

SECURE AND RESILIENT
COMPUTING FOR INDUSTRY AND HUMAN DOMAINS

Volume 2
SECURE AND RESILIENT
SYSTEMS, NETWORKS AND INFRASTRUCTURES
Edited by Vyacheslav Kharchenko

Secure and Resilient Industrial Control
Systems

Computer Systems and Networks
Security and Resilience

Security and Resilience of Web- and
Cloud-Systems

Risk Analysis of Systems of Systems
Security and Resilience

Human-Machine Engineering for Security
Critical and Resilient Systems

Security Management Systems



Volume 2. SECURE AND RESILIENT SYSTEMS, NETWORKS AND INFRASTRUCTURES



MULTI-
LECTURE
BOOK

SECURE AND RESILIENT
COMPUTING FOR INDUSTRY
AND HUMAN DOMAINS

VOLUME 2
SECURE AND RESILIENT
SYSTEMS
NETWORKS AND INFRASTRUCTURES

2017



Co-funded by the
Tempus Programme
of the European Union



Co-funded by the
Tempus Programme
of the European Union

**Ministry of Education and Science of Ukraine
National Aerospace University n. a. N. E. Zhukovsky
“Kharkiv Aviation Institute”**

**V. Sklyar, V. Kharchenko, E. Babeshko, A. Kovalenko, O. Illiashenko,
O. Rusin, A. Panarin, S. Razgonov, D. Ostapiec, I. Zhukovyts'kyi, S. Stirenko,
O. Tarasyuk, A. Gorbenko, A. Romanovsky, O. Biloborodov, I. Skarha-
Bandurova, E. Brezhniev, A. Stadnik, A. Orekhov, T. Lutskiv, V. Mokhor,
O. Bakalynskiy, A. Zhylin, V. Tsurkan, M. Q. Al-sudani, Yu. Ponochovnyi**

SECURE AND RESILIENT COMPUTING FOR INDUSTRY AND HUMAN DOMAINS.

**Secure and resilient
systems, networks and
infrastructures**

Multi-book, Volume 2

V. S. Kharchenko eds.

**Tempus project
Serein 543968-TEMPUS-1-2013-1-EE-TEMPUS-JPCR
Modernization of Postgraduate Studies on Security and Resilience for
Human and Industry Related Domains**

2017

V. Sklyar, V. Kharchenko, E. Babeshko, A. Kovalenko, O. Illiashenko, O. Rusin, A. Panarin, S. Razgonov, D. Ostapec, I. Zhukovyts'kyy, S. Stirenko, O. Tarasyuk, A. Gorbenko, A. Romanovsky, O. Biloborodov, I. Skarha-Bandurova, E. Brezhniev, A. Stadnik, A. Orekhov, T. Lutskiv, V. Mokhor, O. Bakalynskiy, A. Zhylin, V. Tsurkan, M. Q. Al-sudani, Yu. Ponochovnyi. **Secure and resilient computing for industry and human domains. Volume 2. Secure and resilient systems, networks and infrastructures** / Edited by Kharchenko V. S. – Department of Education and Science of Ukraine, National Aerospace University named after N. E. Zhukovskiy “KhAI”, 2017.

Reviewers:

Dr. Peter Popov, Centre for Software Reliability, School of Informatics, City University of London

Prof. Stefano Russo, Consorzio Interuniversitario Nazionale per l'Informatica (Naples, Italy)

Prof. Todor Tagarev, Centre for Security and Defence Management, Institute of Information and Communication Technologies of the Bulgarian Academy of Sciences;

Prof. Jüri Vain, School of Information Technologies, Department of Software Tallinn University of Technology

The second volume of the three volume book called “Secure and resilient computing for industry and human domains” contains materials of the lecture parts of the study modules for MSc and PhD level of education as well as lecture part of in-service training modules developed in the framework of the SEREIN project "Modernization of Postgraduate Studies on Security Resilience for Human and Industry Related Domains"¹ (543968-TEMPUS-1-2013-1-EE-TEMPUS-JPCR) funded under the Tempus programme are given. The book material covers fundamentals issue of secure and resilient computing, in particular, description of related standards, methods of cryptography, software security assurance and post-quantum computing methods review.

The descriptions of trainings, which are intended for studying with technologies and means of assessing security guarantees, are given in accordance with international standards and requirements. Courses syllabuses and description of practicums are placed in the correspondent notes on practicums and in-service training modules.

Designed for engineers who are currently or tend to design, develop and implement information security systems, for verification teams and professionals in the field of quality assessment and assurance of cyber security of IT systems, for masters and PhD students from universities that study in the areas of information security, computer science, computer and software engineering, as well as for lecturers of the corresponding courses.

The materials in the book are given in a form “as is”, desktop publishing of this book is available in hard copy only.

© V. Sklyar, V. Kharchenko, E. Babeshko, A. Kovalenko, O. Illiashenko, O. Rusin, A. Panarin, S. Razgonov, D. Ostapec, I. Zhukovyts'kyy, S. Stirenko, O. Tarasyuk, A. Gorbenko, A. Romanovsky, O. Biloborodov, I. Skarha-Bandurova, E. Brezhniev, A. Stadnik, A. Orekhov, T. Lutskiv, V. Mokhor, O. Bakalynskiy, A. Zhylin, V. Tsurkan, M. Q. Al-sudani, Yu. Ponochovnyi. 2017

This work is subject to copyright. All rights are reserved by the authors, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms, or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

¹ *This project has been funded with support from the European Commission. This publication (communication) reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.*

15 METHODS OF INDUSTRIAL CONTROL SYSTEMS SECURITY ASSESSMENT

15.1 Industrial Control Systems security: a problem statement

This module provides materials concerning security assessment of industrial control systems (ICS). Typical ICS includes supervisory control and data acquisition (SCADA) systems networked with distributed control systems (DCS). DCSs and other control systems are usually based on Programmable Logic Controllers (PLC).

ICSs are typically used in industries such as electric, water and wastewater, oil and gas, transportation, chemical, pharmaceutical, food and beverage, and discrete manufacturing (e.g., automotive, aerospace, etc.) SCADA systems are generally used to control dispersed assets. DCS are generally used to control production systems within a local area such as a factory using control [1].

ICSs consist of combinations of control components (e.g., electrical, mechanical, hydraulic, pneumatic) that operate together to achieve an industrial objective such as manufacturing and transportation. The control part of the system includes the specification of the desired outputs or performance. Control can be fully automated or may include a human in the loop. Systems can be configured to operate open-loop, closed-loop, and manual mode. In open-loop control systems the output is controlled by established settings. In closed-loop control systems, the output has an effect on the input in such a way as to maintain the desired objective. In manual mode the system is controlled completely by humans. A typical ICS may contain numerous control loops, Human Machine Interfaces (HMI), and remote diagnostics and maintenance tools built using an array of network protocols. Some critical processes may also include safety systems.

The basic structure of an ICS with key components is shown in Fig. 15.1.

A control loop utilizes sensors, actuators, and controllers (PLCs) to manipulate some controlled process. A sensor is a device that produces a measurement of some physical property and then sends this information as controlled variables to the controller.

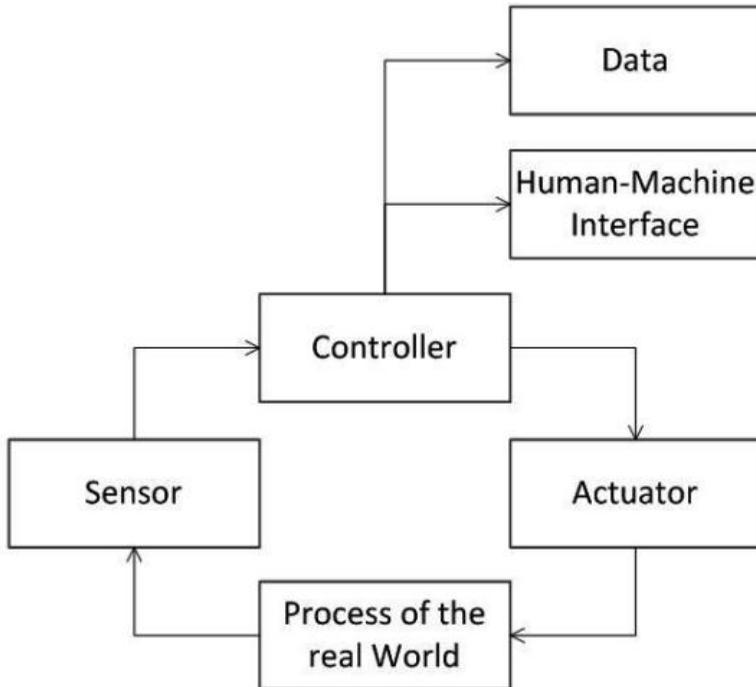


Fig. 15.1 – Main components of Industrial Control Systems

The controller interprets the signals and generates corresponding manipulated variables, based on a control algorithm and target set points, which it transmits to the actuators. Actuators such as control valves, breakers, switches, and motors are used to directly manipulate the controlled process based on commands from the controller.

Operators and engineers use HMI to monitor operation and configure set points, control algorithms, and to adjust and establish parameters in the controller. The HMI also displays process status information and historical information. Diagnostics and maintenance utilities are used to prevent, identify, and recover from abnormal operation or failures.

While control systems used in manufacturing and distribution industries are very similar in operation, they are different in some aspects. Manufacturing industries are usually located within a plant-centric area of a factory or, when compared to geographically dispersed

distribution industries. Communications in manufacturing industries are usually performed using local area network (LAN) technologies that are typically more reliable and high speed as compared to the long-distance communication wide area networks (WAN) used by distribution industries. The ICS used in distribution industries are designed to handle long-distance communication challenges such as delays and data loss posed by the various communication media used. The security controls may differ among network types.

Typical SCADA hardware (see Fig. 15.2) includes a control server placed as the Main Terminal Unit (MTU) at a control center, communications equipment (e.g., radio, telephone line, cable, or satellite), and one or more geographically distributed field sites consisting of Remote Terminal Units (RTUs) and/or PLCs, which controls actuators and/or monitors sensors. The control server stores and processes the information from RTU inputs and outputs, while the RTU or PLC controls the local process. The communications hardware allows the transfer of information and data back and forth between the control server and the RTUs or PLCs. An Intelligent Electronic Device (IED), such as a protective relay, may communicate directly to the control server, or a local RTU may poll the IEDs to collect the data and pass it to the control server. IEDs provide a direct interface to control and monitor equipment and sensors [2,3].

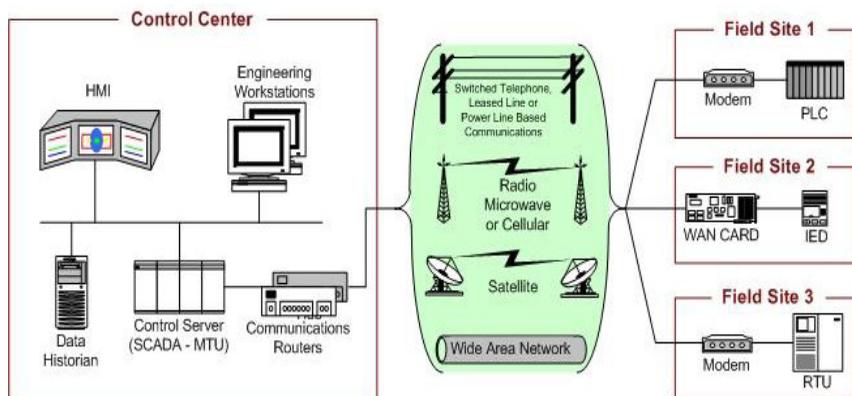


Fig. 15.2 – SCADA System General Layout (source: NIST SP 800-82)

Fig. 15.3 shows control of a manufacturing process being performed by a PLC over a fieldbus network. The PLC is accessible via a programming interface located on an engineering workstation, and data is stored in a data historian, all connected on a LAN [4,5].

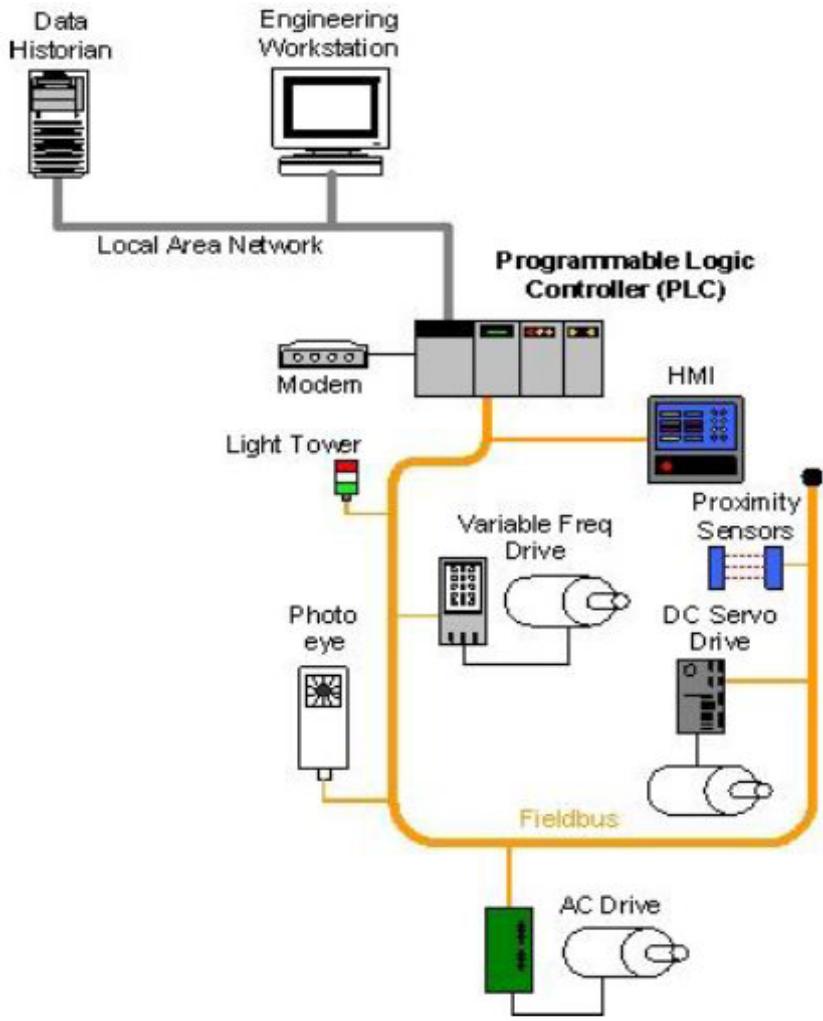


Fig. 15.3 – SCADA System General Layout (source: NIST SP 800-82)

The United States Department of Homeland Security takes into account the following sixteen critical infrastructure sectors, which, probably, are applicable for any of national infrastructure: Chemical, Commercial Facilities, Communications, Critical Manufacturing, Dams, Defense Industrial Base, Emergency Services, Energy, Financial Services, Food and Agriculture, Government Facilities, Healthcare and Public Health, Information Technology, Nuclear Reactors, Materials, and Waste, Transportation Systems, Water and Wastewater Systems.

ICSs have some specific features which make them different from other Information Technologies (IT) systems. ICS control the physical world and IT systems manage data. It raises many ICS characteristics, including different risks and priorities. Some of these include significant risk to the health and safety of human lives, damage to the environment, and financial issues such as production losses, and negative impact to a nation’s economy. ICS have different performance, resilience, safety and reliability requirements, and also use operating systems and applications that may be considered unconventional in a typical IT network environment. Security protections must be implemented in a way that maintains system integrity during normal operations as well as during times of cyber-attack. Table 15.1 provides summary of IT system and ICS differences according to NIST SP 800-82 statements [1].

Table 15.1 – Summary of IT system and ICS differences

Category	Information Technology System	Industrial Control System
Performance Requirements	<p>Non-real-time. Response must be consistent. High throughput is demanded. High delay and jitter may be acceptable. Less critical emergency interaction. Tightly restricted access</p>	<p>Real-time. Response is time-critical. Modest throughput is acceptable. High delay and/or jitter is not acceptable. Response to human and other emergency interaction is critical. Access to ICS should be</p>

Category	Information Technology System	Industrial Control System
	control can be implemented to the degree necessary for security	strictly controlled, but should not hamper or interfere with human-machine interaction
Availability (Reliability) Requirements	Responses such as rebooting are acceptable. Availability deficiencies can often be tolerated, depending on the system's operational requirements	Responses such as rebooting may not be acceptable because of process availability requirements. Availability requirements may necessitate redundant systems. Outages must be planned and scheduled days/weeks in advance. High availability requires exhaustive pre-deployment testing
Risk Management Requirements	Manage data. Data confidentiality and integrity is paramount. Fault tolerance is less important – momentary downtime is not a major risk. Major risk impact is delay of business operations	Control physical world. Human safety is paramount, followed by protection of the process. Fault tolerance is essential, even momentary downtime may not be acceptable. Major risk impacts are regulatory non-compliance, environmental impacts, loss of life, equipment, or production
System Operation	Systems are designed for use with typical operating systems.	Differing and possibly proprietary operating systems, often without

Category	Information Technology System	Industrial Control System
	Upgrades are straightforward with the availability of automated deployment tools	built-in security capabilities. Software changes must be carefully made, usually by software vendors, because of the specialized control algorithms and perhaps modified hardware and software involved
Resource Constraints	Systems are specified with enough resources to support the addition of third-party applications such as security solutions	Systems are designed to support the intended industrial process and may not have enough memory and computing resources to support the addition of security capabilities
Communications	Standard communications protocols. Primarily wired networks with some localized wireless capabilities. Typical IT networking practices	Many proprietary and standard communication protocols. Primarily wired networks with some localized wireless capabilities. Subnetworks including radio wire and wireless (radio and satellite). Networks are complex and sometimes require the expertise of control engineers
Change Management	Software changes are applied in a timely fashion in the presence of good security policy and procedures. The procedures are often	Software changes must be thoroughly tested and deployed incrementally throughout a system to ensure that the integrity of the control system is

Category	Information Technology System	Industrial Control System
	automated	maintained. ICS outages often must be planned and scheduled days/weeks in advance. ICS may use operating systems that are no longer supported
Managed Support	Allow for diversified support styles	Service support is usually via a single vendor
Components Lifetime	Lifetime on the order of 3 to 5 years	Lifetime on the order of 10 to 15 years (for some domains, with opportunity up to 30 years of operation)
Components Location	Components are usually local and easy to access	Components can be isolated, remote, and require extensive physical effort to gain access to them

15.2 Analysis of known Industrial Control Systems treats, vulnerabilities, malware and cyber incidents

15.2.1 ICS threats

A threat is any circumstance or event with the potential to adversely impact organization operations, assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service [3].

Threats have some intent or method that may exploit of vulnerability through either intentional or unintentional means, this intent or method referred to as the threat source.

A vulnerability is a weakness in an information system (including an ICS), system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source.

A threat event is an event or situation that has the potential for causing undesirable consequences or impact. When a threat event occurs it becomes an incident that actually or potentially jeopardizes the confidentiality, integrity, or availability of an information system or the information the system processes, stores, or transmits or that constitutes a violation or imminent threat of violation of security policies, security procedures, or acceptable use policies.

Threats to ICS can come from numerous sources, which can be classified as the following:

- Adversarial threats are caused by individuals, groups, organizations, or states that seek to exploit the organization's dependence on cyber resources (e.g., information in electronic form, information and communications technologies, and the communications and information-handling capabilities provided by those technologies);
- Accidental threats are caused by erroneous actions taken by individuals in the course of executing their everyday responsibilities;
- Structural threats are caused by failures of equipment, environmental controls, or software due to aging, resource depletion, or other circumstances which exceed expected operating parameters;
- Environmental threats are caused by Natural disasters and failures of critical infrastructures on which the organization depends, but which are outside the control of the organization.

It is necessary to create a risk management strategy for the ICS that protects the system against these possible threat sources. The threat source must be well understood in order to define and implement adequate protection.

15.2.2 ICS vulnerabilities

Understanding the source of vulnerabilities and predisposing conditions can assist in determining optimal mitigation strategies. Predisposing conditions are properties of the organization, mission/business process, architecture, or information systems that contribute to the likelihood of a threat event. The groups of vulnerabilities may be the following [3,6]:

- Policy and procedure vulnerabilities; can be considered, for example, such vulnerabilities, as absence of formal ICS security

training and awareness program, inadequate incident detection and response plan, etc.;

- Architecture and design vulnerabilities; can be considered, for example, such vulnerabilities, as non-controlled traffic in security network, no security perimeter defined, etc.;

- Configuration and maintenance vulnerabilities; can be considered, for example, such vulnerabilities, as absence of patch maintenance, inadequate change control and testing of security changes, Denial of Service (DoS), absence of critical configuration backup, poor passwords management, inadequate access controls, inadequate malware protection, etc.;

- Physical vulnerabilities; can be considered, for example, such vulnerabilities, as lack of backup power, physical access of unauthorized personnel, unsecured physical ports, lack of defense against environmental and electromagnetic impacts, etc.;

- Software development vulnerabilities; can be considered, for example, such vulnerabilities, as improper data validation, inadequate authentication and access authorization, etc.;

- Communication and network vulnerabilities; can be considered, for example, such vulnerabilities, as improper firewalls and routers configuration, using of unsecure industry-wide ICS protocols, lack of integrity checking for communications, etc.

15.2.3 ICS security incidents

Possible security incidents for ICS may face include the following:

- Blocked or delayed information through ICS networks, which could disrupt ICS operation;

- Unauthorized changes to instructions, commands, or alarm thresholds, which could damage, disable, or shut down equipment, create environmental impacts, and/or endanger human life;

- Inaccurate information sent to system operators, either to disguise unauthorized changes, or to cause the operators to initiate inappropriate actions, which could have various negative effects;

- ICS software or configuration settings modified, or ICS software infected with malware, which could have various negative effects;

- Interference with the operation of equipment protection systems, which could endanger costly and difficult-to-replace equipment;

– Interference with the operation of safety systems, which could endanger human life.

The first described ICS related cyber security incident happened in 1982. Thomas Reed, senior US national security official, claims in his book “At the Abyss” [7] that the United States allowed the USSR to steal pipeline control software from a Canadian company. This software included a Trojan Horse that caused a major explosion of the Trans-Siberian gas pipeline in June, 1982. The Trojan ran during a pressure test on the pipeline but doubled the usual pressure, causing the explosion. “In order to disrupt the Soviet gas supply, its hard currency earnings from the West, and the internal Russian economy, the pipeline software that was to run the pumps, turbines, and valves was programmed to go haywire, after a decent interval, to reset pump speeds and valve settings to produce pressures far beyond those acceptable to pipeline joints and welds.” The scheme to plant bugs in Soviet software was masterminded by Gus Weiss, who at the time was on the National Security Council and who died last year. Soviet agents had been so keen to acquire US technology, that they didn’t question its provenance. Russian newspaper sources deny the report, saying an explosion did take place, but it was caused by poor construction, not by planted software. “What the Americans have written is rubbish,” said Vasily Pchelintsev, who in 1982 headed the KGB office in the Tyumen region, the likely site of the explosion described in the book.” The software sabotage had two effects, explains Reed. The first was economic. By creating an explosion with the power of a three kiloton nuclear weapon, the US disrupted supplies of gas and consequential foreign currency earnings. But the project also had important psychological advantages in the battle between the two superpowers. “By implication, every cell of the Soviet leviathan might be infected,” Reed writes. “They had no way of knowing which equipment was sound, which was bogus. All was suspect, which was the intended endgame for the entire operation.

At the same time, many researcher conclude, that the above situation could not happen (<http://ogas.kiev.ua/perspective/vzryv-kotorogo-ne-bylo-581>). Firstly, gas transportation system in the USSR was not been equipped with digital control. Secondly, gas pressure increasing was handled by diverse protection system. Thirdly, the

described explosion with the power of a three kiloton is physically impossible in the described conditions.

Any case, this incident is considered in many data bases as the first documented cyber weapon.

NIST SP 800-82 describes the following notorious incidents related with ICSs [3].

Bellingham, Washington Gasoline Pipeline Failure. In June 1999, 900 000 liters (237 000 gallons) of gasoline leaked from a 16 in. (40.64 cm) pipeline and ignited 1.5 hours later causing 3 deaths, 8 injuries, and extensive property damage. The pipeline failure was exacerbated by control systems not able to perform control and monitoring functions. “Immediately prior to and during the incident, the SCADA system exhibited poor performance that inhibited the pipeline controllers from seeing and reacting to the development of an abnormal pipeline operation.” A key recommendation from the NTSB report issued October 2002 was to utilize an off-line development and testing system for implementing and testing changes to the SCADA database.

Maroochy Shire Sewage Spill. In the spring of 2000, a former employee of an Australian organization that develops manufacturing software applied for a job with the local government, but was rejected. Over a two-month period, the disgruntled rejected employee reportedly used a radio transmitter on as many as 46 occasions to remotely break into the controls of a sewage treatment system. He altered electronic data for particular sewerage pumping stations and caused malfunctions in their operations, ultimately releasing about 264 000 gallons of raw sewage into nearby rivers and parks.

CSX Train Signaling System. In August 2003, the Sobig computer virus was blamed for shutting down train signaling systems throughout the east coast of the U.S. The virus infected the computer system at CSX Corp.’s Jacksonville, Florida headquarters, shutting down signaling, dispatching, and other systems. According to Amtrak spokesman Dan Stessel, ten Amtrak trains were affected in the morning. Trains between Pittsburgh and Florence, South Carolina were halted because of dark signals, and one regional Amtrak train from Richmond, Virginia to Washington and New York was delayed for more than two hours. Long-distance trains were also delayed between four and six hours.

Northeast Power Blackout. In August 2003, failure of the alarm processor in First Energy's SCADA system prevented control room operators from having adequate situational awareness of critical operational changes to the electrical grid. Additionally, effective reliability oversight was prevented when the state estimator at the Midwest Independent System Operator failed due to incomplete information on topology changes, preventing contingency analysis. Several key 345 kV transmission lines in Northern Ohio tripped due to contact with trees. This eventually initiated cascading overloads of additional 345 kV and 138 kV lines, leading to an uncontrolled cascading failure of the grid. A total of 61 800 MW load was lost as 508 generating units at 265 power plants tripped.

Davis-Besse nuclear power plant. In August 2003, the Nuclear Regulatory Commission confirmed that in January 2003, the Microsoft SQL Server worm known as Slammer infected a private computer network at the idled Davis-Besse nuclear power plant in Oak Harbor, Ohio, disabling a safety monitoring system for nearly five hours. In addition, the plant's process computer failed, and it took about six hours for it to become available again. Slammer reportedly also affected communications on the control networks of at least five other utilities by propagating so quickly that control system traffic was blocked.

Zotob Worm. In August 2005, a round of Internet worm infections knocked 13 of DaimlerChrysler's U.S. automobile manufacturing plants offline for almost an hour, stranding workers as infected Microsoft Windows systems were patched. Plants in Illinois, Indiana, Wisconsin, Ohio, Delaware, and Michigan were knocked offline. While the worm affected primarily Windows 2000 systems, it also affected some early versions of Windows XP. Symptoms include the repeated shutdown and rebooting of a computer. Zotob and its variations caused computer outages at heavy-equipment maker Caterpillar Inc., aircraft-maker Boeing, and several large U.S. news organizations.

Taum Sauk Water Storage Dam Failure. In December 2005, the Taum Sauk Water Storage Dam suffered a catastrophic failure releasing a billion gallons of water. The failure of the reservoir occurred as the reservoir was being filled to capacity or may have possibly been overtopped. The current working theory is that the reservoir's berm was

overtopped when the routine nightly pump-back operation failed to cease when the reservoir was filled. According to the utility, the gauges at the dam read differently than the gauges at the Osage plant at the Lake of the Ozarks, which monitors and operates the Taum Sauk plant remotely. The stations are linked together using a network of microwave towers, and there are no operators on-site at Taum Sauk.

Browns Ferry-3 PLC Failure. In August 2006, Tennessee Valley Authority was forced to manually shut down one of their plant's two reactors after unresponsive PLCs problems caused two water pumps to fail and threatened the stability of the plant itself. Although there were dual redundant PLCs, they were connected to the same Ethernet network. Later testing on the failed devices discovered that they would crash when they encountered excessive network traffic.

Stuxnet Worm. Stuxnet was a Microsoft Windows computer worm discovered in 2010 that specifically targeted industrial software and equipment. The worm initially spread indiscriminately, but included a highly specialized malware payload that was designed to target only specific SCADA systems that were configured to control and monitor specific industrial processes. Once the machine is infected, Stuxnet looks to see if the computer is running Siemens' Simatic WinCC or PCS 7 software. The malware then automatically uses a default password that is hard-coded into the software to access the control system's Microsoft SQL database. The password has been available on the Internet for several years. An estimated 10,000 machines, mostly in US, Iran, Iraq and Indonesia, reported infections within the first week. Iranian sources confirmed that the Stuxnet malworm shut down uranium enrichment at Natanz for a week from November 16 to 22, 2010. The centrifuge spinning speed was fluctuating without the monitors detecting any malfunction. The International Atomic Energy Agency (IAEA) director, Yukiya Amano, reported the shutdown to the IAEA board in Vienna on Tuesday, November 23, 2010. [8].

Brute Force Attacks on Internet-Facing Control Systems. On February 22, 2013 ICS-CERT received a report from a gas compressor station owner about an increase in brute force attempts to access their process control network. The forensic evidence contained 10 separate IPs and additional calls of a similar nature from additional natural gas pipeline asset owners, which yielded 39 additional IPs of concern. Log

analysis showed a date range from January 16, 2013 but there have been no reports since March 8, 2013.

German Steel Mill Attack. In 2014, hackers manipulated and disrupted control systems to such a degree that a blast furnace could not be properly shut down, resulting in “massive” – though unspecified – damage.

Blackout in Ukrainian power system. Hackers have used highly destructive malware and infected, at least, three regional power authorities, causing blackouts across the Ivano-Frankivsk region of Ukraine on December 23, 2015. Power outages were caused by remote cyber intrusions at three regional electric power distribution companies (Oblenergos) impacting approximately 225,000 customers. While power has been restored, all the impacted Oblenergos continue to run under constrained operations. Over the past year, the group behind BlackEnergy has slowly ramped up its destructive abilities. The KillDisk malware that hits the Ukrainian power companies contained similar functions but was programmed to delete a much narrower set of data. KillDisk had also been updated to sabotage two computer processes, including a remote management platform associated with the ELTIMA Serial to Ethernet Connectors used in industrial control systems.

The USA Industrial Control Systems Cyber Emergency Response Team of National Cybersecurity and Communications Integration Center (NCCIC/ICS-CERT) periodically issues annual reports which provide information concerning ICS vulnerabilities and cyber incidents around the USA [9]. ICS-CERT’s mission is to reduce risk to the Nation’s critical infrastructure by strengthening control systems security and resilience through public-private partnerships. ICS-CERT has been involved in investigation BlackEnergy Malware, which caused power outage in Ukraine at December 23, 2015.

In 2015, ICS-CERT responded to 295 cyber incidents. This represented a 20 percent increase over FY 2014. The Critical Manufacturing Sector nearly doubled to a record 97 incidents, becoming the leading sector for ICS-CERT in FY 2015. The Energy Sector had the second most incidents with 46 incidents, and the Water and Wastewater Systems Sector was third with 25.

In 2015, the ICS-CERT vulnerability coordination team handled 486 vulnerabilities. ICS-CERT reduced the average number of days to

close a ticket from 108 days in 2014 to 55 days in 2015 and closed 76 percent of tickets that had been open over 365 days [9].

Many relevant records concerning security incidents, vulnerabilities and other issues can be found in the following online resources:

- Repository of Industrial Security Incidents (RISI) at the link <http://www.risidata.com/>;

- U.S. National Vulnerability Database supported by NIST at the link <https://nvd.nist.gov/>;

- Alerts of the U.S. Computer Emergency Readiness Team (US-CERT) which provide timely information about current security issues, vulnerabilities, and exploits at the link <https://www.us-cert.gov/ncas/alerts>;

- Newly developed vulnerabilities search engine VULNERS, which integrated search results from many databases at the link <https://vulners.com>.

15.3 Risk management for Industrial Control Systems

A risk management process should be employed throughout an organization, using a three-tiered approach to address risk at the organization level; mission/business process level; and information system level (IT system and ICS). The risk management process is carried out seamlessly across the three tiers with the overall objective of continuous improvement in the organization's risk-related activities and effective inter-tier and intra-tier communication among all stakeholders having a shared interest in the mission/business success of the organization [3,10].

Assessing risk requires that organizations identify their threats and vulnerabilities, the harm that such threats and vulnerabilities may cause the organization and the likelihood that adverse events arising from those threats and vulnerabilities may actually occur.

General risk-assessment concept in relation with security assurance is presented on Fig. 15.1. This concept is received from Security Common Criteria (ISO/IEC 15408, see Section 2.3 of this multi-book). All security entities, such as assets, threats, vulnerabilities, risk, countermeasures and other, are closely related parts of general security framework.

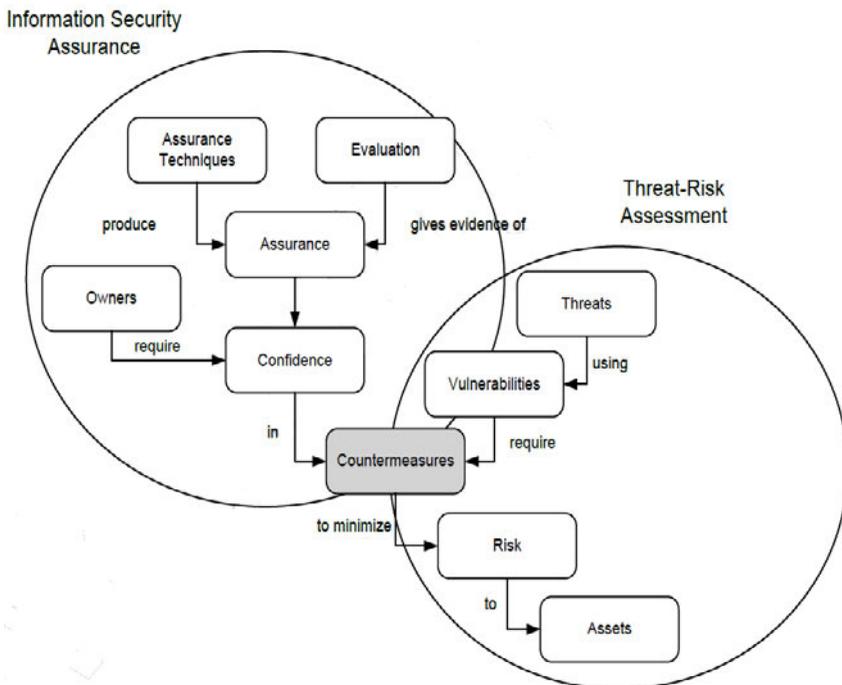


Fig. 15.1 – Risk assessment concept in relation with security assurance (source: ISA/IEC 62443)

Possible impact of ICS on environment, people health, production and other forms potential harms which are related with risks. If evaluate probability of each impact level, risk assessment inputs will be determined (see Table 15-1) [3].

Table 15.1 – Possible risk assessment inputs based on definitions of ICS harm levels

Risk Category	Low harm	Moderate harm	High harm
Injury	Cuts, bruises requiring first aid	Requires hospitalization	Loss of life or limb
Financial loss	\$ 1 000	\$ 100 000	> \$ 1 000 000

Risk Category	Low harm	Moderate harm	High harm
Environmental release	Temporary damage	Lasting damage	Permanent damage, off-site damage
Interruption of production	Minutes	Days	Weeks
Public image	Temporary damage	Lasting damage	Permanent damage
Security concerns	Minor injuries; Ensuring uptime	Moderate injuries; Capital investment	Major injuries/loss of life; Trade secrets; Basic social services losing; Regulatory compliance corruption

The nature of ICS means that when an organization does a risk assessment, there may be additional considerations that do not exist when doing a risk assessment of a traditional IT system. Because the impact of a cyber incident in an ICS may include both physical and digital effects, risk assessments need to incorporate those potential effects. Risk assessment of ICS should include the following specific issues [3,10]:

- Impacts on safety and use of safety assessments;
- Physical impact of a cyber incident on an ICS, including the larger physical environment; effect on the process controlled, and the physical effect on the ICS itself;
- The consequences for risk assessments of non-digital control components within an ICS.

The culture of safety and safety assessments is well established within the majority of the ICS user community. Information security risk assessments should be seen as complementary to such assessments though the assessments may use different approaches and cover different areas. Safety assessments are concerned primarily with the physical world. Information security risk assessments primarily look at

the digital world. However, in an ICS environment, the physical and the digital are intertwined and significant overlap may occur.

Evaluating the potential physical damage from a cyber incident should incorporate: 1) how an incident could manipulate the operation of sensors and actuators to impact the physical environment; 2) what redundant controls exist in the ICS to prevent an impact; and 3) how a physical incident could emerge based on these conditions. A physical impact could negatively impact the surrounding world through multiple means, including the release of hazardous materials (e.g., pollution, crude oil), damaging kinetic forces (e.g., explosions), and exposure to energy sources (e.g., electricity, steam). The physical incident could negatively impact the ICS and supporting infrastructure, the various processes performed by the ICS, or the larger physical environment. An evaluation of the potential physical impacts should include all parts of an ICS, beginning with evaluating the potential impacts on the set of sensor and actuators. Each of these domains will be further explored below.

Evaluating the impact of a cyber incident on the physical environment should focus on potential damage to human safety, the natural environment, and other critical infrastructures. Human safety impacts should be evaluated based on whether injury, disease, or death is possible from a malfunction of the ICS. This should incorporate any previously performed safety impact assessments performed by the organization regarding both employees and the general public. Environmental impacts also may need to be addressed. This analysis should incorporate any available environmental impact assessments performed by the organization to determine how an incident could impact natural resources and wildlife over the short or long term. In addition, it should be noted that ICS may not be located within a single, controlled location and can be distributed over a wide physical area and exposed to uncontrolled environments. Finally, the impact on the physical environment should explore the extent to which an incident could damage infrastructures external to the ICS (e.g., electric generation/delivery, transportation infrastructures, and water services).

In addition to the impact on the physical environment, the risk assessment should also evaluate potential effects to the physical process performed by the ICS under consideration, as well as other systems. An incident that impacts the ICS and disrupts the dependent process may

cause cascading impacts into other related ICS processes and the general public's dependence on the resulting products and services. Impact to related ICS processes could include both systems and processes within the organization (e.g., a manufacturing process that depends on the process controlled by the system under consideration) or systems and processes external to the organization (e.g., a utility selling generated energy to a nearby plant).

The impacts on the ICS cannot be adequately determined by focusing only on the digital aspects of the system, as there are often non-digital mechanisms available that provide fault tolerance and prevent the ICS from acting outside of acceptable parameters. Therefore, these mechanisms may help reduce any negative impact that a digital incident on the ICS might have and must be incorporated into the risk assessment process. For example, ICS often have non-digital control mechanisms that can prevent the ICS from operating outside of a safe boundary, and thereby limit the impact of an attack (e.g., a mechanical relief pressure valve). In addition, analog mechanisms (e.g., meters, alarms) can be used to observe the physical system state to provide operators with reliable data if digital readings are unavailable or corrupted.

Safety systems may also reduce the impact of a cyber incident to the ICS. Safety systems are often deployed to perform specific monitoring and control functions to ensure the safety of people, the environment, process, and ICS. While these systems are traditionally implemented to be fully redundant with respect to the primary ICS, they may not provide complete redundancy from cyber incidents, specifically from a sophisticated attacker. The impact of the implemented security controls on the safety system should be evaluated to determine that they do not negatively impact the system.

Evaluating the impact of an incident must also incorporate how the impact from the ICS could propagate to a connected ICS or physical system. An ICS may be interconnected with other systems, such that failures in one system or process can easily cascade to other systems either within or external to the organization. Impact propagation could occur due to both physical and logical dependencies. Proper communication of the results of risk assessments to the operators of connected or interdependent systems and processes is one way to mitigate such impacts.

15.4 Intrusion Modes and Effect Criticality Analysis (IMECA)

IMECA implementation can be started from as named GAP-analysis in security practices and requirements implementation.

Main principle in the security assessment is the use of process-product approach consisting in determination of the possible discrepancies in the final product and development process. One of the fundamental concepts behind the idea of the approach is the concept of GAP, which is determined as a set of discrepancies of any single process within the lifecycle [11,12] of ICS that can introduce some anomalies (e.g. vulnerabilities) in a product and/or cannot reveal (and eliminate) existing anomalies in a product.

To perform GAP-analysis the special taxonomy of notions was developed. The taxonomy covers the notions of process, product, intrusion, discrepancy, gap, anomaly, vulnerability, attack and threat, taking into account that vulnerabilities lie dormant until the right circumstances arise (in this case when under “the right circumstances” the usage of vulnerability, i.e. the successful attack, is meant). Main notions in Fig. 15-2 are process, product and threat. Processes are being implemented through the development stages of I&C system lifecycle model in order to produce products. Also, products can be vulnerable to intrusions of various types that can affect the product. Results of implementation of the processes can have effects on possible consequential changes in such processes. Each process comprises activities, and, in a case of “non-ideal” process, some of them can contain discrepancies. A segment of such discrepancies (related to the use of inappropriate tool or introduced by human, or due to shortcoming of technique, etc.) is GAP.

Depending on ICS under consideration, each GAP should be represented in a form of a formal description which determine all discrepancies (between “ideal”, i.e. described in requirements case and the real one). Such formal description should be made for a set of discrepancies identified within the GAP [13,14].

For the formal description for GAP and its further analysis the special technique IMECA is proposed to use. The IMECA analysis is a refinement of FMECA-analysis (failure modes, effects and criticality

analysis) which takes into account intrusions to the system and could be applied to security informed safety approach.

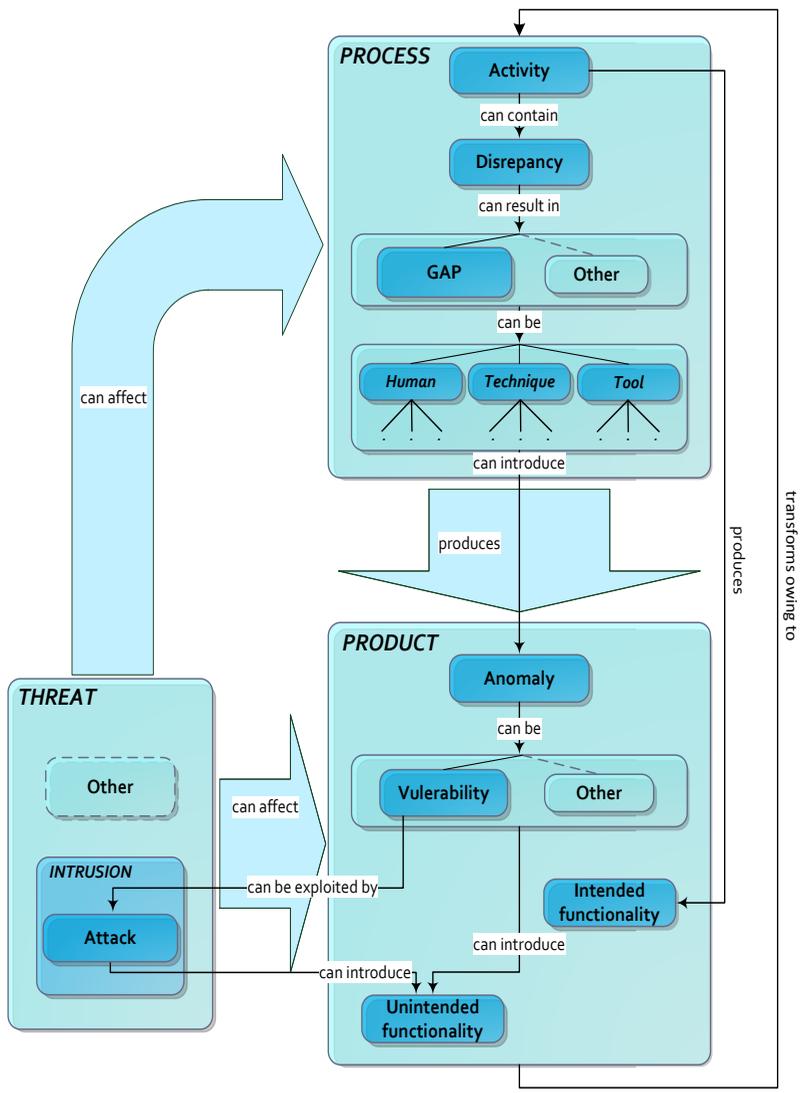


Fig. 15.2 – Taxonomy of Notions for GAP-analysis

Each identified GAP should be represented by a single local IMECA table and each discrepancy inside the GAP can be represented by a single row in that local IMECA table taking into consideration process-product features of the ICS and programmable components. For each GAP, a separate table that contains all the vulnerabilities identified in the GAP analysis is created. All separated tables are combined into general IMECA table.

During the assessment of ICS systems, IMECA can be used in addition to standardized FMECA for safety-related domains, because each vulnerability can become a failure in a case of intrusion into such systems.

During the performance of GAP analysis, the identification of discrepancies (and the corresponding vulnerabilities in case of cyber security assessment), can be implemented via separate detection/analysis of problems caused by human factors, techniques and tools, taking into account the influence of the development environment. Then, after all identified vulnerabilities are prioritized, it is possible to assure security of I&C system by implementing of appropriate countermeasures.

Criticality matrix is depicted on Fig. 15-3 (worst-case criticality diagonal for the matrix; acceptable values of risks are below the diagonal).

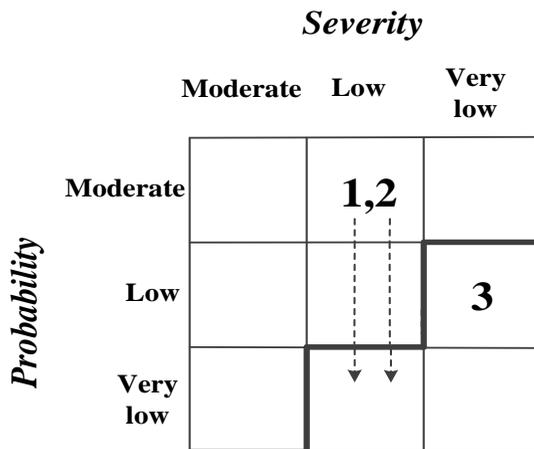


Fig. 15.3 – Criticality Matrix

Each of the numbers inside the matrix represents an appropriate row number of IMECA table. In any case, probability and frequency of successful attacks could change over time depending on the evolution of methods, increase of knowledge about the control and protection system, and other causes. Therefore security measures have a much shorter life time than safety measures and need unfortunately more frequent updates. From security assurance point of view, the possible way of risk reduction is in decreasing of attacks' occurrence probability, since related damage is constant. Cases of probability, decreasing for rows 1, 2, and 3 are denoted by dotted lines with arrows: the problem is in decreasing of the probability by the degree sufficient to move row of IMECA table below the criticality diagonal. Such decreasing of the probability can be achieved, for example, by implementation of certain process countermeasures.

To decrease the risk of manual errors, the tool for the SIS-oriented assessment automation is described. The tool is based on joint use of abovementioned models and techniques, is proposed. The tool allows conducting the joint use of the following analysis techniques: GAP and IMECA. The block-scheme of main stages of analysis is shown on Fig. 15-4.

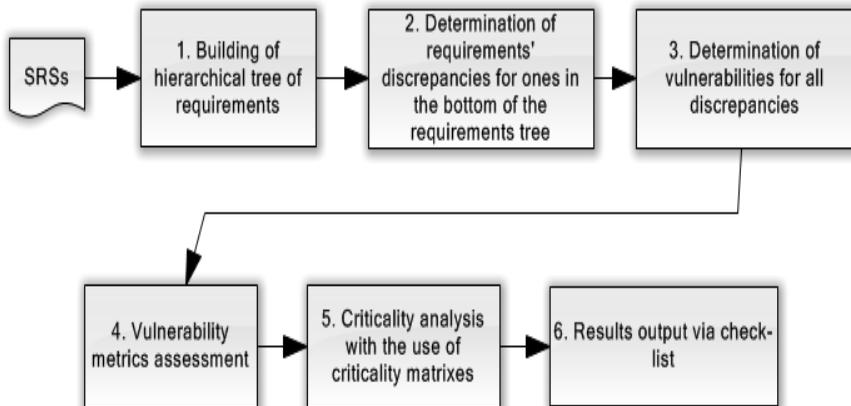


Fig. 15.4 – Workflow of IMECA performance

The ideal system is represented by requirements profile, which contains all elements of the system of process on the different levels of decomposition.

Input data is requirements profile. Requirements could be divided into different levels hierarchically. After determination of quantity of requirements levels the list of requirements for each level is composed. Levels of requirements are filled alternately from top to bottom. When filling one level, for each requirement of the current level the requirements on the lower level, which expand, clarify or detail it, are created. As a result, the requirement at one level can meet one or more requirements of the level below (Step 1 and Step 2).

After input of requirements their analysis at the lowest level is conducted. It is assumed that requirement could be violated, i.e. GAP is introduced artificially and detailed further. During the analysis of the requirement, the specific violations that may possibly occur depending on the nature of requirement are pointed out. In such way GAP is represented as a set of violations of a certain requirement, which could take place in the critical ICS under consideration. At this stage the IMECA-tables are formed for each discrepancy (Step 3, Step 4). It could also be defined more options, which could be determined by expert assessment or additional methods of analysis. One of the required parameters is the likelihood and critical impact on the system. The additional parameters also could be defined with the help of expert assessment or with the use of additional methods of analysis.

Above the parameters under assessment are the likelihood and impact on the criticality of the system. Quantitative parameters can be determined by peer review or other auxiliary tools and techniques.

For each GAP, a separate table that contains all the vulnerabilities identified in the GAP analysis is created. Each of the vulnerabilities is determined by the criticality matrix. With the help of criticality matrix on the basis of vulnerability parameters the metric should be calculated and resulting conclusion for vulnerability shall be made. For the criticality matrix the set of valid parameters is defined.

If any of the parameters of the vulnerability are not included in the allowed range, a decision that the vulnerability is present in the system and requires fixing is made (Step 5).

The presence of discrepancy is determined on the basis of criticality matrix. Check-list is formed from the requirements and a

conclusion about their implementation (Step 6). Example of check-list is shown on Fig. 15-5.

Requirements and GAPS		Discrepancy? Yes/No
Req1	Req11	YES
	Req12	NO
Req2	Req21	NO
	Req22	NO
Req_i	Req_i1	NO
	Req_im	NO

Requirements and GAPS		Discrepancy? Yes/No
Req1		YES
		NO
Req2		NO
		NO
Req_i		NO
		NO

Fig. 15.5 – Check-List for Requirements Testing

Отсылка на 11 раздел в плане разработки tool для GAP-IMECA оценки

Conclusions

Typical ICS includes supervisory control and data acquisition (SCADA) systems networked with distributed control systems (DCS). DCSs and other control systems are usually based on Programmable Logic Controllers (PLC).

Summary of IT system and ICS differences includes the following:

- Performance Requirements;
- Availability (Reliability) Requirements;
- Risk Management Requirements;
- System Operation;
- Managed Support;
- Components Lifetime;
- Components Location.

A threat is any circumstance or event with the potential to adversely impact organization operations, assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service.

Threats have some intent or method that may exploit of vulnerability through either intentional or unintentional means, this intent or method referred to as the threat source.

A vulnerability is a weakness in an information system (including an ICS), system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source.

ICS threats, vulnerabilities, possible and happened incidence are considered in this section.

ICS risk management process is considered. General risk-assessment concept is received from Security Common Criteria in accordance with ISO/IEC 15408 “Information technology – Security techniques – Evaluation criteria for IT security”. ICS risk categories include injury, financial loss, environmental release, interruption of production, public image, and security concerns.

GAP-analysis is determined as finding of a set of discrepancies of any single process within the lifecycle of ICS that can introduce some anomalies (e.g. vulnerabilities) in a product and/or cannot reveal (and eliminate) existing anomalies in a product. So such analysis can discover GAP in requirements compliance.

The Intrusion Modes and Effect Criticality Analysis (IMECA) is a refinement of FMECA which takes into account intrusions to the system and could be applied to security informed safety approach.

Assurance of dependability, security and safety of critical ICS must be done with a special care, because of their development under strict constraints related to resources and cost. It should be done iteratively, rather than the disposable decision.

Questions to self-checking

1. Describe the main components of Industrial Control Systems (ICS).
2. Which are the main differences between ICS and typical IT system?
3. What are relations between security threats, vulnerabilities, risks, and countermeasures?
4. Which are typical ICS threats?
5. Which are typical ICS vulnerabilities?
6. Which are probable security incidents that can occur in ICS?

7. Give some examples of known ICS security incidents.
8. Which risks are possible in ICS?
9. Which are the main features for ICS risk management and assessment?
10. Describe the main issues of GAP-analysis.
11. Describe the main issues of IMECA.
12. Which are the main steps of IMECA?

References

1. NIST SP 800-82 Revision 2, Guide to Industrial Control Systems (ICS) Security: Supervisory Control and Data Acquisition (SCADA) Systems, Distributed Control Systems (DCS), and Other Control System Configurations such as Programmable Logic Controllers (PLC). – National Institute of Standards and Technologies, 2015. – 247 p.
2. E. Knapp. Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems. – Syngress, 2011. – 360 p.
3. A. Sajid, H. Abbas, K. Saleem. Cloud-Assisted IoT-Based SCADA Systems Security: A Review of the State of the Art and Future Challenges // IEEE Access. – Volume 4. – 2016. – P. 1375-1384.
4. D. Pal, J. Vain. Generating optimal test cases for real-time systems using DIVINE model checker // 2016 15th Biennial Baltic Electronics Conference (BEC). – P. 99-102.
5. O. Netkachov, P. Popov, K. Salako. Model-Based Evaluation of the Resilience of Critical Infrastructures Under Cyber Attacks // Proceeding of 9th International Conference (CRITIS 2014). – P. 231-243.
6. Common Cybersecurity Vulnerabilities in Industrial Control Systems. – U.S. Department of Homeland Security, 2011. – 76 p.
7. T. Reed. At the Abyss: An Insider's History of the Cold War. – Ballantine Books, 2004.
8. K. Zetter. Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon. Crown Publishers, 2014. – 433 p.
9. National Cybersecurity and Communications Integration Center / Industrial Control Systems Cyber Emergency Response Team

(NCCIC/ICS-CERT). 2015 Year in Review. – U.S. Department of Homeland Security, 2016. – 22 p.

10. NIST SP 800-39, Managing Information Security Risk: Organization, Mission, and Information System View. – National Institute of Standards and Technologies, 2011. – 88 p.

11. V. Sklyar. Cyber Security of Safety-Critical Infrastructures: Case Study for Nuclear Facilities // Information & Security. – Vol. 21, No.1, 2012. – P. 98-107.

12. V. Kharchenko S., A. Andrashov A., A. Kovalenko A., O. Siora, V. Sklyar. Gap-and-IMECA-Based Assessment of I&C Systems Cyber Security // Complex Systems and Dependability. – Springer-Verlag, 2012. – Advances in intelligent and soft computing. – P.149-164.

13. V. Kharchenko, O. Iliashenko, A. Kovalenko, V. Sklyar, A. Boyarchuk. Security Informed Safety Assessment of NPP I&C Systems: GAP-IMECA Technique and Security // Proceedings of 22nd International Conference on Nuclear Engineering ‘ICONE2014’.

14. V. Kharchenko, A. Kovalenko, V. Sklyar, O. Siora. Security Assessment of FPGA-based Safety-Critical Systems: US NRC Requirements Context // Proceedings of the International Conference on Information and Digital Technologies (IDT 2015). – P. 117-123.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ К РАЗДЕЛУ 2

DCS – Distributed Control Systems

ICS – Industrial Control Systems

ICS-CERT – Industrial Control Systems Cyber Emergency

Response Team

IEC – International Electrotechnical Commission

IED – Intelligent Electronic Device

IEEE – Institute of Electrical and Electronics Engineers

IMECA – Intrusion Modes and Effect Criticality Analysis

DoS – Denial of Service

IT – Information Technologies

ISA – International Society of Automation

ISO – International Standardization Organization

HMI – Human Machine Interfaces

LAN – Local Area Network

NIST – National Institute of Standards and Technology

NIST SP – NIST Special Publication

PLC – Programmable Logic Controllers

SCADA – Supervisory Control and Data Acquisition

WAN – Wide Area Networks

АННОТАЦИЯ

В разделе рассмотрены особенности SCADA систем. Проанализированы отличия АСУ ТП от типовых информационных систем. Выполнен анализ угроз, уязвимостей и возможных нарушений информационной безопасности АСУ ТП. Рассмотрены произошедшие за последние 20 лет инциденты, связанные с информационной безопасностью АСУ ТП. Рассмотрен подход к оцениванию и управлению рисками. Дана характеристика методу оценивания информационной безопасности на основе IMECA (Intrusion Modes and Effect Criticality Analysis).

У розділі розглянуто особливості SCADA систем. Проаналізовано відмінності АСУ ТП від типових інформаційних систем. Виконано аналіз загроз, вразливостей та можливих порушень інформаційної безпеки АСУ ТП. Розглянуто інциденти, що пов'язані з інформаційною безпекою АСУ ТП, які відбулися за останні 20 років. Розглянуто підхід щодо оцінювання та управління ризиками. Надано характеристику методу оцінювання інформаційної безпеки на основі IMECA (Intrusion Modes and Effect Criticality Analysis).

SCADA systems features are discussed in the section. Differences between Industrial Control Systems (ICS) and IT systems are analyzed. Analysis of threats, vulnerabilities and possible incidents is performed for ICS. ICS security incidents for the last 20 years are considered. Risk management and assessment analysis is considered. Information security assessment method based on IMECA (Intrusion Modes and Effect Criticality Analysis) is presented.

16 METHODS OF INDUSTRIAL CONTROL SYSTEMS SECURITY AND RESILIENCE ASSURANCE

16.1 Security Concept of Industrial Control Systems

Result of many standards considering allows representing existing security requirements to Industrial Control Systems (ICS) related with a restricted set of categories (see Fig. 16.1).

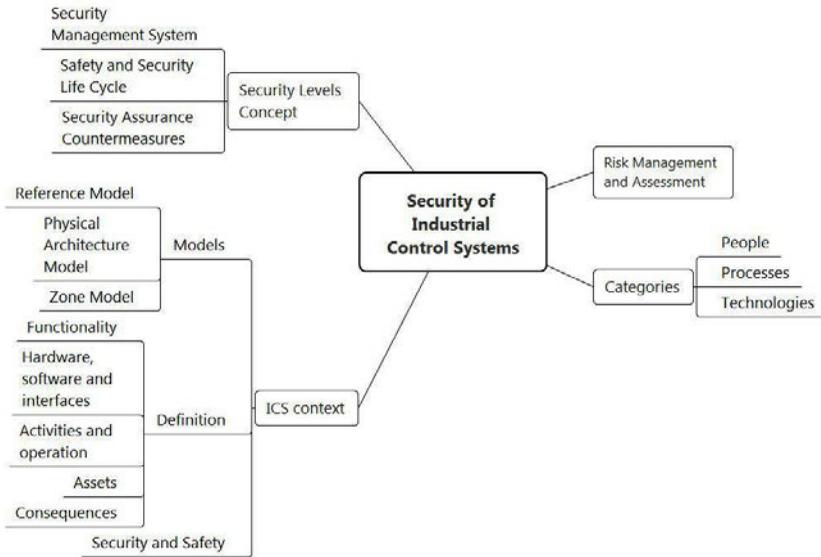


Fig. 16.1 – Security concepts and requirements taxonomy

This conceptual security requirements taxonomy include four the main parts:

- Risk management and assessment as a corner stone for definition of acceptable risks levels and countermeasures for risks reduction (see Section 15.3);

- Categories of security features implementation which include triad “People – Process – Technologies” (see Section 16.2);

– ICS context which drive to define requirement taking into account specifics of ICS; this concept includes three types of models (reference, physical architecture and zone models) as well as functionality, components, assets and other definitions (see Section 16.3), and security and safety coordination issues (see Section 16.4);

– ICS security levels (see Section 16.5) concept which grades risk levels for ICS separated parts and establishes different life cycle processes (see Section 16.6) and countermeasures (see Section 16.7) for different security levels.

The above parts are described below excepted Risk Management Assessment described in Section 15.3 of this multi-book.

The following sections use statements of ISA/IEC 62443 “Security for Industrial Automation and Control Systems” (see Section 2.4 of this multi-book) and NIST SP 800-82 [1].

16.2 Security and resilience assurance based on “People – Process – Technologies” triad

This approach is specified in ISA/IEC 62443 standards series.

The core and foundational principle of the ICS Information Security Management System (ISMS) is the “People – Process – Technologies” categories triad (see Fig. 16.1) [2,3].

Specific recommendations for included in “People” category contain the following:

– Resourcing: It is essential to have the appropriate staffing levels and time commitment to perform the tasks associated with the ICS ISMS (e.g. log reviews, patching, risk assessment, etc.);

– Roles and Responsibilities: Define who owns the process, who supports the process, and the respective responsibilities. There is a commonly used method for assigning those Responsible, Accountable, Consulted, and Informed (RACI) for each task of the process;

– Relationships: Make a concerted effort to break down the traditional relationship barriers between the Control and Business IT groups at all management levels of the organization. The goal is to have cooperative relationships across the different functional areas of the company, and organizational levels. This also applies to the relationship

between the Asset Owner and the Vendor or Integrator responsible for installing and maintaining the ICS;

- Intent, Buy-In, and Support: Ensure that all personnel have the intent and motivation to uphold cyber security policies, practices, and ensure continual improvement. People must be entirely supportive of the security program;

- Training and Capability: Ensure that personnel are adequately qualified to perform the duties associated with ICS security, and that new capabilities are developed where they may not have existed in the organization before (e.g. risk analysis, security intelligence, vulnerability management);

- Awareness and Influenced Decision Making: Ensure that personnel have sufficient awareness and understanding of security policies and security processes as it will influence their decision making and voluntary use of these processes.

“Processes” category may be implemented in several ways, but are built upon smaller hierarchical components of documentation, such as following:

- Policy: This is a formal, brief, and high-level statement or plan that embraces an organization's general beliefs, goals, objectives, and acceptable procedures for a specified subject area;

- Standard: This is a formal document that establishes mandatory requirements, engineering, technical criteria, methods, etc. A standard is meant to convey a mandatory action or rule and is written in conjunction with a policy;

- Process: A process typically describes the act of taking something through an established and usually routine set of procedures or steps to convert it from one form to another, such as processing paperwork to grant physical or cyber access, or converting computer data from one form to another;

- Guideline: These are not required as part of a policy framework, but they can play an important role in conveying best practice information to the user community. Guidelines are meant to “guide” users to adopt behaviors which increase the security posture of a network, but are not yet required (or in some cases, may never be required).

Definition of the specific structure and content of policies, standards, processes and guidelines is the responsibility of the ICS asset

owner. The ISA/IEC 62443 addresses the following subjects in this category:

- Security Policy;
- Organization of Security;
- Asset Management;
- Human Resources Security;
- Physical and Environmental Security;
- Communications and Operations Management;
- Access Control;
- Systems acquisition, development and maintenance;
- Incident Management;
- Business Continuity Management;
- Compliance.

For a traditional information system, requirements and details on each of these subjects can be found in ISO/IEC 27002 “Code of Practice for Information Security Management” (see Section 2.2 of this multi-book). For an ICS ISMS, the unique requirements and enhancements can be found in ISA/IEC 62443-2-1 “Requirements for an IACS Security Management System”.

“Technology” category includes all of the technical security capabilities and controls in place to ensure the availability, integrity, and confidentiality of the ICS. This includes solutions for authentication, access control, encryption, as those technical measures are applied to reduce the security risks to the ICS (see Section 16.7).

The objective of technology relative to ICS is to ensure that security risks are reduced and security-related business processes could be automated where feasible.

16.3 Industrial Control Systems models and definitions

The basis for identifying the security needs and important characteristics of the environment at a level of details necessary to address security issues can be expressed with three models (see Fig. 16.1), each of which is described below [4,5].

A reference model establishes a frame of reference for the more detailed information that follows. It describes a generic view of an integrated manufacturing or production system, expressed as a series of

logical levels. The reference model used by the ISA/IEC 62443 series of standards appears in Fig. 16.2, including the following levels:

– Level 4 (Enterprise Business Systems): This level includes the functions involved in the business-related activities needed to manage a manufacturing organization. For the purposes of this standard, engineering systems are also considered to be in this level;

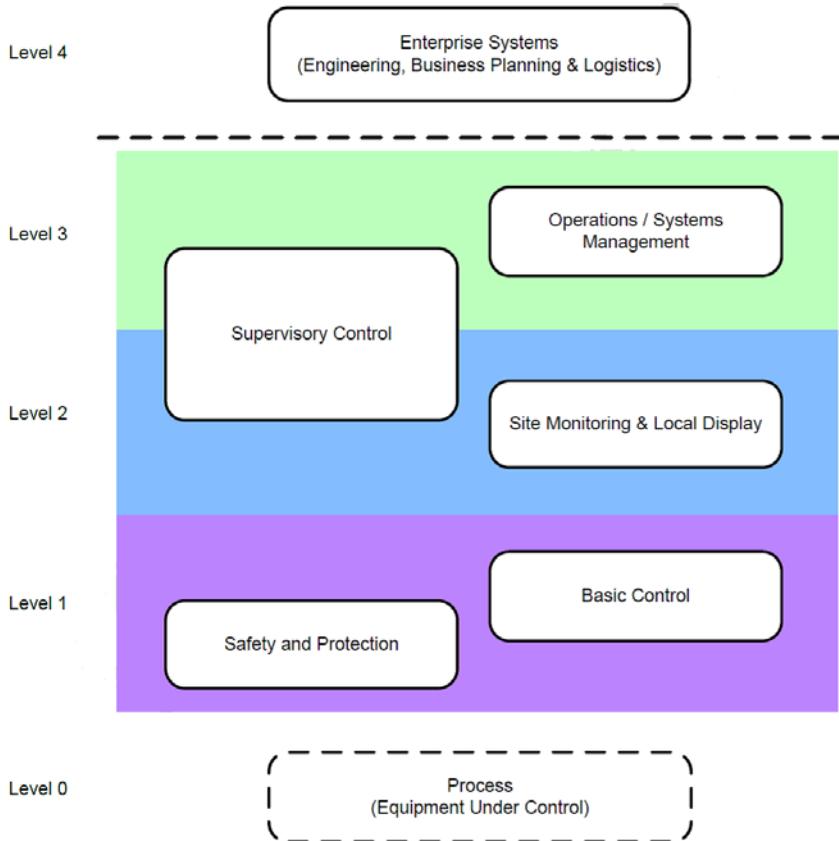


Fig. 16.2 – Reference model of Industrial Control Systems
(source: ISA/IEC 62443)

– Level 3 (Operations Management): This level includes the functions involved in managing the work flows to produce the desired

end products. Examples include dispatching production, detailed production scheduling, reliability assurance, and site-wide control optimization;

- Level 2 (Supervisory Control): This level includes the functions involved in monitoring and controlling the physical process. There are typically multiple production areas in a plant or facility;

- Level 1 (Local or Basic Control): This level includes the functions involved in sensing and manipulating the physical process. Process monitoring equipment reads data from sensors, executes algorithms if necessary, and maintains process history. It includes continuous control, sequence control, batch control, and discrete control. Equipment at this level includes, but is not limited to DCS and PLC. Also included in Level 1 are safety and protection systems that monitor the process and automatically return the process to a safe state if it exceeds safe limits. This category also includes systems that monitor the process and alert an operator of impending unsafe conditions. Safety and protection systems often have additional safety requirements that may not be consistent or relevant to cyber security requirements;

- Level 0 (Process): This level is the actual physical process, which includes a number of different types of production facilities in all sectors including, but not limited to, discrete parts manufacturing, hydrocarbon processing, product distribution, pharmaceuticals, pulp and paper, and electric power. It includes the sensors and actuators directly connected to the process and process equipment.

A physical architecture model is used to describe the various operational components and how they are connected. The details are specific to each individual system under consideration. It is common for an organization to have a single generic model that has been generalized to cover all operating facilities. An example of a simplified reference architecture model for a manufacturing function is shown in Fig. 16.3.

A zone model is derived from the physical architecture model. The assets are grouped into entities (e.g., business, facility, site, or ICS) that are then analyzed for security policies and hence requirements. Fig. 16.4 is an example of a zone model. This model provides the context for assessing common threats, vulnerabilities, and the corresponding countermeasures needed to attain the level of security

required to protect the grouped assets. After grouping assets in this manner, a security policy is defined for all assets that are members of the zone. The results of this analysis are used to determine the appropriate protection required based on the activities performed in the zone.

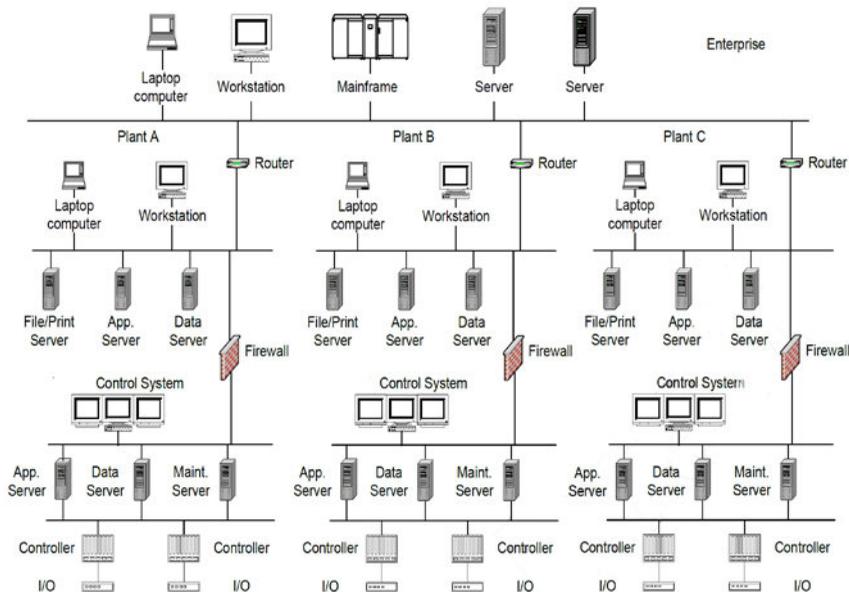


Fig. 16.3 – Physical architecture model of Industrial Control Systems (source: ISA/IEC 62443)

Every situation has a different acceptable level of security. For large or complex systems, it may not be practical or necessary to apply the same level of security to all components. Differences can be addressed by using the concept of a zone, defined as a logical or physical grouping of physical, informational, and application as sets sharing common security requirements. This concept can be applied in an exclusive manner where some systems are included in the security zone and all others are outside the zone. A conduit is a particular type of zone that groups communications that can be logically organized into a grouping of information flows within and also external to a zone.

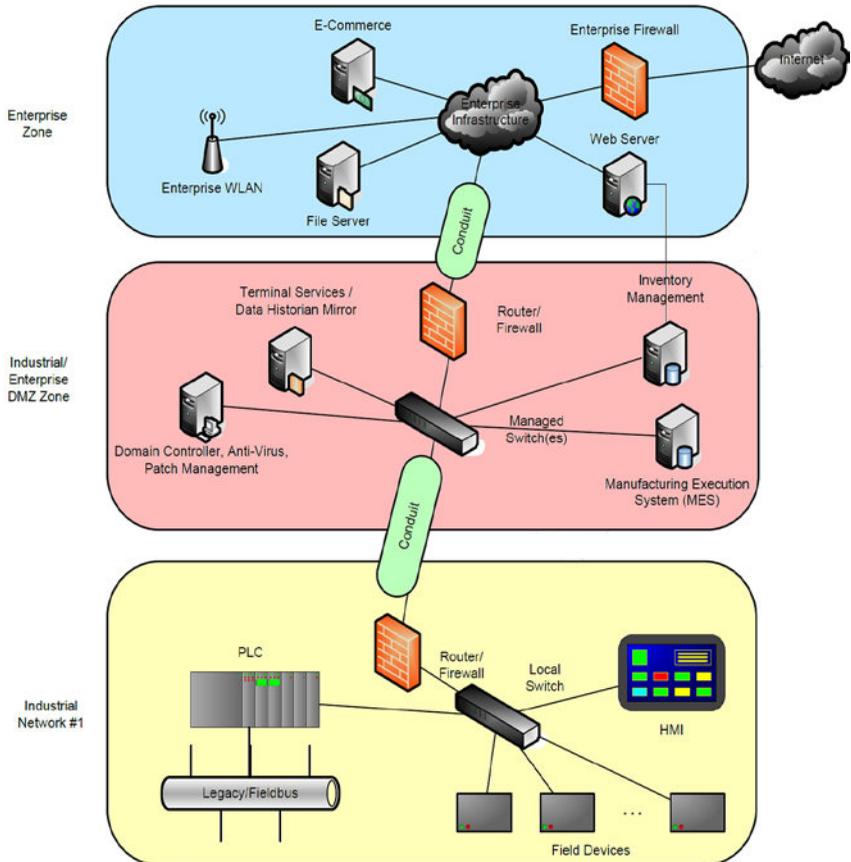


Fig. 16.4 – Zone model of Industrial Control Systems
(source: ISA/IEC 62443)

Channels are the specific communication links established within a communication conduit.

In order to fully articulate the systems and components, the range of coverage may be described from several perspectives, including (see Fig. 16.1):

- Range of functionality included;
- Systems and interfaces;
- Criteria for selecting included activities;

- Criteria for selecting included assets;
- Consequence based criteria.

The scope of ICS security can be described in terms of the range of functionality within an organization's information and automation systems. This functionality is typically described in terms of one or more models.

It is also possible to describe the ICS in terms of connectivity to associated systems and interconnectivity of hardware and software components. All issues that can affect or influence the safe, secure, and reliable operation of industrial processes should be covered.

Activities associated with manufacturing operations includes the following: predictable operation of the process, process or personnel safety, process reliability or availability, process efficiency, process operability, product quality, environmental protection, compliance with relevant regulations, and product sales or custody transfer affecting or influencing industrial processes.

ICS are usually related with assets for which security is essential to the protection. This range of coverage includes systems whose compromise could result in the endangerment of public or employee health or safety, loss of public confidence, violation of regulatory requirements, loss or invalidation of proprietary or confidential information, environmental contamination, and/or economic loss or impact on an entity or on local or national security.

It shall be taken in account ICS compromise could result in any or all of the following situations: endangerment of public or employee safety, environmental protection, loss of public confidence, violation of regulatory requirements, loss of proprietary or confidential information, economic loss, impact on entity, local, state, or national security.

16.4 Security and safety relation in Industrial Control Systems

Safety and security are included in ICS context in Fig. 16.1, however, relation of these attributes requires a separated study. In an ICS context the subjects of security and safety are closely linked. A failure to secure an ICS can in turn result in a potentially unsafe system under control [6,7].

Safety Instrumented Systems (SIS), represent one layer of protection that may be implemented in order to reduce risks associated

with ICS. Traditional risk assessment methodologies in the past, have generally excluded the potential for cyber related attacks to cause safety related incidents. Given that targeted attacks on ICS have occurred and these systems are increasingly being connected to other business systems, they represent a significant potential for common mode failure. As a result, it is necessary in today's world to include cyber security in the overall risk assessment. Without addressing cyber security throughout the entire safety lifecycle, it is impossible to adequately understand the relative integrity of the various layers of protection that involve instrumented systems, including the SIS.

The increasing inter-connectivity of control systems is equally important to industry since new benefits also bring new challenges. Open industrial networks that seamlessly coexist in broader Ethernet systems are being used to link various plant -wide control systems together and connect these systems into expansive, enterprise-level systems via the Internet. As the pace of control system and enterprise network architecture convergence quickens, industrial security depends on staying both flexible and vigilant and successfully controlling the space. What may be considered adequate protection today should evolve as vulnerabilities are identified and new threats emerge.

The discovery of malware that specifically targets industrial control systems brought industrial security to the forefront in manufacturing (see Section 15.2 of this multi-book). As a result, there is growing recognition of the risks and real-world threats that are capable of disrupting control system operation and adversely affecting safety.

An approach to handle requirements to ICS security and functional safety in a general framework is described below.

A set of ICS functional safety requirement can be found in series of industrial standards, for example, IEC 61508 “Functional safety of electrical/electronic/programmable electronic safety-related systems” or IEC 61511 “Functional safety – Safety instrumented systems for the process industry sector”.

These functional safety requirements can be divided in some following categories:

- Requirements to functional safety management;
- Requirements to functional safety life cycle;

– Requirements to systematic (system and software design) failures avoidance;

– Requirements to random (hardware) failures avoidance.

A scope of the above requirements is highly dependent from as named Safety Integrity Level (SIL) which establishes relation between ICS risk level and a scope of the related safety assurance countermeasures. The discussed approach can be represented in a view of a diagram (see Fig. 16.5).

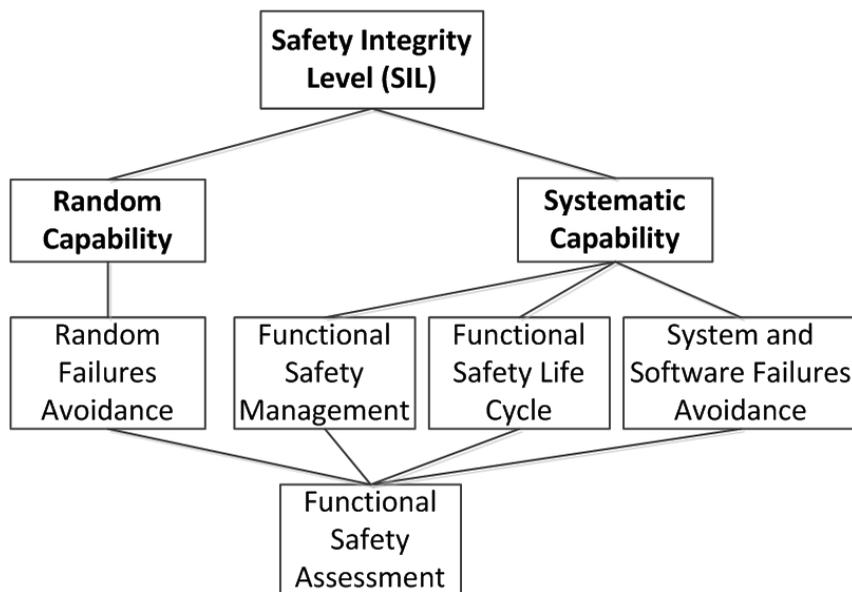


Fig. 16.5 – A concept of ICS safety requirements

The above approach can be applied for ICS security concept (see Fig. 16.1). Firstly, Security Levels shall be implemented for ICS taken into account risks levels (see Section 16.5). Secondly, ISMS shall be implemented and coordinated with functional safety management issues. Thirdly, a common security and safety life cycle shall be established to cover all the process of ICS development, verification and validation (see Section 16.6). Fourthly, common safety and security risks shall be avoided to implement coordinated countermeasures

against random (hardware) and systematic (system and software design) failures (see Section 16.7). Examples of common safety and security random failures avoidance countermeasure are redundancy, self-diagnostic, electromagnetic disturbances protection and others. Examples of common safety and security systematic failures avoidance (attacks avoidance for security) are access control and configuration control. Fifty, assessment shall be periodically performed for both, security and safety. The discussed approach is the base for security and safety coordination, as it is represented on Fig. 16.6.

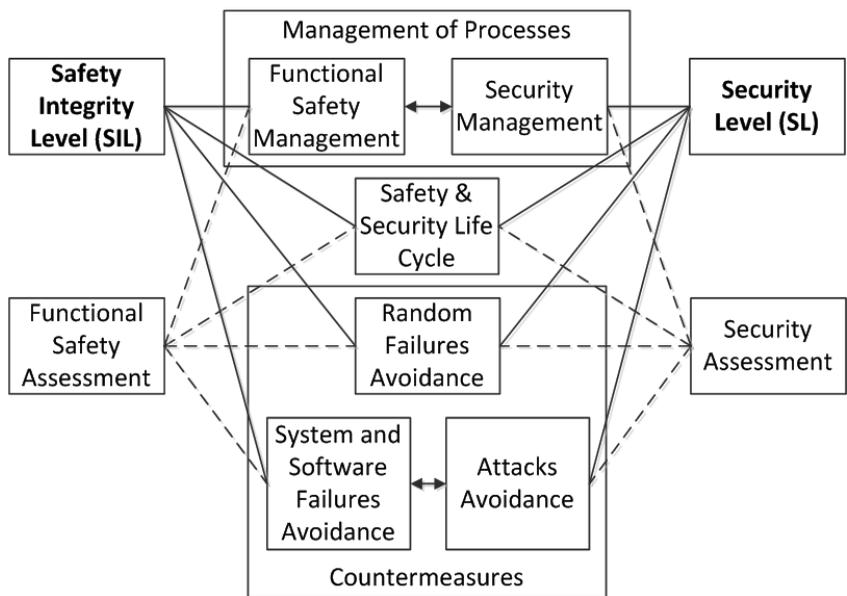


Fig. 16.6 – A concept of ICS harmonized security and safety requirements

16.5 Security Levels of Industrial Control Systems

Safety systems have used the concept of Safety Integrity Levels (SIL) for almost two decades. This allows the safety integrity capability of a component or the safety integrity level of a deployed system to be represented by a single number that defines a protection factor required

to ensure the health and safety of people or the environment based on the probability of failure of that component or system. The process to determine the required protection factor for a safety system, while complex, is manageable since the probability of a component or system failure due to random hardware failures can be measured in quantitative terms. The overall risk can be calculated based on the consequences that those failures could potentially have on Health, Safety and Environmental (HSE). Security systems have much broader application, a much broader set of consequences and a much broader set of possible circumstances leading up to a possible event. Security systems are still meant to protect HSE, but they are also meant to protect the industrial process itself, company proprietary information, public confidence and national security among other things in situations where random hardware failures may not be the root cause. In some cases, it may be a well-meaning employee that makes a mistake, and in other cases it may be a devious attacker bent on causing an event and hiding the evidence. The increased complexity of security systems makes compressing the protection factor down to a single number much more difficult [8-10].

Security levels provide a qualitative approach to addressing security for a zone. As a qualitative method, security level definition has applicability for comparing and managing the security of zones within an organization. As more data become available and the mathematical representations of risk, threats, and security incidents are developed, this concept will move to a quantitative approach for selection and verification of Security Levels (SL). It will have applicability to both end user companies, and vendors of ICS and security products. It will be used to select ICS devices and countermeasures to be used within a zone and to identify and compare security of zones in different organizations across industry segments.

The asset owner will be required to come up with their own definition of what those classifications mean for their particular application. The long-term goal is to move as many of the security levels and requirements to quantitative descriptions, requirements and metrics as possible to establish repeatable applications of the standard across multiple companies and industries. Achieving this goal will take time, since more experience in applying the standards and data on industrial security systems will need to be acquired to justify the quantitative approach.

When mapping requirements to the different Security Levels, standard developers need some frame of reference describing what the different Security Levels mean and how they differ from each other. The goal is to propose such a frame of reference.

The following Security Levels are proposed in ISA/IEC 62443:

– Security Level 0: No specific requirements or security protection necessary. SL 0 has multiple meanings depending on the situation in which it is applied. In defining SL it would mean that the component or system fails to meet some of the SL 1 requirements. This would most likely be for components or systems that would be part of a larger zone where other components or systems would provide compensating countermeasures;

– Security Level 1: Protection against casual or coincidental violation. Casual or coincidental violations of security are usually through the lax application of security policies. These can be caused by well-meaning employees just as easily as they can be by an outsider threat. Many of these violations will be security program related and will be handled by enforcing policies and procedures. A simple example would be an operator able to change a set point on the engineering station in the process control zone to a value outside certain conditions determined by the engineering staff. The system did not enforce the proper authentication and use control restrictions to disallow the change by the operator. Another example would be a password being sent in clear text over the conduit between the process control zone and the Demilitarized Zone (DMZ), allowing a network engineer to view the password while troubleshooting the system;

– Security Level 2: Protection against intentional violation using simple means with low resources, generic skills and low motivation. Simple means do not require much knowledge on the part of the attacker. The attacker does not need detailed knowledge of security, the domain or the particular system under attack. These attack vectors are well known and there may be automated tools for aiding the attacker. They are also designed to attack a wide range of systems instead of targeting a specific system, so an attacker does not need a significant level of motivation or resources at hand. An example would be a virus that infects the maintenance workstation in the Plant DMZ zone spreading to the process control engineering workstation since they both use the same general purpose operating system. Another example

would be an attacker compromising a web server in the enterprise network by an exploit downloaded from the Internet for a publicly known vulnerability in the general purpose operating system of the web server. The attacker uses the web server as a pivot point in an attack against other systems in the enterprise network as well as the industrial network;

– Security Level 3: Protection against intentional violation using sophisticated means with moderate resources, ICS specific skills and moderate motivation. Sophisticated means require advanced security knowledge, advanced domain knowledge, advanced knowledge of the target system or any combination of these. An attacker going after a Security Level 3 system will likely be using attack vectors that have been customized for the specific target system. The attacker may use exploits in operating systems that are not well known, weaknesses in industrial protocols, specific information about a particular target to violate the security of the system or other means that require a greater motivation as well as skill and knowledge set than are required for Security Level 1 or 2. An example of sophisticated means could be password or key cracking tools based on hash tables. These tools are available for download, but applying them takes knowledge of the system (such as the hash of a password to crack). Another example would be an attacker that gains access to the functional safety PLC through the serial conduit after gaining access to the control PLC through a vulnerability in the Ethernet controller;

– Security Level 4: Protection against intentional violation using sophisticated means with extended resources, ICS specific skills and high motivation. Security Level 3 and Security Level 4 are very similar in that they both involve sophisticated means used to violate the security requirements of the system. The difference comes from the attacker being even more motivated and having extended resources at their disposal. These may involve high-performance computing resources, large numbers of computers or extended periods of time. An example of sophisticated means with extended resources would be using super computers or computer clusters to conduct brute-force password cracking using large hash tables. Another example would be a botnet used to attack a system using multiple attack vectors at once. A third example would be an organized crime organization that has the

motivation and resources to spend weeks attempting to analyze a system and develop custom “zero-day” exploits.

Security Levels have been broken down into three different types: target, achieved and capability. These types, while they all are related have to do with different aspects of the security life cycle.

Target Security Levels (SL-T) are the desired level of security for a particular system. This is usually determined by performing a risk assessment on a system and determining that it needs a particular level of security to ensure its correct operation.

Achieved Security Levels (SL-A) are the actual level of security for a particular system. These are measured after a system design is available or when a system is in place. They are used to establish that a security system is meeting the goals that were originally set out in the target Security Levels.

Capability Security Levels (SL-C) are the security levels that component or systems can provide when properly configured. These levels state that a particular component or system or component is capable of meeting the target Security Levels natively without additional compensating controls when properly configured and integrated.

Security Levels are based on the seven Foundational Requirements for security:

- Identification and authentication control;
- Use control;
- System integrity;
- Data confidentiality;
- Restricted data flow;
- Timely response to events;
- Resource availability.

Instead of compressing Security Levels down to a single number, it is possible to use a vector of Security Levels that uses the seven above Foundational Requirements instead of a single protection factor.

Zones and conduits approach (see Fig. 16.4) is closely related with SL concept. Every situation has a different acceptable level of security. For large or complex systems, it may not be practical or necessary to apply the same SL to all components. Differences can be addressed by using the concept of a zone, defined as a logical or physical grouping of physical, informational, and application assets sharing common security

requirements. This concept can be applied in an exclusive manner where some systems are included in the security zone and all others are outside the zone.

A conduit is a particular type of zone that groups communications that can be logically organized into a grouping of information flows within and also external to a zone. Channels are the specific communication links established within a communication conduit.

ISMS is a second issues which is dependent from SL from the point of view of requirements level.

For ICS, a risk management process as well as ISMS should be employed throughout an organization, using a three-tiered approach to address risk at the organization level; mission/business process level; and information system level (IT system and ICS) [1].

ICS security programs should always be part of broader ICS safety and reliability programs at both industrial sites and enterprise ISMS, because cybersecurity is essential to the safe and reliable operation of modern industrial processes.

16.6 Security and safety life cycle of Industrial Control Systems

As if was defined in Section 16.4, security and safety can be implemented in a fame of common life cycle. Typical ICS life cycle include four the main stages [5]:

- ICS development, what is responsibility of ICS vendor;
- ICS installation and commissioning at the operation site, what is responsibility of a system integrator;
- ICS operation and maintenance, what is responsibility of an operator (assets owner);
- ICS decommissioning, what is also responsibility of an operator (assets owner).

From the point of security view, operation is the most important phase because security features are running and maintaining during ICS operation. However, the most complicated structure is implemented for ICS development, since in accordance with standards requirements this part of life cycle has to have a V-shape [7] (see Fig. 16.7). Development phases are signed with usual lines and verification and validation phases are signed with dash-lines.

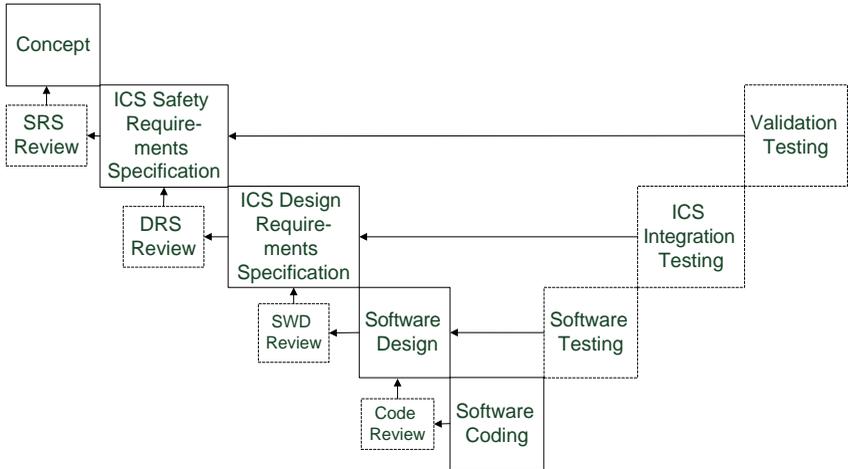


Fig. 16.7 – V-shape security and safety life cycle (development stage)

The main issues of phase content and goals are given in Table 16.1.

Table 16.1 –Life cycle content (development stage)

Phase name	Activities
Concept	Developments of top-level conceptual document, which states recognition of needs for plants and processes automation including hazards, threats and risks analysis
SRS	Developments of ICS functional requirements specification (“black box”) including modes, timing, interfaces, signals, self-diagnostics and others
SRS Review	SRS verification against Concept requirements
DRS	Developments of ICS architecture requirements specification (“white box”) including detailed structure and behavior description
DRS Review	DRS verification against SRS requirements

Phase name	Activities
Software Design	Developments of algorithms and data structure for every software module
Software Design Review	Software Design documents verification against DRS requirements
Software Coding	Writing a software source code in accordance with Software Design documents
Code Review	Software code verification against Software Design documents requirements including Static Code Analysis
Software Testing	Functional and structural testing of software code against Software Design documents requirements
Integration Testing	Functional testing of integrated ICS components against DRS requirements
Validation Testing	Functional testing of ICS against SRS requirements

The main security features implementation for specified life cycle stages and phases are given below.

During concept phase, ICS security implementation includes the following activities:

- Recognize need for protection of property, assets, services, or personnel
- Start developing the security program
- Document assets, services, and personnel needing some level of protection
 - Document potential internal and external threats to the enterprise
 - Establish security mission, visions, and values;
 - Develop security policies for industrial automation and control systems and equipment, information systems and personnel.

During SRS development phase, ICS security implementation includes the following activities:

- Continue developing the security program;
- Establish security functional requirements for ICS and equipment, production systems, information systems, and personnel;

- Perform vulnerability assessment of facilities and associated services against the list of potential threats;
- Discover and determine legal requirements for ICS;
- Perform a risk analysis of potential vulnerabilities and threats;
- Categorize risks, potential impacts to the enterprise, and potential mitigations;
- Segment security work into controllable tasks and modules for development of functional designs;
- Establish network functional definitions for security portions of ICS.

During DRS development phase, ICS security implementation includes the following activities:

- Development of the security program is completed in this phase
 - Define functional security requirements for enterprise zones, plant zones, and control zones;
 - Potential activities and events are defined and documented to perform the functional requirements and implement plans for a secured enterprise;
 - Define functional security organization and structure;
 - Define functions required in the implementation plan;
 - Define and publish security zones, borders, and access control portals;
 - Complete and issue security policies, and procedures;
 - Design physical and logical systems to perform the functional requirements previously defined for security;
 - Conduct training programs;
 - Initiate asset management and change management programs;
 - Design borders and access control portals for protected zones;
- During software design and software coding phases, the designed before security features shall be implemented into the ICS components.
- During installation and commissioning stage, ICS security implementation includes the following activities:
- Physical security equipment, logical applications, configurations, personnel procedures are installed to complete the secured zones and borders within the enterprise;
 - Access control portal attributes are activated and maintained;
 - Training programs are completed;

- Asset management and change management programs are functional and operating;

- Security system turnover packages are completed and ready for acceptance by operations and maintenance personnel.

During operation and maintenance stage, ICS security implementation includes the following activities:

- Security equipment, services, applications and configurations are completed and accepted by operations and maintenance;

- Personnel are trained, and continued training is provided on security matters;

- Maintenance monitors security portions of enterprise, plant, or control zones and keeps them functioning properly

- Asset management and change management is operational and maintained

- Risk reviews, internal and external audits are conducted.

During decommissioning stage, ICS security implementation includes the following activities:

- Obsolete security systems are properly disassembled and disposed of;

- Security borders are updated or recreated for zone protection;

- Access control portals are created, redefined, reconfigured, or closed;

- Personnel are briefed about changes in the security systems and items along with the impact to associated security systems;

- Intellectual property is properly collected, documented, and securely archived or destroyed.

Processes and management maturity issues are closely related with life cycle issues. It is possible to describe the relative maturity of a security program in terms of a life cycle that consists of several phases. Each of these phases consists of one or more steps. So company level should be implemented or maintained through ICS life cycle.

16.7 Survey of security and resilience assurance countermeasures for Industrial Control Systems

This Section is mainly based on statements of the document “NIST SP 800-82. Guide to Industrial Control Systems (ICS) Security”, which

provides guidance for establishing secure ICSs [1], including SCADA, DCS and PLCs.

It is typically not possible to achieve the security objectives through the use of a single countermeasure or technique. A superior approach is to use the concept of defense in depth, which involves applying multiple countermeasures in a layered or stepwise manner. For example, intrusion detection systems can be used to signal the penetration of a firewall.

Defense-in-depth strategy should be implemented to assure security and resilience for ICS. ICS defense-in-depth may include the following countermeasures and means:

- Developing security policies, procedures, training and educational material that applies specifically to the ICS;
- Considering ICS security policies and procedures based on the Homeland Security Advisory System Threat Level, deploying increasingly heightened security postures as the Threat Level increases;
- Addressing security throughout the lifecycle of the ICS from architecture design to procurement to installation to maintenance to decommissioning;
- Implementing a network topology for the ICS that has multiple layers, with the most critical communications occurring in the most secure and reliable layer;
- Providing logical separation between the corporate and ICS networks (e.g., inspection firewall(s) between the networks, unidirectional gateways);
- Employing a DMZ network architecture (i.e., prevent direct traffic between the corporate and ICS networks);
- Ensuring that critical components are redundant and are on redundant networks;
- Designing critical systems for graceful degradation (fault tolerant) to prevent catastrophic cascading events;
- Disabling unused ports and services on ICS devices after testing to assure this will not impact ICS operation;
- Restricting physical access to the ICS network and devices;
- Restricting ICS user privileges to only those that are required to perform each person's job (i.e., establishing role-based access control and configuring each role based on the principle of least privilege);

- Using separate authentication mechanisms and credentials for users of the ICS network and the corporate network (i.e., ICS network accounts do not use corporate network user accounts);

- Using modern technology, such as smart cards for Personal Identity Verification;

- Implementing security controls such as intrusion detection software, antivirus software and file integrity checking software, where technically feasible, to prevent, deter, detect, and mitigate the introduction, exposure, and propagation of malicious software to, within, and from the ICS;

- Applying security techniques such as encryption and/or cryptographic hashes to ICS data storage and communications where determined appropriate;

- Expeditiously deploying security patches after testing all patches under field conditions on a test system if possible, before installation on the ICS;

- Tracking and monitoring audit trails on critical areas of the ICS;

- Employing reliable and secure network protocols and services where feasible.

Technological security assurance countermeasures are based mainly on implementation of secure ICS architecture including the following issues:

- Network segmentation and segregation;

- Boundary protection;

- Encryption;

- Firewalls and DMZ establishing;

- Protocols choice and control;

- Redundancy and fault tolerance;

- Authentication and authorization;

- Monitoring, logging, and auditing;

- Incident detection, response, and system recovery.

Conclusions

The conceptual security requirements taxonomy, based on analysis of security standards requirements (mainly ISA/IEC 62443 and NIST SP 800-82) includes four the main parts:

- Risk management and assessment as a corner stone for definition of acceptable risks levels and countermeasures for risks reduction;
- Categories of security features implementation which include triad “People – Process – Technologies”;
- ICS context which drive to define requirement taking into account specifics of ICS; this concept includes three types of models (reference, physical architecture and zone models) as well as functionality, components, assets and other definitions, and security and safety coordination issues;
- ICS security levels concept which grades risk levels for ICS separated parts and establishes different life cycle processes and countermeasures for different security levels.

The core and foundational principle of the ICS Information Security Management System is the “People – Process – Technologies” categories triad. Each of the category includes own requirements and recommendations.

The basis for identifying the security needs and important characteristics of the environment at a level of details necessary to address security issues can be expressed with three ICS models reference model, physical architecture model and zone model.

In order to fully articulate the systems and components, the range of definitions should be described from several perspectives, including the following:

- Range of functionality included;
- Systems and interfaces;
- Criteria for selecting included activities;
- Criteria for selecting included assets;
- Consequence based criteria.

Safety and security relations are also included in ICS context. In an ICS context the subjects of security and safety are closely linked. A failure to secure an ICS can in turn result in a potentially unsafe system under control.

Requirements to ICS security and safety can be harmonized in accordance with the following taxonomy:

- Security Level (SL) – Safety Integrity Level (SIL);
- Security and functional safety management;
- Security and safety life cycle;

- Random (hardware) failures avoidance, systematic (system and software design) failures avoidance, attacks avoidance;
- Security and functional safety assessment.

Security levels provide a qualitative approach to addressing security for a zone. As a qualitative method, security level definition has applicability for comparing and managing the security of zones within an organization. As more data become available and the mathematical representations of risk, threats, and security incidents are developed, this concept will move to a quantitative approach for selection and verification of Security Levels.

The following Security Levels are proposed in ISA/IEC 62443:

- Security Level 0: No specific requirements or security protection necessary;
- Security Level 1: Protection against casual or coincidental violation;
- Security Level 2: Protection against intentional violation using simple means with low resources, generic skills and low motivation;
- Security Level 3: Protection against intentional violation using sophisticated means with moderate resources, ICS specific skills and moderate motivation;
- Security Level 4: Protection against intentional violation using sophisticated means with extended resources, ICS specific skills and high motivation.

Security and safety can be implemented in a frame of common life cycle. Typical ICS life cycle include four the main stages:

- ICS development, what is responsibility of ICS vendor;
- ICS installation and commissioning at the operation site, what is responsibility of a system integrator;
- ICS operation and maintenance, what is responsibility of an operator (assets owner);
- ICS decommissioning, what is also responsibility of an operator (assets owner).

From the point of security view, operation is the most important phase because security features are running and maintaining during ICS operation. However, the most complicated structure is implemented for ICS development, since in accordance with standards requirements this part of life cycle has to have a V-shape. The main security features implementation should be specified for life cycle stages and phases.

It is typically not possible to achieve the security objectives through the use of a single countermeasure or technique. A superior approach is to use the concept of defense in depth, which involves applying multiple countermeasures in a layered or stepwise manner.

Technological security assurance countermeasures are based mainly on implementation of secure ICS architecture including the issues like network segmentation and segregation, boundary protection, encryption, and other.

Questions to self-checking

1. Describe ICS security concept based on taxonomy security standards requirements.
2. Describe requirements and recommendations for included in “People” category from the “People – Process – Technologies” triad.
3. Describe requirements and recommendations for included in “Process” category from the “People – Process – Technologies” triad.
4. Describe requirements and recommendations for included in “Technologies” category from the “People – Process – Technologies” triad.
5. Describe a purpose and a structure of ICS reference model.
6. Describe a purpose and a structure of ICS physical architecture model.
7. Describe a purpose and a structure of ICS physical zone model.
8. Which types of definitions should be added to fully describe ICS and components?
9. What is relation between ICS security and safety?
10. How can be harmonized requirements to ICS security and safety?
11. Describe a concept of Security Levels.
12. Which Security Levels can be implemented in ICS?
13. Which stages are included in ICS life cycle?
14. What is structure of V-shape ICS development life cycle?
15. How should security features be implemented through ICS life cycle?
16. What is a concept of defense in depth strategy?

17. Give examples of countermeasures for defense in depth implementation.

18. Provide examples of implementation of secure ICS architecture.

References

1. NIST SP 800-82 Revision 2, Guide to Industrial Control Systems (ICS) Security: Supervisory Control and Data Acquisition (SCADA) Systems, Distributed Control Systems (DCS), and Other Control System Configurations such as Programmable Logic Controllers (PLC). – National Institute of Standards and Technologies, 2015. – 247 p.

2. R. Hawkins, I. Habli, D. Kolovos, R. Paige, T. Kelly. Weaving an Assurance Case from Design: A Model-Based Approach // Proceeding of 2015 IEEE 16th International Symposium on High Assurance Systems Engineering (HASE). – P.110-117.

3. P. Clark, C. Irvine, T. Levin, T. Nguyen, D Shifflett, D. Miller. Initial Documentation Requirements for a High Assurance System. Lessons Learned. Technical Report NPS-CS-06-007. Naval Postgraduate School, Monterey, California, USA, 2006. – 10 p.

4. R. Bloomfield, M. Bendele, P. Bishop, R. Stroud, S. Tonks. The risk assessment of ERTMS-based railway systems from a cyber security perspective: methodology and lessons learned // Proceeding of 1st International Conference, RSSRail 2016. – P. 3-19.

5. V. Kharchenko, V. Sklyar, E. Brezhnev, A. Boyarchuk, O. Starov, C. Phillips. University-Industry Cooperation in Cyber Security Domain: Multi-Model Approach, Tools and Cases // Practitioners Proceedings 2016 University-Industry Interaction Conference (UIIC). – P. 265-283.

6. V. Sklyar, V. Kharchenko, E. Bakhmach, A. Andrashov. FPGA-Based I&C Systems: A Technological Trick or a Way to Improve NPPs Safety and Security? // Proceedings of 20th International Conference on Nuclear Engineering – Proceeding of 20th International Conference on Nuclear Engineering ‘ICONE20-POWER 2012’.

7. V.S. Kharchenko, V.V. Sklyar. Assurance Case Driven Design for software and hardware description language based

systems // Radioelectronic and Computer Systems. – 2016. – No. 5(79). – P. 98-103.

8. S. Drimer. Security for volatile FPGAs, UCAM CL-TR-763. – University of Cambridge, 2009. – 169 p.

9. T. Huffmire, C. Irvine, T. Nguyen, T. Levin, R. Kastner, T. Sherwood. Handbook of FPGA Design Security. – Springer, 2010. – 177 p.

10. R. Bloomfield, K. Netkachova, R. Stroud. Security-Informed Safety: If It's Not Secure, It's Not Safe // Software Engineering for Resilient Systems Lecture Notes in Computer Science, Volume 8166, Springer Berlin Heidelberg, 2013. – P. 17-32.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ К РАЗДЕЛУ 2

DCS – Distributed Control Systems
DMZ – Demilitarized Zone
DRS – Design Requirement Specification
HSE – Health, Safety and Environmental
IEC – International Electrotechnical Commission
ICS – Industrial Control System
ISMS – Information Security Management System
ISA – International Society of Automation
ISO – International Standardization Organization
NIST – National Institute of Standards and Technology
NIST SP – NIST Special Publication
PLC – Programmable Logic Controllers
SCADA – Supervisory Control And Data Acquisition
SIL – Safety Integrity Level
SIS – Safety Instrumented Systems
SL – Security Level
SRS – Safety Requirement Specification

АННОТАЦИЯ

Раздел содержит концептуальное описание таксономии требований к информационной безопасности АСУ ТП, основанный на анализе требований стандартов (в первую очередь, ISA/IEC 62443 и NIST SP 800-53). Данная таксономия включает четыре основные части:

- управление рисками и оценивание рисков;
- категории реализации свойств информационной безопасности, включающие триаду «Персонал – Процессы – Технологии»;
- контекст, подчеркивающий особенности АСУ ТП;
- уровни информационной безопасности, определяемые уровнями рисков для составных частей АСУ ТП; на основе данных уровней устанавливается объем реализации процессов жизненного цикла и защитных контрмер.

Данный подход применяется в качестве основы для выбора стратегии обеспечения информационной безопасности.

Розділ містить концептуальний опис таксономії вимог до інформаційної безпеки АСУ ТП, заснований на аналізі вимог стандартів (в першу чергу, ISA/IEC 62443 і NIST SP 800-53). Дана таксономія включає чотири основні частини:

- управління ризиками та оцінювання ризиків;
- Категорії реалізації властивостей інформаційної безпеки, що включають триаду «Персонал - Процеси - Технології»;
- Контекст, що підкреслює особливості АСУ ТП;
- Рівні інформаційної безпеки, що визначаються рівнями ризиків для складових частин АСУ ТП; на основі даних рівнів встановлюється обсяг реалізації процесів життєвого циклу і захисних контрзаходів.

Даний похід застосовується в якості основи для вибору стратегії забезпечення інформаційної безпеки.

The section contains description of conceptual security requirements taxonomy, based on analysis of security standards (mainly

ISA/IEC 62443 and NIST SP 800-82), which includes four the main parts:

- Risk Management and Assessment;
- Categories of security features implementation which include triad “People – Process – Technologies”;
- ICS context which drive to define requirement taking into account specifics of ICS;
- ICS security levels concept which are graded by risk levels for ICS separated parts and establishes different life cycle processes and countermeasures for different security levels.

Such approach is used as the base to choose a strategy of ICS security assurance.

17 METHODS AND TECHNIQUES OF FPGA BASED INDUSTRIAL CONTROL SYSTEMS SAFETY AND SECURITY ASSESSMENT

17.1 FPGA-based Industrial Control Systems Safety and Security Assessment: A Problem Statement

Nowadays industrial control systems (ICSs) are widely used by the world industry in various areas in forms of Instrumentation and Control systems for Nuclear Power Plants, on-board computer-based systems, electronic medical systems, etc. The problems of ICS safety and security assessment should be discussed taking into account trends of computer technologies development. One of the contemporary trends is dynamically growing application of relatively novel complex electronic components, particularly, Field Programmable Gates Arrays (FPGAs) in the most critical ICS.

This module provides materials concerning safety and security assessment of FPGA-based ICS.

FPGA is a convenient technology not only for implementation of auxiliary functions (transformation and preliminary processing of data, diagnostics, etc), it is also effective for implementation of safety important ICS control functions. Application of the FPGA technology is more reasonable than application of software-based technology (microprocessors) in many cases [1].

Moreover, FPGA technology is now being trend in SCSs implementation that inevitably leads to new challenges in various aspects of such systems design, operation and maintenance requiring new approaches, techniques and appropriate requirements

The following FPGA features are important for safety and security assessment:

- development and verification are simplified due to apparatus parallelism in control algorithms implementation and execution for different functions, absence of cyclical structures in FPGA projects, identity of FPGA project presentation to initial data, advanced testbeds and tools, verified libraries and IP-cores;

- existing technologies of FPGA projects development (graphical scheme and library blocks in CAD environment; special hardware describing languages VHDL, Verilog, Java HDL, etc; microprocessor emulators which are implemented as IP-cores) allow increasing a number of possible options of different project versions and multi-version ICS;

- fault-tolerance, data validation and maintainability are improved due to use of: redundancy for intra- and inter-crystal levels; possibilities of implementation of multi-step degradation with different types of adaptation; diversity and multi-diversity implementation; reconfiguration and recovery in the case of component failures; improved means of diagnostics;

- FPGA reprogramming is possible only with the use of especial equipment (it improves a security); stability and survivability of FPGA projects are ensured due to the tolerance to external electromagnetic, climatic, radiation influences, etc.

Generally, it is difficult to perform throughout ICS safety and security assessment for several reasons, including:

- complex fault-tolerant architecture;
- usage of multiversion technologies;
- large number of different components.

Also, there are a lot of assessment techniques that are changing and progressing constantly.

Many authors in the field have emphasized the usefulness of particular techniques as well as their restrictions [2-4]. In this module available information on different techniques is being colligated.

One of the most challenging security problems in a modern world is security of various safety-critical ICSs considering increasing attack rate on assets by use of vulnerabilities. Such systems can contain wide set of general and technology-specific vulnerabilities. Number of vulnerabilities and threats become more and more owing to application of different types off-the-shelf (OTS), first of all, commercial-OTS (COTS).

Furthermore, the possibilities of development of unified technique for ICS safety and security assessment are discussed. This technique should enable validation and eventual certification of safety, security and reliability of ICS via modeling and analysis as well as simulation

and experimentation. The use of different approaches is important since it confers a high level of confidence in results.

17.2 Taxonomies of ICS's attributes

17.2.1 Possible attributes and taxonomies of ICSs

One of the most important attributes of ICS is dependability [5]. Dependability of a system is the ability to deliver required services (or perform functions) that can justifiably be trusted. Dependability is a complex attribute of an ICS that can be represented by a set of primary attributes, including:

- reliability: continuity of correct (required) services;
- availability: readiness for correct services;
- survivability: ability to minimize loss of quality and to keep capacity of fulfilled functions under failures caused by internal and external reasons;
- safety: absence of catastrophic consequences for the user(s) and the environment;
- integrity: absence of improper system alternations;
- confidentiality: absence of unauthorized disclosure of information;
- high confidence: ability of correct estimation of services quality, i.e. definition of trust level to the service;
- maintainability: ability to undergo modifications and repairs;
- security: the protection from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability.

In turn, safety attribute of ICS can have some particular (or secondary) attributes depending on exact system, environment and conditions that have influence on the primary attribute. Here, we distinguished the following attributes (see Fig. 17.1): reliability, security and trustworthiness, and we denoted their two-way influence.

We should note that such particular attributes may be defined for each of primary attributes, thus, representing hierarchical structure of ICS's generic attributes set. Moreover, those secondary and further

attributes may turn to be common for different primary attributes due to their incomplete “orthogonality”.

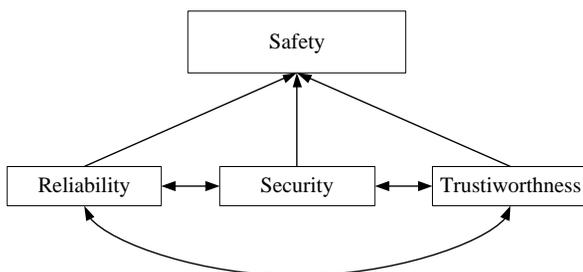


Fig. 17.1 – Taxonomy of safety attribute

17.2.2 Metrics of Interference

Thus, we can state that a set of ICS attributes can be represented in a form of i -level hierarchical model, and each of i levels contains k_i attributes. As an example, Fig. 17.2 represents an element of last two levels of an ICS attributes hierarchical model consisting of i levels.

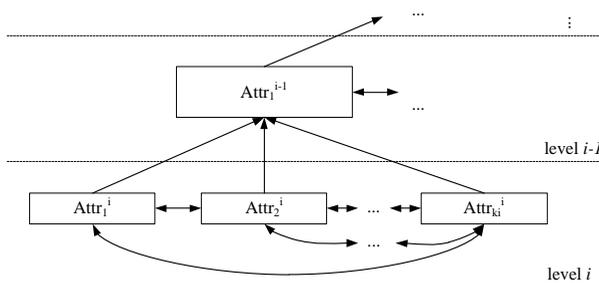


Fig. 17.2 – Levels of ICS attributes hierarchy

One of the possible ways to reveal criticality of two-way influence for ICS’s attributes, is in creating of attributes influence matrix. Such a problem can be solved, in particular, in the following ways:

1. Create a set of n “local” influence matrixes for i hierarchical levels; each of the matrixes consists of k_i attributes (see Fig. 17.3), and, therefore of k_i rows. Such number n can be calculated using the following equation:

$$n = \sum_{x=1}^{i-1} k_x \quad (17.1)$$

The number of rows in each matrix associated with the level m , where $m=[1, i-1]$, is equal to a number of attributes (k_m) at the lower level $m+1$: for example, the local matrix for a single attribute of $i-1$ level consists of k_i rows.

A set of such “local” influence matrixes represents the case of a metric mostly intended for independent assessment of the ICS’s attributes within the single level.

2. Create the single “global” influence matrix where each of all the n attributes (see Eq. (17.1)) is reflected by a single row and appropriate column (see Fig. 17.4).

“Global” influence matrix can be considered as another metric, which is suitable for assessment of the ICS as a whole.

Thus, on the one hand, such metrics allow sharing ICS resources in order to assure the required level of security (a vertical related to different levels in Fig. 17.2), on another hand, they allow optimizing the use of the resources (within the same level, see Fig. 17.2).

		Attr _{k_{i-1}} ⁱ⁻¹		
		<i>low</i>	<i>medium</i>	<i>high</i>
Attr ₁ ⁱ			✗	
Attr ₂ ⁱ				✗
⋮	
Attr _{k_i} ⁱ		✗		

Fig. 17.3 – Local influence matrix

	$Attr_1^i$...	$Attr_{ki}^i$...	$Attr_1^1$...	$Attr_{ki}^1$
$Attr_1^i$		
\vdots
$Attr_{ki}^i$	L	
\vdots
$Attr_1^1$	M	...	L	
\vdots
$Attr_{ki}^1$	L	...	H	...	M	...	

Fig. 17.4 – Global influence matrix

17.3 Extension of DLC-based analysis for ICSs

Development process of modern ICSs requires strong formalized processes for both design and verification and validation (V&V) activities. Thus, development life cycle (DLC) of an ICS can be represented in a form of V-model. To illustrate an example of such model, we present software systematic capability and the DLC (the V-model) in Fig. 17.5.

In terms of the whole system, such V-model implies development of certain artifacts (or components) after completing specific design activities. Each artifact is under strong verification activities in order to prevent unauthorized design and/or functionality of the system.

FPGA technology is now being widely used by the world industry and more often in ICSs for various areas. Application of FPGA technology allows developers to implement intended functions in a convenient and reliable way.

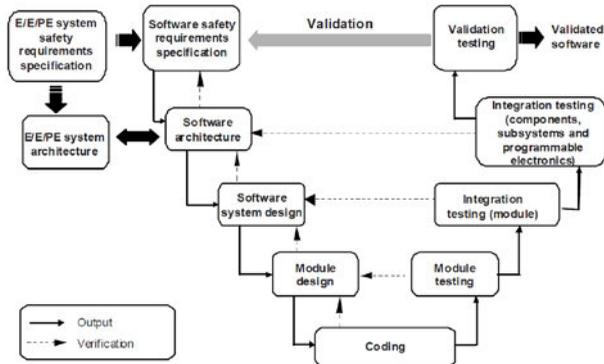


Fig. 17.5 – Software systematic capability and the DLC (the V-model)

Modern trend is in that ICSs are being complex, containing plenty of components, and often based on FPGA technology. In order to consider all the features of such complexity and used technologies, the analysis of ICS attributes should be performed. In such a case, overall DLC of a ICS can be represented in a form of a set of particularly overlapped “sub-V-models” corresponding to each of ICS components’ DLCs. Each of “sub-V-models” covers component-specific development stages and contains appropriate return points.

In a general case, both start point and length of a component’s DLC are different from ICS’s overall DLC due to various reasons. Hence, it is possible to separate all “sub-V-models” of components DLCs to perform comprehensive assessment of required attribute related with the component. Such complete set of all “sub-V-models” for each of the ICS components DLCs forms a plane, or component-oriented V-model of ICS’s DLC (see Fig. 17.6).

Further, it is possible to associate DLC of exact attribute with each of the ICS’s components within the component-oriented V-model. A set of components’ attributes, again, forms another one plane – attributes plane. Hence, we already have two planes: for components and attributes, and, in a bundle with the DLC, it is possible to address the aspect under interest in three-dimension space defined by the three coordinates, which are related with the ICS component, ICS attribute, and DLC stage (see Fig. 17.7).

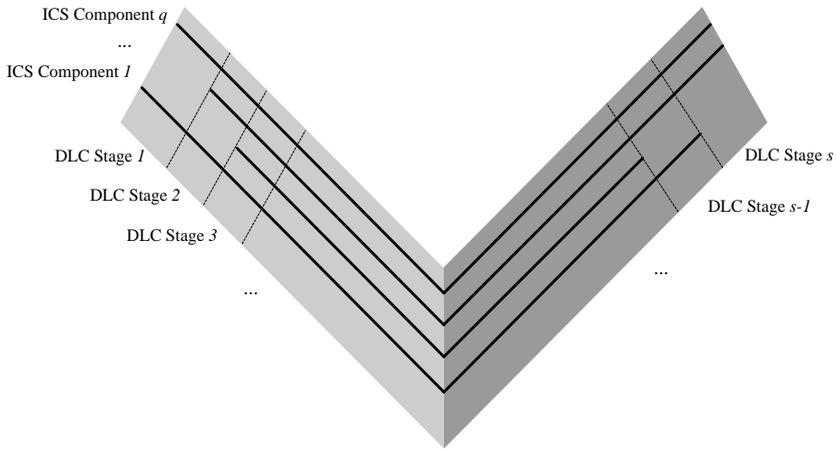


Fig. 17.6 – Component-oriented V-model of ICS's DLC

Thus, now we can talk about attribute-oriented extension to component-oriented V-model of ICS's DLC (see Fig. 17.8). Such approach allows us to independently assess each of SCS components and attributes of the component during the component-specific DLC stage.

The proposed extension allows separation of specific DLC stages for each of components' attributes (for example, safety, security, etc.) to reveal discrepancies of appropriate development processes that can potentially result in anomalies (for example, faults for safety or vulnerabilities for security) of the final product (i.e. ICS or its component).

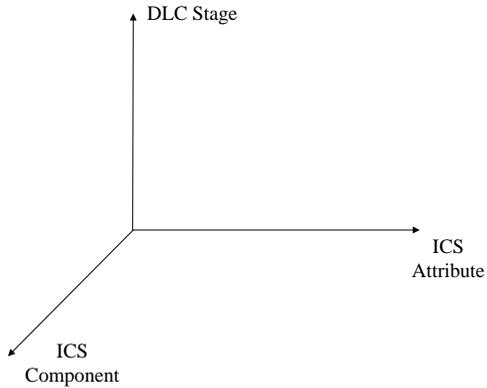


Fig. 17.7 – Three-dimension space

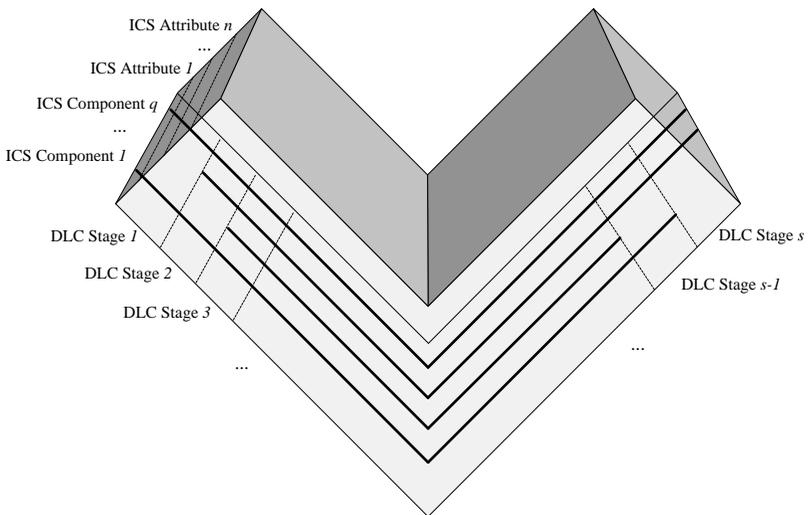


Fig. 17.8 – Attribute-oriented V-model of ICS's DLC

17.4 Establishment of a Secure Development and Operational Environment

Secure Operational Environment is defined as the condition of having appropriate physical, logical and administrative controls within a facility to ensure that the reliable operation of ICSs are not degraded

by undesirable behavior of connected systems and events initiated by inadvertent access to the ICS.

The establishment of a Secure Development and Operational Environment (SDOE) [9-10] in the context of US NRC's RG 1.152, refers to the following aspects:

- measures and controls taken to establish a secure environment for development of the safety ICS against undocumented, unneeded and unwanted modifications;

- protective actions taken against a predictable set of undesirable acts that could challenge the integrity, reliability, or functionality of a ICS during operations.

Phases of the waterfall life cycle model (WLCM) form a framework for describing specific guidance(s) for the protection of digital safety systems and the establishment of an SDOE via identification and mitigation of potential weakness or vulnerabilities in each of the phases that may degrade the SDOE or degrade the reliability of the system.

WLCM includes: concepts; requirements; design; implementation; test; installation, checkout, and acceptance testing; operation; maintenance; retirement. Each of the phases consists of some prescribed activities performed in order to establish and maintain a SDOE. One of the most important activities during concept phase is assessment of vulnerabilities for both development and operational environments. Such assessment forms framework of security requirements to implementation of further life cycle activities and additional secure design solutions to the system under development.

Typical output for vulnerabilities assessment activity is appropriate report describing all the identified vulnerabilities related to both development and operational environments that, in turn, forms the basis for implementation of specific security assurance processes for the ICS. They should be followed in life cycle phase: for example, in development phase, the set of activities, including measures and controls taken to establish a secure environment for development of the ICS against undocumented, unneeded and unwanted modifications

ICS security vulnerabilities classification is presented in Fig. 17.9. SDOE establishment process requires that the development process

should identify and mitigate potential vulnerabilities in each phase of the life cycle.

17.5 Security-oriented analysis of safety-critical ICS

The proposed approach is based on IMECA technique [6-7], as one of the modification of Failure Mode, Effects, and Criticality Analysis (FMECA), which is usually applied to assess reliability and safety. Some “non-ideal” development processes can result in various problems in the corresponding products. Each transition between two consequent (p-1, p) products is accomplished by the implementation of a prescribed process (j) using specific tools under prescribed techniques. Thus, process is a set of sub-processes related to the developer (human), technique, tool, and some of them may contain problems (Fig. 17.10). Such problems can result in product anomalies.

All transition scenarios between two consequent products due to implementation of a process, possibly containing gaps, are presented in Fig. 17.11. In this way, the following statements are true:

1. Presence of gaps within Process_j results in anomalies in Product_p even if Product_{p-1} is “ideal”.
2. Presence of anomalies within Product_{p-1} can be eliminated by “ideal” Process_j in many cases by V&V processes; however, it does not apply to design processes.

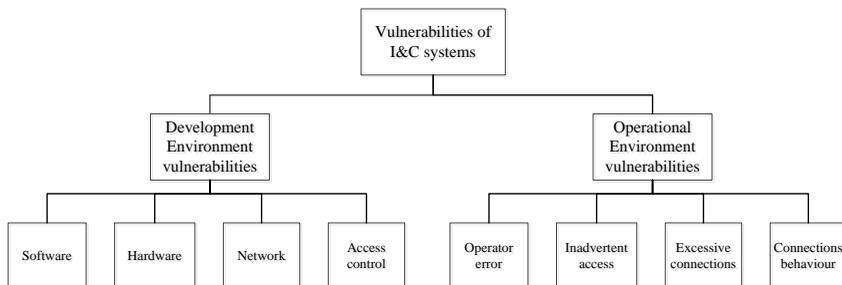


Figure 17.9 – ICS security vulnerabilities classification

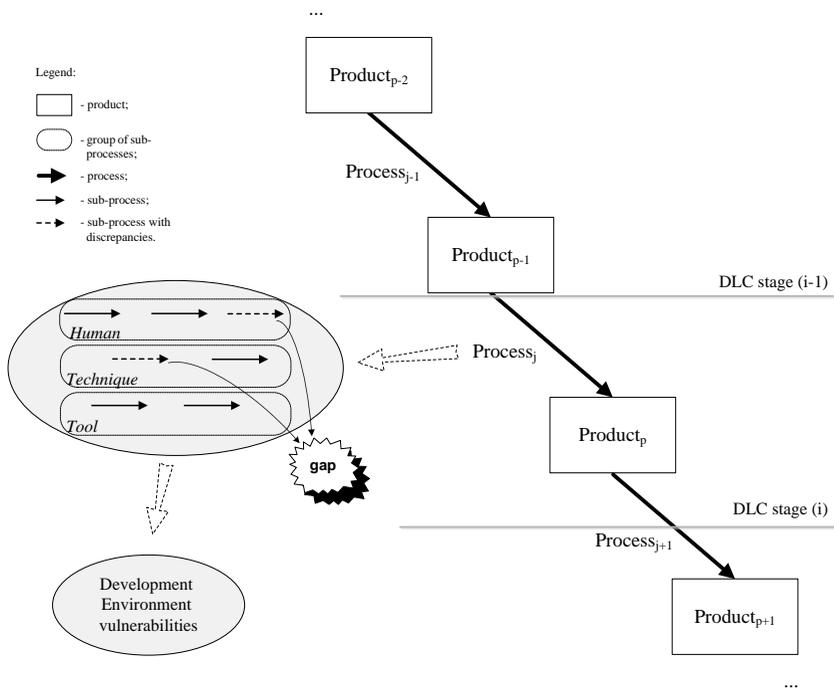


Figure 17.10 – Development process in the ICS development life cycle model

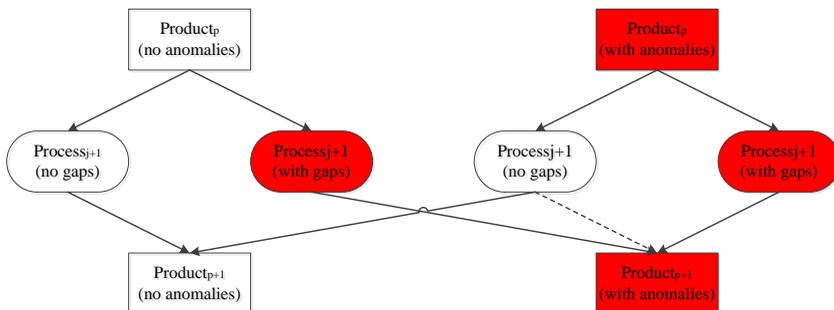


Figure 17.11 – Possible transition scenarios between two consequent products

In terms of security, such process gaps represent sources of security threats, which, in turn, can exploit certain vulnerabilities of the

development process in order to implement successful attack resulting in introduction some anomalies into the product. Table 17.1 represents interrelations between the threat sources and possible types of vulnerabilities related to the development environment. In this way, vulnerabilities types that can be exploited by certain threat(s) are designated by “+” symbol. Such interrelations should be considered during choice of appropriate countermeasures in order to reduce security risks.

Table 17.1 – Interrelations between the threat sources and types of vulnerabilities

		Vulnerabilities			
		Hardware	Software	Network	Access control
Threat source	Human	+	+	+	+
	Technique	+	+	+	
	Tool	+	+	+	

17.6 Extension of gap-and-IMECA-based approach

The activities, required to implement the approach, comprise several consequent steps intended for a comprehensive analysis and assessment of ICS [8]. They are depicted in Fig. 17.12.

The key idea of assessment is in the application of the process-product approach. Therefore, the life cycle model of ICSs should include detailed representation of life cycle processes and appropriate products. Then, it is possible to identify problems (or discrepancies) within the model, i.e. gaps. In general, such gaps may reflect various aspects of the ICS, depending on what system properties are assessed (for example, safety and security).

Hence, depending on the ICS aspects under assessment, each gap should be represented in a form of a formal description; such formal description should be made for a set of discrepancies identified within the gap. The IMECA technique is the most convenient, in our opinion, to perform such description: each identified gap can be represented by a single local IMECA table and each discrepancy inside the gap can be

represented by a single row in that local IMECA table. In this way, complete traceability of life cycle processes, appropriate products and inherent properties of corresponding discrepancies can be achieved. As a result, the number of local IMECA tables would correspond to the number of identified gaps, and the number of rows within each local IMECA table would correspond to the number of identified discrepancies within the appropriate gap.

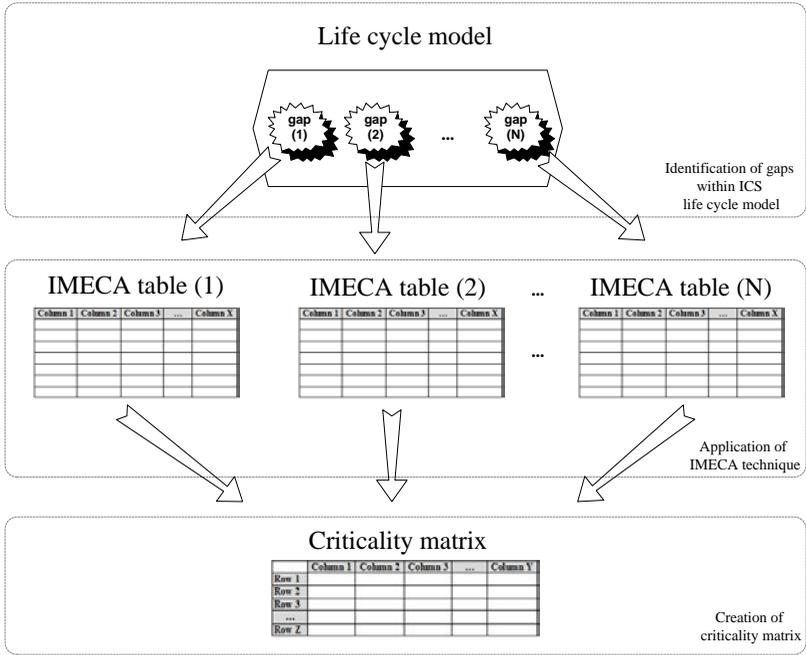


Fig. 17.12 – The principal stages of ICS assessment

After completing the appropriate columns, for example on the basis of expert assessment, for all local IMECA tables, each gap being represented by a set of discrepancies with appropriate numerical values. Data within each row of local IMECA tables reveal, in explicit form, the weaknesses of the ICS aspect under assessment: for example, in terms of safety – system faults and failures, in terms of security – intrusion probability and severity.

Further, in order to implement the approach, the following cases are possible, depending on the scope of the assessment:

1. Assessment of the ICS as a whole. Then, a set of particular IMECA tables (which represent all the identified gaps by a set of discrepancies) should be integrated into the single global IMECA table that reflects the whole system. In this case, each row of the global IMECA table forms the basis for creating a global criticality matrix.

2. Assessment of particular (sub-)systems within the ICS. In this case, it is possible to create an appropriate set of local criticality matrixes that correspond to certain (sub-)systems, based on a set of local IMECA tables.

Integration of local criticality matrixes into a global one is carried out in accordance with the following rule:

$$e_{yz}^G = \bigcup_{k=1}^n e_{yz}^{L_k} , \quad (17.2)$$

where e^G is an element of the global criticality matrix, e^{L_k} is the corresponding element of the k-th local criticality matrix, and n is the total number of local criticality matrixes (equal to total number of gaps).

Moreover, the scales for the numerical values of a discrepancy (for example, its probability and severity) for local criticality matrixes can be set to the same value in order to eliminate the necessity of additional analysis during the creation of a global criticality matrix.

In both cases, the highest risk of the selected assessment aspect corresponds to the highest row in the criticality matrix. In a case of independent gaps and discrepancies, the total risk of R can be calculated using the following equation:

$$R = \sum_{i=1}^n \sum_{j=1}^m p_{ij} D_{ij} , \quad (17.3)$$

where n is the total number of gaps, m is the total number of rows in the IMECA table, p is the occurrence probability, and D is the corresponding damage.

Moreover, the criticality matrix can be extended to be K -dimensional (where $K > 2$) that allows us to consider, for example, the amount of time required to implement the appropriate countermeasures for the assessed ICS.

For example, during the assessment of security, the prioritization of vulnerabilities identified on the basis of process-product approach, should be performed according to their criticality and severity, representing their corresponding stages in the cyber security assurance of the given ICS. The main goal of this step is to identify the most critical security problems within the given set. Prioritization may require the creation of a criticality matrix, where each vulnerability is represented within single rows. In such cases, it is possible to manage the security risks of the whole ICS via changing the positions of the appropriate rows within the matrix (the smallest row number in the matrix corresponds to the smallest risk of occurrence).

During the performance of GA, the identification of discrepancies (and the corresponding vulnerabilities in case of security assessment), can be implemented via separate detection/analysis of problems caused by human factors, techniques and tools, taking into account the influence of the development environment.

Then, after all identified vulnerabilities are prioritized, it is possible to assure security of the ICS by implementing of appropriate countermeasures. Such countermeasures should be selected on the basis of their effectiveness (also, in context of assured coverage), technical feasibility, and cost-effectiveness. But there is an inevitable trade-off between a set of identified vulnerabilities and a minimal number of appropriate countermeasures, which allows us to eliminate vulnerabilities or to make them difficult to be exploited by an adversary. The problem of choosing such appropriate countermeasures is an optimization problem and is still challenging.

17.7 Assessment of FPGA-based ICS Cyber Security

17.7.1 Life cycle model of FPGA-based ICS

Basis of modern critical ICS is usually formed by FPGA chips, which are used in various hardware components. Vulnerabilities of FPGA technology can unintentionally arise or can be introduced by an

adversary during different stages of FPGA chip life cycle. A model of FPGA-based ICS life cycle [11-13] is depicted in Fig. 17.13, and includes:

- 1) stages implemented by FPGA chip vendor:
 - a stage of FPGA chip design (Stage 1);
 - a stage of FPGA chip manufacturing (Stage 2)
 - a stage of FPGA chip packaging and testing (Stage 3);
- 2) stages implemented by ICS developer:
 - a stage of FPGA electronic design (which describes ICS's logic) development for integration into FPGA chip (Stage 4);
 - a stage of FPGA electronic design implementation and testing (Stage 5);
- 3) a stage implemented by user of ICS:
 - a stage of operation of FPGA-based ICS at intended location (Stage 6).

There are factors that can contribute to intended or unintended introduction of vulnerabilities into FPGA-based ICS during implementation of various processes for the following life cycle stages:

- use of malicious tools (EDA tools or CAD tools) during either FPGA chip designing by a vendor or during FPGA electronic design development by an ICS developer;
- use of compromised devices during integration of developed FPGA electronic design into FPGA chip by an ICS developer;
- use of IP-cores from third-party vendors during development of FPGA electronic design by an ICS developer;
- the presence of adversaries (insiders) in development teams.

Some vendors of FPGA chips do not have own manufacturing capacity: in such a case, after implementation of design processes for FPGA chip, that includes application of appropriate tools, they place orders for chip manufacturing among appropriate foundries. Such foundries can introduce additional vulnerabilities into FPGA chips by stealing or modifying FPGA design. Moreover, supply chain of manufactured FPGA chips to developer of ICS is usually traceable and can be audited that, however, does not reduce its importance from point of view of cyber security assurance problem for FPGA-based ICSs.

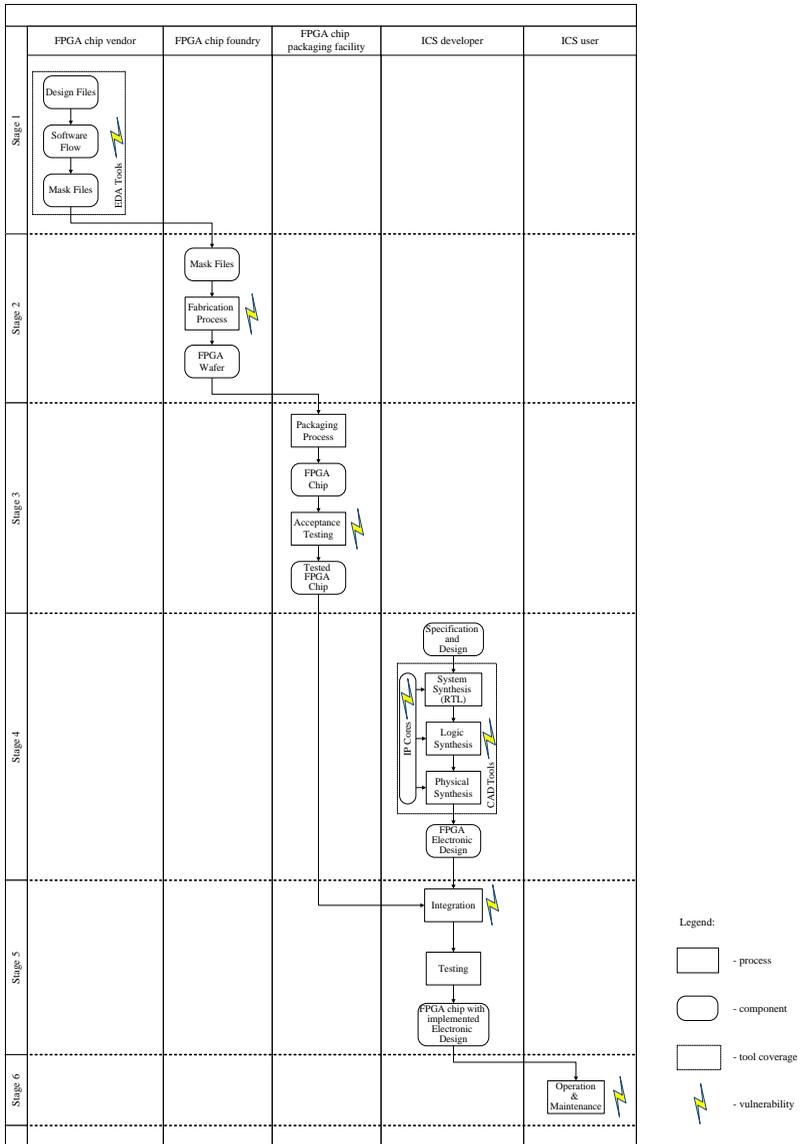


Fig. 17.13 – Life cycle model of FPGA-based ICS

Most of life cycle stages of FPGA chip and FPGA-based ICS are implemented using software tools. Such tools are usually used, for example, during design of printed circuit boards for FPGA chips, in development of FPGA electronic designs, during simulations, etc. Hence, developers of tools for design automation, in turn, can introduce new vulnerabilities into FPGA-based ICSs being developed.

Some vulnerabilities can be introduced into FPGA-based ICSs by their designers via using of IP-cores in FPGA electronic design. IP-core is completed functional description intended for integration into FPGA electronic design, which is being developed. IP-cores can be either in a form of modules for hardware description languages or in a form of compiled netlists. IP-cores are used by designers to save their resources and time. IP-cores can be produced by FPGA chip vendor or third-party vendors, and, in order to assure cyber security of FPGA-based ICS, it is necessary to facilitate safe distribution and integration of such IP-cores by designers of ICSs.

17.7.2 Method of gap-and-IMECA-based assessment for FPGA-based ICS

So, proposed gap-and-IMECA-based approach, as applied to cyber security assessment, can be expressed in the following activities sequence:

Step 1. Identification of security gaps lists for all the components (or modules) of ICS, being assessed, during each life cycle stage. Such lists should include both process gaps (in terms of discrepancies) and product cyber security gaps (in terms of vulnerabilities).

Step 2. Determination of an appropriate set of vulnerabilities for each identified process gap, security gap and possible scenarios to exploit the vulnerabilities. So, for each identified discrepancy or vulnerability, there should be created local IMECA table that reflects: attack mode, attack nature, attack cause, occurrence probability, effect severity, type of effects, and countermeasures.

Step 3. Performance of GA on the basis of IMECA-technique: each gap (identified during Step 1) being represented by one or several rows in a local IMECA table, where the number of such rows corresponds to the number of appropriate discrepancies or

vulnerabilities identified during Step 2. GA should be performed in order to reveal appropriate cyber security risks.

Step 4. Assessment of appropriate columns (occurrence probability and effect severity) in each particular IMECA table, for example, on the basis of expert evaluation. Then, each row of such a local IMECA table represents security weaknesses, which should be analyzed further (during Step 6) in context of the whole ICS.

Step 5. Analysis of cyber security risks of ICS components during different stages: each row in local IMECA tables forms the basis for creation of security criticality matrix, which reveals the weaknesses of appropriate components in a visual form. The highest cyber security risk corresponds to the highest row in security criticality matrix.

17.8 Combined Usage of Safety and Security Assessment Techniques

There are a lot of well-known techniques that can be used for FPGA-based ICS safety and security assessment. Using these techniques it is possible to perform quantitative and/or qualitative assessments. Qualitative assessments though lacking the ability to account, are very effective in identifying potential failures within the ICS.

Fig. 17.14 provides classification of classic assessment techniques.

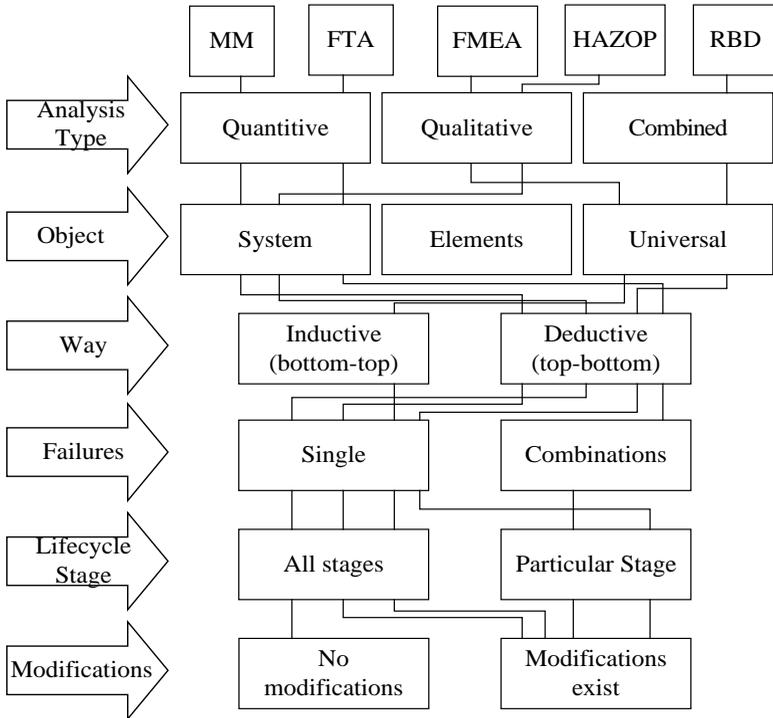


Fig. 17.14 – Classification of assessment methods

Some work could be performed so as to identify possible combination of techniques, possible results are shown in Fig. 17.15. To carry out safety and security analysis it is necessary to have ICS technical documentation (this information is obtained from ICS project) and reliability data of ICS components (is obtained from component vendors).

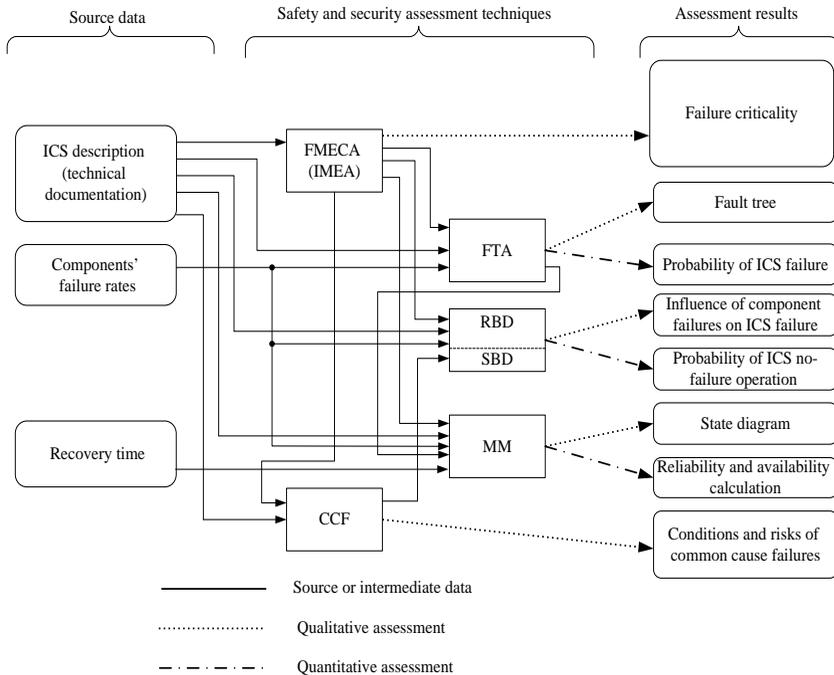


Fig. 17.15 – Combined usage of safety and security assessment methods

The first stage of FPGA-based ICS safety analysis is FMECA (Failure modes, effects and criticality analysis). During this stage all possible failure mechanisms and failure rates for all components involved and quantify failure contribution to overall ICS safety are analysed.

In FMECA qualitative and quantitative results are obtained. Failure mode in FMECA refers to the way a failure might occur. Failure effect is the consequence of failure from the system's point of view. Failure criticality is assigned to each failure mode to get quantitative parameters.

FMECA is carried out early in the FPGA-based ICS development life cycle to find ways of mitigating failures and thereby enhancing reliability through design.

A traditional FMECA uses potential component failures as the basis of analysis. Component failures are analysed one by one, and therefore important combinations of component failures might be overlooked. Environmental conditions, external impacts and other such factors are analysed in FMECA only if they produce component failures; external influences that do not produce component failures (but may still produce ICS failure) are often overlooked.

That's why it is not sufficient to use only FMECA during FPGA-based ICS analysis.

Figure 17.16 show an example of combination of methods. Parallel connections show possible options for usage of different methods. If resources allow, both methods connected in parallel can be used to increase credibility of obtained results. If no, we solve task of choosing the most appropriate method.

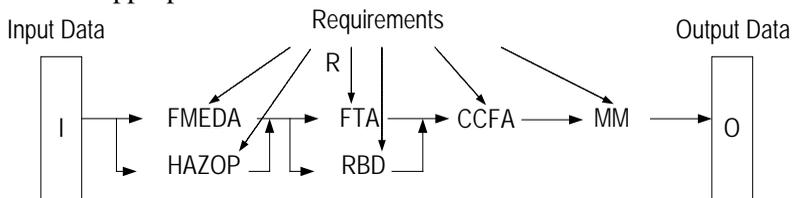


Fig. 17.16 – Example of combined usage of safety and security assessment methods

In other words, during analysis it's necessary:

- to analyze input data required for each method;
- to analyze output data that each method allows to obtain;
- to analyze and choose variants of «horizontal» method combination (when method uses outputs of another method as inputs) and «vertical» one (when results obtained by different methods are compared).

Figures 17.17-17.19 show possible paths of method usage.

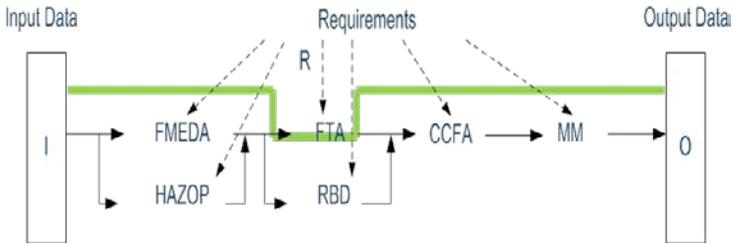


Fig. 17.17 – Example: usage of FTA only

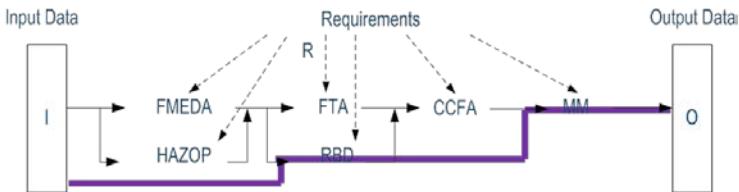


Fig. 17.18 – Example: usage of RBD and MM combination

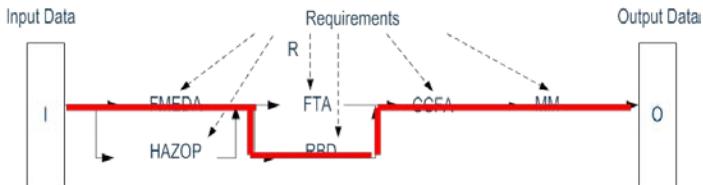


Fig. 17.19 – Example: usage of FMEDA, RBD and MM combination

Fig. 17.17 shows usage of FTA method without any combinations, this is a traditional approach. Fig. 17.18 presents possible combination of RBD and Markov models, where Markov models are used to obtain quantitative results from qualitative RBD model. Fig. 17.19 adds preliminary FMEDA analysis to the path shown in Fig. 17.18, so as to construct RBD model more precisely considering previously analyzed failure modes.

To take into account external impacts it is possible to use IMECA described earlier in this module.

Results of FMECA and IMEA are used during further FTA (Fault Tree Analysis), RBD / SBD (Reliability / Safety (Security) Block Diagram), CCF (Common Cause Failure Analysis), and also during Markov modeling.

Reliability block diagram (RBD) is a graphical analysis technique, which expresses the concerned system as connections of a number of components in accordance with their logical relation of reliability. Safety (security) block diagram (SBD) is a similar technique that treats safety (security) aspects.

Fig. 17.20 shows RBD and SBD principles. Set of FPGA-based ICS components is split into the following groups:

- components that can't lead to FPGA-based ICS failure C_w ;
- components that can lead to ICS failure, but system state would be safe C_{nws} ;
- components that can lead to ICS failure, but system state would be unsafe C_{unws} .

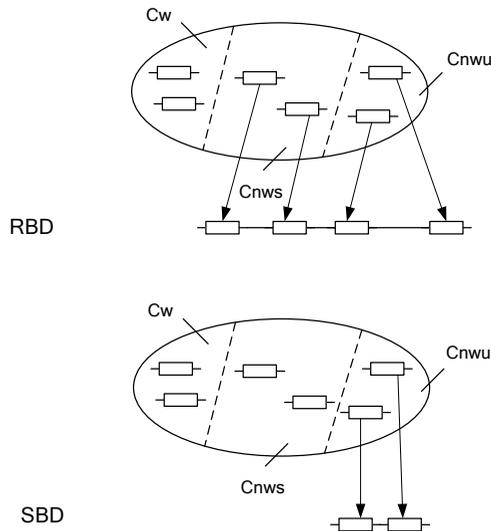


Fig. 17.20 – Reliability and safety (security) block diagrams: principles of development

While RBD treats all possible failures (both C_{nws} and C_{nwu} are included into RBD), SBD treats only components that can lead to unsafe (or unsecure) situation (only C_{nwu} are included). That gives possibility to concentrate on safety (security) aspect and to simplify all following calculations.

During RBD (SBD) it is possible to use list of all components that can cause ICS failure which has been obtained during FMECA. Then ICS architecture (number of components, software and hardware versions, type of diversity, check and reconfiguration means) and sets of different faults must be taken into account so as to calculate safety and security indicators.

17.9. Conclusions

A problem of ICS analysis and assessment is still challenging due to the fact that such systems consist of interconnected complex components with different functions and different nature. The majority of modern ICSs are being FPGA-based; hence, it is impossible to perform their assessment without consideration of all specific details, including interference of various SCS's attributes and the special features for all the technologies used. In this module, some problems related to assessment of safety and security aspects of ICSs were discussed, including problems and features of security environment establishment process, describing in sufficient details its particular stages.

To assess dependability and safety of FPGA-based ICS, it is not enough to use only one assessment method. Combined usage of different methods and further methods' enhancements are possible solutions. Elements of such methods' usage were presented and discussed in this module.

Questions to self-checking

1. Which techniques could be used for ICS safety and security analysis?
2. Which FPGA features is important to consider during ICS safety and security assessment?

3. For which purpose DLC shall be analyzed during ICS assessment?
4. What is an idea behind application of process-product approach in ICS assessment?
5. What is the main idea in method of gap-and-IMECA-based assessment for FPGA-based ICS?
6. Which combinations of methods could be used to perform comprehensive FPGA-based ICS safety and security analysis?
7. What for safety and security block diagrams used?

References

1. V. Kharchenko, V. Sklyar (edits), FPGA-based NPP instrumentation and control systems: Development and safety assessment. RPC "Radiy", National Aerospace University "KhAI", State STC on Nuclear and Radiation Safety, Kharkiv-Kirovograd, Ukraine (2008) 380 p.
2. N.G. Leveson, The need for new paradigms in safety engineering. In book "Safety-critical systems: Problems, process and practice" by C. Dale and T. Anderson. Springer London (2009) 3-20.
3. L. Tong and X. Cao, Methodology for reliability allocation based on fault tree analysis and dualistic contrast, Nuclear Science and Techniques 19 (4) (2008) 251-256.
4. H. Yoshikawa, Distributed HMI system for managing all span of plant control and maintenance, Nuclear engineering and technology, 41 (3) (2009) 237-246.
5. V. Kharchenko, O. Illiashenko, A. Kovalenko, V. Sklyar, A. Boyarchuk. Security Informed Safety Assessment of NPP I&C Systems: GAP-IMECA Technique // Proceedings of 22nd International Conference on Nuclear Engineering, Volume 3: Next Generation Reactors and Advanced Reactors; Nuclear Safety and Security. – 2014
6. A. Gorbenko, V. Kharchenko, O. Tarasyuk, A. Furmanov, F(I)MEA-technique of web services analysis and dependability ensuring, Lecture Notes in Computer Science, 4157/2006 (2006) 153-167.
7. E. Babeshko, A. Gorbenko, V. Kharchenko, Applying F(I)MEA-technique for SCADA-based industrial control systems dependability assessment and ensuring, Proceedings of IEEE DepCoS-

RELCOMEX Conference, June 26-28, Szklarska Poreba, Poland (2008) 309-315.

8. V. Kharchenko, M. Yastrebenetsky (ed.) Nuclear Power Plant Instrumentation and Control Systems for Safety and Security. – IGI Global, 2014. – 450 p.

9. V. Kharchenko, A. Kovalenko, O. Siora, V. Sklyar. Security assessment of FPGA-based safety-critical systems: US NRC requirements context // Proceedings of 2015 International Conference on Information and Digital Technologies (IDT), pp.132-138

10. V. Kharchenko, A. Kovalenko, V. Sklyar. Secure environment establishment for FPGA-based safety-critical systems // Proceedings of East-West Design & Test Symposium (EWDTS) 2015 IEEE, pp. 1-5, 2015.

11. CJ. Clark. FPGA Security, FPGA Configuration, FPGA Bitstream, FPGA Authentication. Business Considerations for Systems with RAM-Based FPGA Configuration. Intellitech, 2009

12. S. Drimer. Security for volatile FPGAs. Technical Report UCAM-CL-TR-763. – University of Cambridge, 2009

13. M. Majzoobi, F. Koushanfar, M. Potkonjak. FPGA-oriented Security. Introduction to Hardware Security and Trust. – Springer, 2011.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ К РАЗДЕЛУ 17

CCFA – Common Cause Failure Analysis
COTS – Commercial Off-The-Shelf
DLC – Development Life Cycle
FPGA – Field Programmable Gate Array
FMEA – Failure Modes and Effects Analysis
FMECA – Failure Modes, Effects and Criticality Analysis
FMEDA – Failure Modes, Effects and Diagnostics Analysis
FTA – Fault Tree Analysis
HAZOP – Hazardous Operations Analysis
ICS – Industrial Control Systems
IMECA – Intrusion Modes, Effects and Criticality Analysis
RBD – Reliability Block Diagram
SBD – Safety (Security) Block Diagram
V&V – Verification and Validation

АННОТАЦИЯ

В разделе рассмотрены аспекты оценки функциональной и информационной безопасности промышленных систем управления, включая вопросы процесса установки безопасного окружения и описания его определенных этапов. Рассмотрены методы оценки функциональной и информационной безопасности промышленных систем управления, основанных на FPGA. Представлены возможности совместного использования методов.

У розділі розглянуто аспекти оцінки функціональної та інформаційної безпеки промислових систем управління, включаючи питання встановлення безпечного середовища і опис його певних етапів. Розглянуто методи оцінки функціональної та інформаційної безпеки промислових систем управління, заснованих на FPGA. Представлено можливості спільного використання методів.

In this module, some problems related to assessment of safety and security aspects of ICSs were discussed, including problems and features of security environment establishment process, describing in sufficient details its particular stages. Methods of safety and security assessment of FPGA-based ICS are discussed. Possibilities of method combinations are presented.

18 METHODS AND TECHNIQUES OF MULTI-VERSION INDUSTRIAL CONTROL SYSTEMS CYBER SECURITY ASSESSMENT AND ASSURANCE

18.1 Diversity for security: case assessment for FPGA-based safety-critical systems

18.1.1 Four challenges for I&C safety assessment and assurance

Industrial safety critical instrumentation and control systems (I&Cs) such as reactor trip systems, on-board aviation systems, railway blocking and signaling systems, etc. are facing more with information (in general and cyber, in particular) security threats and attacks. It concerns most sensitive in point of view safety nuclear domain [1]. Nowadays there is a gap in understanding how to assess safety of industrial I&Cs considering the following:

- firstly, the security issues; security related threats are more and more challengeable for safety critical application. As a result security informed safety conception is intensively developed the last years, in particular for NPP I&Cs [2];
- secondly, the features of FPGA technology and FPGA-based systems as a specific target for intruders. Security aspects for FPGA design and implementation are analyzed in [3-5]. These works allow to systemize different vulnerabilities and threats, and better to understand which of them should be taken into account to assure security;
- thirdly, an application of diversity approach as a mean of minimizing common cause failure risks. In this case two (or more) channels are used in different combinations for obtaining the needed functionality and ensuring of required level of safety. Techniques of development and safety assessment of FPGA-based multi-version industrial systems (MVI&Cs) are researched in [6-8]. However, it is required to analyze influence and features of diversity application in point of view security;

- fourthly, using of case-based proved paradigm. Really, to assure trustworthiness of security assessment for such extremely complex systems, more formalized (and independent in sense of expert errors and uncertainties) techniques are required.

18.1.2 Diversity for safety and security of FPGA-based I&Cs

Diversity is a part of more general principle D3 (Defense-in-Depth&Diversity) [8] applied to provide trusted, fault- and intrusion-tolerant design and operation of I&Cs. Defense-in-Depth is a horizontal/sequential echelon of defense, diversity is a vertical/parallel part of once [11].

18.1.2.1 Diversity related standards for safety and security

There are a lot of international standards and national guides containing requirements for implementation and assessment of diversity. Among them are:

- a) IEC standards:
 - IEC 61513:2001. NPPs - I&Cs important to safety – general requirements for systems;
 - IEC 60880 2006. NPPs - I&Cs important to safety - SW aspects for computer-based systems performing category A functions;
 - IEC 61508 :2011. Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems;
- b) IAEA standards :
 - IAEA NS-G-1.1:2001. Software for Computer Based Systems Important to Safety in NPPs;
 - IAEA NS-G-1.3:2002. I&Cs important to safety in NPPs;
 - IAEA NP-T-1.5:2009. Protecting against CCFs in Digital I&C Systems of NPPs ;
- c) IEEE and NUREG (USA) standards :
 - IEEE std.7-4.3.2:2003. IEEE standard criteria for digital computers in safety systems of NPPs;
 - NUREG/CR-7007:2009. Diversity Strategies for NPP I&C Systems, NUREG/CR-7007 ORNL/TM-2009/302.
- d) National guides and norms :

- DI&C-ISG-02, Diversity and Defense-in-Depth Issues, Interim Staff Guidance (USA);
- BTP 7-19, Guidance for Evaluation of D&DiD In Digital I&C Systems (USA);
- NP 306.5.02/3.035. Requirement on nuclear and radiation safety for I&Cs important to safety in NPPs (Ukraine), etc.

There are standards for other critical domains where diversity as an approach is postulated or requirements to its application are described. For example, requirements to diversity for automotive systems are determined by standard IEC 26262. This standard contains requirements regarding application of software and hardware diversity for on-board vehicle systems.

Generally, the standards are not enough detailed to make all necessary decisions concerning diversity: type of diversity selection and combining, process and product diversity volume assessing and grounding, etc. It is very important that they do not take into account two issues :

- features of FPGA technology what complicates their application and
- security issues for safety assessment.

18.1.2.2 Assessment of safety and security of FPGA-based I&Cs

18.1.2.2.1 Comparison of diversity for SW- and FPGA-based I&Cs

FPGA-based technology provides new possibilities for implementation of diversity principle and additional options [7, 8]. The features of FPGA technology increase a number of diversity kinds and enlarge a set of possible diversity-oriented decisions.

General diversity classification scheme was presented by "cube of diversity" with three coordinates: "stage of the life cycle" – "level of project decisions" and "type of version redundancy" [8]. Using this classification we can analyse safety and security issues for FPGA-based systems and traditional SW-based I&Cs, first of all, for NPPs.

Table 18.1 summarizes variety of diversity attributes from NUREG-CR/7007:2009 for NPP I&Cs and their accordance with kinds of version redundancy of FPGA-based systems.

Table 18.1. Diversity attributes and correspondent FNI&Cs version redundancy kinds.

DIVERSITY ATTRIBUTES (NUREG-CR/7007:2009)	KINDS OF VERSION REDUNDANCY (FPGA-BASED I&Ss)
Design	Diversity of electronic elements (EE)
Different technologies	Different manufacturers of EEs; Different technologies of EEs production
Different approaches within a technology	Different technologies of EEs production
Different architectures within a technology	Different families of EEs
Equipment Manufacturer	Diversity of electronic elements (EE)
Different manufacturers of fundamentally different equipment designs	Different manufacturers of EEs
Same manufacturer of fundamentally different equipment designs	Different families of EEs
Different manufacturers of same equipment design	Different manufacturers of EEs
Same manufacturer of different versions of the same equipment design	Different EEs of the same family
Logic Processing Equipment	Diversity of project development languages
Different logic processing architectures	
Different logic processing versions in same architecture	
Different component integration architectures	Joint use of graphical scheme language and hardware description language (HDL)
Different data flow architectures	Joint use of graphical scheme language and HDL
Function	Diversity of CASE-tools
Different underlying mechanisms to accomplish safety function	Combination of couples of diverse CASE tools and SSs
Different purpose, function, control logic, or actuation means of same underlying mechanism	Different SSs
Different response time scale	
Life-Cycle	Diversity of CASE-tools
Different design companies	Combination of couples of diverse CASE-tools and HDLs
Different management teams within the same company	Combination of diverse CASE-tools and HDLs

DIVERSITY ATTRIBUTES (NUREG-CR/7007:2009)	KINDS OF VERSION REDUNDANCY (FPGA-BASED I&Ss)
Life-Cycle	Diversity of CASE-tools
Different designers, engineers, and/or programmers	Different HDLs
Different implementation/validation teams	
Signal	Diversity of CASE-tools, Diversity of scheme specification (SS)
Different reactor or process parameters sensed by different physical effect	Combination of couples of diverse CASE tools and SSs
Different reactor or process parameters sensed by the same physical effect	
The same process parameter sensed by a different redundant set of similar sensors	
Logic	Diversity of CASE-tools, Diversity of scheme specification (SS)
Different algorithms, logic, and program architecture	Combination of couples of diverse CASE-tools and HDLs
Different timing or order of execution	Different CASE tools configurations
Different runtime environments	Different CASE tools
Different functional representations	Different HDLs

18.1.2.2.2 Diversity and security

Table 18.2 shows results of research on diversity attributes from NUREG-CR/7007 which could be applied to mitigate CCF in diverse SW- and HW/FPGA-based systems with the same vulnerabilities in both versions. Different vulnerabilities in both versions have four grades: VH – very high, H –high, M – medium, L – low.

Gradation is based on risk reduction after appliance of a certain diversity attribute. In this case diversity is considered as a countermeasure for elimination of harmful consequences after successful attacks.

Table 18.2. Diversity attributes as a countermeasure.

DIVERSITY ATTRIBUTES (NUREG-CR/7007:2009)	VULNERABILITIES			
	Software		Hardware	
	common vulnerability	different vulnerabilities	common vulnerability	different vulnerabilities
Design				
Different technologies	H	H	H	H
Different approaches within a technology	M	M	M	M
Different architectures within a technology	L	L	L	L
Equipment Manufacturer				
Different manufacturers of fundamentally different equipment designs	H	H	H	H
Same manufacturer of fundamentally different equipment designs	HM	HM	HM	HM
Different manufacturers of same equipment design	M	M	M	M
Same manufacturer of different versions of the same equipment design	L	L	L	L
Logic Processing Equipment				
Different logic processing architectures	H	H	H	H
Different logic processing versions in same architecture	HM	HM	HM	HM
Different component integration architectures	M	M	M	M
Different data flow architectures	L	L	L	L
Function				
Different underlying mechanisms to accomplish safety function	H	H	H	H
Different purpose, function, control logic, or actuation means of same underlying mechanism	M	M	M	M
Different response time scale	L	L	L	L
Life-Cycle				
Different design companies	H	H	H	H
Different management teams within the same company	HM	HM	HM	HM
Different designers, engineers, and/or programmers	M	M	M	M
Different implementation/validation teams	L	L	L	L
Signal				
Different reactor or process parameters sensed by different physical effect	H	H	H	H

DIVERSITY ATTRIBUTES (NUREG-CR/7007:2009)	VULNERABILITIES			
	Software		Hardware	
	common vulnerability	different vulnerabilities	common vulnerability	different vulnerabilities
Different reactor or process parameters sensed by the same physical effect	M	M	M	M
The same process parameter sensed by a different redundant set of similar sensors	L	L	L	L
Logic				
Different algorithms, logic, and program architecture	H	H	H	H
Different timing or order of execution	HM	HM	HM	HM
Different runtime environments	M	M	M	M
Different functional representations	L	L	L	L

18.1.2.3 Diversity as a countermeasure

Table 18.3 summarizes some attacks on FPGA-based I&Cs and results of security assessment using IMECA-analysis [2,8]. Countermeasures are employed to thwart such tampering attacks. The table contains countermeasures strategies which could be applied as requirements from Regulatory Guide 5.71:2010 (Cyber Security Programs For Nuclear Facilities, U.S. NRC) to eliminate the attack causes and, moreover, FPGA-based MV I&Cs diversity kind and its attributes as a countermeasures.

Thus diversity of FPGA-based MV I&Cs is reviewed as a countermeasure and mitigation strategy for ensuring of security and safety of systems. Criticality matrix (see Fig. 18.1) shows how application of different FPGA-based I&Cs diversity kinds and its attributes will decrease the level of overall risk.

Table 18.3. IMECA-analysis of attacks on FPGA-based I&Cs.

No	Attack mode	Attack nature	Attack cause	Occurrence probability	Effect severity	Type of effects	Countermeasures (including RG 5.71)	FPGA-based I&C diversity kinds and its attributes
1	Readback	Active	Absence of chip security bit and/or availability of physical access to chip interface (e.g., JTAG)	M	H	Obtaining of secret information by adversary	<ul style="list-style-type: none"> • The use of security bit; • Application of physical security controls; (B.1.18 Insecure and Rogue Connections, Appendix B to RG 5.71, Page B-6) 	<u>Diversity of (EE):</u> <ul style="list-style-type: none"> • Different technologies of EEs production
2	Cloning	Active	Storing of decoded configuration	H	H	Obtaining of configuration data by adversary	<ul style="list-style-type: none"> • Checking of chip's internal ID before powering up an electronic design; • Encoding of configuration file; • Storing of configuration file within FPGA chip (requires internal power source) 	<u>Diversity of EE:</u> <ul style="list-style-type: none"> • Different technologies of EEs production; • Different element kinds of EE families
3	Brute force	Active	<ul style="list-style-type: none"> • Search for a valid output attempting all possible key values; • Exhaustion of all possible logic inputs to a device in order; • Gradual variation of the voltage input and other environmental conditions 	L	M	Leak of undesirable information	Detecting and documenting unauthorized changes to software and information, (C.3.7, Appendix C to RG 5.71, Page C-7)	<u>Diversity of project development languages</u> <ul style="list-style-type: none"> • Combination of couples of diverse CASE-tools and HDLs

No	Attack mode	Attack nature	Attack cause	Occurrence probability	Effect severity	Type of effects	Countermeasures (including RG 5.71)	FPGA-based I&C diversity kinds and its attributes
4	Fault injection (glitch)	Active	<ul style="list-style-type: none"> Altering the input clock; Creating momentary over- or under-shoots to the supplied voltage 	M	H	<ul style="list-style-type: none"> Device to execute an incorrect operation Device left in a compromising state Leak of secret information 	<ul style="list-style-type: none"> Making sure all states are defined and at the implementation level, verifying that glitches cannot affect the order of operations; Detection of voltage tampering from within the device; Clock supervisory circuits to detect glitches 	<u>Diversity of EE:</u> <ul style="list-style-type: none"> Different manufacturers of EEs; Different technologies of EEs production; <u>Diversity of SS</u> <ul style="list-style-type: none"> Different SSs; Combination of diverse CASE tools and SSs

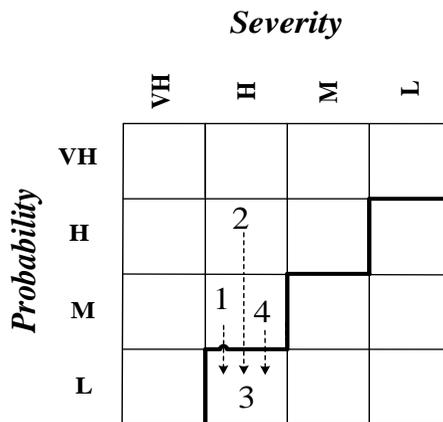


Figure 18.1. Criticality matrix

18.1.3 Security case development

18.1.3.1 Advanced security assurance case

The idea of cybersecurity case for evaluation of security of MV I&Cs lays in applying of Advanced Security Assurance Case ASAC proposed by [9] which is built taking into account requirements to version kinds of systems.

DRAKON was used as a graphical modeling language for representation of cybersecurity case based on ASAC. It was developed from former USSR space program Buran (analogue of Space Shuttle). DRAKON, stands for "friendly algorithmic language that provides clarity." Initially DRAKON was developed for capturing requirements and building software that controls spacecraft [10]. As a language of requirements modeling was chosen IDEF0 notation. Notation IDEF0 allows to show the steps of the evaluation unambiguously (in the form of a directed graph), for each step to determine the evaluated property and evidences necessary for the evaluation, the subjects of assessment, and standards.

If the assessment is subject to a complex (composite) requirement, so each step (or block of IDEF0-diagram) can be decomposed for a detailed description of sub-properties evaluation procedure.

18.1.3.2 Building of ASAC

The result of the analysis of requirements of assurance class "Vulnerability analysis" AVA_VAN.3 from International Standard ISO/IEC 15408 is presented in the form of ontological graph (see Fig. 18.2). The graph accurately and unambiguously (in the accepted notation) describes the subject area (i.e. basic notions/concepts and relations between them). It contains diversity requirements for ensuring of cybersecurity of I&Cs (as countermeasures, Table 18.3) marked in light-blue fillings.

Completeness of scope of assessment is ensured by using ontological graphs of two kinds of object-oriented and process-oriented ontology. Requirements of assurance class "Vulnerability analysis" AVA_VAN.3 from IEC 15408 are depicted in form of properties (Fig. 18.3), evidences (Fig. 18.4) and corresponding actions of an expert

(Fig. 18.5) as results of ontological analysis of diversity requirements for secure I&Cs (marked with blue and dark-blue) and represented in established ASAC form on figure.

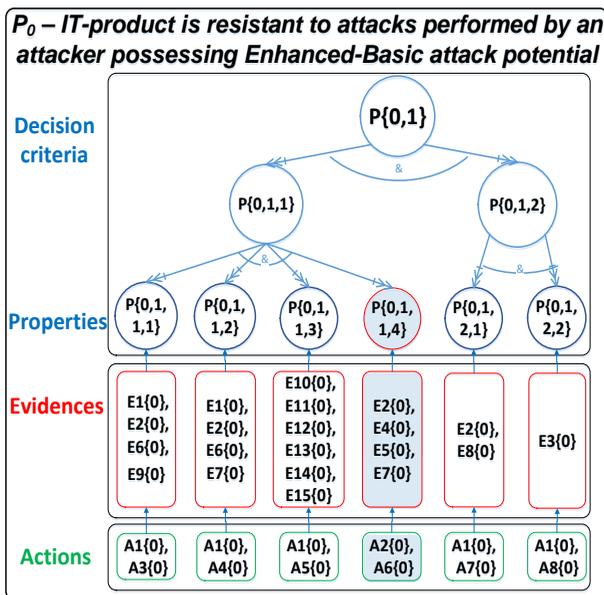


Figure 18.2. Ontological model in form of graph

P _i {j}	PROPERTIES
P{0}	Resistant of the TOE to attacks performed by an attacker possessing Basic attack potential
P{0,1}	Readiness of the TOE for testing
P{0,1,1}	Consistent of the TOE with ST
P{0,1,1,1}	Conformity of the TOE reference with the CM capabilities (ALC_CMC) sub-activities and ST introduction
P{0,1,1,2}	Consistent of the all TOE configurations with ST
P{0,1,1,3}	Conformity of the testing environment to the security objectives for the operational environment described in the ST
P{0,1,1,4}	Conformity of the TOE to diversity requirements
P{0,1,2}	Accuracy of the TOE installing
P{0,1,2,1}	Successfulness completion of the AGD_PRE.1
P{0,1,2,2}	Successfulness of the TOE install and start up, using the supplied guidance only

Figure 18.3. Properties of ASAC represented in tabular form

Ei{J}	EVIDENCES
E1{0}	TOE is suitable for testing
E2{0}	Security Target
E3{0}	Guidance documentation
E4{0}	Information is publicly available to support the identification of potential vulnerabilities
E5{0}	Current information regarding potential vulnerabilities (e.g. from an evaluation authority)
E6{0}	Basic functional specification
E7{0}	Security architecture description
E8{0}	Implementation representation of the TSF
E9{0}	Basic modular design
E10{0}	Applicability of different technologies of EEs production
E11{0}	Applicability of different element kinds of EE families
E12{0}	Applicability of different manufacturers of EEs
E13{0}	Applicability of different Ss
E14{0}	Applicability of combination of diverse CASE tools and Ss
E15{0}	Applicability of combination of couples of diverse CASE-tools and HDLs

Figure 18.4. Evidences of ASAC represented in tabular form

Ai{J}	ACTIONS
A1{0}	Obtain the evidences
A2{0}	Obtain the diversity applicability evidences
A3{0}	Check the conformity of the TOE reference with the CM capabilities (ALC_CMC) sub-activities and ST introduction
A4{0}	Check the consistent of the all TOE configurations with ST
A5{0}	Check the conformity of the testing environment to the security objectives for the operational environment described in the ST
A6{0}	Check the conformity of the TOE to diversity requirements
A7{0}	Check the successfulness completion of the AGD_PRE.1
A8{0}	Check the successfulness of the TOE install and start up, using the supplied guidance only

Figure 18.5. Actions of ASAC represented in tabular form

18.2 Hardware Diversity and Modified NUREG/CR-7007 Based Assessment of NPP I&C Safety

18.2.1 Cases of Hardware diversity application

A. Soft-core and hardwired processors

These two types of processors could be used for the same tasks, but their structure is different in terms of realization. The main differences between them are as follows:

- *Speed.* Hardwired processors - 100's of MHz up to 1GHz+), soft-core processors - 250MHz and less (usually less than 200MHz). Hardwired processors will achieve much faster processing speeds since they are optimized and not limited by fabric speed;

- *Modification.* Hardwired processors are fixed and cannot be modified (though it can take advantage of custom logic in FPGA fabric for processing). Soft-core can be easily modified and tuned to specific requirements, contain more features, custom instructions, etc.

- *Multi-core.* Soft-core processors can be used with multiple cores.

- *Power efficiency.* Hardwired processors tend to be more energy efficient than soft-cores. As an example, unused gates in an FPGA can sometimes be turned off, but usually there are far more active circuits in a soft-core processor than in a purpose-design hardware processor. All of that will lead to unwarranted energy consumption.

- *Cost.* Specialized hardware processor applied for a specific task will cost less than its soft-core implementation. Implementing a processor in FPGA is very resource intensive, particularly if there is a need for highly intensive computing power. The equivalent hardware processor is much cheaper.

Here are several diverse decisions based on the soft-core processor Nios and its hardware analogue:

- Comparison of the task execution time for Nios and VHDL for the FPGA and human resource costs. As an example, the task execution time of an Ethernet network controller using the Lan91C111 chip, controlled by the Nios soft-core processor and the task execution time for a network controller module written in VHDL.

- Implementation of a truncated version of the LAN controller (without TCP / IP support except Ping) will take about 2 years of human

resources. Using soft-core processor and LAN91C111 help solve this task within a week with full support of all the main communication protocols.

- The performance of network packages is measured in milliseconds. The processing time of 1 package for a soft-core processor is up to 100 μ s, although on hard logic this process is many times faster. For the Ping command, the response time is less than 1 millisecond and is the smallest unit of measure, so the reaction time is negligible.

- The VHDL program has extremely poor configuration flexibility for various tasks. Depending on the size of the data (the number of frames), it is necessary to form a different bandwidth of the data bus, and memory buffer size. Each task has its own timing principle, which also poses a problem in the system flexibility.

B. Diversity of data transmission methods and interfaces

As an example, the interface RS485 could be reviewed – high reliability, a large number of consumers on a single data bus, but a relatively low rate of exchange and limited distance. The Address / Data bus is high speed, the distance is limited by internal FPGA connections, or at best by a printed circuit board, which has a very negative effect on the reliability of information transfer.

C. Cyclic redundancy check (CRC)

There is a question on feasibility of using CRC-64 with respect to lower modes, for example, CRC-16. In practice, the most common data bus is 8 or 16 bits, and in rare cases, 32 bits. This is because ADCs operate mainly in 8-16 bit mode. When transmitting, a date, there is no need to transmit the day with a 32-bit number. In practice, it is more viable transmitting the data of 2 ADC channels (total 32 bits) using CRC-64. Typically, high costs are associated with large data packages due to the following reasons. Take into account the time spent on calculating the checksum for a huge incoming dataset, generating a reciprocal array, calculating the checksum for the array, and transmitting it back. Under tight time constraint (i.e. 10 milliseconds for polling of 14 modules or more), time can be critical.

D. Self-diagnostics

Self-diagnostics is reviewed based on a program tabular method for calculating the CRC checksum, software and hardware method for calculating the CRC polynomial:

- Use the table method in program calculation of CRC-8 program memory, followed by a comparison of the result with the constant formed by the TCL-script at the stage of Quartus compilation.
- A hybrid software-hardware method for counting CRC-8 using Nios and a hardware calculation module using polynomial method and linking them via input-output ports.
- Comparative analysis of the speed for generating checksum values using a range of methods.

E. Diversity as a method for performance

This section provides a brief information of the application of diversity principle for the performance gain due to solving different tasks:

- Use a soft-core processor, to solve mathematical formulas and process some data arrays is not always optimal from the point of view of productivity.

- DMA (Direct Memory Access): DMA is invoked when connecting 2 modules using RS485 protocol. The processor frequency is 48Mhz, the transmission speed is 2Mbit per second. To ensure maximum data transfer speed, one must expect the transmission of each byte to fill the transmit buffer with the next byte, or use interrupts, which is highly undesirable in the nuclear industry, or use DMA for parallel transfer (copying) of data to the transmission, while the processor solves other problems. The processor is only involved in programming the DMA controller.

- In some modules, especially those using analog data DAC (Digital-to-analog converter), ADC (Analog-to-digital converter), ADC error correction must be removed in the input data. It should be turned to a single data range and linearized depending on the selected sensor type. A huge list of calculations using multiplication and division will take place. A hybrid method solves the problem. Complex calculations are coded in hard logic. Soft-core processor Nios only sets the initial data and takes the results. The excellent performance and flexibility will be achieved.

F. Diverse softcore 16-bit and 32-bit processor Nios

- Different number of processor commands compiles different code.
- Differences in data calculation registers and data bus.

G. Microcontroller and FPGA as a diverse systems

- In the microcontroller, it is not possible to use a hybrid method for solving tasks. Sometimes this possibility is presented, but it is very limited in hardware.

H. Diverse Avalon Interfaces in Quartus 9 and Quartus 16

- The performance of the new bus is increased by eliminating several bus signals from the bus (Ready/Request, Wait/Request).

- The versatility and flexibility of the new tire has been reduced. The use of DMA in many cases has become impractical.

I. Program code diversity

- Use of pointers relative to access to array elements: different processing times for different sets of commands, and different sets of code for the same output.

- While and for loop: single result, different sets of code, different execution time for the sets of code.

18.2.2 Assessment of hardware diversity

To analyze the impact of hardware diversity types on safety, the technique for assessment should be chosen and/or adapted. There are a several assessment techniques NUREG-A, CLB-A, etc. [21,22].

18.2.2.1 NUREG/CR-7007-based diversity assessment technique

NUREG/CR-7007 [12] presents a method basing on the double level diversity classification. It consists of diversity types and subtypes. The following types of diversity are considered: design diversity as application of different software, FPGA and hardware based approaches; equipment manufacturer diversity as difference in vendors and manufactures of system components; functional diversity as difference in physical functions to perform general task (e.g. shutdown of the reactor); signal diversity as differences in sensed parameters to initiate protective action; life-cycle diversity as involvement of different human resources in appropriate processes for assurance of system safety; logic diversity as differences between systems in terms of algorithms, logic, and program architecture,

timing and/or order of execution, runtime environment, etc.; logic processing equipment diversity as differences in logic processing architecture, logic processing version in the same architecture, component integration architecture, data-flow architecture, etc.

The main tool of diversity assessment technique according with A. NUREG/CR-7007 (NUREG-A) is two level check-list filled by the experts during system analysis [22]. The check-list contains the evidences (column details) supporting variant diversity assessment. The expert marks diversity types DT_i and subtypes DST_{ij} (using value Yes or No) by documentation analysis using a set of special tools. The weight of i-th diversity type WD_i depends on rate of application in I&Cs. Values of metrics MD_i are calculated considering priorities PR_{ij} (importance degree in point of decreasing common cause failure, while $j = \{1, \dots, ND_i\}$, where ND_i is number of diversity subtypes) of diversity subtypes:

$$MD_i (PR_{ij}) = \frac{j}{(1 + 2 + \dots + ND_i)} , \quad (18.1)$$

where j is a number of priority.

General metric of diversity:

$$GMD = \sum_{i=1}^{ND} WD_i MD_i , \quad (18.2)$$

If there are a few diversity subtypes DST_{ij}

$$MD_i = \sum_{j=1}^{ND_i} B_{ij} MD_{ij} , \quad (18.3)$$

where B_{ij} is a Boolean value which is equal to 0, if subtype DST_{ij} is not applied, and is equal to 1 if vice versa.

Such simple technique does not facilitate the calculation of diversity that considers more detailed classifications involving three or more attribute levels [22]. Besides, GMD determines a maximum value because application of any sub-subtypes $DSST_{ijk}$ for diversity subtype DST_{ij} postulates value MD_i .

18.2.2.2 Modification of NUREG/CR-7007 - based technique

The following options of NUREG-A technique development are possible taking into account more detailed specification of hardware diversity classification (for subtypes and sub-subtypes of diversity for logic diversity, logic processing equipment diversity and others):

- Extend the subtypes sets (increasing NDi). In this case, the MDi calculation procedure is the same and reprioritization of diversity subtypes is required. As an example, for a design, equipment manufacturer and logic processing equipment diversity, a few additional subtypes can be added (see Fig. 18.6).

Attribute criteria		Rank	DCE WT
DESIGN	Design		
	Different technologies	1	0,333
	Different approaches within a technology	2	0,267
	Different architectures	3	0,200
	Different FPGA and CPU performing the same tasks	4	0,133
	32-bit)	5	0,067
LOGIC PROC.EQUIP.	Logic Processing Equipment		
	Different logic processing architectures	1	0,286
	Different logic processing versions in same architecture	2	0,238
	Different component integration architectures	3	0,190
	Different data flow architectures	4	0,143
	Different data transmissions methods and interfaces	5	0,095
	Different types of logic implementation: soft-core and hardwired processors (FPGA and Nios)	6	0,048

Figure 18.6. Diversity assessment tool spreadsheet

- Extend the hierarchy of diversity classification (types-subtypes-subtypes). In this case, MDi is presented as maximal value. To calculate a more accurate value it's needed to prioritize sub-subtypes DSSTijk in frame of subtype DSTij. The procedure for prioritization and calculating metrics can be the same as for NUREG-A. In this case the formula for GMD is as follows:

$$GMD = \sum_{i=1}^{ND} WD_i \sum_{j=1}^{ND_i} B_{ij} \sum_{k=1}^{ND_{ij}} B_{ijk} MD_{ijk} , \quad (18.4)$$

where B_{ijk} is a Boolean value for sub-subtypes similar B_{ij} for subtypes.

MD_{ijk} is calculated and is similar $MDi(PRIj)$ (see (18.1)) considering priorities for sub-subtypes $DSST_{ijk}$, see Fig. 18.7.

Attribute criteria		Rank	DCE WT
DESIGN	Design		
	Different technologies	1	0,500
	Different approaches within a technology	2	0,200
	Different architectures	3	0,167
	Different FPGA and CPU performing the same tasks	3.1	0,660
	Different versions of soft-core processor Nios (16-bit and 32-bit)	3.2	0,330
LOGIC PROC. EQUIP.	Logic Processing Equipment		
	Different logic processing architectures	1	0,400
	Different types of logic implementation: soft-core and hardwired processors (FPGA and Nios)	1.1	0,667
	Different data transmissions methods and interfaces	1.2	0,333
	Different logic processing versions in same architecture	2	0,300
	Different component integration architectures	3	0,200
	Different data flow architectures	4	0,100

Figure 18.7. Modified diversity assessment tool spreadsheet

By filling of particular diversity types/subtypes into check-list can automatically reveal corresponding ones (note: expert marks INH = inherent (i) against them). After filling the check-list, the diversity metrics are calculated as sum of weighted values of diversity types/subtypes (attributes and criteria). The diversity metric obtained after calculation is not normalized and can take any values in the range $\{0 - 1.76\}$. In this method, the diversity metric of the value 1.0, is considered as acceptable for two-version I&Cs.

18.3 Diversity for Safety and Security of Embedded and Cyber Physical Systems

18.3.1 Industrial Cases. Diversity for Safety and Security

18.3.1.1 Reactor trip system

Application of diversity approach in safety critical NNP I&C systems is normative requirement of national and international standards [6,8,23]. An example of two-version system is reactor trip system (RTS) based on FPGA platform developed and implemented by RPC Rادی [8,25].

The RTS (Fig. 18.8,a) consists of two identical in point of view functionality and structure systems (main and diverse) connected according with logic 1-out-of-2 (OR). Both systems have M-out-of-N structure as a rule 2-out-of-3, but channels of systems are based on different hardware, FPGA and software designs.

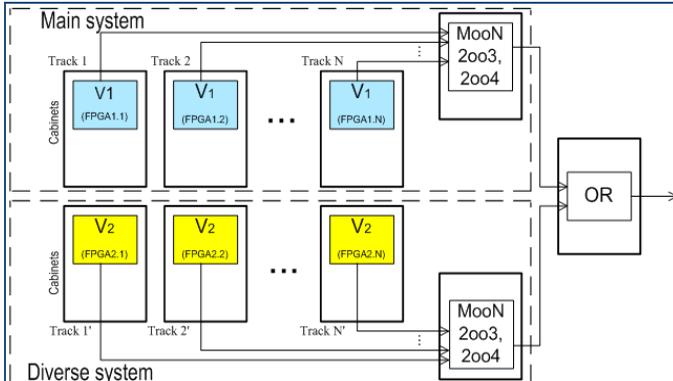
Reliability block diagram of the RTS is shown on the Fig. 18.8,b. This model describes a case with ideal diversity when system versions have not join design faults (components f_{d1} and f_{d2}). Hence such system tolerates design (software or FPGA) faults and physical (hardware and FPGA) faults.

This RBD takes into account common design faults of the versions (red element). They can cause CCF. Detailed research results of this system and RTS with other two-version structure considering version CCF risks are described in [27].

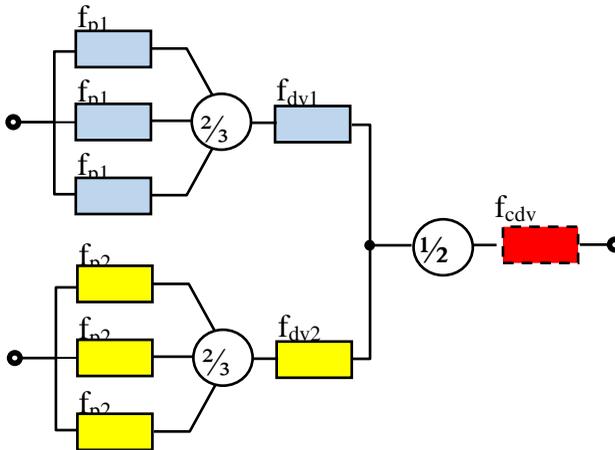
Application of FPGA technology and FPGA-based platforms increase a number of diversity types and enlarge a set of possible diversity-oriented decisions. The following cases are possible and applied in NPP I&C systems:

- Central processing unit CPU1 vs CPU2 (different chips, manufacturers, languages and tools);
- FPGA vs CPU (main system is developed using FPGA, diverse system is developed using microcontroller);

FPGA1 vs FPGA2 (different manufacturers, sub-technologies: SRAM-based, Flash, Anti-fuse, different development and verification techniques and tools).



a



b

Figure 18.8. Structure (a) and reliability block diagram (b) of RTS

Diversity allows improving some attributes of security (integrity and availability) for safety critical systems. Table 18.4 shows how different diversity types (according with classification [6]) can influence on security (integrity) of safety critical system [28]. Fuzzy expert scale of assessment (H – high, HM – high to medium, M – medium, L – low) has been chosen because calculating of quantitative metrics is complex separate task.

Table 18.4. Influence of diversity application on security

Diversity Attributes (NUREG-CR/7007:2009)	<i>Vulnerability mitigation</i>
Design	
Different technologies	H
Different approaches within a technology	M
Different architectures within a technology	L
Equipment Manufacturer	
Different manufacturers of fundamentally different equipment designs	H
Same manufacturer of fundamentally different equipment designs	HM
Different manufacturers of same equipment design	M
Same manufacturer of different versions of the same equipment design	L
Logic Processing Equipment	
Different logic processing architectures	H
Different logic processing versions in same architecture	HM
Different component integration architectures	M
Different data flow architectures	L
Function	
Different underlying mechanisms to accomplish safety function	H
Different purpose, function, control logic, or actuation means of same underlying mechanism	M
Different response time scale	L
Life-Cycle	
Different design companies	H
Different management teams within the same company	HM
Different designers, engineers, and/or programmers	M
Different implementation/validation teams	L
Signal	
Different reactor or process parameters sensed by different physical effect	H
Different reactor or process parameters sensed by the same physical effect	M
The same process parameter sensed by a different redundant set of similar sensors	L
Software	
Different algorithms, logic, and program architecture	H
Different timing or order of execution	HM
Different runtime environments	M
Different functional representations	L

Gradation is based on risk reduction of successful attacks on version vulnerabilities depending on applied diversity types or subtypes.

Hence diversity is considered as a countermeasure for elimination of harmful consequences after successful attacks on vulnerabilities.

It must be emphasized that application of version redundancy can worsen other important security attribute such as confidentiality. This is partly because the intruder can attack one of the systems and access information. Besides, failed (interim) shutdown may be caused such intrusion as well. Hence in this case effect of “weak link in the chain” is possible and must be taken into account.

18.3.1.2 On-board aviation system

Very interesting multi-version structure of on-board flight control system (FCS) has been developed for A-340 and A-380 (Fig. 18.9) [26]. The BCS consists of two diverse systems: primary (PCS) and secondary (SCS). Both systems consist of duplicated subsystems (three and two correspondingly). All five duplicated subsystems have the same two CPU-based channels.

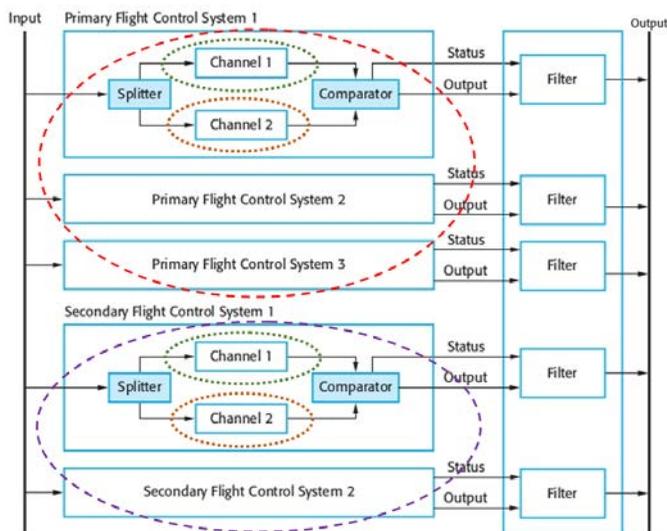


Figure 18.9. Structure of control systems of A340, A380 [26]

Applied diversity types are described by Table 18.5. The following diversity types are implemented:

- Equipment diversity of CPUs (different manufacturers and designs of CPUs for first and second channels Ch1 and Ch2 of all five subsystems);
- Equipment diversity of printed circuit boards (PCB) (different manufacturers and designs of PCSs for channels Ch1 and Ch2 of all five subsystems);
- Software diversity of PCS and SCS (different algorithms, different operation systems (OS) and applied software (ASW)).

Table 18.5. Analyses of diversity types in control system of A340, A380

Systems	Components		Versions		CCF Model
			Ch1	Ch2	
PCS 3×(Ch1, Ch2)	HW	CPU	V _{CPU1}	V _{CPU2}	
		PCB	V _{PCB1}	V _{PCB2}	
	SW	ASW	V _{ASW1}		
		OS	V _{OS1}		
SCS 2×(Ch1, Ch2)	HW	CPU	V _{CPU1}	V _{CPU1}	
		PCB	V _{PSB1}	V _{PSB2}	
	SW	ASW	V _{ASW2}		
		OS	V _{OS2}		

The models of CCF for different diversity types are illustrated by last column of the table. Colors of version fault subsets corresponds to Fig. 18.9.

In contrast to RTS where diversity is divided on two systems (linear-parallel diversity) the FCS is based on so called matrix diversity. This principle of diversity distribution on systems has more complex model of CCF and must be supported high reliable on-line testing. Model for security assessment of such embedded system is more complex as well.

18.3.1.3 On-board vehicle and railway systems

The standard IEC 26262 contains requirement to application of diversity in on-board vehicle computer-based systems [29]. Two types of diversity are described in the standard: hardware based on use of different hardware platforms and software diversity based on use different system or/and applied software.

This huge industry domain hasn't experience on application of the diversity for on-board computer safety critical systems. However taking into account requirements and recommendations of the high level standard they will be implemented [30,31]. In this case experience of other safety critical domains including aviation and NPP I&C systems can be used [32].

In contrast to automotive domain diversity (functional, hardware and software) is applied in railway safety related systems very intensively (see review of design decisions in [33]).

18.3.2 SOA-Based System Case. Diversity for Security

18.3.2.1 Basic approach to architecture development

Diversity can be successfully applied in business systems, in particular, in SOA (service oriented architecture)-based systems. It's explained existing a lot of targeted services with identical functionality and different configuration and components. In this case natural redundancy and diversity can be implemented.

Usually SOA consists of four components: operation system (OS), web-server (WS), application server (AS) and database (DB). Possibilities of use of diverse components in one SOA are formally described by four-level graph.

The functionally identical components are located at the one level, links between nodes (components) describe compatibility of the component [34]. Fig. 18.10 is a fragment of complete graph [35]).

SOA-based multi-version system is developed by selection and configuration of the compatible components linked in two or three different ways at the graph (for two- or three-version systems respectively).

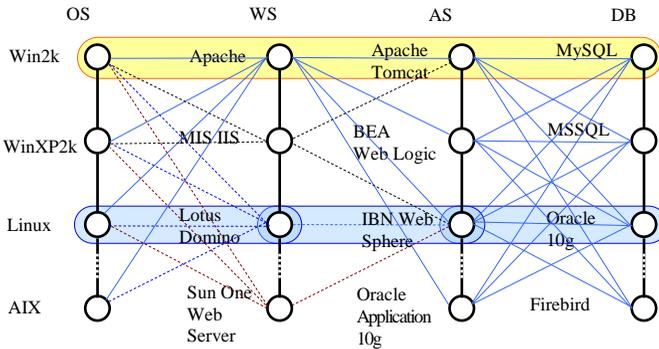


Figure 18.10. Graph of SOA components compatibility and version choice

Every node of the graph relates a set of vulnerabilities. Information about vulnerabilities for commercial software components is acceptable in open databases (NVD and others [35,36]).

18.3.2.2 Security block diagram

Using such information the sets of component and configuration vulnerabilities V_{Ci} can be obtained and used to develop the security block diagram for one SOA version (Fig. 18.11,a). Different configurations (versions) have different sets of vulnerabilities

Hence all pairs of configurations (ways of graph) can be assessed by diversity metrics considering a subset of common vulnerabilities and relation number of common and individual vulnerabilities [35]. For such pairs simplified security block diagram (Fig.18.11,b) takes into account “insecurity” of components and the configuration as a whole (red element).

Thus all diverse configurations are assessed and ranked according with security indicators similar RBD-based reliability assessment. Then complexity, costs and other metrics are calculated and optimal SOA for two-version system can be selected. Due to graph-based description of multi-version life cycle task of MVS development (choice and complexing diversity types) can be formulated and solved as a task of numeration and selection of ways on graph.

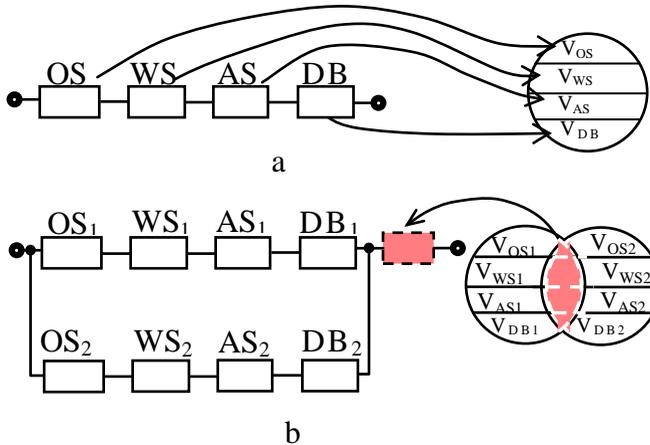


Figure 18.11. Security block diagrams for one-(a) and two(b) version SOA

More detailed technique and tool to assess and configure the best architecture according with criterion “security-cost” and taking into account reliability and security metrics of configurators and connectors (for separated reservation of components) as well, attack profile and possibilities of dynamical reconfiguration of multi-version SOA in clouds, Markov’s models and benchmarking experiences are described in [35,37,38].

These results confirm effectiveness of application of diversity approach in web- and cloud-based business systems and possibilities realization of some safety related functions using such technologies.

18.3.3 Industrial Case. Diversity for Survivability

18.3.3.1 Post-accident monitoring system for critical infrastructure

Diversity principle are applied to assure safety, security and survivability more complex cyber physical systems such as smart energy grid including digital substations and NPPs [40], pre- and post-

accident monitoring system of nuclear reactor and power plant as a whole.

After Fukushima emergency the implementation of reliable and survivable post-accident monitoring systems (PAMS) is requirement of national and international regulatory bodies. PAMSs are necessary for other critical infrastructures (chemical enterprises, oil-gas transport systems and so on).

Existed NPP PAMSs are based on wired networks (WRN) connecting sensor area with the crisis centre. Reliability and survivability of such systems are assured by redundancy of equipment, cable communications and other components. In case of severe accident WRN-based PAMS can be added by wireless network (WLN) more resilient to physical failures.

To assure stable work of WLN-based PAMS subsystem after accident in conditions of powerful jamming a special means are required to support reliable transmission of data considering probable failures of WRN. For that and to improve survivability of PAMS introduction of drone fleet subsystem (DS) has been suggested in [40]. The structure of integrated WRN&WLN&DS-based PAMS is shown on Fig. 18.12.

The following principles of embedding of drone fleet system functioning are the following.

- The drone fleet is located permanently at a considerable distance from the NPP. The communication network (WLN +DS) is deployed after the accident and drones fly to the accident zone.

- Drones fleet is divided by the role and equipment into: repeaters (Slave), that work together on a principle of “one leader” and if the “leading drone-repeater” (Master) is damaged then other drone-repeater takes Master functions; observers (equipped with a TV camera), that enable to run the continuous visual monitoring of the accident location; additional sensors, that can be located in drones or be dropped down in certain places). Drones should be able to change their role by upgrading equipment at the location base.

- Measurement and control modules are equipped with backup batteries, blocks of wireless communication, as well as, self-testing and self-diagnostic systems.

To meet the system requirements the self-adaptability, self-testing and self-healing procedures are used.

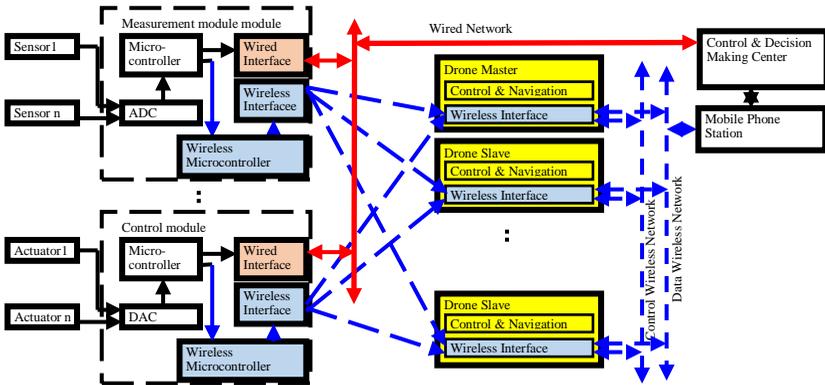


Figure 18.12. The structure of integrated PAMS [39]

18.3.3.2 Reliability and survivability models

Reliability block diagrams for initial WRN-based PAMS (a) and for one of the possible most simple options of integrated system (b) are shown on Fig. 18.13. These models take into account failure of the following elements:

- sensors for WRN and WLN (SeWR and SeWL);
- microcontrollers with AD/DA converters and interfaces for WRN and WLN (M&IR and M&IL relatively);
- WRN and station WRS; WLN,
- drones (DS) and station WLS;
- decision making systems located at the crisis centre.

In fact WLN&DS system of PAMS is diverse for WRN system. Both systems have redundant elements. For first one sliding redundancy of sensors, wireless network and drones are applied. Such flexible redundant structure has more high reliability and survivability because decreases risk of common cause multiple failures [41].

Survivability of integrated PAMS is calculated using diagrams of degradation and combinatorial assessment of probabilities for different states.

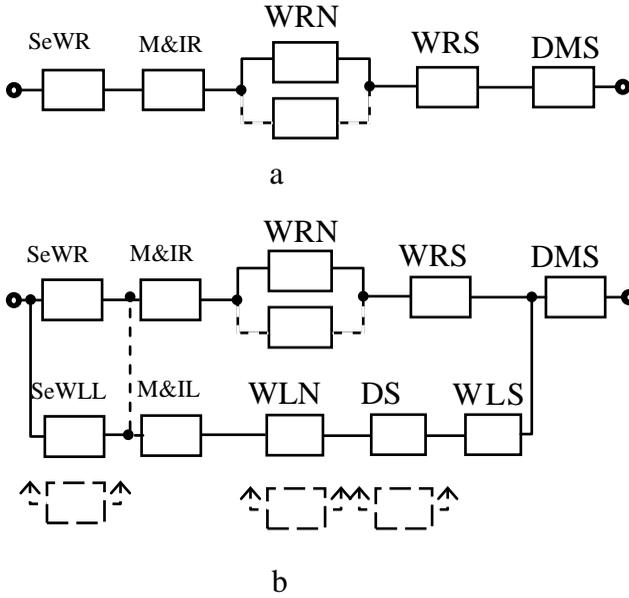


Figure 18.13. Security block diagram of one (a) channel and multi-version (b) SOA

Conclusions

1. The section describes cybersecurity assurance technique of multi-version FPGA-based I&Cs. Requirements profile is formulated using the best practices from the following international regulations. The section summarizes research results on using of security informed safety assessment of FPGA-based MV I&Cs by development of security case based on ASAC. This case considers requirements from Common Criteria and added requirements for diversity as a countermeasure and CCF risk reduction strategy. Security assurance case tends to reducing of uncertainty of safety assessment taking into account influence of security (cybersecurity) to safety.

2. The analysis conducted in this section shows that there is a strong need for the development of new regulation procedures that will cover application of diversity issues which consider new technologies and its interconnections. Existing diversity normative base should be enhanced in a 3 directions – scope, depth and rigor to provide more detailed description of possible applied techniques and tools for quantitative assessment.

The modification of NUREG/CR-7007 based diversity assessment technique has been provided. This section has discussed modified techniques to assess metrics encompassing hardware diversity types. It considers a weighted value of the diversity, while the original technique merely overestimate the assessment.

3. Implementation of diversity and D3 principle is “expensive pleasure” therefore its application must be grounded, actual diversity level must be assessed by quantitative way (or qualitative if assumed by regulator), and required level of diversity or acceptable risks of CCF in developed system must be proved.

Diversity allows improving not only of reliability and safety and security as well. As safety and security are very important and closely related (there is circle “safety-security” via system and environment) application of diversity can assure multiple effect.

On the other side there are limitations of diversity application for improving of some security attributes, for example confidentiality.

Analysis of industrial cases allows concluding that implementation of diversity requires high level of design, verification and maintenance teams.

Questions to self-checking

1. What challenges are there for I&C safety assessment and assurance?
2. Which standards for safety and security are related to diversity?
3. How diversity attributes from NUREG-CR/7007 are related to kinds of versions redundancy for FPGA-based I&Cs?
4. What is gradation of vulnerabilities in both versions of FPGA-based systems based on?
5. What is Advanced Security Assurance Case?
6. What steps are needed to build ASAC?
7. What cases of hardware diversity application are there?
8. What techniques for assessment of hardware diversity are there?
9. How diversity metric is calculated according to NUREG-CR/7007?
10. How can diversity types influence on security of safety critical systems?
11. Which diversity types are implemented in control systems of A340 and A380?
12. How SOA-based multi-version system is developed?
13. How to build security block diagram for one-version and two-versions SOA?
14. What is post-accident monitoring system?
15. How survivability of integrated PAMS is calculated?

References

1. V. Sklyar, Cyber Security of Safety-Critical Infrastructures: A Case Study for Nuclear Facilities, *Information & Security An international Journal*, 28, 1 (2012)
2. V. Kharchenko, O. Illiashenko, A. Kovalenko, et. al. Security Informed Safety Assessment of NPP I&C Systems: GAP-IMECA Technique, ICONE 22, Prague, Czech Republic (2014)
3. B. Badrignans, J. Danger, V. Fischer, G. Gogniat, L. Torres, *Security Trends for FPGAs* (Springer, 2011)
4. T. Huffmire, C. Irvine, T. Nguyen, T. Levin, R. Kastner, T. Sherwood, *Handbook of FPGA Design Security* (Springer, 2010)
5. M. Tehranipoor, C. Wang (edits), *Introduction to Hardware Security and Trust* (Springer, 2012)
6. NUREG/CR-7007 ORNL/TM-2009/302 (2009)
7. V. Kharchenko, V. Sklyar (edits). *FPGA-based NPP Instrumentation and Control Systems: Development and Safety Assessment*, (KhAI , 2008)
8. M. Yastrebenetsky, V. Kharchenko (edits). *NPP I&S for Safety and Security*, (IGI-Global, USA, 2014)
9. O. Illiashenko, O. Potii, D. Komin. *Advanced security assurance case based on ISO/IEC 15408, DepCoS-RELCOMEX*, Brunów, Poland (2015)
10. V. Parondzhanov, *How to improve the work of your mind* (Delo, Russia, 2001)
11. N. G. Bardis, N. Doukas, O. P. Markovski. *Burst Error Correction Using Binary Multiplication without Carry*, MILCOM 2011 Military Communications Conference, Baltimore, MD (2011)
12. United States Nuclear Regulatory Commission., “Diversity Strategies for Nuclear Power Plant Instrumentation and Control Systems”, NUREG/CR-7007, Office of Nuclear Regulatory Research, 2010.
13. IEEE Std 7-4.3.2-2016. *IEEE Standard Criteria for Programmable Digital Devices in Safety Systems of Nuclear Power Generating Stations*.
14. IAEA Safety Standards Series No. SSR-2/1. *Safety of Nuclear Power Plants: Design. Specific Safety Requirements*. International Atomic Energy Agency, Vienna, 2016.

15. IAEA Nuclear Energy Series No. NP-T-3.17. Application of Field Programmable Gate Arrays in Instrumentation and Control Systems of Nuclear Power Plants. International Atomic Energy Agency, Vienna, 2016.

16. ISO 26262:2011. Road vehicles – Functional safety

17. IEC 61508:2010 Ed.2. Functional safety of electrical/electronic/programmable electronic safety-related systems

18. United States Nuclear Regulatory Commission, “Method for Performing Diversity and Defense-in-Depth Analyses of Reactor Protection Systems”, NUREG/CR-6303, Office of Nuclear Regulatory Research, 1994.

19. International Atomic Energy Agency, “Radiation Aspects of Design for Nuclear Power Plants,” IAEA S-G-1.3, Vienna, Austria, 2005.

20. International Electrotechnical Commission, “Instrumentation and Control Systems Important to Safety—Requirements to Cope with Common Cause Failure (CCF),” IEC 62340, Geneva, Switzerland, 2008.

21. V. Kharchenko, O. Siora, V. Duzhyi, D. Rusin, “Standard analysis and tool-based assessment technique of NPP I&C systems diversity”, 22nd International Conference on Nuclear Engineering, 2014.

22. V. Kharchenko, E. Babeshko, K. Leontiev, V. Duzhy, “Diversity for safety and security of NPP I&C: post NUREG/CR 7007 stage”, NPIC-HMIT Proceedings, San-Diego, USA, 2017.

23. IEC 61508-2009. Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems, 2009.

24. R. Bloomfield, K. Netkachova, R. Stroud. Security-Informed Safety: If It’s Not Secure, It’s Not Safe. Proceedings of the 5th WS Software Engineering for Resilient Systems, SERENE2013, Kyiv, Ukraine/A. Gorbenko, A. Romanovsky, V.Kharchenko (edits), Springer, 2013.

25. V. Kharchenko, A. Siora, E. Bakhmach. Diversity-Scalable Decisions for FPGA-Based Safety-Critical I&Cs: from Theory to Implementation. Proceedings of the 6th ANS International Topical Meeting on Nuclear Plant Instrumentation, Controls, and Human Machine Interface Technology, NPIC&HMIT2009, Knoxville, TN, USA: American Nuclear Society, 2009.

26. I. Sommerville. Software Engineeringh, 9th Edition, Addison-Wesley, 2011.
27. V. Kharchenko, A. Volkoviy, O. Siora, V. Duzhyi. Metric-Probabilistic Assessment of Multi-Version Systems: Some Models and Techniques. Dependable Computer Systems: Advances in Intelligent and Soft Computing, Springer, Volume 97, 2011.
28. V. Kharchenko, O. Illiashenko. Diversity for Security: Case Assessment for FPGA-based Safety Critical Systems. Proceedings of the 20th International Conference on Conference on Circuits, Systems, Communications and Computers, CSCC2016, Corfu Island, Greece, 2016.
29. ISO 26262-1-2011. Road vehicles – Functional safety, 2011.
30. D. Negi, N. Bagri, V. Agarwal, Redundancy for Safety-Compliant Automotive and Other Devices, EDN Network, 2014
31. C.Turner Safety and Security for Automotive SoC Design, WS ARM, Seoul-Korea, Taipei-Taiwan, 2016 http://www.arm.com/files/pdf/20160628_B02_ATF_Korea_Chris_Turner.pdf
32. V. Kharchenko. Diversity for Safety of Systems and Software in Context of the Standard ISO/IEC26262. 13th Workshop on Automotive on Software and Systems, Milano, Italy, 2015. <http://www.ices.kth.se/upload/events/76/7577a93992a142f9b6e51b0a40698831.pdf>
33. I. Malynyak. Functional Diversity Design of Safety Related Systems. Proceedings of 11th International Conference ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer, ICTERI-TheRMIT2015, Lviv, Ukraine, 2015.
34. A. Gorbenko, V. Kharchenko, O.Tarasyuk, A. Furmanov. F(D)MEA-Technique of Web-services Analysis and Dependability Ensuring Rigorous Development of Complex Fault-Tolerant Systems, LNCS 4157. Springer. 2006.
35. V. Kharchenko, A. Gorbenko (editors). Web, Grid Cloud Technologies for Dependable IT-Infrastructures.-Project TEMPUS-SAFEGUARD, National Aerospace University KhAI, 2013.
36. National Vulnerability Database, National Institute of Standards and Technologies, USA, 2016 <https://nvd.nist.gov/>

37. V. Kharchenko, Alaa Mohammed Abdul-Hadi, A. Boyarchuk, Yu. Ponochovny. Web Systems Availability Assessment Considering Attacks on Service Configuration Vulnerabilities. Proceedings of DepCoS-RELCOMEX 2014, Brunow, Poland, Springer, 2014.

38. V. Kharchenko, A. Abdul-Hadi, A. Boyarchuk, Yu. Ponochovnyj. Web Systems Availability Assessment Considering Attacks on Service Configuration Vulnerabilities. Seria «Information and Communication Technologies in Education, Research, and Industrial Applications Communications in Computer and Information Science» Vol. 469 / V. Ermolayev et al (edits), Springer International Publishing Switzerland, 2014

39. E. Brezhnev, V. Kharchenko, A. Boyarchuk, J. Vain. Cyber Diversity for Security of Digital Substations under Uncertainties: Assurance and assessment. Proceedings of the 19th International Conference on Conference on Circuits, Systems, Communications and Computers, CSCC2015, Zakynthos Island, Greece, 2015.

40. A. Sachenko, V. Kochan, V. Kharchenko et al. Mobile Post-Emergency Monitoring System for Nuclear Power Plants. Proceedings of the 12th ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer, ICTERI-TheRMIT2016, Kyiv, Ukraine, 2016.

41. V. Kharchenko, A. Sachenko, V. Kochan, H. Fesenko. Reliability and Survivability Models of Integrated Drone-Based Systems for Post Emergency Monitoring of NPPs. Proceedings of the International Conference on Information and Digital Technologies, IDT2016, Rzeszow, Poland, 2016.

42. V. Kharchenko, V. Sklyar, E. Brezhnev, V. Duzhyi. FPGA Platform-Based Multi-Version NPP I&C Systems: Diversity Assessment and Selection of Variants. Proceedings of the 6th ANS International Topical Meeting on Nuclear Plant Instrumentation, Controls, and Human Machine Interface Technology, NPIC&HMIT 2015, Charlotte, NC, USA: American Nuclear Society, 2015.

43. V. Kharchenko, A. Siora, V. Sklyar, A. Volkoviy. Multi-Diversity Versus Common Cause Failures: FPGA-Based Multi-Version NPP I&C Systems. In Proceedings of the 7th ANS International Topical Meeting on Nuclear Plant Instrumentation, Controls, and Human Machine Interface Technologies, NPIC&HMIT 2010, Las-Vegas, Nevada, CA, USA: American Nuclear Society, 2010.

44. A. Kornecki, N. Subramanian, J. Zalewski Studying Interrelationships of Safety and Security for Software Assurance in Cyber-Physical Systems Proceedings of the Federated Conference on Computer Science and Information Systems, 2013.
45. L. Pullum. Software Fault Tolerance Techniques and Implementation, Artech House Computing Library, 2001.
46. B. Littlewood. The Impact of Diversity upon Common Mode Failures. Reliability Engineering and System Safety, Vol.5, 1, 1996.

АННОТАЦИЯ

В разделе описан метод обеспечения кибербезопасности многоверсионных ИУС на основе ПЛИС. В разделе обобщены результаты исследования по применению оценки безопасности МВ ИУС на основе ПЛИС путем разработки примера, основанного на ASAC.

Представлена модификация метода оценки диверсности NUREG/CR-7007. В разделе рассматриваются модифицированные методы расчета метрик, охватывающие виды аппаратной диверсности.

В разделе также представлен анализ реальных систем, который позволяет сделать вывод, что для реализации диверсности требуется высокий уровень команд проектирования, верификации и сопровождения.

У розділі розглянуто метод забезпечення кібербезпеки багатoversійних ІУС на основі ПЛІС. У розділі узагальнені результати дослідження щодо застосування оцінки безпеки БВ ІУС на основі ПЛІС шляхом розробки прикладу на основі ASAC.

Представлена модифікація методу оцінки диверсності NUREG/CR-7007. У розділі розглянуті модифіковані методи розрахунку метрик, що охоплюють види апаратної диверсності.

У розділі також представлено аналіз реальних систем, який дозволяє зробити висновок, що для реалізації диверсності потрібен високий рівень команд проектування, верифікації та супроводу.

The section describes cybersecurity assurance technique of multi-version FPGA-based I&Cs. The section summarizes research results on using of security informed safety assessment of FPGA-based MV I&Cs by development of security case based on ASAC.

The modification of NUREG/CR-7007 based diversity assessment technique has been provided. This section has discussed modified techniques to assess metrics encompassing hardware diversity types.

The section also provides analysis of industrial cases which allows concluding that implementation of diversity requires high level of design, verification and maintenance teams.

MODULE 2 METHODS AND MEANS OF INFORMATION PROTECTION

CONTENTS SECTION

- 2.1 Classification of methods and means of information protection. Services and mechanisms of information protection
- 2.2 Identification and authentication
- 2.3 Typical models of authentication
- 2.4 Access control mechanisms

2.1 Classification of methods and means of information protection. Services and mechanisms of information protection

Totality of methods, facilities and events of information protection consists of software methods, hardware facilities, protective transformations, and also organizational events (fig. 2.1) [1].

Hardware facilities	Software facilities	<i>Protective transformations</i>	Organizational events
---------------------	---------------------	-----------------------------------	-----------------------

Fig. 2.1. Totality of methods, facilities and events of information protection

Essence *of hardware or scheme protection* consists of that in devices and technical equipments of treatment of information a presence is envisaged of the special technical decisions, providing protection and control of information.

Here also take the technical means of information protection (TMIP or TIP). Usually under TMIP technical equipments that is intended for protection to information and are not part of the computer system are understood.

For example: screening devices, localizing electromagnetic radiations or charts odd-even checks of information, carrying out control after the rightness of information transfer between the different devices of the system behave to hardware protection.

Software methods of protection are totality of algorithms and programs, providing differentiation of access and exception of the unauthorized use of information.

Essence of methods *of protective transformations* consists of that information storable in the system and transferrable on communication channels appears in some code, exclusive possibility of her direct use. Also

sometimes the methods of protective transformations attribute to the software (or to the hardware) methods of protection.

Organizational events of protection consist of totality of operating under a selection and verification of personnel participating in preparation and exploitation of the programs and information, strict regulation of development and functioning of the informative system process.

Also the methods of information protection can be classified on functional signs. For example, on the aims of actions, on the types of threats, to direction of providing of protection, on objects, on the levels of scope, on activity [1].

Only the complex use of different protective events can provide reliable protection of information, because every method or means has the weak and strong parties.

Service of security is totality of mechanisms, procedures and other managing influences, realized for reduction of the risk related to the threat [1].

For example, services of identification and authentication help to shorten the risk of threat of the unauthorized user. Some services provide protecting from threats, and other services provide finding out realization of threat. Such services of registration or supervision can exemplify.

We will name some services of security:

- **identification and authentication** are security service guaranteeing, that the only authorized persons work in the informative system;

- **access control** is security service guaranteeing, that resources IS used by the settled method;

- **confidentiality of data and reports** is security service guaranteeing, that data of the informative system, software and reports, are closed for the not authorized persons;

- **integrity of data and reports** is security service guaranteeing, that data of the informative system, software and reports, are not changed by incompetent persons.

- **control of participants of cooperation** is security service guaranteeing, that subjects participating in cooperation will not be able to give up participating in him. In particular, a sender will not be able to deny a parcel reports(control of participants of co-operating with confirmation of sender) or recipient not able to deny receiving message(control of participants of co-operating with confirmation of recipient).

- **registration and supervision** are security by means of that the use of all resources of the informative system can be traced service.

The brought list over of security services it is necessary to examine as possible, but not obligatory to application fully.

2.2 Identification and authentication

2.2.1 General information

Service of identification and authentication - one of the most essential services in the systems of information protection, exactly this service stands on the first border of defence.

Authentication of user (subject) is understood as establishing of his authenticity.

Identification of subject (user) is understood as recognition, i.e. determination of user or user process. Authentication is usually produced after identification.

Authorizing (sanctioning) is granting permission of access to the resource of the system [1 - 5].

At included in the system an user must produce information, qualificatory legality of entrance and right on access, i.e. to name itself (procedure of identification). Verification of accordance of the produced identification information to the user (i.e. whether there is an user those, by whom named itself is procedure of authentication) is further produced, plenary powers of user are determined and access is settled an user to the certain resources (to the objects) of the system (authorizing).

Authentication requires, that an user was in any case known to the system. She is usually based on setting to the user of user id (so-called login). However the system can trust the declared identifier without confirmation of his authenticity. Establishment of authenticity is possible at presence of for the user of unique features and what them anymore, less than risk of substitution of legal user. Requirement, qualificatory the necessity of authentication, the politician of informative safety is taken (obviously or unobviously) into account in majority.

2.2.2 Factors of authentication

Yet to appearance of computers for authentication different distinctive descriptions were used. All well-known methods of authentication adapt oneself for the use in the modern computer systems as far as being of corresponding effective on a cost decisions for their application.

The methods of authentication are usually classified in accordance with the distinctive descriptions used by them, and we will classify descriptions in terms of three factors. The type of the distinctive description used for authentication of users (see a table. 2.1) has every factor in basis [2, 5].

Table 2.1. Factors of authentication

Factor	Class of the	Examples
--------	--------------	----------

	authentication system	
Something, well-known to the user: password	System of password authentication	Password, personal identifying code (PIN-code), combination of lock of safe
Something, having for an user: token	System of property (hardware) authentication	Token, smart card, confidential data, built-in in a device, key of mechanical lock
Something, inherent to the user: biometrics	System of biometric authentication	Finger-prints, picture of retina of eye, documentary photograph

Something, well-known to the user: password. Distinctive description is secret information that is unknown to the uninitiated people. To the computers this could be a pronouncing voice password or memorized combination for a lock. In the computer systems it can be password, passphrase or personal identification number (PIN).

Basic dignities of password authentication :

- 1) The cheapest and easy for realization from the point of view of developer mechanism of authentication.
- 2) Bearableness of distinctive description. The memorized secret word is ideal means for authentication of moving users, i.e., for people that is connected to the system from unforeseeable remote places.

In most informative systems the mechanism of authentication and authentication is used on the basis of chart user id (login) / password. Authentication that depends exceptionally upon passwords can not provide adequate protection often, as passwords have weak points.

Basic lacks of password authentication:

- 1) Efficiency of passwords depends on secrecy, and them it is heavy to save in a secret. There is plenty of methods to find out or intercept a password, and usually there is not a method to find out active secret service to causing of damage.
- 2) Development of methods of attacks was done for housebreakers by determination of passwords, usually selectable people, relatively simple business. Even if the difficult guessed passwords get out, they are written down somewhere, if necessary to have at a hand. But, certainly, a written password is more vulnerable in the plan of possible theft, what memorized.
- 3) If users force to use the passwords, generated from casual symbols that it is difficult to guess, then to the users difficult to memorize them.

At the choice of passwords it is possible to use the special programs verifications of passwords, allowing to define whether new passwords are easy for guessing and impermissible.

Mechanisms for the use only of passwords, especially those that pass a password in an open kind (in an uncoded form) vulnerable at a supervision and intercept. It can become a serious problem, if the informative system has out-of-control connections with external networks.

Something, having for users: device of authentication. Distinctive description is possessing the authorized people by some certain object. Before appearance of computers this could be printing with the personal signature or key from a lock. In the computer systems it can be no more than file of data, containing distinctive description. Often description is built in a device, for example in a card with a magnetic stripe, smart card or in the calculator of password. Similar things are named the devices of authentication. Description can be even built-in in the large enough piece of equipment and to appear not very bearable.

Basic dignities of property authentication:

1) Authentication on the basis of devices of authentication more difficult than all to go round, because an unique physical object that a man must possess is used, to enter the system.

2) Unlike passwords, a proprietor always can at once say, if a device was stolen, and he is difficult to partake him with somebody yet and simultaneously to have the opportunity of included in the system.

Basic weak points of property authentication:

1) Higher cost of realization.

2) Risk of loss of device of authentication.

3) Risk of refuse of apparatus.

4) A problem can be also and bearableness.

Something, inherent to the users: biometrics. Distinctive description is some physical feature unique for a person. Before appearance of computers this could be the personal signature, portrait, finger-print or writing description of original appearance of man. In the computer system distinctive description of physical person is measured and compared to the before obtained data taken off from the personality set for certain. In well known methodologies for authentication voice of man, finger-prints, writing signature, form of hand or feature of eyes, is used. Similar things are named a biometrics.

Basic dignities of biometric authentication:

1) Biometric authentication usually is the easiest approach for those people that must pass authentication. In most cases the well projected

biometric system simply takes statements per man and correctly executes authentication.

2) Distinctive description is portable, because it is a part of body of proprietor.

Basic lacks of biometric authentication:

1) As a rule, as compared to other systems, an equipment is expensive in acquisition, setting and exploitation.

2) At the controlled from distance use biometric testimonies are subject to the risk of intercept: a kidnapper can reproduce the record of testimonies, to mask itself under a proprietor or use them, to hunt down that.

3) If biometric indexes get in bad hands, then their proprietor does not have a method of filling in of damage, because biometric features it is impossible to change.

4) In addition, from the point of view of the system, the process of authentication builds. Difficult also to do the system sensible enough, that she rejected extraneous users and here from time to time did not reject it.

5) Biometric indexes also can be confessed by worthless because of physiological changes and bodily harms.

However, in spite of defects, a biometrics remains promising methodology.

In other words, success of passing of authentication always depends on anything lost, damaged or forgotten. In actual fact there is not one best method of authentication. A choice depends on certain risks, with that it is necessary to clash to the computer system and expenses (as an equipment and administration) that a proprietor is ready to bear. Computer centers often depend upon passwords because of low cost: realization does not require the purchase of the special apparatus and implementation of works on setting and service. Organizations use other methods, only in those cases, when potential losses from errors in treatment of passwords exceed the cost of introduction of these methods.

All factors of authentication have the defects, and the separately taken factors can provide the required level of defence not always. In such cases it is possible to use mechanisms authentications that enter two or three factors. The similar systems often name the systems with strong authentication, as advantages of one factor can block defects other. Cards for ATMS that always require the memorized personal identification number are a well-known example of two-factor authentication.

General for all three factors is that distinctive description contacts unambiguous character with a person. However, the simple mechanisms of authentication are often based on that distinctive description sticks to in a secret.

2.2.3 Password authentication. Attacks to the systems of password authentication. Estimation of firmness of passwords

Main dignities of the systems of password authentication are their simplicity and usualness for users, and also cheapness of realization for developers. The systems of password authentication a long ago occupied the niche in most informative systems. It is now accepted to distinguish two basic types of such systems: systems of authentication after non-permanent passwords and system of authentication after frequent passwords. Most distribution was got exactly by frequent passwords. At the correct use they can provide acceptable strength security.

The key element of the systems of password authentication (by the factor of authentication) is something, well-known to the legal user, i.e. password. Thus, before the developers of the systems a dilemma appears: to allow to the users independently to elect passwords, or envisage possibility of their generation in the system. In first case, usually, users choose passwords that is easily memorized, and consequently and easily guessed by malefactors (proper names, to give and others like that). In second case, passwords are proof to guessing, but to memorize to their users difficult. Such passwords users often write (in a notebook, from the reverse of carpet of mouse, keyboard of и etc.) down somewhere, that increases possibility of their opening (declassifying), thus there is a problem of storage of such passwords.

It is necessary to mark that now exists plenty enough of various requirements both to the semantic passwords and to those passwords, that is generated by the special, as a rule programmatic, by facilities. In addition, all rules of choice of passwords improve constantly. The rules and requirements most generalized and popular now in relation to the choice of passwords of users are recommendations for password protection of IBM [6, 7] and advices and recommendations in relation to the passwords of Microsoft [8, 9].

IBM governed in relation to creation of passwords envisage the following:

- A password must contain not less than six symbols.
- A password must contain not less than two alphabetical symbols and not less than single digital or special character. For example, for English it exists 72 possible symbols are 52 letters of the Roman alphabet (including upper and lower cases), 10 numbers and 10 special characters ('!', '@', '#', '\$', '%', '^', '&', '*', '(', ')').
- The password of user must not coincide with his login or regenerate login (written upside-down or cyclic moved).

- A new password must differ from an old man at least three symbols.
- It is forbidden to use symbols or letters that in succession, words of language and others like that
- For the best protection it is recommended to define for passwords the maximal term of action (for example, 8 weeks), after that a password must be changed.

From position of Microsoft, a reliable password answers such requirements:

- Consists at least of seven symbols.
- Does not contain the user (login), actual name or name of enterprise name.
- Does not contain a momentous word.
- Considerably differs from passwords that was used before.
- Contains symbols that behave to each of four groups :
 - Uppercases ('A', 'D', 'C',..)
 - Lowercases ('a', 'b', 'c',..)
 - Numbers ('0' - '9')
 - Symbols, that the keyboards indicated on the keys, but is not alphanumerical ('', '~', '!', '@', '#', '\$', '%', '^', '&', '*', '(', ')', '_', '+', '-', '=', '|', '}', '|', '_, ':', '"', ';', '"', '<', '>', '?', ',', '.', '/').

By an example reliable to the password *there* is 'J*of p2leO4>F'.

A password can answer practically all terms reliable to the password, but here to remain unreliable. For example, password '*of Hello2U!*' unreliable enough, although answers almost all requirements. Password '*of H!elZl2o'*' is reliable, as a semantic word in him is broken to pieces by symbols, numbers and other letters.

What touches passwords that is generated, then usually specialists erect requirements to them to that a password showed a soba the casual or practically casual (pseudorandom) sequence of symbols (or bits). I.e. quality of such to the password depends on that, as far as the quality is used generator of casual sequences. One of the most widespread methods of verification of sequences of symbols (numbers) there is the use of the special statistical tests on a chance, for example tests of NIST [10]. These tests determine the measure of chance of binary sequences that is descendant the random generators of numbers. Tests are based on different statistical properties inherent only to the casual sequences.

On frequent passwords it is possible the basic types of attacks to consider the attacks of guessing (method of attempts and errors or method of «brute force») and attacks a dictionary (if a password is semantic).

In case of attacks of guessing a malefactor sorts out the possible variants of password consistently, while will not get access to the system [2]. Clear

that implementation of such attacks becomes complicated, if to extend the range of possible values of base secret information (i.e. to increase the amount of possible passwords). Accordingly, firmness of secret information can be estimated by the count of general amount of attempts (so-called suppositions) necessary for implementation of attack after the method of attempts and errors. The amount of such attempts represents a chance or entropy of certain base secret information. As really the strong mechanism of password authentication needs plenty enough of suppositions for implementation of attack of guessing after the method of attempts and errors, then such amount for reduction of numbers above that operations are conducted usually give as bit space.

Bit space of number is an amount of binary bits, that is needed for presentation of this number [2]. Bit space represents the volume of memory, that is needed for storage of number.

At comparison of efficiency of different methodologies of authentication usually execute the estimation of their firmness exactly to the attacks of guessing. It is thus possible to use to one of two compatible indexes of complication of attack of guessing - by middle space of attack or mean time of guessing.

Average space of attack is named the bit space, that answers the amount of attempts, that a malefactor must execute [2]. Every element (unit) that is taken into account in middle space of attack shows a soba one calculable operation of c by eventual time of implementation (for example, one hashing of the password, implementation of one attempt of entrance to the system of n etc.).

If all possible values of base secret are equiprobable, then at implementation of attack of guessing on the average it is necessary to check the half of such values. Thus, average space of attack must represent the necessity of search for the half of possible values of base secret.

Average space of attack (in bats) is determined after a formula [2]:

$$V_{av} = \log_2 \frac{S}{2}, \quad (2.1)$$

where S is an amount of combinations of base secret (for the systems of password authentication is an amount of possible passwords).

For the estimation of factor of time the concept of rate R (attempts for a second), with that separate supposition can be executed, is used. Thus, average time of attack of guessing can be defined after a formula [2]:

$$T_{av} = \frac{2^{V_{av}}}{R}. \quad (2.2)$$

It is necessary to notice that more universal means of comparison of firmness of the different systems of authentication is average space of attack, as it mean time depends on a rate, that, in turn, substantially depends on the fast-acting of the computer system of malefactor.

Essence of dictionary attacks consists in that a malefactor owns some dictionaries and consistently tries to give words from these dictionaries as a password. Such dictionaries contain well-known (already broken) passwords usually, the names are own and nouns of certain language. For the increase of efficiency of attack a malefactor can also « transpose » words from dictionaries, i.e. to change the register of separate or all symbols of word, to change the lay-out of keyboard, conduct a transliteration, change letters alike numbers or symbols, to add numbers at the beginning or at the end of word.

2.2.4 Biometric authentication

2.2.4.1 Application of biometry

Because a biometrics has the unique personal descriptions in basis, then she can be used in three different, but constrained applications [2, 5].

Authentication.

Decided task: is it possible to confirm that the user name behaves to the that man that presented him?

Biometric indexes in this case act part base secret or means of verification, and they must close coincide with a record in a base for a corresponding user.

Identification.

Decided task: having a standard of biometric indexes, is it possible to bind them to the unique man or, at the worst, with less of persons?

To such application the classic case of the use of finger-prints behaves law enforcement authorities. Like application requires the presence of vast base of these standards of biometric indexes, that, probably, contains the sought after standard. The most effective systems of authentication are those, that have most bases of these standards of biometric indexes. For example, the Federal bureau of investigations of the USA (FBI) systematic collects finger-prints from the beginning of 20th century, and this collection is the central element of automated system of authentication on finger-prints, AFIS (Automated Fingerprint Identification System).

Determination of unicity.

This variant of the applied task of authentication purchased practical sense with appearance of cheap computer methods of realization of biometric methods.

Decided task: whether it is possible, having a standard of biometric indexes, to define whether there is their proprietor in a database.

For example, such technology is used by the USA financed by a government by organizations on delivery of manuals for verification that, whether a declarant did not register oneself on drawing allowance several times.

Distinction between the task of authentication and determination of unicity insignificantly, but it is important. Both applications use identical methodology, but serve to the different aims. Appendixes, decision the task of authentication, plenitudes try to obtain: an ideal database contains data on living everybody or at least on each of certain group. The appendixes related to the decision of task of unicity touch people entering into relationships with organization or enterprise, and use a biometrics, to guarantee that every physical person is registered not more than once. The systems of authentication aspiring to plenitude, the systems of determination of unicity have usually absent that, is peculiar to. Important technical distinction between these applications consists of their scale: combination of biometric elements considerably becomes complicated with the height of their amount. The systems of authentication can raise to the arbitrarily largenesses.

The use of similar applications put the question of confidentiality of information about the personal life. Because the biometric measuring is unique for separately taken everybody, watching of movement of people becomes possible by watching of their biometric indexes. For example, each, who visited the place of crime some time, can be invited on an interrogation only on that simple reason, that his finger-print or other biometric index looks like that was discovered in flagrant delict. If engaging in electronic commerce companies plugged collection of biometric indexes in the procedure used by them, then would could it was to watch after a man, conducting a search on the records of the on-line transactions done with the use of his biometric character. Disturbs some, that the finger-prints collected by biometric applications can get in an identification database similar to the system AFIS FBI [2].

In unstanding time not many companies engaging in advancement at the market of the biometric systems are concentrated on the decision of problem of encroaching upon private life. Actually, a public concern compels many producers of the biometric systems this problem to emphasize safety and providing of confidentiality of the personal information in their systems, that those were confessed by consumers. But it satisfies critics not always. There always is a risk of «functional creep», when biometric information with development of the system can appear

used in completely another quality. It is although possible to deprive the system of certain functions (for example, viewing of finger-prints from the place of crime), there are no guarantees, that such function will not appear later. As soon as organization becomes the possessor of biometric indexes, there are no methods to prevent their use in other aims. The systems «know» how to combine biometric indexes with present for them standards, but they do not «understand», one or another indexes were for what presented them.

2.2.4.2 Biometric methodologies

Practically all equipment created for measuring of unique descriptions of man can be used for biometric authentication. The cost of such equipment can arrive at ten and hundreds of thousands of dollars. However, in a number of cases, biometric authentication can be organized on the basis of the use of equipment, entering in the complement of the ordinary personal computer, for example, of keyboard or microphone. Now distinguish about one and a half ten of different types of biometric methodologies that can be used for authentication [2]. All these methodologies it is accepted to divide into two kinds: the methodologies, related to measuring of distinctive physical parameters of man (static descriptions) and the methodologies related to measuring of behavioral descriptions (dynamic descriptions).

Static methodologies envisage measuring of physical descriptions of human body, that in an ideal must be unique for all humanity or, at least, for greater part of people. On measureable description must not render influence normal vital functions of man. Also description must not substantially change during long time.

Dynamic methodologies are based on determination of behavioral type of man unique biometric indexes correspond that. The characteristic feature of such methodologies is that authentication can be each time produced on the basis of different data. For example, the system of authentication can offer to the user to say a certain phrase.

Basic biometric methodologies are presented on a fig. 2.2.

Biometric methodologies

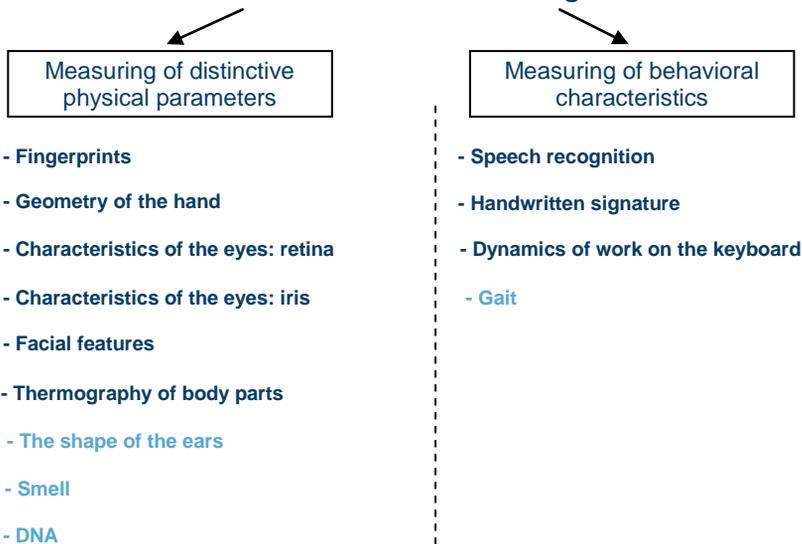


Fig. 2.2. Biometric methodologies

2.2.4.3 Elements of the biometric system

Without depending on the used biometric methodology, all biometric systems of authentication have a general structure plugging in itself general elements (see a fig. 2.3) [2].

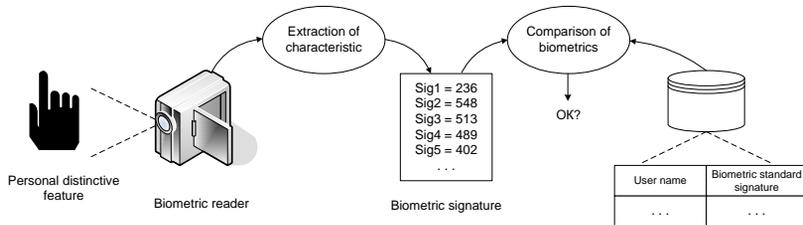


Fig. 2.3. Elements of the biometric system

Distinctive description of man is read and digitised by means of the special reading device. Principles of organization and construction of reading devices (as well as all other elements of the system) differentiate for

the different systems of biometric authentication and depend on the type of the used methodology. From the data obtained from a reading device the system distinguishes data describing the required description, i.e. produces extraction of description. As a result of procedure of extraction of description the so-called biometric signature is formed. A biometric signature is a set of values of the distinguished parameters of description. For authentication of user the system produces the selection of standard biometric signature from a record corresponding to this user in the base of users and compares her to the got biometric signature. Thus authentication will be successful at the partial, but near enough matching. The standard of biometric signature is added to the base during registration of user. Usually for his forming produce a few read-outs of description and obtained data are averaged.

2.2.4.4 Exactness of biometry

One of important questions, that decides the developers of the biometric systems there is a question of being of balance between probability of erroneous refuse to the legal user and by probability of erroneous confirmation of illegal user. The increase of one of these indexes brings other over to reduction. Those what harder requirements on blocking of «stranger», the more possibility of blocking «ours». Thus, it is necessary to find a compromise between the comfort of users of the system and strength (see a fig. 2.4) security [2].

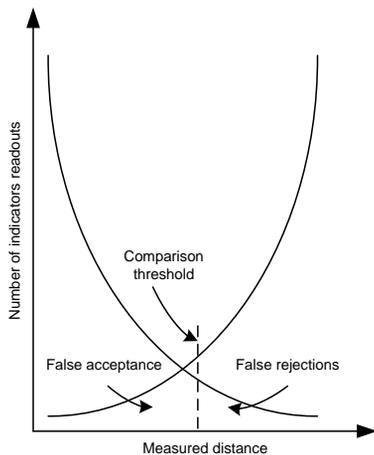


Fig. 2.4. Threshold of combination of erroneous refuses and confirmations

The basic indexes of exactness of the biometric systems are the false acceptance rate (rate of I kind errors, FAR) and false rejection rate (rate of II kind errors, FRR). The values of coefficients of FAR and FRR correspond to certain position of threshold of combination. These coefficients are determined, accordingly, as a stake of erroneous confirmations and abandonments from the general amount of attempts. Connection between the coefficients of errors it is accepted to show by means of receiver operating characteristics curve – ROC curve (see a fig. 2.5).

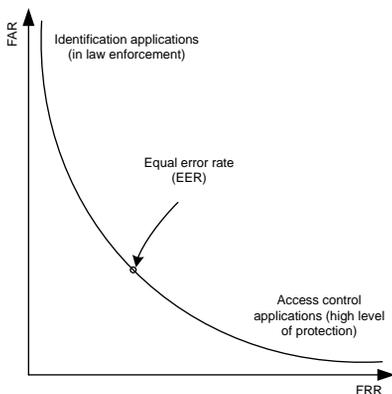


Fig. 2.5. Receiver operating characteristics curve

Every point of curve corresponds to position of threshold of combination and certain values of coefficients of errors. Point in that the values of coefficients of FAR and FRR are equal named the point of equal error rate – EER. The subzero location of EER point testifies to good balance between erroneous refuses and confirmations.

As be obvious from a fig. 2.5, the appendixes related to judicial authentication require the subzero value of coefficient of erroneous refuses and can assume the high value of coefficient of erroneous confirmations. In the appendixes related to the management by access, the subzero value of coefficient of erroneous confirmations is required, let even by the cost of numerous erroneous refuses [2].

For rough comparison of complication of attacks on passwords with attacks on biometry it is also possible to take advantage of the middle space of attack, described higher. At the calculation of middle space for the biometric systems the value of coefficient of erroneous confirmations is

used:

$$V_{av} = \log_2 \frac{2}{k_{FAR}}, \quad (2.3)$$

where k_{FAR} is a value of coefficient of FAR.

2.2.5 Property authentication

2.2.5.1 General information about property authentication

The main feature of property authentication is that an user must have some device of authentication. From the point of view of informative safety authentication with the use of devices has such basic features [2, 5]:

- an user must physically possess the device of authentication for the receipt of access to the system;
- to make the doublet of good device of authentication heavily or beside the purpose;
- possibility of loss of device of authentication an user, that results in impossibility of receipt of access to the system;
- possibility of theft of device of authentication for an user.

Main reason of popularity of authentication with the use of devices is absence of the difficulties related to memorizing. The device of authentication can contain considerably more difficult base secret, than man is able to memorize. At most, that required from an user is this memorizing of PIN - code of access to the device.

Authentication devices divide into active and passive (see a fig. 2.6).

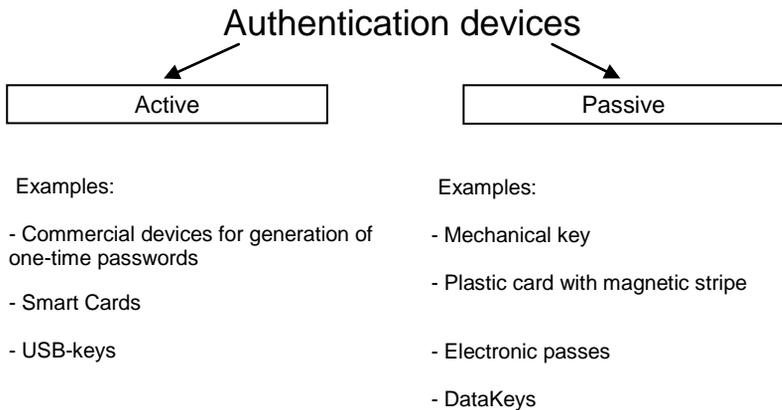


Fig. 2.6. Classification of authentication devices

The device of authentication in any case carries in itself some base secret. For making of doublet of device a malefactor must have a copy of base secret.

Passive devices of authentication are some device of storage of base secret. At passing of authentication a device gives this secret [11, 12].

A general lack of passive devices is possibility easily to copy them. Such defect decides the use of the second factor of authentication, usually PIN - code.

Active devices can in different circumstances generate a different weekend data. Such devices do not need to pass the base secret for authentication of proprietor. These data are used for some operation (for example, generations of non-permanent password). An encryption is often used for this purpose.

Distinguish two types of active devices of authentication. The first type of devices is hardwarily realized devices that envisage the generation of different set of reports at every session of authentication. Thus, there is not sense to reproduce the previous set of reports a malefactor. Reports usually take from the keyboard of computer. The example of such devices are the calculators of non-permanent passwords, made as trinkets.

The devices of the second type are connected directly to the computer and have difficult protocols of authentication. The example of such devices are smart cards or USB- keys.

2.2.6 One-time passwords

2.2.6.1 The calculation of one-time passwords with the use of counters and clocks

Principle of work of the systems of authentication on one-time passwords (OTP) consists in that a password operates only in one current session of authentication. An attempt to give this password next time will result in a failure. For the calculation of series of such passwords on the side of user the special devices are often used. Certainly, the calculation of one-time passwords can be programmatic.

Systems of authentication on the basis of one-time passwords it is necessary to consider separately, as, from one side, authentication takes place with the use of passwords, but, with other, devices are used.

One of two strategies of generation of one-time passwords is usually used [2]:

- with the use of counters,
- with the use of clocks.

Counter device of authentication combine a base secret with the indications of the synchronized counter (see a fig. 2.7).

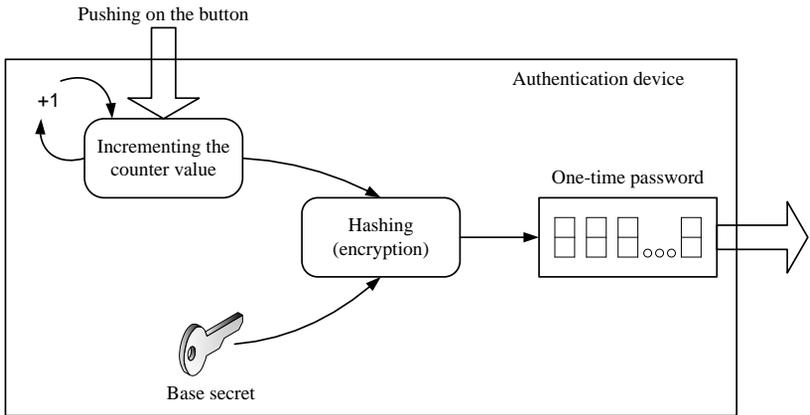


Fig. 2.7. Calculation of OTP on the counter device of authentication

The calculation of password on a device takes place as follows. An user pushes button on a device. A device is increased the state of internal counter and produces his hashing (or encryption) with the use of unique value of base secret. A base secret can be used as a key of encryption or can be combined with the indications of counter at hashing. After it a device formats the result and outputs a new one-time password on a display. An user enters the got password by means of keyboard of computer. On a server the base of registration records of users is conducted with the actual states of counters and base secrets. On the basis of these data a server generates the password and checks his accordance to the password given by an user.

In such systems, if an user by chance pushed button on a device, desync of counters of device and server is possible. A problem can decide a few ways. For example, user can be asked for two serial passwords or along with a password a device can give the state of a few least bits of counter. On the basis of such data a server can define whether there were the given passwords really generated by the device of this user and further to correct the storable in his registration record state of counter.

The clock devices of authentication combine a base secret with the indications of the synchronized clock / timer (see a fig. 2.8).

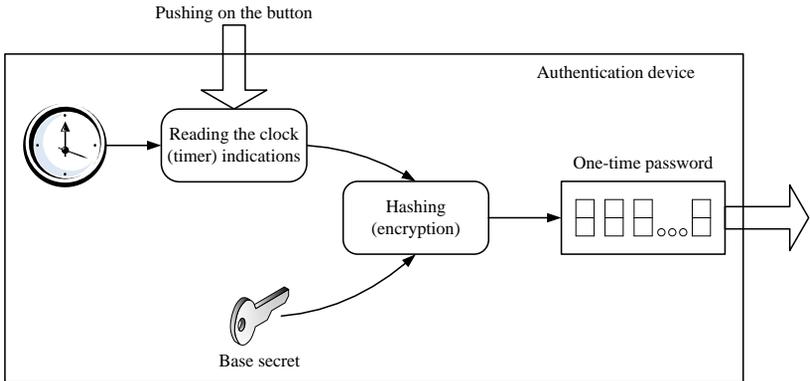


Fig. 2.8. Calculation of OTP on the clock device of authentication

The calculation of password on a device takes place as follows. An user pushes button on a device. A device reads the indications of internal clock / timer, combines them with the unique value of base secret and produces their hashing (encrypting). After it a device formats a result and outputs the value of new one-time password on a display. On a server a base secrets and current statuses of clock are registered. A server generates a next password and compares it to the entered by user. Updating of the state of clock / timer is ordinary with a period of 1 second. Those after a 1 second a device can generate a new password already.

At the use of such chart of authentication the situations of desync of indications of clock are possible on a device and on a server (a clock can fall behind or hurry). It is also necessary to take into account that on an input and transmission of password on a server substantial time is required. These questions decide with the help of so-called time window. His size can vary from a few tens of seconds to a few minutes. A server checks up accordance of the given password to one from time window.

2.2.6.2 One-time passwords on principle a «query – response»

It is considered that first principle «query – response» was applied by Bob Bosen in the computer game «80 space raiders». At a start a game gave out the random number of query on a screen. An user needed to find an answer for this number in the table fastened on a box from a game [2].

Principle of authentication with the use of mechanism «query – response» is shown on a fig. 2.9.

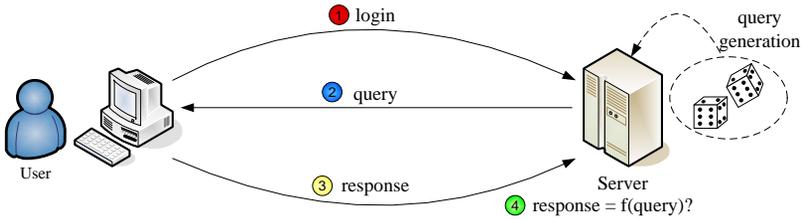


Fig. 2.9. Chart of generation of password on a mechanism a «query - response»

Stages of authentication:

- 1) Input (transmission on a server) of the user name;
- 2) Generation of random number of query and transmission to his user;
- 3) Calculation of one-time password on the basis of number of query and transmission it to server;
- 4) Verification of accordance of one-time password to the query and authorizing.

An answer in such systems is also one-time password. For the calculation of answer and his verification a function with a secret parameter is used.

Advantages of one-time passwords a «query - response» (as compared to other systems of generation of one-time passwords):

- Absence of necessity of synchronization of client and server;
- Open standards of authentication devices.

Mechanisms a «query is an answer» are widely used for authentication of financial transactions on banking. ISO 16609 standard [13] (and his predecessors ISO 8730 and ISO 8731) envisages the use of symmetric cryptoalgorithms for forming of MAC – codes (message authentication code). The chart of authentication device, supporting the indicated standards, is presented on a fig. 2.10.

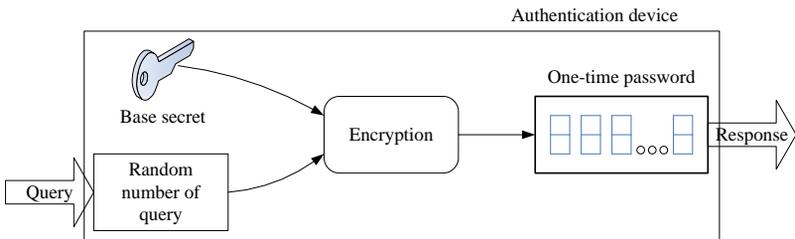


Fig. 2.10. Chart of authentication device on principle a «query - response»

By the basic type of attacks on authentication on principle a «query - response» with the use of MAC - codes there are an intercept of queries and one-time passwords and surplus on the method of tests and errors of possible values of base secret. For the best protecting from attacks it is recommended to use more proof and modern cryptoalgorithms.

2.3 Typical models of authentication

At planning of architecture of the system of authentication it is expedient to use the prepared templates or typical models. There are four basic typical models of authentication [2, 5]:

- Local,
- Direct,
- Indirect,
- Autonomous.

2.3.1 Local authentication

Such template is used in the desktop systems (see a fig. 2.11). User works with the system straight. All system (including the mechanisms of authentication and access control) takes place into one physical perimeter of safety. The proprietor of the system (or the person authorized by him) conducts and renews the database of authentication into this perimeter.

The lacks of such model are complication of administration, simplicity of breaking of perimeter and complication of organization of the multi-user mode.

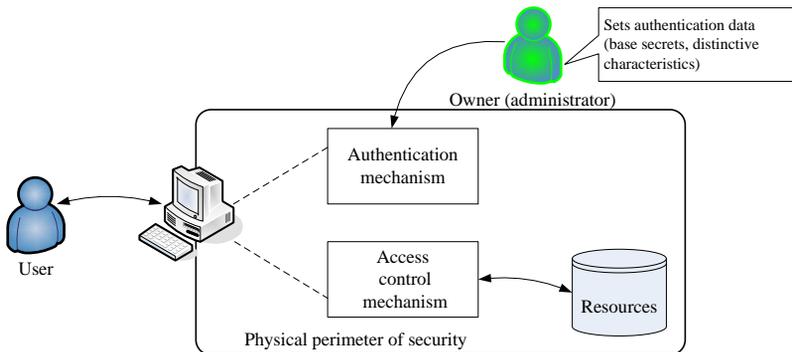


Fig. 2.11. Template of local authentication

2.3.2 Direct authentication

This model is used in the relatively old server systems of local networks and time-sharing systems (see a fig. 2.12). The system can remotely collectively use great number of different users. Mechanisms of authentication and access control of the system is also been situated into one physical perimeter of safety. A proprietor conducts and supports the database of authentication into every system. Users are out of physical perimeter and not protected. Their co-operating with a server is produced straight. The point of service executes authentication and passes confirmative information in a server.

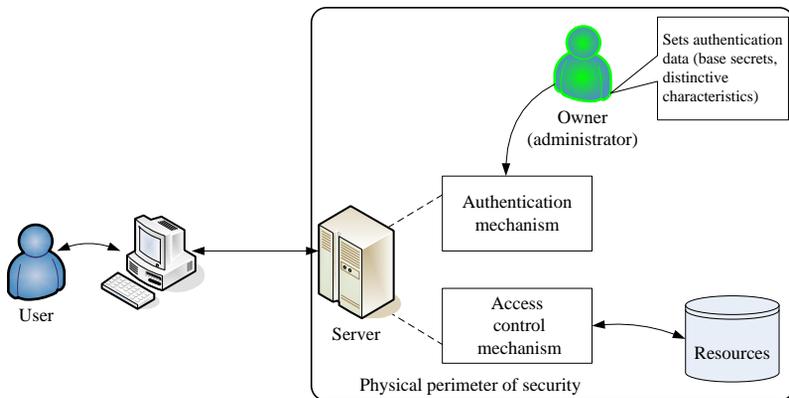


Fig. 2.12. Template of direct authentication

The lack of this model is possibility of intercept of traffic between an user and server. It lays on limits on the use of biometrics and multiple passwords.

2.3.3 Indirect authentication

A model is used in modern network servers with protocols of RADIUS, Kerberos etc.(see a fig. 2.13). The system contains a few points services that require a access control and can take place in different places. Users call to services of the system remotely as necessary. A proprietor conducts and supports one database of authentication for all system. There are a few points of service. A mechanism of authentication is not in the point of service, but in the special server of authentication. All other services (servers) do not make decision on authentication straight, and use the server of authentication.

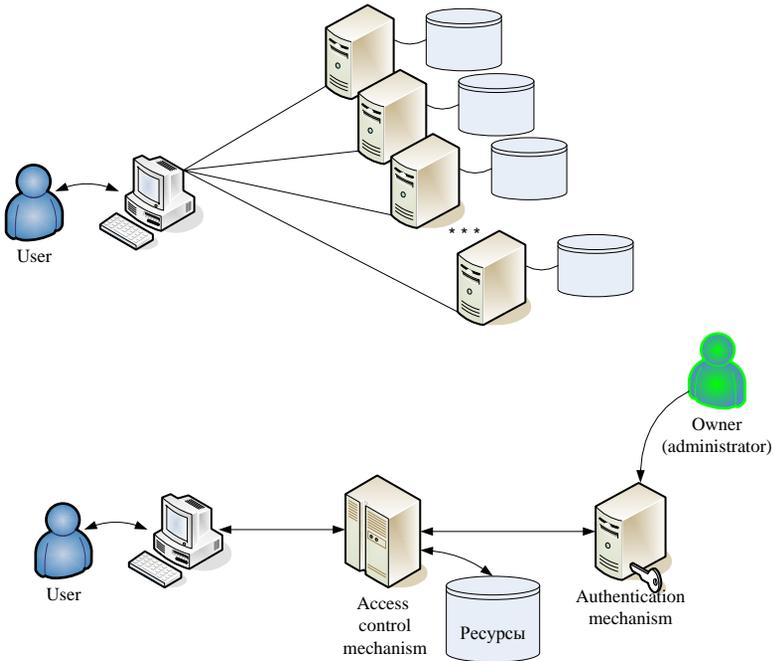


Fig. 2.13. Template of indirect authentication

2.3.4 Autonomous authentication

Such model is used in the systems with the infrastructure of the public key, containing numerous autonomous components (see a fig. 2.14). These components are able to accept exact decisions on a management by access even in default of connection with other systems for the receipt of data on authentication. A proprietor accedes to the risk, that decision can be made with the use of out-of-date data on a access control and authentications and can give a wrong result. The system is up-diffused, it is therefore impossible to depend upon the centralized server of authentication in real time. Management by authentication here centralized. A proprietor is independent organization (the organ of certification). Autonomous authentication combines the features of the first three models.

The lack of model is a difficult process of depriving users of authority.

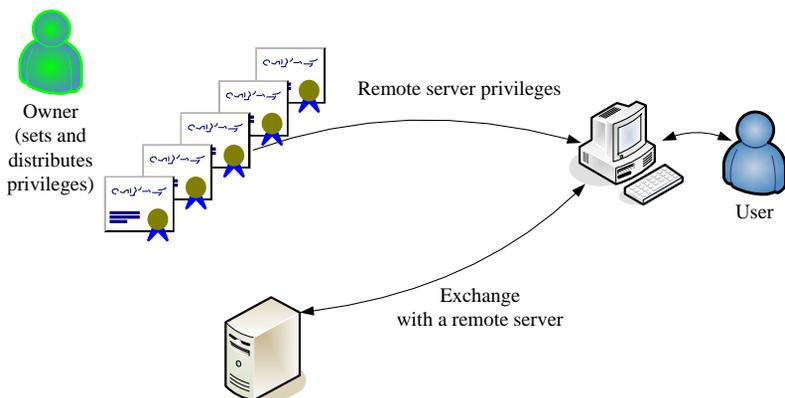


Fig. 2.14. Template of autonomous authentication

2.4 Access control mechanisms

2.4.1 Access control mechanisms and politics of safety

The questions of providing of informative safety are regulated by politics of safety (PS). Politics sets the «rules of life of the system».

Typical questions reflected in politics of safety [14]:

- order of including and exception of users in / from the system;
- order of grant of access to the resources;
- rules of access control to the resources;
- order of storage and distribution of information and other

The most general technical requirements are determined by normative documents and plug in itself:

- Functional requirements - describe management rules access of subjects (users, programs) to the objects (to the resources) of the system, authentication and verification of authenticity of subjects of access, registration of critical events and control of integrity of resources of the system;

- Requirements on the guarantees of architecture (requirements on the special reliability) - describe the order of design, development, testing and documenting of the system, allowing to attain the set level of reliability of nocensors of the system, exception of errors and undeclared possibilities in the programs and equipment.

Access control is the most essential part of functional requirements and serves as basis for classification of the systems on the level of security of information. Requirements on the guarantees of architecture describe the technological aspects of development of the systems.

2.4.2 Access dispatcher

Access control of subjects to objects is performed by the access dispatcher (see a fig. 2.15) [14].

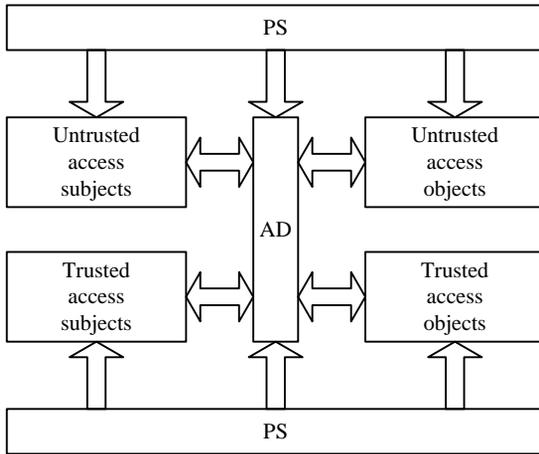


Fig. 2.15. Model of access dispatcher

A access dispatcher (AD) is a subsystem that identifies objects and subjects and makes decision about the grant of access on the basis of comparison of service attributes (SA). SA is some accounting information: the names, passwords of users, address of computers, mark of safety of resources and users.

Accounting information, the configuration parameters of the information security system and codes of the programs behave to the digit of the trusted objects. Their reading and change is produced by the trusted persons on the trusted interface.

Requirements to AD:

- intercept of all appeals from subjects to the objects of access;
- exception of alternative access to the protected objects;
- simplicity of engineering analysis of correctness of realization of functions.

Practice shows that for most systems of politician of safety can be realized by two case frames by access - discretionary and capability-based.

2.4.3 Discretionary access control

A discretionary (arbitrary, electoral, confidence) access control is base on grant of access to the objects the proprietors of these objects [1, 14].

For every pair there is a subject - an object is set list of rights for a subject in relation to an object. A decision on a grant or refuse in access depends on the type of the object inquired by a subject.

Totality of rules setting the types of access for objects is usually recorded in the type of so-called access control table (matrices of access). For every user object a proprietor is set, possessing an unlimited right for the change of attributes of access on the created (and belonging to him) objects, i.e. by a right for the grant of the objects to other subjects.

For description of matrix of access individual or group permissions heritable from higher objects or individually appointed by a proprietor (see a fig. 2.16) are used.

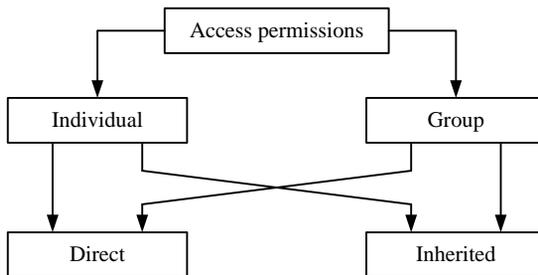


Fig. 2.16. Rights for access are in discretionary models

Group permissions are determined between functional groups (in a number of the subjects grouped on a certain sign) and aggregated objects (for example, by the great number of hierarchically well-organized resources of the file system).

Direct permissions suppose that an appropriation to the object of attributes of differentiation of access is produced every time at his creation by a subject. Heritable permissions suppose defence of the created objects by default.

For formal description of rules of differentiation of access in discretionary models management lists are usually used by access (ACL - access control list). Every element of ACL of object contains the rules of access to him.

A discretionary access case frame is used in most DB and industrial OS (including families of MS Windows and Unix).

Lacks of discretionary case frame by access:

- 1) Service attributes (ACL) on the basis of that made decision after the giving of access are related to the objects of access only locally. For example, at moving of file through a network from one computer to other it

will be necessary again to form ACL.

2) The matrix of access is determined for objects that is «visible» to the user (for example, files, devices etc.) and does not take into account cooperation between processes. In the multi-user systems cooperation between processes can result in the unnoticeable for an user loss of information.

3) In a discretionary model it is difficult to organize work with information of different level of secrecy in the mode, exclusive «mixing» of information of different vultures and categories. For example, such system is vulnerable to the attacks the trojan programs: a proprietor starts the trojan program on its own behalf, the program reinstalls rights for access on interesting her resources and malefactor able to get access to these resources.

2.4.4 Mandatory access control

The mandatory-based (force, administrative, plenipotentiary) methods of access control are based on an appropriation to the objects and subjects of marks of safety (MS) [1, 14].

The mark of safety of subject describes his reliability, mark of safety of object, is a degree of closed of the information contained in him.

The marks of safety usually get out from the arcwise well-organized great number of $M=\{M_1, M_2, \dots, M_N\}$ and appointed to the objects and subjects of access at their creation in the system.

For comparison of marks of safety of subjects and objects it is necessary to set the rules of comparison of elements of M . In practice the choice of M is produced by one of two methods.

First method: choice of great number of consistently increasing natural numbers of M_H . Comparison of elements of such great numbers is produced naturally and simply. Such great numbers are usually named hierarchical classifications.

Second method: forming of elements of great number as subsets of some arcwise unregulated great number of M_U . For example, as M_U it is possible to choose a great number $\{(m_1), (m_1, m_2), (m_1, m_3), (m_1, m_2, m_3)\}$, thus none of pairs of m_i is comparable in ordinary sense. Then, comparison can be executed on the basis of including of great numbers. Obviously, great numbers containing identical elements equal. Great number, containing (m_1, m_3) a less great number containing (m_1, m_2, m_3) . Such classifications are named unhierarchical.

A capability-based management will realize the inductive model of safety access. A model of safety is inductive, if for the system once set in

the safe state her being is guaranteed in the safe state in future.

Usually the marks of safety consist of two parts - level of secrecy and list of categories. The levels of secrecy form a well-organized great number, for example:

- top-secret;
- secretly;
- confidentially;
- unsecretly.

Categories form an unregulated set and intended for description to the subject domain data behave to that. The mechanism of categories allows to divide information to the section-by-section, that assists the best security. A subject can get access only to the categories. The certain list of categories depends on the subject domain of the system.

One mark of safety prevails above the second, if her level of secrecy not below and all categories of the second mark are included in her list of categories.

A subject can read information from an object, if his mark of safety prevails above the mark of safety of object. A subject can write down information in an object, if the mark of safety of object prevails above to his well-aimed safety. Obviously, that a subject can read and write down, if mark of subject and object equal.

2.4.5 Role based access control

A role access case frame is one of varieties of discretionary model [4, 5, 15].

A role is some set of rights for access that is inherent to the users with certain functional (by a post) duties.

Complication of administration of the system is determined by the number of present in her connections (rights for access) between subjects and objects. The amount of roles far less than, than subjects and their rights. Thus, the use of roles results in simplification of administration. From a fig. 2.17 evidently, that the order of amount of connections between subjects and objects in a discretionary model is equal $n*m$, and in role model is equal $n+m$.

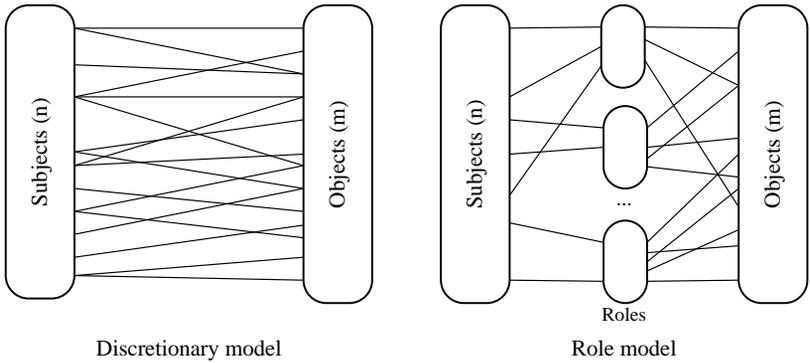


Fig. 2.17. Connections between subjects and objects in discretionary and role based models

By means of role model it is possible to realize principle of division of duties. Between roles the static or dynamic relations of incompatibility can be certain, i.e. to impossibility to one subject in turn or simultaneously to activate both roles.

There is a standard of role models - Role Basis Access Control (RBAC) [15]. In the models of such type the inheritance of roles (see a fig. 2.18) can be used also.

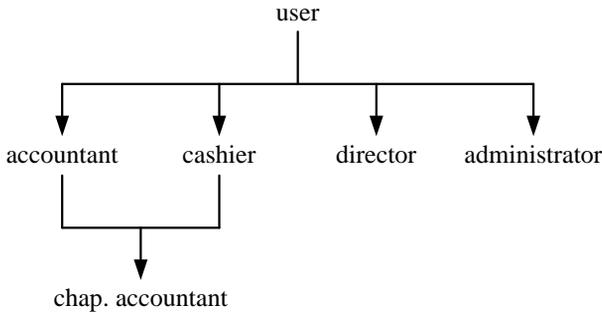


Fig. 2.18. Tree of inheritance of roles

Questions for self-control

1. Name basic services and mechanisms of information protection?
2. What is authentication?
3. Name the basic factors of authentication?
4. What is a device of authentication?

5. Name the possible variants of the use of biometrics?
6. What can be attributed to the elements of the biometric system?
7. What indexes is characterized exactness of biometrics?
8. What is a bit space?
9. What is an average space of attack?
10. Name the basic typical models of authentication?
11. What is a controller of access?
12. Explain the concept of subject in?
13. Explain the concept of object in access control system?
14. Name the basic frames of access control?
15. What is a matrix of access?
16. What is a mark of safety?
17. Explain the concept of role in access control system?

Bibliography

1. Домарев В.В. Безопасность информационных технологий. Методология создания систем защиты – К.: ООО «ТИД «ДС», 2001. – 688С.
2. Smith R. Authentication: From Passwords to Public Keys Addison – Wesley Professional, 2001. – 576P.
3. Todorov D. Mechanics of User Identification and Authentication: Fundamentals of Identity Management – Auerbach Publications Taylor & Francis Group, 2007. – 760P.
4. Ballad B., Ballad T., Banks E. Access Control, Authentication, And Public Key Infrastructure – Jones & Bartlett Learning LLC, 2011. – 391P.
5. Reid P., Smith R. User Authentication Systems and Role-Based Security – Pearson Custom Publishing, 2004. – 265P.
6. Пароли для пользователей https://www.ibm.com/support/knowledgecenter/ru/SS8CCV_7.6.0.3/com.ibm.mbs.doc/user/c_passwords_users.html
7. Как задать требования к паролям https://www.ibm.com/support/knowledgecenter/ru/SSFGJ4_7.6.0/com.ibm.mbs.doc/mbs_common/t_set_password_reqs.html
8. Рекомендации по созданию паролей [https://msdn.microsoft.com/ru-ru/library/cc540536\(v=exchsrvcs.149\).aspx](https://msdn.microsoft.com/ru-ru/library/cc540536(v=exchsrvcs.149).aspx)
9. Пароль должен отвечать требованиям сложности [https://technet.microsoft.com/ru-ru/library/hh994562\(v=ws.11\).aspx](https://technet.microsoft.com/ru-ru/library/hh994562(v=ws.11).aspx)
10. Rukhin A. and others A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications // NIST Special Publication 800-22 Revision 1a. – 2010. – <http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf>.

11. Tammet T., Vain J., Puusepp A., Reilent E., Kuusik A. RFID-based Communications for a Self-Organising Robot Swarm //Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems, 2008. – P.45-54.

12. Tammet T., Vain J., Kuusik A. Using RFID tags for robot swarm cooperation // WSEAS Transactions on Systems 2006/5. – P.1121 – 1128.

13. ISO 16609:2012 Financial services. Requirements for message authentication using symmetric techniques

14. Тарасюк М.В. Защищенные информационные технологии. Проектирование и применение – М.: СОЛОН-Пресс, 2004. – 192С.

15. Sandhu R., Ferraiolo D.F., Kuhn D.R. The NIST Model for Role Based Access Control: Toward a Unified Standard Proceedings // 5th ACM Workshop on Role Based Access Control, July 26-27, 2000, Berlin, P.47 – 63.

Module 1. Information security. Definitions, norms, standards

PART 5. SYSTEMS AND NETWORKS SECURITY AND RESILIENCE

MODULE 1 INFORMATION SECURITY. DEFINITIONS, NORMS, STANDARDS

CONTENTS SECTION

1. Information, its features and form
 - 1.1.1. General concepts of information and its features.
 - 1.1.2. Limited access information and its properties (integrity, confidentiality, availability).
 - 1.1.3. Forms of information and channels of distribution.
- 1.2. Threats to information security
 - 1.2.1. The concept of an Automated System (AS) information processing.
 - 1.2.2. Information Security.
 - 1.2.3. Threats to Information Security and their classification in a number of basic features.
- 1.3. Regulatory and methodological support of information protection in the USA, Canada, the EU and the Russian Federation
 - 1.3.1. Trusted Computer System Evaluation Criteria (Orange book).
 - 1.3.2. Federal Criteria for Information Technology Security (FCITS).
 - 1.3.3. Information Technology Security Evaluation Criteria (ITSEC)
 - 1.3.4. Canadian Security Criteria (STSREC)
 - 1.3.5. State Technical Guidance Documents Russia On information protection.
 - 1.3.6. Common Criteria for Information Technology Security Evaluation (ISO / IEC 15408: 1999).
 - 1.3.7. Short review of the cybersecurity strategies and developments in the cybersecurity policy area in the Eastern Partnership region.
- 1.4. Regulatory and methodological support in the field of technical protection of information in Ukraine
 - 1.4.1. State Service for Special Communication and Information Protection of Ukraine (State Service): history, tasks, rights and obligations.
 - 1.4.2. General provisions for the protection of information in computer systems from unauthorized access (ND TPI 1.1-002-99).
 - 1.4.3. Criteria for evaluating the security of information in computer systems from unauthorized access (ND TPI 2.5-004-99).

Module 1. Information security. Definitions, norms, standards
1.4.4. Classification of automated systems and standard functional profiles manufacturing security information from unauthorized access (ND TPI 2.5-005-99).

1.1 Information, its features and form

1.1.1 General Concepts of Information and its Features

The word "Information" comes from the Latin «Informatio», which means the reduction, clarification, acquaintance. The notion of "Information" is the base of information security and computer science courses, but it is impossible to define it through other, more "simple" concept.

The notion of "Information" is used in various sciences, while in every science the notion of "Information" is associated with a variety of systems concepts.

You can select the following approach to the definition of information:

* **Traditional** (everyday) - used in computer science: information - is information, knowledge, messages about the status of that person perceives from the outside world through the senses (sight, hearing, taste, smell, touch).

* **Probability** - is used in an information theory: information - is information about the objects and phenomena of the surrounding environment, their parameters, properties, and able to reduce the existing degree of uncertainty about them, and incomplete knowledge.

For a man: Information - this knowledge he receives from various sources using the senses.

Knowledge is divided into two groups:

1. Declarative - declaration of the words (statements, reports) begin with the words "I know that ...";

2. Procedure - define actions to achieve any goal, starting with the words "I know how to ..."

Information Classification [1, 2]:

- According to the methods of perception - visual, auditory, tactile, olfactory, gustatory;

- By submission of forms - text, numerical, graphical, musical, combination, etc.

According to public value:

Mass - everyday, socio-political, aesthetic;

Module 1. Information security. Definitions, norms, standards

Special - scientific, technical, administrative, production;

Personal - our knowledge, skills, intuition.

Basic property information:

Objectivity - which is independent of any opinion;

Reliability - reflects the true state of affairs;

Completeness - is sufficient for understanding and decision-making;

Relevance - is important and essential for the present time;

Value (utility, relevance) - provides a solution to this problem, we need to make the right decisions;

Clarity (Clarity) - expressed in a language accessible to the recipient.

Additional features:

1) Attribute properties (attribute - an integral part of something). The most important among them are: - discrete (information consists of separate parts, signs) and continuity (the ability to store information);

2) The dynamic properties associated with changes in data over time:

- Copy - duplication of information;

- Transmission from the source to the consumer;

- Translation from one language to another;

- Transfer to another medium;

- Aging (physical - media, moral - axiological).

3) Practical features - information volume and density.

Information stored, processed and transmitted in the symbol (sign) form. The same information can be presented in different forms: 1) the sign writing, consisting of a variety of signs including emit a character in the form of text, numbers, special. characters; graphics; a table, etc .; 2) a sign or signal; 3) In the oral verbal (conversation).

Submission of information by means of language as sign systems, which are based on a specific alphabet and have the right to perform operations on characters.

Language - a certain sign system of reporting. Exist:

Natural languages - spoken languages in oral and written form. In some cases, spoken language can replace the body language, the language of special characters (eg, road);

Formal languages - languages specific to different areas of human activity, which are characterized by rigidly fixed alphabet, more stringent rules of grammar and syntax. This is the language of music (notes), the language of mathematics (numbers, mathematical signs), number systems, programming languages, etc.

Module 1. Information security. Definitions, norms, standards

At the core of any language is an alphabet - a set of symbols / characters. The total number of characters of the alphabet is called the power of the alphabet.

Media - media or the physical body for the transmission, storage and playback. (This power, light, heat, sound, radio signals, magnetic and optical discs, prints, photographs, etc.).

Information processes - processes related to the receipt, storage, processing and transmission of information (ie, actions to be taken with the information). This is a process during which changes the information content or its form of presentation.

Information resources of the state, society, individual organizations and individuals are determined by value, with appropriate expression and material requiring protection from various influences that could reduce their value.

Rationale for information security. Information resources of the state, society, individual organizations and individuals are determined by value, with appropriate expression and material requiring protection from various influences that could reduce their value.

1.2. Limited Access Information and its Properties (integrity, confidentiality, availability)

Classified information and its properties (integrity, confidentiality, availability) based on state standard - UKRAINE GOST 3396.2-97 (Data protection. Technical Data Protection. Terms and Definitions Protection of information. Technical protection of information. Terms and definitions) [3].

There is fore types of information to be technical protection (Fig.1):

- 1. Information** - Learn about the processes and phenomena (ISO 2226).
- 2. Classified information** - information, the right of access to which is restricted established legal norms and (or) rules.
- 3. Secret information** - Classified information, which contains information constituting state or other secret by law.
- 4. Confidential information** - Limited access information in the possession, use or dispose of some natural or legal persons or the state of access and order which set them.

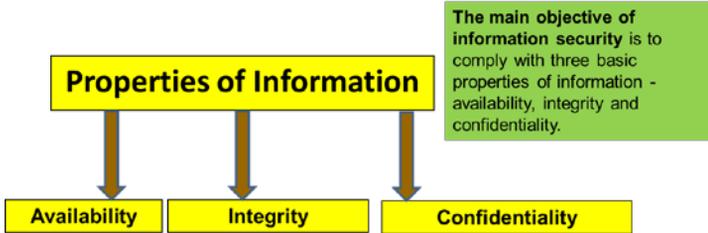


Fig. 1 – Properties of Information

Availability of information - this is its property that allows the entity that has the right to get information in the form required subject, in a place that should be subject, and in the time required for the subject.

Integrity of information - this is her property, which is that the information can not be changed (this includes deleted) unauthorized on the subject (it could be a man, and a computer program, and computer hardware and any other effects such as strong magnetic radiation, fire or flood).

Confidentiality of information - the property whose value is set by the owner of information reflecting the restriction of access to it, according to the existing legislation.

Additionally distinguish ethical property information: **Reliability** - a property that determines the degree of confidence in it. The Ukrainian regulations is not being used.

Information, Forms and Features

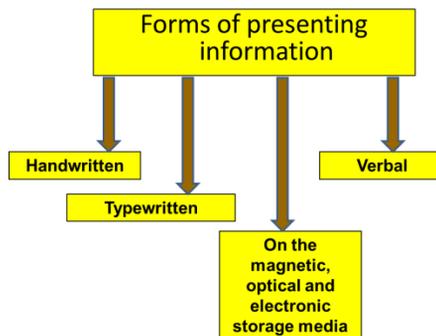


Fig.2 – The forms of presenting information

Module 1. Information security. Definitions, norms, standards
The Law of Ukraine “On Information”. [5].

According to the Law, Information is documentary or publicly announced news about events and phenomena in society, state and environment.

The Law shall apply to informational relations arising in all spheres of life and activity of society and state while getting, using, spreading and keeping information.

The subjects of informational relations are:

- Ukrainian citizens;
- legal entities;
- state;
- foreigners;
- stateless persons;
- foreign legal entities;
- foreign states;
- International organizations.

Objects of informational relations are documentarily or publicly announced information about events and phenomena in the spheres of politics, economy, culture, healthcare, as well as in social, ecological, international and other spheres.

All Ukrainian citizens, legal entities and state bodies have the right to information. It means the possibility of free getting, using, spreading and keeping information necessary for the exercise of their rights, freedoms and legal interests, fulfillment of tasks and functions. Realization of the right to information by citizens, legal entities and state shall not violate public, political, economic, social, spiritual, ecological and other rights, freedoms and legal interests of other citizens, as well as rights and interests of legal entities. Each citizen shall be ensured free access to information referring to him personally.

According to the Law, informational activity is a totality of actions aimed at satisfaction of informational needs of citizens, legal entities and state.

Module 1. Information security. Definitions, norms, standards

The key directions of informational activity are:

- politics;
- economy;
- social sphere;
- spiritual sphere;
- ecology;
- science and technology;
- International sphere.

The types of informational activity are:

- getting information (obtaining, acquisition, accumulation of documentary or publicly announced information by citizens, legal entities or state);
- using information (satisfaction of informational needs of citizens, legal entities and state);
- spreading information (distribution, publication, sale of documentary or publicly announced information);
- keeping information (ensuring the proper condition of information and its material carriers).

According to the Law, the main types of information are as follows:

- statistical information (official documentary state information giving quantity characteristics of mass phenomena and processes which happen in economic, social, cultural and other spheres of life);
- administrative information (data) (official documentary data giving quantity characteristics of mass phenomena and processes which happen in economic, social, cultural and other spheres of life and which are collected, used, spread and kept by bodies of state power (except for bodies of state statistics), bodies of local self-government, legal entities according to the legislation, for fulfillment of administrative duties and tasks which are in their competence);
- mass information (publicly spread printed and audio visual information);
- information about activity of state bodies of power and bodies of local and regional self-government (official documentary information created in the course of current activity of legislative, executive and judicial power, bodies of regional and local self-government);
- legal information (totality of documentary or publicly announced information about law, its system, sources, realization, legal facts, legal relations, crimes and fight against them, their preventive measures etc.);

Module 1. Information security. Definitions, norms, standards

- information about personality (totality of documentary or publicly announced information about personality);
- information of reference encyclopedic character (systematized, documentary or publicly announced information about social, state life and environment);
- sociological information (documentary or publicly announced information about relation of separate citizens and social groups to social events and phenomena, processes, facts).

By regime information is divided into:

- open information;
- information with limited access.

Information with limited access by its legal regime is divided into:

- confidential information (information which is in possession, use and disposal of separate natural persons or legal entities and is spread by their wish according to the provided by them conditions);
- secret information (information containing the facts making state and other types of secret as provided for by law. Spreading such information causes damage to person, society and state).

Pursuant to the Law, information is the object of ownership right of natural persons, legal entities and state. Grounds for arising the ownership right to information are:

- making information with one's own strengths and at one's own expense;
- agreement on making information;
- agreement containing conditions of the ownership right to information transfer to other person.

The Law determines rights and obligations of participants of informational relations.

The Law sets prohibition of censorship and interference in professional activity of journalists and mass media by the bodies of state power or bodies of local self-government, by their officials.

The Law also establishes responsibility for violation of the legislation on information.

1.1.3 Forms of Information and Channels of Distribution

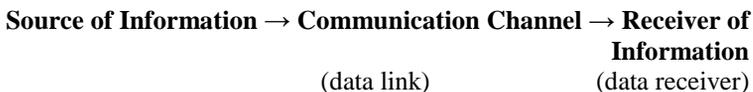


Fig. 3 - Channels of Information's distribution

For information process requires a source of information, channel information, and the consumer. Source transmits (sends) the information and its receiver receives (accepts). The information transmitted from the source reaches to the receiver via a signal (code). Changing the signal provides information.

Ukraine State Standards DSTU 3396.2-97. Protection of information/Technical protection of information/Terms and definitions [3].

Technical Information Leakage

6.1 **Media Information.** Material object containing classified information. Carrier of information is material object, including information from limited access.

6.2 **Informative signal.** Physical field and (or) a chemical containing classified information. Informative signal, physical and chemical field, containing information with limited access.

6.3 **Technical information leakage set of media information environment of its distribution and technical means of intelligence (of technical) channel aggregates.** Leaks of information carrier of information's environment. Technical channel information leakage totality of the information carrier, the medium of its distribution and technical reconnaissance.

6.3.1 **Arbitrarily (technical) information leakage;** Unintentional information leakage. Technical information leakage, in which the media and (or) medium to spread spontaneously formed (Inadvertent technical information leakage channel technical channel information leaks in which media and (or) the medium of their distribution are formed spontaneously).

6.3.2 **Artificial (technical) information leakage;** Deliberate information leakage (Intentional (technical) information leakage channel).

6.4 **Side electromagnetic radiation and indication;** Electromagnetic radiation and the suggestion that the result is a by-functioning technical means and can be a carrier of information.

Module 1. Information security. Definitions, norms, standards

Questions for self-control.

1. What is information?
2. Presentation forms of the information.
3. The concept of information and its features.
4. Basic property of the information.
5. Information threats species classification.
6. Forms of information and channels of its distributions.
7. What is the resource-based approach to information technology?
8. Threats to Information Security and their classification in a number of basic features.

1.2 Threats to information security

The Strategy of Cybersecurity of Ukraine is approved

President Poroshenko signed a decree, which brought into effect the decision of the National Security and Defense Council of Ukraine on 27 January 2016 "On the Cybersecurity Strategy of Ukraine".

The document notes that along with the advantages of the modern digital world and the development of information technologies, cases of illegal collection, storage, use, destruction, distribution, personal data, illegal financial transactions, thefts and fraud in the Internet are actively spreading. Modern information and communication technologies can be used to commit terrorist acts, including by violating the regular operating modes of automated process control systems at infrastructure facilities. Politically motivated activity in cyberspace in the form of attacks on government and private Internet sites is becoming more widespread.

The concept notes: "The economic, scientific and technical, information sphere, the sphere of state administration, the defense-industrial and transport complexes, the electronic communications infrastructure, the security and defense sector of Ukraine are becoming increasingly vulnerable to intelligence and subversive activities of foreign special services in cyberspace. This is facilitated by a broad, sometimes dominant, presence in the information infrastructure of Ukraine of organizations, groups, individuals directly or indirectly associated with the Russian Federation. "

Module 1. Information security. Definitions, norms, standards

The goal of the Cybersecurity Strategy of Ukraine is to create conditions for the safe functioning of cyberspace, its use in the interests of the individual, society and the state.

The basis of the national cyber security system will be the Ministry of Defense of Ukraine, the State Service for Special Communications and Information Protection of Ukraine, the Security Service of Ukraine, the National Police of Ukraine, the National Bank of Ukraine, intelligence agencies.

The Strategy provides for a set of measures, priorities and directions for ensuring cybersecurity of Ukraine, in particular, the creation and operational adaptation of public policies aimed at the development of cyberspace and achieving compatibility with relevant EU and NATO standards, the formation of a competitive environment in the field of electronic communications, the provision of information security services and Cyber defense; Attraction of expert potential of scientific institutions, professional and public associations to the preparation of draft concept papers in this field; Increase digital literacy of citizens and a culture of safety behavior in cyberspace; The development of international cooperation and support for international initiatives in the field of cybersecurity, including the deepening of Ukraine's cooperation with the EU and NATO.

In addition, the NSDC instructed the Cabinet of Ministers, together with the SBU, the Foreign Intelligence Service and with the participation of the National Institute for Strategic Studies, to approve within two months a plan of measures for 2016 to implement the Cybersecurity Strategy of Ukraine and further develop and approve such plans annually and inform about the implementation status.

The Council also decided to establish the National Coordinating Center for Cybersecurity as the working body of the National Security and Defense Council.

1.2.1 The Concept of an Automated System Processing

Automated Information System [Automated System] - interrelated set of tools, methods and personnel used for the storage, processing and issuance information in order to achieve this goal [4].

Current understanding of the information system is to use as the main technical means of information processing computer equipment. In addition, the technical implementation of the information system itself did not mean like that are not taken into account the role of humans for which designed and produced information without which its preparation and presentation.

It is also necessary to understand the difference between computers and information systems. Computers equipped with specialized on-brought into the means and technical base is a tool for information systems. The necessary components of an information system is the staff that interacts with computers and telecommunications.

Automation System Job is to serve two opposing flows of information, input new information and issuing current information on request.

Since the main objective information system - customer service, it is constructed so that the answer to any question was granted quickly and was quite full. This is ensured by the standard procedures for information and the fact that these systems are in order.

In the field of information systems solve several major problems:

- analysis and forecasting flows of information, conveyed in society;
- study ways of presentation and storage of information, creation of special languages for formal description information of different nature, development of special methods of compression and encoding information, annotating three-dimensional documents and their abstracts;
- construction of various procedures and technical means for their implementation, with which you can automate the process of extracting information from documents that are not designed for computers, and focused on the perception of man;
- creating information retrieval systems that are able to accept requests for data warehousing, formulated in natural (human) language, as well as special requests for languages such systems;
- networking, storage, processing and transmission of information, which include information databases, terminals, processing centers and communications.

The structure of the information system is a set of its individual parts, called subsystems.

Subsystem - this part of the system highlighted on any grounds.

Module 1. Information security. Definitions, norms, standards

The overall structure of the information system can be viewed as a set of subsystems regardless of scope. In this case we speak of the structural features of the classification and subsystems providing call. Thus, the structure of any information subsystem can be represented by providing a set of subsystems.

Among providing subsystems usually distinguish information, technical, mathematical, software, organizational and legal support.

Appointment subsystem information support is timely formation and delivery of reliable information for decision making.

Information support [dataware] - set a single system of classification and coding of information, standardized documentation systems, schemes of information flows circulating in the organization and methodology of databases.

Legal regulatory and organizational maintenance of information's systems – The Law of Ukraine “On protection of information in telecommunication systems” {come into effect of 07.05.1994} [4].

This Law regulates relations in the field of information protection in information, telecommunication and information and telecommunication systems.

Definitions

In this Law the following terms have the following meanings:

Information owner - a natural or legal person who owns the information; owner of the system - natural or legal person who owns the system;

User of information in the system - natural or legal person who, in accordance with legislation received the right to access information in the system;

Information (automated) system - organizational and technical system in which information processing technology is implemented using hardware and software;

Telecommunications system - a set of hardware and software designed for the exchange of information by transmission, emission or reception of it in the form of signals, signs, sounds, moving or still images, or otherwise;

Information and Telecommunication System - a set of information and telecommunication systems, information processing in the act as a whole;

Information security system - activities aimed at preventing unauthorized actions on information in the system.

The Automated System

Automated System (AS) - is the organizational and technical system that implements information technology and combines:

- Computing System (CS);
- Physical Environment;
- Personnel;
- Information that is processed.

CS - a set of software and hardware designed for data processing.

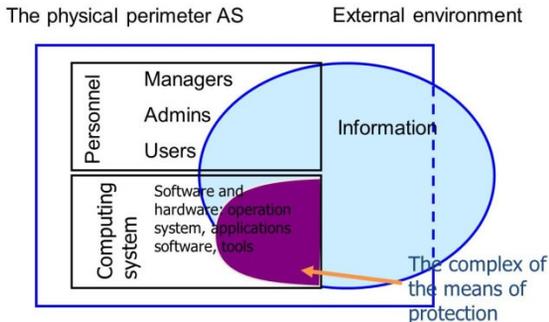


Fig.4 – The complex of the means of protection at information and telecommunication system

Information and Telecommunication Systems (ITS) include any system that meets one of three types of automated systems:

Information System - organizational and technical system that implements the technology of information processing by means of computers and software;

Telecommunications system - organizational and technical system that implements the technology of information exchange by means of hardware and software by transmitting and receiving information in the form of signals, signs, sounds, images, or other means;

Integrated system - a set of two or more related information and (or) telecommunication systems, in which the operation of one (or more) of them depends on the results of another operation (other) so that this totality in the process of interaction can be viewed as a single system.

1.2.2. Information Security

Information Security - state media, which ensured the preservation of the information properties defined security policy.

The purpose of protection - observance of the technology of information processing. For this:

- Protection features specified information (confidentiality, integrity, availability)

- Protect hardware and software against damage

The tasks of protection:

- Anti defined set of threats

- Performing a given security policy

Information security in AS

Activities relating to the security of information processed in the AS, and the AS as a whole, which can prevent or complicate the possibility of threats, and reduce the size of the potential loss as a result of threats.

Ukraine State Standards DSTU 3396.2-97 Protection of information. Technical protection of information. Terms and definitions [3] .

The threat to information

Information's leak. Uncontrolled distribution of information, which leads to its unauthorized reception.

Violating the integrity of the information. Distortion of information, its destruction or destruction.

Block information. Impossibility authorized access to information.

Threat for information. Leak, the possibility of blocking or destroying the integrity of information. Threats to information may arise during the use of technology or technologies, imperfect information security.

Threats model information. Formalized description of the methods and means of implementing threats to information.

Access to Information. The ability to obtain, information processing, blocking and (or) breach of integrity.

Unauthorized access (to information). Access to information for which violates the established order of its mandated and (or) legal norms.

Embedded device; bookmark. Secretly installed technical means which jeopardizes information.

Software bookmark. Secretly introduced a program that poses a threat to the information contained in the computer.

Module 1. Information security. Definitions, norms, standards

Computer virus. A program that multiplies and spreads spontaneously. A computer virus may interfere with the integrity of the information, software and (or) mode of computing.

Special effects. The impact on technical measures leading to realization of threats to information.

Technical intelligence. Unauthorized obtaining information through technical means and its analysis. Technical intelligence model. Formalized description of methods, tools and technical intelligence capabilities.

1.2.3 Threats to Information Security and their Classification in a Number of Basic Features

Adverse effects: effect, which reduces the value of information resources.

Threat: any circumstances or events that can cause violation of information security and (or) damage to AS.

Attack: attempt to implement of the threat.

The vulnerability of the system: the inability of the system to resist implementation of a set of threats or threats.

Secure automated system: automated system that can provide protection of information processed from certain threats.

Questions for self-control.

1. The concept of information security.
2. The basic components.
3. The importance of the problem.
4. Characteristics of threats to information security.
5. The basic principles and methods of information security.
6. Automated systems general information security.
7. What are the models, methods and tools, components target resources to meet the challenges, problems, challenges?
8. The main threats to information security.
9. Basic definitions and criteria for classification of threats to information systems.
10. Threats to availability.
11. Malicious software.
12. The concept of information security.
13. The basic components.
14. The importance of the problem.

- Module 1. Information security. Definitions, norms, standards
15. Characteristics of threats to information security.
 16. The basic principles and methods of information security.
 17. Managing information security risks. Basic concepts. The main stages of risk management.

3. Regulatory and methodological support of information protection in the USA, Canada, the EU and the Russian Federation

3.1 Trusted Computer System Evaluation Criteria (Orange book)

Appointment Information Security Standards [20]. Eliminating defects defense - a tactical way of building secure systems, or approach "from bottom". Let's get rid of some vulnerabilities and increase security of the system. Does not assess the completeness or fulfillment of this task, or the current level of protection.

Standards define a strategic approach to building secure systems, or approach "from above".

- Security system - characteristic quality, for it does not exist units.
- Different experts offer different ways to improve the security of the system and differently appreciate it.

The only way to establish the scale of assessment of security systems, and coordinate the views of various experts, is to develop a standard.

The standard must regulate:

- The concept of information security;
- Approaches to achieve it;
- Requirements for systems and ways of their implementation;
- A system of evaluation criteria of security systems;
- Procedures for the evaluation of these criteria.

The standards provide a framework for the approval requirements to systems developers:

- Consumers (users and owners of the information that they processed);
- Experts who assess the security of information systems (and if necessary - also public or institutional bodies that authorize the operation of the system and process certain information in it).

The development of standards for information security:

TCSEC - Evaluation criteria of protected computer systems of US Department of Defense ("Orange Book"), 1983.

ITSEC - European safety criteria for information technology in 1991.

Module 1. Information security. Definitions, norms, standards
"Guidance Documents Technical Commission of Russia" 1992.
FCITS - Federal safety criteria Information Technology USA, 1992.

Trusted Computer System Evaluation Criteria

Trusted Computer System Evaluation Criteria. - US Department of Defense. - CSC-STD-001-83, 1983.

Trusted Network Interpretation. - National Computer Security Center. NCSC-TG-005, Version 1, 1987.

Trusted Database Management System Interpretation. - National Computer Security Center. NCSC-TG-021 Version 1 1991.

The Interpreted Trusted Computer System Evaluation Criteria Requirements. - National Computer Security Center. NCSC-TG-007-95, 1995.

Determining the safety system - system that supports access control such that only authorized users or processes acting on their behalf, are able to read, write, create and erase information.

Six requirements for secure system:

- Security policy.
- Labels security.
- Identification and authentication.
- Registration and accounting.
- Correctness remedies.
- Continuity of protection.

TCSEC: list of requirements:

- Privacy policy;
- Discretionary access control;
- Reusing objects;
- Security labels;
- Integrity label security
- Exports seen information;
- Labels authority of subjects;
- Labels devices;
- Access control credentials;
- Audit of the security system;
- Identification and authentication;
- Direct interaction with complex means of protection;
- Registration and accounting events;
- Correctness operation;
- System Architecture;
- Integrity systems;

Module 1. Information security. Definitions, norms, standards

- Analysis of covert channels;
- Security management;
- Restoration:
 - correctness development;
 - security testing;
 - design and verification of specifications;
 - configuration;
 - distribution.
- Documentation:
 - user guide security;
 - guidelines security administrator;
 - documentation of the testing process;
 - documenting the development process.

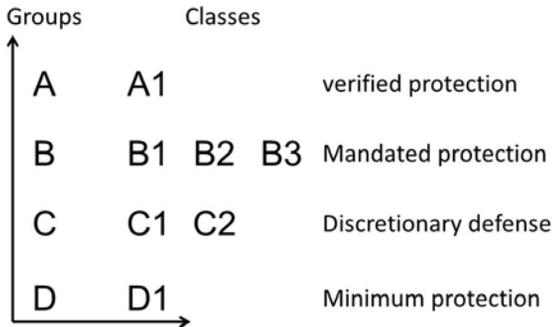


Fig. 5 - TCSEC: classification systems

Conclusions of TCSEC

The first attempt to create a unified security standard that is designed for developers, consumers and professionals with certification computer's system.

Oriented systems of special (military) applications, mainly in operating systems (dominated by the requirements of confidentiality).

Criteria guarantees of protection means and adequacy of security policies developed enough.

Assessment system comes to check compliance with a predefined classes. Scale classes are hierarchical.

Module 1. Information security. Definitions, norms, standards

1.3.2 Federal Criteria for Information Technology Security (FCITS).

Federal Criteria for Information Technology Security [Federal Information Technology Security Criteria] - information security standard developed by the National Institute of Standards and Technology USA (NIST) and the National Security Agency (NSA) in 90 years for use in US federal standard of information processing (Federal Information Processing Standard), which was to replace the "Orange Book". Version 1.0, 1992.

"Federal Criteria" cover virtually the entire spectrum problem related to the protection and safety, as well as include all aspects of confidentiality, integrity and efficiency. The main objects of application security requirements criteria are products of information technology (IT products) and information processing systems.

The key concept of the concept of information security "Federal Criteria" is the concept of the protection profile.

According to the "Federal Criteria" development process of information processing is carried out in a sequence of these basic steps:

- development and analysis of the profile of protection;
- developing and qualifying the analysis of IT products;
- configuration and certification of information processing.

"Federal Criteria" regulate only the first phase of the scheme - the development and analysis of the profile of protection. The process of creating IT products and layout of information processing are outside this standard.

Definition of universal and open for further development of a set of basic security requirements that impose to modern information technology.

Improving the existing requirements and safety criteria.

Reconciliation among themselves the requirements and criteria for IT security technologies adopted in different countries.

The statutory core principles of information security.

FCITS - stages of product development IT

Development and analysis of the profile of protection.

The requirements set out in protection profiles define the functionality of IT products with security and operating conditions, the observance of which is guaranteed by compliance requirements.

Profile protect analyzed for completeness, consistency and technical correctness.

Development and analysis of products qualifying IT.

Module 1. Information security. Definitions, norms, standards

Developed IT products subjected to independent analysis, which - coincidence of the characteristics of the product profile protection requirements.

The layout and certification of information processing in general.

The resulting system must meet the requirements stated in the profile protection.

FCITS - structure protection profile

Description. Classification information is necessary to identify the profile in a special file cabinet.

Rationale.

Description of environmental exploitation, threats to the security provided for, and methods of using the product IT.

Functional requirements for IT products.

Requirements for product development IT technology.

Requirements analysis product qualification process IT.

FCITS - stages of product development IT.

Development and analysis of the profile of protection. The requirements set out in protection profiles define the functionality of IT products with security and operating conditions, the observance of which is guaranteed by compliance requirements. Profile protect analyzed for completeness, consistency and technical correctness.

Development and analysis of products qualifying IT. Developed IT products subjected to independent analysis, which - coincidence of the characteristics of the product profile protection requirements.

The layout and certification of information processing in general. The resulting system must meet the requirements stated in protection profiles

FCITS - profile structure protection.

Description profile protection. Classification information is necessary to identify the profile in a special file cabinet.

Justification profile protection. Description of environmental exploitation, threats to the security provided for, and methods of using the product IT.

Functional requirements for IT products.

Requirements for product development IT technology.

Requirements analysis product qualification process IT.

FCITS - functional requirements for IT products.

Module 1. Information security. Definitions, norms, standards

Implementing security policies, audit policies; Identification and authentication; Registration in the system; Providing direct interaction with complex remedies; Registration and record events.

Policy access control, discretionary access control; Access control credentials; Control hidden channels.

Politics ensure the efficiency, control over the distribution of resources; Providing resistance to failure.

Security Management.

Monitoring the interactions.

Logical defense complex remedies.

Physical protection of complex remedies.

Self-control is a means of protection.

Initialization and restoration of complex remedies.

Limitation of privileges when working with complex remedies.

Easy to use complex remedies.

FCITS - Requirements for technology development.

The process of development: Definition of a plurality of functions of complex remedies according to functional requirements;

Implementation of complex remedies: Determination of functional components of complex remedies; Definition interface complex remedies; Decomposition of complex remedies functional modules; Structuring complex remedies domains of security; Minimizing the functions and structure of complex remedies.

Guarantees the complex remedies.

Testing and analysis of complex remedies: Testing complex functions remedies; Analysis of possibilities breach of security; Analysis of covert channels.

Development environment: Tools; Controls the development process; The procedure of distribution.

Documentation: Documentation of the functions of complex remedies; Complete product documentation for IT (interfaces, components, modules, the structure of complex remedies, methods of design, source code and hardware specifications); Documentation of testing and analysis of IT products; Documenting Process Testing functions; Documenting analysis capabilities breach of security; Documenting the analysis of covert channels; Documentation of environmental and process development.

Escort: User Documentation; Guide security administration; The procedure for updating versions and bug fixes; The installation procedure.

Module 1. Information security. Definitions, norms, standards

FCITS - Requirements for the qualification process analysis.

Analysis: Analysis of architecture; Analysis of implementation.

Control: Control development environment; Control IT product support process.

Testing: Testing complex functions remedies manufacturer; Independent testing functions of complex remedies.

FCITS - Conclusions.

First proposed the concept of the protection profile.

The standard set of three independent claims.

Functional safety requirements are well structured.

Requirements for technology development make producers use modern programming techniques, allowing to confirm the safety of the product.

Requirements analysis process qualification generalized and do not contain specific methodologies.

No overall level of security through universal scale, independent ranking of proposed requirements of each group.

1.3.3. Information Technology Security Evaluation Criteria (ITSEC)

ITSEC - European criteria of the security systems.

The Information Technology Security Evaluation Criteria (ITSEC) is a structured set of criteria for evaluating computer security within products and systems. The ITSEC was first published in May 1990 in France, Germany, the Netherlands, and the United Kingdom based on existing work in their respective countries. Following extensive international review, Version 1.2 was subsequently published in June 1991 by the Commission of the European Communities for operational use within evaluation and certification schemes.

Since the launch of the ITSEC in 1990, a number of other European countries have agreed to recognize the validity of ITSEC evaluations.

The ITSEC has been largely replaced by Common Criteria, which provides similarly-defined evaluation levels and implements the target of evaluation concept and the Security Target document.

ITSEC Concepts.

The product or system being evaluated, called the target of evaluation, is subjected to a detailed examination of its security features culminating in comprehensive and informed functional and penetration testing. The degree

Module 1. Information security. Definitions, norms, standards of examination depends upon the level of confidence desired in the target. To provide different levels of confidence, the ITSEC defines evaluation levels, denoted E0 through E6. Higher evaluation levels involve more extensive examination and testing of the target.

Unlike earlier criteria, notably the TCSEC developed by the US defense establishment, the ITSEC did not require evaluated targets to contain specific technical features in order to achieve a particular assurance level. For example, an ITSEC target might provide authentication or integrity features without providing confidentiality or availability. A given target's security features were documented in a Security Target document, whose contents had to be evaluated and approved before the target itself was evaluated. Each ITSEC evaluation was based exclusively on verifying the security features identified in the Security Target.

ITSEC - safety classes.

F-C1, F-C2, F-B1, F-B2, F-B3 - meet TCSEC;
F-IN - increased requirements for integrity;
F-AV - increased requirements to ensure efficiency (real-time);
F-DI - distributed systems with special requirements for integrity;
F-DC - distributed systems with special requirements for confidentiality;
F-DX - distributed systems with special requirements for confidentiality, integrity and inability denial of authorship.

ITSEC - criteria guarantees.

- Performance measures;
- Matching set of remedies set goals;
- Mutual consistency of different tools and protection mechanisms;
- The ability to withstand attacks remedies;
- The possibility of practical use of architectural flaws remedies;
- Easy to use remedies;
- The possibility of practical use of functional deficiencies remedies.

The criteria for correctness.

- The process of developing:
 - Specification of safety requirements;
 - Development architecture;
 - Creating a working draft;
 - Realization.
- Development environment:
 - Controls configuration;

Module 1. Information security. Definitions, norms, standards
Used programming languages and compilers;
Safety development environment.

- Operational documentation:
 - User guide;
 - Guidelines administrator.
- Operating Environment:
 - Delivery and installation;
 - Starting and operation.

Performance measures.

- Matching set of remedies set goals;
- Mutual consistency of different tools and protection mechanisms;
- The ability to withstand attacks remedies;
- The possibility of practical use of architectural flaws remedies;
- Easy to use remedies;
- The possibility of practical use of functional deficiencies remedies.

Conclusion of the ITSEC

The main achievement - the introduction of the concept of safeguards and indicate a particular scale for criteria guarantees.

Failure of a single hierarchical grading scale systems.

ITSEC closely related TCSEC (not completely separate document).

Recognition of the possible availability of certified deficiencies in systems and the introduction of the criterion of the possibility of using flaws protection.

1.3.4. Canadian Security Criteria (CTCPEG)

The Canadian Trusted Computer Product Evaluation Criteria (CTCPEC) is a computer security standard published in 1993 by the Communications Security Establishment Canada to provide an evaluation criteria on IT products. It is a combination of the TCSEC (also called the Orange Book) and the European ITSEC approaches.

The Common Criteria for Information Technology Security Evaluation (abbreviated as Common Criteria or CC) is an international standard (ISO/IEC 15408) for computer security certification. It is currently in version 3.1 revision 4 [20].

Common Criteria is a framework in which computer system users can specify their security functional and assurance requirements (SFRs and

Module 1. Information security. Definitions, norms, standards SARs respectively) through the use of Protection Profiles (PPs), vendors can then implement and/or make claims about the security attributes of their products, and testing laboratories can evaluate the products to determine if they actually meet the claims. In other words, Common Criteria provides assurance that the process of specification, implementation and evaluation of a computer security product has been conducted in a rigorous and standard and repeatable manner at a level that is commensurate with the target environment for use.

Common Criteria is used as the basis for a Government driven certification scheme and typically evaluations are conducted for the use of Federal Government agencies and critical infrastructure.

Key concepts.

Common Criteria evaluations are performed on computer security products and systems.

Target Of Evaluation (TOE) – the product or system that is the subject of the evaluation.

The evaluation serves to validate claims made about the target. To be of practical use, the evaluation must verify the target's security features. This is done through the following:

Protection Profile (PP) – a document, typically created by a user or user community, which identifies security requirements for a class of security devices (for example, smart cards used to provide digital signatures, or network firewalls) relevant to that user for a particular purpose. Product vendors can choose to implement products that comply with one or more PPs, and have their products evaluated against those PPs. In such a case, a PP may serve as a template for the product's ST (Security Target, as defined below), or the authors of the ST will at least ensure that all requirements in relevant PPs also appear in the target's ST document. Customers looking for particular types of products can focus on those certified against the PP that meets their requirements.

Security Target (ST) – the document that identifies the security properties of the target of evaluation. The ST may claim conformance with one or more PPs. The TOE is evaluated against the SFRs (Security Functional Requirements. Again, see below) established in its ST, no more and no less. This allows vendors to tailor the evaluation to accurately match the intended capabilities of their product. This means that a network firewall does not have to meet the same functional requirements as a database management system, and that different firewalls may in fact be evaluated

Module 1. Information security. Definitions, norms, standards against completely different lists of requirements. The ST is usually published so that potential customers may determine the specific security features that have been certified by the evaluation.

Security Functional Requirements (SFRs) – specify individual security functions which may be provided by a product. The Common Criteria presents a standard catalogue of such functions. For example, a SFR may state how a user acting a particular role might be authenticated. The list of SFRs can vary from one evaluation to the next, even if two targets are the same type of product. Although Common Criteria does not prescribe any SFRs to be included in an ST, it identifies dependencies where the correct operation of one function (such as the ability to limit access according to roles) is dependent on another (such as the ability to identify individual roles).

The evaluation process also tries to establish the level of confidence that may be placed in the product's security features through quality assurance processes:

Security Assurance Requirements (SARs) – descriptions of the measures taken during development and evaluation of the product to assure compliance with the claimed security functionality. For example, an evaluation may require that all source code is kept in a change management system, or that full functional testing is performed. The Common Criteria provides a catalogue of these, and the requirements may vary from one evaluation to the next. The requirements for particular targets or types of products are documented in the ST and PP, respectively.

Evaluation Assurance Level (EAL) – the numerical rating describing the depth and rigor of an evaluation. Each EAL corresponds to a package of security assurance requirements (SARs, see above) which covers the complete development of a product, with a given level of strictness. Common Criteria lists seven levels, with EAL 1 being the most basic (and therefore cheapest to implement and evaluate) and EAL 7 being the most stringent (and most expensive). Normally, an ST or PP author will not select assurance requirements individually but choose one of these packages, possibly 'augmenting' requirements in a few areas with requirements from a higher level. Higher EALs do not necessarily imply "better security", they only mean that the claimed security assurance of the TOE has been more extensively verified.

So far, most PPs and most evaluated STs/certified products have been for IT components (e.g., firewalls, operating systems, smart cards).

Module 1. Information security. Definitions, norms, standards

Common Criteria certification is sometimes specified for IT procurement. Other standards containing, e.g., interoperation, system management, user training, supplement CC and other product standards. Examples include the ISO/IEC 17799 (Or more properly BS 7799-1, which is now ISO/IEC 27002) or the German IT-Grundschutzhandbuch [22].

Details of cryptographic implementation within the TOE are outside the scope of the CC. Instead, national standards, like FIPS 140-2 give the specifications for cryptographic modules, and various standards specify the cryptographic algorithms in use.

More recently, PP authors are including cryptographic requirements for CC evaluations that would typically be covered by FIPS 140-2 evaluations, broadening the bounds of the CC through scheme-specific interpretations.

Some national evaluation schemes are phasing out EAL-based evaluations and only accept products for evaluation that claim strict conformance with an approved PP. The United States currently only allows PP-based evaluations. Canada is in the process of phasing out EAL-based evaluations.

1.3.5. State Technical Guidance Document of Russia Federation “On information protection”.

The concept of protection of computer equipment from unauthorized access to information.

Computer equipment. Protection against unauthorized access to information. Indicators of security against unauthorized access to information.

Automated Systems. Protection against unauthorized access to information. Classification of the Automated Systems and the requirements for data protection.

Computer equipment: 7 classes from 7 to 1.

Automated Systems: 3 groups, 9 classes.

Automated Systems Group 3, which operates one user who has been admitted to all the information. 3A and 3B classes.

Automated Systems Group 2, in which users have the same access privileges to all information. Classes 2A and 2B.

Group 1. Multiuser Automated Systems processes information at different levels of confidentiality, users have different access rights. Classes 1D, 1G, 1B, 1A.

Module 1. Information security. Definitions, norms, standards

Guidance documents Russia on information protection: conclusions.

Like the "Orange Book", the documents focus on the military use of the system.

The concept of "security policy" is treated exclusively as the maintenance of secrecy and lack of unauthorized access. Remedies are guided exclusively at countering external threats.

There is no requirement for protection from health threats and the adequacy of the implementation of security policies. By the very structure of the system and its operation requirements are not imposed.

Use single universal scale of the degree of protection. Ranking requirements by grade with absolute ease.

1.3.6. Common Criteria for Information Technology Security Evaluation (ISO / IEC 15408: 1999).

Orange Book (first evaluation standard, the US Department of Defense) [20] with the improvement has grown into a series of standards, which are summarized in the "Harmonized criteria for European countries", on the basis of which was created by ISO / IEC 15408: 1999, which is often called Common Criteria - Common Criteria (CC). They founded GOST 15408-1999.

CCITSE contains two basic types of security requirements:

- functionality (applicable to the functions (services) Security and implementing their mechanisms) (identification, authentication, access control, auditing, etc.);
- assurance requirements (imposed on the technology development, testing, vulnerability analysis, delivery, maintenance, operational documentation, to all stages of information technology product life cycle).

The main advantages of CCITSE are:

1. A reasonably complete set of security requirements of information technology;
2. Separation of the security requirements into functional requirements and security assurance requirements;
3. Ordering and classification requirements of the hierarchy;
4. Component leveling requirements of families and classes according to the degree of completeness and stiffness, as well as their grouping into functional requirements and performance packages confidence levels;

Module 1. Information security. Definitions, norms, standards

5. Metastandard - CC help to form sets of requirements as set out in CC standardized structures (protection profiles and security targets), which is already oriented to a specific product, but do CC no matter what is not oriented;

6. Open for future build a set of requirements.

The main disadvantages are the CCITSE:

1. Some obsolete - as with any static document in our rapidly changing world;

2. It is not very clear and logical division;

3. The bad and illogical classification;

4. Meta standard and related vagueness.

Basic concepts and principles of the CCITSE:

Evaluation object (EO) - specific design, hardware and software product or information system - a specific embodiment of information technologies with a specific purpose and specific conditions of use, or usable alone or designed to work in the other systems.

Evaluation object does not exist on their own, and in the security environment, which includes everything that has to do with its security. In particular:

1. Administrative Environment - position security policies and programs, taking into account the features of the EO;

2. Procedural environment - the physical environment of the EO and physical protection measures, staff, and his knowledge, experience, made operational and other procedures;

3. Software and hardware environment - the purpose of evaluation of the object and the expected field of application, resources that require protection by means of EO, and the like).

4. Security Assumptions - isolated object of evaluation from the general context, determine the limits of the consideration (the truth of these assumptions are accepted without proof, and of many possible shown only that obviously needed to ensure the security of the EO);

5. EO security threats, the presence of which in the medium is established or suspected (as a result of the analysis of the set of feasible threats selects only those damages which requires reduction);

6. Security policy provisions to be applied to the evaluation of the project;

7. The legal environment - laws and regulations affecting the EO.

8. Security objective - formulated on the basis of assumptions, taking

Module 1. Information security. Definitions, norms, standards into account the threats and security policies and regulations aimed at countering threats can be purely procedural - non-technical - and software and hardware

9. Safety requirements - EO requirements that achieve security objectives.

"Common Criteria" it includes an extensive library of security requirements, which we will consider throughout the course, and which helps to develop specific security requirements required to achieve the security objectives of evaluation object. This library is structured requirements as follows:

1. Classes - the biggest taxon, grouping requirements in subjects;
2. The families within the class distinguished requirements for rigor, and other characteristics;
3. Components - minimum requirements that appear as a single entity in the development and operation;
4. Elements - indivisible requirements.

For each specific "Common Criteria" used the concept of protection and security job profile. It - regulations governing the protection. Protection Profile contains a standard set of security requirements to be met by a product or a system of a certain class. Security Target contains a set of specific safety requirements for the design, the implementation of which can solve the problem of security. In developing the often used as packages of requirements - frequently used components together, united to achieve certain security purposes (combination is safety purposes).

By grouping the safety requirements in the protection and the safety profiles of the job, check whether the EO and its features with these requirements by answering the questions:

- Is responsible for yourself EO security function functional requirements? (Theoretical aspect)
- Do not errors in the implementation of safety features allowed? (Practical aspect)

"Common Criteria" does not prescribe a specific methodology for the development of the EO, but provide for a presentation of several project levels with its decomposition and detail. For safety requirements should be functional specification, then the top-level project, the required number of intermediate levels, the lower level of the project, then, depending on the product type, source code or hardware diagrams and, finally, the

Module 1. Information security. Definitions, norms, standards implementation in the form of executable files, hardware products and so on EO between levels, must demonstrate compliance with the submission, that is the essence of all the higher levels are required to appear and "lower" and "below" there is no place unnecessary entities, not due to the needs of higher levels.

Evaluation methodology.

This topic and the subsequent topics we will look briefly, since it relates more to security management than directly to the Data Protection Act. Following the principles of structural decomposition, the developers have identified in the evaluation of three tasks (steps): entrance task, the task of evaluation, the output task. Input problem has to do with the objects submitted for evaluation or documents regulating its activity (evidence). Its purpose - to make sure that the version of the EO provided for evaluation is correct and properly protected. The challenge is to assess:

- Security Target evaluation;
- Control evaluation EO configuration;
- Evaluation of the documentation for the transfer of the TOE user and operational documentation;
- Assessment of the developer documentation;
- Assessment guidelines;
- Assessment of support EO life cycle;
- Assessment tests;
- Testing;
- Vulnerability analysis assessment.

Allowed random inspection certificates, tests, covert channel analysis results, the requirements for content and presentation of certificates, sample testing. In other situations, this method can only be applied in exceptional cases. The sample size must be justified mathematically and economically, but in the analysis of the implementation of evaluation object, it must be at least 20%. Errors discovered during random inspection, divided into systematic and random. After correcting the bias necessary to make a new selection; after random this is not required.

The necessary assessment element - verification of the internal consistency of each of the presented evidence, as well as foreign mutual consistency of different evidence. Internal consistency checked first for entities with multiple views: For specifications and projects at all levels, as well as guidelines. Checking external coherence is made to function definitions, security settings, procedures and events related to security, because these descriptions may be contained in different documents.

Module 1. Information security. Definitions, norms, standards

Output task set itself the task to formulate comments and receive the technical evaluation report - the main output document, enabling repeated use of evaluation findings. The recommended structure of the report is as follows:

- Introduction;
- Architectural (high-level) description assessment examination of the main components;
- A description of the evaluation process, the applied methods, methodologies, tools, and standards;
- Representation of evaluation results;
- Conclusions and recommendations;
- A list of the evidence submitted;
- A list of abbreviations, a glossary of terms;
- A list of comments.

Confidence Safety Requirements

Confidence in the interpretation of "Common Criteria" - is the basis for the confidence that the product meets the IT security objectives. Trust is provided through active research (evaluation) of IT products. A trust security requirement cover the entire life cycle of IT products and involves the following steps:

- estimated Security Target (ST) and the Protection Profiles (PP), have become sources of safety requirements;
- analyzes the different views of the object of the project assessment and correspondence between them, as well as compliance with the requirements of each security;
 - checked the processes and safety procedures and their application;
 - examines the records;
 - verified the evidence;
 - analyzes the tests and their results;
 - analyzes the vulnerability assessment of the object;
 - conduct independent testing, including test "hacks" (hereinafter referred to as penetration testing).

Each requirement of trust belongs to one of three types:

- Developer action;
- Presentation and content of the evidence;
- Evaluator action.

Assurance requirements are divided into 10 classes, 44 families and 93 component. Here are the main classes:

- APE - Protection Profile evaluation;

Module 1. Information security. Definitions, norms, standards

- ASE - Security Target evaluation;
- ADV - development;
- ALC - support life cycle;
- ACM - configuration management;
- AGD - leadership;
- ATE - testing;
- AVA - vulnerability assessment;
- ADO - Delivery and operation;
- AMA - confidence support.

Permission seven trust **evaluation levels** (EAL), containing useful practical application of a combination of components, ordered according to the degree of amplification. To increase the level of trust will help to further action:

- expanding the boundaries of object of evaluation;
- increase the level of detail considered aspects of the EO;
- increasing consideration of severity, use of more formal methods of verification.

1.3.7. Short review of the cybersecurity strategies and developments in the cybersecurity policy area in the Eastern Partnership region.

1.3.7.1 European Union Cybersecurity Strategy

The EU Cybersecurity Strategy [23] outlines the European Union's vision to promote an open, safe and secure cyberspace. The approach is articulated through five strategic priorities:

- Achieve cyber resilience;
- Drastically reduce cybercrime;
- Develop cyberdefence policy and capabilities;
- Develop the industrial and technological resources for cybersecurity;
- Establish a coherent international cyberspace policy for the European Union and promote core EU values.

To implement the strategy, the Commission has proposed a Directive concerning measures to ensure a high common level of network and information security across the Union (Network Information Security Directive) [24] with the following elements:

- Establishment of Computer Emergency Response Teams (CERTs): Member States are required to adopt a national strategy that sets out concrete policy and regulatory measures to maintain a level of network and

Module 1. Information security. Definitions, norms, standards information security. This includes designating a national competent authority for information security and setting up a CERT that is responsible for handling incidents and risks.

- Co-operation network: the competent authorities in EU Member States and the European Commission will form a co-operation network to coordinate against risks and incidents affecting network and information systems. The network will exchange information between authorities, provide early warnings on information security issues and agree on a co-ordinated response in accordance with an EU NIS co-operation plan.

- Security requirements: Member States must ensure that public and private sector take appropriate technical and organisational measures to manage the security risks to networks and ICT; these must guarantee a level of security appropriate to the risks and should prevent and minimise the impact of security incidents affecting the services they provide.

- Incident reporting: public and private sector must also notify the competent authority of incidents that have a significant impact on the continuity of these services. This is probably the most contentious item and both incident thresholds and the scope of public and private sector entities are still to be finalised. Where the security incident involves personal data, there may be a requirement to notify data protection authorities and individuals affected either existing EU data protection laws or the proposed EU data protection regulation which may be adopted in 2015 or thereafter.

- Use of standards: Member States are encouraged to use standards, such as ISO2700x series [25] for the implementation of security requirements.

- Enforcement: the competent authorities in each Member State are to be given powers to investigate cases of non-compliance of public bodies and market operators, which may include undergoing a security audit. They may also report criminal incidents to law enforcement authorities and work with data protection authorities where incidents involve personal data. The competent authorities and the single points of contact should be civilian bodies, subject to full democratic oversight and should not fulfil any tasks in the field of intelligence, law enforcement or defence or be organisationally linked in any way to bodies active in those fields [26].

- The strategy also highlights the importance [27] of adhering to the protection of fundamental rights, freedom of expression, personal data and privacy as well as ensuring democratic and efficient multi-stakeholder governance.

1.3.7.2 European Network Information Security Agency (ENISA)

Module 1. Information security. Definitions, norms, standards

ENISA published ‘An evaluation Framework for National Cybersecurity Strategies’ in November 2014 [28]. The Framework aims to evaluate cybersecurity strategies currently in place in eighteen European Union Member States. ENISA identified similarities between the different European strategies and their objectives; cybersecurity strategies often have objectives articulated around clusters (objectives), as does the European Cybersecurity Strategy [29]:

- To achieve cyber resilience: develop capabilities and cooperating efficiently within the public and private sector;
- To secure critical information infrastructures;
- To reduce cybercrime;
- To develop the industrial and technological resources for cybersecurity;
- To contribute to the establishment of an international cyberspace policy.

The Framework defines a set of key performance indicators (KPIs) to measure the effectiveness of a strategy. Examples of KPIs include the effective functioning of a CERT, a legislative framework, public-private cooperation, risk assessments for the national critical infrastructure, capacity building and the availability of a budget.

The European Union External Action Service (EEAS) also carried out activities related to cybercrime [30].

1.3.7.3 Good practice study on cybercrime reporting mechanisms

In September 2014, the Council of Europe published a Good practice study on cybercrime reporting mechanisms [31]. Building on the experience of several existing reporting mechanisms from both public and private sectors around the world (Belgium, EU, France, Mauritius, Netherlands, UK, USA), it aims at providing advice to countries which are considering or are in the process of setting up their own cybercrime reporting mechanisms.

The recommendations provided in this study are relevant for cybersecurity strategies as cybercrime reporting mechanisms contribute to identifying trends and fostering cooperation and information sharing. Besides their diversity, cybercrime reporting mechanisms share the fact that they make a positive contribution to the fight against cybercrime, in particular in the following aspects:

- Providing actionable information/complaints which can be the basis for

Module 1. Information security. Definitions, norms, standards investigations and prosecutions;

- Identification of cybercrime threats on citizens and organisations;
- Understanding and measuring trends;
- Establishing a channel of communication between citizens (victims/witnesses of cybercrime) and the authorities/initiatives in charge;
- Coordination between law enforcement and public authorities;
- Fostering a culture of public/private cooperation and information sharing.

1.3.7.4 Assessment criteria for a cybersecurity strategy

Analyzing examples from UK, Czech Republic, Estonia, ENISA and the EU NIS Directive, supporting measures are identified that will be used to assess the current status of EAP country strategies. These developments include:

- Cybersecurity strategy identification of cybercrime prevention as a key objective
- Establishment of computer emergency response teams (CERTs);
- Cooperation on both national and international levels;
- Cooperation with private sector;
- Multi-stakeholder governance;
- Support of economic growth;
- Mandating minimal technical safeguards;
- Reporting mechanisms;
- Education and capacity building;
- Protecting fundamental rights, freedom of expression, personal data and privacy;
- Follow-up to strategy and action plans (evidence of country ownership).

1.3.7.5 Assessment criteria for a cybercrime strategy

The developments in the UK are in many ways representative of the approach chosen by many countries and institutions listed in the section 'International Developments': the issue of cybersecurity and cybercrime is addressed in one single, holistic strategy aiming at directing government's resources and activities in an integrated policy.

Elements of a cybercrime strategy – or more precisely of a strategy on cybercrime and electronic evidence – may comprise:

Module 1. Information security. Definitions, norms, standards

- Cybercrime reporting mechanisms;
- Prevention;
- Legislation, incl. safeguards and data protection
- Specialised units;
- Interagency cooperation;
- Law enforcement training;
- Judicial training;
- Public/private cooperation;
- Effective international cooperation;
- Financial investigations and prevention and control of fraud, money laundering and terrorist financing;
- Specific measures for the protection of children online.

These elements are also largely reflected in the Strategic Priorities adopted in Kyiv meeting under the CyberCrime@EAP project in October 2013.

For the purposes of the present report, the assessment will primarily focus on:

- Public/private cooperation in particular cooperation between law enforcement authorities and Internet Services Providers (ISPs), CERTs;
- International cooperation;
- Establishment of platforms for reporting on cybercrime.

1.3.7.6 Ukrainian draft cybersecurity strategy

Ukraine has prepared a draft Cybersecurity Strategy [32]. Provision for cybersecurity is regulated by its national Constitution, laws on the Main Principles of Domestic and Foreign Policy and On the Main Principles of National Security, Ukraine's National Security Strategy, Ukraine's Information Security Doctrine and the Council of Europe Convention on Cybercrime ratified by the Law of Ukraine No 2824 of 7 September 2005.

Objectives of this draft Strategy include:

- Devising main policy directions on cybersecurity, in particular, the creation of a regulatory framework, harmonised with international standards;
- Creating an advanced, flexible national cybersecurity system for efficient cooperation of government agencies responsible for the enforcement of cybersecurity;

Module 1. Information security. Definitions, norms, standards

- Creating conditions for cooperation between public and private sectors, society and the State in countering cyber-threats and for international cooperation on cybersecurity;
- Creating conditions for the protection of national information infrastructure, primarily, objects of critical information infrastructure;
- Creating conditions for the development of a system that prepares cadres in cybersecurity sphere.

Priority measures in the first implementation phase (2015 - 2016) will focus on the development and improvement of the regulatory framework, particularly to ensure the functioning of the national cybersecurity system, the preparation of the Armed Forces of Ukraine for cyber warfare, basic preparation of cadres specialising in countering cyber-threats, and conditions for cooperation between public and private security sectors on combating cybercrime and greater attention to informing the public and businesses about cybersecurity.

The second phase (2017 - 2018) is planned to focus on the improvement of international rules of conduct in cyberspace and the international regulatory framework to address cybersecurity-related challenges to national and international security, the completion of the national cybersecurity system, the implementation of programs to support domestic innovative products to enhance cybersecurity and fostering development of the computer emergency response team network in Ukraine.

The third phase (2018 and beyond) will be adjusted on the basis of the assessment of its effectiveness and emerging challenges.

Ukraine's CERT is the CERT-UA (www.cert.gov.ua).

1.3.7.6.1 International cooperation

The draft strategy advocates international collaboration in the following areas:

- Support international initiatives in the cybersecurity sphere considering Ukraine's national interests;
- Help prevent the militarisation of cyberspace;
- Ensure Ukraine's participation in European and regional cybersecurity enforcement systems and strictly abide by Ukraine's international obligations in cybersecurity ;

Module 1. Information security. Definitions, norms, standards

- Enhance international cooperation on combating cyber-terrorism, cybercrime and cooperation on cybersecurity at the national and departmental levels;
- Contribute to international rules of government's conduct in cyberspace and improve the international legal framework to address cybersecurity-related challenges to national and international security.

1.3.7.6.2. Public/private cooperation

The strategy outlines the importance of creating conditions for cooperation between public and private sectors, society and the State in countering cyber-threats and for international cooperation on cybersecurity.

1.3.7.6.3. Multi-stakeholder governance

In order to enhance cybersecurity, the strategy foresees that the State, in partnership with the private sector, citizens and civil society, must take part in the creation and implementation of the national strategy. Furthermore, a key element for its enforcement lies in the coordination of public authorities, institutions, private sector, research institutions, professional associations and non-governmental organisations in the cybersecurity sphere.

1.3.7.6.4. Support of economic growth

The Strategy establishes as a priority the creation of economic preconditions for the development and enforcement of security of the national information infrastructure and its resources.

1.3.7.6.5. Mandating minimal technical safeguards

Cyber protection is defined as a set of organisational, regulatory, military, operational and technical measures with the aim of enforcing cybersecurity. The strategy should include a clause regarding strict compliance with legal provisions protecting government information resources, cryptographic and technical protection of information, including protection of personal information by the heads of bodies controlling objects of critical information infrastructure.

1.3.7.6.6. Education and capacity building

Module 1. Information security. Definitions, norms, standards

The Strategy refers to several areas regarding education and capacity building. For example, it recommends changes in academic plans and curricula of secondary and higher education institutions and in research and development plans of senior officials and management.

1.3.7.6.7. Protecting fundamental rights

The Strategy stresses the importance of safeguarding the rights and freedoms of Ukrainian citizens, including the right to privacy and freedom of communication. A draft law 'on the ensuring the cybersecurity of Ukraine' is in preparation providing for the protection of individual and societal vital interests in cyberspace and identifying the main areas for cybersecurity enforcement.

Questions for self-control.

1. Interpretation of the "Orange Book" for the configuration of network information security.

2. Providing high availability automated systems. The availability of the automated systems. Basic concepts. Fundamentals measures to ensure high availability.

3. Security policy. Definitions and basic concepts of security policy. Models of information security policy. Discretionary Security Policy. Mandated security policy. Role security policy. The value of information security policy for the enterprise.

4. Standard ISO/IEC 17799. The main concept. Functional requirements. Requirements trust security.

5. Basic concepts of software and technical level of information security. Features modern information systems for information security. Architectural security.

6. Administrative level of information security. Basic concepts. Sync application security lifecycle systems.

7. Standards and specifications in the field of information security. Basic concepts. Mechanisms security. Network security services. Administration of security

8. Administrative level of information security. Basic concepts. Sync application security lifecycle systems.

9. Basic definitions and criteria for classification of threats to information systems.

1.4 Regulatory and methodological support in the field of technical protection of information in Ukraine

1.4.1 State Special Communications Service of Ukraine: history, tasks, rights and obligations

Decree of the President of Ukraine of 07.11.2005 № 1556/2005 «On observance of human rights during the operational and technical measures" have been identified the need to establish a State Service for Special Communications and Information Protection as a central executive body with special status, defining it the main tasks of implementing the state policy on protection of state information resources in data networks, functioning state system of government communication National system of confidential communication, cryptographic and technical protection of information.

State Service established pursuant to the adopted February 23, 2006 The Law of Ukraine "On State Service for Special Communications and Information Protection of Ukraine" on the basis of the liquidated Department of Special Telecommunication Systems and Information Protection of Security Service of Ukraine.

Regulatory and legal basis of the system of information security of Ukraine as a prerequisite for the legality of its operation

It is recognized that scientific progress is impossible without large-scale implementation in public life and management activities of the State in various areas of science, technology and production of modern information technology, computer technology, information networks and telecommunications.

In modern conditions of development of the information society is actively developing the information sector, which combines information, information infrastructure, including information networks, information relations between actors of the sector, consisting in the collection, creation, dissemination and use of information. Information relations figure prominently in shaping the information policy of the state in modern society as well as business and personal life of each person. This, in turn, necessitates the development and improvement of legal means of regulating social relations in the sphere of information activities. It is clear that in a democratic state of law such relations should be based on a modern

Module 1. Information security. Definitions, norms, standards regulatory framework that regulates activities in the information sphere.

Since the 90s of XX century. Ukraine is gradually overcoming the difficult path to a highly legal society based on democratic principles and transparency of information. Since independence in our country formed a new national law regulating social relations in the information sector, including ensuring information security.

Legislative rules in this area significantly affect the legal regulation of relations between society and its members and the state, between natural and legal persons. That is, at the present stage information relations act, on the one hand, the external manifestation of any relationship in life, society and citizens on the other - those foundations, which formed the law in other areas of their existence.

When creating a modern and efficient system of information security essential importance, the appropriate regulatory framework, without which it is impossible to cover all areas of society under a single legal framework, develop a national concept of state and effectively implement the policy of national security in the information sphere. This means that all of the action for the protection and realization of national interests of Ukraine in any area and at any level are primarily based on the current legislation of Ukraine, confirm the legitimacy of the system of national security. However, in a democratic society, such actions of national security must comply with national legislation and universally recognized international legal norms and be controlled by the public.

In view of the legality of the operation is one of the main requirements for the system of information security. This legitimacy should be based on a set of laws and regulations aimed at creating the necessary conditions for the protection of national interests in the information and other spheres of life.

In particular, the availability of necessary and sufficient regulatory framework and its implementation mechanisms and control allows the system to Ukraine's national security function effectively in the modern world.

Legislation should perform information security primarily three main functions:

1. Adjust the relationship between the subjects of information security, determine their rights, duties and responsibilities.
2. Regulatory ensure the actions of information security at all levels - namely, man, society and state.
3. Establish procedures applying different forces and means of

Module 1. Information security. Definitions, norms, standards information security.

The most urgent task in the field of information security today is the formation of the provisions of national law information on legal provision of information in the field relevant actors, especially state agencies that are charged with the state related functions.

Over the years of independence, Ukraine laid the legal foundations of information security in particular was worked out great array of regulations where the basic powers of the state in the information sphere. Acts of national law which regulate the activity of state bodies, organizations and citizens in the information sector, establish the authority of state bodies to ensure the information security of Ukraine.

In view of the regulatory framework for the national security of Ukraine in the sphere of information should be considered in view of the existing hierarchy of regulations.

At the highest level, we consider the norms of the Constitution of Ukraine, which reinforce the conceptual provisions of national security of Ukraine in all spheres of its existence, and the Concept of Ukraine's National Security Doctrine of Information Security of Ukraine and the Law of Ukraine "On National Security of Ukraine". These documents take into account the basic provisions of international treaties and agreements ratified by Ukraine concerning its national security.

At the second level constitutive laws consider the direction which specifies important provisions on national security in the information sphere ("The Basic Principles of Information Society in Ukraine in 2007-2015", "On Information", "On State Secrets", "On National program of Informatization", "On the Concept of the National Informatization program", "On Radio Frequency resource", "On Telecommunications", "On Protection of Information in Telecommunication Systems", "On Protection of Public Morality").

At the third level - the laws of Ukraine institutional level where fixed basic forms of state agencies in the national security in the information and other spheres of the individual, society and state (eg "On Defense of Ukraine", "On the Armed Forces of Ukraine", "On National Security of Ukraine" and "On State Service of Special Communication and Information Protection", "On Police " and "On Prosecution" and "On State of Emergency", etc.).

Module 1. Information security. Definitions, norms, standards

The structure of the regulatory framework of the national security of Ukraine in the sphere of information occupy a special place decrees and orders of the President of Ukraine, and acts (acts, decrees) of the Cabinet of Ministers of Ukraine. These regulations are illegal and issued to specify and quality meet the challenges of information security.

Ministries and departments of Ukraine within the limits specified by the law of competence and responsibility on the basis of current legislation on national security of Ukraine and in accordance with the decisions of the President of Ukraine, the Cabinet of Ministers of Ukraine to develop departmental orders, instructions, regulations aimed at implementing protection programs vital human interests, society and the state in the information sphere.

An important role in Ukraine's legislation on national security play acts of normative and prescriptive local authorities - decisions on national security (against the consequences of natural disasters, industrial accidents and disasters, epidemics, the maintenance of public order, etc.) that are binding on all enterprises, institutions and organizations, as well as officials and citizens on the territory of the same authority.

1.4.2. General Provisions for the Protection of Information in Computer Systems from Unauthorized Access (ND TPI 1.1-002-99) [6].

Approved by Department of Special Telecommunication Systems and Information Protection of Security Service of Ukraine from "28" April 1999. Number 22 as amended by Order of the Administration of State Service of 28.12.2012 number 806.

This normative document defines the methodological framework (concept) solve problems protecting information in computer systems and the creation of regulatory and methodological documents regulating the question:

- Defining requirements for protecting computer systems from unauthorized access;
- The creation of secure computer systems and protect them from unauthorized access;
- Assess the security of computer systems and their suitability to meet the challenges of the consumer.

Module 1. Information security. Definitions, norms, standards

Information resources of the state or society as a whole as well as individual organizations and individuals represent some value, have appropriate financial expression and require protection from a variety of inherently influences that could reduce the value of information resources. The influences that lead to a decrease in the value of information resources, called unfavorable. The potential adverse impact of known risk.

Protection of information processed in the Automation System (AS), is the establishment and maintenance of a viable state system of measures, both technical (engineering, software and hardware) and non-technical (legal, institutional) that can prevent or complicate the possibility of threats, and reduce potential losses. In other words, information security designed to ensure security of information processed and the AS in general, that such a state that preserves the properties given information and speakers that work it. The system specified measures to ensure data protection in the AS called complex system of information protection.

The essential problem of information security in the AS can be solved organizational measures. However, with the development of information technology trend growth needs of technical measures and remedies.

Main types of protection.

The Automated System (AS) is an organizational and technical system that combines computer system, physical environment, staff and processed information. To distinguish between the two main areas of the technical means of information in the Automated System - is the protection of AS and processed information from unauthorized access and protect information leaks from technical channels (optical, acoustic, protection from leaks of side channels and induce electromagnetic radiation).

ND TPI 1.1-002-99 [6] dedicated to issues of protection from unauthorized access and building protection against unauthorized access, functioning as a part of a computer system speakers.

The ultimate goal of all measures to protect information that are sold are information security during its processing in the AS. Data protection must be ensured at all stages of the life cycle of the AS, at all stages of information processing in all modes of operation. The life cycle of the Automated System include the development, implementation, operation and decommissioning.

The main threat information.

In analyzing the problem of protection from unauthorized access information that may circulate in the Computer System is usually

Module 1. Information security. Definitions, norms, standards considered only information objects that serve reception / information sources and information flows (pieces of information that are sent between objects) regardless of their physical characteristics carriers.

Threats of information processed in the Automated System depends on the characteristics of the Operation Computer System, the physical environment, staff and processed information. Threats may or objective nature, such as changing the conditions of the physical environment (fire, flood, etc.), or failures of the Operation Computer System, or subjective, such as human error or malicious action. Threats with subjective nature may be accidental or intentional. The attempt of threats called attack.

With all great ways of classifying threats most suitable for the analysis is the classification of threats on the result of their exposure to information that is a breach of confidentiality, integrity and availability of information.

Security policy information.

In information security policy should be understood set of laws, rules, restrictions, and recommendations and so on, the procedure of processing information and focused on protection from certain threats.

The security policy should define the Automated System resources that need protection, in particular to establish categories of information processed in the Automated System.

Information Security policy implemented by various Computer System will vary not only that they are implemented security features can protect against various types of threats, but also due to the fact that the resources of the Computer System can vary significantly.

1.4.3. Criteria for Evaluating Information Security in Computer Systems from Unauthorized Access (ND TPI 2.5-004-99) [8].

In the process of assessing the ability of computer systems to ensure the protection of processed information from unauthorized access requirements are considered two forms:

- Requirements for protection functions (security services);
- Requirements for guarantees.

Criteria security of computer systems considered as a set of functional services. Each service is a set of features to withstand a certain set of threats. Each service can include multiple levels. The higher the level of services provided by better protection against certain types of threats.

Functional criteria are divided into four groups, each of which describes

Module 1. Information security. Definitions, norms, standards the requirements for services that provide threat protection one of the four basic types.

Confidentiality.

The threats related to the unauthorized introduction of information are privacy threats. If there are requirements to limit the possibility of observing the information is appropriate services to be found in the "Criteria of Confidentiality." This section describes these services (in brackets are the symbols for each service): confidential confidentiality, administrative confidentiality, the reuse of objects, analysis of covert channels, confidentiality in the exchange (export / import).

Integrity.

The threats related to unauthorized modification of information endanger integrity. If there are requirements to limit the possibility of modifying information, the appropriate services to be found in the "Criteria of Integrity." This section describes the following services: confidential integrity, administrative integrity, and integrity pullback in the exchange.

Accessibility.

Threats related to abuse the possibility of using a computer system or processed information, endanger availability. If there are requirements to protect against denial of access or protection faults, the appropriate services to be found in the "Affordability Criteria". This section describes the following services: use of resources, resistance to failure, hot replacement, disaster recovery.

The observability.

Identification and control of the actions users control the computer system are the subject observability services and handling. If there are requirements to control the actions of users or legality of access and the ability of complex remedies to function, then appropriate services belongs to the "Criteria of the Observability " such as the following services: registration, identification and authentication, trusted channel, segregation of duties, integrity of complex remedies, self-test, in the exchange of authentication, sender authentication (no disclaimer of the authorship), authentication recipient (no disclaimer of receipt).

Criteria warranty.

The criteria to assess the presence of security services in the computer

Module 1. Information security. Definitions, norms, standards system and assess correct implementation services. The criteria include requirements for security architecture of complex remedies, development environment, consistency of development, testing complex remedies environment operation and maintenance documentation.

1.4.4 Classification of Automated Systems and Standard Functional Profiles Manufacturing Security Information from Unauthorized Access (ND TPI 2.5-005-99) [9].

Class 1: one computer per one single user complex: separate workstation that is not connected to the network.

Class 2: localized user per multicomputer complex:

- local network of computers;
- multi terminal server;
- several computers that are not connected to the network but are in the common room and perform common tasks.

Class 3: many allocated users per distributed multi multicomputer complex: the main feature - the impossibility of full control of the territory in which the Automated System.

Semantics profile.

Description Profile consists of three parts:

- Alpha-numerical identifier;
- Equal sign;
- The list of service levels, taken in braces.

Identifier (ID) includes:

- Designation of class AC (1, 2 or 3);
- Alpha part that describes the types of threats against which protection is provided (K and / or C and / or D);
- Account number and optional lettering version.

All of the Identifier separated by a period.

For example, 2.K.4 - functional profile number four, which defines the requirements for Class 2 AS intended for processing, the basic requirement for the protection of which is privacy.

Version can serve in particular to indicate the strengthening of certain services within the profile.

For example, capacity building registration will lead to a new version.

Adding some significant changes, particularly the addition of new services, or may lead to a new profile or to that profile will be related to

Module 1. Information security. Definitions, norms, standards
another class or subclass of the Automated System.

Questions for self-control.

1. Legislative level of information security of Ukraine. What is the legislative level information security and why it is important? A review of current legislation in the field of information security. Legislation on general purpose information security.

2. Security policy. Definitions and basic concepts of security policy. Models of information security policy. Discretionary Security Policy. Mandated security policy. Role security policy. The value of information security policy for the enterprise.

3. Criteria for evaluating information security in computer systems from unauthorized access (ND TPI 2.5-004-99) [8].

4. General provisions for the protection of information in computer systems from unauthorized access (ND TPI 1.1-002-99) [6].

5. Automated system classification standard functional profiles and security (ND TPI 2.5-005-99) [9].

6. The classification of automated system and standard functional profiles protection (ND TPI 2.5-005-99).

Bibliography

1. Ukraine GOST 3396.0-96. Information protection. Technical protection of information. Substantive provisions.

2. Ukraine GOST 3396.1-96. Information protection. Technical protection of information. The conduct of work.

3. Ukraine GOST 3396.2-97. Information protection. Technical protection of information. Terms and definitions.

4. The Law of Ukraine. "On protection of information in automated systems" (Verkhovna Rada, 1994, № 31, st.286) (introduced in the decree number BP 81/94-BP of 05/07/94, BD, 1994, number 31, st.287)

5. The Law of Ukraine. "On information" №-XII 2657 of 2 October 1992.

6. The regulatory document on technical protection of information ND TZI 1.1-002-99. General provisions for the protection of information in computer systems from unauthorized access.

7. The regulatory document on technical protection of information ND TZI 1.1-003-99. Terminology in the field of information security in computer systems from unauthorized access.

Module 1. Information security. Definitions, norms, standards

8. The regulatory document on technical protection of information ND TZI 2.5-004-99. Criteria for evaluating information security in computer systems from unauthorized access.

9. The regulatory document on technical protection of information ND TZI 2.5-005-99. Classification of automated systems and standard functional profiles processed information protection from unauthorized access.

10. The regulatory document on technical protection of information ND TZI 3.7-001-99. Guidance for the development of technical specifications for a comprehensive information security system in the automated system.

11. The regulatory document on technical protection of information ND TZI 1.4-001-2000 typical situation of information security service in the automated system.

12. The regulatory document on technical protection of information ND TZI 1.1-001-99. Technical information security on program-controlled public automatic telephone exchange. Substantive provisions.

13. The regulatory document on technical protection of information ND TZI 2.7-001-99. Technical information security on program-controlled public automatic telephone exchange. The order of performance.

14. The regulatory document on technical protection of information ND TZI 2.5-001-99. Technical information security on program-controlled public automatic telephone exchange. Specifications of functional protection services.

15. On the concept (Principles of State Policy) of Ukraine's national security. Resolution of the Parliament of Ukraine). Supreme Council (VVR), 1997, № 10, p. 85 (As amended by the Law № 2171-III of 21.12.2000, BD, number 9, Article 38).

16. PEMVN-95. Interim recommendations for technical protection of information leakage channels from side electromagnetic radiation and induction.

17. The regulatory document on technical protection of information ND TZI 2.5-002-99. Technical information security on program-controlled public automatic telephone exchange. Specifications safeguards.

18. The regulatory document on technical protection of information ND TZI 2.5-003-99. Technical information security on program-controlled public automatic telephone exchange. Specifications trust estimates correct implementation of protection.

19. The regulatory document on technical protection of information ND TZI 2.3-001-99. Technical information security on program-controlled public automatic telephone exchange. Methods of assessing security (basic).

Module 1. Information security. Definitions, norms, standards

20. Common Criteria for Information Technology Security Evaluation. National Institute of Standards and Technology & National Security Agency (USA), Communication Security Establishment (Canada), UK IT Security and Certification Scheme (United Kingdom), Bundesamt für Sicherheit in der Informationstechnik (Germany), Service Central de la Sécurité des Systèmes (France), National Communications Security Agency (Netherlands). Version 2.1. August 1999

21. ITU-T Recommendation X.800 Security architecture for Open Systems Interconnection for CCITT application.

22. International Standard ISO/IEC 17799. Information technology - Code of practice for information security management. First edition 2000-12-01.

23.

http://ec.europa.eu/information_society/newsroom/cf/dae/document.cfm?doc_id=1667.

24. <http://ec.europa.eu/digital-agenda/en/news/eu-cybersecurity-plan-protect-open-internet-and-online-freedom-and-opportunity-cyber-security>

25. <http://www.27000.org/>

26.

http://www.europarl.europa.eu/sides/getDoc.do?type=TA&language=EN&reference=P7-TA-2014-0244#def_1_2 (Amendment 11)

27. <http://ec.europa.eu/digital-agenda/en/news/eu-cybersecurity-plan-protect-open-internet-and-online-freedom-and-opportunity-cyber-security> (pp 15, 16)

28. https://www.enisa.europa.eu/activities/Resilience-and-CIIP/national-cyber-security-strategies-ncsss/an-evaluation-framework-for-cyber-security-strategies-1/an-evaluation-framework-for-cyber-security-strategies/at_download/fullReport

29. http://eeas.europa.eu/policies/eu-cyber-security/cybsec_comm_en.pdf

30. http://eeas.europa.eu/enp/documents/progress-reports/index_en.htm

31.

http://www.coe.int/t/dghl/cooperation/economiccrime/cybercrime/GLACY/Reports/2688_6_4_GLACY_study_Rep_Mechanisms_v5_ENG.pdf

32.

http://www.niss.gov.ua/public/File/2013_nauk_an_rozrobku/kiberstrateg.pdf
f. See Oleksandr V. Potii; Oleksandr V. Korneyko; Yrii I. Gorbenko, 'Cybersecurity in Ukraine: Problems and Perspectives', http://connections-qj.org/system/files/32.01_potii_korneyko_gorbenko.pdf?download=1 for a comprehensive summary.

Module 1. Information security. Definitions, norms, standards

33. Vain, J.; Halling, E.; Kanter, G.; Anier, A.; Pal, D. (2016). Automatic Distribution of Local Testers for Testing Distributed Systems. In: Arnicans, G.; Arnicane, V.; Borzovs, J.; Niedrite, L. (Ed.). Databases and Information Systems IX : Selected Papers from the Twelfth International Baltic Conference, DB&IS 2016 (297–310). Amsterdam: IOS Press. (Frontiers in Artificial Intelligence and Applications; 291).

34. Risk Assessment and Resilience for Critical Infrastructures. Workshop Proceedings. 25-26 April 2012. Ranco, Italy. European Commission Joint Research Centre Institute for the Protection and Security of the Citizen, Via Enrico Fermi 2749, TP 210, 21027 Ispra (VA), Italy. EUR 25398 EN. ISBN 978-92-79-25589-2 (pdf). ISSN 1831-9424 (online).

35. Peter Popov. Preliminary Interdependency Analysis (PIA): Method and Tool Support. Centre for Software Reliability, City University London, UK. Software Engineering for Resilient Systems: Third International Workshop, SERENE 2011, Geneva, Switzerland, September 2011. LNCS 6968.

36. Balasubramaniyan, S.; Srinivasan, S.; Buonopane, F.; Subathra, B.; Vain, J.; Ramaswamy, S. (2016). Design and verification of Cyber-Physical Systems using TrueTime, evolutionary optimization and UPPAAL. *Microprocessors and Microsystems*, 42, 37–48, 10.1016/j.micpro.2015.12.006.

37. Roberto Baldoni, Luca Montanari Editors. 2013 Italian Cyber Security Report - Critical Infrastructure and Other Sensitive Sectors Readiness. Università degli Studi di Roma La Sapienza. 2014. <https://www.sicurezza nazionale.gov.it/sisr.nsf/sicurezza-in-formazione/la-cyber-security-in-italia.html>

38. M. Ficco, B. Di Martino, R. Pietrantuono, S. Russo. Optimized Task Allocation on Private Cloud for Hybrid Simulation of Large-Scale Critical Systems. *Future Generation Computer Systems (FGCS)*. DOI: 10.1016/j.future.2016.01.022, Available online 13 February 2016.

MODULE 3 SOME SAFETY ISSUES OF TCP/IP NETWORKS

CONTENT SECTION

3.1	Common security issues of TCP/IP networks.....	
3.2	Classification of attacks on computer networks	
3.2.1	The nature of the impact	
3.2.1	For the purpose of exposure.....	
3.2.3	At the beginning of the implementation of the impact.....	
3.2.4	The feedback from the attacked object.....	
3.2.5	The location of the subject of the attacks relative to the attacked object	
3.2.6	The layer of the reference model ISO/OSI, which is the impact	
3.3	An overview of some typical remote attacks	
3.3.1	Network traffic Analysis – listening to the communication channel	
3.3.2	Substitution of the trusted object or subject distributed computing system	
3.3.3	The false object in the network	
3.3.4	The use of false object for the organization of a remote attack on a network	
3.3.5	Denial of Service (DoS).....	
3.4	Mechanisms for the implementation of some network attacks	
3.4.1	Sniffing (Eavesdropping)	
3.4.2	Disadvantages of using a remote search algorithms.....	
3.4.3	Port scanning.....	
3.5	Firewalls (FW),	
3.5.1	Firewall connection scheme.....	
3.5.2	Traffic filtration.....	
3.5.3	Performance of functions of intermediary.....	
3.5.4	Features of gateway shielding at various levels of model OSI	
3.6	An Intrusion Detection System (IDS)	
3.7	An Intrusion Prevention System (IPS)	
	Conclusion	
	Questions for self-control	
	Bibliography	

3.1 Common security issues of TCP/IP networks

In the early 70-ies of the last century the idea arose to combine many lived heterogeneous networks into a single network. It was necessary to create a group of standard protocols that would not depend on the type of equipment and software. In 1974 appeared the work of Vinton Cerf and Robert Kahn [1], which proposed a set of protocols, later called TCP/IP. The most important of this set was the IP Protocol (Internet Protocol), which is to ensure the promotion of data blocks (packets) from one computer to another via some route through a number of intermediate networks. TCP (Transmission Control Protocol) provides end-to-end data delivery between application processes running on network nodes.

Protocols TCP/IP are the basis for constructing intranets and the global network Internet. Despite the fact that the development of TCP/IP was funded by the U.S. Department of defense, TCP/IP does not have absolute protection and allows different types of attacks.

3.2 Classification of attacks on computer networks

The literature describes a large number of attacks in TCP/IP networks [2-10]. In [2] describes more than 80 types of attacks. Proposed various classifications such attacks. For example, in [2] proposed the following classification of attacks:

- Stealing Passwords;
- Social Engineering;
- Bugs and Backdoors;
- Authentication Failures;
- Protocol Failures;
- Information Leakage;
- Denial-of-Service;
- Botnets;
- Active Attacks.

The manual also includes the classification proposed in [8,33].

3.2.1 The nature of the impact

Passive

A passive impact on the network element (network node, the set of network nodes, a network segment) we call the effect, which has no direct effect on the operation of the facility, but may violate its security policy.

Example: listening to the channel.

Active

Active influence on the object let us call the impact of having a direct influence on his work (configuration, operability, etc.) and violate the adopted it security policy.

Example: attack "Denial of service"

3.2.2 For the purpose of impact

Breach of confidentiality of information or system resources

The interception may lead to a breach of its confidentiality (if the information is not encrypted). An example of interception can serve as a listening channel on the network. In this case there is unauthorized access to information without the possibility of its distortion.

The interception service information, Protocol messages can give information about the used networking protocols, operating systems installed on hosts, etc.

Scanning IP addresses or ports the transport layer can provide information about the resources of the system

Breach of integrity of information

The possibility of distortion of information means a full control over the information flow between system objects (for example, the attack "False ARP-server" described in the section 3.4.2), or the ability to send messages on behalf of another entity (the attack "Substitution of the trusted object", described in section 3.3.2). Thus, it is obvious that the distortion of information leads to the violation of its integrity.

Operability (availability) of the system

In this case it is assumed the attacker obtaining unauthorized access to information. Its main objective is to ensure that the operating system on the target object is out of order and for all other objects in the system access to the resources of the attacked object would be impossible. An example of such attacks is DoS – denial of service.

3.2.3 At the beginning of the implementation of the impact

Attack on request from the attacked object

In this case, the attacker expects transmissions from the potential target of attack is a type of request, which will be a condition of commencement of exposure. For example, if you attack a "False ARP-server" attacking waiting for a ARP request, which will give a false ARP-response (see section 3.4.2).

The attack upon the occurrence of the expected event on the target object

In this case, the attacker carries out a constant monitoring of operating system the remote target of an attack (the Internet has a large number of programs for remote tracking computer - AeroAdmin, NeoSpy, etc.) and when a specific event occurs the system starts the impact. As in the previous case, the initiator of the implementation of the attack itself acts as the target object.

Unconditional attack

In this case, the commencement of the attack course against the target of attack, i.e. the attack is immediate and without regard to the state of the system and the target object. An example of such attack can be listening to the communication channel, most versions of DoS. Therefore, in this case the attacker is the initiator of the beginning of the attack.

3.2.4 The feedback from the attacked object

Feedback

Remote attack is carried out in the presence of feedback from the attacked object, characterized by the fact that some of the requests submitted to the attacked object, the attackers need to answer, and, consequently, between the attacker and the target of the attack, there is feedback that allows an attacker to respond adequately to all the changes occurring on the target object. An example could be the creation of a TCP connection. Conducting such an attack is difficult (for attackers) the fact that in most cases you need to impersonate a trusted entity to receive the reply message.

Without feedback (forward attack)

Attacks of this type are usually carried out on a transfer target object, single posts, the answers to which the attacker does not need. These messages are sent either in the form of ICMP messages or messages sent with UDP Protocol. A similar attack can be called unidirectional remote attack.

3.2.5 The location of the subject of the attacks relative to the attacked object

A network segment is a part of a local network separated from the other repeater, hub, bridge or router. I.e., the set of machines to transfer data between which sufficient link layer protocol.

Introsegment attack

Subject (attacking the program or the operator who are directly involved in the impact) and the victim (host, router) are in the same segment. Only the location of one segment allows an attacker to listen (and not always).

Intersegment attack

The subject and object of attack are in different segments.

In practice, intersegment attack to carry out much more difficult than intrasegment, as the attacker cannot access the channel and there is no possibility of direct listening.

Intersegment remote attack is much more dangerous than intrasegment. This is due to the fact that in the case of intersegment attack object and attacking it directly can be at a distance of thousands of kilometers away from each other, which may significantly impede measures to repel the attack.

3.2.6 The layer of the reference model ISO/OSI, which is the impact

The physical layer. Need access to the physical channel cable for wired networks. For wireless (radio) networks, it is possible to listen to the radio waves which provide communication between the network nodes.

The data link layer. The access channel allows you to listen to the transmitted information (images and content), shaping about requests and responses in the data-link level protocols (e.g., ARP protocol).

The network layer. At this layer, the attack is implemented using service packages and network layer protocols. For example, using ICMP messages or control messages between routers.

The transport layer. At this layer, the scanning of the ports, about the formation of UDP or TCP message service protocols, which provided an exchange of UDP datagrams or TCP segments (some routing protocols, the DNS protocol etc.). UDP or TCP messages are used in many varieties of DoS attacks.

The Application layer. At this layer the interference in the work of application programs. As an example – interference with DBMS (SQL injection).

3.3 An overview of some typical remote attacks

Regardless of the network protocols, topologies of computer network infrastructure and implementation mechanisms of remote effects on network nodes is invariant with respect to the peculiarities of a particular system.

This is because computer networks are designed based on the same principles, and therefore have almost the same security problems.

Typical remote attack is remote destructive impact of the information, software implemented via communication channels and are characteristic of any distributed computing system [8].

3.3.1 Network traffic analysis – listening to the communication channel

Network nodes (objects) communicate with other data sets. An attacker connecting to the communication channels of the network has the ability to intercept this data and to perform. In more detail the mechanisms of interception are described below in 3.4.1.

Analysis of network traffic allows: first, to examine the logic of network. Knowledge of the logic of network allows in practice to simulate and implement typical remote attacks.

Secondly, the interception of the data flow exchanged between network nodes, violate the confidentiality of the information exchanged between two network subscriber. Note that there is no possibility of modification of the traffic and the analysis is possible only within one network segment.

The nature of the impact analysis of network traffic is a passive effect. The implementation of this attack without feedback leads to the violation of confidentiality of information within the same network segment at the data link layer of OSI. The beginning of the attack is certainly relevant to the target of attack.

3.3.2 Substitution of the trusted object or subject distributed computing system

In the case where the distributed forces uses unstable algorithms for identification, authentication of remote objects, it is possible to model remote attack, which consists in the transmission through channels of communication messages on behalf of arbitrary object or subject network. There are two varieties of this typical remote attack:

Attack during a connection

In the process of establishing a connection (e.g. TCP connection) network entities to exchange certain information that uniquely identifies the connection. This exchange is usually called handshake.

In case of the established virtual connection attack will be assigning itself attacking the rights of the trusted entity interaction when connecting to the object of attack, allowing the attacker to conduct a session object of a distributed system on behalf of a trusted entity. To mount an attack of this

type is necessary to overcome the system of identification and authentication of messages.

Attack without an established connection

For service messages in a network is often used to transfer single messages, do not require confirmation, is not required to make the connection. For example, ICMP messages, UDP datagrams.

Attack without a set virtual connection is to transfer the service message on behalf of a network control devices, such as routers. Sending false control messages can lead to serious disruptions of distributed computing system (for example, to change its configuration).

3.3.3 The false object in the network

a) The introduction the false object in the network by imposing a false route

A route is a sequence of network nodes (routers) at which data is transmitted from source to receiver. Each router has a special table called routing table in which for each destination specified further route (the address of the next router).

The main goal of the attack is related to the imposition of a false route, is to change the initial routing to the network entity so that the new route passed through the object host by the attacker.

The implementation of this model of a remote attack is any unauthorized use of network management protocols to change the initial routing tables.

Control protocols allow you to:

- to share information between routers – the protocol messages of routing protocols (RIP, OSPF, etc.);
- to notify hosts about the new route (ICMP-messages: Redirect, Router Advertisement/Solicitation);
- remotely manage routers (special SNMP protocol – Simple Network Management Protocol).

To change the routing attacker needs to send on the network specific protocols network management of special service message on behalf of the network control devices (e.g., routers). As a result of successful changes in the route the attacker will have full control over the flow of information exchanged between two network object.

b) The introduction of the network the false object by using the shortcomings of the algorithms for remote search

In the network remote objects are often not initially have enough information needed for addressing of messages (the address of the network adapter, IP address, web server, etc.). To obtain such information, uses a

variety of algorithms for remote search, which consists in transmitting over the network a special type of search queries, and waiting for replies to the request with the required information. Examples of such queries are based on algorithms for remote search, can serve as ARP and DNS request on the Internet (more on this attack is discussed in 3.4.2).

There is a possibility:

- the attacking object is sent to intercept the query and send a response about where to point the data, the use of which will lead to addressing the attacker about the object. In the future, the entire flow of information between subject and object interaction will pass through the false object.

- periodic transmission to the target object is pre-prepared false answer, without receiving a search query; when the transfer target object of a search query about the response, the attacker will immediately be a success.

3.3.4 The use of false object for the organization of a remote attack on a network

Breeding information flow and maintaining it at about the object

In packages exchange data fields in addition there are service fields that do not pose a direct attacking of interest. To directly receive the transmitted file should be carried out on the false semantic object dynamic data flow analysis for its selection.

Modification of the information

- a) modification of transmitted data;
- b) modification of the transmitted code:
 - the introduction of destructive software;
 - change the logic of the executable file.

The substitution of information

False object not only to modify, and replace them with information intercepted. If the modification of the information leading to its partial distortion, the substitution of its a complete change.

3.3.5 Denial of Service (DoS)

Result of application of this remote attack - violation on the attacked object of operability of the relevant service of provision of remote access, that is impossibility of receiving remote access from other objects of network – denial of service! Usually subject to the attack is the server of the large company. The task of the server consists in that, permanently to expect receiving a request for connection from a remote object. In the event of such a request on the possibility to transfer a response to the request

object, which either permit the connection or not. Interaction of the server and the client is normal happens to TCP connection use. The number of possible connections is restricted to server resources (volume of a random access memory, high-speed performance, throughput of channels). The task of attacking – to exhaust the server resources [11].

If the system does not provide rules limiting the number of requests received from one object (es) per unit time, the attacker transmits the address of such one with the number of requests to the target object, which will allow traffic (directed "storm" requests).

If the facility is not provided by means of authentication of the sender address, the attacker sends to the target object is an infinite number of anonymous connection requests on behalf of other objects.

There is also the possibility for the attacker to transfer to the attacked object is incorrect, a specially selected query. In this case, the remote system may loop processing of the request, the buffer overflow with consequent system hang.

DoS attack, which is performed simultaneously with a large number of computers, called DDoS (Distributed Denial of Service). This attack is appropriate, if you want to cause a denial of service is well protected by a large company or government organization [12,13,32].

The first thing an attacker attempts to compromise a number of nodes and gets to have administrator rights. The captured units are installed Trojans. Such computers are called zombie computers. Next, the attacker sends certain commands captured by computers and those in turn realize the powerful DoS attack on the target computer.

There are also programs for voluntary participation in DDoS attacks.

An example of DDoS is shown in Fig. 3.1-3.3 [13]. Spamhaus, a spam-prevention service based in Europe, was the victim of one of the largest known cyberattacks. The attackers tried to overwhelm Spamhaus's servers using what is known as a distributed denial of service attack. This technique harnessed the power of relatively few computers to generate as much as 300 gigabits a second of traffic — an attack so large it disrupted Internet service for millions of users in Europe.

1. The attackers send commands to about 1,000 computers under their control.

2. Each computer, pretending to be Spamhaus, sends requests for information to a type of Internet server called an open resolver. An estimated 100,000 resolvers are involved in the attack.

3. The resolvers respond with a much larger message than the initial request, amplifying the size of the attack.

4. Spamhaus cannot handle the amount of traffic and ceases to respond to legitimate traffic.

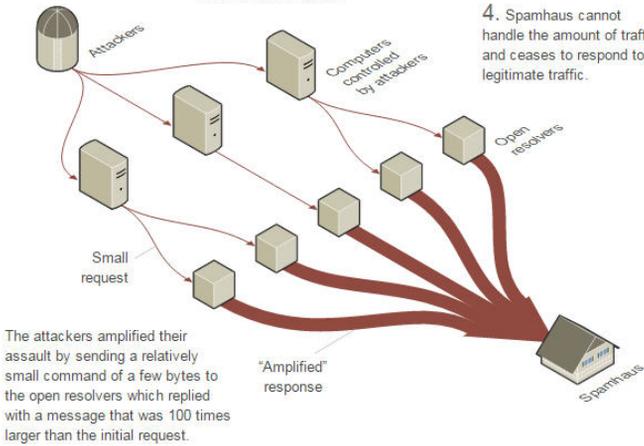


Fig. 3.1. Example of DDoS. The initial attack, reprinted from [13]

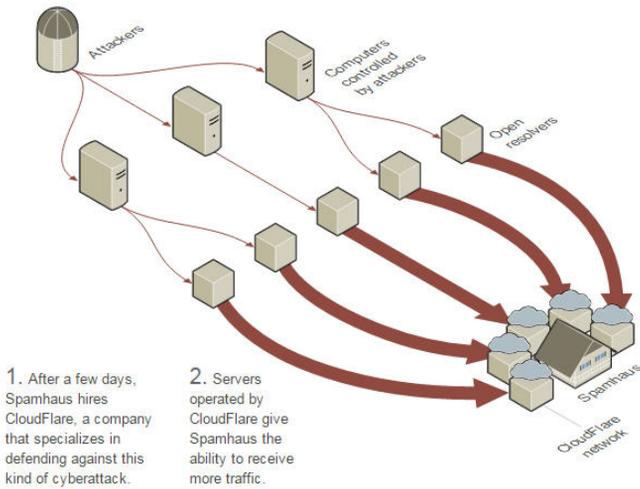


Fig. 3.2. Example of DDoS. The response, reprinted from [13]

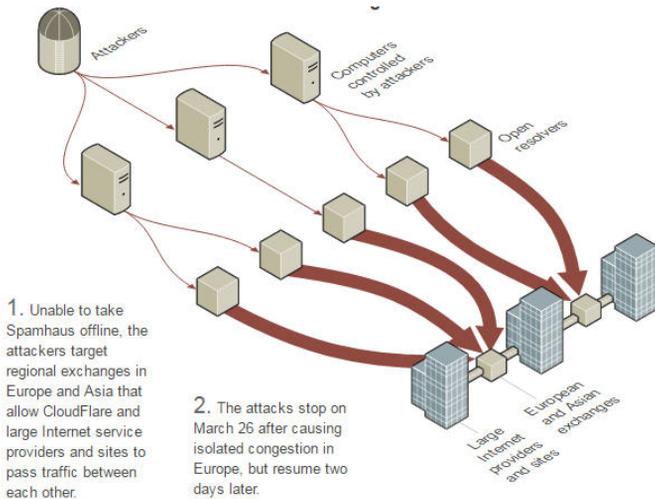


Fig. 3.3. Example of DDoS. A new target, reprinted from [13]

3.4 Mechanisms for the implementation of some network attacks

3.4.1 Sniffing (Eavesdropping)

Sniffing is the interception of packets transmitted between two computers. Interception can occur at any point of the route data. In a local network interceptor can be any node in the network, to the Internet ISP [7].

In networks based on TCP/IP, all information is transmitted mostly in plaintext (including all sorts of sensitive data - passwords and logins). So it is very advantageous to configure on a single machine software, which will browse all the packets flowing on the network and check to see if they contain any passwords. This is the sniffing. And this software called a sniffer.

Sniffer is a program that gathers traffic from the local network, useful for both attackers and network administrators

A sniffer sees only the data incoming and outgoing from the machine on which it is installed. The rest of the information flowing in the LAN, it is not available. A sniffer can capture traffic in the network segment in which it is installed, if you switch the network card to the desired mode of operation. The network card can be installed in one of the following modes:

- "*discriminating mode*" collection coming only on MAC address of the network card.

- "*promiscuous mode*" - collect all traffic that passes through network card of the computer running the sniffer.

Examples of sniffers [7]:

tcpdump is a free sniffer for a variety of UNIX platforms;

windump - free version of tcpdump for Windows XP / 7 / 10;

sniffit - free sniffer for multiple UNIX platforms;

dsniff - a free set of tools, which is based on sniffer running on UNIX;

snort - a free multi-platform sniffer.

Sniffing through hubs (passive sniffing)

A hub creates a translational environment available to all systems on the LAN (Fig. 3.4.)

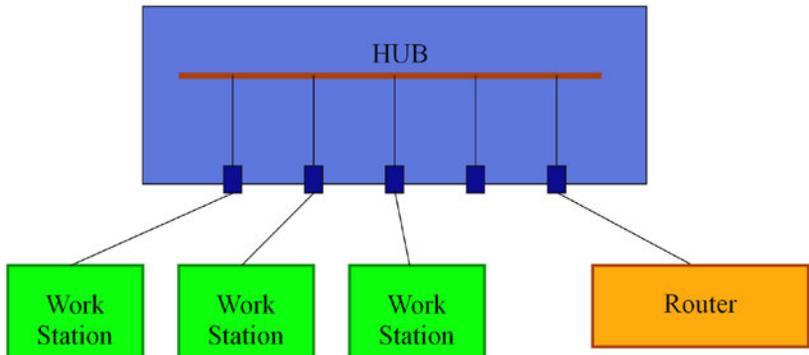


Fig. 3.4. Local network on the basis of the hub

Protection against passive sniffing

- Using the software, scans for listening (AntiSniff). The principle of antisniffing programs is to measure the response time of the hosts on the network queries and the definition, do not account for hosts to handle "extra" traffic.

- Traffic encryption [14].

Sniffing through switches (active sniffing)

The switch looks at the MAC destination address of each frame passing through it, guiding this frame only on the port connected to the host with the specified address (in Fig. 3.5 this is conventionally shown as a switching

matrix). On the attacker's host receives frames only, directed to his address (and to broadcast address). However, there are techniques that allow the attacker to listen to network traffic that uses the switches (look 3.4.2).

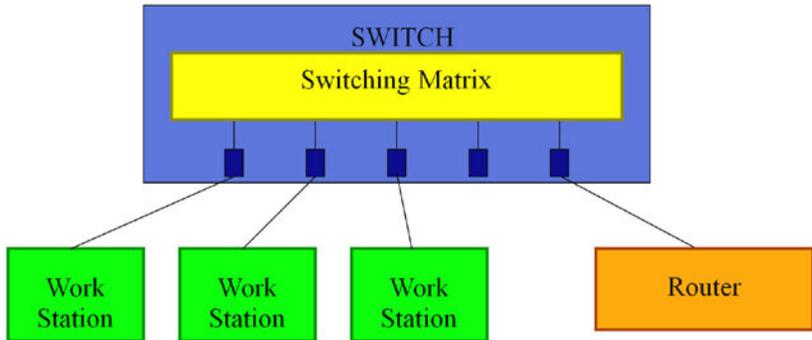


Fig.3.5. Local network on the basis of the switch

Interference switches with overload

If the input switch receives a frame with the MAC address of the sender, which is not in the switch member, the switch remembers the address (and its corresponding port where this address was received) in the buffer memory. In the future, if any port on the switch will receive the frame with this MAC destination address, switch, scanning the buffer memory will find the port that the corresponding MAC address, and send the frame to that port. The attacker uses a method based on the switch memory overflow false sender MAC addresses (for example, by using Macof from the Dsniff package). With the depletion of memory resources, some switches begin to forward data at all links in the network associated with the switch [7].

To protect against this mechanism of scanning is not recommended switch models, subject to the above disadvantage (the other switch models for exhaustion of the memory resource cease to remember the following MAC addresses.

3.4.2 Disadvantages of using a remote search algorithms

Remote lookup in the ARP Protocol

ARP (Address Resolution Protocol — link level Protocol solves the problem of conversion is known to the sender IP address to a hardware address (MAC address) [7,8,10].

Prepared to send the IP packet should be placed in the link layer frame and sent to the MAC address of that host, which matches the IP destination address (if the host in the same network where the sending host) or the MAC address of the router (if the destination host is on another network). The correlation between IP destination address and the corresponding MAC address of the sending host looks in its own ARP table. If the desired IP addresses in the ARP table does not have a MAC address the frame with an ARP request (Fig. 3.6) sends to all machines on the network (a broadcast address in the header).

Each machine of the network, accepted the ARP request, compares its own IP address with the IP address in the request. If the IP address matched the MAC address of the sender of the request sends a response containing the IP address of the responding machine and its MAC address.

Hardware protocol type	For ethernet – 0x0001
Netware protocol type	For IP – 0x0800
Hardware address length	For ethernet – 6
Netware protocol address length	For IPv4 – 4
Operation cod	For sender: request – 0x0001; reply – 0x0002
Sender hardware address	
Sender netware address	
Target hardware address	
Target netware address	

Fig. 3.6. The format of the ARP-protocol communication

Attack "false ARP-server"

An attacker who is in the network segment of the attacked, received the ARP-request, transmit over the network to the requesting host the false ARP-answer, which specifies the MAC-address of the network adapter attacking station – false ARP-server [7,8,10] (Fig. 3.7, 3.8).

The host that took about an answer, write down your ARP-table compliance with the specified IP address (e.g., IP-address of the router), MAC-address of the attacker and all packets with this IP-address will send an attack that browsing and possibly distorting these packages can transmit their router.

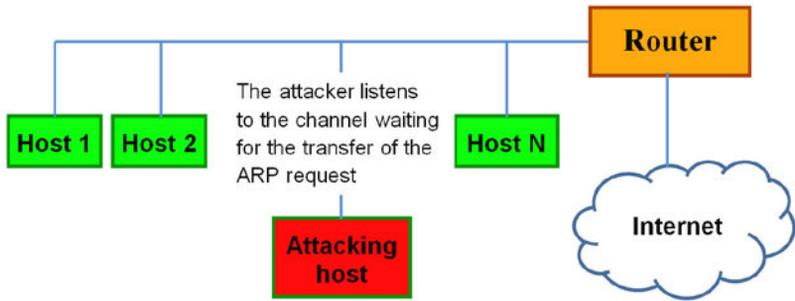


Fig. 3.7. ARP inquiry expectation phase

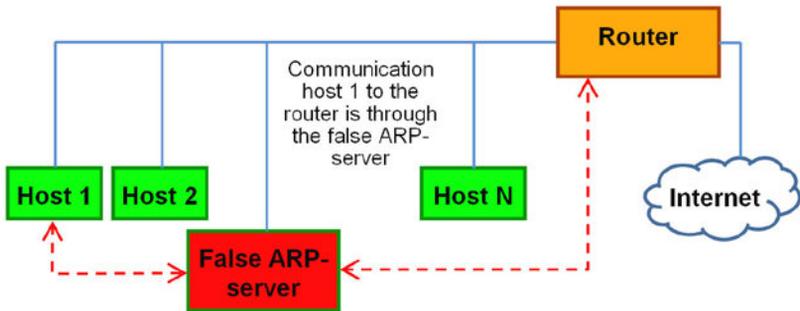


Fig. 3.8. Phase of reception, the analysis, influence and transmission of the intercepted information on the false ARP server

ARP-attack without listening

Type of attack — ARP-spoofing (Fig. 3.9) [15]. While it is possible using, for example, arbitrary ARP (gratuitous ARP) to replace the MAC-address in the ARP-tables of hosts—"victims" as a result, the packets will be sent to another device (although the IP-address will remain unchanged).

Before you perform ARP-spoofing in the ARP-table of nodes A and B, there are records with IP- and MAC-addresses of each other. The exchange of information is carried out directly between the nodes A and B (green arrow). In the course of the ARP-spoofing computer C, performing the attack, sends ARP-replies (without the query):

- node A: IP-address with node B and MAC-address node C;
- node B: IP-address with node A and MAC-address node C.

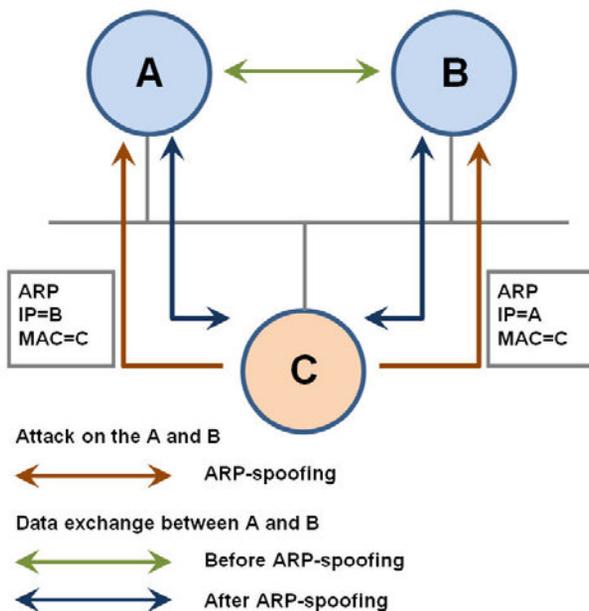


Fig. 3.9. The scheme of ARP-spoofing attack

Due to the fact that computers support spontaneous ARP (gratuitous ARP), they modify their own ARP-table and put to record, where instead of the real MAC-addresses of the computers A and B is the MAC-address of the computer C (red arrows).

After the attack is executed when the computer A wants to transfer a packet to the computer B, it finds record (it corresponds to the computer C) in the ARP-table and defines from it the MAC-address of the receiver. The packet sent on this MAC-address comes to the computer C instead of the receiver. The computer C then can relay a packet to the one to whom it is really addressed — i.e. to the computer B (blue arrows) .

Protection from false ARP-server

1) *Using Arpwatch, BitComet, AntiARP programs.* These programs monitor ARP activity on the specified interfaces. Can detect the attack of ARP-spoofing, but can't prevent it. To prevent attacks requires the intervention of a network administrator. The administrator must maintain a database of matching MAC- and IP-addresses of all nodes in the network

and use the Arpwatch program that listens on the network and notifies the administrator about the noticed violations.

2) **VLAN organization.** If the local network is divided into set of VLANs, the attack of ARP-spoofing can be applied only to computers that are in the same VLAN. The ideal situation from the point of view of security is the availability of only one computer and router interface in one VLAN. Attack of ARP-spoofing to such a segment is impossible.

3) **Use static ARP- tables.** You can avoid the attack of ARP-spoofing by setting up the ARP table manually. Then the attacker will not be able to update the ARP table by sending ARP responses for interfaces of computers.

3.4.3 Port scanning

IP - packages acting on a transport (TCP/UDP) layer of host get organized by the operating system as a great number of points to the access of different network applications of this host points [16,17]. In terms of TCP/IP such system points are named ports. On every host present ($2^{16}-1$) ports of TCP and ($2^{16}-1$) ports of UDP (port 0 - reserve). If network application is active, then the port related to this application is considered "open", at closing of network application the port related to him passes to the state "closed". Examples of standard port numbers:

TCP port 21 - File Transfer Protocol (FTP);

TCP port 23 - Telnet;

TCP port 25 - Simple Mail Transfer Protocol (SMTP);

TCP port 80 - the World Wide Web (WWW, HTTP protocol);

TCP port 666 - Doom (computer game).

A scanning is preparatory operation, secret service. Purpose – to define, what ports of host are opened, i.e. what applications are started. Since a list of active («opened») ports will be made, the phase of active actions begins.

TCP-scanning

TCP-scanning is based on the protocol of a TCP-connection, which is called "three-step handshake".

Protocol of creation of TCP-connection except for the addresses of port-source and port-recipient in heading of TCP-segment uses the special control bits of heading – flags:

URG (urgent pointer) - to use the pointer of urgency, has the special value in the field of TCP-header;

ACK (acknowledgement) is a bit of confirmation, used for a return of previous packages receipt;

PSH (push) is a function of "pushing" through, is used for more rapid migration of data on a TCP-layer;

RST (reset) - digging up because of arising up error;

SYN (synchronize) is synchronization of numbers of sequence, used for establishment of session of connection;

FIN - digging up connections, if no more data act from a sender.

Three-step handshake protocol (his elements are represented on a Fig. 3.10) [7] will be realized the exchange of three service segments of TCP between a client and server. Only specific flags and numbers in the header of these segments at the figure shows

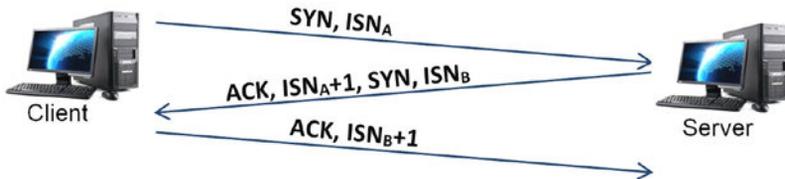


Fig. 3.10. Three-step handshake protocol

Protocol allows to define the initial numbers of sequence of bits of ISN_A , ISN_B (in the flow of data which client and server will be exchanged in a TCP-session). At a scan-out these numbers are not analyzed, analyzed usually only flags and possible ICMP-answers.

Types of scans [7]

"Polite" scanning: TCP-connect

Obvious method, is based on the principles of creating TCP-connections, and consisting of serial communication on the various ports of the TCP scan object SYN requests to create a connection. If the port is open, the scan request is received the SYN response ACK; if the port is closed - the response is a RST or ICMP-message "unreachable port". In the case of open port, the attacker completes the three-step handshake (TCP-ACK) and terminates the connection by sending a TCP FIN.

Disadvantage:

- can be easily detected on a scan target, as each TCP connection is recorded in the system log;
- large enough scan time.

"Half-open" scanning: TCP-SYN

In difference from the previous method, attacking, in case of open port (reception in reply to TCP-SYN answer TCP-ACK) finishes a session (the third stage) transfer not standard TCP-ACK, and transfer TCP-RST, interrupting connection before it has been established.

Such operation in system magazine is not registered, since the third stage of the report is absent also connection is not established. But if in system in which there is a scanning, a firewall is present, such scanning also can be registered.

Second advantage TCP-SYN of scanning is its speed as connection is broken off before its installation.

Scanning TCP-FIN, Xmas Tree, TCP-Null

FIN-scanning - is established only flag FIN;

XmasTree-scanning - flags FIN, PSH and URG are established;

Null-scanning - no flags in TCP heading are established.

These three types of scanning use an imperceptible opening in TCP RFC to divide ports on opened and closed. When the system meeting the requirements RFC is scanned, any package which is not containing the established bit SYN, RST or ACK, will cause sending TCP-RST in the answer in case the port is closed, or will not entail any answer if the port is opened.

Key feature of these types of scanning is their ability imperceptibly to bypass some package filters.

This method does not work for Windows systems which do not follow specifications RFC.

Scanning TCP-ACK

This type of scanning differs from other themes, that it is not capable to define open port. It are used for revealing of rules of package filters, and also for definition of ports filtered by them.

Often package filters are adjusted so that to forbid inquiries from external sources (such inquiries begin with TCP-SYN) to own network applications. I.e. the package filter will block TCP-segments with SYN flag.

Thus access to internal clients to external servers is usually resolved. For access to external servers the internal client realizes the three-step handshake protocol, sending TCP-SYN and expecting TCP-SYN-ACK in the answer. As such connection by a filtration rule is usually authorized, such answer (with flag ACK) passes the network filter (Fig.3.11).

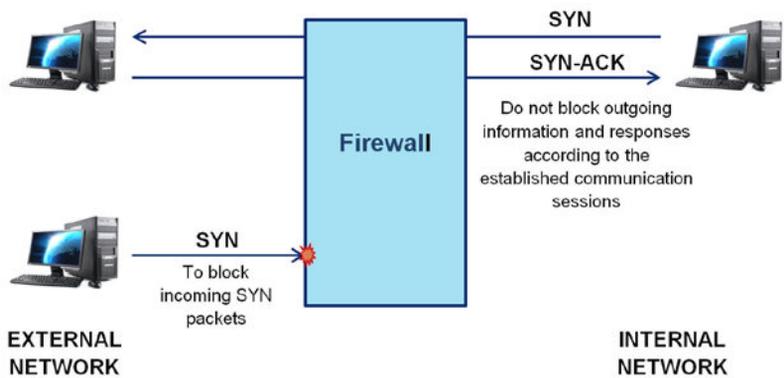


Fig. 3.11. Outgoing sessions and answers are allowed, and incoming requests for opening of connection are blocked

The inquiry segment at TCP-ACK scanning's contains established only ACK flag. At scanning of not filtered systems the open and both closed ports will return a RST segment in answer.

Ports that do not respond or send back ICMP error message is marked as filtered (Fig. 3.12).

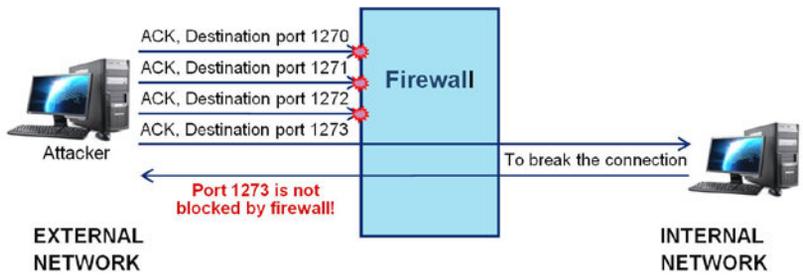


Fig. 3.12. ACK scanning

Protection against scanning of ports

Correct adjustment of a firewall (it is expedient to remember, for example, history of legal exchanges).

Use of Intrusion Detection System (IDS).

3.5 Firewalls (FW)

The firewall (shielding gateway) is called locally or functionally distributed software (software and hardware) means for realizing the control of information coming into the protected system and/or exiting the protected system [2,9,18,19].

3.5.1 Firewall connection scheme

For counteraction to not authorized gateway access the firewall should settle down between a protected network of the organization which are internal, and potentially hostile external network. Thus all interactions between these networks should be carried out only through the gateway screen. Organizational the screen is a part of a protected network (Fig. 3.13).

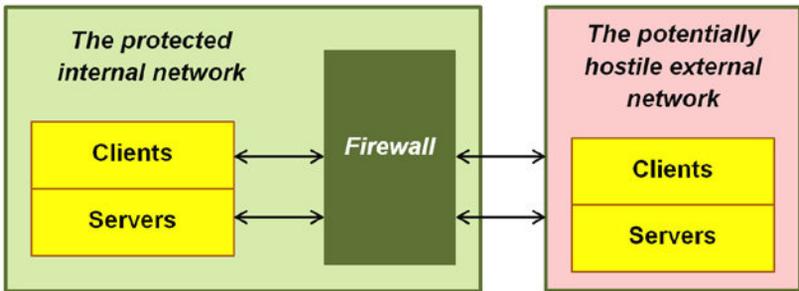


Fig. 3.13. Firewall connection scheme

Firewall can carry out two groups of functions [9,19]:

1) *The filtration of information streams passing through it.*

Filtering of information flows is that they are selectively passed through the screen, perhaps with the implementation of certain reforms and send a notice that his data in the pass denied.

2) *Intermediary at realization of gateway connections.*

The shielding gateway carries out functions of intermediary by means of the special programs named shielding agents or is simple – programs-intermediaries. The given programs are resident and forbid direct transfer of packages of messages between an external and internal network.

Complex firewall must have possibility of the analysis and use of following elements:

- Information on connections — information from all seven layers in a package.
- Histories of connections — the information received from the previous connections.
- Application layer Conditions – information on a condition, received from other applications.
- Aggregating elements — calculations of the various expressions based on all factors set forth above.

3.5.2 Traffic filtration

The filtration is carried out on the basis of a set of the rules which are preliminary loaded into the firewall and being expression of network aspects of the accepted policy of safety. Therefore it is convenient to represent the firewall as sequence of the filters processing an information stream (Fig. 3.14).

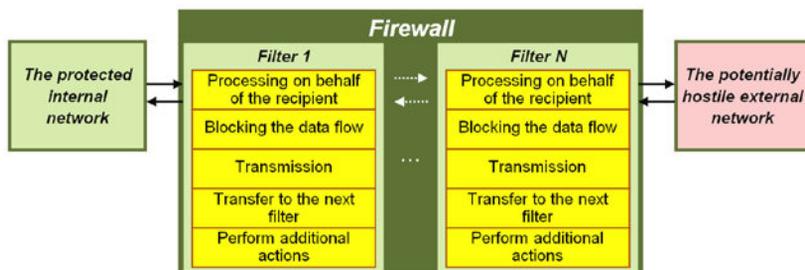


Fig. 3.14. Firewall traffic filtering

Each of filters is intended for interpretation of separate rules of a filtration by performance of following stages:

1. The information analysis by the criteria set in interpreted rules, for example, to addresses of the addressee and the sender or as the appendix for which this information is intended.

2. Acceptances on the basis of interpreted rules of one of following decisions:

- Not to pass data;
- To process data on behalf of the addressee and to return result to the sender;

- To transfer data to the following filter for analysis continuation;
- To pass data, ignoring following filters.

As criteria of the analysis of an information stream following parameters can be used:

- Service fields of packages of the messages, containing network addresses, identifiers, addresses of interfaces, numbers of ports and other significant data;
- Direct contents of packages of the messages, checked, for example, on presence of computer viruses;
- External characteristics of a stream of the information, for example, time, frequency characteristics, volume of given etc.

3.5.3 Performance of functions of intermediary

Functions of programs-intermediaries (shielding agents):

- Users identification and authentication;
- Check of authenticity of received and transferred data;
- Access differentiation to resources of an internal or external network;
- Filtration and transformation of a stream of messages, for example, dynamic search of viruses and transparent enciphering of the information;
- Translation of internal network addresses for proceeding packages of messages;
- Registration of events, reaction to set events, and also the analysis of the registered information and generation of reports;
- Caching the data requested from an external network.

3.5.4 Features of gateway shielding at various layers of model OSI

On some layers of the network model OSI presents different types of firewalls (Fig. 3.15).

Bridge shield

The given class firewalls functioning at 2-nd layer of model OSI, is known also as transparent (stealth), hidden, shadow firewalls.

Bridge filters have appeared rather recently and represent a perspective direction of development of technologies of gateway shielding. The filtration of the traffic is carried out by them at channel layer, i.e. bridge shields work with frames.

Similar FW it is possible to carry to advantages:

- There is no necessity for change of options of a corporate network, it is not required additional network configuration of firewall interfaces.

- High efficiency. As it is simple devices, they do not demand the big expenses of resources.
- The transparency. For this shield its functioning at 2-d layer of model OSI is key. It means, that its network interface has no IP-address and is invisible to world around. Attacking will not know at all, that there is firewall, checking their each package.

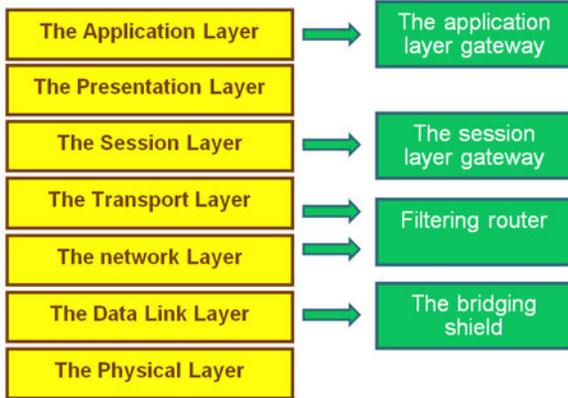


Fig. 3.15. Types of the gateway screens functioning at separate layers of the network model OSI

The shielding router (the package filter)

The shielding router (the package filter) is intended for a filtration of packages of messages and provides transparent interaction between internal and external networks (Fig.3.16) [2]. It functions at network and transport layers of reference model OSI.

The decision on that to pass or reject data, is accepted for each package independently on the basis of the set rules of a filtration.

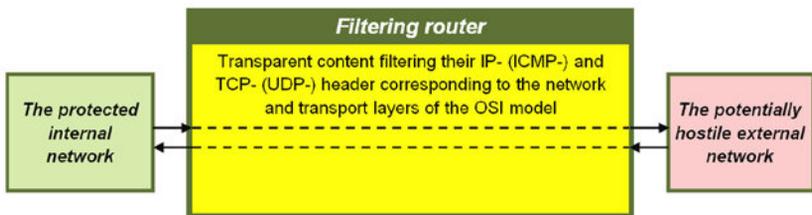


Fig. 3.16. Shielding router (the package filter)

For decision-making headings of network and transport layers are analyzed.

As analyzed fields IP- and TCP (UDP)-headings of each package act:

- The address of the sender;
- The address of the addressee;
- Package type;
- Flag of a fragmentation of a package;
- Number of port of a source;
- Number of port of the addressee.

Advantages of shielding routers concern:

- Simplicity of the shield, and also its configuration and installations procedures;
- Transparency for program applications and the minimum influence on productivity of a network;
- The low cost caused by that any router to some extent represents possibility of a filtration of packages.

Disadvantages of shielding routers:

- Do not support many necessary functions of protection, for example, authentication of ending nodes, cryptographic closing of packages of messages, and also check of their integrity and authenticity;
- Are vulnerable for such widespread network attacks, as a fake of initial addresses and not authorized change of contents of packages of messages.

The session layer gateway (shielding transport)

It is intended for the control of TCP-connections and translation IP-addresses at interaction with an external network (Fig. 3.17).

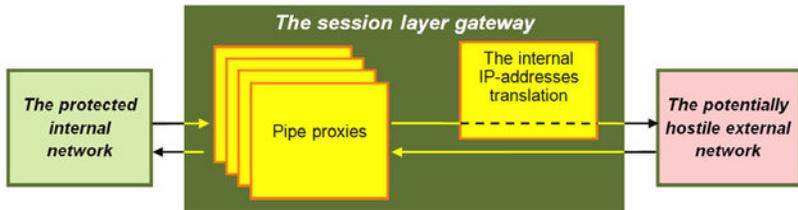


Fig. 3.17. Scheme of functioning of the gateway of session layer

For the control of virtual connections in gateways of session layer special programs which *name channel intermediaries* (pipe proxies) are used. These intermediaries establish virtual channels between internal and external networks, and then supervise transfer on these channels of the packages generated by appendices TCP/IP [21].

The gateway of the session layer supplements shielding router with functions of the control of connections and translations internal IP-addresses.

The internal IP-address translation or Network address translation (NAT) is a method of remapping one IP address space into another by modifying network address information in Internet Protocol (IP) datagram packet headers while they are in transit across a traffic routing device [].The main use of NAT is to limit the number of public IP addresses an organization or company must use, for both economy and security purposes.

IP masquerading is a technique that hides an entire IP address space, usually consisting of private IP addresses, behind a single IP address in another, usually public address space. The address that has to be hidden is changed into a single (public) IP address as "new" source address of the outgoing IP packet so it appears as originating not from the hidden host but from the routing device itself. Because of the popularity of this technique to conserve IPv4 address space, the term NAT has become virtually synonymous with IP masquerading.

The firewall sees the request from the computer with the internal IP. It then makes the same request to the Internet using its own public address, and returns the response from the Internet resource to the computer inside the private network. From the perspective of the resource on the Internet, it is sending information to the address of the firewall. From the perspective of the workstation, it appears that communication is directly with the site on the Internet. When NAT is used in this way, all users inside the private network access the Internet have the same public IP address when they use the Internet. That means only one public addresses is needed for hundreds or even thousands of users.

Using NAT in this way allows network engineers to more efficiently route internal network traffic to the same resources, and allow access to more ports, while restricting access at the firewall. It also allows detailed logging of communications between the network and the outside world.

Additionally, NAT can be used to allow selective access to the outside of the network, too. Workstations or other computers requiring special access outside the network can be assigned specific external IPs using NAT, allowing them to communicate with computers and applications that require a unique public IP address. Again, the firewall acts as the intermediary, and

can control the session in both directions, restricting port access and protocols.

NAT is a very important aspect of firewall security. It conserves the number of public addresses used within an organization, and it allows for stricter control of access to resources on both sides of the firewall.

Disadvantages of the gateway of the session level:

- The control and protection of contents of packages of messages is not provided;
- Not supported by user authentication and end nodes, as well as other features help protect your network.

Therefore a gateway of session layer apply as addition to an application gateway.

Application layer gateway (shielding gateway)

Application layer gateway is shown in Fig. 3.18

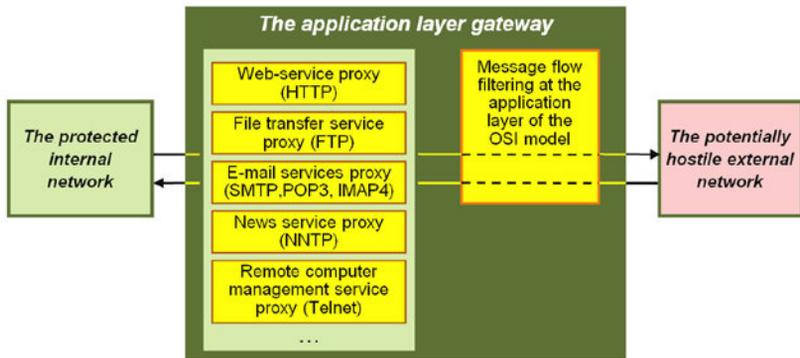


Fig. 3.18. Scheme of functioning of an application gateway

Protection functions:

- Users identification and authentication at attempt of an establishment of connections through a firewall;
- Check of authenticity of the information transferred through a gateway;
- Access differentiation to resources of internal and external networks;

- Filtration and transformation of a stream of messages, for example, dynamic search of viruses and transparent enciphering of the information;
- Registration of events, reaction to set events, and also the analysis of the registered information and generation of reports;
- Caching the data requested from an external network.

If in a network the applied gateway entering and proceeding packages can be transferred only for those services for which there are corresponding intermediaries works (Fig.3.18) [2,21].

Intermediaries of an applied gateway provide check of contents of processed packages. They can filter separate kinds of commands or the information in messages of reports of application layer which it are entrusted for serving.

At adjustment of an applied gateway and the description of rules of a filtration of messages such parameters, as are used:

- the service name;
 - admissible time range of its use;
 - restrictions on contents of the messages connected with given service;
 - computers from which it is possible to use service;
 - identifiers of users;
 - schemes of authentications;
- etc.

The application layer gateway possesses following important advantages:

- at the expense of possibility of performance of the overwhelming majority of functions of intermediary, provides the highest level of protection of a local network;
- protection at application layer allows to carry out a considerable quantity of additional checks, reducing thereby probability of carrying out of the successful attacks based on lacks of the software;
- at infringement of working capacity of the application gateway is blocked through passage of the packets between the shared networks that does not reduce the security of the protected network in case of failures.

Disadvantages of an application layer gateway:

- high cost (for example, cost of a Cisco PIXf 535 is about 50 thousand dollars);
- big enough complexity of the firewall, and also of its installation and configuration procedures;
- high demands on performance and resource consumption of computer platform;

- absence of "transparency" for users and capacity reduction at realization of gateway interactions.

3.6 An intrusion detection system (IDS)

An intrusion detection system (IDS) is a device or software application that monitors a network or systems for malicious activity or policy violations. Any detected activity or violation is typically reported either to an administrator or collected centrally using a security information and event management (SIEM) system.

An IDS differs from a firewall in that a firewalls limit access between networks to prevent intrusion and do not signal an attack from inside the network. An IDS evaluates a suspected intrusion once it has taken place and signals an alarm.

IDSes having sensors (Fig. 3.19) to detect signatures of attacks or behavioral activity to determine malicious behaviors.

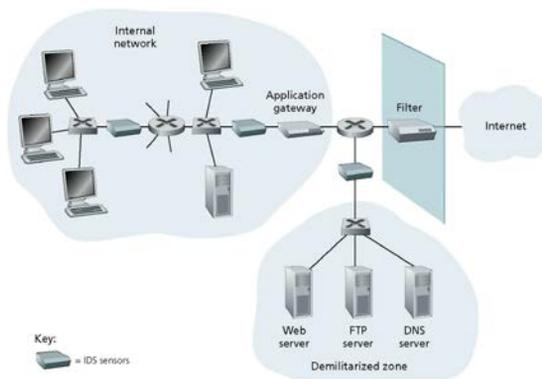


Fig. 3.19. An organization deploying a filter, an application gateway, and IDS sensors (from [20])

An IDS can be used to detect a wide range of attacks, including network mapping (emanating, for example, from nmap), port scans, TCP stack scans, DoS bandwidth-flooding attacks, worms and viruses, OS vulnerability attacks, and application vulnerability attacks.

The most common classifications are network intrusion detection systems (NIDS) and host-based intrusion detection systems (HIDS) [21,22].

1. Network-based intrusion prevention system (NIPS): monitors the entire network for suspicious traffic by analyzing protocol activity.
2. Host-based intrusion prevention system (HIPS): an installed software package which monitors a single host for suspicious activity by analyzing events occurring within that host.

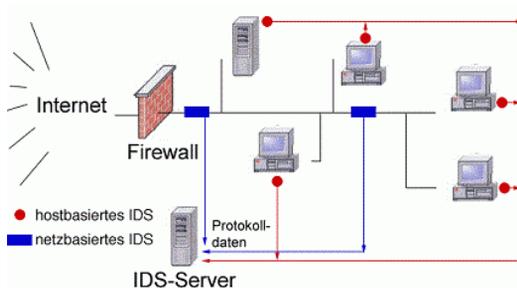


Fig. 3.20. IDS-sensors in NIPS and HIPS (from [23])

Ways to detect an intrusion [20,24]

Signature Detection:

It is also known as misuse detection, it tries to identify the events that indicate an abuse for system. It is achieved by creating models of intrusions. Incoming events are compared with the intrusion models for detection and decision. While making signature the model should detect the incoming intrusion without making any impact to regular traffic, only malicious traffic should match the model or else false alarm will be raised.

A signature-based IDS maintains an extensive database of attack signatures. Each signature is a set of rules pertaining to an intrusion activity. A signature may simply be a list of characteristics about a single packet (e.g., source and destination port numbers, protocol type, and a specific string of bits in the packet payload), or may relate to a series of packets. The signatures are normally created by skilled network security engineers who research known attacks. An organization's network administrator can customize the signatures or add its own to the database. If a packet (or series of packets) matches a signature in the database, the IDS generates an alert.

Signature-based IDS systems, although widely deployed, have a number of limitations. Most importantly, they require previous knowledge of the attack to generate an accurate signature. In other words, a signature-based IDS is

Anomaly Detection

It is termed as “not-use detection” and it differs from the signature recognition model. The model consist of a database of Anomalies. Any event that is identified with the database is called anomaly. Any deviation from the normal use is consider as Attack.

Protocol Anomaly detection

This technique based on the anomalies specific to a protocol, this model integrated with IDS recently. This identifies TCP/IP specific flaws with network. Protocols are created with specifications, know as RFCs(RFC1192) for dictating proper use and communication.

New approaches to detection

Many modern means of detection of threats on the basis of expert systems use the rules identifying the known attacks. These rules are set by the administrator, automatically created by system or use both options. Rules are used by system for removal of outputs about a protection status on the basis of these data. These expert systems need frequent up-dating because new threats permanently appear. "Rule check" systems are insufficiently flexible therefore insignificant variations of details of the attack can lead to the fact that the system won't react to the attack.

Very often malefactors bypass the set protective equipment. The attacks are carried out for very short term and a diversity of threats permanently increases that doesn't allow to find and prevent them standard protective equipment. Therefore different approaches for the solution of this problem are offered, the systems based on new approaches (a fuzzy logic, neural and neuro-fuzzy systems, etc.) are developed.

Expert systems can give to the user definite answer on a question of compliance of analyzable characteristics, and those which are stored in the database. Unlike expert systems, the system based on a fuzzy logic [25] and biological approaches (neural networks [26,27], hybrid neuro-fuzzy networks [28]) carries out information analysis and gives the chance of an assessment of data, their comparing with characteristics which she is taught to recognize

3.7 An Intrusion Prevention System (IPS)

An Intrusion Prevention System (IPS) — program or hardware system of network and computer safety. IPS finds invasions or violations of safety and automatically protects from them.

The IPS systems can be considered as extension of Intrusion Detection Systems (IDS) as the task of tracing of the attacks remains identical.

The main difference between Intrusion Prevention System (IPS) and an Intrusion Detection System (IDS) is that an IPS is implemented in-line whereas and IDS sits off to the side (Fig. 3.21).

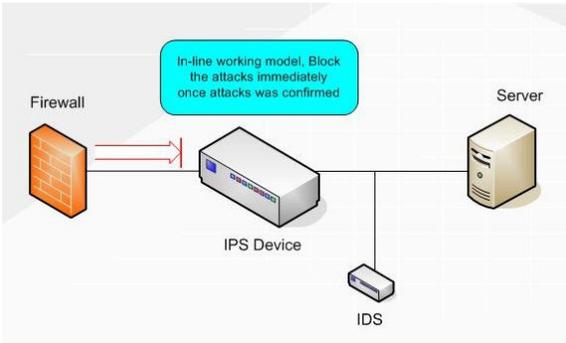


Fig. 3.21. IPS location in the protection system (from [28])

So basically all traffic is directed through the IPS, which can then block or allow the packets based on policy. It can also perform a level of correction or modification if required.

The IPS monitors the network much like the IDS but when an event occurs, it takes action based on prescribed rules. Security administrator can define such rules so the systems respond in the way they would

The obvious benefit of IPS is that it can take automated action in real time. This can be to block an attack in action or stop the malware connecting to a command and control server or with application layer IPS prevent data loss.

IPS operates on the In-line mode i.e. the sensor is placed directly in the network traffic path, inspecting all traffic at wire speed as it passes through the assigned port pair. In-line mode enables the sensor to run in a protection/prevention mode, where packet inspection is performed in real time, and intrusive packets are dealt with immediately, the sensor can drop

malicious packets. This enables it to actually prevent an attack reaching its target.

IDPSs typically record information related to observed events, notify security administrators of important observed events, and produce reports. Many IDPSs can also respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which involve the IDPS stopping the attack itself, changing the security environment (e.g., reconfiguring a firewall), or changing the attack's [29]

IPS, as well as firewall, can operate at different network levels [30]

IPS technologies use several response techniques, which can be divided into the following groups [31].

The IPS stops the attack itself: Terminate the network connection or user session that is being used for the attack

Block access to the target from the offending user account, IP address, or other attacker attribute. Block all access to the targeted host, service, application, or other resource.

The IPS changes the security environment: The IPS could change the configuration of other security controls to disrupt an attack. Common examples are reconfiguring a network device such as firewall, router, and switch to block access from the attacker.

The IPS changes the attack's content: IPS technologies can remove or replace malicious portions of an attack to make it benign.

One of the most common problems with an IDS is that it generates so many alerts that the Operations Center Security (SOC) simply cannot investigate all alerts.

Another challenge with an IPS is that because all packets go through it, the IPS also needs to be as resilient as the services that sit behind it, in a denial of service attack the IPS can be a easier target than the servers, because you can exhaust its CPU, memory etc.

Despite the problems, IDPSs have become a necessary complement to the security infrastructure of almost every organization.

Conclusion

Protocols TCP/IP are the basis for constructing intranets and the global network Internet. TCP/IP does not have absolute protection and allows different types of attacks.

The literature describes a large number of attacks in TCP/IP networks and proposed various classifications such attacks. One of the possible classifications divides attack as follows: the nature of the impact (passive, active); for the purpose of exposure (breach of confidentiality of

information or system resources, breach of integrity of information, operability (availability) of the system); at the beginning of the implementation of the impact (attack on request from the attacked object, the attack upon the occurrence of the expected event on the target object, unconditional attack); the feedback from the attacked object (feedback, without feedback); the location of the subject of the attacks relative to the attacked object (intra-segment attack, inter-segment attack); the layer of the reference model ISO/OSI, which is the impact.

An overview of some typical remote attacks. Typical remote attack is remote destructive impact of the information, software implemented via communication channels and is characteristic of any distributed computing system.

Analysis of network traffic allows: first, to examine the logic of distributed system and, secondly, the interception of the data flow exchanged between distributed objects.

In the case where the distributed system uses unstable algorithms for identification, authentication of remote objects, it is possible to model remote attack, which consists in the transmission through channels of communication messages on behalf of arbitrary object or subject network.

There is the possibility of introducing the false object in the network by imposing a false route and by using the shortcomings of the algorithms for remote search (RIP and DNS protocols). As examples of mechanisms for the implementation of certain network attacks are considered sniffing (intercept packets transmitted between two computers), disadvantages of using the remote search algorithms (remote search in the ARP protocol), Denial of Service (DoS, DDoS), a port scanning (set of type port scanning).

Considered a protection mechanism as firewalls, which can perform two groups of functions: the filtration of information streams passing through it and intermediary at realization of gateway connections.

On separate layers of the OSI network model can operate different types of gateway screen: the bridging shield, filtering router (the package filter), the session layer gateway (shielding transport), the application gateway. Each of these FW carries out its protection functions group.

Intrusion detection system (IDS) gathers and analyzes information from within a computer or network to identify unauthorized access, misuse and possible violation's. IDS also can be referred as a packet sniffer which intercepts packets travel along various communication mediums. All the packets are analyzed after they captured.

Intrusion prevention is the process of performing intrusion detection and attempting to stop detected possible incidents. Intrusion detection and prevention systems (IDPS)¹ are primarily focused on identifying possible

incidents, logging information about them, attempting to stop them, and reporting them to security administrators. In addition, organizations use IDPSs for other purposes, such as identifying problems with security policies, documenting existing threats, and deterring individuals from violating security policies. IDPSs have become a necessary addition to the security infrastructure of nearly every organization.

Questions for self-control

- 1) Classification of network attacks by the nature of impact? Examples.
- 2) Classification of network attacks by the impact purpose? Examples.
- 3) Classification of network attacks by a condition of the beginning of implementation of impact? Examples.
- 4) Classification of network attacks by existence of back coupling with the attacked object? Examples.
- 5) Classification of network attacks by layout of the subject concerning the attacked object? Examples.
- 6) Classification of network attacks by the level of an ISO reference model on which is carried out impact? Examples.
- 7) The characteristic and mechanism of implementation of the standard remote attack "A network traffic analysis".
- 8) The characteristic and the mechanism of implementation of the standard remote attack " Substitution of the trusted object or subject distributed computing system".
- 9) The characteristic and the mechanism of implementation of the standard remote attack " The introduction the false object in the network by imposing a false route".
- 10) The characteristic and the mechanism of implementation of the standard remote attack " The introduction of the network the false object by using the shortcomings of the algorithms for remote search ".
- 11) Characteristic and mechanism of implementation of standard remote attack «Denial-of-Service».
- 12) 11) Characteristic and mechanism of implementation of standard remote attack «Distributed Denial-of-Service».
- 13) Describe the attack "The false ARP server" without interception of an ARP request.
- 14) Specify methods of protection against the attack "The false ARP server".
- 15) Describe method of TCP port scanning "Polite scanning"
- 16) Describe method of TCP port scanning ""Half-open" scanning "

- 17) 15) Describe method of TCP port scanning " Scanning TCP-ACK "
- 18) Protection from port scanning?
- 19) Basic functions of firewalling?
- 20) Traffic filtering by a firewall?
- 21) Functions of mediation of a firewall?
- 22) Representation of firewalls at the different OSI levels?
- 23) The filtering routers. Assignment, advantages, disadvantages?
- 24) The session layer gateways. Assignment, advantages, disadvantages?
- 25) Assignment of channel intermediaries (within the shielding transport)?
- 26) The application layer gateways. Assignment, advantages, disadvantages?
- 27) Describe the generalized structure of an application gateway. What is the principle of its operation?
- 28) What is the difference between an intrusion detection system and a firewall?
- 29) Features of Network-based intrusion prevention system (NIPS)?
- 30) Features of Host-based intrusion prevention system (HIPS)?
- 31) Ways of detecting intrusions?
- 32) Specify the main difference between the Intrusion Prevention System (IPS) and the Intrusion Detection System (IDS).
- 33) What methods of response do IPAs use?

Bibliography

1. V.G.Cerf, R.F.Kacn. A protocol for packet network interconnection. *IEEE Transactions on Communication*. №5. 1974
2. William R. Cheswick, Steven M. Bellovin, Aviel D. *Firewalls and Internet Security: Repelling the Wily Hacker (2nd Edition)*. Addison-Wesley Professional. 2003. 464 p.
3. Christos Kalloniatis. *Security Enhanced Applications for Information Systems*. InTech. 2012. 234 p. DOI: 10.5772/2345
4. Ali Ismail Awad, Aboul Ella Hassanien, Kensuke Baba (Eds.). *Advances in Security of Information and Communication Networks*. First International Conference, SecNet 2013.Cairo, Egypt, September 3-5, 2013.Proceedings. *Communications in Computer and Information Science*. 2013. Vol. 381. 249 p.
5. *Computer security handbook. – 5th ed.* Edited by Seymour Bosworth, M.E. Kabay, Eric Whyne. Wiley, 2009. 2040 p.

6. Barry Lunt, Dale Rowe and Joseph Ekstrom. Cyber Security. In: *Security Enhanced Applications for Information Systems*. Edited by Christos Kalloniatis. InTech. 2012. 224 p. DOI: 10.5772/2345
7. Ed Skoudis, Tom Liston. *Counter Hack Reloaded, Second Edition: A Step-by-Step Guide to Computer Attacks and Effective Defenses*. Prentice Hall. 2005. 784 p.
8. Медведовский И.Д., Семьянов П.В., Леонов Д.Г. *Атака на Internet*. 2-е изд. М.: ДМК-Пресс. 2006 . 336 с. [In Russian: Medvedovsky I., Semyanov P., Leonov D. *The attack on the Internet*. 2d ed. Moscow: DMK-Press]
9. Шаньгин В.Ф. *Защита компьютерной информации. Эффективные методы и средства*. М.: ДМК Пресс. 2008. [In Russian: Shangin V. *Protection of computer information. Effective methods and tools*. Moscow.: DMK-Press.]
10. Мамаев М., Петренко С. *Технологии защиты информации в Интернете. Специальный справочник*. Спб.: Питер. 2002. 848 с. [In Russian: Мамаев М., Petrenko S. *Technologies of information protection on the Internet. Special reference book*. SPb .: Piter.]
11. William (Chuck) Easttom. *Computer Security Fundamentals, 3rd Edition*. Pearson IT Certification. 2016. 448 p.
12. CloudFlare blog, Deep Inside a DNS Amplification DDoS Attack. 30.10.2012. Available at: <https://blog.cloudflare.com/deep-inside-a-dns-amplification-ddos-attack/>
13. A. McLEAN, G. Gates, A. Tse. How the Cyberattack on Spamhaus Unfolded. 2013. Available at: http://www.nytimes.com/interactive/2013/03/30/technology/how-the-cyberattack-on-spamhaus-unfolded.html?_r=1&action=click&contentCollection=Global%20Business&module=RelatedCoverage&pgtype=article®ion=EndOfArticle&
14. Stallings W. *Cryptography and Network Security: Principles and Practic*. 6 ed. Pearson Education, Inc. 2014. 752 p.
15. Andrew Lockhart. *Network Security Hacks: Tips & Tools for Protecting Your Privacy*. O'Reilly Media, Inc. 2006. 480 p.
16. Andrew S. Tanenbaum. *Computer Networks (5th edition)*. Prentice Hall, Indian International Ed 2010. 960 p.
17. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 4-е изд. СПб.: Питер. 2010. 944 с. [In Russian: Olifer V., Olifer N. *Networks. Principles, technologies, protocols: Textbook for universities*. 4th ed. SPb .: Piter.]
18. Omar Santos. *End-to-End Network Security: Defense-in-Depth*. Cisco Press. 2007. 480 p.

19. Анин Б. *Защита компьютерной информации*. СПб.: БХВ-Петербург. 2000. 384 с. [In Russian: Anin B. *Protection of computer information*. SPb.: BHV-Piterburg]
20. Kurose, James F. *Computer networking : a top-down approach* / James F. Kurose, Keith W. Ross.—6th ed. publishing as Addison-Wesley. 2013. 862pp.
21. Karen Scarfone Peter Mell. *Guide to Intrusion Detection and Prevention Systems (IDPS). Recommendations of the National Institute of Standards and Technology*. National Institute of Standards and Technology. Gaithersburg. 2007. 127 p.
22. John R. Vacca. *Managing Information Security*. Syngress. 2010. 137 p.
23. Intrusion-Detection-System (IDS). 2016. Available at: <https://gbhackers.com/intrusion-detection-system-ids/>
23. Engin Kirda; Somesh Jha; Davide Balzarotti. Recent Advances in Intrusion Detection. In: *12th International Symposium, RAID 2009*. Saint-Malo, France, September 23–25, 2009. P. 162.
24. R. Shanmugavadivu R., Nagarajan N. Network intrusion detection system using fuzzy logic. *Indian journal of computer science and engineering*. 2011, vol. 2, No. 1, pp. 101-111. Available at: <http://www.ijcse.com/docs/IJCSE11-02-01-034.pdf>
25. Крижановский А.В. Применение искусственных нейронных сетей в системах обнаружения атак. *Доклады Томского университета систем управления и радиоэлектроники*, №2(18), часть 1., 2008. – С. 104-105. [In Russian: Krizhanovsky A. Application of artificial neural networks in systems for detecting attacks. In: *Reports of the Tomsk state university of control systems and radio electronics*.]
26. Morady M. System for intrusion Detection and Classification of Attacks. 2013. Available at: <http://research.cs.queensu.ca/~moradi/148-04-MM-MZ.pdf>
27. Слеповичев И.И. и др. Обнаружение DDoS атак нечеткой нейронной сетью. *Известия Саратовского университета. Новая серия. Серия математика. Механика. Информатика*. – 2009. №36 том 9. – С. 84-89. [In Russian: Slepovichev I. ed at. DDoS attack detection using fuzzy neural network. In: *Izv. Saratov Univ. (N.S.), Ser. Math. Mech. Inform.*] Available at: <http://cyberleninka.ru/article/n/obnaruzhenie-ddos-atak-nechetkoy-neyronnoy-setyu>
28. 4a - Rakkhi Samarasekera. How does an Intrusion Prevention System (IPS) work? 2011. Available at: <https://www.quora.com/How-does-an-Intrusion-Prevention-System-IPS-work>

29. 2a - Karen Scarfone Peter Mell. Guide to Intrusion Detection and Prevention Systems (IDPS). Computer Security Division Information Technology Laboratory National Institute of Standards and Technology Gaithersburg. 2007. Available at: <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-94.pdf>
30. 1a - Neil Desai. Intrusion Prevention Systems: the Next Step in the Evolution of IDS. 2003. Available at: <https://www.symantec.com/connect/articles/intrusion-prevention-systems-next-step-evolution-ids>
31. What is IPS and how Intrusion Prevention System Works. Available at: <http://www.aboutonlinetips.com/what-is-ips-and-how-intrusion-prevention-system-works/>
32. Shahedeh Khani, Cristina Gacek, Peter Popov. *Security-aware selection of Web Services for Reliable Composition*. 2015. Available at: https://www.researchgate.net/publication/282691779_Security-aware_selection_of_Web_Services_for_Reliable_Composition
33. Oleksandr Netkachov, Peter Popov, Kizito Salako. *Quantification of the Impact of Cyber Attack In Critical Infrastructures*. A. Bondavalli et al. (Eds.): SAFECOMP 2014 Workshops, LNCS 8696. Springer International Publishing Switzerland 2014. pp. 316–327.

MODULE 4 VIRTUAL PRIVATE NETWORKS

CONTENT SECTION

4.1	The general principles of construction of the protected communication channels
4.2	Protection at network level. IPSecurity Protocol.....
4.2.1	Architecture IPsec
4.2.2	IPsec Security Association (SA)
4.2.3	IPsec operation modes
4.2.4	AH protocol
4.2.5	Protocol of encoding (Encapsulation Security Payload – ESP).....
4.2.6	SAD (Security Associated Database) and SPD (Security Policy Database) databases.....
4.3	Protection at the transport layer. SSL and TLS protocols. OpenVPN.....
4.3.1	SSL assignment
4.3.2	SSL architecture.....
4.3.3	SSL Record Protocol
4.3.5	SSL Alert Protocol.....
4.3.5	SSL Handshake Protocol
4.3.6	OpenVPN Protocol
4.4	Protection at the data link layer. PPTP and L2TP protocols
4.4.1	PPTP protocol.....
4.4.2	L2TP protocol.....
	Conclusion
	Questions for self-control
	Bibliography

4.1 The general principles of construction of the protected communication channels

Association of the local networks and separate computers through an open environment of an information transfer in the uniform virtual network providing safety of circulating data, name *the virtual protected network* (VPN) [1–6].

Protection of the information in the course of transfer on open communication channels is based on performance of following functions:

- 1) authentication the co-operating parties;
- 2) cryptographic closing of transferred data;
- 3) acknowledgement of authenticity and integrity of transferred data;
- 4) protection against repetition, a delay and removal of messages;
- 5) protection against negation of the facts of departure and reception of messages.

The protected channel can be constructed by means of the system means realized at different layers of model OSI (see Fig. 4.1). If for protection of data the report of one of top levels (application, presentation or session) such way of protection does not depend on is used what networks (IP or IPX, Ethernet or ATM) are applied to transportation of data. On the other hand, the application thus becomes dependent on the concrete report of protection, i.e. for application such report is not transparent.

A secure channel at the highest, application layer inherent another drawback is the limited scope. The protocol protects only a specific network service.

7 - The Application Layer	Affect the applications do not depend on network technology	S/MIME
6 - The presentation layer		SSL (TLS)
5 - The session layer		
4 - The transport layer		
3 - The network layer	Transparent to the application, depend on the network technology	IPSec
2- The data link layer		PPTP
1 - The physical Layer		

Fig. 4.1. VPN at different layers of model OSI

The Secure Socket Layer (SSL) [1,2,7,8] protocol and its new open implementation Transport Layer Security (TLS) became the most known protocol of a secure channel working at the following, presentation layer.

Lowering of level of the protocol turns it into much more universal remedy of protection. Now any applications and any application layer protocols can use the uniform protocol of protection.

The below in a stack means of a secure channel are realized, the it is simpler to make them the transparent for applications and application protocols. At the network and channel levels dependence of applications on protocols of protection disappears absolutely. However here we face other problem — dependence of the protocol of protection against specific network technology.

The IPSec protocol working at the network layer is compromise option. On the one hand, it is transparent for applications, and with another — it can work practically on all networks as it is based on the widespread IP protocol.

4.2 Protection at network level. IPSecurity Protocol

4.2.1 IPSec Architecture

IPSecurity (IPSec) - is a set of protocols (see Fig. 4.2) related issues encryption, authentication, and protection when transporting IP-packets [1,2,6,7,10].

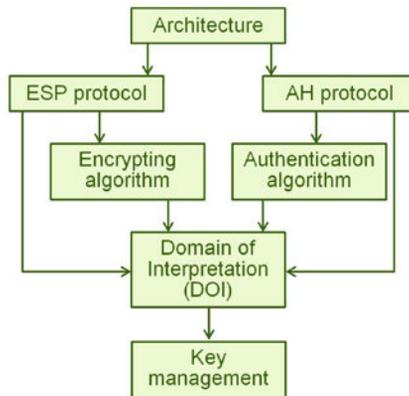


Fig. 4.2. IPSec Architecture

A basic purpose of IPSec protocols — maintenance of safe data transmission on networks IP. The use of IPSec ensures:

- integrity i.e. that data by transmission weren't distorted, lost or duplicated;
- authenticity i.e. that data were transferred by that sender who proved that he that for whom gives itself;
- confidentiality i.e. that data are transferred in the form preventing their unauthorized viewing.

Three protocols are the center of IPSec: authentication protocol (Authentication Header, AH), protocol of encoding (Encapsulation Security Payload, ESP) and keys exchange protocol (Internet Key Exchange, IKE). Functions on maintenance of a secure channel are distributed between these protocols as follows:

- the AH protocol guarantees integrity and authenticity of data;
- the ESP protocol ciphers transmitted data, guaranteeing confidentiality, but it can support authentication and integrity of data also;
- the IKE protocol solves the auxiliary problem of automatic provision to ending points of the channel of the secret keys necessary for operation of authentication protocols and data encryption.

4.2.2 IPSec Security Association (SA)

In order that the AH and ESP protocols could perform the work on protection of transmitted data, the IKE protocol sets logical connection between two ending points which in standards of IPSec wears the name "Secure Association" (SA) [1]. Secure Association – a set of the rules, procedures, parameters applied to support of service of protection of a transport flow. The SA parameters (Fig. 4.3) define what of two protocols, AH or ESP is applied to data protection, what functions fulfills the protocol of protection. Very important parameter of safe association is cryptography material, i.e. the secret keys used in operation of the AH and ESP protocols.

The formatting method defines how headers are created and what part of these headers and data of the user will be protected in data transfer process.

SPI (Security Parameter Index) – the SA identifier. It defines how the receiving side will process the arriving data stream.

Secure association parameters shall suit both ending points of a secure channel. Secure association (SA) represents the unidirectional (simplex) logical connection therefore in case of a double-sided data interchange it is necessary to set two IPSec SA.



Fig. 4.3. Between two ending points of VPN logical connection which in standards of IPSec wears the name "Secure Association" is set

4.2.3 IPSec operation modes

For execution of the tasks on support of secure data transfer the AH and ESP protocols include the additional control footing in the packets processed by them, making out it in the form of headers. The AH and ESP protocols can protect data in two modes: transport and tunnel. In the transport mode transmission of an IP packet through a network is executed by means of original header of this packet (Fig. 4.4).

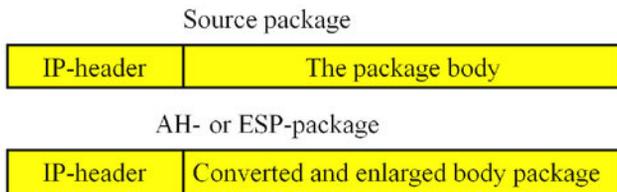


Fig. 4.4. Conversion of an IP packet in the IPSec transport mode

In the tunnel mode (Fig. 4.5) the initial packet is located in a new IP packet and data transfer on a network is executed based on header of a new IP packet.

Application of this or that mode depends on requirements imposed to data protection and also on a role which is played in networks by the node finishing a secure channel. So, the node can be a host (a finite node) or the gateway (the intermediate node). Respectively, there are three diagrams of IPSec application: "host-host", "gateway-gateway" and "host-gateway"

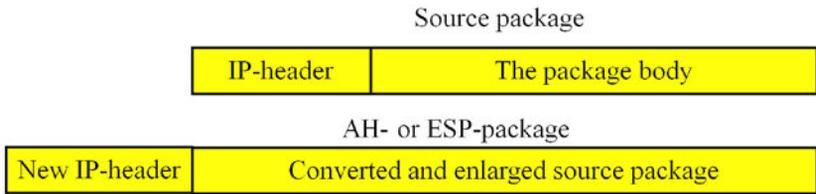


Fig. 4.5. Conversion of an IP packet in the IPSec tunnel mode

Case 1 (Fig. 4.6)

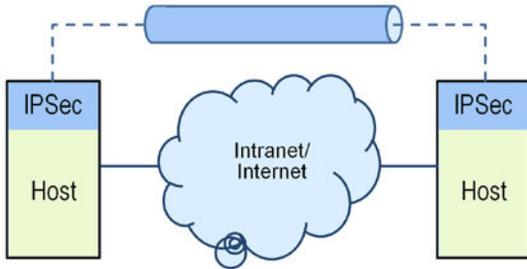


Fig. 4.6. The scheme of IPSec application "host-host"

In such scheme:

- Through protection is provided;
- Internet has no concept about SA and does not participate in it.

The transport mode of protection though it is allowed also tunnel mode is used.

Case 2 (Fig. 4.7)

In such scheme hosts are released from care concerning application SA. It is supposed, that hosts are connected to safety gateways (IP-compatible routers) safe connections.

The tunnel mode of protection is used.

The combination of cases 1 and 2 (Fig. 4.8) is possible. Such combined use of two SA allows to protect reliably the traffic and in an internal network.

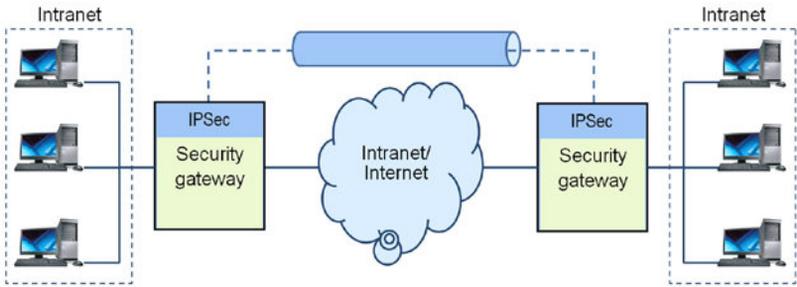


Fig. 4.7. The scheme of IPSec application "gateway-gateway"

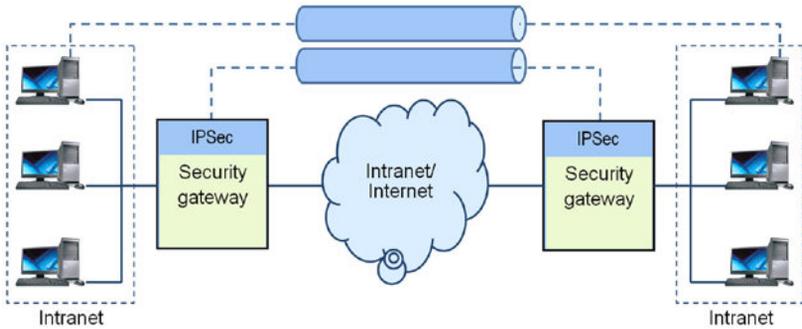


Fig. 4.8. A combination of a case 1 and a case 2

Case 3 (Fig. 4.9)

The secure channel will be organized between a distant host at which IPSec, and the gateway protecting a traffic for all hosts entering the Intranet network of the enterprise works.

The distant host can use when sending packets to the gateway the transport or tunnel mode.

The gateway sends a packet to a host only in the tunnel mode.

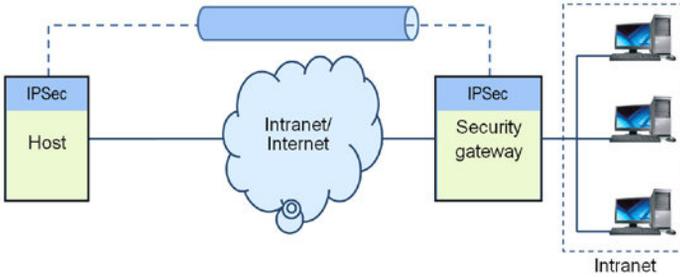


Fig. 4.9. The scheme of application IPSec "host- gateway"

4.2.4 AH Protocol

It allows the reception party to be convinced, that:

- the package has been sent by the party from which the given association is established;
- the package contents have not been deformed;
- the package is not the duplicate of some before the received package.

For performance of these functions AH protocol uses header of a following kind (see Fig. 4.10).

0	8	16	31
Next Header	Payload Length	Reserved	
Security Parameters Index (SPI)			
Sequence Number (SN)			
Authentication Data (variable length)			

Fig. 4.10. AH protocol header structure

In the field "Next Header" is underlined a code of type of the protocol of higher layer, i.e. the protocol which message is placed in the field of data of IP package.

In the field "Payload Length" contains length of AH header.

The "Security Parameters Index" (SPI) field used for communication of a package with the secure association provided for it.

The “Sequence Number” (SN) field specifies sequential number of a packet and is applied to protection against its false reproduction when the third party tries to reuse the intercepted protected packets sent by the valid authenticated sender.

“Authentication Data” field – a code of authenticity and integrity of the message – hash-function (MD5 or SHA1 or another) with a key – Message Authentication Cod (MAC)

For computation the hash-function undertakes the following information:

- fields of IP-header which do not change along the line;
- AH-header (except the data field of authentication);
- all data of the upper layer protocol.

Transport and tunnel modes of the AH protocol

Location of the AH header in a packet depends on in what mode — transport or tunnel — the secure channel is configured. The resultant packet in the transport mode looks as it is shown on Fig. 4.11.

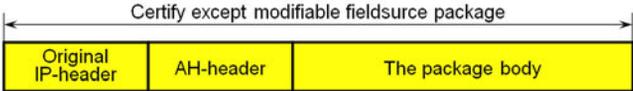


Fig. 4.11. A resultant packet of the AH protocol in the transport mode

In the tunnel mode the gateway IPsec accepts the outgoing packet going through it as transit goods and creates for it an external IP packet with new IP-header. The AH protocol in the tunnel mode protects all fields of the initial packet, and also invariable fields of an external packet new header (see Fig. 4.12).

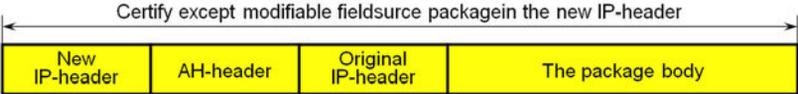


Fig. 4.12. A resultant packet of the AH protocol in the tunnel mode

4.2.5 Protocol of encoding (Encapsulation Security Payload – ESP)

The ESP protocol solves two groups of problems. The functions similar to functions of the AH protocol concern to the first of them, is a support of

authentication and integrity of data on the basis of the digest, and to the second — data protection from unauthorized viewing by encoding of transmitted data.

For the decision of the tasks the ESP protocol uses the service fields of the following format (see Fig. 4.13).

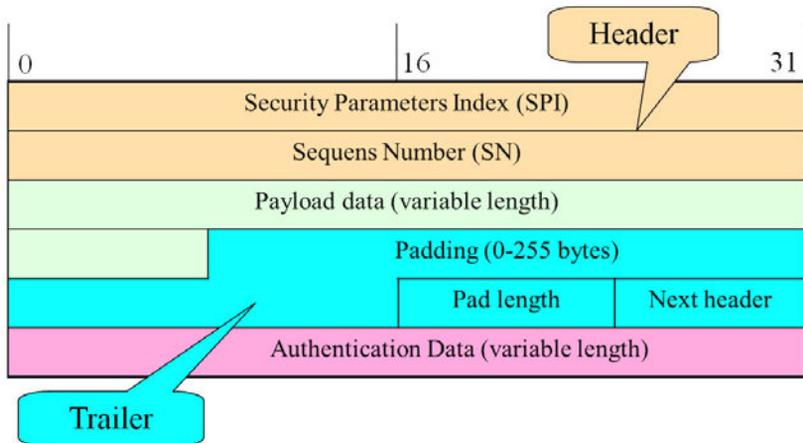


Fig. 4.13. IPsec ESP format

The service fields is divided into two parts separated by the data field (Payload Data). The first part which will be designated further as ESP title is formed by two SPI and SN fields and is placed before the data field. Remaining service fields of the ESP protocol are located at the end of a packet. Directly the data field is followed by a so-called trailer, which includes filler (Padding), length of filler (Pad Length), and also the pointer on the protocol of the following layer (Next Header). The optional field of monitoring of integrity (Authentication Data) finishes a packet.

Some service fields are similar to AH header fields: “Next Header”, “SPI”, “SN”, “Authentication Data”. But there are also two additional fields – “Padding” and “Pad Length”. Padding can be necessary in three cases. First, it is necessary for normal operation of some encryption algorithms that the ciphered text contained the multiple number of units of a certain size. Secondly, the format of title of ESP requires that the data field came to an end on boundary of four bytes. And, at last, Padding can be used for concealment of the valid packet size for the purpose of support of so-called partial confidentiality of a traffic.

Transport and tunnel modes of the ESP protocol

In Fig. 4.14 placement of header fields of ESP in the transport mode, and also its field of protection on two groups of functions is shown.

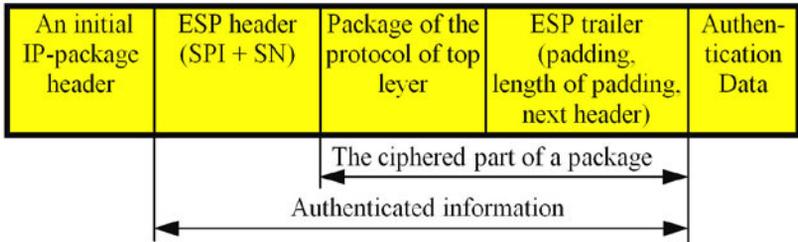


Fig. 4.14. Placing of ESP header fields in a transport mode, and also its area of protection on two groups of functions

In this mode ESP doesn't cipher IP packet header, otherwise the router won't be able to read a header fields and it is correct to realize advance of a packet between networks. Also the SPI and SN fields which shall be transferred in open form weren't among the ciphered fields in order that the arrived packet could be carried to a certain association and to be protected from false reproduction of a packet.

Unlike the AH protocol, monitoring of integrity and authenticity of data in the ESP protocol (optional function) doesn't extend to header of the initial packet, and for this reason it makes a sense to apply both protocols jointly — ESP to encoding, and AH to integrity monitoring.

In the tunnel mode the header of the initial IP packet is located after ESP header and completely is among securable fields, and the header of an external IP packet isn't protected by the ESP protocol (see Fig. 4.15).

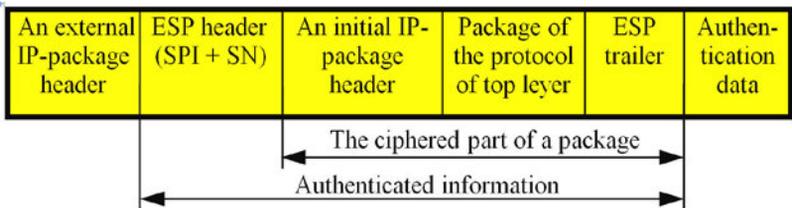


Fig. 4.15. ESP header fields placing in a tunnel mode, and also its area of protection on two groups of functions

4.2.6 SAD (Security Associated Database) and SPD (Security Policy Database) databases

To determination of a method of protection which shall will be applied to a traffic in each node supporting IPsec two types of databases are located: security associations databases (SAD), and security policy databases (SPD).

In case of establishment of logical connection, two sides accept a row of the agreements regulating process of transmission of a data stream in between. Agreements are fixed in the form of a set of parameters. For secure association such parameters are: type and an operation mode of the protocol of protection (AH or ESP), cryptography techniques, secret keys, value of the current issue of a packet in association and other information. Sets of the current parameters defining all active associations are stored on both terminal nodes of a secure channel in the form of databases of security associations SAD. Each IPsec node supports two bases of SAD — one for the proceeding associations, and another for entering.

Other type of the database — the database of a policy of security (SPD) — sets compliance between IP packets and processing rules set for them. The records SPD consist of fields of two types — fields of the selector of a packet and a fields of a policy of protection for a packet with this value of the selector (see Fig. 4.16).

The selector consists of the following feature set based on which it is possible to select type of a traffic which needs to be protected definitely with a big level of detailing:

- IP addresses of a source and assignment;
- ports of a source and assignment (i.e. TCP or UDP ports);
- types of a transport layer protocol (TCP, UDP);
- user name in the DNS or X.500 format;
- a system name (a host, the safety gateway, etc.) in the DNS or X.500 format

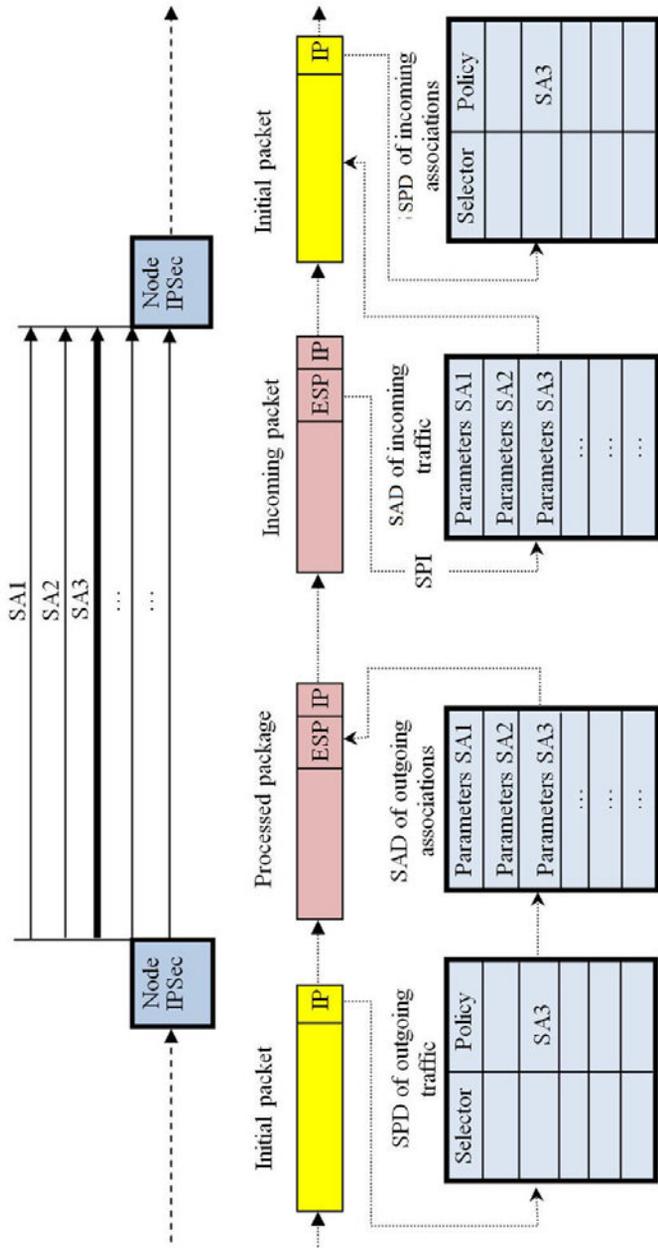


Fig. 4.1.6. SAD and SPD databases using

4.3 Protection at the transport layer. SSL and TLS protocols

In case of protection of Web the widespread decision is protection investment of funds directly over the TCP protocol. An example of the modern implementation of such approach are the SSL standard (Secure Socket Layer — the protocol of secure sockets) and its newer version — the TLS standard (Transport Layer Security — the protocol of protection of the transport layer) [1,2,8] safe data transfer in Internet. At this layer for practical implementation of this approach there are two opportunities. The most common decision is implementation of means of SSL (or TLS) in a set of the appropriate protocols that provides transparency of security features for applications. At the same time means of SSL can be built in also application programs. For example, browsers of Google and Microsoft Internet Explorer, and also the majority of Web servers have the built-in support of SSL.

4.3.1 SSL assignment

Protection of information exchange:

- confidentiality due to encoding;
- authentication (at the expense of the sign-code signature) the server and, option, the client.

According to the SSL (TLS) protocol between ending points of the virtual area network secure tunnels are created. Initiators of each protected tunnel are the client and the server functioning on computers in ending points of the tunnel.

Operation of the SSL protocol is described in terms of two important concepts — a session of SSL and the SSL connection [1].

- **Connection.** Connection of SSL is called the transport (in terms of the OSI model) providing service of some suitable type. Such connections represent equal relations between nodes to SSL. Connections are temporal. Each connection is associated only with one session. Between any couple of reported sides (for example, between HTTP-applications of the client and servers) it is possible to install a lot of the protected connections.

- **Session.** The session of SSL is a communication between the client and the server. Sessions are created by the handshake protocol SSL (SSL Handshake Protocol). The session defines a set of parameters of cryptography protection which can be used by several connections.

The SSL (TLS) protocol provides two stages of interaction of the client and the server during the forming and support of securable connection:

- SSL connection establishment;

- the protected interaction.

4.3.2 SSL architecture

The SSL protocol is designed to provide a possibility of reliable protection of open data transfer with use of the TCP protocol. Strictly speaking, SSL represents not one protocol, but two layers of protocols, as shown in Fig. 4.17.

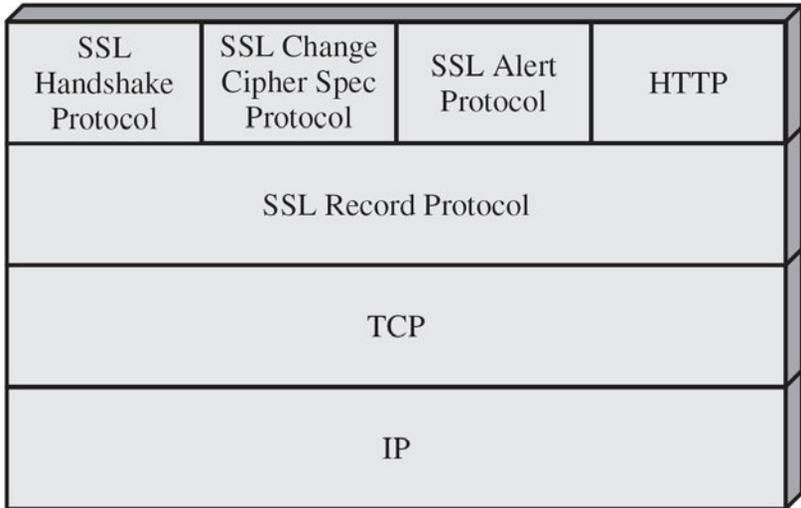


Fig. 4.17. SSL Protocols stack, reprinted from [1]

SSL Record Protocol provides a basic set of the security features applied by protocols of higher layers.

SSL Handshake Protocol is designed to create a SSL-connection. SSL Change Cipher Spec Protocol and SSL Alert Protocol are used for control of a data interchange of SSL and are considered below.

4.3.3 The SSL Record Protocol

Provide the protected interaction of the sides in the created connection and provides support of two following services:

- Confidentiality due to the symmetric encoding. The handshake protocol SSL during creation of connection defines the secret key for encoding general for the Client and the Server.
- Integrity of messages due to use of MAC.

In Fig. 4.18 the general diagram of operation of the SSL Record Protocol is shown. This protocol, having received the message for sending, at first fragments data, breaking them into units of the suitable size, if necessary executes data compression, applies MAC computation algorithm, ciphers data, adds title and transfers the created record to TCP segment.

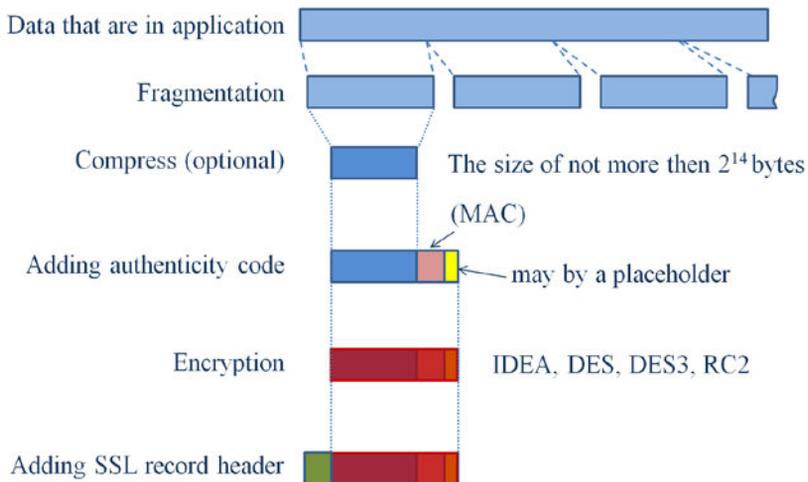


Fig. 4.18. SSL Record Protocol operation, reprinted from [1]

The first step is fragmentation. The message received from the application of higher layer is divided into units no more than 2^{14} bytes (16384 bytes). Then as an optional opportunity is applied compression. Such compression shall happen without loss and shan't increase the unit size more than for 1024 bytes. In the SSLv3 specifications (and also in the current version of TLS) by default compression algorithms aren't applied.

The following step is computation of a code of authenticity of the message (MAC value) for data. For this purpose is the common secret key serves.

Then the compressing message together with the MAC value added to it is ciphered with use of the symmetrical encoding. Encoding shan't increase

block length more than by 1024 bytes therefore the general size of the unit can't exceed $2^{14}+2048$ bytes.

When using algorithms of block encryption after MAC value it can be necessary to add padding. After bytes of padding are followed by 1-byte value specifying total length of padding.

The completing step in operation of the protocol of the record SSL is creation of the header consisting of the following fields (Fig. 4.19):

- Contents type (8 bits). Defines the protocol of the lying higher than the layer by means of which this fragment shall be processed. Use of change_cipher_spec, alert, handshake and application_data values is provided.
- Major version (8 bits). Specifies the main version number of the used SSL protocol. For SSLv3 this field contains value 3.
- Minor version (8 bits). Specifies additional version number of the applied SSL protocol. For SSLv3 this field contains value 0.
- Length of an oblate fragment (16 bits). Length in bytes of this fragment of the clear text (or an oblate fragment if compression is used). The most admissible value is equal $2^{14} + 2048$.

Content type	Major version	Minor version	Compressed length
--------------	---------------	---------------	-------------------

Fig 4.19. SSL Record Header

4.3.4 SSL Change Cipher Spec Protocol

This protocol generates a one-byte message containing a value of 1 (Fig. 4.20). The sole purpose of this message is an indication to start up the standby state parameters in the current state, which leads to a renewal of cipher suites used for this connection.



Fig. 4.20. SSL Change Cipher Spec Protocol format

4.3.5 SSL Alert Protocol

The protocol of notification (Alert Protocol) is intended for transmission to other side of the notifications concerning SSL operation participating in a data interchange.

Any message generated by this protocol consists of two bytes (Fig. 4.21). The first byte contains the value designating respectively a level of warning (1) or level of an unremovable error (2). If the level of an unremovable error is specified, the SSL protocol immediately breaks off this connection. The second byte contains the code designating a specific sense of notification.

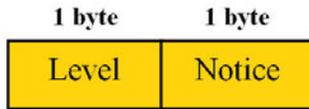


Fig. 4.21. Message of the Alert Protocol

4.3.5 SSL Handshake Protocol

This protocol allows the server and the client to execute mutual authentication, and also to agree on encryption algorithms, computation of MAC and cryptographic keys which will be applied to data protection then sent to the SSL records. The handshake protocol shall be used prior to transfer of the application programs data.

The handshake protocol shall be used prior to transfer of these application programs. At the same time several messages which exchange the client and the server are generated. All of them have the format shown in Fig. 4.22.



Fig. 4.22. Message of the Handshake Protocol

Any such message contains three following fields.

- Type (1 byte). Specifies one of 10 admissible types of the message. Admissible types of messages are given in Tab. 4.1.

Table 4.1.

SSL Handshake Protocol messages types

Message Type	Parameters
<code>hello_request</code>	null
<code>client_hello</code>	version, random, session id, cipher suite, compression method
<code>server_hello</code>	version, random, session id, cipher suite, compression method
<code>certificate</code>	chain of X.509v3 certificates
<code>server_key_exchange</code>	parameters, signature
<code>certificate_request</code>	type, authorities
<code>server_done</code>	null
<code>certificate_verify</code>	signature
<code>client_key_exchange</code>	parameters, signature
<code>finished</code>	hash value

- Length (3 bytes). Length of message in bytes.
- Contents (≥ 0 bytes). The parameters connected to the message of this type (see Tab. 4.1).

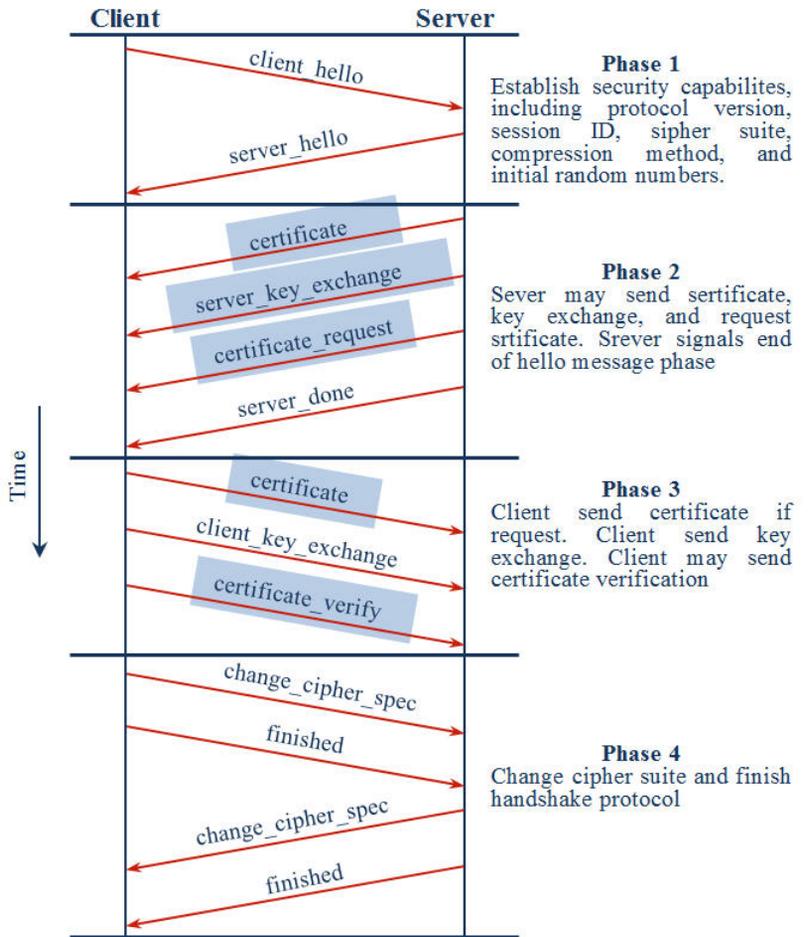
Diagram of operation of the SSL Handshake Protocol

In Fig. 4.23 the diagram of message exchange in case of installation of logical connection between the client and the server is shown. Process of exchange can be provided consisting of four main stages shown on Fig. 4.23.

After the third stage there is a formation of cryptographic keys. In the beginning (phase 2 and phase 3) value of a preliminary key (*pre_master_secret*) is created, and then both sides calculate value of the main key (*master_secret*). For transmission each other of *pre_master_secret* value the sides have two choices.

- ***RSA***. The 48-byte *pre_master_secret* key generated by the client is ciphered by means of public key of the RSA server and goes the client to the server. The server will decipher the received cipher text by means of the personal key and recovers *pre_master_secret* value.
- ***Diffie-Hellman's method***. Both the client and the server generate public keys on Diffie-Hellman's algorithm. After exchange of these

keys each side executes a certain computation by Diffie-Hellman's method as a result of which shared pre_master_secret value turns out.



Note: Shades transfers are optional or situation-dependent messages that are not always sent

Fig. 4.23. SSL Handshake Protocol Action

Co-operative main secret key represents 48-byte value (384 bits) generated for this session during the protected exchange of keys. The procedure of formation of the main secret key is shown on Fig. 4.24. For formation of the main master key except a preliminary master key prior data and one-time random numbers which the client and the server made an exchange at the first stage (phase 1) are used.

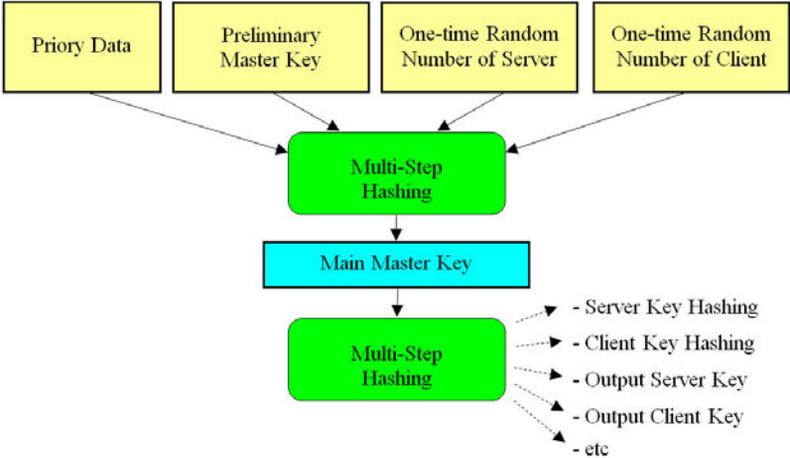


Fig. 4.24. SSL Encrypting Keys Formation

For operation of the SSL record protocol required: client MAC secret key for record, server MAC secret key for record, client secret key for encrypt, server secret key for decrypt, client initialization vector for record and server initialization vector for record. All these parameters are generated from the main master key by means of application of function of hashing to the main master key for obtaining the protected sequence of bytes of sufficient length. The procedure of generation of keys from the main master key is similar to the procedure of generation of the main master key from preliminary master key and is shown on Fig. 4.24.

4.3.6 OpenVPN Protocol

OpenVPN is a fairly new open source technology that uses the OpenSSL library andSSLv3/TLSv1 protocols, along with an amalgam of other technologies, to provide a strong and reliable VPN solution. One of its

major strengths is that it is highly configurable, and although it runs best on a UDP port, it can be set to run on any port, including TCP port 443. This makes traffic on it impossible to tell apart from traffic using standard HTTPS over SSL (as used by for example Gmail), and it is therefore extremely difficult to block.

Another advantage of OpenVPN is that the OpenSSL library used to provide encryption supports a number of cryptographic algorithms (e.g. AES, Blowfish, 3DES, CAST-128, Camellia and more), although VPN providers almost exclusively use either AES or Blowfish. 128-bit Blowfish is the default cipher built into OpenVPN

How fast OpenVPN performs depends on the level of encryption employed, although technically speaking IPSec is faster than OpenVPN because encryption/decryption is performed in the kernel, and because it allows for multi-threading, which OpenVPN does not.

OpenVPN has become the default VPN connection type, and while natively supported by no platform, is widely supported on most through third party software (including both iOS and Android).

4.4 Protection at the data link layer. PPTP and L2TP protocols

4.4.1 PPTP protocol

Point-to-Point Tunneling Protocol (PPTP) is the oldest of the protocols used in VPNs. It was originally designed as a secure extension to Point-to-Point Protocol (PPP). It adds the features of encrypting packets and authenticating users to the older PPP protocol [12].

PPTP uses a control channel over TCP and a GRE tunnel operating to encapsulate PPP packets.

Generic Routing Encapsulation (GRE) is a tunneling protocol developed by Cisco Systems that can encapsulate a wide variety of network layer protocols inside virtual point-to-point links over an Internet Protocol network.

The PPTP specification does not describe encryption or authentication features and relies on the Point-to-Point Protocol being tunneled to implement security functionality. However, the most common PPTP implementation shipping with the Microsoft Windows product families implements various levels of authentication and encryption natively as standard features of the Windows PPTP stack.

In the Microsoft implementation, the tunneled PPP traffic can be authenticated with PAP, CHAP, MS-CHAP v1/v2 .

Password Authentication Protocol (PAP) is a password-based authentication protocol used by Point to Point Protocol (PPP) to validate users. PAP is considered a weak authentication scheme. Among PAP's deficiencies is the fact that it transmits unencrypted passwords over the network. PAP is therefore used only as a last resort when the remote server does not support a stronger scheme such as CHAP or EAP.

CHAP provides protection against replay attacks by the peer through the use of an incrementally changing identifier and of a variable challenge-value. CHAP requires that both the client and server know the plaintext of the secret, although it is never sent over the network. Thus, CHAP provides better security as compared to Password Authentication Protocol (PAP) which is vulnerable for both these reasons.

CHAP is a three-way process whereby the client sends a code to the server, the server authenticates it, and then the server responds to the client. CHAP also periodically re-authenticates a remote client, even after the connection is established.

MS-CHAP is the Microsoft version of the Challenge-Handshake Authentication Protocol, CHAP. The protocol exists in two versions, MS-CHAPv1 and MS-CHAPv2. Currently, dropped support for MS-CHAPv1.

EAP-TLS is seen as the superior authentication choice for PPTP;[13] however, it requires implementation of a public-key infrastructure for both client and server certificates. As such, it may not be a viable authentication option for some remote access installations.

PPTP uses Microsoft Point-to-Point Encryption (MPPE) to encrypt packets.

MPPE uses the RSA RC4 algorithm to provide data confidentiality. The length of the session key to be used for initializing encryption tables can be negotiated. MPPE currently supports 40-bit and 128-bit session keys.

MPPE session keys are changed frequently; the exact frequency depends upon the options negotiated, but may be every packet. All keys are derived from the cleartext authentication password of the user. RC4 is stream cipher; therefore, the sizes of the encrypted and decrypted frames are the same size as the original frame.

The Point-to-Point Tunneling Protocol (PPTP) is an obsolete method for implementing virtual private networks, with many known security issues.

PPTP has been the subject of many security analyses and serious security vulnerabilities have been found in the protocol. The known vulnerabilities relate to the underlying PPP authentication protocols used, the design of the MPPE protocol as well as the integration between MPPE and PPP authentication for session key establishment [14-17].

In more detail, the PPTP protocol is described in [6,18].

4.4.2 L2TP protocol

L2TP appeared as a result of combining of the PPTP and Layer 2 Forwarding (L2F) protocols [19,20].

The principal advantage of L2TP is that this protocol allows to create the tunnel not only on the IP networks, but also in such as ATM, X.25 and a Frame Relay.

The Layer 2 Tunneling Protocol (L2TP) is a client-server protocol that allows the Point-to-Point Protocol (PPP) to be tunneled across a network. L2TP encapsulates Layer 2 packets, such as PPP, for transmission across a network. An L2TP access concentrator (LAC), configured on an access device, receives packets from a remote client and forwards them to an L2TP network server (LNS) on a remote network. The LNS functions as the logical termination point of the PPP session tunneled by the LAC from the remote client. Fig. 4.25 shows a simple L2TP topology.

In spite of the fact that L2TP works like the link protocol of the OSI model, actually it is a session layer protocol and uses the registered UDP port 1701.

L2TP applies the UDP protocol as transport and uses the identical message format both for control of the tunnel, and for transfer of data. L2TP in implementation of Microsoft uses UDP packets containing the encoded packets of PPP as control messages.

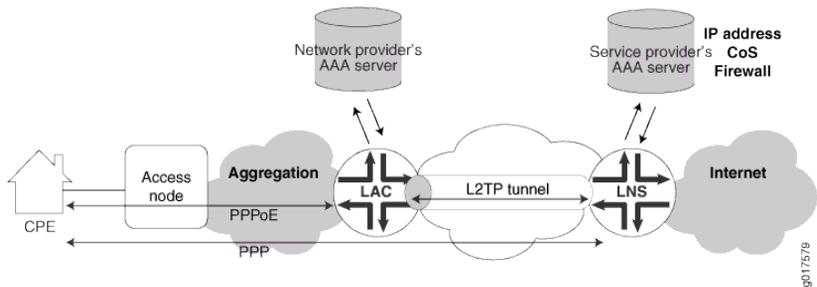


Fig 4.25. Typical L2TP Topology (from [20])

L2TP does not include any encryption capabilities on its own, so it is often combined with an encryption protocol. The most common encryption protocol used with L2TP is IPsec

As well as in a case with PPTP, L2TP begins assembly of a packet for transmission to the tunnel with the fact that PPP title, then L2TP title is added to a field of details of PPP at first. The packet received thus is encapsulated by UDP. The L2TP protocol uses UDP port 1701 as port of the sender and the receiver.

Depending on IPsec selected like a trust relationships policy, L2TP can cipher UDP messages and add to them title and the termination Encapsulating Security Payload (ESP), and also the termination IPsec Authentication. Then encapsulation in IP is made. The IP title containing source addresses and the receiver is added. In conclusion of L2TP executes the second PPP encapsulation for data preparation for transmission.

The computer receiver accepts data, processes title and the termination PPP, cleans IP title. Through IPsec Authentication of an information field of IP is carried out, and the ESP title of IPsec helps to decrypt a packet.

IPsec encryption has no major known vulnerabilities, and if properly implemented may still be secure.

L2TP/IPsec encapsulates data twice which slows things down, but this is offset by the fact that encryption/decryption occurs in the kernel and L2TP/IPsec allows multi-threading (which OpenVPN does not.) The result is that L2TP/IPsec is theoretically faster than OpenVPN.

Conclusion

Association of the local networks and separate computers through an open environment of an information transfer in the uniform virtual network providing safety of circulating data, name the virtual protected network (VPN)

The protected channel can be constructed by means of the system means realized at different layers of model OSI

A secure channel at the highest, application level inherent another drawback is the limited scope. The protocol protects only a specific network service.

The Secure Socket Layer (SSL) protocol and its new open implementation Transport Layer Security (TLS) became the most known protocol of a secure channel working at the following, presentation level.

The IPsec protocol working at the network layer is transparent for applications and can work practically on all networks as it is based on the widespread IP protocol.

Three protocols are the center of IPsec: authentication protocol (Authentication Header, AH), protocol of encoding (Encapsulation Security Payload, ESP) and keys exchange protocol (Internet Key Exchange, IKE).

In order that the AH and ESP protocols could perform the work on protection of transmitted data, the IKE protocol sets logical connection between two ending points which in standards of IPSec wears the name "Secure Association" (SA). Secure Association – a set of the rules, procedures, parameters applied to support of service of protection of a transport flow.

The AH and ESP protocols can protect data in two modes: transport and tunnel. In the transport mode transmission of an IP packet through a network is executed by means of original header of this packet. In the tunnel mode the initial packet is located in a new IP packet and data transfer on a network is executed based on header of a new IP packet.

AH Protocol allows the reception party to be convinced, that: the package has been sent by the party from which the given association is established the package contents have not been deformed; the package is not the duplicate of some before the received package.

The ESP protocol solves two groups of problems. The functions similar to functions of the AH protocol concern to the first of them, is a support of authentication and integrity of data on the basis of the digest, and to the second — data protection from unauthorized viewing by encoding of transmitted data.

In case of protection of Web the widespread decision is protection investment of funds directly over the TCP protocol. An example of the modern implementation of such approach are the SSL standard (Secure Socket Layer — the protocol of secure sockets) and its newer version — the TLS standard (Transport Layer Security — the protocol of protection of the transport layer) safe data transfer in Internet.

SSL assignment: protection of information exchange: confidentiality due to encoding; authentication (at the expense of the sign-code signature) the server and, option, the client.

The SSL (TLS) protocol provides two stages of interaction of the client and the server during the forming and support of securable connection: SSL connection establishment; the protected interaction.

SSL represents not one protocol, but two layers of protocols.

SSL Handshake Protocol is designed to create a SSL-connection. SSL Change Cipher Spec Protocol and SSL Alert Protocol are used for control of a data interchange of SSL and are considered below.

The SSL Record Protocol provide the protected interaction of the sides in the created connection and provides support of two following services: confidentiality due to the symmetric encoding. The handshake protocol SSL during creation of connection defines the secret key for encoding general for the Client and the Server; integrity of messages due to use of MAC.

OpenVPN has become the default VPN connection type, and while natively supported by no platform, is widely supported on most through third party software (including both iOS and Android).

The functional capabilities of PPTP and L2TP are various. L2TP can be used not only on IP networks, the official messages for creation of the tunnel and transfer of data on it use an identical format and protocols. PPTP can be applied only on IP networks, and it needs the separate TCP connection for creation and use of the tunnel. L2TP over IPSec offers more security levels, than PPTP, and can guarantee almost 100 percent safety important for data structure.

Features of L2TP do it by very perspective protocol for creation of the virtual area networks

Questions for self-control

- 1) Give the definition of a virtual private network (VPN).
- 2) Specify the composition of protocols in IPSec. Explain the purpose of each of them.
- 3) What is the "IPSec security Association"? What components it consists?
- 4) What features of tunnel mode IPSec? Provide a generalization of the scheme mode.
- 5) What are the features of a transport mode IPSec? Provide a generalization of the scheme mode.
- 6) Describe the use of VPN tunnels based on IPSec when connecting to remote hosts.
- 7) Describe the use of VPN tunnels based on IPSec, when connecting secure gateways.
- 8) Describe the use of VPN tunnels based on IPSec when a remote host connection to the security gateway.
- 9) Specify the purpose of the AH Protocol.
- 10) Specify the purpose of the ESP Protocol.
- 11) Explain the purpose and structure of the SAD and the SPD.
- 12) Explain the use of the SAD and SPD for the formation of outgoing traffic.
- 13) Explain the use of the SAD and SPD for processing incoming traffic.
- 14) Specify the purpose` of SSL.
- 15) Describe the architecture of SSL.
- 16) What is a "session" in the SSL?
- 17) What is a "connection" to SSL?

- 18) Describe the purpose and format of the SSL Alert Protocol.
- 19) Describe the purpose and format of the SSL Handshake Protocol.
- 20) Describe the General scheme of operation of the SSL Record Protocol.
- 21) What features of the OpenVPN protocol?
- 22) Assignment of the PPTP protocol?
- 23) Mechanisms of protection of the PPTP protocol?
- 24) What features of the L2TP protocol?
- 25) Mechanisms of protection of the PPTP protocol?

Bibliography

1. Stallings W. *Cryptography and Network Security: Principles and Practic. 6 ed.* Pearson Education, Inc. 2014. 752 p.
2. Stallings W., Lawrie B. *Computer security: principles and practice, 2 ed.* Pearson Education, Inc. 2011. 816 p.
3. Omar Santos. *End-to-End Network Security: Defense-in-Depth.* Cisco Press. 2007. 480 p.
4. Dwayne Williams, Roger Davis, Chuck Cothren, Et al. *Principles of Computer Security, Fourth Edition.* McGraw-Hill. 2016. 768 p.
5. Шаньгин В.Ф. *Защита компьютерной информации. Эффективные методы и средства.* М.: ДМК Пресс. 2008. [In Russian: Shangin V. *Protection of computer information. Effective methods and tools.* Moskow.: DMK-Press.]
6. Uyless Black. *Intrnet Security Protocols. Protection IP Traffic.* Prentice Hall PTR. 2000. 304 p.
7. William Stallings. *Network security Essentials: Applications and standards. Fourth edition.* Pearson Education, Inc. 2011. 432 p.
8. *Free eBook: Beginners Guide to Digital SSL Certificates.* 2011. Available at: <http://www.onlineprogrammingbooks.com/free-ebook-beginners-guide-to-digital-ssl-certificates/>
9. Graham Bartlett, Amjad Inamdar. *IKEv2 IPsec Virtual Private Networks: Understanding and Deploying IKEv2, IPsec VPNs, and FlexVPN in Cisco IOS.* Cisco Press. 2016. 656 p.
10. Douglas Crawford PPTP vs L2TP vs OpenVPN vs SSTP vs IKEv2. 2014. Available at: www.bestvpn.com/blog/4147/pptp-vs-l2tp-vs-openvpn-vs-sstp-vs-ikev2/
11. William (Chuck) Easttom. *Computer Security Fundamentals, 3rd Edition.* Pearson IT Certification. 2016. 448 p.
12. William (Chuck) Easttom. *Computer Security Fundamentals, 3rd Edition.* Pearson IT Certification. 2016. 448 p

13. Choosing EAP-TLS or MS-CHAP v2 for User-Level Authentication. *Microsoft TechNet*. March 28, 2003

14. Bruce Schneier, *Cryptanalysis of Microsoft's Point to Point Tunneling Protocol (PPTP)*. Available at: <https://www.schneier.com/academic/paperfiles/paper-pptp.pdf>

15. Richard Chirgwin. *Marlinspike demos MS-CHAPv2 crack. 'The strength of a single DES encryption' not enough*. 2012. Available at: http://www.theregister.co.uk/2012/07/31/ms_chapv2_crack/

16. Moxie Marlinspike. *Divide and Conquer: Cracking MS-CHAPv2 with a 100% success rate*. 2012. Available at: <https://www.cloudcracker.com/blog/2012/07/29/cracking-ms-chap-v2/>

17. Dj Walker-Morgan. *Microsoft says don't use PPTP and MS-CHAP*. 2012. Available at: <http://www.h-online.com/security/news/item/Microsoft-says-don-t-use-PPTP-and-MS-CHAP-1672257.html>

18. Безопасность и резильентность систем и сетей. Практикум / И.В. Жуковицкий, Д.А. Остапец, С.А. Разгонов, А.П. Заец - Под ред. Жуковицкого И.В. – Харьков: Национальный аэрокосмический университет имени Н.Е. Жуковского «ХАИ». – 2017. – 131 с. [In Russian: System and Networks Security and Resilience. Practical workshop / I.V. Zhukovytskyu, D.A. Ostapets, S.A. Razgonov, A.P. Zaec – Ed. Zhukovytskyu I.V. – Kharkov: National Aerospace University named after N.E. Zhukovsky "Khai"].

19. Райан Норман. *Выбираем протокол VPN*. Windows IT Pro/RE. 2001. № 07. Available at: <https://www.osp.ru/winitpro/2001/07/175027/>

20. *L2TP for Subscriber Access Overview*. 2016. Available at: https://www.juniper.net/documentation/en_US/junos/topics/concept/subscriber-management-l2tp-overview.html#fig-typical-l2tp-topology/

21. Mantas, G., Lymberopoulos, D. & Komninos. *PKI security in large-scale healthcare networks*. Journal of Medical Systems, 36(3). pp. 1107-1116 (2012).

RESILIENCE MECHANISMS OF SYSTEMS AND NETWORKS

CONTENT SECTION

- 5.1 General concepts of resilience
- 5.2 Resilience disciplines.....
 - 5.2.1 Challenge tolerance
 - 5.2.2 Disciplines relating to trustworthiness
 - 5.2.3 Robustness and complexity
- 5.3 ResiliNets framework and strategy
- 5.3.1 Scope and Definition
- 5.3.2 ResiliNets axioms
- 5.3.3 ResiliNets strategy
- 5.3.4 ResiliNets design principles
- 5.4 Framework for resilience
- 5.4.1 The approach to the formation of the framework
for resilience structure
- 5.4.2 Resilience control loop
- 5.5 Resilience metrics framework
- 5.6 Understanding challenges and risks
- 5.7 Defense and dynamic adaptation architecture
- Conclusion
- Questions for self-control
- Bibliography

5.1 General concepts of resilience

The vulnerabilities of the current Internet and the need for greater resilience are widely recognized. Resilience evidently cuts through several thematic areas, such as information and network security, fault tolerance, software dependability, and network survivability.

In [1] indicates that no matter what strategy is adopted, breaches will occur. It is nearly impossible to take advantage of our connectedness without being at risk. Defensive technologies such as firewalls, passwords, encryption, physical barriers, and authentication mechanisms are important to maintain but alone have not been effective in eliminating breaches or predicting where the next attack will occur. Their value as stand-alone security measures will be of limited use in fighting increasingly sophisticated, innovative, and well-funded cyber criminals.

The emerging challenge is to find more predictive methods of identifying threats, mitigating their impact, and managing an agile cyber security operation that will both creatively and effectively maintain protection. Such protection can be implemented by means of a resilient enterprise

Designing a Resilient Enterprise

What is resilience? Merriam-Webster's dictionary defines resilience [2] as:

- the capability of a strained body to recover its size and shape after deformation caused especially by compressive stress;
- an ability to recover from or adjust easily to misfortune or change.

Resilience evidently cuts through several thematic areas, such as information and network security, fault tolerance, software dependability, and network survivability. Scientific investigations concerning the resilience of large, natural systems date back to the mid-1970s, typified by the pioneering work of Holling on the resilience and stability of complex ecological systems [3].

During the past decade, system resilience has received increased attention due to research efforts in several system domains. Examples include multipartner projects such as IRIS (Infrastructure for Resilient Internet Systems [4]) in the United States and ReSIST (Resilience for Survivability in IST [5]) in Europe.

Contemporary definitions of system resilience differ somewhat according to the assumed nature of a system's application environment. A common property, however, is the ability to cope with unanticipated system and environmental conditions that might otherwise cause a loss of acceptable service (failure) [6].

For example, in the context of applications where safety is the principal concern (particularly human safety, where failures can result in the loss of lives), Hollnagel (in the Prologue of [7]) defines resilience as:

The intrinsic ability of a system to adjust its functioning prior to, during, or following changes and disturbances, so that it can sustain required operations under both expected and unexpected conditions.

The US DoD “Engineered Resilient Systems” initiative defines a resilient system to be one which

“... is trusted and effective out of the box in a wide range of contexts, easily adapted to many others through reconfiguration or replacement, with graceful and detectable degradation of function”.

The following development of the ReSIST definition of resilience is quoted verbatim from the Laprie paper cited earlier [8]: “With such ubiquitous systems, what is at stake is to maintain dependability, i.e., the ability to deliver service that can justifiably be trusted in spite of continuous changes. Our definition of resilience is then:

The persistence of service delivery that can justifiably be trusted, when facing changes”.

Although concern with unanticipated conditions is not explicit in the above definition, it becomes obvious once “changes” are defined in various ReSIST documents. In particular, they introduce a “prospect” dimension of change that includes an unforeseen category: see Fig. 5.1 (taken from [2]).

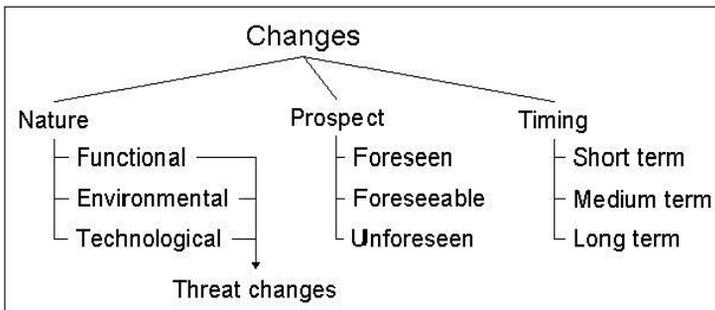


Fig. 5.1. ReSIST Classification of Changes, reprinted from [2]

5.2 Resilience disciplines

There are a number of relevant disciplines that serve as the basis of network resilience, and for which [9] a broad definition of resilience

subsumes.

At the highest level, we divide the disciplines into two categories, as shown in Fig. 5.2. On the left side are challenge tolerance disciplines that deal with the design and engineering of systems that continue to provide service in the face of challenges. On the right sides are trustworthiness disciplines that describe measurable properties of resilient systems. The relationship between these two is robustness, which formally is the performance of a control system when perturbed, or in our context, the trustworthiness of a system when challenged.

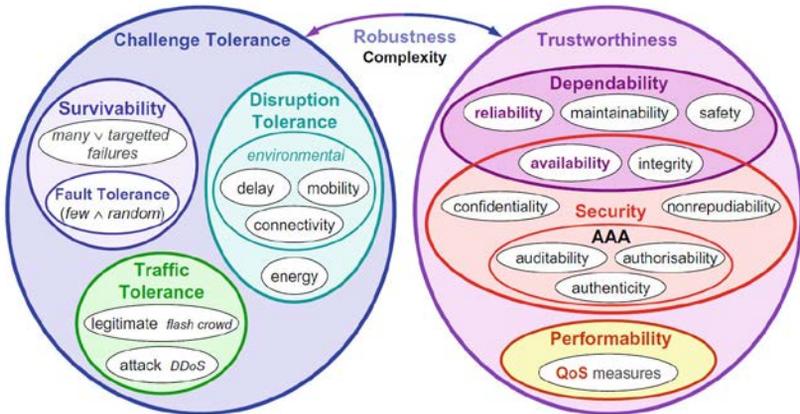


Fig. 5.2. Resilience disciplines, reprinted from [9].

5.2.1 Challenge tolerance

The first major subset of resilience disciplines deals with the problem of how to design systems to tolerate the challenges that prevent the desired service delivery. These challenges can be subdivided into (1) component and system failures for which fault tolerance and survivability are concerned, (2) disruptions of communication paths for which disruption tolerance is concerned, and (3) challenges due to the injection of traffic into the network, for which traffic tolerance is concerned.

Fault tolerance

Fault tolerance is one of the oldest resilience disciplines, and is defined as the ability of a system to tolerate faults such that service failures do not result.

Fault tolerance relies on redundancy as a technique to compensate for the random uncorrelated failure of components. Fault tolerance is not sufficient to provide coverage in the face of correlated failures, and therefore is necessary but not sufficient to provide resilience. Thus, fault tolerance can be considered a subset of survivability.

Survivability

The emergence of and dependence on the Internet lead to the realization that new techniques were needed for unbounded networks that could be affected by correlated failures for which fault-tolerant design techniques are not sufficient. Survivability is the capability of a system to fulfill its mission, in a timely manner, in the presence of threats such as attacks or large-scale natural disasters. This definition captures the aspect of correlated failures due to an attack by an intelligent adversary, as well as failures of large parts of the network infrastructure.

In addition to the redundancy required by fault tolerance, survivability requires diversity so that the same fate is unlikely to be shared by parts of the system undergoing correlated failures.

While survivability is significantly more difficult to quantify than fault tolerance, it has been formalized as a set-theoretic and state-machine based formulation [10]:

$$\text{Survivability } \{S; E; D; V; T; P\},$$

where S is the set of acceptable service specifications, E describes the ways in which the system can degrade based on external challenges, D are the practical values of E , V is the relative ordering of service values $S \times D$; $T \subseteq S \times S \times D$ is the set of valid transitions between service states S given a challenge D , and P are the service probabilities that some $s \in S$ must meet dependability requirements.

Disruption tolerance

Another major type of challenge that is unique to communication networks comes from challenges in the communication environment that make it difficult to maintain stable end-to-end connections between users. Disruption tolerance is the ability of a system to tolerate disruptions in connectivity among its components, consisting of the environmental challenges: weak and episodic channel connectivity, mobility, unpredictably-long delay, as well as tolerance of energy (or power) challenges.

Traffic tolerance

The last major challenge category is that caused by the injection of traffic into the network. Traffic tolerance is the ability of a system to tolerate

unpredictable offered load without a significant drop in carried load (including congestion collapse), as well as to isolate the effects from crosstraffic, other flows, and other nodes. In defining traffic as a challenge, we mean traffic beyond the design parameters of the network in its normal operation. Traffic challenges can either be unexpected but legitimate such as from a flash crowd [11], or malicious such as from a distributed denial-of-service (DDoS) attack [12]. It is important to note that while DDoS detection is an important endeavour, network resources are impacted regardless of whether traffic is malicious or not. Furthermore, a sufficiently sophisticated DDoS attack is indistinguishable from normal traffic, and thus traffic tolerance mechanisms are important whether or not attack detection mechanisms are successful.

5.2.2 Disciplines relating to trustworthiness

Trustworthiness is defined as the assurance that a system will perform as expected [13], which must be with respect to measurable properties. The trustworthiness disciplines therefore measure service delivery of a network, and consist of (1) dependability, (2) security, and (3) performability.

Dependability

Dependability is the discipline that quantifies the reliance that can be placed on the service delivered by a system [14,15], and consists of two major aspects: availability and reliability. Important to both of these aspects are the expected values of the failure and repair density functions. The basic measures of dependability are the MTTF (mean time to failure), which is the expected value of the failure density function, and the MTTR, which is the expected value of the repair density function. The mean time between failure is the sum of these two [16]:

$$MTBF = MTTF + MTTR.$$

Availability is readiness for usage, which is the probability that a system or service will be operable when needed, and is calculated as

$$A = MTTF/MTBF.$$

Reliability is continuity of service, that is the probability that a system or service remains operable for a specified period of time:

$$R(t) = \Pr(\text{no failure in}(0; t)) = 1 - Q(t),$$

where $Q(t)$ is the failure cumulative distribution function.

These notions of dependable systems have been codified by IFIP WG 10.4 [17] and ANSI T1A1 [18] and are commonly applied to network dependability.

Security

Security is the property of a system, and the measures taken such that it protects itself from unauthorized access or change, subject to policy [19]. Security properties include AAA (authenticity, authorisability, auditability), confidentiality, and nonrepudiability. Security shares with dependability the properties of availability and integrity.

Performability

Performability [20] is the property of a system such that it delivers performance required by the service specification, as described by QoS (quality of service) measures such as delay, throughout or goodput, and packet delivery ratio.

5.2.3 Robustness and complexity

Two disciplines lie outside challenge tolerance and trustworthiness, but describe their relationship to one another (robustness) and overall characteristics (complexity).

Robustness

Robustness is a control-theoretic property that relates the operation of a system to perturbations of its inputs [21,22]. In the context of resilience, robustness describes the trustworthiness (quantifiable behaviour) of a system in the face of challenges that change its behaviour.

Complexity

Complexity science has an important relationship to resilience and the robustness of systems, because resilience mechanisms such as self-organization and autonomic behaviour increase complexity, and increased complexity may result in greater network vulnerability.

5.3 ResiliNets framework and strategy

The ResiliNets initiative [23] has developed a framework for resilient networking [24], initially as part of the Autonomic Network Architecture (ANA) [25,26] and Postmodern Internet Architecture (PoMo) [27,28] projects, serving as the basis of the ResumeNet (Resilience and Survivability for Future Networking: Framework, Mechanisms, and Experimental Evaluation) project [29,30]. This initiative was heavily influenced by the frameworks described above, and can be viewed as a

successor and synthesis of all of them.

5.3.1 Scope and Definition

The resilient and survivable networking initiative (ResiliNets) is investigating the architecture, protocols, and mechanisms to provide resilient, survivable, and disruption-tolerant networks, services, and applications.

Resilience is the ability of the network to provide and maintain an acceptable level of service in the face of various challenges to normal operation [31]:

- unusual but legitimate traffic load (e.g. flash crowds);
- high-mobility of nodes and subnetworks;
- weak, asymmetric, and episodic connectivity of wireless channels;
- unpredictably long delay paths either due to length (e.g. satellite) or as a result of episodic connectivity;
- attacks against the network hardware, software, or protocol infrastructure (from recreational crackers, industrial espionage, terrorism, or warfare);
- large-scale natural disasters (e.g. hurricanes, earthquakes, ice storms, tsunamis, floods);
- failures due to mis-configuration or operational errors;
- natural faults of network components.

Relationship of resilience to survivability and disruption tolerance

The primary difference between definition [31] of resilience vs. survivability and disruption tolerance is that resilient networks are engineered to tolerate legitimate but unpredictably high-traffic loads (such as flash crowds), while maximizing the service provided to other users of the network, as well as being resistant to attack.

Survivability is the capability of a system to fulfill its mission in a timely manner, even in the presence of attacks or failures [31], including large scale natural disasters.

Disruption tolerance is the ability for end-to-end applications to operate even when network connectivity is not strong (weak, episodic, or asymmetric) and the network is unable to provide stable end-to-end paths.

Thus survivability and disruption tolerance are necessary but not sufficient for resilience.

Relationship of resilience to fault tolerance

Fault tolerance the ability of a system or component to continue normal operation despite the presence of hardware or software faults [31].

Fault tolerant systems are generally engineered only to tolerate isolated random natural failures. Thus, fault tolerance is necessary but not sufficient for survivability (and therefore resilience). We do believe that we can learn from past work in fault tolerance, particularly by extending work in design methodology and metrics.

5.3.2 ResiliNets axioms

Axioms provide the basis for any systematic framework; in [9] present four basic self-evident tenets that form the basis for the ResiliNets strategy.

A0. Faults are inevitable; it is not possible to construct perfect systems, nor is it possible to prevent challenges and threats.

A1. Understanding normal operation is necessary, including the environment, and application demands. It is only by understanding normal operation that we have any hope of determining when the network is challenged or threatened.

A2. Expectation and preparation for adverse events and conditions is necessary, so that defences and detection of challenges that disrupt normal operations can occur. These challenges are inevitable.

In [9] further classify adverse events and conditions by severity as mild, moderate, or severe, and categorize them into two types:

(1) Anticipated adverse events and conditions are ones that we can predict based either on past events (such as natural disasters), and attacks (e.g. viruses, worms, DDoS) or that a reasoned threat analysis would predict might occur.

(2) Unanticipated adverse events and conditions are those that we cannot predict with any specificity, but for which we can still be prepared in a general sense. For example, there will be new classes of attacks for which we should be prepared.

A3. Response to adverse events and conditions is required for resilience, by remediation ensuring correct operation and graceful degradation, restoration to normal operation, diagnosis of root cause faults, and refinement of future responses.

5.3.3 ResiliNets strategy

In [2] ResiliNets strategy is formalized as a two-phase strategy $D^2R^2 + DR$, as shown in Fig. 5.3. At the core are passive structural defences. The first active phase, D^2R^2 : Defend, Detect, Remediate, Recover, is the inner control loop and describes a set of activities that are undertaken in order for a system to rapidly adapt to challenges and attacks and maintain an

acceptable level of service. The second active phase DR: Diagnose, Refine, is the outer loop that enables longer-term evolution of the system in order to enhance the approaches to the activities of phase one.

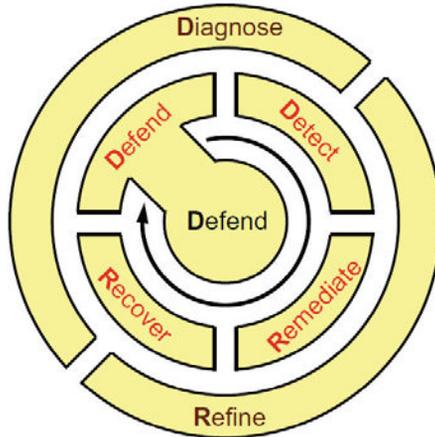


Fig. 5.3. ResiliNets strategy, reprinted from [9].

D²R² inner loop

The first strategy phase consists of a passive core and a cycle of four steps that are performed in real time and are directly involved in network operation and service provision.

- S1. Defend against challenges and threats to normal operation.
- S2. Detect when an adverse event or condition has occurred.
- S3. Remediate the effects of the adverse event or condition.
- S4. Recover to original and normal operations.

DR outer loop

The second phase consists of two background operations that observe and modify the behaviour of the D^2R^2 cycle: diagnosis of faults and refinement of future behaviour.

- S5. Diagnose the fault that was the root cause.
- S6. Refine behaviour for the future based on past D^2R^2 cycles.

This is an ongoing process that requires that the network infrastructure, protocols, and resilience mechanisms be evolvable.

In [11] indicates that it is essential to solve the problem of resilience on all levels (Fig. 5.4), both from a network architectural perspective as well as

from a protocol layering and plane viewpoint. Starting from the bottom-up, each level is made as resilient as practical (understanding cost and resource tradeoffs). Higher levels are themselves organized into resilient structures using the resilient lower-level building blocks.

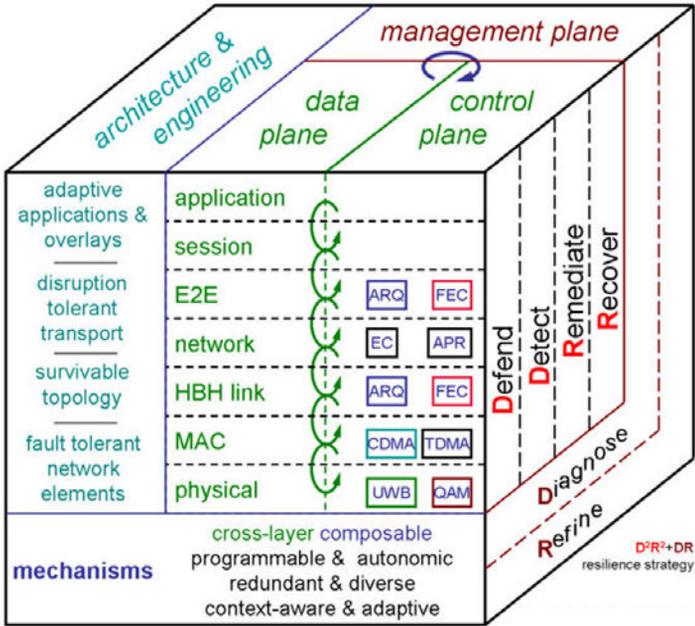


Fig.4. Cube strategy $D^2R^2 + DR$ resiliens, reprinted from [31]

5.3.4 ResiliNets design principles

In [9] ResiliNets principles is formalized as a four groups (Fig. 5.4.).

Prerequisites

Five principles span the domain of prerequisites necessary to build a resilient system (Fig. 5.4).

P1. Service requirements of applications need to be determined to understand the level of resilience the system should provide. In this sense, resilience may be regarded as an additional QoS property along with conventional properties such as performance.

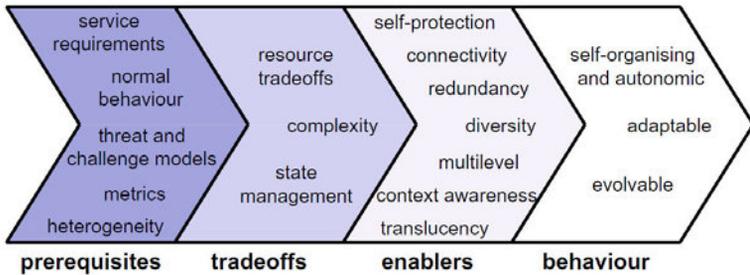


Fig. 5.4. Resilience principles, reprinted from [9]

P2. Normal behaviour of the network is a combination of design and engineering specification, along with monitoring while unchallenged to learn the network's normal operational parameters. This is a fundamental requirement (A1) for detecting (S2) challenges.

P3. Threat and challenge models are essential to understanding and detecting potential adverse events and conditions.

P4. Metrics quantifying the service requirements and operational state are needed to measure the operational state (in the range normal – partially-degraded – severely-degraded) and service state (in the range acceptable M impaired M unacceptable) to detect and remediate (S1–2) and quantify resilience to refine future behaviour (S6).

P5. Heterogeneity in mechanism, trust, and policy are the realities of the current world.

The Global Internet is a collection of realms of disparate technologies. Furthermore, realms are defined by trust and policy, across which there is tussle. Resilience mechanisms must deal with heterogeneous link technologies, addressing, forwarding, routing, signaling, traffic, and resource management mechanisms.

Design tradeoffs

Three principles describe fundamental tradeoffs that must be made while developing a resilient system (Fig. 5.4).

P6. Resource tradeoffs determine the deployment of resilience mechanisms. The relative composition and placement of these resources must be balanced to optimize resilience and cost. Resources to be traded against one another include bandwidth, memory, processing, latency, energy, and monetary cost.

P7. Complexity of the network results due to the interaction of systems at multiple levels of hardware and software, and is related to scalability. The

degree of complexity [33,34,35] must be carefully balanced in terms of cost vs. benefit, and unnecessary complexity should be eliminated.

P8. State management is an essential part of any large complex system. It is related to resilience in two ways: First, the choice of state management impacts the resilience of the network. Second, resilience mechanisms themselves require state.

Enablers

Seven principles are enablers of resilience that guide network design and engineering (Fig. 5.4).

P9. Self-protection and security are essential properties of entities to defend against challenges (A2) in a resilient network. Self-protection is implemented by a number of mechanisms, including but not limited to mutual suspicion, the AAA mechanisms of authentication, authorization, and accounting, as well as the additional conventional security mechanisms of confidentiality, integrity, and nonrepudiation.

P10. Connectivity and association among communicating entities should be maintained when possible based on eventual stability, but information flow should still take place even when a stable end-to-end path does not exist based on the eventual connectivity model.

P11. Redundancy in space, time, and information increases resilience against faults and some challenges if defences (S1) are penetrated. Redundancy refers to the replication of entities in the network, generally to provide fault tolerance.

P12. Diversity is closely related to redundancy, but has the key goal to avoid fate sharing. Diversity in space, time, medium, and mechanism increases resilience against challenges to particular choices. Diversity consists of providing alternatives so that even when challenges impact particular alternatives, other alternatives prevent degradation from normal operations.

P13. Multilevel resilience is defined with respect to protocol layer, protocol plane, and hierarchical network organization

P14. Context awareness is needed for resilient nodes to monitor the network environment (channel conditions, link state, operational state of network components, etc.) and detect adverse events or conditions.

P15. Translucency [36,37] is needed to control the degree of abstraction vs. the visibility between levels.

Behaviour needed for resilience

The last group of three principles encompass the behaviours and properties a resilient system should possess (Fig. 5.4).

P16. Self-organizing and autonomic behaviour is necessary for network resilience that is highly reactive with minimal human intervention. A

resilient network must initialize and operate itself with minimal human configuration, management, and intervention.

P17. Adaptability to the network environment is essential for a node in a resilient network to detect, remediate, and recover from challenges. Resilient network components need to adapt their behaviour based on dynamic network conditions, in particular to remediate (S3) from adverse events or conditions, as well as to recover (S4) to normal operations. At the network level, programmable and active network techniques enable adaptability.

P17. Evolvability is needed to refine (S6) future behaviour to improve the response to challenges, as well as for the network architecture and protocols to respond to emerging threats and application demands.

5.4 Framework for resilience

5.4.1 The approach to the formation of the framework for resilience structure

The resilience framework builds on work by Sterbenz et al. [38], whereby a number of resilience principles are defined, including a resilience strategy, called D2R2 + DR: Defend, Detect, Remediate, Recover, and Diagnose and Refine. The strategy describes a real-time control loop to allow dynamic adaptation of networks in response to challenges, and a non-real time control loop that aims to improve the design of the network, including the real-time loop operation, reflecting on past operational experience.

The framework represents the systematic approach to the engineering of network resilience. At its core is a control loop comprising a number of conceptual components that realize the real-time aspect of the D2R2 + DR strategy, and consequently implement network resilience. Based on the resilience control loop, other necessary elements of our framework are derived, namely resilience metrics, understanding challenges and risks, a distributed information store, and policy-based management.

5.4.2 Resilience control loop

Based on the real-time component of the $D^2R^2 + DR$ strategy, in [39] was developed a resilience control loop, depicted in Fig. 5.1, in which a controller modulates the input to a system under control in order to steer the system and its output towards a desired reference value. The control loop forms the basis of systematic approach to network resilience — it defines

preserve the target service provision level or ensure its graceful degradation. This adaptation is directed using resilience knowledge, not shown in Fig. 5.6, such as policies and challenge models. It is expected a cost of remediation in terms of a potentially unavoidable degradation in quality of service (QoS), which should not be incurred if the challenge abates. Consequently, the network should aim to recover to normal operation after a challenge has ceased.

The purpose of the background loop in the $D^2R^2 + DR$ strategy is to improve the operation of the resilience control loop such that it meets an idealized system operation. This improvement could be in response to market forces, leading to new resilience targets, new challenges, or suboptimal performance. The diagnose phase identifies areas for improvement, including defense, that are enacted through refinement. In reality, and for the foreseeable future. It is expected this outer loop to be realized with human intervention.

5.5 Resilience metrics framework

Defining a resilience target requires appropriate metrics. It would be ideal, as stated in [4,7], to express the resilience of a network using a single value, \mathbf{R} , in the interval $[0,1]$, but this is not a simple problem because of the number of parameters that contribute to and measure resilience, and due to the multilayer aspects in which each level of resilience (e.g., resilient topology) is the foundation for the next level up (e.g., resilient routing). In [39,40] proposed to model resilience as a two-dimensional state space in which the vertical axis \mathbf{P} is a measure of the service provided when the operational state \mathbf{N} is challenged, as shown in Fig. 5.7. Resilience is then modeled as the trajectory through the state as the network goes from delivering acceptable service under normal operations S_0 to degraded service S_c . Remediation improves service to S_r and recovery returns to the normal state S_0 . Maybe measure resilience at a particular service level as the area under this trajectory \mathbf{R} .

In order to optimize the resilience of a network, it should be addressed at all levels, in the sense that each layer does the best it can, given practical constraints. These constraints are often based on the cost of resilience. Therefore, resilience must be analyzed at each layer individually as well as for the network as a whole. For this purpose, the metrics framework supports multilevel resilience evaluation. Formally, resilience \mathbf{R}_{ij} is defined at the boundary B_{ij} between any two adjacent layers L_i, L_j [40].

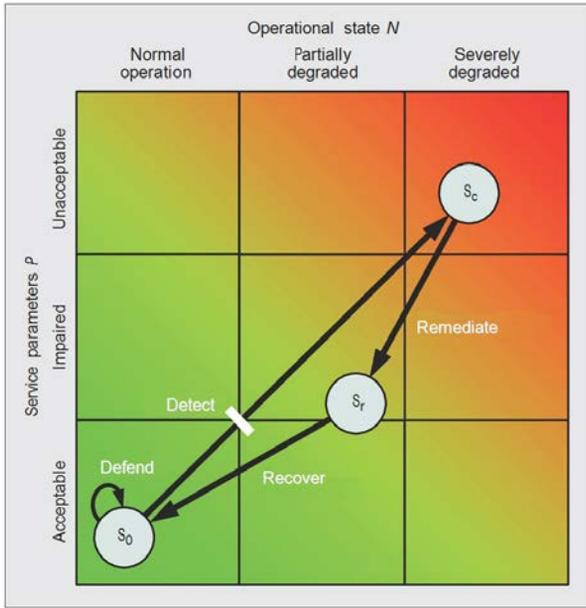


Fig. 5.7. Resilience state space, reprinted from [40]

Based on the formulation discussed earlier, let there be a set of k operational metrics $\mathbf{N} = \{N_1, N_2, \dots, N_k\}$ that characterize the state of the network below the boundary B_{ij} . Similarly, let there be a set of l service parameters $\mathbf{P} = \{P_1, P_2, \dots, P_l\}$ that characterize the service from layer i to layer j . Resilience R_{ij} at the boundary B_{ij} is then evaluated as the transition of the network through this state space. The goal is to derive the \mathbf{R}_{ij} as a function of \mathbf{N} and \mathbf{P} . In the simplest case \mathbf{R}_{ij} is the area under the curve obtained by plotting \mathbf{P} vs. \mathbf{N} on a multivariate piecewise axis. In the multilevel analysis, as shown in Fig. 5.8, the service parameters at the boundary B_{ij} become the operation metrics at boundary $B_{i+1,j+1}$. In other words, the service provided by a given layer becomes the operational state of the layer above, which has a new set of service parameters characterizing its service to the layer above.

This state space approach provides a way of representing and reasoning about multilevel resilience. One of the uses of the state space concept is to represent resilience classes, which offer a possible simplification for network and service providers when they wish to describe resilience in a Service Level Agreement.

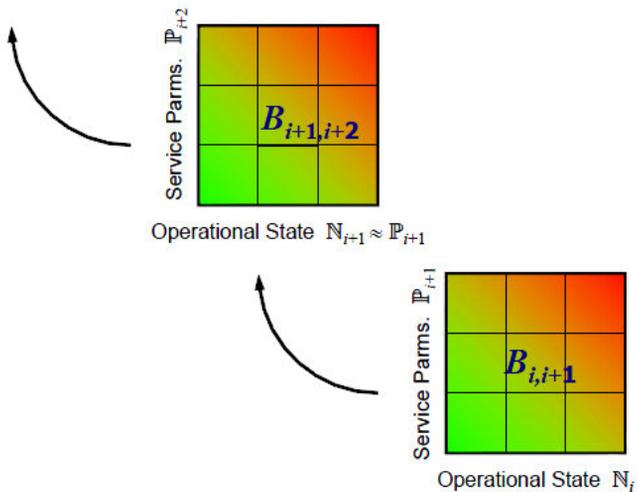


Fig. 5.8. Resilience across multiple levels, reprinted from [40]

A key aspect of metrics framework is the notion of a metric envelope. For a given metric m , or the combination \mathbf{R} as shown in Fig. 5.9, we map the trajectory of the best, average and worst case of the metric's behaviour in response to the increasing intensity of a challenge (greater values of k).

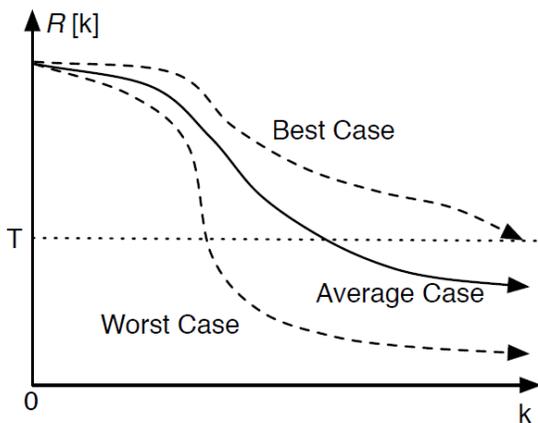


Fig. 5.9. The metric envelope concept, reprinted from [41]

5.6 Understanding challenges and risks

Engineering resilience has a monetary cost. To maximize the effectiveness of the resources committed to resilience, a good understanding of the challenges a network may face is mandatory. In [39] described a structured risk assessment approach that identifies and ranks challenges in line with their probability of occurrence and their impact on network operation

Central to determining the impact of a challenge is to identify the critical services the network provides and the cost of their disruption: a measure of impact. Various approaches can be used to identify the critical services, such as discussion groups involving the network's stakeholders. Networked systems are implemented via a set of dependent subsystems and services (e.g., web and Session Initiation Protocol (SIP) services rely on Domain Name Service (DNS)). To identify whether challenges will cause a degradation of a service, it is necessary to explicate these dependencies.

The next phase is to identify the occurrence probabilities of challenges (challenge_prob). Some challenges will be unique to a network's context (e.g., because of the services it provides), while others will not. In relation to these challenges, shortcomings of the system (e.g., in terms of faults) should be identified. The aim is to determine the probability that a challenge will lead to a failure (fail_prob). Can be used tools, analytical modeling, and previous experience (e.g., in advisories) to help identify these probabilities. Given this information, a measure of exposure can be derived using the following equation [39]:

$$\text{exposure} = (\text{challenge_prob} \times \text{fail_prob}) \times \text{impact}.$$

With the measures of exposure at hand, resilience resources can be targeted at the challenges that are likely to have the highest impact.

However, to be able to make autonomic decisions about the nature of a wide range of challenges and how to respond to them — a necessary property of resilient networks — a broader range of information needs to be used. In addition to traditional network monitoring information, context information, which is sometimes “external” to the system can be used.

In [39] described a Distributed Store for Challenges and their Outcome (DISco), which uses a publish-subscribe messaging pattern to disseminate information between subsystems that realize the real-time loop. Such information includes actions performed to detect and remediate challenges. Information sources may report more data than we can afford or wish to relay on the network, particularly during challenge occurrences. DISco is

able to aggregate information from multiple sources to tackle this problem. Decoupling information sources from components that use them allows adaptation of challenge analysis components without needing to modify information sources. To assist the two phases of the outer loop, DISco employs a distributed peer-to-peer storage system for longer-term persistence of data, which is aware of available storage capacity and demand.

5.7 Defense and dynamic adaptation architecture

The architecture, shown in Fig. 5.10 [39], consists of several subsystems implementing the various tasks of the communication system as well as the challenge detection components and adaptation capabilities. The behavior of all these subsystems is directed by the resilience manager using policies, which are held in a resilience knowledge base. Central to this architecture is DISco, which acts as a publish-subscribe and persistent storage system, containing information regarding ongoing detection and remediation activities.

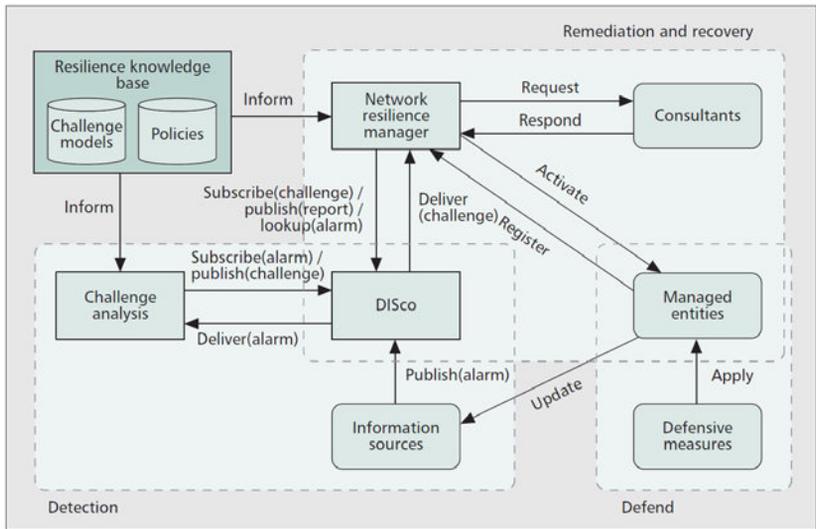


Fig. 5.10. A dynamic adaptation architecture that realizes the resilience control loop, reprinted from [39]

Defensive measures

As a first step, defensive measures need to be put in place to alleviate the impact of challenges on the network. Since challenges may vary broadly from topology-level link failures to application-level malware, defensive measures against anticipated high-impact challenges need to be applied at different levels and locations: in the network topology design phase, and within protocols; across a network domain, as well as at individual nodes. Defensive measures can either prevent a challenge from affecting the system or contain erroneous behavior within a subsystem in such a way that the delivered service still meets its specification.

Detection subsystems

The second step is to detect challenges affecting the system leading to a deviation in delivered service. An incremental approach is needed to challenge analysis. Thereby, the understanding about the nature of a challenge evolves as more inputs become available from a variety of information sources.

Remediation and recovery subsystems

The challenge detection subsystem interfaces with the remediation and recovery subsystem, the third and final step, by issuing alerts to DISco using the `publish(challenge)` primitive. These alerts contain information about the challenge and its impact on the network, in terms of the metrics that are falling short of the resilience target. The network resilience manager takes this information as context data, and, based on policies, selects an adaptation strategy.

Conclusion

Given the dependence of our society on network infrastructures, and the Internet in particular, we take the position that resilience should be an integral property of future networks. In this article, we have described a systematic approach to network resilience.

Contemporary definitions of system resilience is the ability to cope with unanticipated system and environmental conditions that might otherwise cause a loss of acceptable service (failure).

Resilience evidently cuts through several thematic areas, such as information and network security, fault tolerance, software dependability, and network survivability.

ResiliNets axioms:

A0. Faults are inevitable; it is not possible to construct perfect systems, nor is it possible to prevent challenges and threats.

A1. Understanding normal operation is necessary, including the environment, and application demands. It is only by understanding normal operation that we have any hope of determining when the network is challenged or threatened.

A2. Expectation and preparation for adverse events and conditions is necessary, so that defences and detection of challenges that disrupt normal operations can occur. These challenges are inevitable.

ResiliNets strategy is formalized as a two-phase strategy $D^2R^2 + DR$. At the core are passive structural defences. The first active phase, D^2R^2 : **D**efend, **D**etect, **R**emediate, **R**ecover, is the inner control loop and describes a set of activities that are undertaken in order for a system to rapidly adapt to challenges and attacks and maintain an acceptable level of service. The second active phase DR : **D**iagnose, **R**efine, is the outer loop that enables longer-term evolution of the system in order to enhance the approaches to the activities of phase one.

The framework for resilience represents the systematic approach to the engineering of network resilience. At its core is a control loop comprising a number of conceptual components that realize the real-time aspect of the $D^2R^2 + DR$ strategy, and consequently implement network resilience. Based on the resilience control loop, other necessary elements of our framework are derived, namely resilience metrics, understanding challenges and risks, a distributed information store, and policy-based management.

The strategy describes a real-time control loop to allow dynamic adaptation of networks in response to challenges, and a non-real time control loop that aims to improve the design of the network, including the real-time loop operation, reflecting on past operational experience.

Resilience metrics framework can be represented as a two-dimensional state space in which the vertical axis \mathbf{P} is a measure of the service provided when the operational state \mathbf{N} is challenged. Resilience is then modeled as the trajectory through the state as the network goes from delivering acceptable service under normal operations S_0 to degraded service S_c . Remediation improves service to S_r and recovery returns to the normal state S_0 . Maybe measure resilience at a particular service level as the area under this trajectory.

A measure of exposure can be derived using the following equation:

$$\text{exposure} = (\text{challenge_prob} \times \text{fail_prob}) \times \text{impact},$$

where challenge_prob is probabilities of challenges; fail_prob is the probability that a challenge will lead to a failure.

Questions for self-control

- 1) Formulate a concept resilience
- 2) What is a fault tolerance?
- 3) What is a survivability?
- 4) What are the disciplines concerning the reliability system.
- 5) What relationship of resilience to fault tolerance?
- 6) Specify ResiliNets axioms
- 7) Define ResiliNets strategy D^2R^2+DR
- 8) Call four steps the first D^2R^2+DR strategy phase
- 9) Call the operations of the second D^2R^2+DR strategy phase
- 10) Give an explanation of the principles that span the domain of prerequisites necessary to build a resilient system (Fig.4).
- 11) Give an explanation of the principles that describe fundamental tradeoffs that must be made while developing a resilient system (Fig. 5.4).
- 12) Give an explanation of the principles that enablers of resilience that guide network design and engineering (Fig. 5.4).
- 13) Give an explanation of the principles that encompass the behaviours and properties a resilient system should possess (Fig. 5.4).
- 14) Explain the principle of modeling a resilience as a two-dimensional state space in which the vertical axis P is a measure of the service provided when the operational state N is challenged (Fig. 5.7).
- 15) How can we determine a measure of a system exposure?

Bibliography

1. George A. Wright, Terrye N. Schaetzel. *Cyber Security: Designing and Maintaining Resilience*. White paper presented by: Georgia Tech Research Institute Cyber Technology and Information Security Laboratory. 8 p. Available at:
<http://www.globalsciencecollaboration.org/public/site/PDFS/cyber/Cyber%20Security%20white%20paper%20final.pdf>
2. Merriam-Webster. Dictionary. (2013). Available at:

<http://www.merriam-webster.com/>

3. C. S. Holling. Resilience and stability of ecological systems. *Annual Review of Ecology and Systematics*. 1973, vol. 4, no. 1, pp. 1–23.

4. *IRIS: Infrastructure for Resilient Internet Systems*. Available at: <https://pdos.csail.mit.edu/archive/iris/>

5. *Resist noe. Resilience for survivability in IST*. Available at: <http://www.resist-noe.org>.

6. J. F. Meyer. Model-based evaluation of system resilience. In: *43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W)*. 2013, pp. 1-7. DOI: 10.1109/DSNW.2013.6615535.

7. E. Hollnagel, J. Paries & D.D. Woods, J. Wreathall, et al. *Resilience Engineering in Practice: A Guidebook*. Ashcroft, 2011.

8. J.-C. Laprie. From dependability to resilience. In: *Proc. IEEE Int. Conf. on Dependable Systems and Networks*. 2008, vol. Supplemental, pp. G8–G9.

9. James P.G. Sterbenz, David Hutchison, Egemen Çetinkaya, et al. Resilience and Survivability in Communication Networks: Strategies, Principles, and Survey of Disciplines. *Computer Networks: Special Issue on Resilient and Survivable Networks (COMNET)*. June 2010, vol.54, iss.8., pp.1245–1265.

10. J.C. Knight & E.A. Strunk & K.J. Sullivan. Towards a rigorous definition of information system survivability. In: *Proceedings of the DARPA Information Survivability Conference and Exposition DISCEX III*. Washington DC. 2003, pp. 78–89.

11. J. Jung, B. Krishnamurthy, M. Rabinovich. Flash crowds and denial-of-service attacks: characterization and implications for CDNs and web sites. In: *Proceedings of the 11th International Conference on World Wide Web (WWW)*, ACM. New York, NY, USA, 2002, pp. 293–304. DOI: 10.1145/511446.511485.

12. J. Mirkovic, P. Reiher. A taxonomy of DDoS attack and DDoS defense mechanisms. *SIGCOMM Computer Communication Review*. 2004, vol. 34 (2), pp. 39–53. DOI: 10.1145/997150.997156.

13. A. Avizienis, J.-C. Laprie, B. Randell, et al. *Basic concepts and taxonomy of dependable and secure computing*, Technical Research Report TR 2004-47. Institute for Systems Research, the University of Maryland. 2004.

14. P.A. Lee, T. Anderson. *Fault Tolerance: Principles and Practice*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. 1990. ISBN:0387820779.

15. J.-C. Laprie. Dependability: basic concepts and terminology, Draft. *IFIP Working Group 10.4 – Dependable Computing and Fault Tolerance*. 1994.
16. R. Billinton, R. Allan. *Reliability Evaluation of Engineering Systems*. Plenum Press, New York, 1992.
17. A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *Transactions on Dependable and Secure Computing*. 2004, vol 1 (1), pp. 11–33.
18. T1A1.2 Working group. *Enhanced network survivability performance, Technical Report T1.TR.68-2001*. Alliance for Telecommunications Industry Solutions (ATIS), 2001.
19. C. Landwehr. Computer security. *International Journal of Information Security*. 2001, vol. 1 (1), pp. 3–13.
20. J. Meyer. Performability: a retrospective and some pointers to the future. *Performance Evaluation*. 1992, vol. 14 (3–4), pp. 139–156.
21. E. Jen. Robust Design: A Repertoire of Biological, Ecological, and Engineering Case Studies. *Oxford University Press*. 2005.
22. W. Willinger, J. Doyle. Robustness and the Internet: Design and Evolution. *Oxford University Press*. 2005.
23. J.P.G. Sterbenz, D. Hutchison. ResiliNets: Multilevel Resilient and Survivable Networking Initiative Wiki. 2008. Available at: <http://wiki.ittc.ku.edu/resilinet>
24. M. Schöller, J.P.G. Sterbenz, A. Jabbar, D. Hutchison. *First draft of the Resilience and Security Framework*. 2006. Available at: <http://www.ana-project.org/deliverables/2006/D.3.2.-Resilience.pdf>
25. *Autonomic Network Architecture Wiki*. 2006. Available at: <http://www.ana-project.org/>
26. G. Bouabene, C. Jelger, C. Tschudin, ed al. The autonomic network architecture (ANA). *IEEE Journal on Selected Areas in Communications (JSAC)*. 2010, vol. 28 (1) pp. 4–14. DOI:10.1109/ JSAC.2010.100102. ISSN: 0733-8716.
27. B. Bhattacharjee, K. Calvert, J. Griffioen, ed al. *Postmodern Internetwork Architecture, Technical Report ITTC- FY2006-TR-45030-01*. Information and Telecommunication Center, 2335 Irving Hill Road, Lawrence, KS 66045-7612. 2006.
28. J.P.G. Sterbenz, B. Bhattacharjee, K. Calvert, ed al. *PoMo: Postmodern Internetwork Architecture Wiki*, 2008. Available at: <http://wiki.ittc.ku.edu/pomo>
29. *ResumeNet Wiki*. 2009. Available at: <http://www.resumenet.eu/project/index>
30. M. Schöller, P. Smith, C. Rohner, ed at. On realising a strategy for

resilience in opportunistic networks. In: *Proceedings of the EU Future Network and Mobile Summit*. Florence, Italy, in press.

31. James P.G. Sterbenz and David Hutchison. *ResiliNets: Multilevel Resilient and Survivable Networking Initiative*. Available at: <http://www.ittc.ku.edu/resilinet/>

32. M.H. Behringer. Classifying network complexity. In: *Proceedings of the ACM Workshop on Re-architecting the Internet (ReArch)*. ACM. New York, NY, USA, 2009, pp. 13–18. DOI: 10.1145/1658978.1658983.

33 - [1-131] M.S. Blumenthal, D.D. Clark. Rethinking the design of the internet: the end-to-end arguments vs. the brave new world. *ACM Transactions on Internet Technology*. 2001, vol. 1 (1) pp. 70–109. DOI: 10.1145/383034.383037.

34. J.C. Doyle, D.L. Alderson, L. Li, ed at. The “Robust Yet Fragile” Nature of the Internet. *Proceedings of the National Academy of Sciences of the United States of America*. 2005, vol. 102 (41) pp. 14497–14502. ISSN: 00278424.

35. F. Foukalas, V. Gazis, N. Alonistioti. Cross-layer design proposals for wireless mobile networks: a survey and taxonomy. *IEEE Communications Surveys Tutorials*. 2008, vol. 10 (1) pp. 70–85. DOI:10.1109/COMST.2008.4483671. ISSN: 1553-877X.

36. J.P.G. Sterbenz, J.D. Touch. *High-Speed Networking: A Systematic Approach to High-Bandwidth Low-Latency Communication, first ed.* Wiley. 2001.

37. J. P. G. Sterbenz et al. Resilience and Survivability in Communication Networks: Strategies, Principles, and Survey of Disciplines. *Elsevier Computer Networks, Special Issue on Resilient and Survivable Networks*. 2010, vol.54, no. 8, pp. 1243–1304.

38. Paul Smith, David Hutchison, James P. G. Sterbenz, ed al. Network Resilience: A Systematic Approach. *IEEE Communications Magazine*. 2011, Vol. 49, Issue: 7, pp. 88 – 97. DOI: 10.1109/MCOM.2011.5936160.

39. P. Smith, D. Hutchison, J. P.G. Sterbenz (KU), ed al. *Resilience and Survivability for future networking: framework, mechanisms, and experimental evaluation. Final strategy document for resilient Networking*. 2011. 61 p.

40. C. Doerr, J. M. Hernandez, R. Holz, ed al. *Mieghem. Defining metrics for resilient networking (Final). ResumeNet Project Deliverable*. September 2011.

41. Oleksandr Netkachov, Peter Popov, Kizito Salako. *Model-Based Evaluation of the Resilience of Critical Infrastructures Under Cyber Attacks*. In book: *Critical Information Infrastructures Security*. 2016. pp.231-243.

42. C. Esposito, D. Cotroneo, S. Russo. *On Reliability in Publish/Subscribe Services*.

43. *Computer Networks*, Vol. 57, pp. 1318-1343 (2013)

44. D. Cotroneo, A. Pecchia, R. Pietrantuono, S. Russo. *A Method to Support Fault Tolerance Design in Service Oriented Computing Systems*. *International Journal of Systems and Service Oriented Engineering (IJSSOE)*, Vol. 1, Issue 3, 75-89 (2010)

CHAPTER 24 Resilient Internet and Cloud Computing Systems

24.1 Dependability Retrospective of Web and Cloud systems

Modern Internet computing and web services technologies support rapid, low-cost and seamless composition of globally distributed applications, and enable effective interoperability in a loosely-coupled heterogeneous environment. Web services are autonomous, platform-independent computational entities that can be dynamically discovered and integrated into a single service to be offered to the users or, in turn, used as a building block in further composition. They can be provided by third-party companies and hosted on corporate web servers, in the Clouds or private data centres.

The essential principles of service provisioning form the foundation for various modern and emerging IT technologies, such as Cloud Computing (software-as-a-service, SaaS; platform-as-a-service, PaaS; infrastructure-as-a-service, IaaS) and, in more general, Everything-as-a-Service (EaaS).

The service-oriented paradigm of cooperation via the Internet is now widely used in e-science, critical infrastructures, business-critical systems, smart-applications and Internet-of-Things technologies. Failures of such applications can affect people's lives and businesses. Thus, ensuring dependability and security of Web and Cloud-based systems is a must, as well as a challenge.

Cloud computing is an emergent technology supporting the pay-as-you-go paradigm for delivering computing as a service [1]. Reasonable price, unlimited computing and data storage resources cause a growing interest among corporate and individual users in the migration of their applications to the Clouds. However, one of the main stumbling blocks in making Internet and Cloud computing ubiquitous is the potential lack of dependability and security of the services and service-oriented systems due to high complexity, and the inability of customers to justifiably trust in the claimed performance, security, reliability and quality of the third-party services.

A well known and widely adopted dependability conceptual framework [2] was proposed by Avizienis A., Laprie J.C., Randell B., and Landwehr C.E. in 2004. It systematized and generalized a long series of earlier work in various strands from the dependability and security domains. The authors define *dependability* as the ability to deliver service that can justifiably be trusted. The proposed dependability taxonomy are organised into three categories: dependability attributes (availability, reliability, safety, confidentiality, integrity, maintainability), threats to dependability (faults, errors, failures) and means to ensure dependability (fault prevention, fault tolerance, fault removal, fault forecasting).

24.2 Resilience of Ubiquitous Computing Systems

Recent technological advances in information and communication technologies, new paradigms of service-oriented systems, Internet, Cloud and ubiquitous computing as well as modern software development approaches have exposed certain limitations of the dependability concept.

Ubiquitous computing systems often refer to the future large, networked, continuously evolving computing systems constituting complex information infrastructures involving everything from super-computers and huge server “farms” to myriads of small mobile computers and tiny IoT devices.

Ensuring dependability of such systems working in open and changeable environment requires development of new concepts and principles.

The notion of *resilience* has been introduced to fill the dependability gap and to cope with the continuous changes in system requirements, environment and operational conditions, internal system structure and components characteristics in addition to errors, faults and failures traditionally as dealt with by the dependability community.

The notion of resilience has been elaborated in various application domains, including material science, child psychiatry and social psychology, ecology, business and industrial safety.

In ICT systems the term *resilience* was first introduced in the 1970s and 1980s [3, 4, 5, 6]. However, it has been most intensively studied by the research community only during the last decade.

Nowadays many researchers from the dependability community in information and communication science [7, 8, 9, 10, 11, 12] share the following definitions of resilience tightly connected with the dependability notion [2]:

- (i) *the persistence of service delivery that can justifiably be trusted, when facing changes;*
- (ii) *the persistence of the avoidance of failures that are unacceptably frequent or severe, when facing changes;*
- (iii) *the persistence of dependability when facing changes.*

Modern definition of resilience puts forward a notion of recovery after “unforeseen events” and includes the effects of evolution through the “change” concept. Changes here may refer to unexpected failures, intrusions or accidents, increased load, etc.

The changes can be classified according to the three viewpoints [8]:

- (i) *nature*: functional, environmental, or technological (either or both hardware and software);

(ii) *prospect*: foreseen (as in new versioning), foreseeable (as in the advent of new hardware platforms), or unforeseen (as drastic changes in service requests or new types of threats).

(iii) *timing*: short term (e.g., seconds to hours, as in dynamically changing systems like spontaneous or ‘ad-hoc’ networks of mobile nodes and sensors), medium term (e.g., hours to months, as in new versioning or reconfigurations), or long term (e.g., months to years, as in reorganizations resulting from merging of systems in company acquisitions).

Besides, we propose to categorize changes regarding where they have been originated:

(i) *internal* (e.g. changes in system structure or in the states, characteristics (dependability, performance, security) or functionality of system components as far as modern systems are usually composed out of the third-party components which can be out of the general administrative control);

(ii) *external* (e.g. changes in the environment, external resources, operational profiles or user requirements).

In addition to the resilience definition mentioned above, Nicolas Gueffi proposed to view system resilience as its ability to evolve (for example, by versioning or upgrading) during the life cycle towards improving system capabilities to avoid failures and reduce degradations [13].

In their work N. Gueffi and L. Lúcio provide the mathematical definition of the resilience concept and consider *resilience* regarding particular system property. For instance, if one considers security as a property of interest of an evolving e-banking system, that e-banking system would be considered resilient regarding security if the number of successful attacks involving unauthorized money transfers would diminish over the system’s lifetime [14].

It should be noted that there are two main resilience communities working almost independently. Majority of the research works referred above belong to the *European community*, which naturally evolve from the *dependability research school* founded and developed by T. Anderson, B. Randell, J.C. Laprie, A. Avizienis, L. Strigini, K.S. Trivedi, etc.

The second resilience community has been founded and influenced mainly by American researchers originated from the *safety-critical systems research school* led by N. Leveson, D. Woods, E. Hollnagel and others. These researchers focus on how resilience concept can be brought into engineering safety-critical systems [15, 16, 17, 18] and claim that unsafe state may arise because of insufficient or inappropriate system adjustment/adaptation to the changeable operational environment, new demands or threads rather than because internal system failures. Thus, they define resilience as a proactive approach for safety management and adaptation to varying demands and

threats. In [18] the authors define four different aspects of system resilience that can be targeted in different application domains:

- (i) ability to rebound and reconfigure;
- (ii) ability to maintain a desirable state;
- (iii) ability of the systems to withstand stress, and
- (iv) ability to adapt and thrive.

Recently, a notion of resilience has been widely adopted and used in various application domains such as computer networks and large scale networked systems [19, 20, 21], Service-oriented architecture [22], Cloud computing [23, 24], etc.

24.3 The Threat of Uncertainty

24.3.1 Factors of Uncertainty

A significant part of the modern software applications, especially those implementing paradigms of Internet and ubiquitous computing, work in an unstable environment as a part of globally-distributed and loosely-coupled environment, communicating with a number of other devices and services deployed by the third-party companies typically with the unknown QoS, dependability and performance characteristics.

Ensuring and assessing dependability of such systems is complicated when these systems are dynamically built or when their components (i.e. web services, IoT sensors, Smart devices, etc.) are dynamically replaced by the new ones with the same (or similar) functionality but unknown dependability and performance characteristics.

By their very nature hardware and software building block of ubiquitous computing systems are black boxes, as neither their source code, nor their complete specification, nor information about their deployment environments are available. Moreover, their dependability is not completely known and they may not provide a sufficient quality of service. The only known information about, for instance, web services, is their programming interfaces.

Thus, it is often safe to treat them as “dirty” boxes, assuming that they always have bugs and vulnerabilities, do not fit enough, have poor specification and documentation. Modern Internet computing applications are heterogeneous, as their components might be developed by different companies following different standards, fault assumptions, and different conventions and may use different technologies. Finally, the majority of modern distributed applications are built as overlay networks over the Internet. Therefore, their construction and composition are complicated by the fact that

the Internet is a poor communication medium (e.g. it has low quality and is not predictable).

As a result users cannot be confident in availability, trustworthiness, reasonable response time and others dependability characteristics [2] which can vary over wide ranges in a very random and unpredictable manner. In this work we use the general synthetic term *uncertainty* to refer to the unknown, unstable, unpredictable, changeable characteristics and behaviour of modern ubiquitous systems, exacerbated by running them over the Internet as a composition of third-party components (i.e. web services, virtual computing instances, data storages, IoT sensors and Smart devices, etc.).

This uncertainty [25, 26] exhibits itself through the unpredictable response times of the Internet message and data transfers, the difficulty to diagnose the root cause of service failures, the inability to see beyond the interfaces of a service or a device, unknown common mode failures, etc. There are several important consequences of such uncertainty:

- there is *no guaranty of the correctness* of a system response provided to a user (the response can contain hidden errors; it can be an exception or a silence);

- there is *no guaranty of a certain response time* (our practical experiments has showed that the average response time is not typical as its standard deviation often exceeds 100%; at the same time the worst-case execution time can be dozens of times higher than the average value);

- there is *no confidence in dependability and QoS system characteristics* (availability, trustworthiness, security, performance, etc.);

- there is *no objectivity in user experience/knowledge* (it might seem that the same system is highly reliable but has bad performance for some users and unreliable but highly responsive for another ones).

The uncertainty discovered in system components affects dependability of the whole system and will require additional specific resilience techniques. Thus, dealing with such *uncertainty*, which is to an extent in the very nature of the Internet, Clouds, web services, IoT sensors and smart devices is the main challenge, thrown down by the ubiquitous computing systems. These systems should be capable of tolerating faults and potentially-harmful events caused by a variety of reasons including, low or changing (decreasing) quality of components (services), changing characteristics of the network media, component mismatches, permanent and temporary faults of individual services, composition mistakes, service disconnection, changes in the environment and in the policies.

Nowadays there is significant research activity devoted to achieving resilience of modern computing systems. Recent related works [27, 28, 29, 30]

have introduced principles, approaches, models and algorithms to build adaptive and resilient ubiquitous computing systems, services and SOA.

There have been works on incorporating resilience techniques into SOA [31, 32, 33] and Cloud computing architecture [34, 35, 36, 37, 38] for building dependable enterprise systems and mission critical applications, and on integration of Cloud Computing and the IoT technologies [39, 40].

In a series of recent works researchers discuss performance [41, 42] and security [43, 44] issues related to SOA and cloud computing and propose a dynamic fault tolerance models [45] ensuring system resilience.

But even though the existing proposals offer useful means for improving system dependability and resilience by proposing and enhancing particular technologies, most of them do not address the *uncertainty challenge* in addition to the lacking dependability characteristics and changing environment. Proposed techniques exploit the flexibility of the service infrastructure, but the major challenge in utilising these techniques is the *uncertainty* inherent in the services running over the Internet and Clouds.

Besides, hardly any of the studies offer strong mathematical foundation and proofs mostly because, we believe, *there is no general theory to capture uncertainties inherent to SOA, Cloud and Internet computing*.

Uncertainty of the Internet and instability of QoS characteristics (their performance, dependability and security) of system components are such that on-line optimization of redundancy, diversity and time-outs can make a substantial difference in perceived dependability, but currently there are no good tools available for the company to carry out such optimisation in a rigorous manner.

Thus, *uncertainty* needs to be treated as a threat in a way similar to and in addition to faults, errors and failures, traditionally dealt with by the dependability community [2].

24.3.2 Uncertainty Measurement and Quantification

A series of our recent experimental works [26, 46, 47, 48, 49, 25] supports the claim that dealing with the uncertainty inherent in the very nature of the Internet and Clouds is one of the main challenges in building dependable service-oriented systems of the Internet scale.

In particular, to illustrate the problem, our earlier extensive experiments with bioinformatics web services widely used in the DNA in-silico experiments, like BASIS and BLAST [46] show that the response time varies a lot because of various unpredictable factors like Internet congestions and failures, web services overloads, etc. In particular, the BASIS WS response time changes from 300 ms to 120000 ms, 22% of the requests have the response time at least twice larger than the observed minimal value and 3% of

requests have the response time more than 20 times larger. We believe it is impossible to build fast and dependable SOA without dealing with such phenomena.

Network instability as well as the internal instability of WS throughput significantly affects service response time. Because of network congestions and packet losses the response time could increase in an order. Accidental and sharp increase of response time commonly occurs due to short-term network congestions causing packet losses and multiple retransmissions. Besides, our experimental work showed that the Internet is also a subject to long-term congestions that can last hours and days. Thus, QoS of modern and emerging computing systems cannot be ensured without guaranteeing the underlying network QoS, especially in case of using the Internet as a communication medium to build the global and ubiquitous computing systems.

We can also state that the instability of the response time depends on the quality of the network connection used rather than on the length of the network route or number of the intermediate routers.

Because of the Internet, different clients have their own view on system performance and dependability. Each client has his own unique network route to the web service. However, it is likely that some parts of the route can be common for multiply clients. Thus, number of clients simultaneously suffering from the Internet instability depends on the point where network congestions or failures happen.

Retrieval of real distribution laws of system delays is an important for quantifying uncertainty. In [49] we benchmarked web service performance with the purpose to gather response time statistics and apply known distribution laws of random variables (e.g. Exponential, Gamma, Beta, Normal, Weibull or Poisson, etc.) to predict and quantify performance uncertainty. The paper describes the whole research methodology including the technique we used to test hypotheses that the system response time conforms to one of the theoretical distributions.

Our main finding was that the entire statistics gathered over more than four weeks cannot be described by any known theoretical distribution. The more experimental data we used, the worse approximation was provided by all studied distributions. A close approximation can be achieved only within the limited sample intervals (25-50 samples) with the coefficient of variation (CV) in between 5% and 20%.

Our work shows that long-tailed distributions like Beta, Weibull and Gamma fit the experimental data better than others. At the same time the Exponential distribution that typically used for networks simulation and response time analysis does not fit well the stochastic processes happening in such unstable environments as the Internet.

It seems that the research and engineering community needs a new exploratory theory and more complex assumptions to predict and simulate performance and dependability of distributed systems, using the Internet as a communication medium and Clouds as hosting environment.

24.3.3 Timing Failures

Instability of the response time can cause *timing failures* when the time of response arrival or the time during which information is delivered at the service interface (i.e., the timing of service delivery) deviates from the time required to execute the system function. A timing failure may be in the form of early or late response, depending on whether the service is delivered too early or too late [2].

In our experiments multiple clients invoked the same web services. However, different clients caught different number of errors and exceptions, but not all of them were caused by service unreliability. In fact, some clients were successfully serviced whereas others, at the same time, were faced with different problems.

Thus, for complex distributed systems composed of many different services some users may perceive a correct service whereas others may perceive incorrect services of different types due to timing errors or network failures.

Thus, timing errors can become a major cause of inconsistent failures usually referred to, after [50] as the Byzantine failures. These errors might occur in different system components depending on the relative position in the Internet of a particular user and a system (and various system components) they are interacting with, and, also, on the instability points appearing during the execution.

As a result, such systems or their components might be compromised by the client side or network failures, which, actually, are not related to dependability of those systems themselves. However, most of the time, the clients are not very interested in their exact cause. Besides, even if they are, they do not have sufficient mechanisms to disclosure the root causes of such failures.

From different client side perspectives the same Internet application usually has different availability and reliability characteristics. Objective data might be obtained by aggregation of clients' experience and/or by having internal access to the system operational statistics.

24.3.4 Resisting Uncertainty

The novel concepts of IoT, cyber-physical and ubiquitous systems and their application in mission-critical domains will clearly require continued attention to the uncertainty issues. For such systems using the Internet as a communication media and globally-distributed Clouds as platforms for hosting computing and data resources this uncertainty is unavoidable and the systems should be able to provide the trustworthy service in spite of it.

This, in turn, will require developing new resilience engineering techniques and resilience-explicit mechanisms dealing with this threat. The future solutions will need to deal with a number of issues such as uncertainty of fault assumptions, uncertainty of redundant resource behaviour, uncertainty of error detection, etc. The traditional adaptive solutions based on the control feedback will not be directly applicable in the current form as they are intended for predictable behaviour.

Uncertainty has two main consequences. First, it is difficult to assess the dependability and performance of services, and hence it is difficult to choose between them and gain confidence in their dependability. Secondly, it is difficult to execute fault tolerance mechanisms in a (close to) optimal manner, since too much data is missing to make good decisions and exploit all features of the dependability mechanisms.

We believe that uncertainty can be resolved by two means:

- (i) uncertainty removal through advances in data collection and analysis, and,
- (ii) uncertainty tolerance through smart algorithms that improve decisions despite lack of data (e.g., by extrapolation, better mathematical models, etc.).

Improving the prediction of system performance and latency needs more sophisticated procedures for experimental data processing (e.g. using dynamic time slots, rejecting some extreme samples, etc.) beforehand.

Moreover we believe that the more aware the user is about different delays contributing to response time and their uncertainty and also different factors affecting the overall dependability the more intelligent will be his/her choice. Thus, a user can intelligently and dynamically switch between the ISP, Cloud or service providers if she/he understands which delay makes the major contribution to the response time and its instability.

We also should notice that performance and dependability characteristics of Internet and Cloud systems and their components could become out of date very quickly. Once measured their non-functional characteristics cannot be assumed to be true forever. This is why, developing dynamic fault-tolerant techniques and mechanisms setting timeouts on-line and adopting system architecture and its behaviour on the fly are crucial.

Good measurement of uncertainty is important but this is only the beginning. More sophisticated fault-tolerant mechanisms have to be implemented at both, the client and the service sides. Emerging systems should uncover, tolerate and notify clients about potential accident factors at the server side to avoid client side failures.

In turn, a client should implement diagnostics mechanisms distinguishing internal, service side and network failures and using different recovery strategies to handle them in more adequate way. Thus, most of the errors caused by transient network failures might be effectively tolerated by simple retry.

These systems should be also robust to the accidental response time delays. Extremely high delays that happen from time to time could cause mistiming in a composite business workflows incorporating number of different services.

One of the possible solutions for resisting the uncertainty is to use service and path redundancy and diversity inherent to the Internet and Clouds. In [51, 52] we discuss several patterns for dependability-aware service composition allowing us to construct composite service-oriented systems resilient to various failures type (signalled or unsignalled; content, timing or silent failures) using inherent redundancy and diversity of Web Service components existing in the Internet.

24.4 Resilience Principles and Resilient System Architectures

24.4.1 Resilience Principles and Techniques

As ubiquitous systems are under continuous changes or evolutions and non-functional characteristics (e.g. probability of failure, latency, etc.) of their components are uncertain, a central property they should demonstrate is resilience.

Resilience assumes certain principles that a system has to exhibit via appropriate technologies [8]. These principles have a tight interconnection with dependability assurance techniques as shown in Fig. 21.1.

First of all, a system should be capable of *evolvability*, i.e., the ability to successfully accommodate changes. Ability to evolve while executing (i.e. *adaptivity*) is a crucial property for systems operating non-stop.

Assessability during operation is another important principle a resilient system should implement via run-time monitoring, measurement and assessment to demonstrate a justified confidence in its dependability and performance properties.

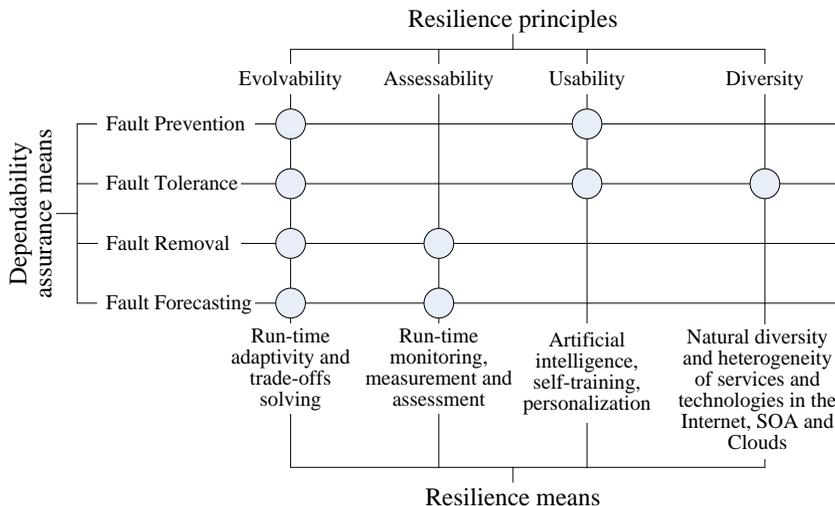


Fig. 24.1. Resilience and dependability principles and means

J.C. Laprie also highlighted the importance of the *usability* principle that pervasive computing systems should implement to prevent users from errors when they interact with such systems.

Finally, resilient systems should take advantage of *diversity* and *heterogeneity* which are naturally present in the Internet, SOA and Clouds in order to prevent system from single points of failures and intrusions.

Service-oriented architecture supports construction of the globally distributed massive-scale systems with growing number of services. This makes it unique in allowing access to a number of services with identical or similar functionalities, provided by different vendors and deployed on different platforms all over the Internet. In other words, SOA possesses the inherent redundancy and diversity of the existing web services [52].

We should use this fact to build dependable ubiquitous computing systems which are resilient to uncertainty of dependability and performance characteristics of web services and other components like IoS sensors, smart devices, etc.

It is obvious that systems developers (systems integrators) and end users should be able to choose and use the most dependable components from the existing ones of similar functionality but diverse nature [53]. However, our approach goes even further. We propose to use available diverse services together with the purpose to improve system resilience to their uncertain dependability and performance characteristics. In the next section we describe resilience patterns allowing to trade-off between system redundancy (which

often defines system cost), dependability and performance taking into account changeable dependability and performance characteristics of system components.

24.4.2 Resilient System Architecture

In [54] we proposed the service-oriented architecture which uses a dedicated middleware for a managed dependable upgrade of web services. That approach has been extended to compose alternative (diverse) services with the identical or similar functionality or replicas of the same service deployed on diverse platforms [51, 52, 55]. Such kind of redundancy based on inherent service diversity can improve dependability, performance and resilience of services composition.

The architecture includes a *mediator* component, which collect the responses from all diverse system components and returns an adjudicated response to a client. In the simplest case the *mediator* is a *voter* (i.e. performs majority voting using the responses from diverse services). It can be also programmed to perform more complex aggregation operation or to provide the best choice according to criterions specified by user (for example, chose the response with the latest time stamp like in NoSQL databases).

The mediator middleware intercepts user's request, relays it to all the diverse services and collects their responses (Fig. 24.2). Each service can return response of several types [52]:

1. Correct response returned before client's application timeout.
2. Evident erroneous response which often results in exception message reporting the problem occurred during service invocation. It can be also a response which value lies beyond the specified acceptance range. If such error occurs, user could retry the same service latter on (if it is suspected that the error was caused by temporal factors) or, most likely, invoke an alternative one.

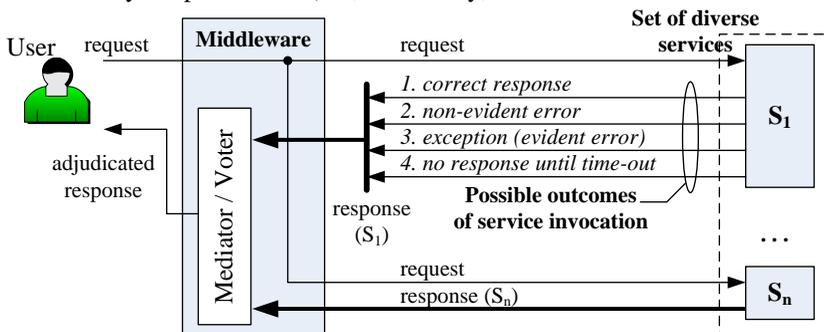


Fig. 24.2. Architecture of resilient services composition

3. Non-evident erroneous response. It can include calculation errors or incorrect data returned to a user. This type of error does not arise immediate exception. Detection of such errors is possible by comparing service response with responses from other diverse services.

4. Timing error (or silence) when service does not return response of any above types to a user until the specified timeout.

The architecture can support several composition patterns meeting different resilience objectives (such as ensuring service availability, responsiveness or trustworthiness), various strategies for invoking diverse services (sequential or simultaneous) and procedures for response adjudication. The basic ones are:

1. ALL (Fig. 24.3) – all available diverse services (or a specified number of them) are invoked concurrently and their responses are used by the mediator to produce an adjudicated response to the consumer (i.e. by voting, or selecting the most actual response with the latest timestamp).

2. FIRST (Fig. 24.4) – all available diverse services are invoked concurrently and the fastest response is returned to the service consumer.

3. QUORUM (Fig. 24.5) – all available diverse services are executed concurrently. The mediator is configured to wait for up to a quorum number of responses to be collected before the specified timeout. In the most general case the exact number of responses can be configured dynamically.

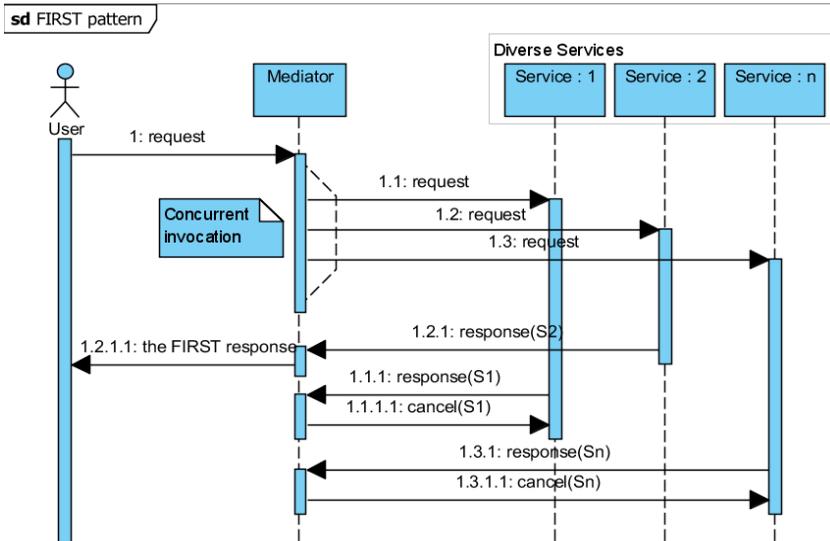


Fig. 24.3. Resilience pattern FIRST

sd ALL pattern

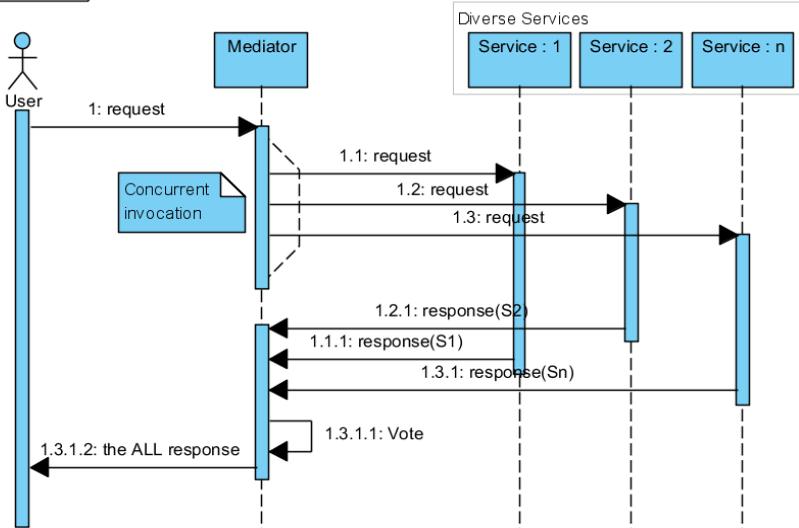


Fig. 24.4. Resilience pattern ALL

sd QORUM pattern

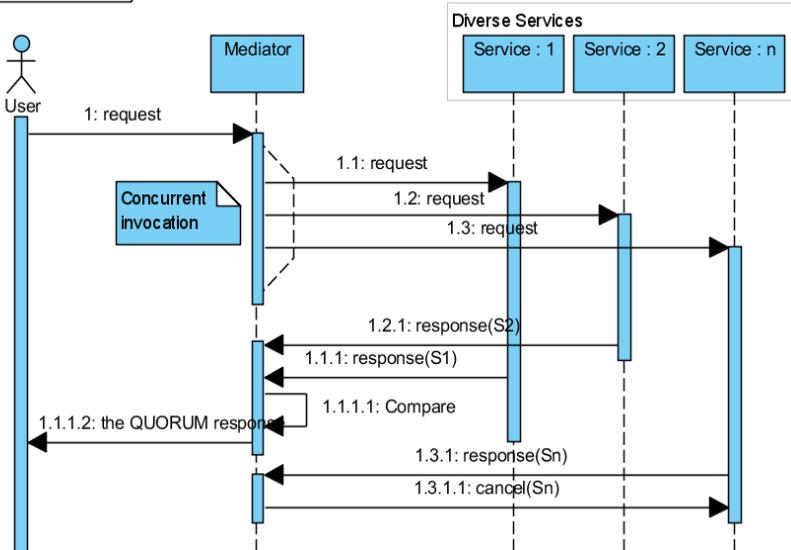


Fig. 24.5. Resilience pattern QORUM

4. **SEQUENCE** – the diverse services are invoked in a sequence (Fig. 24.6). The subsequent service is only invoked if the response received from the previous one is evidently incorrect (i.e. exception) or if there is no response has been received before the application timeout.

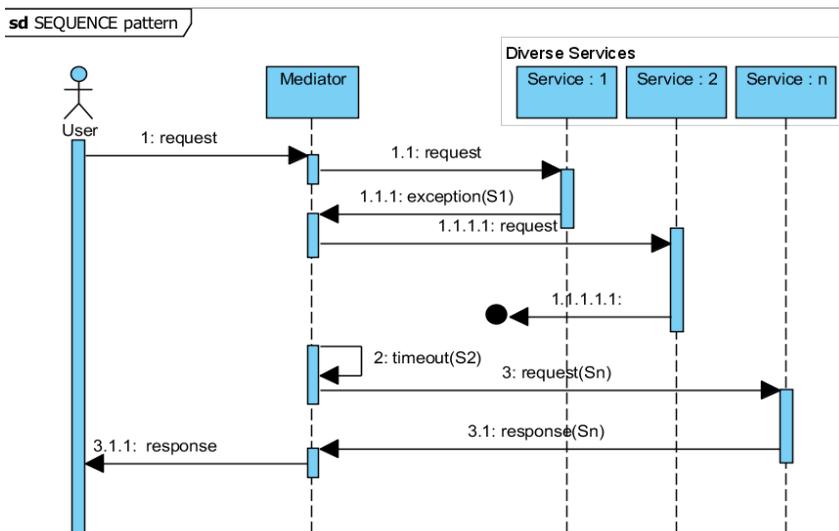


Fig. 24.6. Resilience pattern SEQUENCE

Our simulation and experimental results [52, 55] has shown that all resilience patterns significantly improve service availability and probability of correct response.

The QUORUM and ALL patterns also significantly minimize probability of hidden errors (but can slightly increase probability of exception) at the expense of some performance deterioration. However the QUORUM pattern provides better trustworthiness-to-response-time ratio.

The FIRST pattern significantly improves system responsiveness whereas the SEQUENCE pattern improves dependability characteristics without performing redundant service invocations.

Table 24.1 ranges patterns preference based on their effect on particular dependability and performance property. Ability to switch between patterns and to adapt their parameters (e.g. the total number of invoked diverse services, timeout settings, etc.) support resilience of service-oriented ubiquitous computing systems. Proposed patterns employ engineers with the ability to trade-off between system redundancy (which often defines system cost), dependability and performance.

Table 24.1. Qualitative effectiveness of different resilience patterns

Pattern	Redundancy increase	Trustworthiness enhancement		Availability enhancement		Performance enhancement
		increasing probability of correct response	decreasing probability of hidden error	decreasing probability of exception	decreasing probability of timeout	decreasing response time
ALL	II	I	I	II	I	IV (significant worseness)
FIRST	II	II	II	I	I	I (significant enhancement)
QUORUM	II	I	I	II	I	II (worseness)
SEQUENCE	I	II	II	I	II	III (worseness)

A description of the proposed resilience patterns can be formalized and generalized by introducing a triad (M:N:R), where M – is the total number of diverse system components (e.g. services); N – is the number of diverse system components invoked simultaneously by a mediator; R – is a number of responses from diverse components awaited by a mediator before it return system response to a user.

Thus, the ALL, FIRST, QUORUM and SEQUENCE pattern can be described respectively by the triads (M:M:M), (M:M:1), (M:M:⌊M+1⌋/2) and (M:1:1).

The additional parameter T shall specify the application timeout which limit the time during which mediator waits for responses from the invoked system components.

24.5 Next Generation *Antifragile* Computing Systems

Adaptiveness (adaptability, adaptivity) means that the system is capable to react to observed or act upon expected temporary changes of the system itself, the context/environment (e.g., resource variability or failure scenarios) or users needs and expectations (e.g., responsiveness) [56]. Adaptiveness can be

provided without explicit user involvement. In this case it is termed autonomous behaviour or self-properties, and often involves monitoring, diagnosis, and reconfiguration.

Antifragility is a new concept popularized by N.N. Taleb which he coined in his 2012 book [57] as the next step after resilience. *Antifragility* refers to computing systems and ICT infrastructures that benefit from some form of disorder and is fundamentally different from the concepts of resiliency and robustness [58, 59]. The *resilient system* resists unforeseen disturbances by adapting its functions and structure, though it still stays the same system preserving its identity (i.e. the set of functional and non-functional properties characterizing the system given the specifications of that system).

In contrast, the *antifragile system* self-evolves and gets better. It autonomously adapts own functions, structure, and identity, in order to systematically improve its system-environment fit [60, 61, 62]. Antifragile engineering is a challenge that, once met, would allow systems to [63]:

- (i) self-evolve and self-improve by learning from error, faults and failures;
- (ii) meta-adapt to changing circumstances;
- (iii) self-adjust to dynamically changing environments;
- (iv) self-organize to track dynamically and proactively optimal strategies to provide scalability, high-performance, energy efficiency;
- (v) personalize their aspects and behaviours to meet particular needs of every user.

Conclusion and Questions for the Self-Control

In this work we discuss further development of the dependability concept and its transformation into the concept of resilient ubiquitous computing systems. This transformation is caused by the fact that new generation of computing systems are continuously evolving. They are subject to requirements changes, changes of internal system structure and components characteristics. Moreover, ubiquitous systems have to cope with the continuous changes of the environment and operational conditions in addition to errors, faults and failures traditionally as dealt with by the dependability community.

In general, the notion of resilience can be defined as *persistence of dependability when facing changes* of different nature as described in Section 24.2.

We believe that one of the most crucial factors stimulating the development of resilient systems is *uncertainty* issue. In this work we use the general synthetic term *uncertainty* to refer to the unknown, unstable, unpredictable, changeable dependability and performance characteristics and behaviour of modern ubiquitous systems, exacerbated by running them over the Internet as a composition of third-party components (i.e. web services,

virtual computing instances, data storages, IoT sensors and Smart devices, etc.).

This work is aimed to answer a series of related questions, including:

1. What is a difference between dependability and resilience notions and how do these notions are interconnected?
2. What are ubiquitous computing systems?
3. What are main factors stimulating development of the resilience concept?
4. What changes the notion of resilience has been developed to cope with?
5. What is uncertainty issue? How does uncertainty exhibit itself in computing systems running over the Internet and Clouds?
6. How it is possible to measure and quantify uncertainty?
7. Which approaches can be used to resist uncertainty?
8. What are main resilience principles and technologies to achieve resilience?
9. Which architectural solutions can be employed to support resilience of service-oriented ubiquitous computing systems?
10. What are differences between concepts of *resilient* and *antifragile* computing systems?

References

- [1] R. Buyya, J. Broberg and A. Goscinski, *Cloud Computing: Principles and Paradigms*, John Wiley & Sons, 2011.
- [2] A. Avizienis, J. Laprie, B. Randell and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11-33, 2004.
- [3] P. Dearnley, "An investigation into database resilience," *The Computer Journal*, vol. 19, no. 2, pp. 117-121, 1976.
- [4] L. Svobodova, "Resilient distributed computing," *IEEE Transactions on Software Engineering*, vol. 10, no. 3, pp. 257-268, 1984.
- [5] T. Anderson, Ed., *Resilient Computing Systems*, Collins, 1985.
- [6] T. Anderson, Ed., *Dependability of Resilient Computers*, Blackwell Science Inc., 1989.
- [7] J. Laprie, "Resilience for the Scalability of Dependability," in *4th IEEE International Symposium on Network Computing and Applications (NCA'05)*, 2005.
- [8] J. Laprie, "From dependability to resilience," in *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'08)*, 2008.

- [9] L. Simoncini, "Resilient computing: An engineering discipline," in *International Parallel and Distributed Processing Symposium*, 2009.
- [10] K. Trivedi, D. Kim and R. Ghosh, "Resilience in computer systems and networks," in *International Conference on Computer-Aided Design*, 2009.
- [11] L. Strigini, "Resilience: What Is It, and How Much Do We Want?," *IEEE Security and Privacy Magazine*, vol. 10, no. 3, pp. 72-75, 2012.
- [12] L. Strigini, "Fault Tolerance and Resilience: Meanings, Measures and Assessment," in *Resilience Assessment and Evaluation of Computing Systems*, K. Wolter, A. Avritzer, M. Vieira and A. van Moorsel, Eds., Berlin, Springer-Verlag, 2012, pp. 3-24.
- [13] N. Guelfi, "A Formal Framework for Dependability and Resilience from a Software Engineering Perspective," *Central European Journal of Computer Science*, vol. 1, no. 3, pp. 294-328, 2011.
- [14] L. Lúcio and N. Guelfi, "A precise definition of operational resilience," Luxembourg, 2011.
- [15] E. Hollnagel, D. Woods and N. Leveson, Eds., *Resilience Engineering, Concepts And Precepts*, Ashgate Publishing, 2006, p. 414.
- [16] E. Hollnagel, C. Nemeth and S. Dekker, Eds., *Resilience Engineering Perspectives*, vol. Vol. 1. Remaining Sensitive to the Possibility of Failure, Ashgate Publishing, 2008.
- [17] A. Masys, Ed., *Disaster Management: Enabling Resilience*, Springer, 2015, p. 338.
- [18] P. Longstaff, T. Kowalski and W. Geoghegan, "Translating Resilience: A Framework to Enhance Communication and Implementation," in *5th Symposium On Resilience Engineering*, Soesterberg (Netherlands), 2013.
- [19] C. Queiroz, S. Garg and Z. Tari, "A probabilistic model for quantifying the resilience of networked systems," *IBM Journal of Research and Development*, vol. 57, no. 5, pp. 3:1-3:9, 2013.
- [20] J. Sterbenz, E. Cetinkaya, M. Hameed, A. Jabbar, S. Qian and J. Rohrer, "Evaluation of network resilience, survivability, and disruption tolerance: Analysis, topology generation, simulation, and experimentation," *Telecommunication Systems*, vol. 52, no. 2, pp. 705-736, 2013.
- [21] J. Sterbenz, D. Hutchison, E. Çetinkaya, A. Jabbar, J. Rohrer, M. Schöller and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *The International Journal of Computer and Telecommunications Networking*, vol. 54, no. 8, pp. 1245-1265, 2010.
- [22] D. Rosenkrantz, S. Goel, S. Ravi and J. Gangolly, "Resilience Metrics for

- Service-Oriented Networks: A Service Allocation Approach," *IEEE Transactions on Services Computing*, vol. 2, no. 3, pp. 183-196, 2009.
- [23] R. Ghosh, F. Longo, V. Naik and K. Trivedi, "Quantifying Resiliency of IaaS Cloud," in *29th IEEE Symposium on Reliable Distributed Systems*, 2010.
- [24] S. Kounev, P. Reinecke, F. Brosig, J. Bradley, K. Joshi, V. Babka, A. Stefanek and S. Gilmore, "Providing Dependability and Resilience in the Cloud: Challenges and Opportunities," in *Resilience Assessment and Evaluation of Computing Systems*, K. Wolter, A. Avritzer, M. Vieira and A. van Moorsel, Eds., Springer, 2012, pp. 65-81.
- [25] A. Gorbenko, V. Kharchenko, O. Tarasyuk, A. Romanovsky and Y. Chen, "The Threat of Uncertainty in Service-Oriented Architecture," in *RISE/EFTS Joint International Workshop on Software Engineering for Resilient Systems*, Newcastle-upon-Tyne, UK, 2008.
- [26] A. Gorbenko, Y. Chen, A. Romanovsky and V. Kharchenko, "Measuring and Dealing with the Uncertainty of the SOA Solutions," in *Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions.*, V. Cardellini, E. Casalicchio and K. Castelo Branco, Eds., Hershey, IGI Global, 2011, p. 265–294.
- [27] V. De Florio, Ed., *Technological innovations in adaptive and dependable systems: advancing models and concepts*, Hershey: IGI Global, 2012.
- [28] V. De Florio, Ed., *Innovations and Approaches for Resilient and Adaptive Systems*, Hershey: Information Science Reference, 2013.
- [29] N. C. G. Suri, Ed., *Adaptive, Dynamic, and Resilient Systems*, Boca Raton: CRC press, 2014.
- [30] B. Christensen, "Application Resilience in a Service-oriented Architecture," 2013. [Online]. Available: <http://radar.oreilly.com/2013/06/application-resilience-in-a-service-oriented-architecture.html> .
- [31] O. Erol, M. Mansouri and B. Sauser, "A framework for enterprise resilience using service oriented architecture approach," in *3rd Annual IEEE Systems Conference*, 2009.
- [32] M. Hall-May, M. SurrIDGE and R. Nossal-Ruyeni, "Resilient critical infrastructure management with a service oriented architecture: a test case using airport collaborative decision making," *International Journal of Applied Mathematics and Computer Science*, vol. 21, no. 2, p. 259–274, 2011.
- [33] P. den Hamer and T. Skramstad, "Autonomic Service-Oriented Architecture for Resilient Complex Systems," in *IEEE Symposium on*

Reliable Distributed Systems Workshops, 2011.

- [34] R. Ravishankar, "Cloud architecture for a highly resilient, always-on system," 2014. [Online]. Available: <https://www.ibm.com/blogs/cloud-computing/2014/04/cloud-architecture-highly-resilient-always-system/>.
- [35] O. Diez and A. Silva, "Resilience of Cloud Computing in Critical Systems," *Quality and Reliability Engineering International*, vol. 30, no. 3, p. 397–412, 2014.
- [36] S. Kounev, P. Reinecke, F. Brosig, J. Bradley, K. Joshi, V. Babka, A. Stefanek and S. Gilmore, "Providing Dependability and Resilience in the Cloud: Challenges and Opportunities," in *Resilience Assessment and Evaluation of Computing Systems*, K. Wolter, A. Avritzer, M. Vieira and A. van Moorsel, Eds., Berlin, Springer-Verlag, 2012, pp. 65-81.
- [37] T. Erl, R. Cope and A. Naserpour, "Reliability, Resiliency and Recovery Design Patterns in Cloud Computing," in *Cloud Computing Design Patterns*, Old Tappan, Prentice Hall, 2015, p. 97–166.
- [38] R. Jhawar and V. Piuri, "Fault Tolerance and Resilience in Cloud Computing Environments," in *Computer and information security handbook; 2nd Edition*, J. Vacca, Ed., Waltham, Morgan Kaufmann, 2012, pp. 125-142.
- [39] A. Botta, W. de Donato, V. Persico and A. Pescape, "On the Integration of Cloud Computing and Internet of Things," in *International Conference on Future Internet of Things and Cloud*, 2014.
- [40] A. Botta, W. de Donato, V. Persico and A. Pescape, "Integration of Cloud computing and Internet of Things: A survey," *Future Generation Computer Systems*, vol. 56, p. 684–700, 2016.
- [41] S. Paliwa, "Paliwal Performance Challenges in Cloud Computing," *CMG MeasureIT*, vol. 14, no. 1, 2014.
- [42] N. Zanoon, "Toward Cloud Computing: Security And Performance," *International Journal on Cloud Computing: Services and Architecture*, vol. 5, no. 5/6, pp. 17-26, 2015.
- [43] Q. Nguye and A. Sood, "Building a Resilient Service-Oriented Architecture Environment," *CrossTalk*, vol. 9/10, pp. 27-31, 2013.
- [44] N. Shear, P. Cable, R. Cunningham, V. Gadepally, T. Moyer and A. Yerukhimovich, "Secure and Resilient Cloud Computing for the Department of Defense," *Lincoln Laboratory Journal*, vol. 22, no. 1, pp. 123-135, 2016.
- [45] A. Meshram, A. Sambare and S. Zade, "Fault Tolerance Model for Reliable Cloud Computing," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 1, no. 7, p.

- 600–603, 2013.
- [46] Y. R. A. G. A. K. V. M. S. T. O. Chen, "Benchmarking Dependability of a System Biology Application," in *14th IEEE Int. Conference on Engineering of Complex Computer Systems*, 2009.
- [47] A. Gorbenko, V. Kharchenko, A. Romanovsky and A. Mikhaylichenko, "Experimenting with exception propagation mechanisms in service-oriented architecture," in *4th Int. Workshop on Exception Handling*, 2008.
- [48] A. Gorbenko, A. Romanovsky, V. Kharchenko, S. Mamutov and O. Tarasyuk, "Exploring Uncertainty of Delays as a Factor in End-to-End Cloud Response Time," in *9th European Dependable Computing Conference*, 2012.
- [49] A. Gorbenko, V. Kharchenko, S. Mamutov, O. Tarasyuk, Y. Chen and A. Romanovsky, "Real Distribution of Response Time Instability in Service-Oriented Architecture," in *IEEE International Symposium on Reliable Distributed Systems*, 2010.
- [50] L. S. R. & P. M. Lamport, "The Byzantine Generals Problem," *ACM Trans. Programming Languages and Systems*, vol. 4, no. 3, pp. 382-401, 1982.
- [51] A. Gorbenko, V. Kharchenko and A. Romanovsky, "Vertical and Horizontal Composition in Service-Oriented Architecture," in *Int. Workshop on Methods, Models and Tools for Fault Tolerance*, Oxford, 2007.
- [52] A. Gorbenko, V. Kharchenko and A. Romanovsky, "Using Inherent Service Redundancy and Diversity to Ensure Web Services Dependability," in *Methods, Models and Tools for Fault Tolerance: LNCS 5454*, Berlin; Heidelberg, Springer-Verlag, 2009, p. 324–341.
- [53] Y. Wang and J. Vassileva, "Toward Trust and Reputation Based Web Service Selection: A Survey," *International Transactions on Systems Science and Applications Journal. Special Issue on New tendencies on Web Services and Multi-agent Systems*, vol. 3, no. 2, 2007.
- [54] A. Gorbenko, V. Kharchenko, P. Popov and A. Romanovsky, "Dependable Composite Web Services with Components Upgraded Online," in *Architecting Dependable Systems III, LNCS 3549*, R. de Lemos, C. Gacek and A. Romanovsky, Eds., Berlin, Heidelberg, Springer-Verlag, 2005, p. 92–121.
- [55] O. Tarasyuk, A. Gorbenko, A. Romanovsky, V. Kharchenko and V. Ruban, "The Impact of Consistency on System Latency in Fault Tolerant Internet Computing," in *Distributed Applications and Interoperable Systems, LNCS 9038*, A. Bessani and S. Bouchenak, Eds., Berlin;

- Heidelberg, Springer-Verlag, 2015, pp. 179-192.
- [56] K. Göschka and L. Frohofer, "Closing the Dependability Gap: Converging Software Engineering with Middleware," in *38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2008.
- [57] N. Taleb, *Antifragile: Things That Gain from Disorder*, Random House, 2012, p. 430.
- [58] A. Abid, M. Khemakhem, S. Marzouk, M. Jemaa, T. Monteil and K. Drira, "Toward Antifragile Cloud Computing Infrastructures," *Procedia Computer Science*, vol. 32, pp. 850-855, 2014.
- [59] M. Monperrus, "Principles of Antifragile Software," arXiv:1404.3056, 2014.
- [60] K. H. Jones, "Engineering Antifragile Systems: A Change In Design Philosophy," *Procedia Computer Science*, vol. 32, pp. 870-875, 2014.
- [61] K. Jones, "Antifragile Systems: An Enabler for System Engineering of Elegant Systems," in *2nd International Workshop on Computational Antifragility and Antifragile Engineering*, 2015.
- [62] J. Kephart and D. Chess, "The Vision of Autonomic Computing," *Computer*, , vol. 36, no. 1, pp. 41-50, 2003.
- [63] B. Cheng, R. de Lemos, H. Giese, P. Inverardi and J. Magee, Eds., *Software Engineering for Self-Adaptive Systems. Lecture Notes in Computer Science*, vol. 5525, Springer, 2009.

CHAPTER 25 Vulnerability Study of Computer Systems

25.1 Security and Vulnerability of Multilevel Computing Architectures

Today, security of information and communication systems has become one of the most crucial concerns for both system developers and users. Recent security accidents like those happened to Hollywood Presbyterian Medical Center [1] or San Francisco Municipal Transportation Agency [2] show how vulnerable to attacks our modern society is. They may cost millions of US dollars and even affect human lives.

One of the main reasons of successful attacks, malicious intrusions and virus infections are software vulnerabilities in computer systems, communication equipment, smartphones and other intellectual devices.

Generally speaking, vulnerability is a weakness which allows an intruder to undermine system's information assurance. MITRE Corporation defines vulnerability as a software fault that can be directly used by a hacker to gain access to a system or network [3]. Exploiting vulnerability allows attackers to either execute commands as normal users, or access data violating the specified access restrictions or cause denial of service attack and terminate system services.

Software vulnerabilities are mainly caused by errors and weaknesses in software design and implementation.

A typical computer system consists of hardware and a multitier system architecture playing the role of a deployment environment for the specific application software. For instance, a web application can be created as a set of servlets and server pages, java beans, stored database procedures and triggers running on the top of the software stack in a specific deployment environment. This environment is constructed from a number of software components. Typical examples of system components for web services are operating system (OS), web and application servers (AS and WS), and data base management systems (DBMS).

Vulnerabilities can be discovered in both application software and its deployment environment represented by an operating system and other system-level components.

For instance, CVE-2016-7205 vulnerability refers to a weakness in the Windows family of operating systems when the font library improperly handles specially crafted embedded OpenType fonts [5, 6, 7]. An attacker can successfully exploit this vulnerability through a web-based or file sharing attack scenarios. As a result, a full control upon the affected system can be taken allowing hackers to install programs, view, change or delete data, create new user accounts with administrative rights, etc.

Another example is the CVE-2014-0160 vulnerability [8] in OpenSSL cryptography library that has affected half a million widely trusted websites and services including Yahoo, Amazon Web Services, GitHub, Wikipedia, etc. Existed since December, 2012 and disclosed only in April, 2014 it allowed remote attackers to obtain sensitive information from process memory and even to compromise server secret key via crafted packets that trigger a buffer over-read.

There is no doubt that vulnerabilities in operating systems are one of the most critical security threats as their exploitation can compromise all processes and services running in the operating system and allow attackers to gain access to all data stored on the vulnerable computer. Moreover, vulnerabilities of operating systems and various system components usually are more numerous than vulnerabilities discovered in application software running on the top of it. These vulnerabilities represent threats to system security and dependability that are additional to faults, errors and failures traditionally dealt with by the dependability community.

25.2 Software Vulnerability Lifecycle

Software vulnerability lifecycle has been discussed in a number of research papers [10, 11, 18]. The authors of [19] propose a formal model of the vulnerability lifecycle defining its milestones. Most of the researchers and security analysts mark out 5 main events in a typical vulnerability lifecycle:

- (i) vulnerability creation;
- (ii) vulnerability discovery;
- (iii) vulnerability disclosure;
- (iv) patch availability;
- (v) patch installation.

Exploits or computer viruses can become available during vulnerability lifecycle taking advantage of the particular vulnerability. An exploit is a sequence of commands, a software tool or even a specially generated data (e.g. an infected file) which automates making use of a vulnerability and allows even unskilled users to attack computer systems. Thus, exploit availability is an additional event sliding in between events (i) and (v).

Time intervals between the above mentioned events in the vulnerability life cycle have different risks of system exposure associated with them.

In particular, a special term *days-of-risk* [18] is used to define a period of an increased security risk between the time when a vulnerability is discovered or publicly disclosed to the time when a patch is applied to fix it. Usually the periods of *black*, *gray* and *white risk* are marked out to indicate public awareness of the hazard and to qualify relative risks of exposure (see Fig. 25.1).

Risk value depends on vulnerability severity and other factors. It dramatically increases when an exploit is released in the wild and comes down when software vendor issues a patch to fix vulnerability. Application of anti-virus software, firewalls, intrusion detection systems mitigate risk value.

How public vulnerability disclosure (e.g. through the CVE or NVD databases) affects system security is a question of great debates. On the one hand, malicious hackers can use publicly available information to attack affected computer systems increasing the risk of exposure.

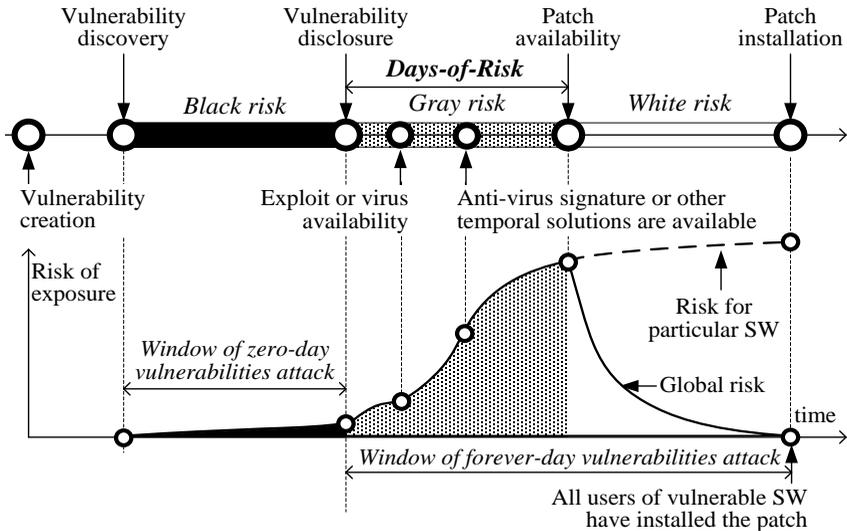


Fig. 25.1. Vulnerability lifecycle and risk of exposure

On the other hand, forewarned users of vulnerable systems are forewarned, so they can take additional prevention actions to mitigate risk of exposure. Besides, public awareness of vulnerability usually pressures vendors to find a fix urgently.

In the paper we investigate *gray risk* (or *post-disclosure risk*) which defines the interval between vulnerability disclosure time and the date when the patch fixing vulnerability becomes available.

25.3 Vulnerability Databases

Nowadays there are a lot of institutions focusing their activity on vulnerability discovery and elimination. They include, undoubtedly, software vendors as well as a number of governmental and independent international

organizations, commercial enterprises and even individuals. Most of these institutions provide publically available vulnerability datasets. The most known and trusted are:

- CVE – Common Vulnerabilities and Exposures database provided by not-for-profit MITRE Inc. (cve.mitre.org). MITRE maintains a list of known vulnerabilities and performs their enumeration by assigning CVE-IDs which are used by many others vulnerability databases to synchronize with CVE and enable data exchange between security databases and products. In 2016 MITRE has assigned more than 9900 vulnerability identifiers.

- NVD is the National Vulnerability Database provided by U.S. National Institute of Standards and Technology (web.nvd.nist.gov). NVD offers a dataset of software security vulnerabilities which is based upon and synchronized with the CVE database. It classifies vulnerability severity and type and also specifies vulnerable software and provides additional meta-data using the Common Vulnerability Scoring System (CVSS), the Common Weakness Enumeration Specification (CWE) and the Common Platform Enumeration Dictionary (CPE). It has reported almost 6000 vulnerabilities disclosed in 2016 which is 16.5 vulnerabilities per day in average. About half of them have been observed in operating systems.

- VNDB is the Vulnerability Notes Database provided by CERT (www.kb.cert.org/vuls/). Most of VNDB entries are covered by CVE and NVD. Though, it have been mentioned that CERT VNDB vulnerability disclosures appear on its website 24–72 hours before they appear in CVE or NVD [14].

- VulnDB is the Risk Based Security’s vulnerability database (www.riskbasedsecurity.com/vulnbd/). VulnDB tracks vulnerabilities in third-party libraries and pretends to provide over 47,000 vulnerabilities that are not found in CVE or NVD. Though, its commercial offering prevents VulnDB from been widely used by researches.

- SecurityTracker is another vulnerability dataset commercially available at securitytracker.com. It almost entirely covers vulnerability entries that have CVE IDs.

Besides, software product vendors often provide information about vulnerabilities in their products as through the security bulletins (e.g. <https://technet.microsoft.com/en-us/security/bulletins.aspx>). At the same time, OSVDB (Open Source Vulnerability Database) and FVDB (Frei’s Vulnerability Database) that recently have been actively used by many researchers seem to be no longer available.

The CVE and NVD databases offer access to vulnerability data sets through the simple search interface available on their web sites or by distributing XML data feeds. Unfortunately, they do not support SQL querying making difficult a direct use of the CVE and NVD repositories for complex

analytics. In our study we have merged together XML data files provided by the CVE and NVD databases and inserted the joint data set into MySQL database. We use CVE-ID as a primary key to uniquely identify particular vulnerability.

CPE identifiers provided by NVD are used to assign particular vulnerability to a certain product out of the three main groups: (i) operating systems, (ii) application software and (iii) hardware components (e.g. routers, graphical cards, embedded devices, etc.). Besides, we store two dates associated with the same vulnerability: when it firstly appeared in the CVE database and when its description was published by NVD.

25.5 Research Methodology

In the work we examine vulnerabilities of 6 popular enterprise operating systems (see Table. 25.1), investigate statistics of vulnerability disclosure and elimination and analyze their criticality.

In contrast to other works investigating software vulnerabilities [9, 10, 11] we focus on examining how vulnerability disclosure rates have being changed over last years for particular operating systems and study how much time OS vendors spend on issuing patches to fix that security flaws and how many yet unfixed vulnerabilities can exist in each OS simultaneously.

Besides, we update research results reported for the earlier versions of some of those OSs by other authors in 2000 [15] and 2006 [12, 16, 17].

Table 25.1. Qualitative effectiveness of different resilience patterns

Operating system	Release date	Linux kernel version
Ubuntu Server 12.04	26.04.2012	3.2.x
Red Hat Enterprise Linux 6	10.11.2010	2.6.32
Novell Linux SUSE Enterprise Server 11 SP2	27.02.2012	3.0.x
Microsoft Windows Server 2012 R2	18.10.2012	-
Apple MacOS Server 10.8	25.06.2012	-
Oracle Sun Solaris 11	09.11.2011	-

Thus, one of our intentions was to analyse how their security and vulnerability have been changed since that time.

Ubuntu Server, Red Hat Enterprise Linux and Novell Linux Enterprise Server are non-monolithic operating systems, thus in our study we also considered vulnerabilities in Linux kernels they use.

In this work we analyse four aspects of operating system vulnerability.

- 1) quantitative analysis and statistics comparison of disclosed and fixed vulnerabilities for different operating systems;
- 2) assessment and analyzing number of days-of-risk for each operating system;
- 3) comparison of vulnerabilities severity and analyzing the most frequent numerous vulnerabilities discovered in various OSs;
- 4) discovery of vulnerabilities that are common for two or more operating systems.

We assume vulnerability disclosure time as a date when a vulnerability firstly appears in the CVE database with the corresponding CVE-ID assigned to it. A time of patch issuing should be derived from vendor's security bulletins. It is noteworthy that, according to our study, vulnerability description usually becomes available in the NVD database the same day or the day after it is mentioned in the vendor security bulletin.

It means that NIST implements so called *responsible disclosure model* by giving stakeholders a time for the vulnerability to be patched before publishing the details in the NVD database. Thus, days-of-risk for a particular vulnerability can be considered as a period of time between a vulnerability is firstly reported in the CVE database until it appears in the NVD database.

25.6 Vulnerability Study of Enterprise Operating Systems

25.6.1 Statistics of Vulnerability Discovery and Elimination

In this section we summaries statistics of vulnerabilities discovered and disclosed in different operating systems since the 1st of January, 2012 and until the 31st of December, 2015 (see Table 25.2). A number of vulnerabilities that had been observed but not fixed by the 1st of January, 2012 are reported as 'Starting'.

In the table we use the following short pseudonyms for operating systems under investigation:

- Ubuntu – Ubuntu Server 12.04;
- RedHat – Red Hat Enterprise Linux 6;
- Novell – Novell Linux Enterprise Server 11 SP2;
- Windows – Microsoft Windows Server 2012 R2;
- MacOS – Apple Macintosh Server 10.8;
- Solaris – Oracle Solaris 11.

Operating systems Red Hat Enterprise Linux 6 and Oracle Solaris 11 had been released before the observed period (see Table 25.1). Other operating systems (Ubuntu Server 12.04, Novell Linux Enterprise server 11 SP2, Microsoft Windows Server 2012 R2 and Apple Macintosh Server 10.8) were released in the beginning of 2012.

Table 25.2. Operating Systems Vulnerability Statistics

Year	Vulnerabilities by groups	Operating System					
		Ubuntu	Windows	RedHat	Novell	MacOS	Solaris
Initial number		14	0	45	26	0	9
2012	Disclosed	58	10	27	31	2	47
	Fixed	28	5	37	35	2	47
	Avg.Sev.	5.11	8.31	4.87	5.16	3.20	4.37
	Avg.DoR	146	132	262	112	94	89
2013	Disclosed	183	59	63	121	59	30
	Fixed	190	51	83	124	58	31
	Avg.Sev.	5.01	7.08	5.05	4.96	4.93	4.73
	Avg.DoR	111	130	122	101	110	75
2014	Disclosed	126	64	26	90	40	32
	Fixed	152	38	33	103	40	26
	Avg.Sev.	5.37	7.25	6.14	5.27	7.85	5.03
	Avg.DoR	55	91	88	51	89	75
2015	Disclosed	141	136	26	32	13	36
	Fixed	147	156	34	37	14	34
	Avg.Sev.	6.18	7.17	5.63	6.09	8.52	4.44
	Avg.DoR	57	101	67	71	47	92
Total	Disclosed	522	269	187	300	114	154
	Fixed	517	250	187	299	114	138
	Avg.Sev.	5.42	7.45	5.42	5.37	6.13	4.64
	Avg.DoR	92	113	135	84	85	83

It is worth mentioning that on the date of the official release some of those operating systems already had vulnerabilities that earlier had been discovered in previous OS versions. In particular, Ubuntu Server 12.04 inherited 25 of such vulnerabilities, Microsoft Windows Server 2012 R2 – 5, Novell Linux Enterprise server 11 SP2 – 30 and Oracle Solaris 11 – 9 vulnerabilities.

Table 25.2 presents the average vulnerability severity level (Avg.Sev.) by aggregating CVSS (Common Vulnerability Scoring System) metrics taken from NVD vulnerability database as well as reports average *days-of-risk* value (Avg.DoR).

During 2012-2015 the largest number of vulnerabilities (522) was disclosed in Ubuntu Server 12.04, the least number (114) – in Apple Macintosh Server 10.8. Novell Linux Enterprise server 11 SP2 and Windows Server 2012 R2 occupy a middle position having 269 and 300 vulnerabilities, respectively. A cumulative graph of disclosed vulnerabilities is depicted in Fig. 25.2.

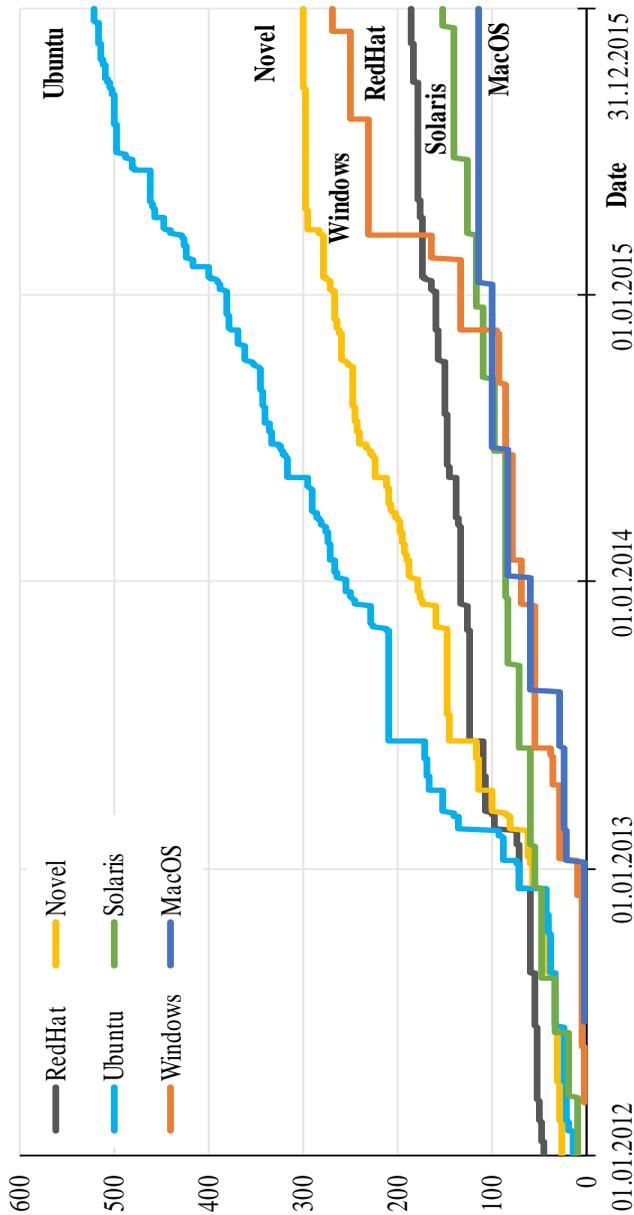


Fig. 25.2. Cumulative number of disclosed vulnerabilities

Using both, the date of vulnerability disclosure and the date when the OS vendor issues a patch to fix it we can plot graphs of *forever-day vulnerabilities* showing how many of known (already disclosed publicly) but yet unfixed vulnerabilities existed every day during 2012-2015 in particular operating system (see Fig. 25.3).

The article [20] coins a new term '*forever-day vulnerability*' defining publicly disclosed vulnerabilities that has not been patched yet and can be hacked. It is in contrast to '*zero-day vulnerabilities*' [19] which are still publically undisclosed vulnerabilities that some hackers have already discovered and can exploit.

Any operating system with forever-day vulnerabilities is always vulnerable unless the software vendor issues a patch and a system administrator installs it.

As far as vulnerability disclosure rate significantly overtop the rate of vulnerability elimination it can happen that an operating system contains up to several dozens of forever-day vulnerabilities at a time. Any of these vulnerabilities could be potentially exploited by hackers to attack the system.

Fig. 25.3 shows that various operating systems have only few days (if any) of vulnerability free operation.

During 2012-2015 OS RedHat has had only 11 of such days while MacOS – 159. All the rest operating systems had not had vulnerability free days at all! It means that OS users and administrators should understand and accept potential risk of running vulnerable system.

In addition, Table 25.3 presents a detailed statistics of forever-day vulnerabilities for each operating system. In average, Ubuntu OS had 33 of such vulnerabilities every day. For OS Windows, RedHat and Novell this number is close to 20 vulnerabilities. MacOS and Solaris had the least average number of forever-day vulnerabilities (10).

25.6.2 Operating Systems Days-of-Risk

Number of disclosed vulnerabilities is often used as the major indicator of software insecurity. However, taking into account how fast software vendors react on vulnerabilities discovered in their products is also important.

Days-of-risk defines a period of time after a vulnerability is discovered/disclosed and until it is eliminated from a system after patch installation. It is also known as '*window of-vulnerability*' or '*days-of-recess*'. However, in this study we do not take into account possible delays between the times when a vendor issues the patch and until a user or a system administrator actually installs it.

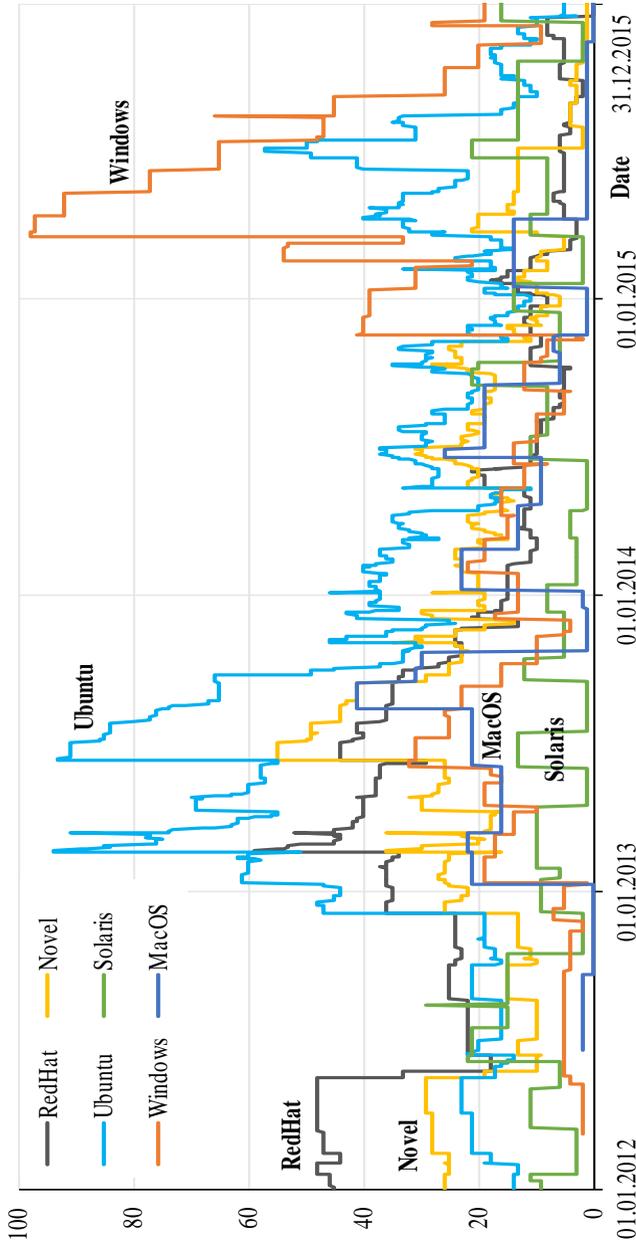


Fig. 25.3. Forever-day vulnerabilities

Table 25.3. Forever Day Vulnerabilities Statistics

Year	Forever Day Vulnerabilities	Operating System					
		Ubuntu	Windows	RedHat	Novell	MacOS	Solaris
2012	Min	13	2	18	9	0	2
	Max	48	7	48	29	2	29
	Average	21	4	33	19	1	10
	Std. deviation	8	1	12	8	1	6
2013	Min	24	1	13	14	0	1
	Max	94	32	59	55	41	13
	Average	60	18	34	30	18	7
	Std. deviation	18	7	9	11	11	4
2014	Min	11	2	4	6	1	1
	Max	46	41	21	31	26	21
	Average	27	16	11	19	13	7
	Std. deviation	8	10	4	5	8	5
2015	Min	3	9	0	1	0	2
	Max	57	98	18	21	14	21
	Average	23	48	6	7	4	9
	Std. deviation	12	27	3	6	6	6
Total	Min	3	1	0	1	0	1
	Max	94	98	59	55	41	29
	Average	33	22	21	19	10	8
	Std. deviation	20	22	15	11	10	6

Days-of-risk can be used to compare efforts that different vendors make to solve security issues and to delivery security updates fixing vulnerabilities. Fig. 25.4 shows how days-of-risk have been changing over years for different operating systems. It includes information taken from Table 25.2 (2012–2015) as well as data reported for earlier versions of studied OSs in [12, 16, 17, 21] by other researchers (depicted using dotted lines). For instance, according to [17] in 1999 Microsoft had an average of 16 days from vulnerability disclosure to patch. RedHat spent only 11 days to fix vulnerabilities while Sun proved itself to be very slow solving security problems in 90 days on average.

In 2006, as reported in [12, 21], days-of-risk parameter for Microsoft Windows series of operating systems (Windows 2000 Professional and Server, Windows XP, Windows Server 2003) was estimated at 29 in average.

At the same time Red Hat took 107 days to deliver security updates for its Enterprise Linux 2.1, 3.0 and 4.0 while Sun spent 168 days to do the same for any Solaris version patched in 2006. In addition, it was estimated that Apple Mac OS X and Novell SUSE Linux Enterprise Server and Desktop (versions 8–10) had 46 and 74 days-of-risk respectively.

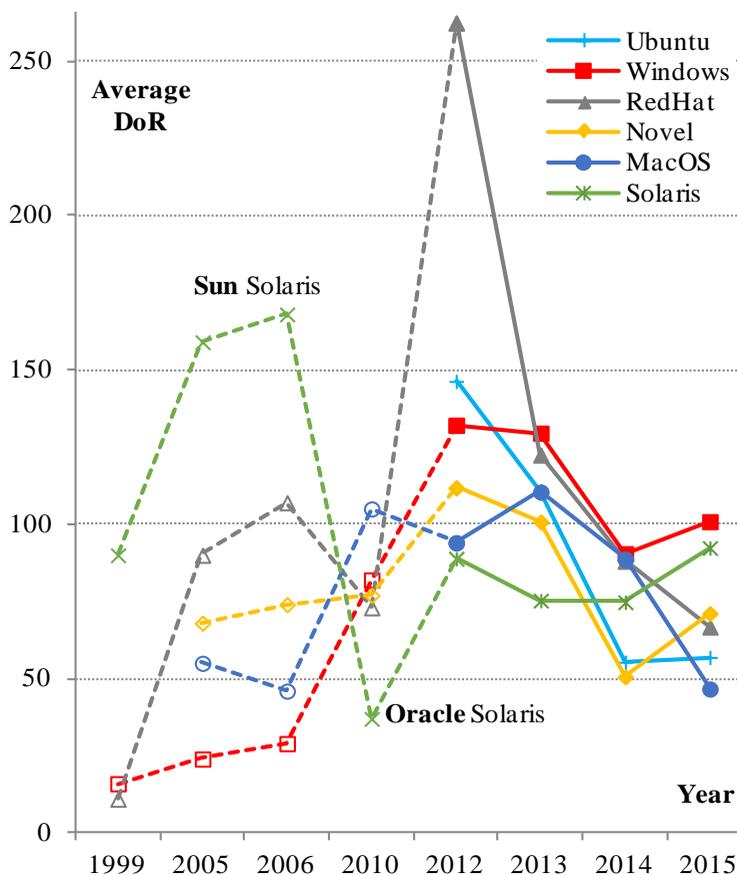


Fig. 25.4. Operating systems average *days-of-risk*

Finally, at the SERENE'2011 workshop the authors reported how days-of-risk changed in 2010 [13].

Figure 25.4 shows that since 2013 there is a tendency towards decreasing days-of-risks. During last two years average days-of-risk for different operating systems varies between 50 and 100 days. Unfortunately, it still means that after vulnerability public disclosure users of affected operating system are remaining vulnerable and unprotected against potential hacker attacks during months and OS vendors know it!

Besides, the statement argued by Jeff Jones in a series of his earlier blog posts [12, 18, 22] that Windows is the platform exposing users to risks for the shortest period of time as compared to other OSs seems to be no longer true.

At the same time, we can see that since Oracle took ownership of Solaris OS in 2009 it has been reacting on new vulnerabilities much faster.

In addition, we have build probability density functions (see Fig. 25.5) defining the relative likelihood for the vulnerability to be patched on a particular day after it was disclosed, that is much more informative than the average days-of-risk. It allows to estimate a probability of issuing patch before the specified date or to define confidence intervals.

It shows, for instance, that vulnerabilities in Novell and Ubuntu operating systems are usually fixed during first 30 days after disclosure. Majority of Windows' vulnerabilities are patched between 60 and 120 days while Apple usually spends from 60 to 90 days to issue security updates for MacOS.

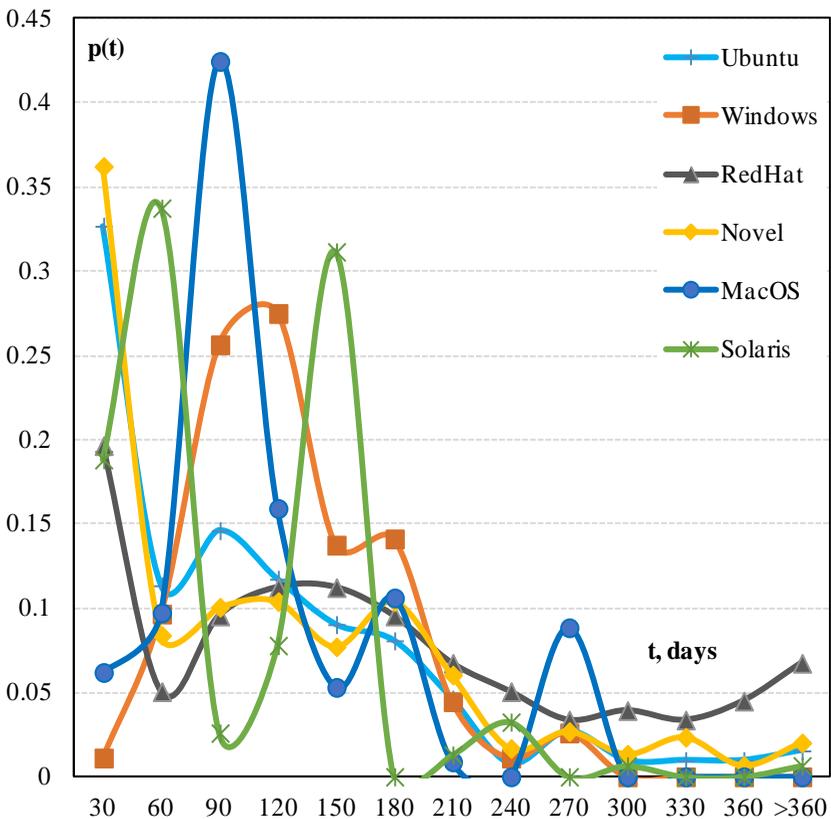


Fig. 25.5. Probability density functions of operating systems *days-of-risk*.

25.6.3 Vulnerabilities Severity Analysis

Severity is an important characteristic quantifying vulnerability impact on system's security. NVD database has adopted Common Vulnerability Scoring System (CVSS) to assign severity scores to software vulnerabilities.

CVSS are calculated based on several metrics that approximate ease of vulnerability exploitation (possibility of remote access, access complexity and need for authentication), vulnerability impact on confidentiality, integrity and availability and other factors [22].

Table 25.4. OSs Vulnerability Severity Statistics

Year	Operating System	Number of vulnerabilities by severity score									
		1	2	3	4	5	6	7	8	9	10
2012	Ubuntu	3	4	1	20	11	11	6		1	1
	Windows					2		2		5	1
	RedHat	3	3	2	7	4	4	2		1	1
	Novell	2	3	1	11	2	8	2		1	1
	MacOS		1		1						
	Solaris	2	6	12	13	7	2	4	1		
	Total:	10	17	16	52	26	25	16	1	8	4
2013	Ubuntu	18	13	6	65	23	29	23	1	2	3
	Windows			1	6	7	5	29		9	2
	RedHat	10	3	1	18	7	15	7			2
	Novell	14	11	5	40	11	19	17		1	3
	MacOS	3	9	3	18	7	16	3			
	Solaris	3	3		17	1	2	3			1
	Total:	48	39	16	164	56	86	82	1	12	11
2014	Ubuntu	4	9	1	54	15	16	23			4
	Windows	3	3	1	9	6	6	16	1	17	2
	RedHat	1	1	1	5	4	4	6		1	3
	Novell	1	9	2	44	8	10	12			4
	MacOS		1	1	3	2	21	3		1	8
	Solaris	1	3	1	13	5	3	6			
	Total:	10	26	7	128	40	60	66	1	19	21
2015	Ubuntu	1	6	8	22	29	18	37	1	8	11
	Windows	3	19	2	8	8	11	48		35	2
	RedHat		4		2	7	6	7			
	Novell		6	1	7	2	2	6		2	6
	MacOS	2					4	6			1
	Solaris	5	3	5	13	1	4	5			
	Total:	11	38	16	52	47	45	109	1	45	20

Scores, as well as the overall severity rating are ranged from 0 to 10, with 10 being the most severe. Besides, vulnerability severity is divided on several qualitative ranges: Low [0.1-3.9], Medium [4.0-6.9], High [7.0-8.9], and Critical [9.0-10.0]. Table 25.4 shows how vulnerability severity has been changing over years for different operating systems.

Vulnerabilities in Oracle Solaris are the least critical. Their average severity is 4.13. The most severe vulnerabilities have been discovered in OS Microsoft Windows (average severity is 6.46) and Apple MacOS (average severity is 5.27). Moreover, amount of critical vulnerabilities [9.0-10.0] disclosed in OS Microsoft Windows is equal to 27% of total. At the same time, amount of such vulnerabilities for other operating systems is less than 9%.

In our research we have checked a widespread hypothesis that software vendors make more efforts on fixing the most critical vulnerabilities firstly. However, a diagram on Fig. 25.6 shows that days-of-risk metric does not actually depend on vulnerability severity.

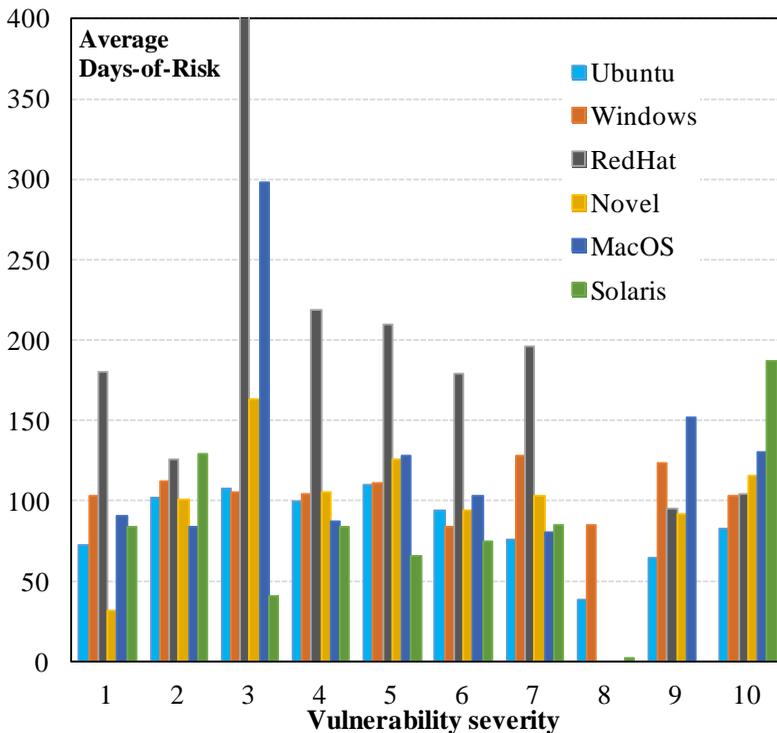


Fig. 25.6. Average *days-of-risk* depending on vulnerability severity

25.6.4 The Most Common Types of OSs Vulnerabilities

NVD vulnerability database classifies all vulnerabilities using CWE scheme. The Common Weakness Enumeration (CWE) is a formal list of software weakness types proposed by MITRE Corporation (<https://cwe.mitre.org/>).

The top ten vulnerability types discovered in different operating systems is presented in Table 25.5. About 90 percent of vulnerabilities in Oracle Solaris are marked as 'Uncategorized'. Thus we took them out of consideration.

The most common OS vulnerabilities by the CWE types (sorted by prevalence) are:

CWE-264 – Weaknesses and mistakes in permissions, privileges, and access controls;

CWE-119 – Improper restriction of operations within the bounds of a memory buffer using lacks of certain programming languages (often C and C++) that do not control bounds for the memory buffer that is being addressed. Vulnerabilities of CWE-119 type usually cause arbitrary code execution, altering the intended control flow, reading protected information or system crash;

CWE-20 – Improper input validation which may result in altered control flow, arbitrary code execution or illegal access to and control of resources;

CWE-200 – Information intentional or unintentional exposure to an actor that is not explicitly authorized to have access to that information;

CWE-399 – Improper management of system resources, e.g. memory allocation or reallocation;

CWE-189 – Numeric errors related to improper calculation or conversion of numbers;

CWE-362 – Concurrent code execution using shared resource with improper synchronization also known as *Race Condition*;

CWE-310 – Cryptographic issues including missing encryption of sensitive data or key management errors;

CWE-94 – Improper control of code generation also known as *Code Injection* which often happens when software allows a user's input to contain code syntax.

CWE-59 – Improper link resolution before file access that allows an attacker to traverse the file system to unintended locations and read/overwrite the contents of unexpected files. The first three types of vulnerabilities have been dominated over years (see Fig. 25.7). Since 2015 their contribution to the total number of discovered vulnerabilities has exceeded 70%.

Table 25.5. The Most Common OS Vulnerability Types

Vulnerability Type	Percentage of vulnerabilities by CWE types						
	Ubuntu	Windows	RedHat	Novell	MacOS	Solaris	Total*
CWE-264	13.41	26.01	15.43	14.62	21.93	-	18.28
CWE-119	19.35	12.45	18.62	15.95	18.42	2.60	16.96
CWE-20	11.49	17.58	9.04	13.62	21.93	1.30	14.73
CWE-200	7.09	9.52	6.91	9.63	7.89	-	8.21
CWE-189	9.58	2.20	10.11	9.97	3.51	3.25	7.07
CWE-399	9.58	4.40	7.98	9.97	2.63	3.25	6.91
CWE-362	4.02	1.83	4.26	5.98	0.00	-	3.22
CWE-310	2.49	1.83	1.06	3.32	7.02	-	3.15
CWE-94	0.38	8.79	0.53	0.00	0.00	-	1.94
CWE-59	1.15	0.37	2.13	0.33	0.00	-	0.80
Others	4.98	7.33	4.26	2.33	7.02	-	5.18
Uncategorized	16.48	7.69	19.68	14.29	9.65	89.61	13.56

*Taking out of consideration Solaris' vulnerabilities

CWE-264 weaknesses usually mean implementation mistakes in permissions, privileges, and access control mechanisms. As a result, vulnerable software cannot properly identify session reuse (CVE-2016-3840), bypasses check for the access, read or write permissions (CVE-2016-2416 or CVE-2016-6536) or relies on client-side authorization that can be easily bypassed via certain changes in local files (CVE-2015-5989).

Improper input validation (CWE-20) is a parent of other widespread vulnerability types including *command injection* (CWE-77), *cross-site scripting* (CWE-79) and *SQL injection* (CWE-89).

CWE-119 weaknesses (e.g. CVE-2016-7277, CVE-2016-4658 or CVE-2016-4598) often allow remote attackers to execute arbitrary code, read protected data or cause a denial of service via a crafted document viewed by victim on the infected web-page (e.g. .jpeg image or .xml file) or downloaded from the Internet and opened on his/her computer (e.g. .doc/.pdf documents or media files).

It is remarkable that numeric errors caused by incorrect calculation or conversion of numbers (CWE-189) affect not only system dependability but also security. This type of errors still remains quite typical causing sometimes sad but striking mistakes.

In particular, the authors of [23] have state that more than 700 papers reporting on various strands of the genomic research over the 10-year period are riddled with errors due to an erroneous conversation of some gene symbols (e.g. MARCH1 or 2310009E13) to date and numbers in Excel spreadsheets.

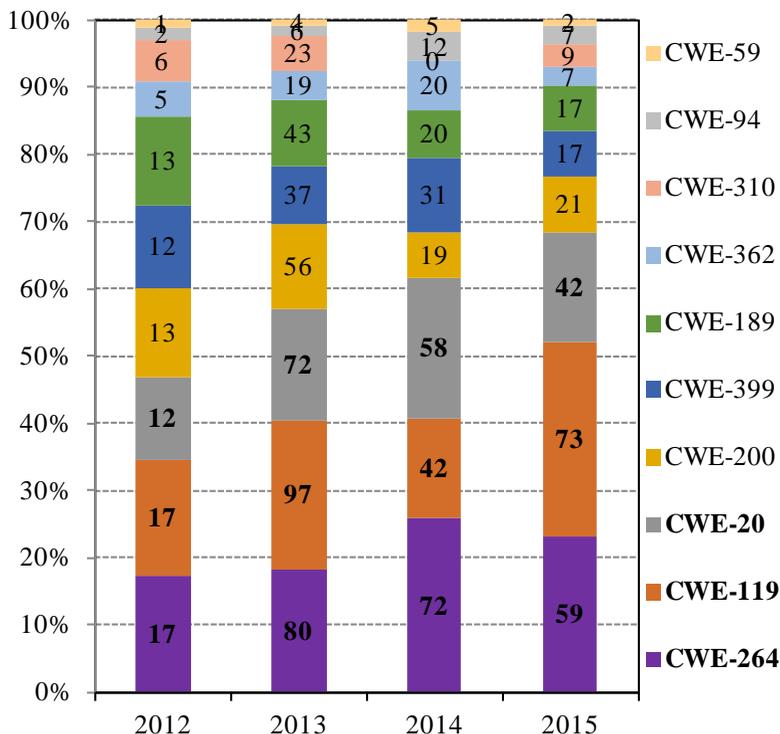


Fig. 25.7. Distribution of CWE by years

25.6.5 Common and Group Vulnerabilities

The most dangerous vulnerabilities are those discovered in more than one operating system. A reason why the same vulnerability is discovered in several OSs is explained by using common vulnerable components (system libraries, third party software components, OS kernels, etc.).

More often group vulnerabilities are discovered in different releases of the same OS or in a family of related operating systems, e.g. BSD Unix (OpenBSD, FreeBSD, NetBSD) or Linux (RedHat, CentOS, Novell, Ubuntu), etc.

However, sometimes hackers and security analysts discover vulnerabilities that are common for even different OS families. For example, the CVE-2008-4609 vulnerability caused denial-of-service attack for a variety of operating systems and their versions, including Linux, BSD Unix, Microsoft Windows, Cisco IOS and possibly many others [24, 25].

The vulnerability manipulated the state of Transmission Control Protocol (TCP) connections exploiting an algorithmic error in protocol implementation in various operating systems. Remote attacker was able to cause connection queue exhaustion by flags manipulation in TCP header of crafted network packets sent to a computer-victim.

Figure 25.8 shows vulnerabilities distributed between Ubuntu, Novell and RedHat operating systems during 2012-2015. Forty seven of them were common for all three operating systems. Besides, there were 3 groups of vulnerabilities shared between pairs Ubuntu and Novell (208), RedHat and Ubuntu (20), and Novell and RedHat (16).

The largest number of group vulnerabilities shared between Ubuntu and Novell operating system are those discovered in Linux kernels (versions 3.2.x and 3.0.x) used by them.

Besides, RedHat and MacOS share the CVE-2013-1824 vulnerability in PHP SOAP parser which allows remote attacker to gain unauthorised access to arbitrary files of operating systems.

The number of vulnerabilities shared by two or more operating systems can be used as a measure of diversity between them [9]. Software diversity has been used as a major fault and intrusion-tolerance mechanism to design safety-critical computer systems. Thus, vulnerability databases (the NVD database in particular) can help in determining the most diverse software products.

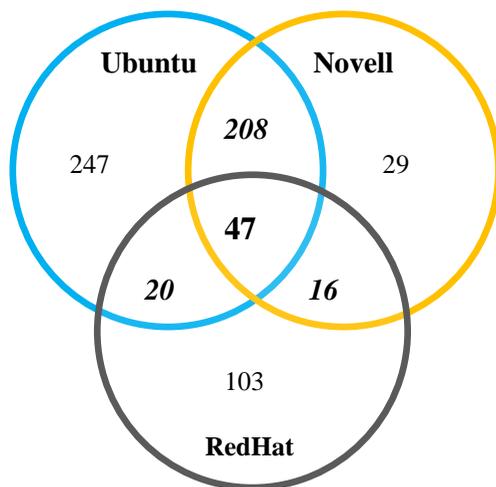


Figure 25.8. Number of individual, group and common vulnerabilities shared by Linux family of operating systems (Ubuntu, Novell and RedHat).

Conclusion and Questions for the Self-Control

This chapter presents a retrospective vulnerability analysis of the popular enterprise operating systems: Ubuntu Server 12.04, Red Hat Enterprise Linux 6, Novell Linux Enterprise Server 11 SP2, Microsoft Windows Server 2012 R2, Apple MacOS Server 10.8 and Oracle Sun Solaris 11.

Significant growth of the total number of vulnerabilities discovered in modern operating systems as well as the general tendency toward increasing their severity demonstrate serious security challenges and risks that OS developers and users face.

It is very important that the crucial parameters affecting system security are not only the total number of vulnerabilities disclosed in a particular software product and their severity but also, so called, days-of-risk which shows how fast software vendors issue patches fixing disclosed vulnerabilities. Our study shows that average days-of-risk for the investigated operating systems varies from 83 days for Oracle Solaris up to 135 days for RedHat.

It is worrying that we have discovered that the rate with which software developers issue security updates in general does not depend on vulnerability severity. Average days-of-risk for the most critical vulnerabilities remains even higher than one calculated for vulnerability of the lowest severity (139 vs 94 days). This uncovers certain shortcomings in security updates development policies adopted by OS vendors and in maintenance management processes they run.

The number of OS vulnerabilities that remain unpatched is growing. The increase of days-of-risk and the raise of a number of forever-day vulnerabilities threaten security and dependability of computer systems.

At the end of the paper we investigated vulnerabilities that were discovered in more than one operating systems. Such vulnerabilities that are common for different operating systems and even different OS families can lead to large-scale hacker attacks and virus epidemics. They also seriously complicate the development of intrusion-tolerant computer systems based on OS diversity.

The results presented in the paper show that numerous vulnerabilities in operating systems cause significant security threats. This calls for implementing *defence-in-depth* principle assuming application of layered security mechanisms incorporating together antivirus software, firewalls, security scanners, intrusion detection systems and other solutions in a way that makes them aware of the recent and current OS vulnerabilities.

The software vendors should clearly pay more attention to improving security of their products which might require significant changes in their software development and maintenance processes. Our experimental work supports our claim that decreasing days-of-risk and reducing a number of

forever-day vulnerabilities is one of the main challenges in building secure operating systems.

This work is aimed to answer a series of related questions, including:

1. What is software vulnerability?
2. How do vulnerabilities affect system security?
3. What are the most popular vulnerability databases?
4. What are differences between CVE and NVD databases?
5. What is software vulnerability life cycle and its major milestones?
6. What are differences between vulnerability discovery and disclosure?
7. How is a risk of exposure changing during the vulnerability life cycle?
8. How has vulnerability of enterprise operating systems been changing over past years?
9. How much time in average OS vendors spend to issue patch fixing vulnerabilities in their products?
10. What is *days-of-risk*? What is the average *days-of-risk* for the investigated operating systems?
11. What are *forever-day vulnerabilities*? What are differences between *zero-day vulnerabilities* and *forever-day vulnerabilities*?
12. How many forever-day vulnerabilities had been observed in various operating systems during 2012-2015?
13. What are the most dominating types of software vulnerabilities observed in different operating systems?
14. How severe in average are vulnerabilities discovered in different operating systems?
15. How diverse are different operating systems taking into account numbers of the common and group vulnerabilities?

References

- [1] B. Clark, "Hackers take hospital offline, demand \$3.6m ransom," [Online]. Available: <http://thenextweb.com/insider/2016/02/15/hackers-take-hospital-offline-demand-3-6m-ransom/>.
- [2] C. Williams, "Passengers ride free on SF Muni subway after ransomware infects network, demands \$73k," [Online]. Available: http://www.theregister.co.uk/2016/11/27/san_francisco_muni_ransomware/.
- [3] MITRE Corporation, "Common Vulnerabilities and Exposures. Terminology," [Online]. Available: <https://cve.mitre.org/about/terminology.html>.
- [4] National Vulnerability Database, "Vulnerability Summary for CVE-2016-7256," [Online]. Available:

- <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-7256>.
- [5] MITRE Inc., "CVE-2016-7256," [Online]. Available: <http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-7256>.
 - [6] Microsoft Inc., "Microsoft Security Bulletin MS16-132 - Critical," [Online]. Available: <https://technet.microsoft.com/en-us/library/security/ms16-132.aspx>.
 - [7] National Vulnerability Database, "Vulnerability Summary for CVE-2014-0160," [Online]. Available: <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-0160>.
 - [8] S. Frei, M. May, U. Fiedler and etc., "Large-scale vulnerability analysis," in *SIGCOMM Workshop on Large-Scale Attack Defense*, 2006.
 - [9] M. Shahzad, M. Zubair Shafiq and A. Liu, "A large scale exploratory analysis of software vulnerability life cycles," in *34th Int. Conf. on Software Engineering (ICSE '12)*, 2012.
 - [10] J. Jones, "Basic Guide to Days of Risk," 2007. [Online]. Available: <http://www.csoonline.com/article/2136934/data-protection/basic-guide-to-days-of-risk.html>.
 - [11] L. D. T. Bilge, "Before we knew it: An empirical study of zero-day attacks in the real world," in *ACM Conference on Computer and Communications Security*, Raleigh, NC, 2012.
 - [12] OSVDB, "Rebuttal: Dark Reading's "9" Sources for Tracking New Vulnerabilities," 2016. [Online]. Available: <https://blog.osvdb.org/2016/10/26/rebuttal-dark-readings-9-sources-for-tracking-new-vulnerabilities/>.
 - [13] M. Garcia, A. Bessani, I. Gashi and etc., "OS Diversity for Intrusion Tolerance: Myth or Reality?," in *IEEE/IFIP 41st Int. Conf. on Dependable Systems & Networks (DSN'2011)*, 2011.
 - [14] J. Reavis, "Linux vs. Microsoft: Who Solves Security Problems Faster?," 2000. [Online]. Available: <http://www.reavis.org/research/solve.shtml>.
 - [15] J. Jones, "Days-of-risk in 2006: Linux, Mac OS X, Solaris and Windows," 2006. [Online]. Available: <http://www.csoonline.com/article/2136935/data-protection/days-of-risk-in-2006---linux--mac-os-x--solaris-and-windows.html>.
 - [16] P. Edmonds, "When It Comes to Protection from Vulnerabilities, Process Trumps "Many Eyes"," 2007. [Online]. Available: <https://technet.microsoft.com/en-us/library/cc512608.aspx>.
 - [17] A. Patrizio, "Report Says Windows Gets The Fastest Repairs," 2007. [Online]. Available:

- <http://www.internetnews.com/security/article.php/3667201>.
- [18] D. Goodin, "Rise of "forever day" bugs in industrial systems threatens critical infrastructure," 2012. [Online]. Available: <http://arstechnica.com/business/2012/04/rise-of-ics-forever-day-vulnerabilities-threaten-critical-infrastructure/>.
- [19] M. Oiaga, "Recount: Windows Still Safest, Tops Mac OS X, Linux and Sun Solaris. But are statistics a true measure of security?," 2007. [Online]. Available: <http://news.softpedia.com/news/Recount-Windows-Still-Safest-Tops-Mac-OS-X-Linux-and-Sun-Solaris-57433.shtml>.
- [20] A. Gorbenko, O. Tarasyuk, V. Kharchenko and A. Romanovsky, "Using Diversity in Cloud-Based Deployment Environment to Avoid Intrusions," *Software Engineering for Resilient Systems*, no. LNCS 6968, p. 145–155, 2011.
- [21] J. Jones, "2006 Client OS Days of Risk," 2007. [Online]. Available: <https://blogs.microsoft.com/microsoftsecure/2007/06/18/2006-client-os-days-of-risk/>.
- [22] Forum of Incident Response and Security Teams, "Common Vulnerability Scoring System, V3 Development Update," 2015. [Online]. Available: <https://www.first.org/cvss>.
- [23] M. Ziemann, Y. Eren and A. El-Osta, "Gene name errors are widespread in the scientific literature," *Genome Biology*, vol. 17, no. 177, 2016.
- [24] National Vulnerability Database, "Vulnerability Summary for CVE-2008-4609," 2008. [Online]. Available: <https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-4609>.
- [25] Cisco Systems, "TCP State Manipulation Denial of Service Vulnerabilities in Multiple Cisco Products," 2009. [Online]. Available: <https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20090908-tcp24>.

CHAPTER 26 Modelling of Secure and Resilient Cloud Systems

26.1 Resilience Models for the Internet and Cloud Computing Systems

26.1.1 Time-Probabilistic Failure Model

Internet and Cloud computing systems as any other complex software may contain faults which may manifest themselves in operation. To every request, a remote system might return either a correct response – that is, succeed – or an erroneous response or exception – that is, fail. Failure behaviour of such globally distributed system is characterised by the probability of failure on demand (*pdf*). This probability can be statistically measured by a client as a ratio between r failures observed in n demands [1]. It can vary between the environments and the contexts (operational profiles) in which a web service is used.

The various factors, which affect the *pdf* may be unknown with certainty. Thus, the value of *pdf* may be uncertain as well. This uncertainty can be captured by a probability density series or probability distribution, built by aggregating usage experience of different clients.

Thus, the response returned to the client by a remote service may be of several types:

1. *Correct result*.
2. *Evident error* – an error that needs no special means to be detected. It concerns exception messages of different types reported to the client and notifying him about denial of the requested service for some reason.
3. *Non-evident (hidden) error* – an error that can be detected only by using a multiversioning at the application level (e.g. diversity of web services used).

However, the distributed nature of the service-oriented architectural model does not guarantee that the client receives a response from the web service within the finite time. If this happens we face so-called timing failures when the response is received too late or is not received at all. Thus, the known dependability definition [2] should be extended for service oriented systems as the “ability to deliver service *within the expected time* that can justifiably be trusted”.

In the Fig. 26.1 we adopt the failure model introduced by Avizienis, et al. in [2] to the distributed nature of the service-oriented systems and, more general, Internet and Cloud computing. The model distinguishes between the two main failure domains: (i) *timing failures* when the duration of the response delivered to the client exceeds the specified waiting time – the application

timeout (i.e. the service is delivered too late), and (ii) *content failures* when the content (value) of the response delivered to the client deviates from implementing the system function.

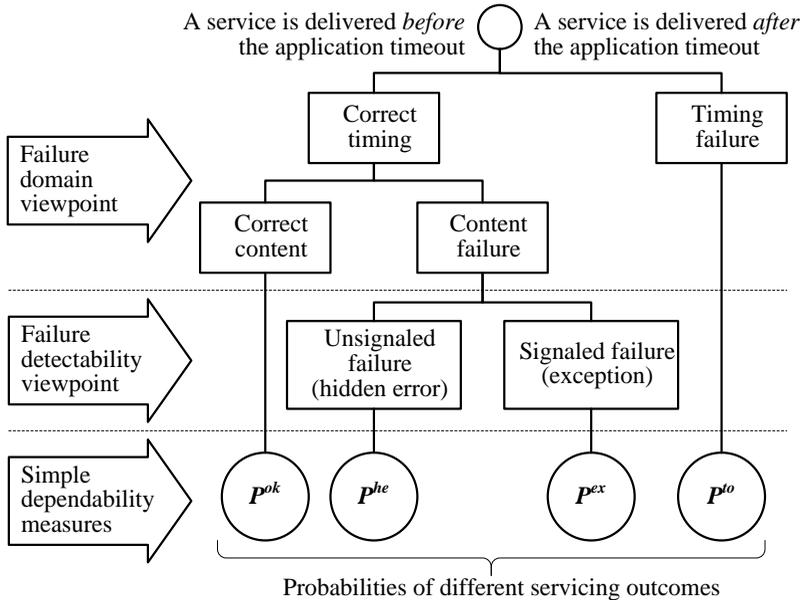


Fig. 26.1. Failure modes for the Interned and cloud computing systems

Probabilities p^{ok} , p^{he} and p^{ex} are conditional probabilities. They are conditioned on the arrival of some response within the timeout. Probabilities p^{ex} and p^{he} refer to failure modes that in the Avizienis’s classification correspond to the *detectability* viewpoint, where they are classified as: *signaled* and *unsignaled* failures, respectively.

The interdependency between probabilities of different servicing outcomes is shown in Fig. 26.2. The proposed time-probabilistic failure model also takes into account response time uncertainty in the form of the probability density function $f_t(t)$. Changing of timeout value causes changing the probability of timeout and, hence changing (redistribution) values of p^{ok} , p^{he} , p^{ex} and p^{to} as long as the sum of all probabilities must be equal to one. Hence, they are functions of a *timeout* setting:

$$p^{ok}(timeout) = p^{ok\infty} \cdot \int_0^{timeout} f_t(t)dt \tag{26.1}$$

$$p^{he}(timeout) = p^{he\infty} \cdot \int_0^{timeout} f_t(t) dt \quad (26.2)$$

$$p^{ex}(timeout) = p^{ex\infty} \cdot \int_0^{timeout} f_t(t) dt \quad (26.3)$$

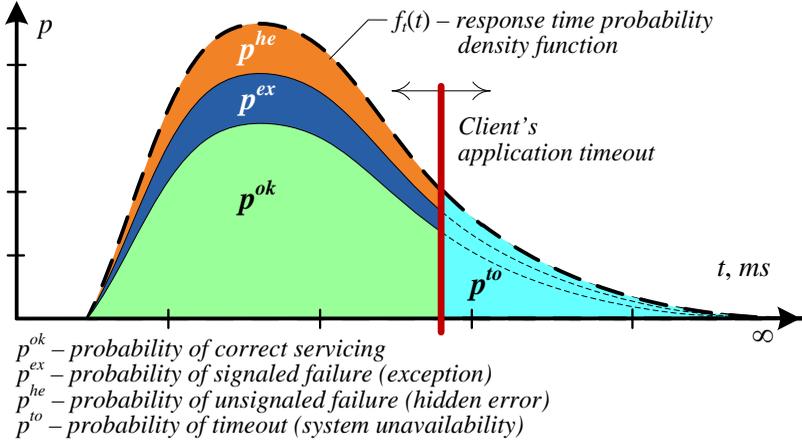


Fig. 26.2. Time-probabilistic failure model: the trade-off between availability and latency depending on time-out setup

where $p^{ok\infty}$, $p^{ex\infty}$, $p^{he\infty}$ are the eventual probabilities of getting a correct, evident and non-evident erroneous results with the unlimited waiting time, i.e. when $timeout \rightarrow \infty$.

The system unavailability can be estimated as the probability of the client receiving a response after the specified application timeout:

$$p^{to}(time-out) = \int_{timeout}^{\infty} f_t(t) dt \quad (26.4)$$

Besides, we introduce the following two measures estimating system latency: T^{av_srv} – average servicing time and T^{av_wait} – average waiting time. The expectation of $f_t(t)$ truncated from the right by a timeout is the average response (servicing) time of those invocations in which the client receives a response of any type before the specified time-out:

$$T^{av_srv}(timeout) = \frac{\int_0^{timeout} t \cdot f_t(t) dt}{F_t(timeout)} \quad (26.5)$$

where $F_t(\text{timeout}) = \int_0^{\text{timeout}} f_t(t)dt$ is the cumulative distribution function of a response time.

The average response (waiting) time $T^{\text{avg_wait}}$ estimated for all invocations, including those when a time-out is triggered, is the sum of $T^{\text{avg_srv}}$ under the specified time-out and a product of the time-out value and the probability of a time-out:

$$T^{\text{avg_wait}}(\text{timeout}) = \int_0^{\text{timeout}} t \cdot f_t(t)dt + \text{timeout} \cdot (1 - F_t(\text{timeout})) \quad (26.6)$$

This is because the waiting time for those invocations for which a time-out is triggered is equal to the time-out value. So, the weight of a tail of $f_t(t)$ truncated by the time-out is concentrated at the truncation border. It is obvious that $T^{\text{av_srv}} \leq T^{\text{av_wait}}$.

Using these equations, systems engineers can trade off between maximizing the service availability and minimizing its latency. Besides, these equations can help to choose appropriate application time-outs, which are the main error detection mechanism here. The practical example of estimation the probabilities of different type of failures and system latency depending on time-out settings can be found in [3].

26.1.2 Trade-offs Between Consistency, Availability and Latency in Resilient Internet and Cloud Computing

The CAP conjecture [4], which first appeared in 1998-1999, defines a trade-off between system availability, consistency and partition tolerance, stating that only two of the three properties can be preserved in distributed replicated systems at the same time. Gilbert and Lynch [5] view the CAP theorem as a particular case of a more general trade-off between consistency and availability in unreliable distributed systems which assume that updates are eventually propagated.

System partitioning, availability and latency are tightly connected. A replicated fault-tolerant system becomes partitioned when one of its parts does not respond due to arbitrary message loss, delay or replica failure, resulting in a timeout. System availability can be interpreted as a probability that each client request eventually receives a response. In many real systems, however, a response that is too late (i.e. beyond the application timeout) is treated as a failure. High latency is an undesirable effect for many interactive web applications. In [6] the authors showed that if a response time increases by as little as 100 ms, it dramatically reduces the probability of the customer

continuing to use the system. Failure to receive responses from some of the replicas within the specified timeout causes partitioning of the replicated system. Thus, partitioning can be considered as a bound on the replica's response time. A slow network connection, a slow-responding replica or the wrong timeout settings can lead to an erroneous decision that the system has become partitioned. When the system detects a partition, it has to decide whether to return a possibly inconsistent response to a client or to send an exception message in reply, which undermines system availability.

The designers of the distributed fault-tolerant systems cannot prevent partitions which happen due to network failures, message losses, hacker attacks and components crashes and, hence, have to choose between availability and consistency. One of these two properties has to be sacrificed. If system developers decide to forfeit consistency they can also improve the system response time by returning the fastest response to the client without waiting for other replica responses until the timeout, though this would increase the probability of providing inconsistent results. Besides, timeout settings are also important. If the timeout is lower than the typical response time, a system is likely to enter the partition mode more often [3]. It is important to remember that none of these three properties is binary. For example, modern distributed database systems, e.g. Cassandra [7], can provide a discrete set of different consistency levels for each particular read or write request. The response time can theoretically vary between zero and infinity, although in practice it ranges between a minimal affordable time higher than zero and the application timeout. Availability varies between 0% and 100% as usual.

The architects of modern distributed database management systems and large-scale web applications such as Facebook, Twitter, etc. often decide to relax consistency requirements by introducing asynchronous data updates in order to achieve higher system availability and allow a longer response time. Yet the most promising approach is to balance these properties. For instance, the Cassandra NoSQL database introduces a tuneable replication factor and an adjustable consistency model so that a customer can choose a particular level of consistency to fit with the desired system latency.

The CAP theorem helps the developers to understand the system trade-offs between consistency and availability/latency [8]. Yet even though this theorem strongly suggests that better consistency undermines system availability and latency, developers do not have quantitative models to help them to estimate the system response time for the chosen consistency level and to achieve a precise trade-off between them. Our interpretation of the CAP theorem and the trade-offs resulting from the CAP is depicted in Fig. 26.3. The application timeout can be considered as a bound between system availability and performance (in term of latency or response time) [9]. Thus, system

– QUORUM – the driver must wait for the responses from a QUORUM of replica web services. It provides a compromise between the ONE and ALL options trading off latency versus consistency. The quorum is calculated as: $(amount_of_replicas / 2) + 1$, rounded down to an integer value. As far as in our experiments we use the replication factor of 3, the quorum is 2.

In this section we present a set of probabilistic models that allow us to build a combined probability density function of system response time by taking into account provided consistency level and incorporating response time probability density functions for each replica.

When the system is configured to provide consistency level ALL, the probability of returning response to the client at time t is equal to the probability that one of the replicas (e.g. the first one) returns its response exactly at time t , i.e. $g_1(t)$ while two other replicas return their responses later than t (by time t), i.e. $\int_0^t g_2(t) = G_2(t)$ and $\int_0^t g_3(t) = G_3(t)$.

So far as we have three replicas, all three possible combinations have to be accounted. As a result, the probability density function of the system response time for consistency level ALL can be defined as following:

$$f_{ALL}(t) = g_1(t)G_2(t)G_3(t) + g_2(t)G_1(t)G_3(t) + g_3(t)G_1(t)G_2(t). \quad (26.7)$$

where $g_1(t)$, $g_2(t)$ and $g_3(t)$ – are response time probability density functions of the first, second and third replicas respectively; $G_1(t)$, $G_2(t)$ and $G_3(t)$ – are response time cumulative distribution functions of the first, second and third replicas respectively.

When the system is configured to provide consistency level ONE, the probability of returning a response to the client at time t is equal to the probability that if only one of the replicas (e.g. the first one) returns its response exactly at time t , i.e. $g_1(t)$, while two other replicas return their responses at the same time or later on, i.e. $\int_t^\infty g_2(t) = 1 - G_2(t)$ and $\int_t^\infty g_3(t) = 1 - G_3(t)$.

Keeping in mind three possible combinations we can deduce the probability density function of the system response time for consistency level ALL as:

$$f_{ONE}(t) = g_1(t)(1 - G_2(t))(1 - G_3(t)) + g_2(t)(1 - G_1(t))(1 - G_3(t)) + g_3(t)(1 - G_1(t))(1 - G_2(t)). \quad (26.8)$$

Deducing the response time probability density function for the QUORUM consistency level is based on a combination of the previous two cases.

The probability of returning response to the client at time t is equal to the probability that one of the replicas returns its response exactly at time t ; one of the two remained replicas returns its response by time t and another one responds at time t or later on. Taking into account all possible combinations the probability density function of the system response time for consistency level QUORUM can be deduced as:

$$\begin{aligned}
 f_{QUORUM}(t) &= (g_1(t)G_2(t) + g_2(t)G_1(t))(1 - G_3(t)) + \\
 &+ (g_1(t)G_3(t) + g_3(t)G_1(t))(1 - G_2(t)) + \\
 &+ (g_2(t)G_3(t) + g_3(t)G_2(t))(1 - G_1(t)).
 \end{aligned} \tag{26.9}$$

Using similar reasoning it is possible to deduce response time probability density functions of a system composed of n replicas:

$$f_{ALL}(t) = \sum_{i=1}^n \left(\frac{g_i(t)}{G_i(t)} \cdot \prod_{j=1}^n G_j(t) \right). \tag{26.10}$$

$$f_{ONE}(t) = \sum_{i=1}^n \left(\frac{g_i(t)}{1 - G_i(t)} \cdot \prod_{j=1}^n (1 - G_j(t)) \right). \tag{26.11}$$

It is extremely hard to build a general form of the probability density function of the system response time for consistency level QUORUM. However, the general reasoning is as following. The composed probability density function should be presented as a sum of m items, where m is a number of k -combinations of n (k is a number of replicas constituting a quorum). Each of the m items is a product of two factors. The first one defines the probability that a particular combination of k replicas return their responses by time t . Another factor defines the probability that the remaining $(n-k)$ replicas return their responses after t .

Let us consider how the proposed models can be applied in practice to estimate system's average response time and build its probability distribution function. Let assume that there is a three-replicated systems. The response time of each replica follows the exponential distribution $f_1(t) = f_2(t) = f_3(t) = \mu \cdot e^{-\mu \cdot t}$, where $\mu = 0.05$ which means that the average value of the response time is equal to 200 ms. Then, using (26.7) – (26.9) we can derive probability density functions of a system response time depending on the chosen consistency level:

$$f_{ALL}(t) = 3\mu \cdot e^{-\mu \cdot t} \cdot (1 - e^{-\mu \cdot t})^2,$$

$$f_{ONE}(t) = 3\mu \cdot e^{-3\mu \cdot t},$$

$$f_{QUORUM}(t) = 6\mu \cdot (e^{-2\mu \cdot t} - e^{-3\mu \cdot t}).$$

Figure 26.4 displays probability density functions of the system response time corresponding to different consistency levels. As it was expected, choosing the strongest consistency level (ALL) significantly increases the average response time, which is equal to 366.6(6) ms in our case. The weakest consistency level (ONE) drops the average response time down to 66.6(6) ms. The QUORUM consistency level provides a compromise between system performance (it allows to slightly decrease the average time down to 166.6674 ms) and consistency.

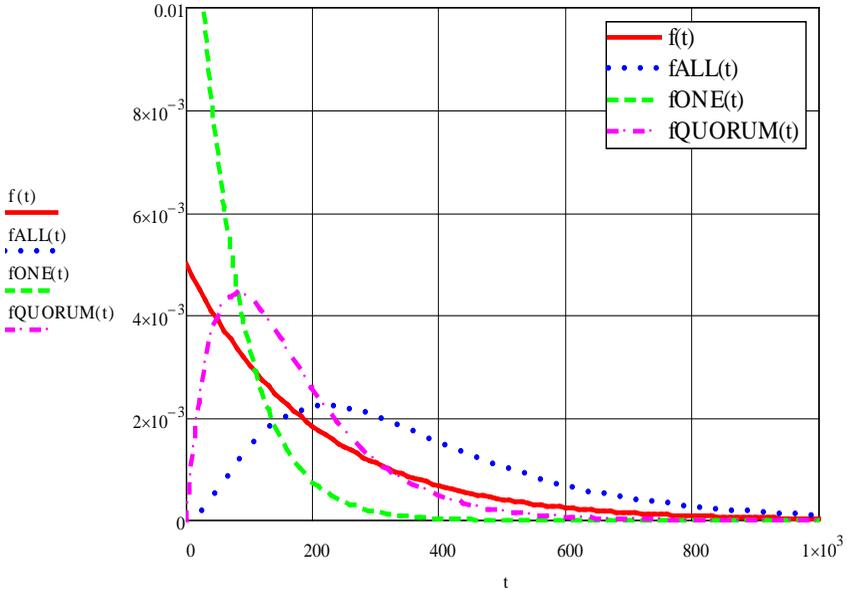


Fig. 26.4. Probability density functions of the replica $f(t)$ and system response times $f_{ALL}(t)$, $f_{ONE}(t)$ and $f_{QUORUM}(t)$

26.1.4 Probabilistic Models of the System Dependability

In this section we introduce theoretical models estimating probabilities of different service outcomes depending on the consistency level provided by a replicated system or chosen by a client. There are two basic assumptions used by the models:

1) probabilities of different servicing outcomes (p^{ok} , p^{he} , p^{ex} , p^{to}) are identical but independent for all replicas/diverse components;

2) if none of replica services has returned any response before the specified timeout a system reports an exception to a client. Hence, $P_{ALL,ONE}^{to} = 0$;

3) it is assumed that replicated services are diverse and, hence, probabilities of common and group failures are equal to zero.

ALL/QUORUM consistency level/pattern. The correct response will be provided to a client if at least two (or more) replicas return correct responses. Other replicas can return responses of different type (including hidden errors; as far as we assume that all services are diverse, hidden errors possibly provided by them will be different). The total number of favorable combinations is C_n^i . Thus a probability of correct servicing can be estimated

$$\text{as } C_n^i \sum_{i=2}^n (p^{ok})^i \cdot (1 - p^{ok})^{n-i}.$$

Besides, if the only one service returns a non-exceptional response while others return exceptions or are timed out, it will be relayed to a client. Taking into account n favorable combinations (C_n^1) the probability of the correct servicing in this case is equal to $n \cdot p^{ok} (p^{ex} + p^{to})^{n-1}$.

Combining these two probabilities we can derive the following:

$$P_{ALL}^{OK} = \sum_{i=2}^n C_n^i \cdot (p^{ok})^i \cdot (1 - p^{ok})^{n-i} + n \cdot p^{ok} (p^{ex} + p^{to})^{n-1}, \quad n > 2.$$

By applying similar reasonings we can estimate probabilities of the hidden error and exception:

$$\begin{aligned} P_{ALL}^{HE} &= n \cdot p^{he} (p^{ex} + p^{to})^{n-1}, \quad n > 2, \\ P_{ALL}^{EX} &= (p^{ex} + p^{to})^n + (p^{he})^n + n \cdot p^{ok} \sum_{i=1}^{n-1} C_{n-1}^i \cdot (p^{he})^i \cdot (p^{to} + p^{ex})^{n-i-1} + \\ &+ p^{he} \sum_{i=1}^{n-2} C_n^{i+1} \cdot (p^{he})^i \cdot (p^{to} + p^{ex})^{n-i-1}. \end{aligned}$$

If we assume that all replicas are copies of the same server (i.e. all replicas are identical) all hidden errors will not be distinguished one from another. It means that the overall probability that a hidden error is returned to a client

increases correspondingly. In this case the above equations should be modified as following:

$$\begin{aligned}
 P_{ALL}^{OK} &= \sum_{i=\frac{n}{2}+1}^n C_n^i \cdot (p^{ok})^i \cdot (1 - p^{ok})^{n-i} + \\
 &+ \sum_{i=1}^{n/2} C_n^i \cdot (p^{ok})^i \cdot \sum_{j=0}^{i-1} C_{n-i}^j \cdot (p^{he})^j \cdot (p^{ex} + p^{to})^{n-i-j}, \\
 P_{ALL}^{EX} &= (p^{ex} + p^{to})^n + \sum_{i=1}^{n/2} C_n^i \cdot C_{n-i}^i \cdot (p^{ok})^i \cdot (p^{he})^i \cdot (p^{ex} + p^{to})^{n-2i}, \\
 P_{ALL}^{HE} &= \sum_{i=\frac{n}{2}+1}^n C_n^i \cdot (p^{he})^i \cdot (1 - p^{he})^{n-i} + \\
 &+ \sum_{i=1}^{n/2} C_n^i \cdot (p^{he})^i \cdot \sum_{j=0}^{i-1} C_{n-i}^j \cdot (p^{ok})^j \cdot (p^{ex} + p^{to})^{n-i-j}
 \end{aligned}$$

Analytical models estimating probability of different servicing outcomes for the QUORUM pattern are identical to those proposed above (where n should be considered as a quorum number of replicas instead of their total number). The only difference is that QUORUM pattern provide less response time than the ALL one.

ONE consistency level/pattern. If the system provides the weakest consistency level when all available services are requested but the only fastest non-exceptional response is returned to a client, the probabilities of different servicing can be estimated as following:

$$\begin{aligned}
 P_{ONE}^{OK} &= (p^{ok})^n + \sum_{i=1}^{n-1} C_n^{n-i} \cdot (p^{ok})^{n-i} \cdot (p^{to} + p^{ex})^i + \\
 &+ \sum_{i=1}^{n-1} \frac{n-i}{n} C_n^{n-i} \cdot (p^{ok})^{n-i} \cdot (p^{he})^i + \\
 &+ \sum_{i=1}^{n-2} i \cdot C_n^i \cdot (p^{ok})^i \cdot \sum_{j=1}^{n-1-i} \frac{C_{n-i}^j}{n-j} \cdot (p^{he})^{n-i-j} \cdot (p^{to} + p^{ex})^j, \\
 P_{ONE}^{EX} &= (p^{ex} + p^{to})^n,
 \end{aligned}$$

$$\begin{aligned}
 P_{ONE}^{HE} &= (p^{he})^n + \sum_{i=1}^{n-1} C_n^{n-i} \cdot (p^{he})^{n-i} \cdot (p^{to} + p^{ex})^i + \\
 &+ \sum_{i=1}^{n-2} i \cdot C_n^i \cdot (p^{he})^i \cdot \sum_{j=1}^{n-1-j} \frac{C_{n-i}^j}{n-j} \cdot (p^{ok})^{n-i-j} \cdot (p^{to} + p^{ex})^j + \\
 &+ \sum_{i=1}^{n-1} \frac{n-i}{n} C_n^{n-i} \cdot (p^{he})^{n-i} \cdot (p^{ok})^i, \quad n > 2.
 \end{aligned}$$

SEQUENCE pattern. Some replicated system can implement SEQUENCE pattern when replicas are invoked in a sequence. A subsequent replica is invoked if the preceding replica returns the exception. It is an extension of the *simple-retry* recovery technique which assumes invoking the same replica again after transient errors have happened.

If we assume that all replicas have the same p^{ok} , p^{he} , p^{ex} , p^{to} probability values their order of invocation does not matter. Hence, we can derive:

$$\begin{aligned}
 P_{SEQUENCE}^{OK} &= p^{ok} \cdot \sum_{i=0}^{n-1} (p^{ex} + p^{to})^i \\
 P_{SEQUENCE}^{EX} &= (p^{ex} + p^{to})^n, \\
 P_{SEQUENCE}^{HE} &= p^{he} \cdot \sum_{i=0}^{n-1} (p^{ex} + p^{to})^i.
 \end{aligned}$$

Analytical models investigation. Figures 26.5-7 shows graphs of p^{ok} , p^{he} , p^{ex} probabilities for different patterns depending on a number of replicas. As an example, we assume that each replicas have identical dependability characteristics: $p_i^{ok}=0.7$, $p_i^{to}=0.2$, $p_i^{ex}=0.05$, $p_i^{he}=0$.

Graphs $PokALL1(n)$, $PheALL1(n)$ and $PexALL1(n)$ correspond to the case when all replicas are assumed as ideally diverse, i.e. when there are no common and group faults shared between them.

Graphs $PokALL2(n)$, $PheALL2(n)$ and $PexALL2(n)$ are build assuming that all replicas are 100% identical.

Analysis shows that probability $Pok(n)$, $Phe(n)$ and $Pex(n)$ graphs for ONE and SEQUENCE patterns are identical despite the fact that ONE pattern provides significantly less response time, as was shown in Section 26.1.3.

Probability of correct servicing for the ALL pattern under the assumption that all replicas are ideally diverse is expectedly higher than when we assume that all replicas are identical.

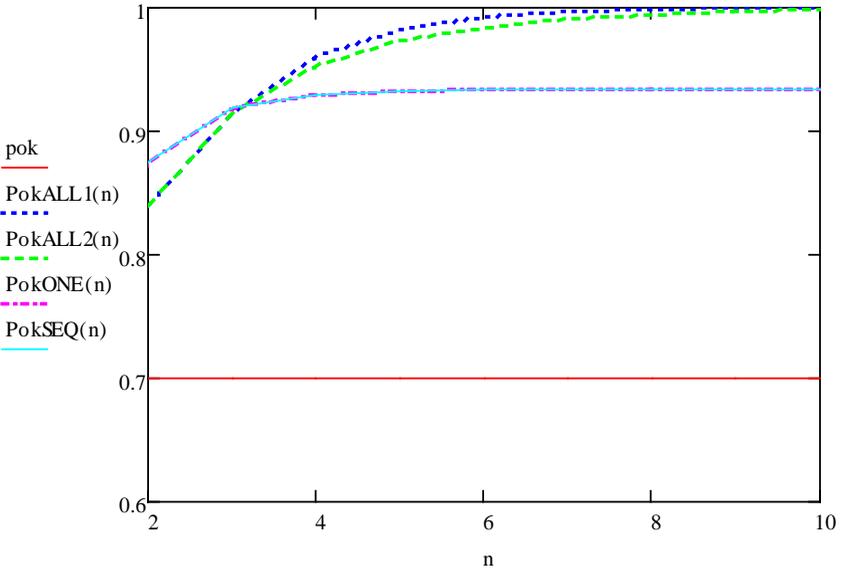


Fig. 26.5. Probability of correct servicing depending on a number of system replicas

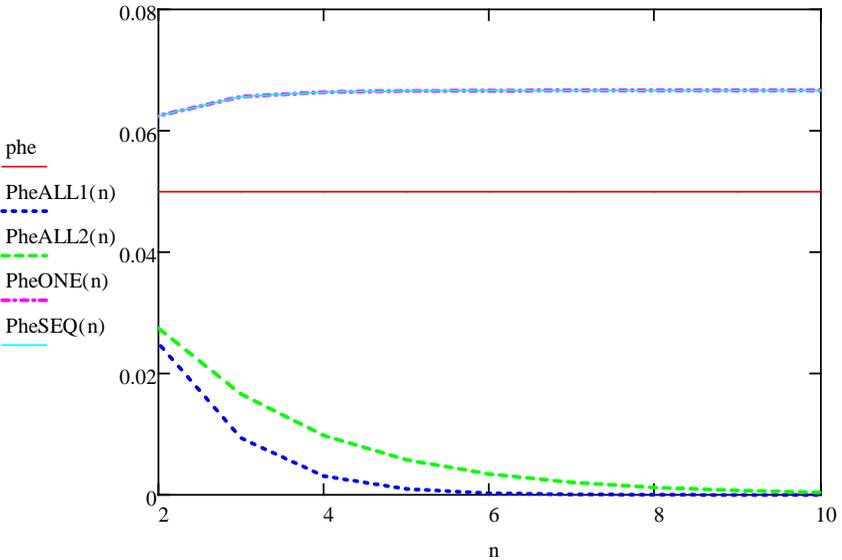


Fig. 26.6. Probability of a hidden error depending on a number of system replicas

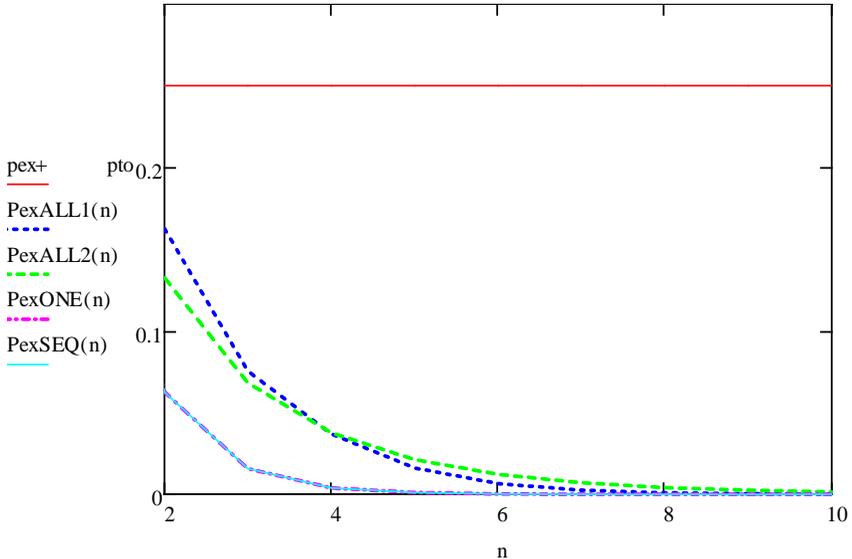


Fig. 26.7. Probability of an exception depending on a number of system replicas

26.2 Modelling intrusion avoidance approach via dynamic reconfiguration of the SW system environment

26.2.1 Diversity of the System Environment

Design diversity is one of the most efficient methods of providing software fault-tolerance [12]. In regard to multitier architecture of web-services, software diversity can be applied at the level of the operating system, web and application servers, data base management systems and, finally, for application software, both separately and in many various combinations.

Platform-independent technologies like Java, Python, Perl, Ruby, PHP, etc. provide the crucial support for applying diversity of different system components. For example, thanks to JVM, Java applications can be run on different operating systems under control of various web and applications servers. This feature provides ability to use the deployment environment which can be dynamically reconfigured by replacing one component by another one of the same functionality (e.g. Linux OS can be replaced with Solaris OS, GlassFish AS with Oracle WebLogic, or IBM WebSphere, etc.).

Table 26.1 lists different diverse components that can be used at various levels of the software system environment.

Table 26.1. Diversity level and diversity components of the system deployment environment

Diversity Level	Diverse system components
Operating Systems (OS)	Windows OS Series, MacOS X Server, Linux, FreeBSD, IBM AIX, Oracle Solaris, HP-UX, etc.
Web-server (WS)	Apache httpd, Oracle iPlanet Web Server, IBM HTTP Server, lighttpd, nginx, Cherokee HTTP Server, etc.
Application server (AS)	GlassFish, Geronimo, Oracle WebLogic, JBoss, Caucho Resin, IBM WebSphere, SAP NetWeaver, Apple WebObjects, etc.
Data-Base management system (DBMS)	MS SQL Server, MySQL, Oracle Database, Firebird, PostgreSQL, SAP SQL Anywhere, etc.

26.2.2 Intrusion-avoidance approach

In [13] we proposed the intrusion avoidance approach which is based on the idea of running at the different levels of the multitier system architecture (OS, WS, AS and DBMS) only the least vulnerable components. Other diverse components should be hold in a stand-by mode.

Fig. 26.8 specifies a behavior of the configuration controller. When a new vulnerability is disclosed, the most vulnerable system component should be replaced with the diverse one having fewer numbers of forever-day vulnerabilities. Such dynamic reconfiguration should also take into account severity and potential harmful consequences of different vulnerabilities, their popularity, availability of exploit code, etc. When a product vendor patches some vulnerability the system can be reconfigured again (after patch installation and re-estimation of the security risks).

The reconfiguration should be performed in a manner that is transparent and inconspicuous for the application software running on the top of the system software stack.

The proposed approach can be implemented by making use of existing virtualisation technologies. A PaaS cloud-platform can be developed to provide a trusted environment for secure deployment of application software and services. It can act as a mediator between clients and existing IaaS Cloud services (e.g. Amazon EC2).

The platform will (see Fig. 26.9) rent IaaS virtual instances, prepare and update diverse VM images, estimate vulnerability level of the diverse VM images, deploy and reconfigure virtual instance transparently to the client's cross-platform applications and services running on the top of it.

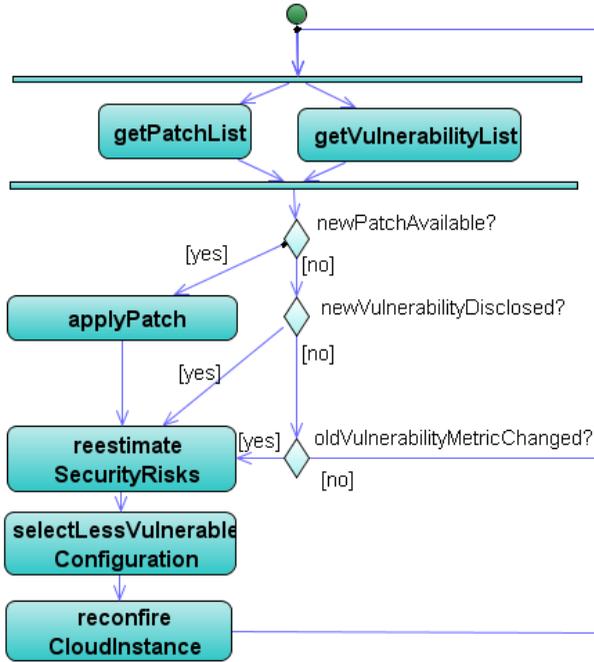


Fig. 26.8. UML Activity diagram specifying system reconfiguration

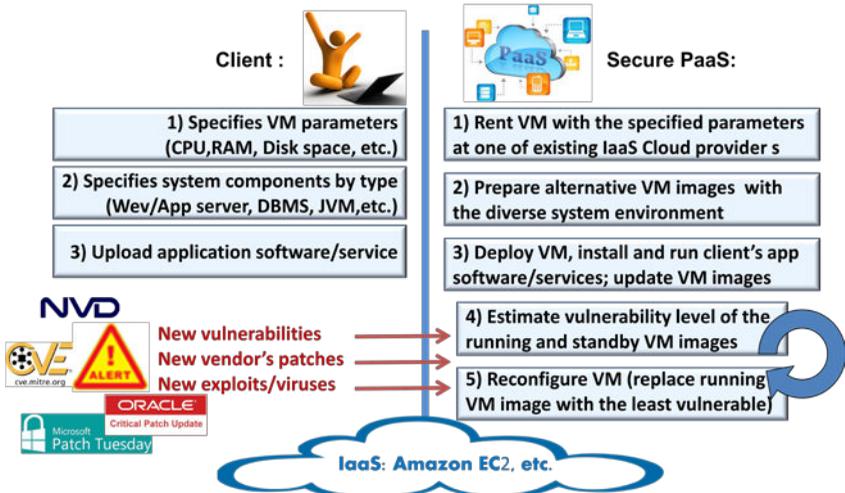


Fig. 26.9. Secure PaaS cloud platform

26.2.3 Intrusion-avoidance modelling

In this section we are demonstrating how system vulnerability can be reduced by employing the proposed intrusion-avoidance technique. We consider a diverse reconfigurable system which is dynamically switching between three Linux-based operating systems:

- Ubuntu Server 12.04;
- Red Hat Enterprise Linux 6;
- Novell Linux SUSE Enterprise Server 11 SP2.

A number of forever-day vulnerabilities (FDV) for above mentioned operating systems reported during 2012-2015 years is shown in Fig. 26.10. It demonstrates how many of known (already disclosed publicly) but yet unfixed (and, hence, can be hacked any time) vulnerabilities existed every day in particular operating system. More details concerning vulnerability study of those operating systems can be found in Chapter 25.

Besides, the bottom graph in Fig. 26.10 shows a number of forever-day vulnerabilities in a reconfigurable diverse system which switches between different operating systems to keeps a number of such vulnerabilities at the minimal level.

Green circles in Fig. 26.10 displays moments of time when switches are happened. The number inside each circle corresponds to the number of a particular switch from Table 26.3. The table reports results of dynamic operating system reconfigurations performed by the configuration controller taking into account the number of forever-day vulnerabilities. To simplify our demonstration we took out of consideration severity of different vulnerabilities.

In our simulation we used Ubuntu Server 12.04 as the initial active operating system. Table 26.3 shows the set of subsequent switches between different operating systems in accordance with the vulnerability discovering and fixing issuing process (see Fig. 26.10).

The table also presents the exact dates and periods of active operation of different operating systems. In our simulation, the overall period of the active operation for different operating systems was:

- Ubuntu Server 12.04 – 147 days;
- Red Hat Enterprise Linux 6 – 698 days;
- Novell Linux SUSE Enterprise Server 11 SP2 – 616 days.

As it can be seen from Table 26.4, the proposed approach to intrusion avoidance allows to hold the minimum possible number of forever-day vulnerabilities dynamically switching between diverse operating systems.

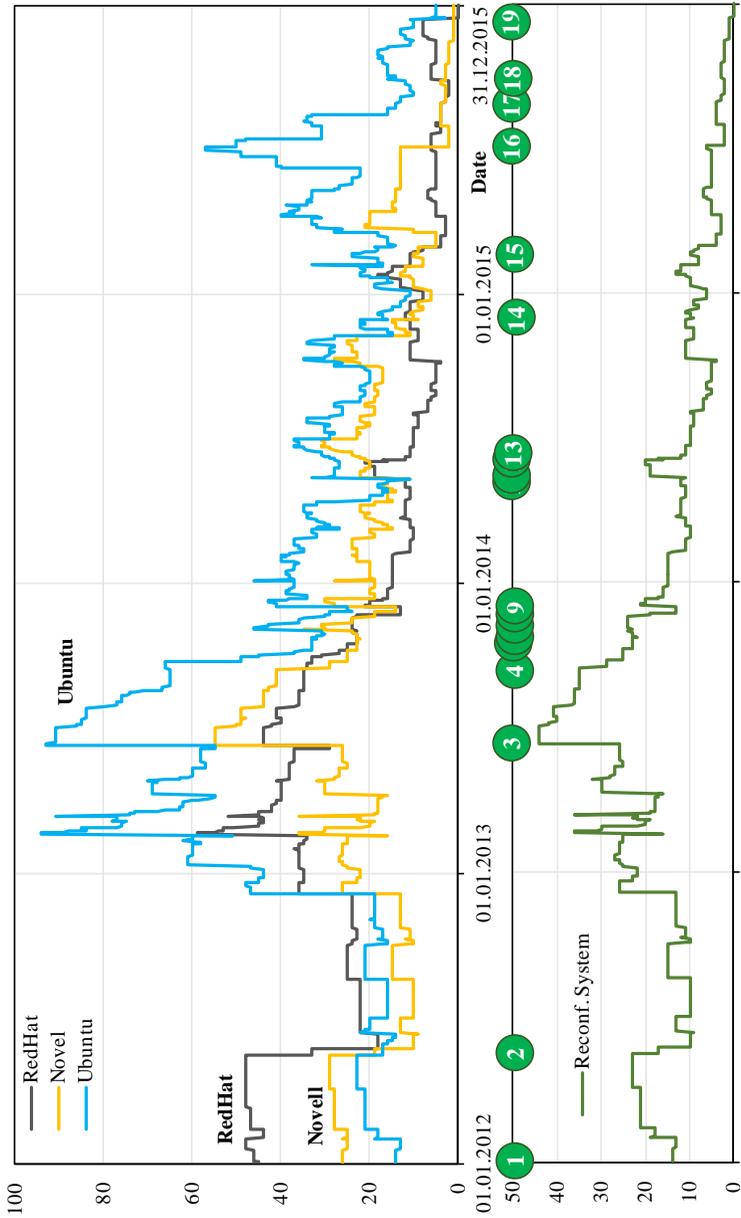


Fig. 26.10. Forever-day vulnerabilities in different OSs and the Reconfigurable Diverse System

Table 26.3. Operating systems reconfiguration summary

No	Operating System on Duty	Duty period			Average number of FDV
		Start Date	End Date	Duration, days	
1	Ubuntu Server 12.04	01.01.2012	24.05.2012	145	20
2	Novell Linux SUSE Enterprise Server 11 SP2	25.05.2012	11.06.2013	383	18
3	Red Hat Enterprise Linux 6	12.06.2013	15.09.2013	96	39
4	Novell Linux SUSE Enterprise Server 11 SP2	16.09.2013	17.10.2013	32	26
5	Red Hat Enterprise Linux 6	18.10.2013	23.10.2013	6	23
6	Novell Linux SUSE Enterprise Server 11 SP2	24.10.2013	24.10.2013	1	22
7	Red Hat Enterprise Linux 6	25.10.2013	19.11.2013	26	24
8	Novell Linux SUSE Enterprise Server 11 SP2	20.11.2013	22.11.2013	3	19
9	Red Hat Enterprise Linux 6	23.11.2013	11.05.2014	170	13
10	Ubuntu Server 12.04	12.05.2014	13.05.2014	2	11
11	Red Hat Enterprise Linux 6	14.05.2014	01.06.2014	19	19
12	Novell Linux SUSE Enterprise Server 11 SP2	02.06.2014	04.06.2014	3	20
13	Red Hat Enterprise Linux 6	05.06.2014	29.11.2014	178	9
14	Novell Linux SUSE Enterprise Server 11 SP2	30.11.2014	23.02.2015	86	9
15	Red Hat Enterprise Linux 6	24.02.2015	05.07.2015	132	5
16	Novell Linux SUSE Enterprise Server 11 SP2	06.07.2015	05.08.2015	31	2
17	Red Hat Enterprise Linux 6	06.08.2015	28.09.2015	54	3
18	Novell Linux SUSE Enterprise Server 11 SP2	29.09.2015	14.12.2015	77	2
19	Red Hat Enterprise Linux 6	15.12.2015	31.12.2015	17	0

The average instant number of the forever-day vulnerabilities would be equal to 15 that is almost 25% as less as the best result achieved by Novell Linux SUSE Enterprise Server 11 SP2 (19 vulnerabilities at once).

Conclusion and Questions for the Self-Control

In the chapter we discuss probabilistic models to define resilience property of the distributed Internet and Cloud computing systems. Besides we propose and simulate intrusion avoidance approach for the secure deployment of application software and services.

Table 26.4. Intrusion avoidance summary

No of FDV per year		Ubuntu	RedHat	Novell	Reconfigurable diverse system
2012	min	13	18	9	9
	max	48	48	29	26
	average	21	33	19	16
	std. dev.	8	12	8	5
2013	min	24	13	14	13
	max	94	59	55	44
	average	60	34	30	28
	std. dev.	18	9	11	8
2014	min	11	4	6	4
	max	46	21	31	20
	average	27	11	19	11
	std. dev.	8	4	5	4
2015	min	3	0	1	0
	max	57	18	21	13
	average	23	6	7	4
	std. dev.	12	3	6	3
Total	min	3	0	1	0
	max	94	59	55	44
	average	33	21	19	15
	std. dev.	20	15	11	10

The resilience can be considered as an ability of a system to trade-off and interplay between key non-functional properties. The designers of the distributed Internet and Cloud systems cannot prevent partitions which happen due to network failures, message losses, hacker attacks and components crashes and, hence, have to choose between availability and consistency. One of these two properties has to be sacrificed. If system developers decide to forfeit consistency they can also improve the system response time by returning the fastest response to the client without waiting for other replica responses until the timeout, though this would increase the probability of providing inconsistent results. In this work we introduce the time-probabilistic failure model and consider a fundamental trade-offs between consistency, availability and system latency. Besides, we propose a set of probabilistic models estimating system response time and system dependability which provide a mathematical framework allowing developers and clients to measure and interplay between different properties depending on their needs.

In the chapter we also discuss the intrusion-avoidance architecture that makes use of system component diversity can significantly improve the overall security of the computing environment used to deploy web services.

The approach proposed to intrusion avoidance is based on dynamical reconfiguration of the system by selecting and using the particular operating system, web and application servers and DBMS that have the minimal number of the forever-day (i.e. known but yet unpatched) vulnerabilities taking also into account their severity.

Such strategy allows us to dynamically control (and to reduce) the number of forever-day vulnerabilities and their severity by the active and dynamic configuration of the deployment environment. This helps the architects to decrease the risks of malicious attacks and intrusions. The intrusion-avoidance architecture mainly relies on the cross-platform technologies like Java, Python, Perl, etc. and the IaaS cloud services providing the crucial support for diversity of the system components, their dynamic reconfiguration and maintenance of the spare configurations. Our simulation using real-life vulnerability statistics shows how the proposed approach can be used to decrease system vulnerability.

This work is aimed to answer a series of related questions, including:

1. How is the traditional dependability model changed to fit distributed nature of the Internet and Cloud systems?
2. How does uncertainty exhibit itself in the distributed internet and cloud systems?
3. What are the main features of the time-probabilistic failure models?
4. How do time-out settings affect system dependability and latency and allow to interplay between them?
5. What are fundamental trade-offs between consistency, availability and latency in resilient internet and cloud computing systems?
6. How to estimate system response time depending on the chosen consistency level?
7. How do dependability parameters (probabilities of different servicing outcomes) depend on the system consistency?
8. How does software diversity affect system security?
9. What approach can be used to mitigate risks of a system been intruded?
10. How can cloud computing technologies be used to implement intrusion avoidance approach?

References

- [1] D. Smith and K. Simpson, *Safety Critical Systems Handbook: A Straightforward Guide to Functional Safety, IEC 61508 and Related Standards*, 3rd ed., Oxford: Butterworth-Heinemann, 2004.
- [2] A. Avizienis, J. Laprie, B. Randell and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE*

- Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11-33, 2004.
- [3] A. Gorbenko, A. Romanovsky, O. Tarasyuk and V. Kharchenko, "Dependability of Service-Oriented Computing: Time-Probabilistic Failure Modelling," in *Software Engineering for Resilient Systems, Lecture Notes in Computer Science (LNCS)*, vol. 7527, P. Avgeriou, Ed., Berlin, Heidelberg, Springer-Verlag, 2012, p. 121–133.
 - [4] E. Brewer, "Towards Robust Distributed Systems," in *19th Ann. ACM Symposium on Principles of Distributed Computing*, 2000.
 - [5] S. Gilbert and N. Lynch, "Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services," *ACM SIGACT News*, vol. 33, no. 2, pp. 51-59, 2002.
 - [6] J. Brutlag, "Speed Matters for Google Web Search," 2009. [Online]. Available: http://services.google.com/fh/files/blogs/google_delayexp.pdf.
 - [7] A. Lakshman and P. Malik, "Cassandra: a decentralized structured storage system," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, pp. 35-40, 2010.
 - [8] D. Abadi, "Consistency Tradeoffs in Modern Distributed Database System Design," *IEEE Computer*, vol. 45, no. 2, pp. 37-42, 2012.
 - [9] A. Gorbenko and A. Romanovsky, "Timeouting Internet Services," *IEEE Security & Privacy*, vol. 11, no. 2, pp. 68-71, 2013.
 - [10] A. Gorbenko, V. Kharchenko and A. Romanovsky, "Using Inherent Service Redundancy and Diversity to Ensure Web Services Dependability," in *Methods, Models and Tools for Fault Tolerance: LNCS 5454*, Berlin; Heidelberg, Springer-Verlag, 2009, p. 324–341.
 - [11] O. Tarasyuk, A. Gorbenko, A. Romanovsky, V. Kharchenko and V. Ruban, "The Impact of Consistency on System Latency in Fault Tolerant Internet Computing," in *Distributed Applications and Interoperable Systems, LNCS 9038*, A. Bessani and S. Bouchenak, Eds., Berlin; Heidelberg, Springer-Verlag, 2015, pp. 179-192.
 - [12] L. Strigini and A. Avizienis, "Software Fault-Tolerance and Design Diversity: Past Experience and Future Evolution," in *4th Int. Conf on Computer Safety, Reliability and Security*, 1985.
 - [13] A. Gorbenko, O. Tarasyuk, V. Kharchenko and A. Romanovsky, "Using Diversity in Cloud-Based Deployment Environment to Avoid Intrusions," *Software Engineering for Resilient Systems*, no. LNCS 6968, p. 145–155, 2011.

27 SYSTEM OF SYSTEMS AS AN OBJECT OF SECURITY ANALYSIS

27.1 Definitions of System of Systems

There is no universally accepted definition of the term “System-of-Systems” yet. In the scheme of things (in general), a system of systems (SoS) is a collection of individual systems that come together to form a larger, more complex system which is greater than the sum of its parts. These individual systems can be combinations of people, activities, software, and hardware.

The next section will present some definitions of SoS out of many possible ones:

Definition 1: Systems of systems are large-scale concurrent and distributed systems that are comprised of complex systems [1].

Definition 2: Enterprise system of systems engineering is focused on coupling traditional systems engineering activities with enterprise activities of strategic planning and investment analysis [1].

Definition 3: System of systems integration is a method to pursue development, integration, interoperability, and optimization of systems to enhance performance in future battlefield scenarios [2].

Definition 4: SoS involves the integration of systems into systems of systems that ultimately contribute to evolution of the social infrastructure [3].

Definition 5: In relation to joint warfighting, system of systems is concerned with interoperability and synergism of command, control, computers, communications, and information (C4I) and intelligence, surveillance, and reconnaissance (ISR) systems [4].

Definition 6: Systems of systems exist when there is a presence of a majority of the following five characteristics: operational and managerial independence, geographic distribution, emergent behavior, and evolutionary development [5]

Definition 7: A system-of-systems (SoS) consists of multiple, heterogeneous, distributed, occasionally independently operating systems embedded in networks at multiple levels, which evolve over time [6].

At an elementary level, there are three points common among these different definitions [7]. First, a SoS is a system itself. Second, the systems that compose a SoS are systems, as their name indicates. Lastly, the constituent systems maintain a level of independence before and after joining a SoS [8] and it is this last point that distinguishes a SoS.

However, a practical definition may be that a system of systems is a “supersystem” comprised of other elements that themselves are independent complex operational systems and interact among themselves to achieve a common goal. Each element of SoS achieves well-substantiated goals even if they are detached from the rest of the SoS.

27.2 System of System classifications

So far as many definitions there are a lot of classifications. The base for classification became the SoS development process, architecture structure, and governance mechanism (Table 27.1).

Directed SoS are owned by a single organization and are developed by integrating systems that are also owned by that organization. This means that there can be a single policy making (governance) body within the organization that can direct the implementation of system policies. The system elements may come from different parts of an organization and may be independently managed.

Collaborative SoS are systems where constituent independent systems are owned and governed by different organizations. System of system governance depends on voluntary participation in a governing body. This governing body cannot direct the implementation of policy as this might conflict with the interests of system owners.

Governance is the way in which policies about the management, operation and evolution are set. Governance is not the same as management. Governance is about aims and objectives; management is about realizing these objectives.

Another classification approach is based on SoS architecture structure. If the component systems are architected so that they can be integrated to work together to fulfill a goal, it is a **dedicated SoS**.

Table 27.1. Classification of SoS

Base for classification	SoS type	Description
SoS development process [9]	directed SoS	integrated SoS, built and managed to fulfill specific purposes
	collaborative SoS	the system is developed through the collaboration of its participants. Constituents interact more or less voluntarily to fulfill agreed central purposes
SoS architecture structure	dedicated SoS	the component systems are architected so that they can be integrated to work together to fulfill a goal
	virtual SoS	subsystems are previously existing architectures that are integrated to meet an immediate mission requirement
Governance-based classification [10]	organizational	SoS where the governance and management of the system lies within the same organization or company
	federated	SoS where the governance of the SoS depends on a voluntary participative body in which all of the system owners are represented.
	coalitions	SoS where there are no formal governance mechanisms but where the organizations involved informally collaborate and manage their own systems to maintain the system as a whole.

If subsystems are previously existing architectures that are integrated to meet an immediate mission requirement, it is a **virtual SoS**. Virtual systems have no central governance and the participants may not agree on the overall purpose of the system. Organizations may

participate in the SoS because it meets some immediate need but may withdraw at any time.

Acknowledged SoS [11] have recognized objectives, a designated manager, and resources at the SoS level, e.g., an SoS Engineering (SoSE) team. But the SoSE team does not have complete authority over the constituent-systems. The constituent-systems maintain their independent ownership, objectives, funding, and development approaches. Figure 27.1 illustrates using unidirectional arrows between the SoSE team and the constituent-systems. The unidirectional arrow means that the SoSE team can provide guidance to the constituent-systems but that the constituent-systems are not required to comply with SoSE requests or to formally report to SoSE teams.

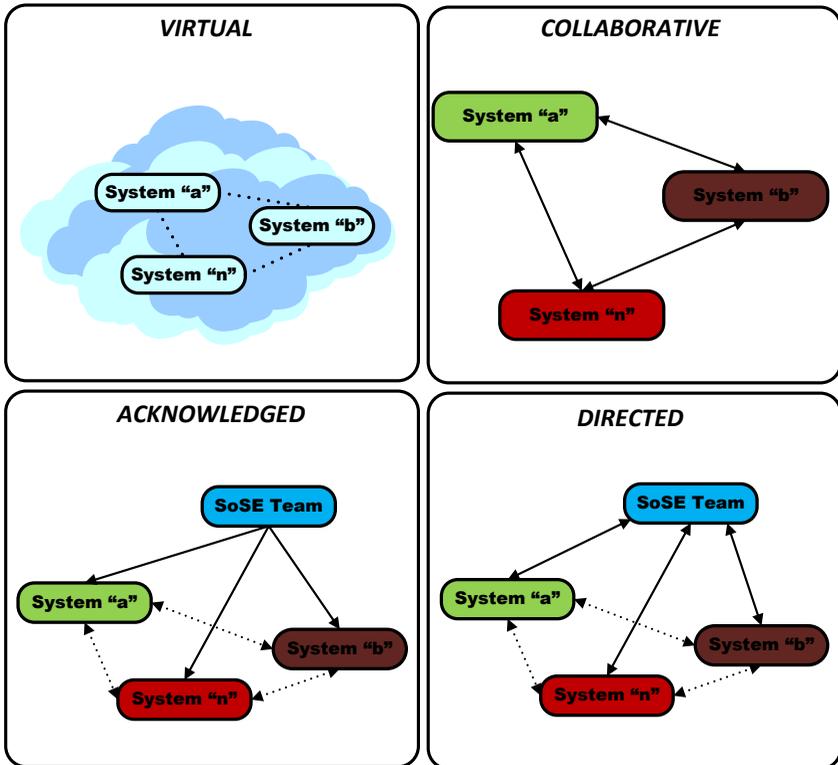


Fig. 27.1: Types of SoS

Organizational systems of systems are SoS where the governance and management of the system lies within the same organization or company. An e-procurement system for a large company that includes an ordering system, an accounting system and an asset management system

Federated systems are SoS where the governance of the SoS depends on a voluntary participative body in which all of the system owners are represented. A disaster management system includes systems from fire, police, and ambulance services.

System of system coalitions are SoS where there are no formal governance mechanisms but where the organizations involved informally collaborate and manage their own systems to maintain the system as a whole. An example is an algorithmic stock trading system that includes individual trading systems from different companies that deal directly with each other.

27.3 Characterization of Systems of Systems

As it mentioned above, Systems of Systems characterized by self-organization, autonomous constituent systems, continuous evolution, scalability and sustainability - provide both economic and social value.

Many characterizations of a system of systems suggest that such systems have the following properties [5]:

1. *Operational Independence of the Individual Systems.* This suggests that a system of systems is composed of systems that are independent and useful in their own right, and if a system of systems is disassembled into the constituent systems, these constituent systems are capable of independently performing useful operations by themselves and independently of one another.

2. *Managerial Independence of the Systems.* This suggests that the component systems generally operate independently to achieve the technological, human, and organizational purposes of the individual organizational unit that operates the system. These component systems are generally individually acquired, serve an independently useful purpose, and often maintain a continuing operational existence that is independent of the larger system of systems.

3. *Geographic Distribution.* Geographic dispersion of the constituent systems in a system of systems is often very large. Often, the individual constituent systems can readily exchange only information and knowledge with one another, and not any substantial quantity of physical mass or energy.

4. *Emergent Behavior.* The system of systems performs functions and carries out purposes that may not reside uniquely in any of the individual constituent systems. The principal purposes supporting engineering of these individual systems and the composite system of systems are fulfilled by these emergent behaviors.

5. *Evolutionary and Adaptive Development.* A system of systems is never fully formed or complete. Development is evolutionary and adaptive over time, and where structures, functions, and purposes are added, removed, and modified as experience of the community with the individual systems and the composite system grows and evolves.

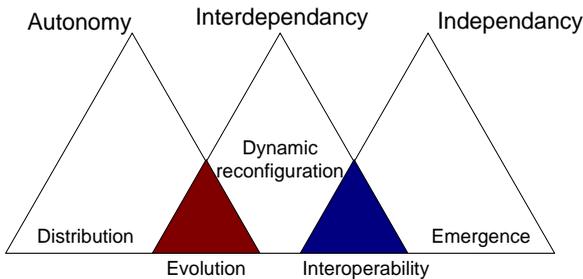


Fig. 27.2: SoS properties

As well as technical complexity, the characteristics of SoS may also lead to significantly increased managerial and governance complexity. Table 27.2 summarises how the different SoS characteristics primarily contribute to different types of complexity:

SoS diversity is an evidence of visible heterogeneity: A SoS should, out of necessity, be incredibly diverse in its capability as a system compared to the rather limited functionality of a constituent system, limited by design. It seems to us that there is a fundamental distinction to be made between requirements-driven design for a conventional system based on its defined scope, and a capabilities-based SoS that must exhibit a huge variety of functions, on an as-

needed basis, in order to respond to rampant uncertainty, persistent surprise, and disruptive innovation [12].

Table 27.2. SoS characteristics and system complexity adapted from [10]

SoS characteristic	Technical complexity	Managerial complexity	Governance complexity
<i>Operational Independence of elements</i>		yes	yes
<i>Managerial Independence</i>	yes	yes	
<i>Geographic Distribution</i>	yes	yes	yes
<i>Emergent Behavior.</i>	yes		
<i>Evolutionary Independence and Adaptive</i>	yes		
Data-intensive	yes		yes
Heterogeneity	yes		

27.4 Architecture and attributes of SoS

A general model of SoS architecture conditionally consists of three hierarchical levels (Fig. 27.3):

1. The low level: system models of single infrastructure.
2. The middle level: interaction model between single infrastructures (so-called as “local system-of-systems”).
3. The high level: global system-of-systems model.

Every system-level acquisition program is required “to identify critical functions and components and manage their risk of compromise” including hardware, software, firmware and information [13], that is, security vulnerabilities of systems are to be addressed as part of systems engineering of the system. This leads to the need of injecting flexibility and adaptability to the system engineering design, to respond to the ever-changing domains of technology, society, economy, legislation and politics, which determine the profiles of service demand and the corresponding expected performance [14]. In this scenario of technologically and structurally evolving (and more and

more interdependent) critical infrastructure (CI), concerns are arising on their vulnerability and risk of failure, i.e. on the danger that [15]:

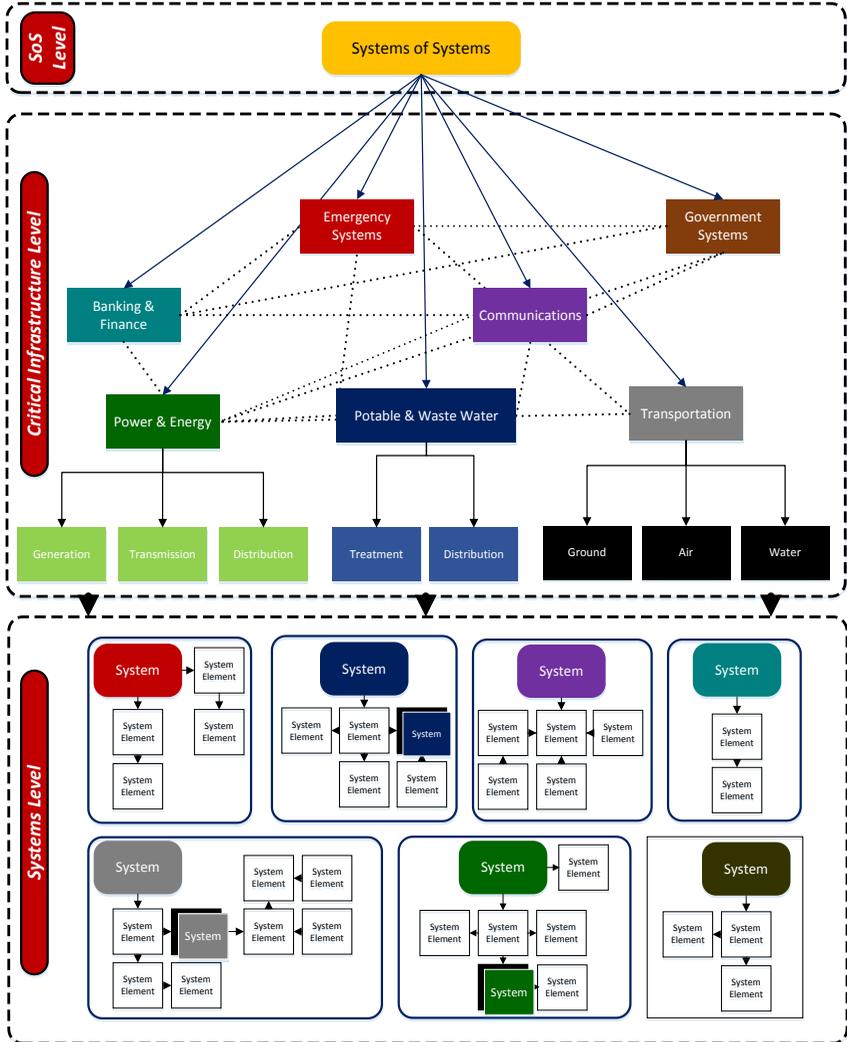


Fig.27.3: System of Systems architecture (adapted from [16])

- the allocated system capacities may not be adequate to support the growing demands in scenarios of greater CI integration and market deregulation;
- the safety margins preventively designed may not be sufficient to cope with the expected and, most of all, unexpected stresses arriving onto the systems.

These issues are difficult to analyze as, due to the SoS complexity, emergent behaviors may arise at system level from the collective response of the elementary components, in ways difficult to predict and manage. More complexity means more people involved, more parts, more interactions, more mistakes in the design and development process, more of everything where hidden insecurities can be found. A complex system means a large attack surface.

The problem is how to deal with the complexity? How to secure such systems when they composed of variety traditional systems and distinguished by the different dynamic properties and when SoS architectures do not typically include security considerations?

Here we set out to analyze a special SoS category, most vulnerable to cyber crimes, Cyber-physical Systems of Systems.

In real world, Cyber-physical Systems (CPS) and SoS have a lot common, both of them have many interactions components and a lot of physical connections. But CPS represents the large, complex, often spatially distributed system with tight interaction of many real-time computing systems and physical systems [17].

Examples of Cyber-physical Systems:

- Airplanes,
- Cars,
- Ships,
- Buildings with advanced HVAC controls,
- Manufacturing plants,
- Power plants.

In the same way real SoS are actually much more complex than the CPS, their specific characteristics includes (a) dynamic reconfiguration (it means that components may be switched on and off (as in living cells), enter or leave SoS (e.g. air traffic control); (b) continuous evolution (continuous addition, removal, and modification of hardware and software over the complete life cycle (often many

years); (c) partial autonomy (as a rule, SoS has a lot of local actors with local authority and priorities. That is, the autonomous systems cannot be fully controlled on the SoS level, they need incentives towards global SoS goals); (d) emerging behavior (the overall SoS shows behaviors that do not result from simple interactions of subsystems. Usually not desired in technical systems, may lead to reduces performance or shut-downs, e.g. power oscillations in the European power grid, oscillations in supply chains, etc.).

Examples of SoS include:

- Smart power grid with power plants and power distribution and control,
- Smart transport systems (rail, traffic management with V2V and V2I facilities for highly automated or autonomous driving, air traffic control systems),
- Advanced manufacturing systems (industry 4.0),
- Mobile co-operating autonomous robotic systems or vehicles,
- Health-care systems,
- Smart buildings and neighbourhoods - from local communities through to smart cities.

Both, CPS and SoS in their generalities turns into particular class of Cyber-physical Systems of Systems (CPSoS), see Table 27.3.

Table 27.3. Similarities and differences in CPS and SoS

Cyber-physical Systems (CPS)	Systems of Systems (SoS)	
	Cyber-physical SoS	
Tight interaction of many distributed real-time computing systems and physical systems	Many interaction components	Dynamic reconfiguration
	Physical connections - material/energy streams - shared recourses (roads, airspace, rails) - communication networks	Continuous evolution
		Partial autonomy
		Emerging behavior

These fusion stands on the sharpest edge of CPS and SoS. In this context, the constituent systems to be considered are not only the

complex ICT systems themselves, but also Cyber-physical systems, i.e. embedded ICT systems with strong relationship to physics, mechatronics and the notion of interaction with each other and with an unpredictable environment. The result may be ‘emergent properties’ - unforeseen or unpredicted behavior that may have critical effects. CPSoS must be adaptable, reconfigurable and extendable during their lifetime, since the classical predictability assumptions of safety and cyber-security assessment and certification no longer hold [18].

Examples of Cyber-physical Systems of Systems:

- Integrated large production complexes
- Transportation networks (road, rail, air, maritime, etc.)
- Large networks of systems (electric, grid, traffic systems, water distribution)
- Smart (energy, water, gas, etc.) networks, supply chains or manufacturing.

To answer the question how to secure such systems we need refer to their composition. In this context we should keep in mind that SoS are heterogeneous and independently operable systems that are networked together for a common goal. (Fig. 27.4).

That is the behavior of the overall SoS depends not only on that of the single systems, but also on the interactions between the constituent systems. SoS conceptual framework identifies three components:

- physical networks such as roads and power grids,
- information networks such as Intranets and databases, and
- social networks such as people, organizations, and processes.

In any SoS there is the potential for interactions between systems to occur that affect the security of the overall system. You already know that SoS components can operate independently when separated from the main system.

The components can also maintain their existence independent of the SoS and as a result of these features, SoS fulfills a common purpose as well as additional purposes of the individual subsystems. This lead to fact that degradation occurring in systems due to an attacks may not critically damage the system itself but it may cause the emergence of critical loss of operability elsewhere in the interconnected SoS.

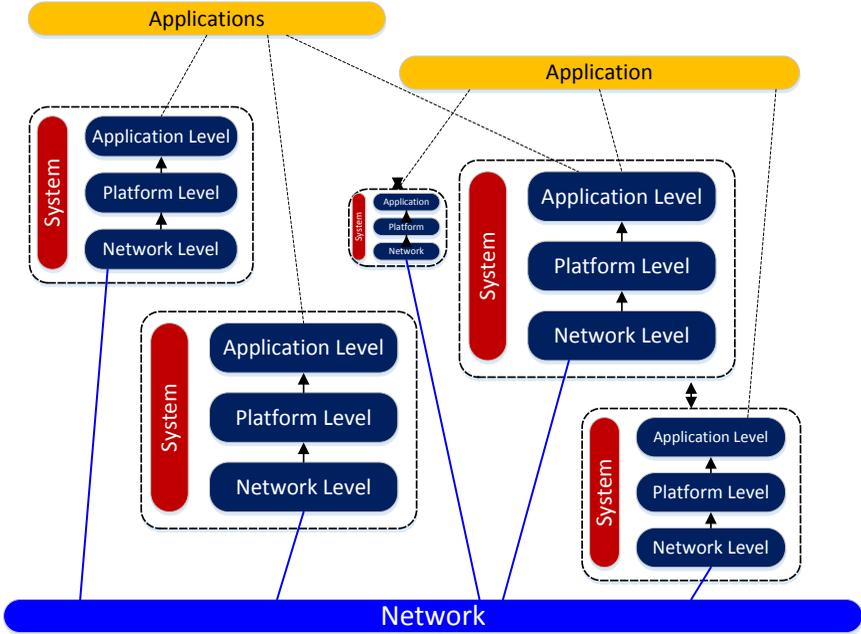


Fig. 27.4: Composition of systems into SoS

For this reasons, the SoS behavior, when a component systems is subject to a cyberattack, can be very different from the behavior of the system under attack, in terms of robustness and reliability.

The consequences of attacks on the SoS cannot be understood by means of the merely evaluation of the behavior of the separate system, but require an assessment of the effect of the interdependencies on the behavior of the whole SoS.

27.5 Interdependencies in SoS

System of systems can be defined in terms of interdependence attribute where a set or arrangements of interdependent systems are connected to provide a given capability. Interdependent infrastructures can correspond to one infrastructure (internal interdependency) or more infrastructures (external interdependency) [19].

Table 27.4. SoS dimensions [20]

		← System of Systems Dimensions →				
	Level	Resources	Operations	Economics	Policy	
Base level	α ($\nu 10^6$)	Vehicles and Infrastructure (e.g. aircraft, ATC facility)	Operating a resource (e.g. pilots, crew, maintenance)	Economics of building / operating / buying / selling / using a single resources	Policies relating to single use (e.g. type certification, flight procedures, etc.)	
Network of Networks	↑	β ($\nu 10^4$)	Collection of resources for a common function (e.g. airport)	Operating resource networks for common function (e.g. airline)	Economics of operating / buying / selling / leasing resource networks	Policies relating to multiple vehicle use (e.g. airport traffic management, noise policies)
	γ ($\nu 10^2$)	Resources in a transport sector (e.g. air transportation)	Operating collection of resource networks (e.g. commercial air operators)	Economics of a business sector (e.g. airline industry)	Policies relating to sectors using multiple vehicles (safety, accessibility, etc.)	
	δ ($\nu 10^1$)	Multiple, interwoven sectors (resources for a national transportation system)	Operations of multiple business sectors (i.e. operators of total national transportation system)	Economics of total national transportation system (all transportation companies)	Policies relating to national transportation policy	
	↓	ϵ ($\nu 10^0$)	Global transportation system	Global operations in the world transportation system	Global Economics of the world transportation system	Policies relating to the global transportation system

Dependencies and interdependencies between the elements of SoS are an important source of risk and risk uncertainty [21].

According to the [22] four types of interdependencies are identified for critical infrastructures:

- Physical: the operation of one infrastructure depends on the material output of the other.
- Cyber: dependency on information transmitted through the information infrastructure.
- Geographic: dependency on local environmental effects that affects simultaneously several infrastructures.
- Logical: any kind of dependency not characterized as physical, cyber or geographic.

All types of interdependencies may provide the tolerance to attacks and failures if well managed (positive impact). *For instance*, technical failures such as abnormal disconnection of a transmission line can be detected by remotely installed devices at a substation and corresponding alarms can be transmitted to a control center via services provided by coupled telecommunication systems in order to prevent further failure propagations. However, these interdependencies might also be a source of threat generating risks, e.g., the risk of cascading failures, which make infrastructures more vulnerable (negative impact). In power blackout events, service disruptions further propagate to other infrastructures (transportation, telecommunication and water supply) and worsen the overall negative impacts [23]. Even though not preventable, these damages may be minimized, if the capabilities of both direct and indirect affected infrastructures are strengthened and effects of interdependencies are recognized [24, 25].

It is important to understand if interdependencies are essential, e.g. the dependency of power grids on the control system, or “parasitic”, e.g. the dependency of the control system on the controlled grid. The latter can be removed or redesigned. Moreover, the human/social element is recognized to play a key role in the operations of infrastructures. To better understand the performance of infrastructures, especially their behavior during and after the occurrence of disturbances (e.g., natural hazards or technical failures), resilience analysis [21, 26, 27, 28, 29] has grown as a proactive approach to

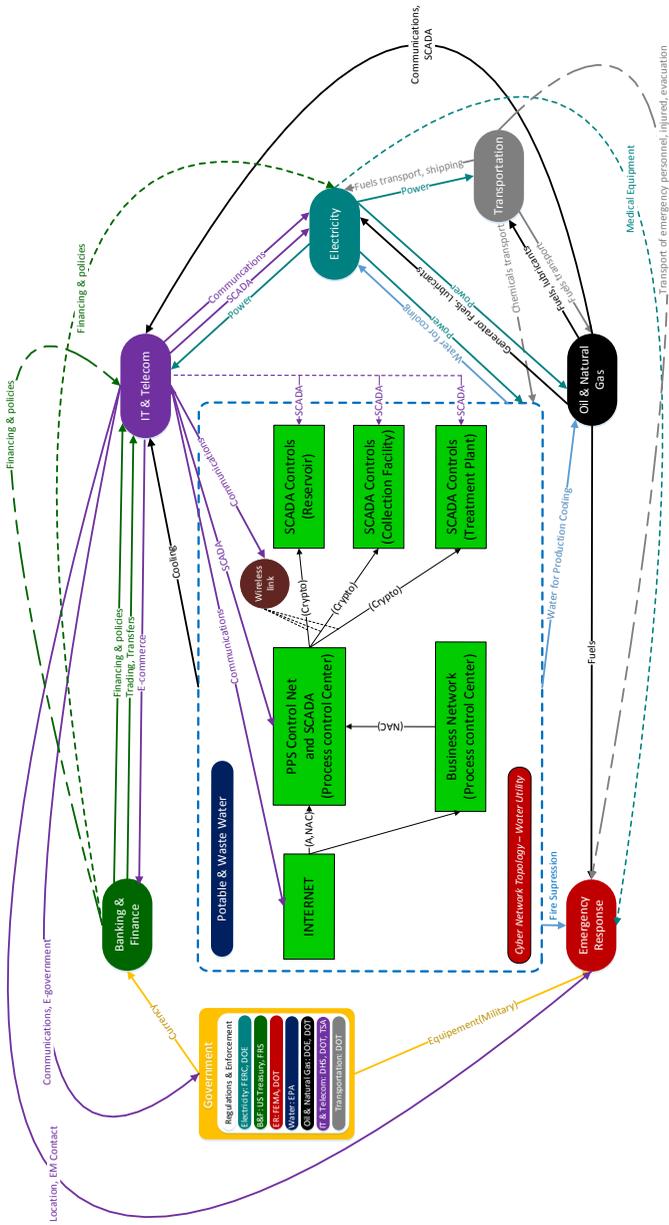


Fig. 27.5: The potable and waste water CPSoS dependencies

enhance the ability of infrastructures to prevent damage before disturbance events, mitigate losses during the events and improve recovery capability after the events, by extending the concept of pure prevention and hardening. Interdependencies among infrastructures dramatically increase the overall complexity of the “systems of systems”. There is therefore a need to consider multiple interconnected infrastructures and their interdependencies in a holistic manner. We distinguish three main approaches of interdependencies [22]:

First the focus can be laid on one critical infrastructure system and on the others critical infrastructures systems it is depending on. Figure 27.5 shows for instance the critical infrastructures the electric power infrastructure is depending on.

In the figure, waste water CPSoS is the supported infrastructure where natural gas, oil, transportation, telecommunications, electric power, banking, and finance are supporting infrastructures.

The second approach, on the contrary can focus on one critical infrastructure system and on others critical infrastructures systems that are depending on the services provided by the system under focus.

Finally, the last approach aims at embracing a whole system of various critical interdependent infrastructures interacting with each others. Figure 27.6 gives an example of some existing interdependencies existing among various critical infrastructures systems.

Pervasive connectivity moves into the safety critical domain:

- by including actuation,
- by penetrating safety critical systems,
- uncertainty and concerns of connectivity and scalability are complemented with timeliness and dependability.

In the view of vulnerability and risk analysis, it is necessary to determine for each infrastructure:

- Which other infrastructure it depends on continuously or nearly continuously for normal operations,
- Which other infrastructures it depends on during times of high stress or disruptions,
- And which it depends on to restore service following the failure of a component or components that disrupt the infrastructure.

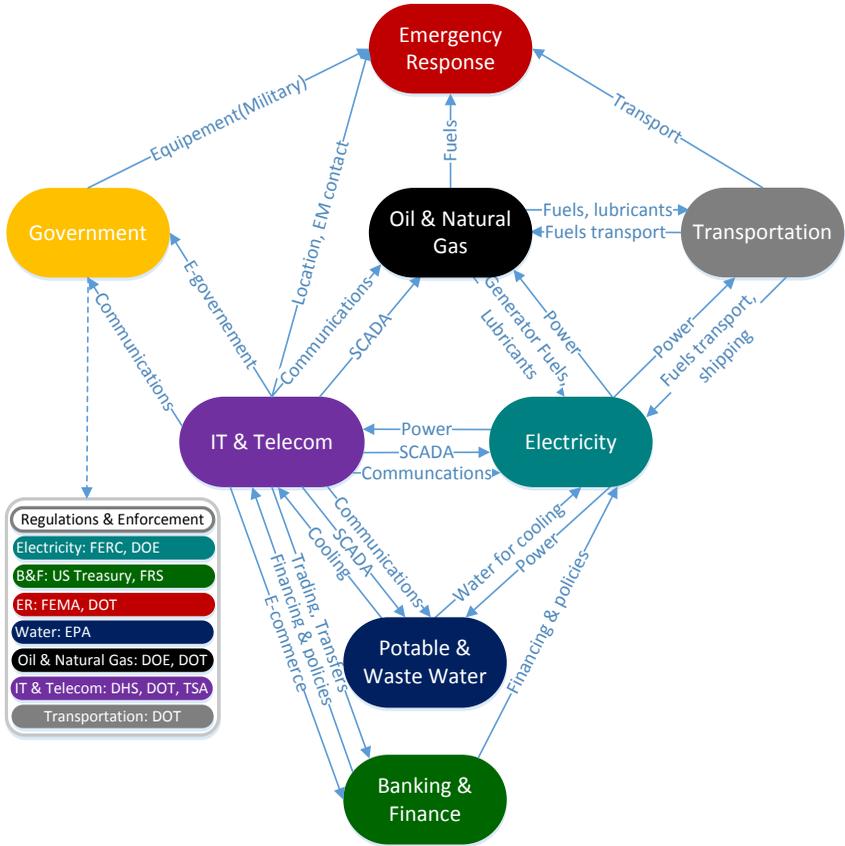


Fig. 27.6: A system of critical infrastructures systems and their interdependencies adapted from [22]

For instance, under normal operating conditions the electric power infrastructure requires natural gas and petroleum fuels for its generators, road and rail transportation and pipelines to supply fuels to the generators, air transportation for aerial inspection of transmission lines, water for cooling and emissions control, banking and finance for fuel purchases and other financial services, and telecommunications for e-commerce and for monitoring system status and system control. During emergencies or after components failures the electric power

infrastructure will have potentially different yet critical dependencies on the same infrastructures.

For example, the utility may require petroleum fuels for its emergency vehicles and emergency generators and road transportation to dispatch repair crews and replacement components [22].

However, the interdependencies in SoS lead to the comprehension that interaction space is not that clear - the domain can be more logical than physical. That why security is an enabling technology in this emerging field because without security those systems would not be possible at all. Only one framework could possibly address so many things: risk management. Current analytical techniques to protection of systems are based on a methodology which identifies critical components of the system; their risks to persistent threats and vulnerabilities; and options for countermeasures to address the risks.

But risk management does not work in unpredictable environments. The first three steps to rolling out a risk management program are to

1. Identify all critical assets.
2. Identify their vulnerabilities.
3. Prioritize them based on risk.

Some relatively common issues for SoS risk management [12] are given in Table 27.5., they provide a starting point to applying the own risk management programs.

Table 27.5. Common SoS Risk Management Issues [30]

No	Issue	Issue Summary
1	Multiply stakeholders	Differences in stakeholder's behaviors will often lead to contention and potentially sub optimal design solutions, funding allocation, schedule priority, and increased risk
2	Multiply risk management processes	Differences in risk management process and their implementation can lead to the omission of risks as well as exaggeration of other risks.
3	Long life cycles	Non-uniform acquisition maturity potentially complicates risk management.
4	Common technical risk	Technical risks are often examined, evaluated, and managed separately, which may not

	classes	provide insight into potential strengths / surpluses and weaknesses / shortfalls.
5	Integration risk	Integration risks is often not explicitly evaluated.
6	Functional performance risk	Functional performance risk is often not explicitly evaluated.
7	Interface complexity	It is generally difficult to evaluate interface complexity and accurately relate it to risk.

27.6 What components of SoS are at cybersecurity risk?

Virtually any element of cyberspace can, at least in theory, pose some level of cybersecurity risk, which is generally thought of as a combined assessment of threat, vulnerability, and impact that gives a measure of the overall potential for harm from vulnerability if no corrective action is taken [31].

There appear to be certain candidate components of cyberspace and associated activities that are sources of potentially significant risk because either major vulnerabilities have been identified or substantial impacts could result from a successful attack. They are

- Components that play critical roles in elements of critical infrastructure. This could include, for example, computer control systems such as SCADAs used in the chemical and energy industries, and the Internet infrastructure. Another example is information held by financial services industries that could be stolen electronically or otherwise compromised.

- Software. In particular, widely used computer programs such as operating systems can be vulnerable to various forms of compromise resulting, for example, in information theft or use of the compromised system as a weapon of attack. This kind of vulnerability has perhaps received more public attention than any other, given that it affects virtually all owners and users of desktop systems.

- Cybersecurity governance. Many observers have expressed concerns that corporations and other organizations, including some involved in critical infrastructure sectors (see below), have not developed governance mechanisms sufficiently responsive to

cybersecurity needs. Weaknesses have been cited with respect to several aspects of cybersecurity governance, including policies, procedures, and personnel management.

– Public knowledge and perception. Observers who have expressed concern about the risk of major cyberattacks from terrorists or other criminals have in many cases pointed to a lack of public awareness about the risk as a weakness, both with respect to lack of knowledge about the steps individuals need to take to defend against attacks and the need for national public- and private-sector effort.

Three modes of malicious attacks on critical infrastructure are generally envisioned:

1) Attacks upon the system: The system itself is the primary target with ripple effects throughout society,

2) Attacks by the system: The population is the actual target, using parts of the system as a weapon,

3) Attacks through the system: The system provides a conduit for attacks on other critical infrastructures.

The strategy to increase cybersecurity in SoS is twofold: cyberattacks can be reduced or contained (thus making the attacks unsuccessful), or the impacts of successful cyberattacks can be reduced, by improving the resilience of systems to such attacks. One fundamental requirement to achieve the latter step is risk evaluation and impact assessment [32]. This evaluation must also account for the possible cascading effects that cyberattacks can have on the operability of a system.

Conclusions

The modern society of today is highly dependent on the network of large infrastructure systems that provide essential services to its inhabitants, including energy, transport, communication, financial, production, emergency, and other services that support day-to-day activities.

‘Systems-of-systems’ is a relatively new term for systems that are composed of independent (autonomous) subsystems that are full-blown systems by themselves in every way. The main purpose of SoS is to provide new services, but with highly interacting and interdependent

ICT systems relying on critical infrastructures, new threats and challenges arise. The increased complexity of analyzing an SoS requires an especially clear understanding of the SoS as a critical prerequisite to the application of these approaches.

There are some trends affect on security race in SoS:

- SoS security becomes everything security. That means that all the things we understand from patching and vulnerabilities to security vs. complexity to network effects become relevant to everyone / everything.

- For many reasons, like complexity the attacks on SoS is easier than defense,

- The more connections meant the more vulnerabilities in one system can affect another,

- New vulnerabilities arise in the interconnections,

- More critical systems mean more power to attackers

- Internet allows criminals to scale and allows attacks from anywhere / everywhere.

- Degradation occurring in systems due to an attacks may not critically damage the system itself but it may cause the emergence of critical loss of operability elsewhere in the interconnected SoS

- An SoS security risk framework is needed to manage the problem of identifying the key elements of risk to SoS

- In cybersecurity area Technology and Law must work together or both will fail.

Questions to self-checking

1. Give definition and properties of System of System.
2. What are the main differences between CPS and SoS?
3. Explain why managerial and operational independence are the key distinguishing characteristics of systems of systems when compared to other complex systems.
4. What specific characteristics of SoS?
5. How the different characteristics of SoS contribute to different types of complexity?
6. Identify three components of SoS conceptual framework.
7. What SoS type is more fragile to critical loss of operability?

8. Is any SoS category, most vulnerable to cyber crimes?
9. Explain how you understand the term ‘emergent properties’.
10. How managerial independence of the systems affect their security?
11. What main approaches of interdependencies?
12. How interdependencies in SoS can affect on their security?
13. When interdependencies may provide the tolerance to attacks?
14. What components of SoS are at cybersecurity risk?
15. What the modes of malicious attacks on critical infrastructure?
16. How to increase cybersecurity of SoS?
17. What approach is essential to manage the problem of SoS security?

References

1. Carlock, P.G., Fenton, R.E., 2001, Systems of systems (SoS) enterprise systems engineering for information-intensive organizations, *Systems Engineering*, 4(4): 242–261.
2. Pei, R.S., 2000, Systems of systems integration (SoSI)—a smartway of acquiring Army C4I2WS systems, *Proceedings of the Summer Computer Simulation Conference*, pp. 134–139.
3. Luskasik, S.J., 1998, Systems, systems of systems, and the education of engineers, *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, 12(1): 11–60.
4. Manthorpe, W.H., 1996, The emerging joint system of systems: a systems engineering challenge and opportunity for APL, *John Hopkins APL Technical Digest*, 17(3): 305–310.
5. Jamshidi, M., 2005, System of systems engineering definitions, *Proceedings IEEE Systems, Man, and Cybernetics Conference*, October, Waikoloa, HI http://ieeesmc2005.unm.edu/SoSE_Defn.htm.
6. DeLaurentis D. Role of humans in complexity of a system-of-systems. In: Duffy VG, editor. *Digital Human Modeling*. Berlin: Springer-Verlag; 2007. p. 363–71.
7. Baldwin, W.C., Sauser, B. Modeling the Characteristics of System of Systems http://boardmansauser.com/Systemomics/Publications_files/IEEESoSE1569199244Final.pdf

8. Shah, Nirav B., Donna H. Rhodes and Daniel E. Hastings. 2007. Systems of systems and emergent system context. Paper 85 presented at the 5th Conference on Systems Engineering Research, March 2007, Hoboken, NJ. http://web.mit.edu/adamross/www/SHAH_CSER07.pdf [accessed February 2, 2009].
9. Maier, M. 1998. Architecting principles for systems-of-systems. *Systems Engineering* 1, no. 4: 267-284.
10. Sommerville, Ian 2014 <http://iansommerville.com/software-engineering-book/files/2015/08/Ch-20-Systems-of-systems.pdf>
11. Dahmann, J. and K. Baldwin. 2008. Understanding the current state of US defense systems of systems and the implications for systems engineering. Proceedings of the IEEE Systems Conference, April 7- 10, in Montreal, Canada.
12. Conrow, Edmund H. "Risk Management for Systems of Systems." 2004 Systems and Software Technology Conference, Salt Lake City, UT, 21 Apr. 2004.
13. Protection of Mission Critical Functions to Achieve Trusted Systems and Networks, Department of Defense Instruction 5200.44, 2012.
14. Tagarev, T., Georgiev, V., & Ivanova, P. (2012). Analytical Support to Critical Infrastructure Protection Policy and Investment Decision-Making. *Information & Security*, 28(1), 13.
15. Zio, E. Challenges in the vulnerability and risk analysis of critical infrastructures / *Reliability Engineering and System Safety* 152 (2016) 137–150.
16. Schmitz W, Neubecker KA. Architecture of an Integrated Model Hierarchy, vol. Final Report, ACIP, 2003.
17. Balasubramaniyan, S., Srinivasan, S., Buonopane, F., Subathra, B., Vain, J., & Ramaswamy, S. (2016). Design and verification of Cyber-Physical Systems using TrueTime, evolutionary optimization and UPPAAL. *Microprocessors and Microsystems*, 42, 37-48
18. D. Schneider, E. Schoitsch, E. Armengaud: "Towards Trust Assurance and Certification in Cyber-Physical Systems"; in *Computer Safety, Reliability and Security*, 33rd International Conference, SAFECOMP 2014, Springer, LNCS 8696, pp. 180- 191.

19. Nan, C., Sansavini, G. A quantitative method for assessing resilience of interdependent infrastructures / Reliability Engineering and System Safety 157 (2017) 35–53.

20. DeLaurentis, D. (2015). Security in a System of Systems Context: Insights from Recent Initiatives Panel: Security for Energy Infrastructures

http://www.purdue.edu/discoverypark/energy/assets/pdfs/brc/Panel%201/Dan%20DeLaurentis_Energy%20Workshop%20slides.pdf

21. Netkachov O., Popov P., Salako K. Model-Based Evaluation of the Resilience of Critical Infrastructures Under Cyber Attacks / Resilience of Critical Infrastructures. Springer Intern.Publ. Switzerland 2016 C.G. Panayiotou et al. (Eds.): CRITIS 2014, LNCS 8985, pp. 231–243, 2016.

22. Rinaldi S.M., Peerenboom J.P., Kelly T.K. (2001). Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies / IEEE Control System Magazine, 0272-1708/01, Washington, December 2001

23. US – Canada Power System Outage Task Force. Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations; 2004.

24. Brummitt CD, D’Souza RM, Leicht EA. Suppressing cascades of load in interdependent networks. Proc Natl Acad Sci 2012;109:E680–9.

25. Buldyrev SV, Parshani R, Paul G, Stanley HE, Havlin S. Catastrophic cascade of failures in interdependent networks. Nature 2010;464:1025–8.

26. Fisher R, Bassett G, Buehring W, Collins M, Dickinson D, Eaton L, et al. Constructing a Resilience Index for the Enhanced Critical Infrastructure Protection Program. Decision and Information Sciences Division, Argonne National Laboratory, Department of Energy, United States; 2010.

27. Linkov I, Creutzig F, Decker J, Fox-Lent C, Kröger W, et al. Risking resilience: changing the paradigm. Nat Clim Change 2014;4:407–9.

28. Madni AM, Jackson S. Towards a conceptual framework for resilience engineering. IEEE Syst J 2009;3:181–91.

29. Ouyang M, Dueñas-Osorio L, Min X. A three-stage resilience analysis framework for urban infrastructure systems. *Struct Saf* 2012;36–37:23–31.

30. Edmund H. Conrow Risk Management for Systems of Systems / *The Journal of Defense Software Engineering* 2005 // <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.636.6840&rep=rep1&type=pdf>

31. Aniello L., Bondavalli A., Ceccarelli A., Ciccotelli C., Cinque M., Frattini F., Guzzo A., Pecchia A., Pugliese A., Querzoni L., Russo S. (2014). Big data in critical infrastructures security monitoring: Challenges and opportunities. arXiv preprint arXiv:1405.0325. <https://pdfs.semanticscholar.org/de55/a260f6c4203b9c6853eae89050bb8b84dff6.pdf>

32. Netkachov, O., Popov, P., & Salako, K. (2014, September). Quantification of the impact of cyber attack in critical infrastructures. In *International Conference on Computer Safety, Reliability, and Security* (pp. 316-327). Springer International Publishing.

Summary

When you have read this chapter, you will: understand what is meant by a system of systems and how this differs from an individual system; understand systems of systems classification, architecture, and the differences between another types of complex systems; understand why interdependencies can both provide the tolerance to attacks and cause the emergence of critical loss of operability; define what components of SoS are at cybersecurity risk; have been introduced to the modes of malicious attacks on critical infrastructure.

28 METHODS AND TECHNIQUES OF SECURITY AND RESILIENCE SoS RISK ANALYSIS

28.1 Basic concepts associated with security risk in SoS

Systems of systems (SoS) consist of dynamic coalitions of systems and services that collaborate to achieve a common goal. Examples of such coalitions include Web Services, Mobile Ad-hoc Networks (MANETs), air traffic control systems, etc [1]. As it mentioned in [2] when engineering traditional systems, the tools and methodologies available are sufficient to provide a solution to a defined problem; the analysis conducted is dominated by technological components; and scoping and framing the problem is easy, since the boundaries are fixed. However, when dealing with SoS, the boundaries become fluid, there is no one right way of dealing with the problem at hand since it is emergent, and engineering these systems of systems becomes a satisficing issue, rather than optimizing [3]. SoS brings new opportunities and new risks. The discussion of risk within the SoS context, therefore, becomes more important and difficult at the same time.

The security process aims at managing risks in accordance with the system's objectives. In general, managing risk is the assumption and convergence of three basic requirements: full information, independent trials and the relevance of quantitative valuation. Since total security is not attainable, this means that our limited resources have to be used efficiently and with purpose.

Risk management is the program and supporting processes to manage information security risk to organizational operations (including mission, functions, image, reputation), organizational assets, individuals, other organizations, and the Nation, and includes: (i) establishing the context for risk-related activities; (ii) assessing risk; (iii) responding to risk once determined; and (iv) monitoring risk over time (fig. 28.1). The essence of risk management lies in the maximizing the areas where we have some control over the outcome, while minimizing the areas where we have absolutely no control over the outcome and linkage between effect and cause is hidden from us. Risk

management is the process that helps us to protect our critical assets and operations with proportional, coherent, and verifiable measures, thus a balanced cost/benefit ratio. This process is a crucial tool in the decision-making process; it allows us to conscientiously make trade-offs, state our security posture, and choose the appropriate measures.



Figure 28.1: The risk management process

Risk analysis and management focus on hostile effects of known unknowns [4]. Since security is not a static state that is present or not present, we ought to define security levels as a continuous cycle that constantly changes over time. Without a proactive approach to security, the levels of security would rapidly decrease over the lifetime of the system. Also, it should be mentioned that products and technologies alone cannot solve security problems they can only provide security

when used efficiently, through consistent and thoroughly defined processes. In this context we can identify two such processes:

- the business continuity planning, which defines how to recover after a disruption or disaster and how to restore the critical functions in order to keep the business going [5],

- the incident management, which describes how to log, record, and resolve security incidents, including legal aspects and evidence management. It is certain that security incidents will occur; we just don't know when they will take place. Therefore, we must anticipate how such incidents will be handled [6]. According to ISO/IEC 13335 security standard “system security consists of defining, achieving, and maintaining the following properties: confidentiality, integrity, availability, non-repudiation, accountability, authenticity, and reliability” [7]. Safety and security in SoS are strongly interdependent and have to be considered together.

Security	Safety
Availability	Availability
Integrity	Integrity
Confidentiality	Reliability
Maintainability	Maintainability
Auditability	
Imputability	
Business criteria	

Figure 28.2: Some common characteristics of safety and security

Authors [8] distinguish at least four types of interdependencies between safety and security: (1) conditional dependencies – security is a condition for safety and vice versa; (2) reinforcement – safety and security countermeasures can strengthen each other; (3) antagonism – they can weaken each other; and (4) independence – no interaction between safety and security.

Recent security and safety standards, such as EBIOS (DCSSI - France) [9, 10], ISO 27005:2011 (IEC - International) [11]; MAGERIT (Ministry of Public Administration - Spain) [12]; OCTAVE (SEI Carnegie Mellon University - USA) [13]; IT Baseline Protection Manual (BSI - Germany) [14]; NIST SP800-30 (NIST - USA) [15] for

security and ESARR [16]; CRAMM (Siemens Insight Consulting - UK) [17]; for safety, show that safety and security ontologies are mostly based on the same concepts and same phases and they can be summarized as follows [18]:

- Identification of the hazards, at the functional level, or capacity level, of the system. The hazards describe, in a generic way, failure modes that impair the safety or security of the system and its environment.

- Identification of the effects (or consequences) of the hazards and estimation of the severity of the effects.

- Identification of the possible causes (safety ontology), or threats scenarios (security ontology) that may induce the hazards, along with their probabilities/frequencies of occurrence (safety ontology), or likelihood (security ontology).

- Then the concept of risk comes naturally from the combination of identified severities and identified probabilities or likelihoods.

SoS risk management approach can be applied to all threats and hazards, including cyber incidents, natural disasters, man-made safety hazards, and acts of terrorism, although different information and methodologies may be used to understand each.

Criticality and risk assessment for multi-infrastructure systems need adapted concepts, definitions and models for these joint systems. The adapted solutions should take into consideration mathematical behavior and the flows causing the interdependencies between the involved infrastructures. Similar to the safety lifecycle, the cybersecurity lifecycle has an assess phase, analysis phase, implementation phase, and operational phase [19].

There are also several activities involved across all phases. We will focus here on the first few stages, particularly on most popular cyber security risk assessment methodologies. Included are also high level risk assessment methods, used in safety area like HAZOP, FMEA and others.

The reason is the traditional approaches, successful in the treatment of safety and reliability issues, among many others, are specified, standardized, and integrated in the SoS landscape as well as can be combined with new security techniques and approaches successfully.

28.2 Classification of the main risk analysis and assessment methodologies

Traditionally, the risk analysis and assessment (RA) techniques are classified into three main categories: (1) the qualitative, (2) the quantitative, and (3) the hybrid techniques (qualitative-quantitative, semi-quantitative). This classification this is very relative and subjective. Fig. 28.3 illustrates the classification of the main risk analysis and assessment methodologies presented in this chapter. Seven integrated safety and security risk assessment methods are represented on the bottom as a part of big picture.

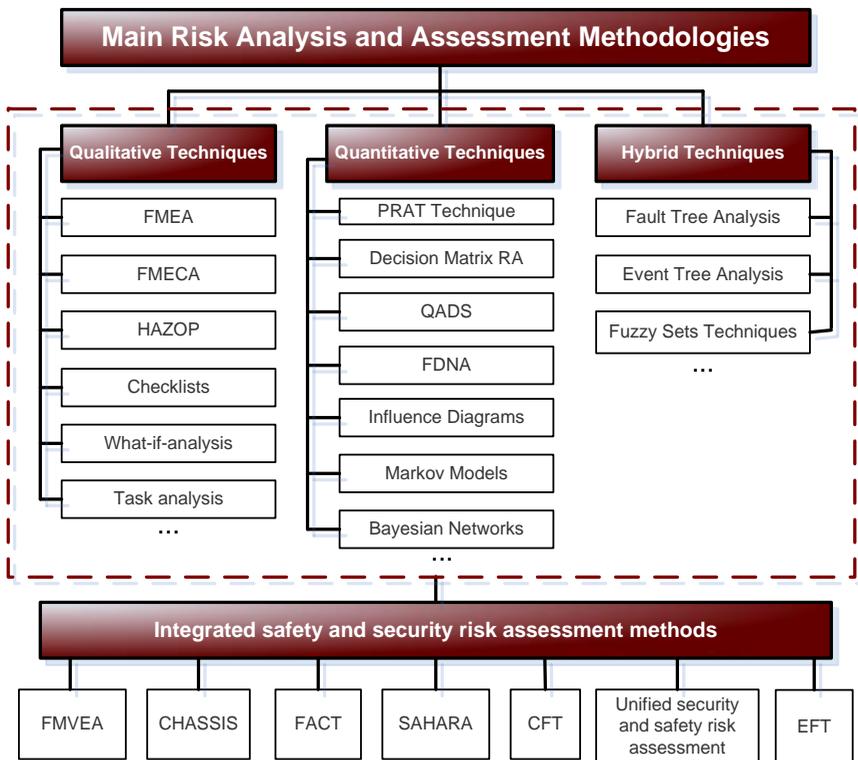


Fig. 28.3 Classification of the main risk analysis and assessment methodologies (adopted from [20])

Qualitative techniques are based both on analytical estimation processes, and on the safety managers-engineers ability. The qualitative risk analysis methodology uses several elements that are interconnected: threats, vulnerabilities, and controls and allows covered entities to assess all potential impacts, whether they are touchable or untouchable. Qualitative methods rate the magnitude of the potential impact of a threat as high, medium, or low. They are the most common measures of the impact of risks. According to quantitative techniques, the risk can be considered as a quantity, which can be estimated and expressed by a mathematical relation, with the help of real accidents' data.

Quantitative risk analysis uses two basic elements: the probability of an event occurring and the losses that may be incurred. Quantitative risk analysis uses one number produced from these elements. This is called the Expected Annual Loss (ALE) or Estimated Annual Cost (EAC). This is calculated for an event by simply multiplying by the probability of potential losses. Therefore, in theory, one may rank events in order of risk (ALE) and make decisions based on that risk. The problem with this type of risk analysis is usually associated with the unreliability and inaccuracy of data. Probability can rarely be accurate and can, in some cases, promote complacency. In addition, control and action steps that often deal with a number of potential events and the events themselves are often inter-related.

The analyses to be performed with **Hybrid techniques** like on Fault Trees can be either qualitative or quantitative. Qualitative analyses show, for instance, which combinations of failures must occur together to cause a top-level failure. Quantitative analysis, on the other hand, calculates the probability of the top event occurring from the probabilities of the basic events. It is important to know that most calculation rules for the probabilistic analysis depend on the assumption that all events are stochastically independent of each other [21]. Hybrid techniques present a great complexity due to their ad hoc character that prevents a wide spreading.

Integrated safety and security risk assessment techniques can consolidate all approaches mentioned above in different combinations depending on system requirements. It is assumed that the combination of the different mindsets and engineering approaches of safety

engineers and security engineers, which are able to work independently from another and mutually benefit from each other's findings, are more likely to be result in higher maturity of analysis.

28.2.1 Qualitative techniques

a) Family of Failure Modes and Effects Analysis (FMEA)

The FMEA technique [23], along with its close relatives, failure modes, effects and criticality analysis (FMECA), and HAZOP [24], are generally the first systematic risk and reliability analysis techniques applied to a system. The purpose of an FMEA is to examine individual components and assess the effect of their failure on the system in which they are used (and on other systems and subsystems) [22]: FMEA is a qualitative method that is typically documented in a tabular format. To accomplish an FMEA, the analyst examines the components of a system one by one, and for each component, considers every known failure mode individually. The analyst writes a description of the failure mode, the method by which that failure would be detected, the effect of the failure, and the expected response of operators or automatic controls to the situation.

FMECA has the purpose of finding the components of a system whose frequent failures can have severe consequences, and are unlikely to be detected. In order to compute criticality according to the criteria provided by industrial management, first fault frequency (F), severity (S) and non-detection probability (D) have to be known for each one of the components in the studied system [25]. Classical FMECA is an empirical approach, where to each factor is given a value between 1 and 10, 10 being the worst possible case in Table 28.1.

Table 28.1 Classic FMECA [25]

F	Frequency	S	Severity	D	Non detection probability
10	Permanent	10	Human death	10	No detection possibilities
5	Frequent	5	Financially or materially consequences	5	A detection system exists but it is not infallible
1	Rare	1	Not serious	1	The detection system is infallible

Computing criticality for the applications used in control centers to supervise and coherently control the electric network needs adapted methodology for assessing the values to each one of the concerned factors. Figure 28.4 presents a deterministic approach to quantify the criticality of SCADA/EMS and Distribution Management System (DMS) functions.

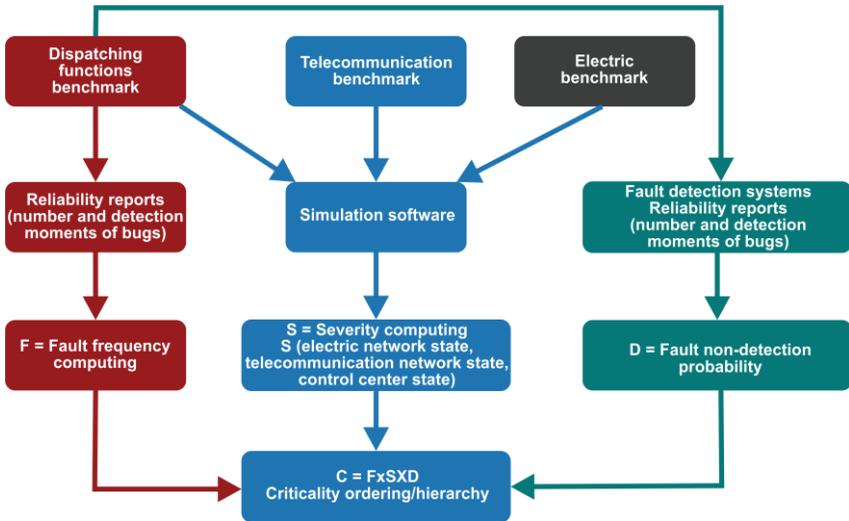


Fig. 28.4 Adapted FMECA for SCADA/EMS/DMS applications

The first tool necessary for the approach is a benchmark made out of electric, telecommunication and control center components that run as a whole unitary system. Another necessary tool is a combined simulator. For each component of the control center the fault frequency can be computed by using software reliability methods and input data from reliability reports [26]. Software reliability is a science descended from classical reliability that takes into consideration the conceptual differences between material and software. Reliability reports list the registered failures of software and the moment of each failure occurrence. Severity can be determined by using coupled simulators as those presented in the above section of this chapter. In order to better

understand cascading failures severity, not only the impact on the electric infrastructure is considered, but the whole coupled infrastructure made out of the power grid and the ICS components is studied. This approach can answer to questions such as “How serious is the fact that the electric network is in normal state but the operator can’t operate it?”

In practical reliability reports, no numeric correlation is done between the faults of dispatching functions/applications (for instance: State Estimation, Load Frequency Control, etc.) and their impact on the electric network [27].

The non-detection probability is the failure occurrence of the detection components. Their computation is the most problematic as the information concerning this aspect is rather poor when referring to operation applications. Study cases inventories of different SCADA/EMS and SCADA/DMS architectures have showed self-surveillance tools included by different control center providers. It is important to keep in mind that the most suitable way to reduce criticality is not by diminishing the severity of failures but by investing in powerful tools meant to detect them in due time.

Once all the factors are computed, they are multiplied and the result gives the criticality of each studied component. Adapted FMECA for electric network software management applications requires a complex procedure. This is why other approaches must be considered.

b) The HAZOP method

A HAZOP (Hazard and Operability) [20, 28, 29] study is related to an FMECA in that it assesses predefined scenarios to determine their probable causes, consequences, and possible remediation actions. It is a formalized methodology to identify and document hazards through imaginative thinking. It involves a very systematic examination of design documents that describe the installation or the facility under investigation. The HAZOP method focuses on qualitative deviations of key system operating parameters from their nominal, normal, or design values. The fundamental philosophy here is that normal operations are generally safe, and deviations from these normal operations are the source of unexpected or unrecognized problems. The objective is to find the “weak link” in the system, and to provide a basis for

developing procedural or engineering controls to reduce any risks so identified. The one-by-one nature of parameter variation in a HAZOP study and failure consideration in an FMEA can neglect the effects of multiple concurrent failures or variations, which may have both significant likelihood and high criticality.

The study is performed by a multidisciplinary team, analytically examining design intent deviations. Generally, a team of six members consisting of team leader, process engineer, operation representative, safety representative, control system engineer, and maintenance engineer is recommended for the study. The HAZOP team members try to imagine ways in which hazards and operating problems might arise in a process plant. To cover all the possible malfunctions in the plant, the HAZOP study team members use a set of ‘guide words’ for generating the process variable deviations to be considered in the HAZOP study. The sets of guide words that are often used are NONE, MORE OF, LESS OF, PART OF, and MORE THAN. When these guide words are applied to the process variables in each line or unit of the plant, we get the corresponding process variable deviation to be considered in the HAZOP study.

c) Checklists

Checklist analysis is a systematic evaluation against pre-established criteria in the form of one or more checklists, which are enumeration of questions about operation, organization, maintenance and other areas of installation safety concern and represent the simplest method used for hazard identification. Checklist analysis is based mostly on interviews, documentation reviews, and field inspections.

Development of a security requirements checklist can be helpful in analyzing controls in an efficient and systematic manner. The security requirements checklist can be used to validate security noncompliance as well as compliance. During development of security requirements checklist, the risk assessment team determines whether the system security requirements (SSR) stipulated for the IT system and collected during system characterization are being met by existing or planned security controls. Typically, SSR can be presented in table form.

SSR checklist contains the basic security standards that can be used to systematically evaluate and identify the vulnerabilities of the

assets (personnel, hardware, software, information), nonautomated procedures, processes, and information transfers associated with a given IT system. Table 28.2 lists security criteria suggested for use in identifying an IT system’s vulnerabilities in each security area according to [15].

Table 28.2 Security criteria [15]

Security area	Security criteria
Management security	Assignment of responsibilities; Continuity of support; Incident response capability; Periodic review of security controls; Personnel clearance and background investigations; Risk assessment; Security and technical training; Separation of duties; System authorization and reauthorization; System or application security plan
Operational security	Control of air-borne contaminants (smoke, dust, chemicals); Controls to ensure the quality of the electrical power supply; Data media access and disposal; External data distribution and labeling; Facility protection (e.g., computer room, data center, office); Humidity control; Temperature control; Workstations, laptops, and stand-alone personal computers
Technical security	Communications; Cryptography; Discretionary access control; Identification and authentication; Intrusion detection; Object reuse; System audit

A checklist analysis is used for high-level or detailed analysis, including root cause analysis; it is used most often to guide boarding teams through inspection of critical vessel systems; it is also used as a supplement to or integral part of another method, especially what-if-analysis, to address specific requirements.

Although checklist analysis is highly effective in identifying various system hazards, this technique has two key limitations:

(1) The structure of checklist analysis relies exclusively on the knowledge built into the checklists to identify potential problems. If the

checklist does not address a key issue, the analysis is likely to overlook potentially important weaknesses.

(2) Traditionally provides only qualitative information. Most checklist reviews produce only qualitative results, with no quantitative estimates of risk-related characteristics [20, 30].

d) What-if-analysis

Security risk analysis mainly consists of ‘what if’ analysis, during which the system is investigated for potential vulnerabilities and threats. It is an approach that uses broad, loosely structured questioning to postulate potential upsets that may result in accidents or system performance problems and determines what things can go wrong and judges the consequences of those situations occurring [29, 30].

The main characteristics of the technique are briefly summarized as follows: It is a systematic, but loosely structured, assessment, relying on a team of experts to generate a comprehensive review and to ensure that appropriate safeguards are in place. Typically is performed by one or more teams with diverse backgrounds and experience that participate in group review meetings of documentation and field inspections. It is applicable to any activity or system. It is used as a high-level or detailed risk-assessment technique. It generates qualitative descriptions of potential problems, in the form of questions and responses, as well as lists of recommendations for preventing problems. Occasionally it is used alone, but most often is used to supplement other, more structured techniques (especially checklist analysis).

The procedure for performing a what-if-analysis consists of the following steps:

(i) Specification and clearly definition the boundaries for which risk-related information is needed.

(ii) Specification the problems of interest that the analysis will address (safety problems, environmental issues, economic impacts, etc.).

(iii) Subdividing the subject into its major elements (e.g. locations on the waterway, tasks, or subsystems), so that the analysis will begin at this level.

(iv) Generation “what-if” questions for each element of the activity or system.

(v) Responding to each of the “what-if” questions and develop recommendations for improvements wherever the risk of potential problems seems uncomfortable or unnecessary.

(vi) Further subdividing the elements of the activity or system, if it is necessary or more detailed analysis is desired.

The section of some elements into successively finer levels of resolution until further subdivision will provide no more valuable information or exceed the system’s control or influence to make improvements. Generally, the goal is to minimize the level of resolution necessary for a risk assessment.

There are follow limitation of this approach: (1) it is based on experience of team members; (2) it is not systematic. The quality of the evaluation depends on the quality of the documentation, the training of the review team leader, and the experience of the review teams. From other side it is fast to implement compared to other qualitative techniques, it can analyze a combination of threats and failures and it is very flexible. It is generally applicable for almost every type of risk assessment application, especially those dominated by relatively simple failure scenarios. A good example and a case study a system which enables the user to perform “what-if” analysis on large distributed in complex data center applications can be found in [31].

e) Task Analysis (TA)

This process analyzes the way that people perform the tasks in their work environment and how these tasks are refined into subtasks and describes how the operators interact both with the system itself and with other personnel in that system. It can be used to create a detailed picture of human involvement using all the information necessary for an analysis in an adequate degree of details [32, 33].

Task analysis involves the study of activities and communications undertaken by operators and their teams in order to achieve a system goal. The result of a task analysis is a Task Model. The task analysis process usually involves three phases:

(i) collection of data about human interventions and system demands,

(ii) representation of those data in a comprehensible format or graph, and

(iii) comparison between system demands and operator capabilities.

The primary objective of task analysis is to ensure compatibility between system demands and operator capabilities, and if necessary, to alter those demands so that the task is adapted to the person. A widely used form of task analysis is the hierarchical task analysis (HTA).

Through its hierarchical approach it provides a well-structured overview of the work processes even in realistically sized examples. HTA is an easy to use method of gathering and organizing information about human activities and human interaction, and enables the analyst to find safety-critical tasks. It is time-consuming in case of complex tasks and requires the cooperation of experts from the application domain, knowledgeable about the task operation conditions.

Other analysis techniques are the Tabular Task Analysis, Timeline Analysis, Operator Action Event and Fault Trees [34], the GOMS-methods (Goals, Operators, Methods, and Selection Rules), Critical Action and Decision Evaluation Technique etc.

28.2.2 Quantitative techniques

f) Attack graphs technique for analyzing or quantifying security risks

A graph is one of the most natural representations of formalisms considering for network security analysis. Attack graph models a full-fledged attack as a sequence of atomic exploits and represents the combination of hosts, network configurations, vulnerabilities and exploits to describe the possible known security attacks. A consolidated view of major attack graph generation and analysis techniques can be found in [35]. Ordinarily, the attack graph is specified as a directed acyclic graph $G = (V, E, P, L)$, where V is a set of vertices that represent pre-conditions, vulnerabilities and exploits and E is a set of edges (arcs) that represent relationships between the pre-conditions, vulnerabilities and exploits. A probability P_i is associated with each vertex that represents the likelihood of an attacker exploiting vulnerability without considering the pre-conditions. An expected loss L_i is associated with each vertex that represents the loss value in

monetary units when the vertex has been exploited. The risk function of a vertex $v \in V$ is defined as the product of the cumulative probability of v with the expected loss of v . More specifically a quantitative calculation of the risk can be given with the following relation:

$$R(T_i) = P(v) \cdot L(v),$$

where $P(v)$ is the cumulative probability that represents the likelihood that a vulnerability associated with vertex is exploited. $L(v)$ is the expected loss in monetary value associated with a vertex v if vertex v is exploited. The set of security metrics for measuring overall security risk are grouped into different families which can be combined into a single score. These families of security metrics include [36]:

- (i) Victimization: scores network services and their vulnerabilities,
- (ii) Size: measures risk in terms of the attack graph size,
- (iii) Containment: measures security risk in terms of the degree with which the attack graph contains attacks across different network protection domains such as different subnets, and
- (iv) Topology: based on graph theoretic properties of the attack graph such as the weakly connected components, strongly connected components, length of maximum shortest path etc.

Attack graph based probabilistic security metric approach uses common vulnerability scoring system (CVSS) values for individual exploits and computes a cumulative score considering the causal relationship among exploits and security conditions.

For attack graph generation a model checking techniques can be used. The construction of an attack graph is based on the assumption that, vulnerability can always be exploited. But, in reality, there is a wide range of probabilities associated with exploitability of vulnerabilities.

g) The decision matrix risk-assessment (DMRA) technique

It is a systematic approach for estimating risks, which is consisting of measuring and categorizing risks on an informed judgment basis as to both probability and consequence and as to relative importance [37]. In [15] an approach for quantifying operational risks – special focus on cyber security risks is described. The combination of a

consequence/severity/impact and likelihood range, gives us an estimate of risk (or a risk ranking).

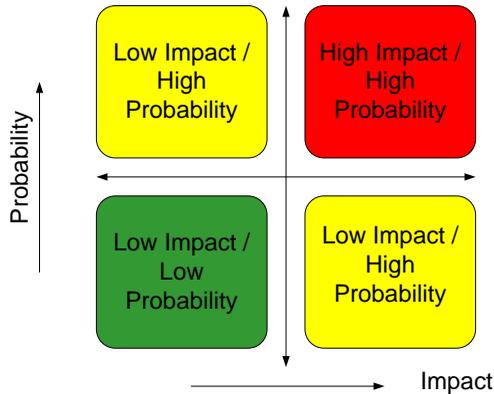


Fig. 28.5 2×2 Risk matrix

More specifically, the product of impact (I) and likelihood (P) provides a measure of risk (R) which is expressed by the relation:

$$R=I \cdot P.$$

The process follows from threat to actual business risk and impact. the first stage assessing the Probability of the threat occurring is performed Table 28.3.

Table 28.3. Sample probability table

Probability Category	Prob. No.	Description
Very High	9	Risk event expected to occur
High	7	Risk event more likely than not to occur
Probable	5	Risk event may or may not to occur
Low	3	Risk eventless likely that not to occur
Very Low	1	Risk event not expected to occur

Once the threats have been identified, the question of assigning impact and probability ratings must be addressed. So, on the next step assessing the Impact should the risk occur is executed.

Table 28.4. Business impact table

Impact	5	4	3	2	1
Financial	> \$ 500 m	\$200-500 m	\$50-\$200 m	< \$50 m	< \$50 m
Customer service & operations	Significant loss of customers due to extensive interruption to service capability
Reputation	Substantial damage to brands resulting from extensive negative national publicity
Legal / regulatory compliance	Loss of license, loss of public listing or substantial penalties on directors
People	Death or severe injury to employees
Customers	Serious financial impact to all customers

Each risk is analyzed for probability and impact and can be assigned in different point ratings:

- (i) a nine point rating (a score between 1 and 9);
- (ii) a five point rating (Very Low, Low, Medium, High, Very High or a score 1 to 5); or
- (iii) a three point rating (Low, Medium, High or score of 1 to 3).

And finally, working out the actual risk by combining the probability with the impact is calculated.

Table 28.5. Samples of probability and impact matrixes

		1	3	5	7	9
Probability	9	9	27	45	63	81
	7	7	21	35	49	63
	5	5	15	25	35	45
	3	3	9	15	21	27
	1	1	3	5	7	9
		Impact				

	1	3	5	7	9
9	9	27	45	63	81
7	7	21	35	49	63
5	5	15	25	35	45
3	3	9	15	21	27
1	1	3	5	7	9

Table 28.6. Scenarios on Risk Matrix

	Very Low	Low	Medium	High	Very High
Very High	Medium	Medium	High	High	High
High	Low	Medium	Medium	High	High
Medium	Low	Medium	Medium	Medium	High
Low	Low	Low	Medium	Medium	Medium
Very Low	Low	Low	Low	Low	Medium

This provides a very good, easy to understand overview of a relatively simple and workable risk assessment process. Besides this, the developed decision matrix risk assessment technique has two key advantages: (a) It differentiates relative risks to facilitate decision making. (b) It improves the consistency and basis of decision. Moreover, it is a quantitative (due to risk measuring) and also a graphical method which can create liability issues and help the risk managers to prioritize and manage key risks.

h) Quantitative assessment of domino scenarios (QADS)

As it mentioned in [38] cyber risk presents a unique concern in the energy sector because an attack on energy infrastructure has the potential to cross from the cyber realm to the physical world – a cyber-attack could cause, for instance, a massive operational failure of an energy asset. Large centralized infrastructures are especially at risk due to the potential ‘domino effect’ damage that an attack on a nuclear, coal, or oil plant could cause. Therefore we include QADS technique particularly related to the cyber industry in this survey. The domino effect is assumed as an accident in which a primary event propagates to nearby equipment, triggering one or more secondary events resulting in

overall consequences more severe than those of the primary event. Furthermore, an accident is usually considered as a “domino event” only if its overall severity is higher or at least comparable to that of the primary accidental scenario, while domino accidental scenarios result from the escalation of a primary accidental event. The escalation is usually caused by the damage of at least one equipment item, due to the physical effects of the primary event. Four elements may be considered to characterize a domino event:

- (i) A primary accidental scenario, which triggers the domino effect.
- (ii) A propagation effect following the primary event, due to the effect of escalation vectors caused by the primary event on secondary targets.
- (iii) One or more secondary accidental scenarios, involving the same or different plant units, causing the propagation of the primary event.
- (iv) An escalation of the consequences of the primary event, due to the effect of the secondary scenarios.

The quantitative assessment of domino accidents requires the identification, the frequency evaluation and the consequence assessment of all the credible domino scenarios, including all the different combinations of secondary events that may be originated by each primary event. The identification of the credible domino scenarios should be based on escalation criteria addressing the possible damage of equipment due to the physical effects generated in the primary scenarios. In the approach to the frequency assessment of domino scenarios, the damage probability of a unit due to a given primary event may be considered independent on the possible contemporary damage of other units.

Thus, if n possible target units are present, a single primary event may cause a maximum of n different secondary events, each having an overall probability to take place equal to $P_{d,i}$. However, each secondary event may take place contemporary to other secondary events.

A single domino scenario may thus be defined as an event involving the contemporary damage of k units resulting in k secondary events, with k comprised between 1 and n .

If each of the n secondary units is labeled by a numerical indicator comprised between 1 and n , a domino scenario may thus be indicated as a vector $J_m^k = [\gamma_1, \dots, \gamma_k]$ whose elements are the indexes of the secondary units involved in the event. Since $k \leq n$, in general more than one domino scenario may involve k units. Therefore, the subscript m of vector J indicates that the single domino scenario is the m^{th} combination of k secondary events.

The number of domino scenarios involving k different secondary events may be calculated by the following expression:

$$S_k = \frac{n!}{(n-k)!k!}.$$

The total number of different domino scenarios that may be generated by the primary event, S_d , can be calculated as follows:

$$S_d = \sum_{k=1}^n S_k = 2^n - 1.$$

The probability of a single domino scenario involving the contemporary damage of k units resulting in k secondary events, can be evaluated as follows:

$$P_d^{(k,m)} = \prod_{i=1}^n \left[1 - P_{d,i} + \delta(i, J_m^k)(2 \cdot P_{d,i} - 1) \right],$$

where the function $\delta(i, J_m^k)$ equals 1 if the i^{th} event belongs to the m^{th} combination, 0 otherwise. The last equation is the algebraic expression obtained from the union of the probabilities of the k events belonging to the m^{th} combination, calculated considering as independent the secondary events. The expected frequency of the m^{th} domino scenario involving k contemporary events, $f_d^{(k,m)}$ may thus be calculated as

$$f_d^{(k,m)} = f_p \cdot P_d^{(k,m)},$$

where f_p is the expected frequency of the primary event that triggers the escalation [39].

i) Functional Dependency Network Analysis (FDNA)

Functional Dependency Network Analysis (FDNA) was originally formulated by Garvey and Pinto [40, 41], who applied it to evaluate the effect of topology, and of possible degraded functioning of one or more systems on the operability of each system in the network. In [42] authors modified FDNA to make it suitable to analyze interdependencies in SoS [43], and applied it to aerospace SoS [44, 45]. The basic ideas and formulation of FDNA (complete description of the method, and its applications in other fields, cf. [43], [44], and [45]), with modifications to tailor it to the analysis of cyberattacks are represented below.

In FDNA, the architecture of SoS is modeled as a directed network (Fig. 28.6). The nodes represent either the component systems or the capability that the SoS is meant to acquire.

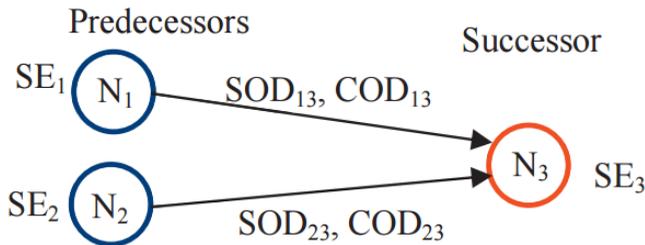


Fig. 28.6 Synthetic FDNA network. N: node. SOD: strength of dependency. COD: criticality of dependency. SE: self-effectiveness.

Accordingly, the links represent the operational dependencies between the systems or between the capabilities. Each dependency is characterized by strength (Strength of Dependency, SOD) and criticality (Criticality of Dependency, COD), that affect the behavior of

the whole SoS in different ways. Strength of dependency accounts for how much the behavior of a system is affected by the behavior of a predecessor system, while criticality of dependency quantifies how the functionality of a system is degraded when a predecessor system is experiencing a major failure. Those inputs can come from expert judgment and evaluation, or may be the result of simulation and experiments.

This method can be used to evaluate the effect of topology, and of possible degraded functioning of one or more systems on the operability of each system in the network. The analysis can be a deterministic evaluation of a single instance of the SoS, or a stochastic quantification of the overall SoS behavior. In the deterministic analysis, given the internal health status (called Self-Effectiveness, SE) of each system, and the properties of the dependencies, FDNA quantifies the operability O_i of each system, based on equations (28.1) – (28.6).

The operability of a node, ranging between 0 and 100, is defined as the “percentage” of effectiveness, that is the level at which the system is currently operating, or the level at which the desired capability is being currently achieved.

The operability of root nodes is equal to their self-effectiveness, since they are not dependent from other nodes:

$$O_i = SE_i \quad (28.1)$$

The operability of nodes that have at least one predecessor is computed as the minimum of two terms, one depending on the SODs, and one depending on the CODs:

$$O_j = \min(\text{SOD}_{-}O_j, \text{COD}_{-}O_j) \quad (28.2)$$

$$\text{SOD}_{-}O_i = \frac{1}{n} \sum_{i=1}^n \text{SOD}_{-}O_{ji} \quad (28.3)$$

$$\text{SOD}_{-}O_{ji} = \text{SOD}_{-}O_{ji}O_i + (1 - \text{SOD}_{ij})SE_j \quad (28.4)$$

$$\text{COD}_{-}O_j = \min(\text{COD}_{-}O_{j1}, \text{COD}_{-}O_{j2}, \dots, \text{COD}_{-}O_{jn}) \quad (28.5)$$

$$\text{COD}_{-O_{ji}} = O_i + \text{COD}_{ij} \quad (28.6)$$

In the stochastic version of FDNA, the self-effectiveness of each system follows a probability distribution; this means that the resulting operability of the nodes is probabilistic. In the previous studies authors [43] proposed FDNA as a tool to identify the most critical nodes in the network, as well as the most important dependencies, in terms of impact on the operability when disruptions occur. The robustness of a SoS can be evaluated as the ability to reduce the loss of operability when partial failures affect one or more systems.

j) Influence Diagrams

An influence diagram is an acyclic probabilistic network that consists of node sets N and arc sets A [46]:

$$G = (N,A).$$

The nodes can represent system states, decisions, or chance or deterministic occurrences, while the arcs represent the conditional dependencies among these occurrences. The nodes ultimately influence a “value node” that quantifies the consequences for each possible combination of occurrences and system states. Conditional probabilities can be applied within the model nodes to represent the probability that a particular event happens given specific conditions in the other nodes to which it is connected (i.e., the states, decisions, or events that influence this node). Thus, an influence diagram consists of four distinct parts: the nodes, the influences upon the nodes (the dependencies among the nodes, as represented by the arcs), the conditional dependencies within each node upon other nodes in the model, and the conditional probabilities themselves.

The influence diagram model (see Fig. 28.7) describes the information security risks in the interaction between the various elements of the relationship. Measure of information security risks is mainly based on the likelihood of security incidents and information on the size of the loss. The interaction between nodes is complex, it seems very difficult to obtain value at risk, but with the gradual refinement of the influence diagram, the relationship between the child nodes thread is getting clear.

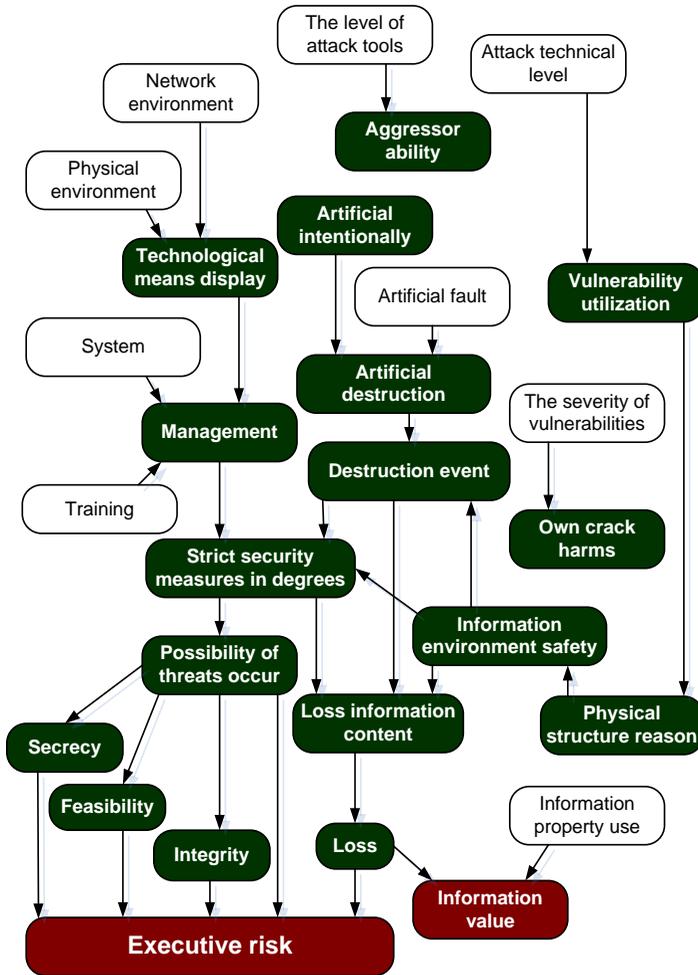


Fig. 28.7: Influence diagram model for risk assessment [47].

The influence diagram method is conceptually similar to the event tree, decision tree, and fault tree methods described further. Probabilistic influence diagram is suitable for the interrelationship and information flow of various possible forms and is capable of multiple sequential evaluation process in order to overcome the traditional

method of analysis of the limitations of the order of unity and improve the efficiency of risk assessment. In addition, the method is not limited to simple binary events as is FTA. This flexibility makes the influence diagram an important tool to the risk analyst. Recent methods for solving influence diagrams [48], which emphasize the development of “paths” (similar to ETA), have enabled influence diagrams to produce highly valuable risk assessment results.

k) Markov Models

The Markov model [49] is a directed graph that captures the concepts of system states and probabilistic transitions between states. To build a Markov model, an analyst examines every relevant configuration of a system – both functional and nonfunctional configurations – and defines them to be states of the system. Then they defines the probability of transfer from each state to every other state (as a function of time and other factors) to complete the model.

An interesting application of Markov modeling is found in the continuous event tree methodology [50] (see section 28.2.3). In this method, the branching operations within an event tree model are viewed as state transitions within the framework of a Markov model. This allows the analyst to determine the population of each state (and, hence, of each branch within the event tree model) as a function of time. The method has been extended to a semi-Markov process to allow for the state and branch transition probabilities to vary as a function of the length of time the system has spent in that state [51].

Another approach uses Markov Latent Effects (MLE) to quantify imprecise subjective metrics through possibilistic or fuzzy mathematics, which are then aggregated using weighted sums to rank the credibility of various threat scenarios [52]. The latent effects represent the influence that one decision element has on another. This approach explicitly evaluates the threat potential, recognizing that full probabilistic assessment is not possible due to a lack of experiences to provide probabilistic data sets.

l) Bayesian Networks for Security and Risk Assessment

This last approach concerns the system security assessment with regard to the potential cyber attacks against the ICS of the power

system. The evaluation of the impact of cyber attacks on the ICS is complicated, because of the uncertainty modeling of this type of intentional acts and their uncertain impact on the system's performance. The intentional nature of the cyber attacks and their occurrence does not exhibit a random characteristic. Thus, answering the following type of questions becomes difficult [22]:

- What is the probability of the implementation of these attacks?
- What is the probability to have a catastrophic impact?

The response requires thinking about the attackers' motivation, possible vulnerabilities exploited, the lack of protection, generation's ICT (age), the criticality of functions performed by attacked technology, among other numerous factors. All these variables contain different uncertainty levels [53]. Moreover, in this problem, different infrastructures can be affected and finally to lead damages in the system operation. It appears that Bayesian networks offer the possibility of including in a single model, different types of uncertainties present into the problem and, of modeling interdependencies between the different infrastructures. Inferences are also made in order to have more extensive information, which is not directly observed based on data, experience, a priori reasoning. In the probabilistic inference, the probability models the uncertainty and the joint probability distribution function describes the dependency relationship between variables/facts.

Bayesian networks (BNs) are one of the most common graphic models representing probabilistic inference. They are employed to understand and to obtain conclusions about interdependency relationships into models. Bayesian network is a directed acyclic graph whose structure describes a set of conditional independence properties about the variables [54]. The network offers for a set of variables $\{X_1, X_2, \dots, X_n\}$ a compact representation of the joint probability distribution. The structure and the numerical parameters of a BN can be elicited from the experts or based on data. Four situations are distinguished in this construction:

(1) Experts build the BN based on determinist knowledge. In this case the modeling does not represent any problem because both laws and variables are known. A typical example in power systems operation is: if the transmission line flow exceeds its capacity, then line is overloaded.

(2) Experts' knowledge is modeled since variables and relationships between them are uncertain. Intuitive reasoning is used for this. In this case, information of a diverse nature is taken into consideration in the modeling. This process is more laborious even if they know this matter or have a lot of experience. It exits a lot of examples in the power system security field as: the modeling of attackers' motivation against ICT, interdependencies between power grid and its associated ICS, human acts and decision in the control loop among other examples.

(3) Experts assign subjective probability values to network parameters: when data are not available or the uncertainty does not obey to randomness, the only possibility is to use a personal measurement of uncertainty or belief in an event.

(4) Bayesian network is built based on data. Variables and links between them can be established from statistical (analysis of correlation, covariance, etc.) and optimization techniques.

Once the BN is created, via the probabilistic inference, posterior probability distributions of variables of interest are quantified. The problem is then to find the probability of a set of query variables given a set of evidence. This is possible by sequential applications of Bayes' theorem. Different possible of inferences are illustrate in Fig. 28.8.

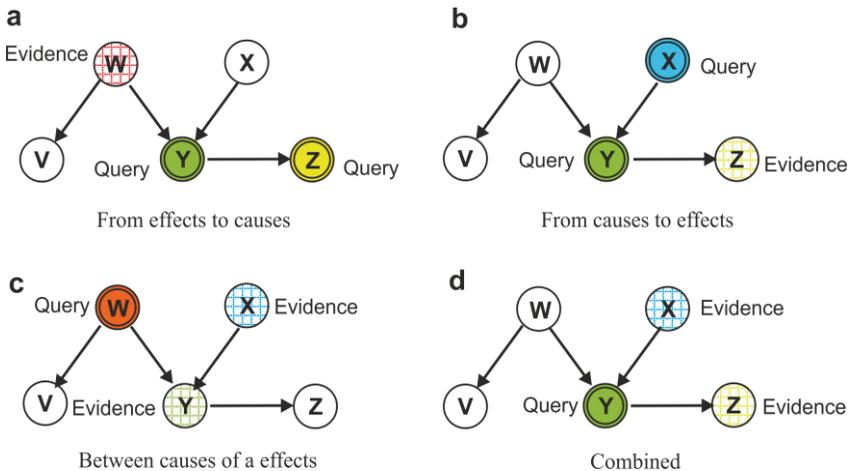


Fig. 28.8 Types of inferences in Bayesian networks

According to [55], “Bayesian networks are direct representations of the world, not of reasoning process”. This is correct because links in the Bayesian networks represent real causal connections and not the flow of information during reasoning. The different inferences above presented can be obtained from these networks by propagating information in any direction.

The possibility of incorporating subjective probability values by using Bayesian networks allows us to model different types of uncertainty [56] and to model the interdependencies between main functions of the ICS and the power grid operation. Additionally, these graphs can also improve the communication between experts of different domains as in the case of ICT’s and power systems’ experts.

An example of interdependency modeling technique for SoS using Bayesian Networks can be found in [57]. Event trees and Bayesian Networks are used to quantify development interdependencies between systems and assess cascading development risks.

28.2.3 Hybrid techniques

By applying different hybrid techniques e.g. the bow tie model (Fig. 28.9), we can determine the threats, vulnerabilities and required controls for cyber-security. It allows to model a hierarchy of causal relationships that could lead to an incident and controls that could reduce the likelihood of the incident occurring (fault tree); and the potential consequences and contingent controls (event tree).

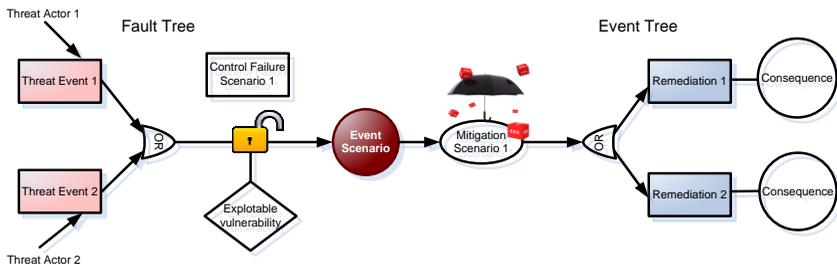


Figure 28.9. Bow Tie model showing Fault Tree and Event Tree

The combination of fault tree and event tree analysis can quickly get complicated. To cope with the complexity the process of fault tree analysis, event tree analysis, and control selection can be modelled using Bayesian Networks [58].

m) Fault-tree analysis (FTA)

FTA is a deductive technique focusing on one particular accident event and providing a method for determining causes of that event. In other words FTA is an analysis technique that visually models how logical relationships between equipment failures, human errors, and external events can combine to cause specific accidents.

Fault trees are constructed from events and gates. Basic events can be used to represent technical failures that lead to accidents while intermediate events can represent operator errors that may intensify technical failures. The gates of the fault trees can be used to represent several ways in which machine and human failures combine to give rise to the accident. For instance, an AND gate implies that both initial events need to occur in order to give rise to the intermediate event. Conversely, an OR gate means that either of two initial events can give rise to the intermediate event [30, 59]. Below it is presented a summary of the graphics most commonly used to construct a fault tree. This technique starts with the undesired event (top event) and determines all the ways in which it could occur. These are displayed graphically in a logical tree diagram.

Once the fault tree has been developed, consideration should be given to ways of reducing or eliminating potential causes / sources. Fault Trees are composed from events with relationships that connect the events together reflecting the structure and relationships within the organization and to the organization environment. As example authors [60] took the e-mail phishing scenario to build a simplified fault tree model (fig. 28.10).

Study the fault-tree model and the list of minimal cut sets allow to identify potentially important dependencies among events. Dependencies are single occurrences that may cause multiple events or conditions to occur at the same time.

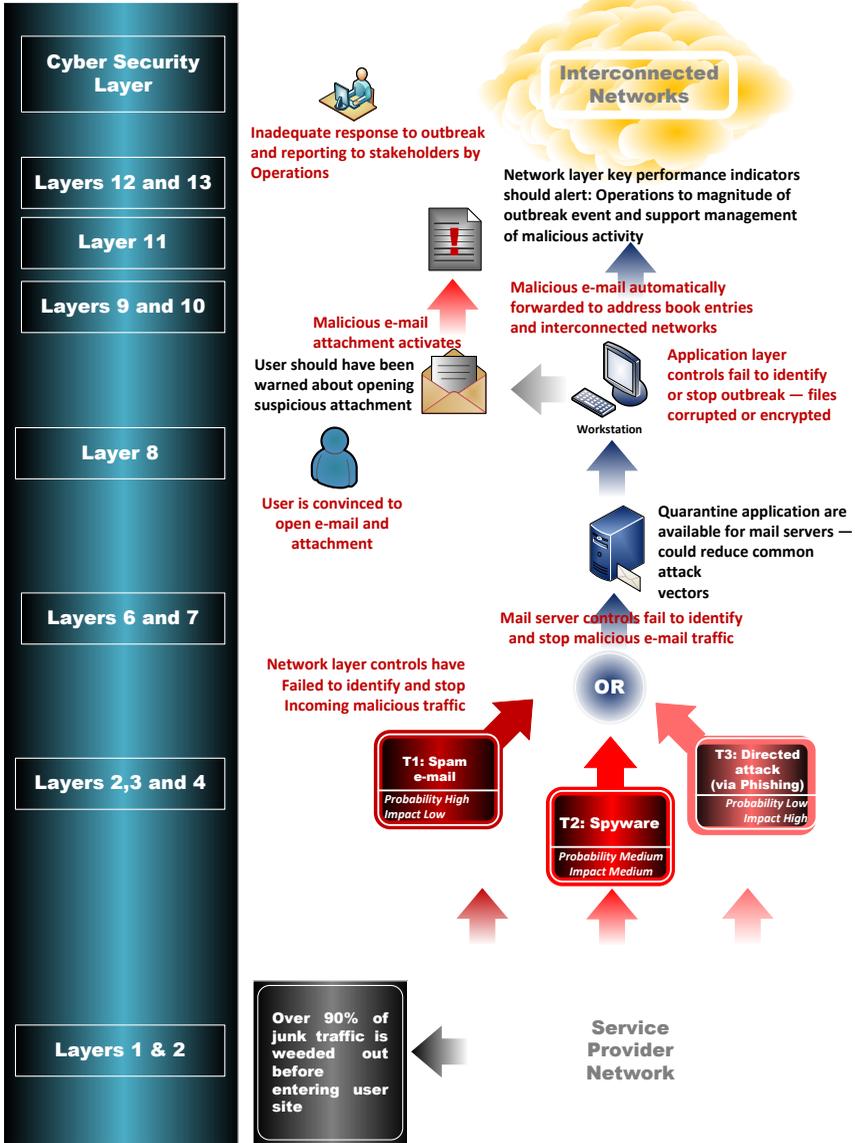


Figure 28.10. Fault tree analysis across the security metrics (adapted from [60])

The further extensions of fault trees methodology are **attack trees** and **attack-fault tree (AFT)**. Attack trees are widely used for security risk assessment. An attack tree is a particular graph that describes the steps of an attack process. It uses the same basic symbols as fault trees: nodes (represent attacks), gates (AND, OR), and edges (path of attacks through the system). Every attack has a final scope or a final motivation. For example, the final scope of a Denial of Service against a WebServer could be to avoid to a set of users the access to the data showed into such web-server. This final scope can be defined briefly as the Goal of the attack [61]. Several authors propose to use additional symbols in attack trees. E.g. dynamic, “trigger” edges [62] can be used in situations when one attack event (e.g. Attack 1) triggers the other (e.g. Attack 2). In this case, Attack 2 can be realizable only if Attack 1 has been completed. An example of an attack tree is shown in Fig. 28.11 it depicts attack process steps of the Stuxnet attack [63].

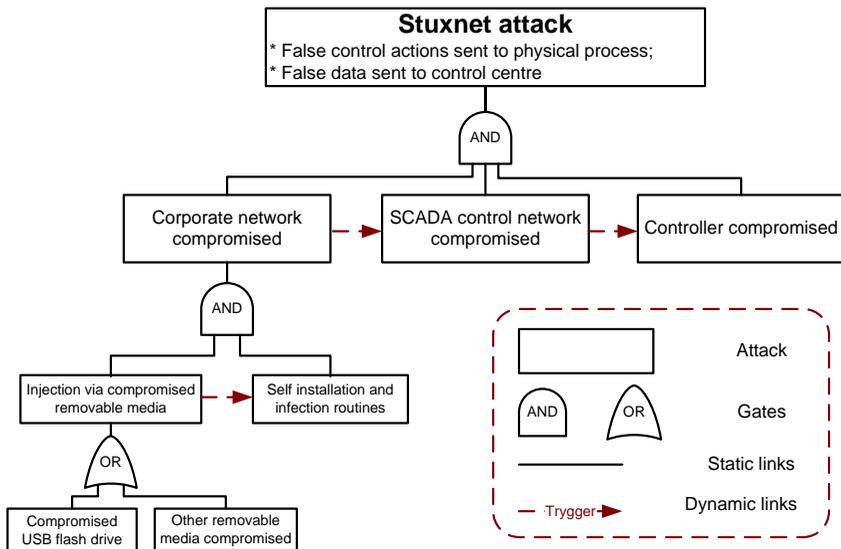


Figure 28.11: An example of attack tree (Stuxnet) [64]

The goal of the Stuxnet attack is to compromise controller, which is controlling a SCADA system. The attack starts with injection via compromised removable media, which could be done by user opening a

compromised file folder either on USB flash drive, or other removable media. Once this step is completed, Stuxnet worm instantaneously starts its self-installation and infection routines, thus there is a “trigger” line between these nodes. After injection and self-installation and infection routines are completed, an attacker is able to compromise corporate network, which allows him to gain access to SCADA control network, and eventually to compromise a controller.

AFT models how a top-level (safety or security) goal can be refined into smaller sub-goals, until no further refinement is possible. In that case, the leaves of the tree model either the basic component failures, the basic attack steps or on demand instant failures. Since subtrees can be shared, AFT represents the directed acyclic graphs, rather than tree [65]. Note that the attack tree is only one example of the ICT security analysis methodologies available today. If necessary, more advanced security assessment tools are available, including, e.g., attack graph [66, 67], ADVISE [68], CyberSAGE [69], attack defense tree [70], etc. These tools support features such as automatic generation of likely attack scenarios based on vulnerability and system information, and more detailed modeling of attacker behavior and attacker/defender interactions.

n) The ETA method (Event Tree Analysis)

Event tree analysis (ETA) is a technique that uses decision trees and logically develops visual models of the possible outcomes of an initiating event. Furthermore, it is a graphical representation of the logic model that identifies and quantifies the possible outcomes following the initiating event. The models explore how safeguards and external influences, called lines of assurance, affect the path of accident chains [30, 71, 72]. In this method, an initiating event such as the malfunctioning of a system, process, or construction is considered as the starting point and the predictable accidental results, which are sequentially propagated from the initiating event, are presented in order graphically. ETA is a system model representing system safety based on the safeties of subevents. It is called an event tree because the graphical presentation of sequenced events grows like a tree as the number of events increase. An event tree consists of an initiating event, probable subsequent events and final results caused by the sequence of

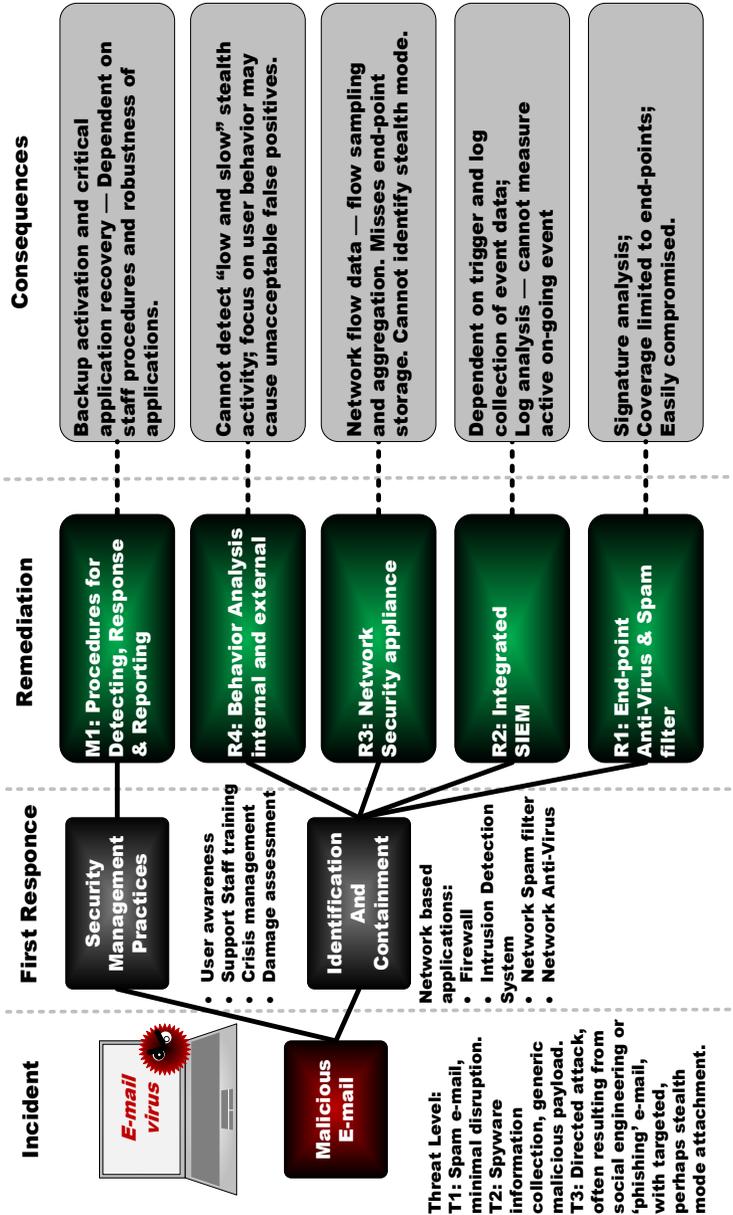


Fig. 28.12 Event tree for mail server receives malicious e-mail

events. From the preceding FTA example, we can choose threat events to create the event tree model. As an illustration, an initiating event the acceptance of the malicious software at the e-mail server is shown in fig. 28.12.

Here the event is assumed has already happened: somehow malicious e-mail data, of unknown threat level, has passed through the network layer control.

Probable subsequent events are independent to each other and the specific final result depends only on the initiating event and the subsequent events following. Therefore, the occurrence probability of a specific path can be obtained by multiplying the probabilities of all subsequent events existing in a path. In an event tree, all events in a system are described graphically and it is very effective to describe the order of events with respect to time because the tree is related to the sequence of occurrences.

In the design stage, ETA is used to verify the criterion for improving system performance; to obtain fundamental information of test operations and management; and to identify useful methods to protect a system from failure.

The ETA technique is applicable not only to design, construction, and operation stages, but also to the change of operation and the analysis of accident causes.

o) Fuzzy set techniques to evaluate the cyber-risk

This technique is highly subjective and its utilization takes into account the imprecise and vagueness of cyber risk metrics [73]. A number of studies have employed fuzzy set theory in risk analysis, such as [73, 74, 75]. Cyber-risk is seen as the possibility for loss of confidentiality, integrity, and availability due to a specific threat and mathematically described by the fuzzy relational function:

$$\text{CyberRisk}_{\text{threat}} = F(\text{Threat}_{\text{asset}}, \text{Vulnerability}_{\text{threat}}, \text{Asset}_{\text{value}}),$$

where the fuzzy arguments are $\text{Threat}_{\text{asset}}$, $\text{Vulnerability}_{\text{threat}}$ and $\text{Asset}_{\text{value}}$.

To assess security risks of a computer and network system, the value of each asset is evaluated for the importance of the asset, and vulnerabilities and threats which may cause damage or loss of asset values are also examined.

The steps involved in the Fuzzy Risk Calculations technique [75] are as follows:

- I. Carry out risk identification;
- II. Perform fuzzification of risk constructs;
- III. Use fuzzy risk assessment and aggregation; iv. Perform fuzzy weighted mean computation; and
- IV. Perform defuzzification.

This approach culminates into a cyber-security vulnerability assessment (CSVA) model assists decision-makers on available risk options. The essence of the CSVA model is to offer decision makers a plausible and pragmatic approach to assessing risks against ICT assets. Authors [76] posited that determining risk amounts to addressing the following questions:

- (1) What could go wrong?
- (2) How many times does it go wrong?
- (3) What is the impact on the organization or what are the consequences?

The answer to the first question could be interpreted as the assets being compromised. The second question requires the evaluation of the possibility of occurrences of these threats. The third question estimates the extent and severity of consequences or the impact level of the risk as result of the exploited vulnerabilities.

28.2.4 Integrated safety and security risk assessment methods

Risk assessment methods like FMEA, FTA, Component Fault Tree (CFT) have been used by safety community whereas the risk assessment methods like Attack Trees, AFT, Attack-Countermeasure Trees (ACT), National Institute of Standards and Technology (NIST) 800-30 Risk Assessment [15] have been used by security community. Several authors used these methods as a starting point for the development of integrated safety and security risk assessment methods [77].

Authors [77] distinguished four ways in which the integrated safety and security risk assessment methods have been developed (see Table 28.7): (1) Combination of a usual safety risk assessment (S_pRA) method and a variation of the usual S_pRA method for security risk assessment. The methods SAHARA and FMVEA come under this category; (2) Combination of a usual security risk assessment method (SRA) and a variation of the usual SRA method for safety risk assessment. The Unified Security and Safety Risk Assessment method comes under this category; (3) Combination of a S_pRA method and a SRA method. The methods FACT Graph, Extended CFT, and EFT come under this category; (4) Others - There is no conventional S_pRA , and conventional SRA method used in the integration. The CHASSIS method comes under this category.

Table 28.7: Some integrated safety and security risk assessment methods [77]

Safety risk assessment method	Security risk assessment method	Integrated safety and security risk assessment method
ISO 26262: HARA	Variation of ISO 26262: HARA	SAHARA
FMEA	Variation of FMEA	FMVEA
Variation of NIST 800-30 security risk estimation	NIST 800-30 security risk estimation	Unified security and safety risk assessment
Fault Tree	Fault Tree	FACT Graph
CFT	Attack Tree	Extended CFT
Fault Tree	Attack Tree	EFT
Safety Misuse Case (involving faulty-systems)	Security Misuse Case (involving attackers)	CHASSIS

p) Security-Aware Hazard Analysis and Risk Assessment (SAHARA)

The steps involved in the SAHARA method [78] are as follows:

I. The ISO 26262 – Hazard Analysis and Risk Assessment (HARA) approach is used to classify the safety hazards according to the

Automotive Safety Integrity Level (ASIL), and to identify the safety goal and safe state for each identified potential hazard. The HARA process focuses your attention on areas most in need by evaluating which populations and facilities are most vulnerable to hazards and to what extent injuries and damages may occur. It answers the fundamental question: "What would happen if a hazard event occurred?";

II. The attack vectors of the system are modeled. To model the attack vectors of the system the STRIDE method can be used [79]. The first steps of the SAHARA approach, combining security and safety analyses, is to quantify the STRIDE security threads in an analog manner as is performed for safety hazards as part of the HARA approach. STRIDE is a classification scheme for characterizing known threats according to the kinds of exploit that are used (or motivation of the attacker) [80];

III. The security threats are quantified with reference to the ASIL analysis, according to the resources (R) and know-how (K) that are required to pose the threat and the threats criticality (T);

IV. The security threats are classified according to the Security Level (SecL). SecL is determined based on the level of R, K, and T;

V. The security threats that may violate the safety goals ($T > 2$) are considered for the further safety analysis.

q) Combined Harm Assessment of Safety and Security for Information Systems (CHASSIS)

Combined Harm Assessment of Safety and Security for Information Systems (CHASSIS) [81] is an approach for requirements engineering via use cases and sequence diagrams, it uses a variation of Unified Modeling Language (UML)-based models for both the safety and security risk assessment.

CHASSIS has two-steps

I. The definition of functional requirements as a basis for the elicitation of safety and security requirements. Users, functions and services are described in use case diagrams and textual descriptions of use cases.

II. The elicitation of safety and security requirements is carried out. Through a brainstorming session with domain as well as safety and

security experts, potential misuses of the system are identified. The names of use cases are combined with hazard and operability study (HAZOP) [28] guide words in order to obtain potential misuses of the system.

r) Failure-Attack-Countermeasure (FACT) Graph

Failure-Attack-Countermeasure (FACT) Graph incorporates various artefacts: safety artefacts (fault trees and safety countermeasures) and security artefacts (attack trees and security countermeasures). In FACT graph, safety and security countermeasure are attached to the relevant faults and attacks, thus it is easy to identify interrelated countermeasures and analyze their interdependencies.

The steps involved in the FACT Graph method [64] are as follows:

I. The fault trees of the system analyzed are imported to start the construction of FACT graph;

II. The safety countermeasures are attached to the failure nodes in the FACT graph;

III. The attack trees of the system analyzed are imported to the FACT graph in construction. This is done by adding an attack-tree to the failure node in the FACT graph with the help of OR gate, if the particular failure may also be caused by an attack;

IV. The security countermeasures are attached to the attack nodes in the FACT graph. This could be done based on the ACT technique [82, 83] where, once attack tree is constructed, and possible security countermeasures are attached to the attack nodes, security analysts can select a set of security countermeasures for implementation, considering a given budget [82].

s) Failure Mode, Vulnerabilities, and Effect Analysis (FMVEA)

With the increasing awareness of the security implications for safety-critical systems, safety assessment methodologies and standards are being extended to explicitly take security into account. A recent example is the Failure Modes, Vulnerabilities and Effects Analysis (FMVEA) approach [84], which extends the well-established Failure Mode and Effects Analysis (FMEA) methodology [23] (see section 28.2.1) and extends the standard approach with security related threat

modes. Figure 28.13 shows the process of applying FMVEA to a system.

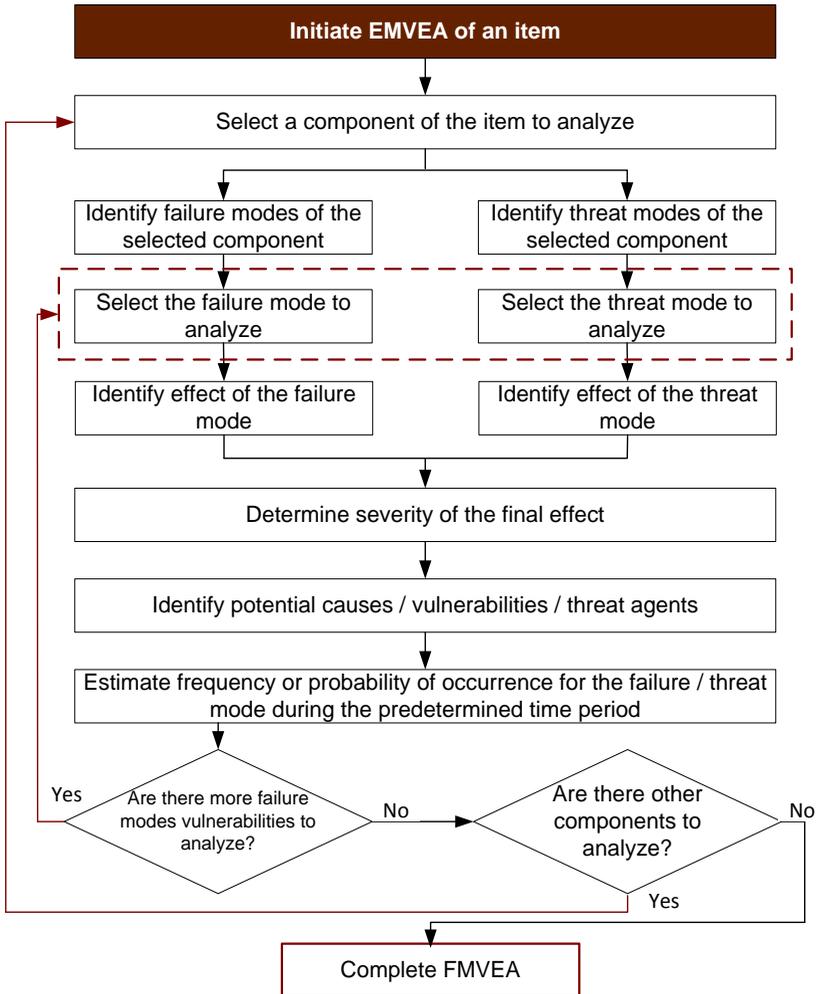


Fig. 28.13 Overview of FMVEA method [85]

The system model is based on a three-level data flow diagram (DFD). Effects of failure and threat modes are presented at the context

level of the diagram, which shows the interaction between the system and its environment. Failure and threat modes are located at the level 1 DFD.

Vulnerabilities and failure causes are based on the level 2 DFDs [85]. FMVEA uses the information about the system architecture for rating risks thus, this technique is more appropriate for later phases in the engineering process such as design and verification, where more information about the system is available.

The steps involved in the FMVEA method [85] are as follows:

I. Making a list of system components functional analysis at the system level;

II. A component that needs to be analyzed from the list of system components is selected;

III. The failure/threat modes for the selected component are identified;

IV. The failure/threat effect for each identified failure/threat mode is identified;

V. The severity for the identified failure/threat effect is determined;

VI. The potential failure causes/vulnerabilities/threat agents are identified;

VII. The failure/attack probability is determined. The attack probability is described as the sum of threat properties and system susceptibility ratings. The threat properties is the sum of motivation and capabilities ratings, whereas the system susceptibility is the sum of reachability and unusualness of the system ratings;

VIII. Finally, the risk number is determined, which is the product of severity rating and failure/attack probability.

t) Unified Security and Safety Risk Assessment

Unified Security and Safety Risk Assessment method mainly adopts an approach similar to the security risk estimation method NIST 800-30 [15]. However, unlike NIST 800-30, this method considers both security and safety risks.

The steps involved in the Unified Security and Safety Risk Assessment method [86] are as follows:

I. System characterization. The software and hardware of the target system are identified to obtain related functionalities, system boundary, and data properties;

II. Identification of threats, hazards, vulnerabilities, and initiating events; the relationship between security and safety has been denoted as a Boolean parameter (v, t, h) , where v is a vulnerability, t is a threat, and h a hazard. If vulnerability v is exploited by threat t triggering an initiating event that causes hazard h , the value of the parameter is 1. Otherwise, it is 0.

III. Control analysis. The current and planned controls are identified;

IV. Determining the threat likelihood;

V. Determining the hazard likelihood;

VI. Asset impact value analysis;

VII. Determining the combined safety-security risk level;

VIII. Providing control recommendations.

IX. Result documentation. The risk assessment reports are provided.

u) Extended Fault Tree (EFT)

As it mentioned above, a **fault tree** (see section 28.2.3) describes how a set of events can concur in order to cause a certain Top Event. In the field of System Security fault trees are usually used to describe the “race conditions” of a system, i.e. the combinations of events which can, in some way, damage its functionalities.

In the same way, **attack trees** are used to describe the steps an attacker must follow in order to successfully exploit some functionalities of a system.

EFT is a technique for the integration of fault-tree and attack tree analysis structures in order to extend the usability of the results of a traditional risk analysis with the consideration of potential malicious attacks. This integration may help in understanding how malicious attackers can take advantage of the failures of components for deploying their hostile actions. An Integrated fault tree and attack tree is shown in fig. 28.14.

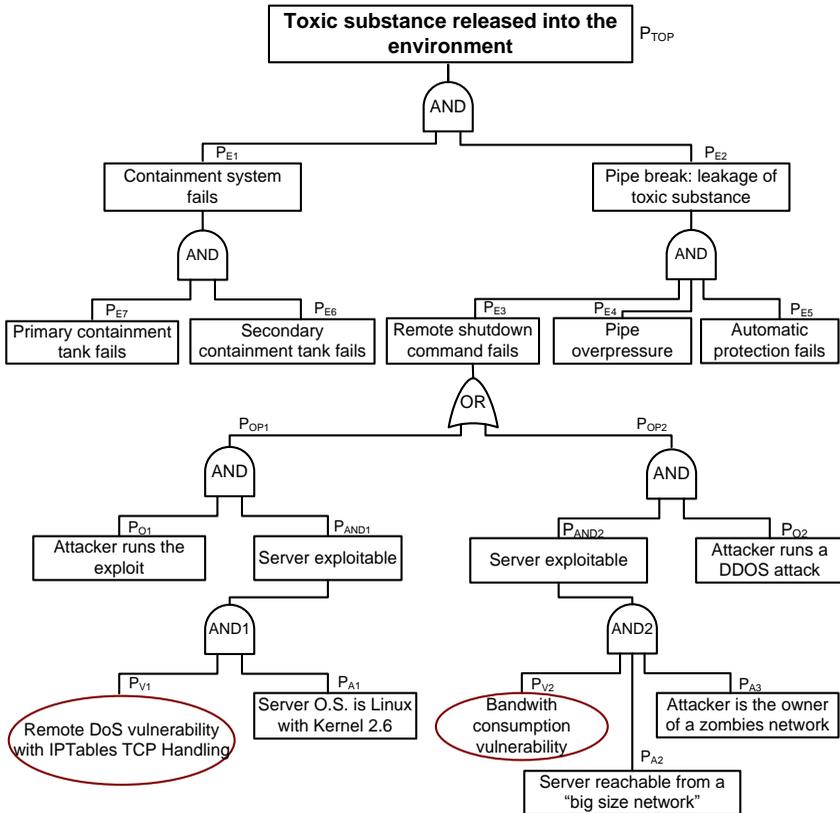


Fig. 28.14 Integrated fault tree and attack tree [61]

The steps involved in the EFT method [61] are as follows:

- I. Development of the fault tree for the system to be analyzed. During this process the random faults should be taking into account;
- II. Extending the fault tree by adding an attack tree to the basic or intermediate event in the fault tree, if the particular event in the fault tree may also be caused by malicious actions. The attack tree concept used in the development of EFT is based on [87];
- III. Quantitative analysis. It is performed based on the formulae defined in [61] to calculate the top event probability.

This approach promises to be helpful not only in the analysis of the risk exposure of ICT systems, but even in the discovery of new complex attack pattern profiles.

v) Extended Component Fault Tree (CFT)

This hybrid approach was proposed in [88] for rating of the events to avoid the problem of assigning probabilities to security-related events.

The steps involved in the extended CFT method [88] are as follows:

I. Development of CFT for the system to be analyzed This could be done based on [21];

II. The CFT is extended by adding an attack tree to the failure node with the help of OR gate, if the particular event may also be caused by an attack;

III. Qualitative analysis. The results of this analysis are ordered lists of Minimal Cut Sets (MCSs). A cut set is a set of basic events which together cause the top level event of the tree. MCSs containing only one event would be single point of failure which should be avoided;

IV. Quantitative analysis. It is conducted by assigning values to the basic events. Therefore, MCSs containing only safety events would have a probability P, MCSs containing only security events would have a rating R, MCSs containing both safety and security events would have a tuple of probability and rating (P, R).

w) The SERA framework

The SERA framework stands from Security Engineering Risk Analysis [89] and defines an approach for analyzing security risk in SoS and software-reliant systems across the software lifecycle. The SERA framework includes the four tasks:

I. Establish the operational context;

II. Identify risk;

III. Analyze risk;

IV. Develop a control plan.

In contrast with traditional security-risk analysis techniques which exploit a simplified view of security risk, where a single threat actor

exploits a single vulnerability in a single system to cause an adverse consequence, the SERA is shared the understanding of the system in its operational environment using multiple models that represent various aspects of the system security. Another difference SERA framework is it based on brainstorming techniques. The analysis team should be an interdisciplinary with members providing diverse skill sets.

For each security risk, the SERA framework requires the following data to be recorded: Security risk scenario; Risk statement; Threat components; Threat sequence; Workflow consequences; Stakeholder consequences; Enablers. When brainstorming is used, participants describe risks based on their tacit understanding of the operational environment.

More information about this technique and a case study illustrating the use of SERA framework for the Wireless Emergency Alert (WEA) system can be found in [90]

28.3 SoS resilience analysis

Resilience assessment methods for cyber systems

Conclusions

The choice of risk assessment methodologies is vast, though in general, the application of the existing approaches is not straightforward. In addition it is not clear whether the needs of emerging goals can be satisfied.

In the literature, one can identify three different approaches: application of RA methodologies to infrastructure, structural analysis, and behavioral analysis. Structural approaches assume the existence of a system of systems topology, which accounts for interdependencies.

Risk analysis of complex SoS requires a systemic and holistic approach that integrates multiple perspectives, models and tools. For all that the cyber risk of SoS must be considered as a part of the overall system risk, while the people assessing the risk should be from different

teams focused on their fields of expertise and system-wide process as well.

There are different ways in which the integrated safety and security risk assessment methods have been developed. A holistic approach must aim at integrating both safety and security concerns with clear expressions of the interplays.

In many cases, there is room for significant innovations.

Some tips for choosing a risk framework: When looking at the ‘frameworks’ that are commonly used in information security circles, you will find that they all came about to address slightly different problems, and therefore may be more suited to one environment or another. Beyond the frameworks themselves, it is essential to consider what risk processes and practices already exist. If something does already exist, be sure to first determine if it can be expanded to meet your needs or how the information security risk frameworks would interoperate with models in other business units. The next step before you select a framework is to look at the core objectives of your system and to make a list of strategic objectives. Finally, before you select any framework or even a risk model, you need to guide the executive team through the exercise of articulating the system’s risk threshold. Have them describe in words, what level of risk they want to escalate to them. For example, if the system’s mission includes a focus on being highly availability, then this might be their risk tolerance statement: “At a minimum, any risk that is likely to result in service outage for all clients longer than the published recovery time objective should be escalated to the executive management team.”

Then you would want to select a framework that could articulate risks in a way that would allow you to compare each risk to this tolerance statement easily. The framework would need to include the same risk factors that your system is focused on.

Questions to self-checking

1. Why we have to combine safety and security risk assessment methodologies in the SoS landscape?
2. Types interdependencies between safety and security in SoS.

3. Specify main categories of risk analysis and assessment techniques.
4. Name the most popular cyber security risk assessment methodologies.
5. What is the difference between hybrid technique and integrated safety and security techniques?
6. For which purpose Bayesian networks are employed?
7. How safety and security risk assessment methodologies can be combined for SoS risk assessment?
8. In what ways the integrated safety and security risk assessment methods can be developed?
9. The steps involved in the FMVEA method.
10. What technique can be applied for the integration of fault-tree and attack tree analysis?
11. How to use the results of integrated safety and security risk assessment in risk treatment?
12. How to address safety and security interactions in Risk Treatment?

References

1. Trivellato D., Zannone N., Etalle S. A security framework for systems of systems. In Policies for Distributed Systems and Networks (POLICY), 2011 IEEE International Symposium. – 2011. – pp. 182-183.
2. Pinto C.A.; McShane M.K., Bozkurt I. System of Systems Perspective on Risk: Towards a Unified Concept // Finance Faculty Publications. Paper 1. – 2012. – pp. 33-46.
3. Keating C., Rogers R., Unal R., Dryer D., Sousa-Poza A., Safford R., Peterson W., Rabadi G. System of systems engineering // Engineering Management Journal. – 2003. – Vol. 15, No. 3. – pp.36–45.
4. Modeling and Simulation Support for System of Systems Engineering Applications / L.B. Rainey, A, Tolk (Eds.), John Willey & Sons, Inc., 2015. – 641 p.

5. Tagarev T., Ivanova P. Modelling Extreme Events for the Purposes of Security Foresight // Radioelectronic and Computer Systems. – vol. 59. – pp. 253-259.

6. **Sitbon**, P. Security in remote services used by EPU's / P. Sitbon, C. Poirier, J. Zerbst, DK Holstein, M. Scherer, R. Evans, Marc Tritschler Electricité de France, Electricité de France, Vattenfall, OPUS Consulting, Alstom, Snowy Hydro Ltd, PA Consulting France, France, Sweden, USA, France, Australia, UK.

7. Houmb S.H., Jurgens G.G., France J.R. An Integrated Security Verification and Security Solution Design Trade-Off Analysis Approach // Integrating Security and Software Engineering – Advances and Future Visions by Mouratidis H., Giorgini P (eds.), Chapter IX. – 2007. – pp. 190-219.

8. Piètre-Cambacédès L., Bouissou M. Modeling safety and security interdependencies with BDMP (Boolean logic Driven Markov Processes) // Proceedings of the IEEE International Conference on Systems Man and Cybernetics (SMC 2010). – 2010. – pp. 2852-2861.

9. ANSSI France. EBIOS 2010 : Expression des besoins et Identification des Objectifs de Sécurité. <http://www.ssi.gouv.fr/>, 2010.

10. Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI). France : EBIOS 2010 - Expression des Besoins et Identification des Objectifs de Sécurité. Méthode de Gestion des Risques, 2010.

11. ISO/IEC 27005:2011 Information technology - Security techniques - Information security risk management, 2011.

12. Ministerio de Administraciones Públicas (MAP). España: MAGERIT versión 3 – Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información. I Método, 2012.

13. Alberts, C., Dorofee, A.: Managing Information Security Risk. The OCTAVE Approach. Addison Wesley, 2005.

14. Bundesamt für Sicherheit in der Informationstechnik (BSI) Deutschland: IT Baseline Protection Manual, 2000.

15. National Institute of Standards and Technology, NIST SP 800-30, "Risk Management Guide for Information Technology Systems," Gaithersburg, MD, July 2002.

16. Eurocontrol. Eurocontrol Safety Regulatory Requirement. Eurocontrol Safety Regulation Commission, 2001.

17. Siemens - Insight Consulting: The Logic behind CRAMM's Assessment of Measures of Risk and Determination of Appropriate Countermeasures, 2005.

18. Sadvandi S., Chapon N., Pietre-Cambacédes L. Safety and security interdependencies in complex systems and sos: Challenges and perspectives // Complex Systems Design & Management. – 2012. – pp. 229-241.

19. Johnson D. Performing a Cybersecurity Risk Assessment as a Component of the PHA - <http://www.exida.com/Blog/performing-a-cybersecurity-risk-assessment-as-a-component-of-the-pha>

20. Marhavilas P.K., Koulouriotis D., Gemeni V. Risk analysis and assessment methodologies in the work sites: On a review, classification and comparative study of the scientific literature of the period 2000-2009 // Journal of Loss Prevention in the Process Industries. – 2011. – vol. 24. – pp. 477-523.

21. Kaiser B., Liggesmeyer P., Mackel, O. A New Component Concept for Fault Trees. // Proceedings of the 8th Australian Workshop on Safety Critical Systems and Software (SCS). – 2003. – vol. 33, pp. 37-46.

22. Securing Electricity Supply in the Cyber Age: Exploring the Risks of Information and Communication Technology in Tomorrow's Electricity Infrastructure / Z. Lukszo, G. Deconinck, Margot P.C. Weijnen (Eds.), Springer Science+Business Media B.V. 2010. – 187 p.

23. International Electrotechnical Commission, "IEC 60812: Analysis techniques for system reliability - procedure for failure mode and effects analysis (FMEA)," 2006.

24. Risk Assessment and Risk Management for the Chemical Process Industry / Greenberg H.R., Cramer J.J. (Eds), Van Nostrand Reinhold, New York, 1991.

25. Stamatis D.H. Failure Mode and Effect Analysis: FMEA from Theory to Execution / American Society for Quality (ASQ), Milwaukee, WI, 1995.

26. Gaudoin O., Ledoux J. Modélisation aléatoire en fiabilité des logiciels / Hermès Science Publications-Lavoisier, Paris, 2007.

27. Hadjsaid N., Tranchita C., Rozel B., Viziteu M., Caire R. Modeling cyber and physical interdependencies – applications in ICT

and power grids // Proc. Power Syst. Conf. Expos. – 2009. – doi:10.1109/PSCE.2009.4840183

28. Ministry of Defence (United Kingdom), “HAZOP studies on systems containing programmable electronics part 2 general application guidance,” May 2000.

29. Nolan D.P. Safety and Security Review for the Process Industries Application of HAZOP, PHA, What-IF and SVA Reviews / Gulf Professional Publishing, Elsevier Inc, 2015. – 186 p.

30. Ayyub B.M. Risk analysis in engineering and economics / Chapman & Hall/ CRC, 2003.

31. Singh R., Shenoy P., Natu M., Sadaphal V., Vin H. Predico: a system for what-if analysis in complex data center applications // Proceedings of the 12th International Middleware Conference, December 12-16, 2011, Lisbon, Portugal. – 2011. – pp. 120-139.

32. Landau K., Rohmert W., Brauchler R. Task analysis. Part I – Guidelines for the practitioner // International Journal of Industrial Ergonomics. – 1998. – vol. 22(1-2). – pp. 3-11.

33. Brauchler R., Landau K. Task analysis. Part II – the scientific basis (knowledge base for the guide) // International Journal of Industrial Ergonomics – 1998. – vol. 22(1-2). – pp. 13-35.

34. Doytchev D. E., Szwillus, G. Combining task analysis and fault tree analysis for accident and incident analysis: a case study from Bulgaria // Accident Analysis and Prevention. – 2008. – doi: 10.1016/j.aap.2008.07.014.

35. Lippmann R., Ingols K. An annotated review of past papers on attack graphs. MIT Lincoln laboratory Project Report, 31 March 2005. Project Report ECS-TR-2005- 054.

36. Noel, S. & Jajodia, S. Metrics suite for network attack graph analytics. In Proceedings of the 9th Annual Cyber and Information Security Research Conference (CISR 2014), Robert K. Abercrombie and J. Todd McDonald (Eds.). ACM, New York, NY, USA, 5-8. doi:10.1145/2602087.2602117

37. Johnson W.G. The management oversight and risk tree // Prepared for the U.S. Atomic Energy Commission, 1973.

38. World energy perspectives. The road to resilience: Managing cyber risks. 2016 - <https://www.worldenergy.org/wp->

[content/uploads/2016/09/20160926_Resilience_Cyber_Full_Report_WEB-1.pdf](#)

39. Cozzani V., Antonioni G., Spadoni G. Quantitative assessment of domino scenarios by a GIS-based software tool // *Journal of Loss Prevention in the Process Industries*. – 2006. – 19(5). – pp. 463-477.

40. Garvey P., Pinto A. *Introduction to Functional Dependency Network Analysis*, MIT, 2009.

41. Garvey P., Pinto A. *Advanced Risk Analysis in Engineering Enterprise Systems*, CRC Press. – 2012.

42. Guariniello C., DeLaurentis D. Communications, information, and cyber security in systems-of-systems: Assessing the impact of attacks through interdependency analysis // *Procedia Computer Science*. – 2014. – vol. 28. – pp. 720-727.

43. Guariniello C., DeLaurentis D. Dependency analysis of system-of-systems operational and development networks // *Procedia Computer Science*. – 2013. – vol. 16. – pp. 265-274.

44. Guariniello C., DeLaurentis D.A. Maintenance and recycling in space: functional dependency analysis of on-orbit servicing satellites team for modular spacecraft // *AIAA SPACE 2013 Conference and Exposition*. – 2013. – p. 5327.

45. Guariniello C., DeLaurentis D. Dependency Network Analysis: Fostering the Future of Space with New Tools and Techniques in Space Systems-of-Systems Design and Architecture. // *IAF International Astronautical Congress*, 2013.

46. Jae M., Apostolakis G.E. The Use of Influence Diagrams for Evaluating Severe Accident Management Strategies // *Nuclear Technology*. – 1992. – vol. 99. – pp. 142-157.

47. Liu H., Zhang C., Wan Y. Research on Information Engineering Surveillance Risk Evaluation Based on Probabilistic Influence Diagram // *2nd IEEE International Conference on Information Management and Engineering, ICIME 2010*. – 2010. – vol. 5. – pp. 362-366.

48. Jansma R.M., Fletcher S.K., Murphy M.D., Lim J.J., Wyss G.D. *Risk-Based Assessment of the Surety of Information Systems* // SAND96-2027, Sandia National Laboratories, Albuquerque, NM, July 1996.

49. McCormick N.J. Reliability and Risk Analysis: Methods and Nuclear Power Applications / Academic Press, New York, 1981.

50. Devooght J., Smidts, C. Probabilistic Reactor Dynamics – I: The Theory of Continuous Event Trees // Nuclear Science and Engineering. – 1992. – vol. 111. – pp. – 229-240.

51. Devooght J., Smidts, C. Probabilistic Reactor Dynamics – III: A Framework for Time-Dependent Interaction Between Operator and Reactor During a Transient Involving Human Error // Nuclear Science and Engineering. – 1992. – vol. 112. – pp. 100-113.

52. Tidwell V., Cooper J.A., Silva C.J., Jurado S. Threat Assessment of Water Supply Systems Using Markov Latent Effects Modeling // Sandia National Laboratories Report SAND2004-0818C, Albuquerque, New Mexico, 2004.

53. Tranchita C., HadjSaid N., Torres A. Risk assessment for power system security with regard to intentional events / Thesis to obtain the degree of Doctor from the Grenoble Institute of Technology and the Los Andes University, 2008.

54. Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference / Morgan Kaufmann, San Mateo, CA, 1988.

55. Pearl J. Bayesian networks / Arbib, M. (ed.) Handbook of Brain Theory and Neuronal Networks, MIT Press, Cambridge, MA. – U.S. Nuclear Regulatory Commission NRC, 2000.

56. Gashi I., Popov P., Stankovic V., Uncertainty explicit assessment of off-the-shelf software: A Bayesian approach // Information and Software Technology. – vol. 51 (2). – pp. 497-511

57. Han S.Y., DeLaurentis D. Development Interdependency Modeling for System-of-Systems (SoS) using Bayesian Networks: SoS Management Strategy Planning // Procedia Computer Science. – 2013. – vol. 16. – pp. 698-707.

58. Risk Assessment and Decision Analysis with Bayesian Networks / Norman Fenton, Martin Neil, 2013.

59. Haimes Y.Y. Models for risk management of systems of systems // Int. J. System of Systems Engineering. – 2008. – Vol. 1, Nos. 1/2. – pp. 222-236.

60. Modelling Cyber Security Risk Across the Organization Hierarchy <http://www.track->

assets.com/files/Modelling%20Cyber%20Security%20Risks%20across%20the%20organization%20Mar16.pdf

61. Fovino I.N., Masera M., De Cian A. Integrating Cyber Attacks within Fault Trees // Reliability Engineering and System Safety. – 2009. – vol. 94, no. 9. – pp. 1394-1402.

62. Kriaa S., Bouissou, M., Pietre-Cambacedes L. Modeling the Stuxnet attack with BDMP: Towards more formal risk assessments. // Proceedings of the 7th International Conference on Risk and Security of Internet and Systems (CRISIS 2012), pp. 1–8 (October 2012), doi:10.1109/CRISIS.2012.6378942.

63. Karnouskos S. Stuxnet worm impact on industrial cyber-physical system security // Proceedings of the 37th IEEE Annual Conference on Ind. Electronics Soc. (IECON 2011). – 2011. – pp. 4490–4494, doi:10.1109/IECON.2011.6120048.

64. Sabaliauskaite G., Mathur A.P. Aligning Cyber-physical System Safety and Security // Cardin, M.A., Krob, D., Cheun, L.P., Tan, Y.H., Wood, K. (eds.) Complex Systems Design & Management Asia 2014. LNCS. – 2015. – pp. 41-53.

65. Kumar R., Stoelinga M. Quantitative security and safety analysis with attack-fault trees, 2017.

66. Sheyner O., Haines J., Jha S., Lippmann R., Wing J. Automated generation and analysis of attack graphs // Proc. of the IEEE Symposium on Security and Privacy, 2002.

67. Ou X., Boyer W., McQueen M. A scalable approach to attack graph generation // Proc. of the ACM Conference on Computer and Communications Security (CCS), 2006.

68. LeMay E., Ford M., Keefe K., Sanders W. H., Muehrke C. Model-based security metrics using ADversary View Security Evaluation (ADVISE) // Proc. of the Conference on Quantitative Evaluation of SysTems (QEST), 2011.

69. Vu A.H., Tippenhauer N.O., Chen B., Nicol D.M., Kalbarczyk Z. CyberSAGE: A Tool for Automatic Security Assessment of Cyber-Physical Systems // Proc. of the Conference on Quantitative Evaluation of SysTems (QEST), 2014.

70. Kordy B., Mauw S., Radomirovic S., Schweitzer P. Foundations of attack-defense trees // Proc. of the conference on Formal Aspects of Security and Trust (FAST), 2011.

71. Beim G. K., Hobbs B.F. Event tree analysis of lock closure risks // *Journal of Water Resources Planning and Management*. – 1997. – vol. 123(3). – pp. 169-178.

72. Hong E.S., Lee I.M., Shin H.S., Nam S.W., Kong J.S. Quantitative risk evaluation based on event tree analysis technique: Application to the design of shield TBM // *Tunnelling and Underground Space Technology*. 2009. – vol. 24(3). – pp. 269-277.

73. Ngai E.W.T., Wat F.K.T. Fuzzy Decision Support System for Risk Analysis in e-Commerce Development // *Decision Support Systems*. – 2005. – vol. 40. – pp. 235-255.

74. Dondo M. A fuzzy risk calculations approach for a network vulnerability ranking system / *Defence R & D Canada-Ottawa*, 2007.

75. Yeboah-Boateng, E. O. Cyber-Security Challenges with SMEs in Developing Economies: Issues of Confidentiality, Integrity & Availability / (CIA). (1 ed.) Institut for Elektroniske Systemer, Aalborg Universitet, 2013.

76. Kaplan S., Garrick J.B., On the Quantitative Definition of Risk // *Risk Analysis*. – 1981. – vol. 1(1), pp. 11-37.

77. Chockalingam S., Hadžiosmanović D., Pieters W., Teixeira A., van Gelder P. Integrated Safety and Security Risk Assessment Methods: A Survey of Key Characteristics and Applications // 11th International Conference on Critical Information Infrastructures Security, CRITIS 2016, 10-12 October 2016, Paris, France.

78. Macher G., Sporer H., Berlach R., Armengaud E., Kreiner C. SAHARA: A security-aware hazard and risk analysis method // *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2015, March 2015. – pp. 621–624.

79. Scandariato R., Wuyts K., Joosen W. A Descriptive Study of Microsoft's Threat Modeling Technique // *Requirements Engineering*. – 2015. – vol. 20, no. 2. – pp. 163-180.

80. Threat risk modeling
https://www.owasp.org/index.php/Threat_Risk_Modeling#STRIDE

81. Raspotnig C., Karpati P., Katta V. A combined process for elicitation and analysis of safety and security requirements / *Lecture Notes in Business Information Processing*, 2012.

82. Roy A., Dong S. K., Trivedi K. S. Scalable optimal countermeasure selection using implicit enumeration on attack

countermeasure trees // Proceedings of the 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012). – 2012. – pp. 1-12.

83. Roy A., Kim D.S., Trivedi K.S. Scalable Optimal Countermeasure Selection Using Implicit Enumeration on Attack Countermeasure Trees // Proceedings of the 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). – 2012. – pp. 1-12.

84. Schmittner C., Gruber T., Puschner P., Schoitsch E. Security application of failure mode and effect analysis (FMEA) // Proc. of the International Conference on Computer Safety, Reliability and Security (SAFECOMP), 2014.

85. Schmittner C., Ma Z., Smith P. FMVEA for Safety and Security Analysis of Intelligent and Cooperative Vehicles // Bondavalli, A., Ceccarelli, A., Ortmeier, F. (eds.) SAFECOMP 2014 Workshops. LNCS. – 2014. – vol. 8696, pp. 282-288.

86. Chen Y., Chen S., Hsiung P., Chou I.: Unified Security and Safety Risk Assessment – A Case Study on Nuclear Power Plant. In: Proceedings of the International Conference on Trusted Systems and their Applications (TSA). – 2014. – pp. 22-28.

87. Fovino, I.N., Masera, M.: Through the Description of Attacks: A Multi-Dimensional View // Gorski J. (eds.) SAFECOMP 2006. LNCS. – 2006. – vol. 4166. – pp. 15-28.

88. Steiner M., Liggesmeyer P. Combination of Safety and Security Analysis – Finding Security Problems that Threaten the Safety of a System. // Workshop on Dependable Embedded and Cyber-physical Systems (DECS). – 2013. – pp. 1-8.

89. Alberts C., Woody C., Dorofee A. Introduction to the Security Engineering Risk Analysis (SERA) Framework. CMU/SEI-2014-TN-025 / Software Engineering Institute, Carnegie Mellon University, 2014.

90. Mead N.R., Woody C.C. Cyber Security Engineering A Practical Approach for Systems and Software Assurance / Pearson Education, Inc., 2017. – 330 p.

Summary

This chapter reviews numerous risk analysis techniques (other similar terms often used are tools, models, or methods) developed and applied in risk analysis across different industries, sectors and activities integrated into SoS. A brief overview summary is provided for each technique, including information on applications, procedures, strengths and limitations, and other characteristics. Detailed information about risk analysis techniques is provided on the base a number of publications.

29 Smart grid safety analysis and assurance

29.1 Introduction to smart grid safety issues

The development of smart grids is important if the global community is to achieve shared goals for energy security and reliability, economic development and climate change mitigation [1-15]. Smart grids enable increased demand response and energy efficiency, integration of renewable energy resources and electric vehicle recharging services, while reducing peak demand and stabilizing the electricity system. Smart grid is a very complicated, multilevel and dynamical system comprised of different ICT-based, interconnected and independent components. The Smart Grid is a next, naturally determined stage of electrical smart infrastructure evolution. The Smart Grid is an upgrade to the current electrical smart system, so it has all of the functionality of our current smart system plus several new functionalities [16] such as: self-healing, motivates and includes the consumer, resists attack, increases smart quality, accommodates all generation and storage options, enables electrical markets, optimizes assets and operates efficiently.

According to [17] major differences between the conventional electric smart system and the smart grid are shown in Fig. 29.1.

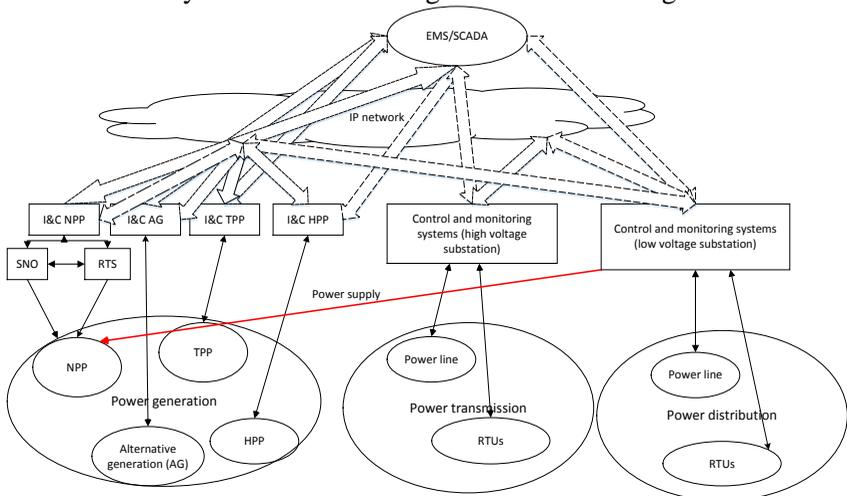


Fig. 29.1 Smart Grid

The conventional electric smart system consists of three components: generation (concentrated fire/water/atomic smart generation), distribution (smart transmission and distribution), and smart consumption by the customers. The flow of electric smart is unidirectional from upstream: generation, transmission, distribution, and then consumption. Smart grid determined as a fully automated smart delivery network that monitors and controls every customer and node, ensuring a two-way flow of electricity and information between the smart plant and the appliance, and all points in between.

Smart grid distributed intelligence, coupled with broadband communications and automated control systems, enables real-time market transactions and seamless interfaces among people, buildings, industrial plants, generation facilities, and the electric network.

Nuclear Power Plants (NPP) is an intrinsic part of future smart grid. The reliable operation of the NPP implies that the grid to which it connects is also efficient, safe and reliable and vice versa. NPPs supply large amounts of energy to the grid as well as relying on it to receive smart for crucial safety operations, especially during emergency conditions. The safe startup, operation and shutdown of NPPs require a reliable and stable smart supply from the smart grid ('off-site smart'). Smart grid stability remains a main issue of NPP safety. Smart grid introduces new risks for NPP safety. From other side NPP safety is influenced by safety of I&C NPP.

The electric substation is extremely strategic to smart grid operations. Compared to other systems in an electric utility network, the substation has the highest density of valuable information needed to operate and manage a smart grid. Considering the many smart grid road maps developed by companies and utilities smart grid substations will become the main objects of interest when smart grid starts. Substations are also important to energy utilities because there are lots of them and a substantial proportion of them are reaching the end of their useful operating life. In developing economies, the demand for new electric transmission and distribution infrastructure means that thousands of new substations are being designed and built.

Nowadays there are no differences among substations in respect to cyber security. But substations with critical loads, such as NPPs, should be given the highest level of importance. Substation security issues are important in the respect to NPP safety. The Smart Grid will introduce

several new security risks related to its communication requirements, system automation, new technologies, and data collection [18].

Hence new type of safety critical smart grid substation (CSGS) should be introduced and given more careful considerations in the respect of its cyber security. NPP and CSGS mutual influences (dependencies) are to be analyzed in order to assure NPP required safety level and mitigate all possible risks involved into their interactions.

Infrastructure interdependency modeling is a relatively new area of research and analysis, but recent events of both natural disasters and malicious acts have shown that the impact of these cross infrastructure relationships can be measured. Many approaches have been used to model infrastructure interaction including for example agent-based models [19], input-output models [20], neural networks [21] and scalable multi-graph methods [22]. As well as differing in their general approach these methods differ widely in the type, size and number of networks being considered. The approaches can be combined in a collective model where different infrastructure networks are encompassed in a single model structure or a distributed type where each network is modeled separately and the results are passed between the models according to some mediating mechanism.

Agent-based models are computer simulations of systems where entities called agents are used to represent The key characteristic of the agent and the simulations is that each agent exists as an individual entity which maintains a state, senses input, and possesses rules of behavior that act upon the inputs and either modify the state or produce an output. the behavior of system components.

The paradigm of modeling and simulation is “garbage in, garbage out.” Having credible and traceable data available to use is key to infrastructure and interdependency modeling. Gathering information on a particular infrastructure is possibly the most significant challenge.

Input-output inoperability models (IIM) are financial models that have been used for analyzing cascade effects in critical infrastructure systems [23]. IIM uses inoperability levels to describe the state of each infrastructure network. A neural network is a collection of densely interconnected simple computing units called artificial neurons loosely based on the architecture of the human brain. Neural networks have been used for reliability analyses on interdependent lifelines [24].

Diversity is well known technique used for I&C NPP safety assurance. According to [25, 26] diversity is one of the general principles used to decrease vulnerability against CCF and provide dependability of I&C. Diversity is used jointly with structure and temporal redundancy types to decrease risks of the CCF.

Application of diversity principle implies implementation and using two or more redundant systems or components to perform an identified function, and redundant systems (components) implemented the attributes of diversity (design, signal, functional, equipment and others).

There are many different approaches for I&C diversity assessment. Basically, they are divided on two groups: the theoretical-set and metric-oriented methods and expert-oriented methods [27-29].

The theoretical-set and metric-oriented methods are based on: Euler's diagram for sets of version design, physical and interaction faults (including vulnerabilities for assessment intrusion-tolerance); matrix of diversity metrics for sets of different faults (individual, group and absolute faults of versions); calculation of diversity metrics by use of Euler's diagrams or other data about results of testing and faults of different versions. Probabilistic methods use reliability block-diagrams (RBDs), their modifications (survivability and safety block-diagrams), Markovian chains, Bayesian method, etc.

Expert-oriented methods use two groups of metrics: diversity metrics for direct assessment of versions and I&C reliability and safety (direct diversity metrics); indirect diversity metrics (product complexity metrics and process metrics); values of these metrics may be used to assess direct diversity metrics. Expert methods are founded on interval mathematics-based assessment of diversity, soft computing-based assessment (fuzzy logic, genetic algorithms), risk-oriented approach and so on. Based on analysis of available publications, we could make a conclusion on the lack of approaches, which deal with uncertainties inherited to expert-based diversity assessment.

Common disadvantages of various approaches discussed above are as follows:

- Smart grid security issues and I&C NPP safety are considered separately. I&C NPP and CSGS safety (security) approaches are kept independent;

- CSGS safety (security) is considered a static attribute;

No consideration provided for mutual influences between CSGS and I&C NPP. Generally, there is a lack of publications devoted to smart grid safety assessment considering influences.

Beside there is also lack of approaches that consider the diversity as a means of smart grid safety assurance. Why not to use the best practices from nuclear safety techniques?

29.2 Smart grid security and safety interrelation

The smart grid always needs to be available, and locking the system during an emergency could cause NPP safety issues. The smart grid security objectives being evaluated are confidentiality, integrity, and availability. In the electrical smart system, electricity must always be available, so this is the most important security objective. Integrity is the next important security objective followed by confidentiality.

Integrity is the next important security objective in the Smart Grid. The Smart Grid uses data collected by various field sensors and agents. This data is used to monitor the current state of the electrical smart system. The integrity of this data is very important. Unauthorized modification of the data, or insertion of data from unknown sources can cause failures or damage in the electrical smart system. The electricity in the smart grid not only needs to be available, but it also has to have quality. The quality of the electrical smart will be dependent on the quality of the current state estimation in the smart system.

An intelligent smart grid relies on real-time, high-bandwidth, two-way open communications to control and monitor smart flows. These communications make the smart grid viable, but also open it to cyber attack. Smart grid technologies will introduce millions of new intelligent components to the electric grid that communicate in much more advanced ways than in the past, namely two-way via open protocols. Because of these open communications among large number of devices, CSGS cyber security becomes critical. Smart grid security is determined by security of CSGS. If not secured CSGS becomes unavailable for NPP connected to it, then it might cause some risks for NPP, and smart grid safety as a whole.

29.2.1 Principles of smart grid safety analysis

The smart safety analysis is carried out taking into consideration principles of dynamism, hierarchy, uncertainty, and influence (interaction) of subsystems.

Principle of dynamical analysis assumes to record changes of system criticality during the operation as a result of changes of its states (transition to state of non-operability). At each stage of life cycle, the criticality assessment specification and adjustment of criticality matrices [30], taking into consideration probable changes, are carried out.

The principle of hierarchy assumes representation of grid structure as a hierarchy.

The principle of influence of subsystem failures of i -level (on subsystem failure criticality of the same level) and influence on subsystems of $(i-1)$ -level (higher) is important.

The safety of all influenced subsystems must be reconsidered.

The principle of uncertainty takes into consideration information incompleteness and uncertainty related to the conditions that cause PG accidents.

The principle of the weakest link risk flow is based on assumption that PG safety might be evaluated on risks associated with the weakest link of the grid.

The SG safety is an integral value composed of grid systems safety values. The grid safety is determined by uncontrolled mutual influence among grid systems. It is worth to note that influence exists on all SG levels and have to be taken into consideration when providing grid systems safety.

29.2.2 The approach for influence formalization in smart grid

The formalization of influences between smart grid systems is very helpful for its safety assessment based on criticality matrices. Generally, criticality matrix is represented as FMECA table. The traditional FMECA [31] is the most widely used reliability analysis technique on the initial stages of system development.

For example, if smart grid system S_1 consists of three subsystems S_{11} , S_{12} , S_{13} then criticality matrix which represents the system S_1 might be presented as shown in the Table 29.1.

Table 29.1 Criticality matrix for system S_1

System S_1		Severity of Failure Mode		
Failure rate		H	M	L
	H		S_{12}	
	M			S_{13}
	L	S_{11}		

Traditionally, the criticality assessment is performed by calculating the criticality accident (failure) as a product of its severity and probability:

$$Crt(S_i) = P(S_i) \times Sev(S_i), \tag{29.1}$$

where S_i is SG system; $P(S_i)$ is probability of S_i accident (failure); $Sev(S_i)$ – severity of accident consequences.

According to the principle of hierarchy, the smart grid structure might be represented as a hierarchy. In this case, the safety of SG systems of higher level hierarchy might be evaluated as a sum of criticalities of smart grid systems of lower level hierarchy. For example, considering the criticalities of S_{11} , S_{12} , S_{13} as subsystems of S_1 , its total criticality could be calculated as:

$$\begin{aligned} Crt(S_1) &= P(S_{11}) \times Sev(S_{11}) + \dots + P(S_{13}) \times Sev(S_{13}) = \\ &= \sum_i^I \sum_j^J P(S_{ij}) \times Sev(S_{ij}). \end{aligned} \tag{29.2}$$

Another approach might introduced considering the weakest link of PG. In this case the system total criticality might be equaled its weakest link criticality.

It is suggested to treat criticality as PG system’s safety inverse value. The more system criticality the less its safety and vice versa.

It should be noted that criticality matrix might be used to represent different states of nature and its influence on SG systems. It is suggested to use the environmental FMECA where different natural

hazards (earthquake, flooding, etc.) are considered as different failures modes characterized by its probability and severity for the nearest PG systems. This probability of system accident (natural disaster) and its severity could be handled as linguistic or numerical variable. Hence, criticality is also treated correspondently either linguistic or numerical variable.

A linguistic variable is characterized by a quintuple $(x, T(x), U, G, M)$ in which x is the name of variable; $T(x)$ is the term set of x , that is, the set of names of linguistic values of x with each value being a fuzzy number defined on U ; G is a syntactic rule for generating the names of values of x ; and M is a semantic rule for associating with each value its meaning.

The set of state Ω_{S_i} of any PG system S_i is determined as:

$$\Omega_{S_i} = \{ \text{Crt}(S_i)=\text{High}, \text{Crt}(S_i)=\text{Medium}, \text{Crt}(S_i)=\text{Low} \}. \quad (29.3)$$

Any accident or failure of smart grid system leads to the change of criticality of all connected systems due to principle of risk flow. When a failure of one system occurs, the criticalities of all dependent systems are recalculated due to influences between systems.

The prognosis and assessment of SG system service life, based on real time measurements, will help to identify grid systems most likely to fail. The potential estimation methods and equipment service life prediction for complicated systems consist of deterministic, statistical, physical-statistical and methods based on expert knowledge. These methods are used to predict the probability of accident of any system S_{ij} of S_i .

This criticality assessment is used to support the subjective expert judgment expressed by linguistic variable on the initial smart grid system state. The more system criticality calculated on (2) the more confident expert's opinion on the criticality of each node of PG.

29.2.3 Types of influences between smart grid and I&C NPP

CSGS – I&C NPP influences can be categorized according to various dimensions in order to facilitate their identification, understanding and analysis. Six dimensions have been identified in [9]. They correspond to: a) the type of CSGS – NPP interdependencies (physical, cyber,

geographic, and logical); b) the smart grid (NPP) environment (technical, business, political, legal, etc.); c) the couplings among the smart grid systems and their effects on their response behavior (loose or tight, inflexible or adaptive); d) CSGS (NPP) characteristics (organizational, operational, temporal, spatial); e) the state of operation (normal, stressed, emergency, repair), the degree to which the smart grid systems are coupled; f) the type of failure affecting the CSGS – NPP states (common cause, cascading, escalating).

The NPP as a part of smart grid constantly interacts with critical smart grid substations. Generally, influence is an ability of I&C NPP (critical smart grid substations) to determine the state, characteristics and behavior of CSGS (NPP).

Generally influences between CSGS – I&C NPP could be classified into different types [19]:

1. Physical influence $I_{phys}^{NPP}(t)$ - a physical reliance on smart flow from smart grid through CSGS to NPP and vice versa.

There are some risks events resulted from uncontrolled changes in this type of influence:

- Loss of off-site smart. A loss of off-site smart interrupts smart to all in-plant loads such as pumps and motors, and to the NPP's safety systems. As a protective action, safety systems will trigger multiple commands for reactor protective trips (e.g. turbine and generator trip, low coolant flow trip, and loss of feedwater flow trip). The reactor protection system will also attempt to switch to an alternate off-site smart source to remove residual heat from the reactor core. If this fails, in-plant electrical loads must be temporarily smarted by batteries and stand-by diesel generators until off-site smart is restored. Trip of an NPP causing degraded grid frequency and voltage. Smart grid substation state due to this event denoted as $S_{I(ph)}^{Subst1}$. NPP state stipulated by this grid event is denoted as $S_{I(ph)}^{NPP}$.

Unless additional smart sources are quickly connected to the grid, this can degrade the grid's voltage and frequency and thus the off-site smart supply to the NPP. The degraded voltage and frequency on the grid can potentially result in NPP protection system disconnecting the degraded off-site smart to the NPP.

A load rejection is a sudden reduction in the electric smart demanded by the grid. Such a reduction might be caused by the sudden opening of an interconnection with another part of the grid that has carried a large load. Smart grid substation state corresponding to this event denoted as $S_{2(ph)}^{Subst1}$.

An NPP is designed to withstand load rejections up to a certain limit without tripping the reactor. An NPP's ability to cope with a load rejection depends on how fast the reactor smart can be reduced without tripping and then how fast the reactor smart output can be increased back to the original level when the fault is cleared. NPP state stipulated by this smart grid event is denoted as $S_{2(ph)}^{NPP}$.

A loss of load is a 100% load rejection, that is the entire external load connected to the smart station is suddenly lost, or the breaker at the station's generator output is opened. Smart grid substation state for this event denoted as $S_{3(ph)}^{Subst1}$. Under this severe condition, it may still be possible to 'island' the NPP so that it smarts only its own auxiliary systems. During this 'house-load' operating mode, the reactor operates at a reduced smart level that is still sufficient to assure enough electricity for its own needs, typically 5% of full smart. NPP state stipulated by this event is denoted as $S_{3(ph)}^{NPP}$. NPP also could influence the smart grid state. For example, if a large NPP (e.g. 10% of the grid's total generating capacity) trips unexpectedly, the result can be a significant mismatch between generation and load on the grid.

NPP state stipulated by this event is denoted as $S_{4(ph)}^{NPP}$. Smart grid substation state corresponding to this event denoted as $S_{4(ph)}^{Subst1}$. System physical states are characterized by values of its operational parameters and processes used to energy generation and transformation. Table 29.2 represents a set of possible NPP states due to smart grid physical influence.

Table 29.2 NPP states (in physical domain) due to smart grid physical influence

$S_{1(ph)}^{NPP}$	Reactor shutdown due to <i>Loss of off-site smart</i>
-------------------	---

$S_{2(ph)}^{NPP}$	Reactor shutdown due to grid load rejection
$S_{3(ph)}^{NPP}$	House-load' operating mode caused by loss of load

There are some risks associated with NPP transition to state $S_{3(ph)}^{NPP}$. In case of either common cause failures of diesel generators or if off-site smart restoration time is more than operation time determined by batteries capacities this design basis accident could grow to nuclear accident. This is a main reason for this event analysis and preventing during NPP operation.

Table 29.3 represents a set of possible states of smart grid substation (in physical domain).

Table 29.3 CSGS states (in physical domain)

$S_{1(ph)}^{Subst}$	Stable operation of substation. Its state correlates with load rejection of the electric smart demanded by the grid
$S_{2(ph)}^{Subst}$	Unstable operational substation parameters due to smart grid degraded voltage or frequency
$S_{3(ph)}^{Subst}$	Complete substation outage due to smart grid faults (Loss of off-site smart)

2. Informational $I_{inf}^{NPP}(t)$ - a reliance on information transfer between NPP and critical smart grid substation (dependencies between I&C CSGS and I&C NPP states). State of both I&Cs depends on information transmitted through the information infrastructure. Informational dependencies exist due to I&C NPP and I&C CSGS electronic, informational links. Critical smart grid substations which provide information for NPP operator could be considered as a critical information infrastructure. This type of influence stipulates dependencies between informational state of I&C smart grid substation and NPP I&C's information state.

Different I&C CSGS information states are highly determined by its cyber security level. Cyber security refers to arrangements to ensure that I&C CSGS equipment is reasonably secure against accidental or

malicious actions that may change the intended operation of CSGSs (its informational state). Since the NPP safety and control systems generally do not interact with network based systems outside the plant, susceptibility for external malicious attack is limited. But cyber attacks can affect the integrity and the availability of CSGS connection to the NPP.

The vulnerability for such attack in smart grid is higher than in traditional smart grid because CSGSs communicate through wireless networks that might be not secure enough. There are some examples of possible cyber attacks from the smart grid to the NPP:

- Fake signals coming from the grid asking the plant to trip or to reduce output;
- Transmission of wrong voltage set points that could make certain voltage sensitive equipment inoperable or cause premature trips;
- Wrong smart and voltage measurements;
- Other changes to plant parameters or plant status that could initiate undesirable behavior.

Table 29.4 represents a set of possible I&C CSGS states (in informational domain).

Table 29.4 A set of possible of I&C CSGS states (in informational domain)

CSGS states (in informational domain)	
$S_{1(\text{inf})}^{\text{subst}}$	Stable information state. No data compromised. I&C services are correct. I&C availability and integrity meet requirement
$S_{2(\text{inf})}^{\text{Subst}}$	Unstable informational state. Some data might be compromised due to insufficient cyber security level. There are some evidences of I&C incorrect services performing. I&C availability (integrity) does not meet requirements
$S_{3(\text{inf})}^{\text{Subst}}$	Control data is likely corrupted. Control data might affect substation operational logic. I&C services are incorrect. I&C availability and integrity does not meet requirement

A set of I&C control data is suffered a Loss of Integrity if some event has caused it to be corrupted or incorrectly altered.

Generally, I&C could change its informational state due to its software, hardware and communications equipment failures or incompatibility; loss of control data integrity, availability, confidentiality due to cyber attacks and natural disasters, etc.

Table 29.5 represents a set of possible state of I&C NPP (in informational domain).

Table 29.5 A set of possible I&C NPP states (in informational domain)

NPP I&C states (in informational domain)	
$S_{1(\text{inf})}^{\text{NPP}}$	I&C operates correctly. All control data is secure. I&C services are correct. I&C availability and integrity meet requirement
$S_{2(\text{inf})}^{\text{NPP}}$	I&C operates correctly, but some data might be corrupted There are some evidences of I&C incorrect services performing. I&C availability (integrity) does not meet requirements
$S_{3(\text{inf})}^{\text{NPP}}$	I&C does not operate correctly. I&C services are incorrect. I&C availability and integrity does not meet requirement

3. Level of cyber security determines the informational state of both I&Cs. Cyber attacks also have the potential to cause major damage to connected electrical equipment.

Hence the new type of influence should be introduced for smart grid-NPP interaction assessment. Information – physical influence $I_{\text{inf}}^{\text{NPP}}(t)$ - an influence of information state of I&C CSGS on physical (technical) state of NPP equipment. Being corrupted information of I&C CSGS might influence physical (technical) state of corresponding NPP equipment. As an example, transmission and distribution systems that have circuit breakers that can open and reclose within 12 to 15 cycles have the potential to cause considerable damage to rotating plant (generators, motors) as a result of a cyber attack that caused such operation. This is because the rotating plant is likely to speed up or slow down during the brief time that the circuit breaker is open, so that the rotating plant will be out of phase with voltage on the grid at the

instant of reclose; this will cause a large transient torque on the plant which can cause physical damage. This potential vulnerability could be exploited through digital protection and control devices such as protective relays, programmable logic controllers, bay controllers and other digital devices that can control circuit breaker operations.

These devices are common protection and control devices found in process control systems and electricity grid substations. The electrical generators, motors and pumps could suffer significant damage if this vulnerability is successfully exploited and as a consequence nuclear safety could be compromised.

Figure 29.2 represents a mutual influence between informational states of I&C CSGS and physical states NPP.

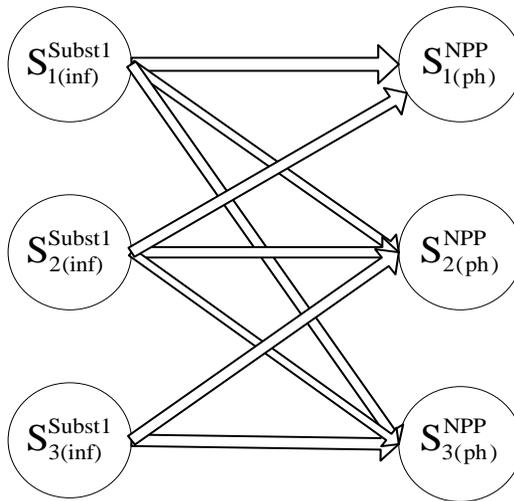


Fig. 29.2 Mutual influence between informational state of I&C CSGS and physical state NPP equipment

Electrical equipment in NPP could be affected if the CSGS in the zone of influence of the nuclear smart plant are not secure. It may be possible to gain access to digital equipment in the substation to execute such malicious control either through communication networks, or through local portals at substations intended for computer connectivity.

4. Geographic $I_{geo}^{NPP}(t)$ - a local environmental event affects components of NPP-smart grid substation (usually the transmission lines) due to physical proximity. Given this influence, events such as an explosion or fire (NPP accident) could affect normal operation of smart grid substation (as an example radioactive contamination could prevent substation from regular maintenance procedure).

In its turn the closer substation to NPP switchyard more likely substation failures affect normal operation of NPP. A grid fault near to NPP which causes a transient depression of the grid voltage to a very low value could lead to trip of small electrical auxiliaries. Some electronic equipment may see the voltage as a loss of supply and stop operating.

This type of influence represents new type of non-operational CSGS state when NPP accident consequences might prevent it from normal operation. These risks have to be evaluated.

5. Organizational $I_{org}^{NPP}(t)$ (influences though policy, regulation, markets). This type of influence could lead to risks caused by uncoordinated NPP – smart grid procedures. Risk sources are human actions stipulated by imperfect policy, regulation, procedures. Market requirements are additional risk sources. It forces NPP (grid operator) to push NPP (grid equipment) operate close to its physical parameters (safety margin).

There is a need for close cooperation between the grid operator and NPP operator in maintenance planning as well as outage planning. Experience has shown that a formal agreement on coordination of planning, including definition of responsibilities is beneficial in ensuring the reliability of the offsite smart from the grid. It is a particularly important to coordinate maintenance activities on plant safety systems with transmission maintenance.

29.3 CSGS - NPP Information – physical influence assessment

It is suggested evaluating NPP and CSGS mutual interaction in information – physical domain using Bayesian belief networks (BBN). Bayesian belief networks are very effective for modeling situations where some information is already known and incoming data is

uncertain or partially unavailable (unlike rule-based or “expert” systems, where uncertain or unavailable data results in ineffective or inaccurate reasoning). These networks also offer consistent semantics for representing causes and effects (and likelihoods) via an intuitive graphical representation. An important fact to realize about Bayesian belief networks is that they are not dependent on knowing exact historical information or current evidence. A seminal advantage of Bayesian methods over frequency methods is that Bayesian methods can in principle always yield an answer, even in situations where frequency methods cannot be used.

It is considered two CSGSs connected to NPP. According to this approach, BBN is constructed for CSGS – NPP Information – physical influence assessment. Nodes (NPP, two CSGSs) are connected by links which represent correspondently physical influence (green color) from one CSGS and informational influence (yellow color) from second CSGS.

Figure 29.3 represents BBN for CSGSs and NPP connected to them.

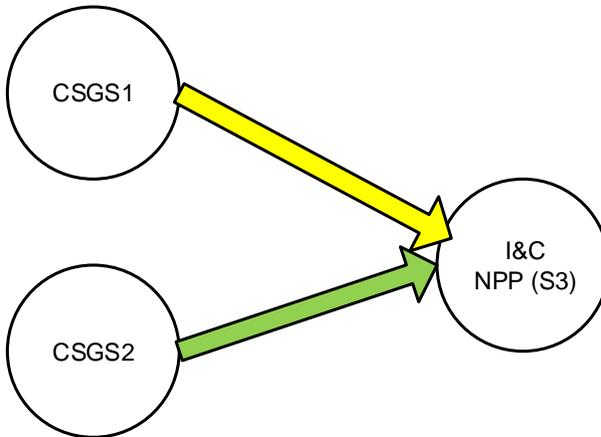


Fig. 29.3 BBN for CSGSs and I&C NPP connected to them.

Every node also has a conditional probability table, or CPT, associated with it. Conditional probabilities represent likelihoods based on prior information or past experience. A conditional probability is stated mathematically as the probabilities of NPP (child node) being at

state $S_{1(ph)}^{NPP}$ considering all possible informational and physical state combinations of CSGS (parents' nodes).

Fragment of linguistic CPT is shown in the Table 29.6.

Table 29.6 Fragment of CPT

S₁			S₂			S₃	
<i>Criticality (security states)</i>			<i>Criticality (availability states)</i>			<i>Criticality (safety states)</i>	
<i>H</i>	<i>M</i>	<i>L</i>	<i>H</i>	<i>M</i>	<i>L</i>	<i>H</i>	...
+			+			$P(\text{Crt}(S_3)=H/\text{Crt}(S_1)=H, \text{Crt}(S_2)=H)=High$...
+				+		$P(\text{Crt}(S_3)=H/\text{Crt}(S_1)=H, \text{Crt}(S_2)=M)=Low$...
					
	+		+			$P(\text{Crt}(S_3)=H/\text{Crt}(S_1)=M, \text{Crt}(S_2)=H)=Low$...

According to [10], probability of NPP being in state $S_{1(ph)}^{NPP}$ ($P(S_{1(ph)}^{NPP})$) depends on the combination of informational and physical states of CSGSs (parent's nodes) and might be determined as:

$$\begin{aligned}
 & P(S_{1(phys)}^{NPP}) = \\
 & P(S_{1(phys)}^{NPP} / (S_{1(inf)}^{Subst1}, S_{1(ph)}^{Subst2})) * P(S_{1(inf)}^{Subst1}) * P(S_{1(ph)}^{Subst2}) + \\
 & + P(S_{1(phys)}^{NPP} / (S_{1(inf)}^{Subst1}, S_{2(ph)}^{Subst2})) * P(S_{1(inf)}^{Subst1}) * P(S_{2(ph)}^{Subst2}) + \\
 & + P(S_{1(phys)}^{NPP} / (S_{1(inf)}^{Subst1}, S_{3(ph)}^{Subst2})) * P(S_{1(inf)}^{Subst1}) * P(S_{3(ph)}^{Subst2}) + \\
 & + P(S_{1(phys)}^{NPP} / (S_{2(inf)}^{Subst1}, S_{1(ph)}^{Subst2})) * P(S_{2(inf)}^{Subst1}) * P(S_{1(ph)}^{Subst2}) + \\
 & + P(S_{1(phys)}^{NPP} / (S_{2(inf)}^{Subst1}, S_{1(ph)}^{Subst2})) * P(S_{2(inf)}^{Subst1}) * P(S_{1(ph)}^{Subst2}) + \\
 & + P(S_{1(phys)}^{NPP} / (S_{2(inf)}^{Subst1}, S_{3(ph)}^{Subst2})) * P(S_{2(inf)}^{Subst1}) * P(S_{3(ph)}^{Subst2}) + \\
 & + P(S_{1(phys)}^{NPP} / (S_{3(inf)}^{Subst1}, S_{1(ph)}^{Subst2})) * P(S_{3(inf)}^{Subst1}) * P(S_{1(ph)}^{Subst2}) + \\
 & + P(S_{1(phys)}^{NPP} / (S_{3(inf)}^{Subst1}, S_{2(ph)}^{Subst2})) * P(S_{3(inf)}^{Subst1}) * P(S_{2(ph)}^{Subst2}) + \\
 & + P(S_{1(phys)}^{NPP} / (S_{3(inf)}^{Subst1}, S_{3(ph)}^{Subst2})) * P(S_{3(inf)}^{Subst1}) * P(S_{3(ph)}^{Subst2}).
 \end{aligned}$$

where $P(S_{1(phys)}^{NPP})$ - probability for NPP being at $S_{1(phys)}^{NPP}$ state;
 $P(S_{1(phys)}^{NPP} / (S_{1(inf)}^{Subst1}, S_{1(ph)}^{Subst2}))$ - conditional probability for NPP to be at $S_{1(phys)}^{NPP}$ state provided I&C CSGS1 being at $S_{1(inf)}^{Subst1}$ informational state and CSGS2 being at $S_{1(ph)}^{Subst2}$ physical state; $P(S_{1(inf)}^{Subst1}) (P(S_{3(ph)}^{Subst2}))$ - probability for I&C CSGS1 being at $S_{1(inf)}^{Subst1} (S_{3(ph)}^{Subst2})$ state is determined by expert

Generally, probability of I&C CSGS1 being at the informational state $S_{1(inf)}^{Subst1}$ might be calculated as:

$$\begin{aligned}
 P(S_{1(inf)}^{Subst1}) &= P(Sec_l) \times P(loss_Int / Sec_l) \times \\
 & P(loss_avail / loss_Int \wedge Sec_l)
 \end{aligned}$$

where $P(Sec_l)$ - probability of CSGS security level being evaluated as (High, Medium, Low);

$P(loss_Int / Sec_l)$ - probability of loss of I&C integrity provided CSGS security level is (High, Medium, Low);

$P(\text{loss_avail} / \text{loss_Int} \wedge \text{Sec_I})$ - probability of loss I&C availability provided CSGS security level is (High, Medium, Low) and loss of I&C integrity.

Thus, knowing the states of CSGS1 and CSGS2 we can assess the criticality state of NPP. The NPP safety state obtained during assessment might be not acceptable. This might happen due to high-level risks caused by influences between systems in smart grid. The smart grid owners shall evaluate these risks and take the decisions to downplay the negative influence between systems.

29.4 Diversity as a means for smart grid safety assurance

29.4.1 Diversity as a means for NPP safety assurance

We suggest using the diversity as a means for smart grid safety assurance considering the different types of influences between smart grid and NPP.

Considering the fact that some NPPs were built many years ago, the main reasons for the NPP I&C modernization are a necessary safety evolution, an operational improvement relating to production efficiency, maintenance considerations; a poor original design with defects, irreversible effects of equipment ageing, periodic renewal, replacement impossible due to obsolescence, lack of support, etc.

Whatever the reasons for such NPP I&C modernization, consideration should be given to the effects of that modernization, which ensure having no impacts that would compromise the safety of the plant and fulfill the requirements for I&C diversity.

The I&C systems in NPPs, associated with reactor protection and safety-system actuations (hereinafter named as Reactor Trip System), typically consist of several elements, such as process sensors, transmitters, sensing lines, and cabling as well as various logic units and switching devices. The RTS's primary function is to protect the reactor core and its coolant system pressure boundary and to assist the Engineered Safety Features Actuation System in limiting the consequences of certain accident conditions. Secondary functions are to provide equipment protection alarms and limiting signals.

During the initial stage of modernization, decision making process should be provided to evaluate the possible alternatives and select the most diverse primary (secondary) RTS, taking into considerations not only diversity values of each alternative, but the expenses required to implement it.

29.4.3 Uncertainty inherited to the task of strategy diversity assessment.

Nowadays, the uncertainties, associated with an alternative I&C diversity assessment, create a demand for the methods to make possible the translation, to a mathematical language, of the intangible values and human experience, improving the available resources in the decision making process in this complicated area.

Usually, in a quantitative setting, the information is expressed by means of numerical values. However, when we work in a qualitative setting, that is, with vague or imprecise knowledge, the information cannot be estimated with an exact numerical value. In that case, a more realistic approach may be to use linguistic assessments instead of numerical values, that is, to suppose that the variables, which participate in the problem area, are assessed by means of linguistic terms [32], [33]. This approach is appropriate for a lot of problems, since it allows a representation of the information in a more direct and adequate form if we are unable to express it with precision.

A linguistic variable differs from a numerical one in that its values are not numbers, but words or sentences in a natural or artificial language. Since words, in general, are less precise than numbers, the concept of a linguistic variable serves the purpose of providing a means of approximated characterization of phenomena, which are too complex or too ill-defined to be amenable to their description in conventional quantitative terms.

In fact, considering the approach suggested in [34], it is often difficult to determine the precise values of diversity attributes' weights and rank of all alternatives on diversity criteria. We need to evaluate all appropriate experience of applications of different diversity approaches in all industrial area, take into account all relevant statistics of I&C failures caused by CCFs etc. A part of this information is often represented as linguistic information, being the expert's subjective

opinions. The transformation and formalization of this linguistic information into precise form without application of special methods is characterized by loss of important information. This is another aspect, which increases the difficulties of I&C diversity assessment.

At the initial stage of selection of secondary (primary) RTS it is more convenient approach for the experts to compare the possible alternatives of primary (secondary) RTS and express their preferences using the natural language expressions.

The experts have to deal with portion of qualitative information stipulated by several types of the following uncertainties:

- Uncertainties caused by lack of sufficient and objective information on RTSs, which could be considered as an alternative for given RTS. The lack of required information is stipulated by policies of some I&C company-manufacturer to conceal the part of information related to its possible shortages and defects. In addition, a part of information on RTS features is confidential and not available for objective expert assessment;
- Strategic Uncertainties caused by dependencies on activities of other subjects involved (directly or indirectly) in the process of selection of alternative RTS (partners, suppliers etc.);
- Uncertainties caused by application of imprecise information (different system parameters) expressed in natural language (for example the linguistic nature of some diversity attributes).

On the one hand, it is possible to neglect all these uncertainties and use deterministic approaches for selection of the most diverse I&C system for a given one. But on the other hand, some of important information might be lost.

The following framework for application of diversity as a means for smart grid safety assurance is given below.

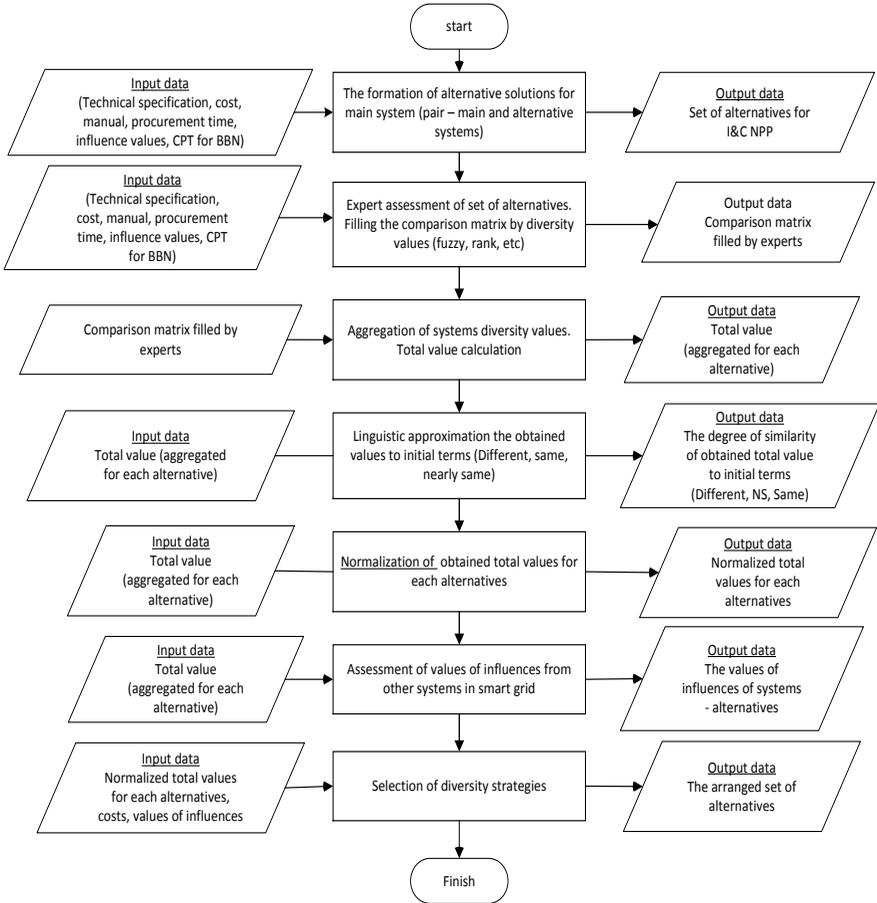


Fig. 29.4 Stages of I&C NPP diversity assessment

Each alternatives as diverse I&C NPP systems are evaluated considering such criteria as Design, Equipment Manufacturer, Equipment Manufacturer, Functional, and other similarities with a primary I&C NPP. The additional criteria to compare is vulnerability to negative influences from other systems. The less influences are the less criticalities are. More information could be found in [34].

Conclusions

The proposed approach may be applied to smart grid safety value prediction, taking into account its systems influence. The approach is based on the use of BBN with capacity to predict the possible safety change due to influences between systems in smart grid. NPP is considered as a one SG system which safety state determined the safety of smart grid as a whole. The SG safety assessment is carried out taking into consideration principles of dynamism, uncertainty and mutual influence of systems. BBN is used to predict the particular criticality of PG system, conditioned by the given type of influence.

Consideration of the difference types of influence allows improving the accuracy of SG safety value. This approach may be also applied to smart grid – NPP informational – physical influence assessment. BBN is used to predict NPP unsafe state considering the combination of informational and physical state of CSGs connected to I&C NPP. Informational state of I&C is characterized by correctness of its logic function due to state of control data.

The next step of approach development is constructing of dynamical BBN which allows evaluating time dependent mutual influences between smart grid and NPP.

Questions to self-checking

1. What are the main benefits from implementation of smart grid technologies?
2. What are the main differences between conventional grid and smart grid?
3. What is a role of NPP in smart grid?
4. Describe the main approaches to model infrastructure interaction (between NPP and smart grid)? Name the common disadvantages of various approaches.
5. Describe the principles of smart grid safety analysis?
6. Name the types of influences between smart grid and I&C NPP?
7. What is specific features of Information – physical influence between smart grid and NPP?
8. Name the features of Information – physical influence assessment approach?

9. What are the differences between informational and technical states?

10. What is nature of uncertainty inherited to the task of strategy diversity assessment?

11. What is the diversity? Describe the stage of I&C NPP diversity assessment?

References

1. Kaijser, A.: The Swedish Infrastructure – Historical Development and Future Challenges (1984)
2. Holmgren, A. and Molin, S.: Using Disturbance Data to Assess Vulnerability of Electric Smart Delivery Systems, Journal Infrastructure systems, American Society of Civil Engineers (ASCE), 243-251 (2006)
3. Bedford, T and Cooke, R. M.: Probabilistic Risk Analysis: Foundation and Methods (2001)
4. Hoyland, A. and Rausald, M.: System Reliability Theory: Models and Statistical Methods (1996)
5. Albert, R. and Barabasi, A.-L.: Statistical Mechanics of Complex Networks, Review of Modern Physics, 23-26 (2002)
6. Albert, K., Albert, I. and Nakarado, G. L.: Structural Vulnerability of North American Smart Grid. Physical review E., 69, 025193 (R) (Rapid Communication), 95-104 (2004)
7. Glass, R.: Simulation and Analysis of Cascading Failure in Critical Infrastructure, Proceeding of Working Together: R&D Partnerships in Homeland Security, 45-56 (2005)
8. Heping, P. and Lin, L.: Fuzzy Bayesian networks a general formalism for representation, inference and learning with hybrid Bayesian networks, IJPRAI, 14(7), 941–962 (2000)
9. Elyasi Komari, L., Kharchenko, V. and Babeshko, E.: Extended Dependability Analysis of Information and Control Systems by FME(C)A-Technique: Models, Procedures, Application, in Proceedings of the Conference on Dependability of Computer systems, 25-32 (2009)
10. Brezhnev, E.: Risk-analysis in critical information control system based on computing with words' model, in Proceeding of 7th

International Workshop on Digital Technologies, Circuit Systems and Signal Processing, 67-72 (2010)

11. Bowles, J. B.: An assessment of PRN prioritization in a failure modes effects and criticality analysis, Volume 47, N: 51-60 (2004)

12. On-line Tutorial on Bayesian nets an dprobability. Available at:<http://www.dcs.qmw.ac.uk/%7Enorman/BBNs/BBNs.htm> [retrieved: February, 2013].

13. L. Zadeh and J. Kacprzyk, Computing with words in Information / Intelligent Systems – Part : Foundation; Part 2, USA, 2001.

14. Causes of destruction of Sayano–Shushenskaya HPP hydraulic unit 2. Available at:

http://zhurnal.lib.ru/b/boris_i_k/prichinyrazruschenijagidroagregata2ssc_hges.shtml [retrieved: December, 2012]

15. European Commission European Smart Grids Technology Platform: Vision and Strategy for Europe’s Electricity, Available at: http://ec.europa.eu/research/energy/pdf/smartgrids_en.pdf.

16. Hidaka, K.:Smart Grid as New Big Opportunity of Information and Communication Technology. Science & Technology Trends. Quarterly Review, No.38, 23-39 (2007)

17. McDaniel, P and McLaughlin, S.: Security and Privacy Challenges in the Smart Grid, IEEE Security Privacy Magazine, vol. 7, no.3,75 -77 (2009)

18. Dudenhoefter, D.D., Permman M.R. & Manic, M. CIMS: A framework for infrastructure interdependency modelling and analysis.(2006)

19. Etola, R., De Porcellinis, S. and Sforna, M. Critical infrastructure dependency assessment using the input-output inoperability model. International Journal of Critical Infrastructure Protection 2, 170-178 (2006)

20. Min, X. & Duenas-Osorio, L.: Inverse Reliability-based Design of Interdependent Lifeline Systems.TCLEE Lifeline Earthquake Engineering in a Multihazard Environment.(2009)

21. Svendsen, N.K. and Wolthusen, S.D. Connectivity models of interdependency in mixed-type critical infrastructure networks. Information Security Technical Report, 12, 44-55 (2006)

22. Setola, R., De Porcellinis, S. & Sforna, M. Critical infrastructure dependency assessment using the input-output

inoperability model. *International Journal of Critical Infrastructure Protection* 2, 170-178 (2009)

23. Rinaldi, S., J. Peerenboom, and T. Kelly: Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies, *IEEE Control Systems Magazine*, IEEE, 11-25 (2009)

24. Kharchenko V., Siora A., Sklyar V., Volkoviy A. Multi-Diversity Versus Common Cause Failures: FPGA-Based Multi-Version NPP I&C Systems, *Proceedings of the Conference NPIC&HMIT, Las-Vegas, Nevada, USA, 85 – 96, 2010.*

25. Volkoviy A., Lysenko I., Kharchenko V., Shurygin O. Multi-Version Systems and Technologies for Critical Applications, *National Aerospace University KhAI, Kharkiv, Ukraine, 34 – 41, 2009.*

26. Kharchenko V., Sklyar V. (eds) *FPGA-based NPP Instrumentation and Control Systems: Development and Safety Assessment*, RPC Radiy, National Aerospace University KhAI, State STC on Nuclear and Radiation Safety, Kirovograd - Kharkiv, 2008.

27. Kharchenko V., Bakhmach E., Siora A., Duzhiy V., Volkoviy A. Assessment of Multi-Version NPP I&C Systems Safety: Metric-Based Approach, *Technique and Tool. Proceedings of the Conference ICONE, Osaka, Japan, 143 – 154, 2011.*

28. Siora A., Sklyar V., Kharchenko V. (n,m)-Version Systems: Taxonomy, Models and Technologies. *Bulletin of Kharkiv National University: Mathematical Modelling. Information Technology. Automated Control Systems, No 4, 34-47, 2008.*

29. Elyasi Komari, V. Kharchenko, and E. Babeshko, Extended Dependability Analysis of Information and Control Systems by FME(C)A-Technique: Models, Procedures, Application, in *Proceedings of the Conference on Dependability of Computer systems, Brunow, Poland, 30 June-02 July 2009, IEEE Computer Society, Los Alamitos, California, pp. 25-32, 2009.*

30. J. B. Bowles, “An assessment of PRN prioritization in a failure modes effects and criticality analysis”, *Journal of the IEST, Volume 47, N: 51-6. 2004.*

31. Zadeh L. From computing with numbers to computing with words—from manipulation of measurements to manipulation of perceptions. *IEEE Trans.Circ.Syst, Fund.TheoryAppl. Vol. 4, No.1, 105 –119, 1999.*

32. Mendel J.M. An architecture of making judgment using computing with words. Int. J.Appl. Math. Comput. Sci., Vol.12, No.3, 325–335, 2002.
33. NUREG/CR-7007. Diversity Strategies for Nuclear Power Plant Instrumentation and Control Systems, United States Nuclear Regulatory Commission, 2009.
34. Eugene Brezhnev, Vyacheslav Kharchenko etc The Cost-Effective approach To selection of Diverse NPP RTS under uncertainty /11 th International Probabilistic Safety Assessment and Management Conference & The Annual European Safety and Reliability Conference, Helsinki, Finland, 20-29 June, 2012

Summary

This chapter suggests the approach for interaction assessment between smart grid substation as the main element of smart grid and NPP. This approach takes into consideration an influence of CSGS security level on NPP safety level, suggests a concept of diversity as a means for smart grid safety assurance.

30 Smart Grid Security And Resilience Analysis And Assurance

30.1 Introduction to smart grid security and resilience

As mentioned in chapter 29, critical infrastructure play a high role in normal operation of modern society. Critical infrastructure operate in all-hazards environment that poses many risk on infrastructure's safety and security. Smart grid is very complicated infrastructure with wide application of information and communication technologies. As mentioned in [1] infrastructure resilience is about "delivering the goods" regardless of disruptive events that may occur. Smart grid resilience is the ability to reduce the magnitude and/or duration of disruptive events (natural or human-centered) and return to normal operation.

The effectiveness of a resilient smart grid depends upon its ability to anticipate, absorb, adapt to, and/or rapidly recover from a potentially disruptive event. All of these features

Absorptive capacity is the ability of the smart grid to endure a disruption without significant deviation from normal operating performance.

Cyber diversity is suggested as one of the general principles of improvement of absorbing and adaptive ability of smart grid as a whole by decreasing a number of common cause failures of smart substations with a critical load, such as Nuclear Power Plant (NPP).

The new metrics of smart substations diversity assessment are introduced in this chapter. It allows estimating the diversity required to decrease risks of CCFs including cyber risks and improve the smart grid resilience.

Nuclear power occupies a unique position in the debate over global climate change as the only carbon-free energy source. Nowadays it is already contributing to world energy supplies on a large scale, and has potential to be expanded if the challenges of safety, nonproliferation, waste management, and economic competitiveness are addressed, and technologically fully mature. So it might be concluded that Nuclear Power Plants (NPP) is an intrinsic part of future smart grid.

The substations which provide links between NPP and PG are extremely strategic to NPP safety. Compared to other systems in an electric utility network, the smart substation has the highest density of valuable information needed to operate and manage a smart grid. Unreliable substation equipment and insufficient cyber security introduce new risks to NPP safety. A successful attack on one of these substations could have fatal and expensive consequences.

Nowadays there are no differences among substations in respect to cyber security. Substation cyber security issues are important in the respect to NPP safety. Smart substations main assets are not only physical facilities, but also information, databases and software applications, different intelligent electronic devices (IEDs) such as breaker controllers, voltage regulators, remote terminal units (RTUs), programmable logic controllers (PLCs). These IEDs are important cyber assets of digital substation.

Substation state of operability could be compromised by IEDs common cause failures (CCFs) which could occur at any substation levels and introduce new risks to NPP safety. Hardware CCFs are failures (or unavailable states) of substation equipment due to a shared cause. For example, NPP I&C's failures' analysis proves that CCFs are significant contributors to I&C incidents. For example, 450 failures (out of 3000) fall on multiple failures during 564 reactor-years [2].

According to [3] Industrial Control Systems (ICS) have the common cyber vulnerabilities. These vulnerabilities are divided on three general categories such as: the vulnerabilities inherent in the ICS product, vulnerabilities caused during the installation, configuration, and maintenance of the ICS and the lack of adequate protection because of poor network design or configuration. For example, through bad coding practices and improper input validation, access can be granted to an attacker allowing them to have unintended functionality or privilege escalation on the systems. Examples of improper input validation identified are within buffer overflows, boundary checking, and code injection. It means that besides hardware CCFs digital substations' IEDs might be prone to cyber common cause failures (CCCFs). Cyber CCFs might be determined as events when cyber assets' availability, confidentiality and integrity are compromised within a specified (short) time interval. The reasons are the common vulnerabilities, tough

coupling within networks between IEDs which might lead to security violation due to human errors, shared input data equipment, environmental events (flooding, storm) and cyber attacks. Thus substations with critical loads, such as NPPs, should be given the highest level of importance in respect of their cyber security. The higher level of cyber security of smart substation with critical load might be achieved through implementation of substations' variety when IEDs with similar functionalities are different and less vulnerable to the same shared cause.

The document [4] has performed the common vulnerabilities assessments on a large variety of systems, and for each assessment, it tailors the assessment and methodology to provide the most value to the customer. In this document all vulnerability identification activities are focused on enabling the identification and remediation of the highest risk ICS cybersecurity vulnerabilities rather than the collection of data for statistical purposes. It also gives the recommendations for ICS vendors and owners on how to reduce the common vulnerabilities of ICS systems. The list of recommendations includes the following: create a Security Culture, Enhance ICS Test Suites, Create and Test Patches, Redesign Network Protocols for Security, etc. It might be unfeasible to implement the joint plan for common vulnerabilities reduction considering such business issues as competition among vendors, lack of coordination, etc.

In [5] the authors consider the application of "defense in depth" for cyber security assurance of electrical distribution systems. "Defense in depth" is a strategy of integrating technology, people, and operations capabilities to establish variable barriers across multiple layers of an organization. These barriers include electronic countermeasures such as firewalls, intrusion detection software/components, and antivirus software, coupled with physical protection policies and training. The cost of implementation of such strategy is not considered in this paper.

In [6, 7] the common vulnerabilities of control systems are also considered. The papers describes the generalized trends in vulnerabilities observed from the assessments, as well as typical reasons for these security issues and the introduction to an effective mitigation strategy. Many of these vulnerabilities result [8, 9] from deficient or nonexistent security governance and administration, as well

as budgetary pressure and employee attrition in system automation. It is also mentioned that defense-in-depth concept should be used to cyber security assurance of cyber components.

It might be noted that the application of smart grid substation variety (diversity) is not considered as a means to decrease CCFs of IEDs including cyber CCFs. There is also no analysis of impact of substation diversity on probability of event when these substations with critical load are failed due to one shared cause. This event is determined as substations CCFs when all of them got unavailable within one short time interval.

30.2 The stages of diversity assessment of smart substation with a critical load

A diversity is one of the general principles used to decrease hardware vulnerability against CCF and provide dependability of I&C. Diversity is the general approach used for decreasing CCF risks of NPP I&C systems. Differences in equipment, development and verification technologies, implemented functions, etc. can mitigate the potential for common faults.

Typically, there are three substations that provide the power supply for NPP. It is presumed in the future that NPP will be connected to PG through the same amount of smart substations. It is important to consider all possible risks which might occur within this interaction, analyze them and mitigate as well.

There are many risks factors for smart substation with critical load. The list of these risk factors includes the following: human failures (on different substation level), hardware failures, software failures, cyber security issues, external events (Fig.30.1).

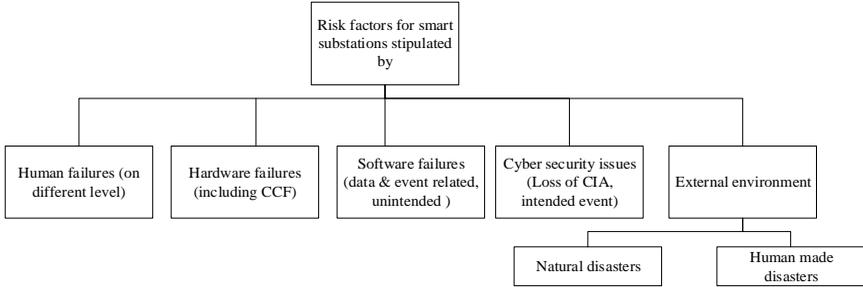


Fig. 30.1 The risks factors for smart grid substation

Generally, the logical structure of the smart substation includes: the process level where the digital signal acquisition, consolidation, processing operations are performed; interval layer measurement and control; station control layer that achieves communication within substation and control system as well as coordination with the substation operational function and the station-level support function based on information sharing. Many of these functions are performed by IEDs.

Due to high level of coupling and interconnections between IEDs they might be prone to hardware and software failures.

Hardware CCFs are subset of dependent failures in which two or more IEDs fault states exist at the same time, or within a short time interval, as a result of a shared cause (root-cause). IED software failure is considered as an inability of its program to continue signal acquisition, consolidation, processing due to erroneous logic (usually systematic failures). There are many techniques to provide the software reliability. Among them are the software fault-tolerance techniques such as software redundancy. Software redundancy is achieved by incorporating some additional software components that are not exactly identical but they are similar in functionality. *Common cause software failures* are subset of software failures in which two or more IEDs fault states exist at the same time, or within a short time interval, as a result of a shared cause (software failure as a root-cause).

Cyber CCF might be determined as an event when IEDs' availability, their data confidentiality and integrity are compromised within a specified (short) time interval. If an intruder gains the access to

substation IEDs then the possible consequences might include: shut down of the substation or any portion of the subsystem controlled by the compromised IEDs; change IEDs settings to degrade their reliability and, subsequently, the power supply service provided by the substation for particular NPP; gather control and protection settings information that could be used in a subsequent attack on other similar substation; plant malicious code that could later trigger a delayed or coordinated attack, etc.

All of these events put the new risks to NPP connected to this smart substation. NPP and power utilities' owners should cooperate with aim to reduce all possible risks caused by loss of external (for NPP) power supply. It means they should make the joint decision on selection of smart substations with the high level of diversity. The cost issues are to be considered as well. In this case the diversity is taken as CCFs mitigation techniques. Considering the IEDs high importance it is presumed that IEDs are significant contributor to substation vulnerabilities and substation cyber security might be achieved by implementation of diverse IEDs within different substations.

The approach suggested for selection of smart substations with diverse IEDs deals with qualitative aspects represented in qualitative terms by means of linguistic variables. Computing with words (CW) has been applied as a computational basis to linguistic decision making of complex situations [10].

To select the most diverse smart grid substations, using the diversity criteria and evaluate the similarity (difference) between IEDs, expert should take into consideration the compelling evidence. Based on these evidences experts evaluate the difference (similarity) between similar IEDs (from different substations) using the linguistic terms: SAME (S), NEARLY SAME (NS), DIFFERENT (D).

30.2.1 The formation of diversity strategies set

The following diversity strategies of IEDs smart substations implementation are considered in this chapter:

- Strategy S_{11} - All smart substations and their IEDs are similar. One vendor develops and produce all substations with no difference in IEDs design, manufacturing, cyber issues, etc.;

- S_{21} - All substations are different and produced by different vendors;
- S_{31} - All substations' IEDs are produced by one vendor but there are some differences between them.

30.2.2 The diversity strategies set's expertise

During this stage experts are supposed to fill the comparison matrixes to evaluate the similarities (differences) between the IEDs in term of hardware and cyber aspects. The expert is supposed to compare the IEDs with similar functionalities from different substations and select the most different between them.

If the particular IED for the first substation is determined then it is required to compare it with the possible alternatives for IEDs with similar functionalities from second and third substation. If the substation automation controller, for example, OM600, the grid automation controller of ABB, is selected for the first substation, according to S_2 strategy, this IED is compared with C264 from Alstom Grid (IED1), GE's D25 from General Electric (IED2) and SICAM (IED3) AK from Siemens. The expert is required to assign the weight of each criterion. The criterion's weight might be expressed either as linguistic value (Low, Medium, High) or any numerical values from [0, 1]. For sake of simplicity the weight of criterion is presented as a scalar value.

Table 30.1 represents the example of diversity assessment for the strategy S_{21} (hardware aspects).

Table 30.1 Checklist for IED CCF (hardware vulnerabilities aspect)

Vulnerabilities criterion	W_k , weight of criterion	IEDs		
		<i>IED1</i>	<i>IED2</i>	<i>IED3</i>
Design				
System Layout/Configuration	0,2	NS	D	NS
Component Internal Parts	0,23	NS	D	D

Vulnerabilities criterion	W_k , weight of criterion	IEDs		
		<i>IED1</i>	<i>IED2</i>	<i>IED3</i>
Design team	0,13	D	D	D
Design procedures	0,24	NS	D	NS
V&V procedures	0,2	NS	D	NS
Manufacturing				
Manufacturing method, and material	0,13	NS	NS	NS
The manufacturing staff	0,27	D	D	D
The same quality control procedure	0,6	NS	NS	NS
Installation				
Installation method, and material	0,33	NS	NS	NS
The Installation staff	0,41	D	D	D
The quality control procedure	0,26	D	D	NS
Operation				
Operation method, and material	0,4	S	NS	S
The Operation staff	0,32	D	D	D
The quality control procedure	0,28	NS	NS	D
Maintenance				
Maintenance/ Test/Calibration Schedule	0,21	NS	D	NS
Maintenance/ Test/Calibration Procedure	0,31	NS	D	NS
Maintenance/Test/Calibration Staff	0,48	D	D	D

Table30.2 represents the example of diversity assessment for the set of strategies S_{21} (cyber aspect).

Table 30.2 Checklist for IEDs CCF (Cyber vulnerabilities aspect)

Vulnerabilities	W_k ,	IEDs
-----------------	---------	------

critterion	weight of critterion	<i>IED1</i>	<i>IED2</i>	<i>IED3</i>
Design				
The coding practices	0,12	NS	D	D
The security requirements	0,21	NS	D	D
The security testing procedure	0,13	NS	NS	NS
The vendor	0,15	D	D	D
The tools used	0,19	S	S	NS
The security culture	0,2	NS	S	D
Installation				
The installation procedure	0,43	D	D	D
The installation team	0,35	D	D	D
The installation tool	0,22	NS	NS	NS
Operation				
The communication links	0,51	S	NS	NS
The port security on network equipment	0,49	NS	D	NS
Configuration				
Patch management procedure	0,22	NS	NS	D
Encryption procedure	0,13	D	NS	NS
Authentication procedure	0,65	D	D	D

The expert is proposed to use linguistic values to evaluate all possible IEDs' alternatives for substations.

In this chapter, we shall use labels represented by triangular fuzzy numbers. A triangular fuzzy number, denoted by $M = \langle m, \alpha, \beta \rangle$, has the membership function:

$$\mu_M(x) = \begin{cases} 0, & \text{for } x \leq m \leq m - \alpha \\ 1 - \frac{m-x}{\alpha}, & \text{for } m - \alpha < x < m \\ 1, & \text{for } x = m \\ 0, & \text{for } x \geq m + \beta. \end{cases} \quad (30.1)$$

The point m , with membership grade 1, is called the mean value and α , β are the left hand and right hand spread of M respectively.

For example, we assign the following semantics to the set of three terms:

$$\text{NS} = (0, 0,25, 0,5), \text{S} = (0,25, 0,5, 0,75), \text{D} = (0,5, 0,75, 1).$$

C. The aggregation stage

During this stage all linguistic values provided by experts are aggregated to obtain a collective assessment for the IED's alternatives. It is provided by calculation of the fuzzy diversity score D_{ij} as an arithmetic mean:

$$D_{ij} = \left(\frac{1}{t} \sum_{k=1}^t w_k \times m_{ij}^t, \frac{1}{t} \sum_{k=1}^t w_k \times \alpha_{ij}^t, \frac{1}{t} \sum_{k=1}^t w_k \times \beta_{ij}^t \right), \quad (30.2)$$

where w_k – weight of k criterion; $\langle m_{ij}^t, \alpha_{ij}^t, \beta_{ij}^t \rangle$ – a triangular fuzzy number that represents one of linguistic values {S, NS, D} assigned by t th expert for S_{ij} diversity strategy. D_{ij} represents a difference between two IEDs. The more value D_{ij} , which corresponds certain diversity strategy S_{ij} , the more diverse both IEDs.

Using the best-fit method [10], the obtained fuzzy diversity score D_{ij} for each IEDs can be mapped back to one (or all) of the defined linguistic terms (SAME, NEARLY SAME, DIFFERENT). The method uses the distance between fuzzy diversity score, represented by fuzzy triangular number for each IEDs and each of the initial linguistic terms to represent the degree to which obtained score, is confirmed to each of them. The distance between the obtained fuzzy diversity score D_{ij}

and the expression SAME, NEARLY SAME, DIFFERENT is defined as follows:

$$d_{ij}^{(r)}(D_{ij}, SAME) = \left[\sum_{j=1}^3 (\mu_{D_{ij}}^j - \mu_{same}^j)^2 \right]^{\frac{1}{2}};$$

$$d_{ij}^{(r)}(D_{ij}, NEARLY\ SAME) = \left[\sum_{j=1}^3 (\mu_{D_{ij}}^j - \mu_{NS}^j)^2 \right]^{\frac{1}{2}}; \quad (30.3)$$

$$d_{ij}^{(r)}(D_{ij}, DIFFERENT) = \left[\sum_{j=1}^3 (\mu_{D_{ij}}^j - \mu_{different}^j)^2 \right]^{\frac{1}{2}}.$$

Hence, each IED is characterized by 3-tuple $\langle d_{ij}^{(1)}, d_{ij}^{(2)}, d_{ij}^{(3)} \rangle$, where $d_{ij}^{(r)}$ - a distance between obtained fuzzy diversity score and corresponding linguistic term (SAME, NEARLY SAME, DIFFERENT).

It should be noted that each $d_{ij}^{(r)}$ ($j = 1, \dots, J$, where j – number of possible alternatives classified as type of S_i strategy) is an unsealed distance. The closer D_{ij} , is to the r th expression, the smaller $d_{ij}^{(r)}$ is. More specifically, $d_{ij}^{(r)}$ is equal to zero if D_{ij} , is just the same as the r th expression in terms of the membership functions. In such a case, D_{ij} should not be evaluated to other expressions at all due to the exclusiveness of these expressions. To embody such features, new indices need to be defined based on $d_{ij}^{(r)}$ ($r = 1, 2, 3$).

Suppose $d_{ij}^{(3)}$ is the smallest among the obtained distances for D_{ij} , and let α_{i1} , α_{i2} , α_{i3} represent the reciprocals of the relative distances between the identified fuzzy diversity score D_{ij} , and each of the defined linguistic terms with reference to $d_{ij}^{(3)}$ (smallest distance). Then, $\alpha_{ij}^{(r)}$ ($r = 1, 2, 3$) can be defined as follow:

$$\alpha_{ij}^{(r)} = \frac{1}{\frac{d_{ij}^{(r)}}{d_{ij}^{(3)}}}, r = 1, 2, 3. \quad (30.4)$$

If $d_{ij}^{(3)} = 0$ it follows that $\alpha_{ij}^{(3)}$ is equal to 1 and the others are equal to 0. Then, $\alpha_{ij}^{(r)}$ ($r = 1, 2, 3$) can be normalized by:

$$\beta_{ij}^{(r)} = \frac{\alpha_{ij}^{(r)}}{\sum_{r=1}^3 \alpha_{ij}^{(r)}}, r = 1, 2, 3. \quad (30.5)$$

Each $\beta_{ij}^{(r)}$ represents the extent to which D_{ij} belongs to the r th defined linguistic terms. Thus, $\beta_{ij}^{(r)}$ could be viewed as a degree of confidence that obtained fuzzy scores for all diversity strategies S_{ij} belong to the r th defined linguistic terms.

Results obtained for the selection of the most diverse substation controller to decrease the cyber vulnerability of smart substation with critical load are presented in the table 30.3. The IED1 is the most diverse IED to OM600 in respect to cyber vulnerabilities.

Table 30.3 Results obtained for all IEDs considered in example.

IED alternatives	Degree to which D_{ij} belongs to the initial terms		
	S	NS	D
<i>IED1</i>	0,12	0,39	0,49
<i>IED2</i>	0,36	0,28	0,38
<i>IED3</i>	0,33	0,63	0,04

30.2.3 The exploitation stage

During this stage all IEDs' alternatives are ranked by using the collective linguistic assessment obtained in the previous stage, taking into account the cost of each IED, C_{ij} . The rational diverse strategy could be found with the following criterion:

$$S_{ij}^* = argmax \frac{\beta_{ij}^{(r)}}{c_{ij}^*}, \quad (30.6)$$

where $\beta_{ij}^{(r)}$ represents the extent to which D_{ij} belongs to the r th defined linguistic terms; C_{ij}^* - cost of S_{ij} reduced to $\sum C_{ij}$, ij – number of alternatives.

The main aim of all stages described above is decrease the IDEs cyber common vulnerabilities of substations with critical load. The more diversity of particular IED the more diversity is achieved among this type of cyber security assets. All smart substation cyber assets should be evaluated in the same way to provide the highest level of cyber security. The result of this approach is the set of diverse smart substation with critical load.

30.3 BBN as a basis for Cyber Common Cause Failure assessment of substations

The connections between NPP and smart substations is represented as BBN with nodes (NPP reactor, safety systems and substations) and edges as the power lines. BBNs are very effective for modeling situations where some information is already known and incoming data is uncertain or partially unavailable. BBN that represents links between reactor unit, its safety systems (RPS, RCIC) and on site, off site power supply is given on Fig. 30.3. Construction and assessment of BBN parameters was performed using Netica 5.12 tool. The Netica APIs are a family of powerful Bayesian Network toolkits.

This BBN allows evaluating CCFs of substations that provide the power supply to NPP. When successful cyber attack on smart substations IDEs is performed then if there is no any substation diversity (all substations IEDs are the same) then probability of CCFs of all substations is 0,729. All substations are prone to the same threats and have the common cyber vulnerabilities.. This scenario describes the situation when the attacker (terroristic organization) performs the successful attempt to compromise the IEDs cyber security and as a result to make the substation to be not operable. It seems to be realistic while all substations are similar.

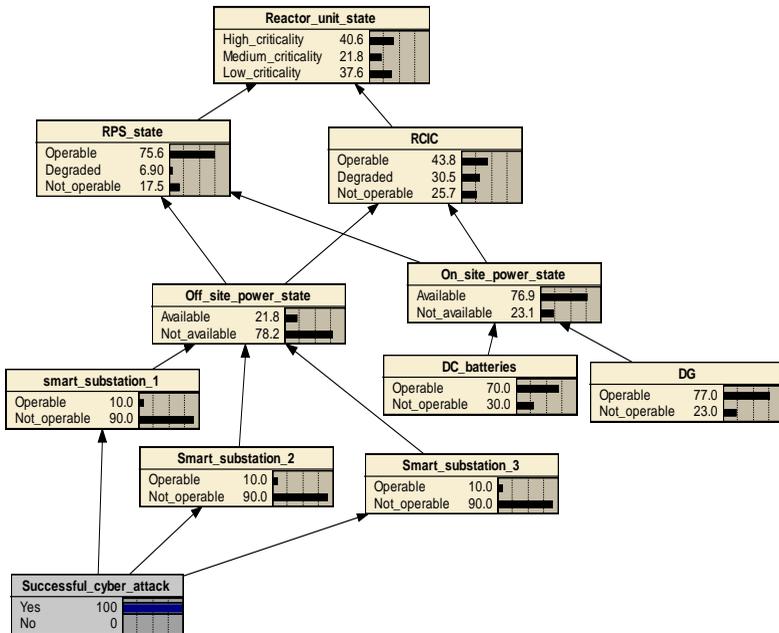


Fig. 30.3 BBN without cyber diversity implementation

We have selected the most different IEDs considering the approaches given above.

BBN presented below is made on parameters that consider diversity in substations. It might be seen that the probability of CCFs is decreased to 0,126.

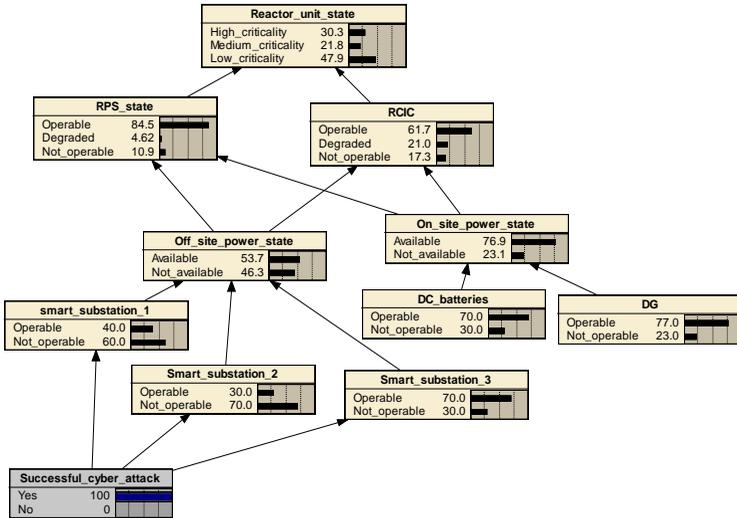


Fig. 30.4 BBN with cyber diversity implementation

Conclusions

The safe operation of NPP requires that smart substation operates in secure manner. In the future if not being treated now the cyber risk of smart substation can compromise NPP safety. To assure the security of smart substation the cyber diversity is suggested in this chapter. Cyber diversity is suggested as one of the general principles of improvement of absorbing and adaptive ability of smart grid as a whole by decreasing a number of common cause failures of smart substations with a critical load, such as Nuclear Power Plant (NPP). All substations with critical load should be selected with diversity in mind. The approach for diversity assessment of such substations based on processing of linguistic values given by experts. Each IED is characterized by fuzzy diversity score of its similarity (difference) with IED that has been already selected. The cost of IED is also taken into consideration. This approach might be useful during the initial stage of substation modernization to assure the required level of cyber security and resilience. BBN is used to evaluate CCF of substations with critical load before and after diversity implementation.

Questions to self-checking

1. Why is smart grid substation important for NPP safety?
2. What is smart grid resilience and how this is linked to cyber security?
3. What is cyber diversity and how it can improve smart grid resilience?
4. What are main features of industrial control systems and how they abnormal operational states can influence smart grid safety and security? What are intellectual digital devices?
5. Name the risks factors for smart grid substation?
6. What are common cause failures? What is a difference between hardware and software CCF?
7. Describe the vulnerabilities criterion used for cyber diversity assessment? Why are they inherited to IED? How to decrease them during system development life cycle?
8. Name the stages of cyber diversity assessment?
9. What are industrial control system's development stages? How security shall be treated on each stages?
10. What are main features of Bayesian networks? How this network is used for CCFs risk assessment?

References

1. Critical infrastructure resilience. Final report and recommendations. National Infrastructure Advisory Council (NIAC), 2009, 54 p.
2. Iloh, J.P.I., Mbachu, C.B., Uzhede, G.O.: An improved merging unit model for substation automation system based on IEC61850. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 13054-13063 (2014)
3. Lisnianski, A., Levitin, G.: Multi-state system reliability: assessment, optimization and applications. World scientific Publishing Co. Pte Ltd, Singapore (2003)
4. Xing, L., Levitin, G.: Combinatorial algorithm for reliability analysis of multistate systems with propagated failures and failure

isolation effect. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 41(6), 1156-1165 (2011)

5. Risk management: a tool for improving Nuclear Power Plant performance, IAEA VIENNA, IAEA-TECDOC-1209, 2001.

6. NISTB Assessments Summary Report: Common Industrial Control System Cyber Security Weaknesses, Idaho National Laboratory Idaho Falls, Idaho 83415, 2010.

7. Common Cybersecurity Vulnerabilities in Industrial Control Systems, Home Land Security, Control Systems Security program, National Cyber security Division, 2011.

8. Max Wandera, Brent Jonasson, Cybersecurity considerations for electrical distribution systems. White Paper WP152002EN, 2014.

9. Common vulnerabilities in critical infrastructure control systems, Sandia National Laboratories Albuquerque, NM 87185-0785, 2nd edition, 2003.

10. Zadeh L. and Kacprzyk J. Computing with Words in Information/Intelligent Systems – Part 1: Foundation; Part 2: Applications. Heidelberg, Germany: Physica-Verlag, vol.1, 187 – 201, 19991

CM3. HUMAN-MACHINE ENGINEERING FOR SECURITY CRITICAL AND RESILIENT SYSTEMS

31.1 Classification of human-machine engineering for resilient systems. Terms and concepts.

Human-machine engineering for resilient systems is a form of engineering based on improving and application of quality assurance model for interaction between a human and the critically used systems.

Human-system interfaces (HSIs) - a human-system interface is that part of the system through which personnel interact to perform their functions and tasks.

Human factors - a body of scientific facts about human characteristics. The term covers all biomedical, psychological, and psychosocial considerations; it includes, but is not limited to, principles and applications in the areas of human factors engineering, personnel selection, training, job performance aids, and human performance evaluation.

Human factors engineering (HFE) - the application of knowledge about human capabilities and limitations to plant, system, and equipment design. HFE provides reasonable assurance that the design of the plant, systems, equipment, human tasks, and the work environment are compatible with the sensory, perceptual, cognitive, and physical attributes of the personnel who operate, maintain, and support the plant.

User-centered design (UCD) - processes a multi-disciplinary activity, which incorporates human factors and ergonomics knowledge and techniques with the objective of enhancing effectiveness and productivity, improving human working conditions, and counteracting the possible adverse effects of use on human health, safety and performance.

Safety Case - approach to ensure the functional safety.

Quality - the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs.

External quality - the extent to which a product satisfies stated and implied needs when used under specified conditions.

Quality model - the set of characteristics and the relationships between them, which provide the basis for specifying quality requirements and evaluating quality.

Quality in use metrics measure the extent to which a product meets the needs of specified users to achieve specified goals with effectiveness, productivity, safety and satisfaction in a specified context of use. Quality in use is assessed by observing representative users carrying out representative

tasks in a realistic context of use. The measures may be obtained by simulating a realistic working environment (for instance in a usability laboratory) or by observing operational use of the product. When measuring quality in use it is important that users are only given the type of help and assistance that would be available to them in the operational environment.

Usability – the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

Accessibility - usability of a product, service, environment or facility by people with the widest range of capabilities.

Effectiveness metrics assess whether users can achieve specified goals with accuracy and completeness in a specified context of use.

Efficiency relates the level of effectiveness achieved to the quantity of resources expended. Efficiency is generally assessed by the mean time taken to achieve the task. Efficiency may also relate to other resources (e.g. total cost of usage). A common measure of efficiency is time on task.

Productivity metrics assess the resources that users consume in relation to the effectiveness achieved in a specified context of use. The most common resource is time to complete the task, although other relevant resources could include the user's effort, materials or the financial cost of usage.

Satisfaction metrics assess the user's attitudes towards the use of the product in a specified context of use.

Safety metrics assess the level of risk of harm to people, business, software, property or the environment in a specified context of use. It includes the health and safety of the both the user and those affected by use, as well as unintended physical or economic consequences.

Compliance - each characteristic should comply with the requirements of the corresponding standard for each software class.

Human-machine interface security – capability of the HMI to protect information so that unauthorized persons or processes are not able to read or modify it but authorized users and processes are provided with the access to it. This requirement refers to the transmitted data as well.

Usability of cybersecurity – a well-designed system needs to make it easy for the user to do the right thing, hard to do the wrong thing, and easy to recover when the wrong things happen anyway.

User-centered security – the innovative approach requiring to take into account the factors of perception, characteristics, needs, abilities and behaviour of users when developing cyber security measures.

Metric - a measurement scale and the method used for measurement. Metrics includes methods for categorizing qualitative data.

31.2 Problems and trends of human-machine engineering

Human-machine interfaces (HMI) are an important part of information and control systems (I&Cs) for commercial and critical domains. The ability to control and manage systems and objects, for instance reactor, aircraft or medical equipment, the efficiency and reliability of the human and I&Cs as a whole, depend on the HMI quality.

The HMI development process is based on the variety of modern technologies and means, such as sensor monitors, high-performance computers and networks.

Scalable and flexible interfaces of the operator's panels allow integrating into the HMI different systems monitoring and control functions, for instance the smart-house control panels [1].

The hardware part of such HMI includes the touch-screen, that gives an access to the general infrastructure of automated house and Internet.

The software part is presented by the smart-house control program, which communicates and operates with a variety of connected devices, such as an air conditioner, CCTV, intercom, lighting equipment, household devices, etc. HMI can also offer some other Internet services, for example food purchase, bills payment and etc.

HMI for the automobile informational systems improves the traffic safety by decreasing the driver's informational overload and thus minimizing the distractions. The context-dependended interface adaptation is proposed in [2], which can be achieved by user personalization.

To guarantee the safety of nuclear power plants (NPP) it is required the modernization and development of new instrumentation and control systems. I&Cs functionality, reliability and effectiveness of human activities depend heavily on human-machine interfaces [3,4].

In [4] authors considered the HMI development issues in case of supporting the operator cognitive activities in the field of nuclear power industry.

One of the nowadays challenges in the HMI creation domain is a development of green human-machine interfaces (GHMI). In contrast to the traditional HMIs, they have such properties as environmental friendliness, adaptability, safety, reliability, etc [5].

Environmental interface is the interface, where information, provided by the mental model, closely matches the managed object, operated by human. The environmental interface provides the information in a form of visual images, which makes its perception and execution easier for processing of cognitive operations. The modern environmental interface is created using the virtual reality technologies (immersion interface) and cognitive graphics.

Based on the environmental interface definition, we can conclude that environmental interface should be matched with the user's mental model, make the perception of the information easier and facilitate the implementation of the cognitive operations of the operator.

Here is what is relevant now in the HMI field: human factor studies in order to reduce the likelihood of errors; analysis of the reliability of operator's actions associated with the risk assessment and taking into account the possible consequences; development of techniques for evaluation of safety [6].

Much attention is given to the issues of the human factor and HMI in the transport systems [7, 8]. HMI will be one of the major topics to which investigations in the field of transport safety are going to be devoted in the nearest future, as marked in [8].

Systems that need to communicate to the driver must be easy to use and always keep the driver doing the principal task that consists in driving a vehicle safely. Good HMI system reduces the informational strain on the driver helping to select the most relevant and important information. Therefore, all the risks associated with the use of such systems should be estimated.

The approach for quality and safety HMI assessment, based on Safety Case methodology is proposed in [9]. According to this methodology, the HMI assessment has to be performed during all stages of the life cycle, and results must be grounded and documented.

The project PRORETA is a research in the area of the HMIs. The research object is the prototype of the cooperative automobile HMI. The PRORETA HMI system implements a huge number of use scenarios, it does not complicate or irritate and ensures the multimode support [10].

One of the variants of cooperative HMI construction - is to use the technology of cloud computing [11]. Cooperative HMI provides the measure values of the parameters of vehicle and driver state in real time via the Internet into the "cloud." Here, the data from all the cars is dynamically processed and transmitted to motoring public.

31.3 Example of human-machine interface of critical and resilient system

Modern I&Cs of the NPP are complex systems of the distributed information processing, where HMI implementation is usually based on workstations. The main purpose of these HMI is to provide staff with the information on the status of the power unit systems, as well as an interface to control the actuators. Information is provided on the monitors of the Main Control Room and workstations for personnel.

The main component of displaying the details about information and control systems is video frames (VF) organized as a number of systems with the multilevel hierarchy and the capability to transfer both from one level of the hierarchy to another, inside the levels and between the systems. In addition, video frames can be called from the menu or from the function keyboard.

VF provide the operator with the technological information in real time in form of mnemonic diagrams (animated fragments of technological schemes or images of technological equipment), diagrams, histograms, tables, charts and so on (fig. 31.1).

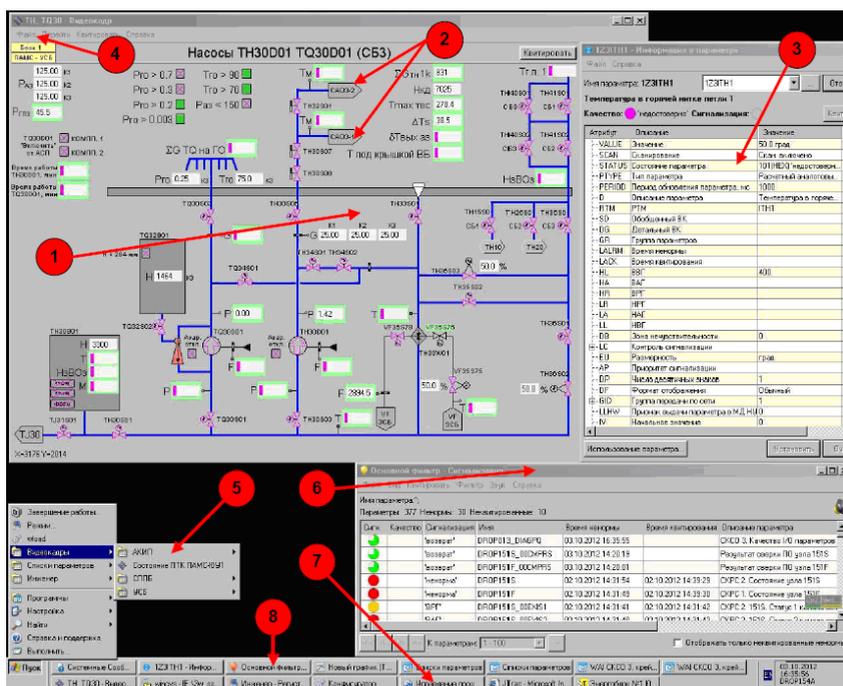


Figure 31.1 – Structure of video frame

1. Video frame window.
2. Video frame elements to display the technical nodes of the system.
3. Nodes parameters information.
4. Main menu of video frame.

5. Main menu of operating system.
6. Windows of additional systems and the signalization system.
7. Task panel of operating system.
8. Window selection button.

31.4 Human-machine interface model

Figure 31.2 presents a model of the human-machine system. Its interface consists of two parts: hardware (HW) and software (SW). Besides monitors, HMI hardware may include a standard keyboard with a trackball or a mouse and a functional keyboard.

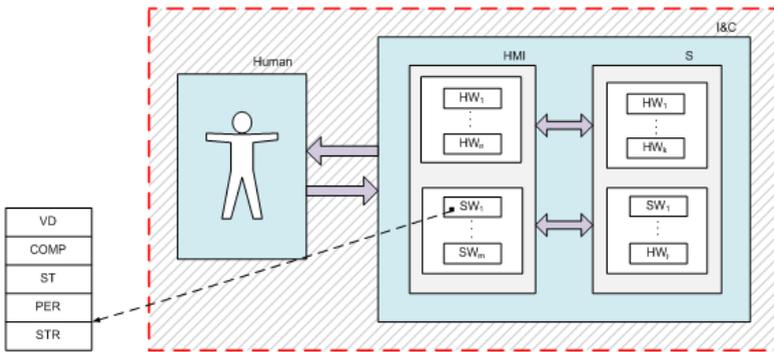


Figure 31.2 -. Model of the human-machine system

Detailed structure of the display system is provided at the design stage. HMI software model can have lots of levels:

$$HMI = \{ STR, PER, ST, COMP, VD \},$$

where *STR* – strategy; *PER* – capabilities; *ST* – structure; *COMP* – layout; *VD* – visual design.

The level of strategy (*STR*) defines objectives of the interface and the user needs; functional specifications and information requirements are determined at the level of capabilities (*PER*), the level of *ST* is for interaction design and informational architecture; layout (*COMP*) and visual design (*VD*) levels define the levels of information and visual design interface. The main factor in achieving high quality HMI is to follow the standards.

31.5 Regulatory framework analysis

Development of HMI is a multi-disciplinary problem. Its scientific rationale and solution requires knowledge of disciplines such as systems design, ergonomics and usability, human factors engineering, software engineering, safety and risk management. There is its own regulatory framework in each of these areas, which regulates approaches, processes, methods and tools for design and evaluation, which may be useful to create an effective methodology for design and evaluation of HMI.

The international standardisation process is an essential mean of ensuring the compatibility of the separate systems. The organisations engaged in standardisation are as follows:

- International Organization for Standardization (ISO);
- International Electrotechnical Commission (IEC);
- Institute of Electrical and Electronics Engineers (IEEE);
- Society of Automotive Engineers (SAE);
- The Internet Engineering Task Force (IETF);
- European Telecommunications Standards Institute (ETSI);
- European Standards Committee (CEN);
- European Committee for Electrotechnical Standardization (CENELEC).

Table 31.1 – Standards and Guidelines

Areas	Principles and recommendations
Standards in the HMIs	<ul style="list-style-type: none"> - IEC 60447:2004 Basic and safety principles for man-machine interface, marking and identification - establishes the main principles of the human-machine interface activation that ensure the control elements to function accurately and timely as well as the safe performance of the equipment in general; - ANSI/HFES 200 Human factors engineering of software user interfaces; - ARINC Specification 661-2. Cockpit display system interfaces to user systems
Standards in ergonomic	<ul style="list-style-type: none"> - ISO 9241:2010 Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems - provides the requirements and recommendations for the human-oriented system design; - ISO CD 23974: Software ergonomics for World Wide Web user interfaces;

	<ul style="list-style-type: none"> - IEC TR 61997: Guidelines for the user interfaces in multimedia equipment for general purpose use; - ISO 15008:2003 Road vehicles - Ergonomic aspects of transport information and control systems - Specifications and compliance procedures for in-vehicle visual presentation; - ISO 15005:2002 Road vehicles - Ergonomic aspects of transport information and control systems - Dialogue management principles and compliance procedures; - ISO 17287:2003 Road vehicles - Ergonomic aspects of transport information and control systems - Procedure for assessing suitability for use while driving
<p>The standards for Intelligent transport systems</p>	<ul style="list-style-type: none"> - ISO/TR 10992:2011 Intelligent transport systems - Use of nomadic and portable devices to support ITS service and multimedia provision in vehicles; - ISO 15662:2006 Intelligent transport systems - Wide area communication - Protocol management information; - ISO/TS 17419:2014 Intelligent transport systems - Cooperative systems - Classification and management of ITS applications in a global context; - ISO/TS 17423:2014 Intelligent transport systems - Cooperative systems - ITS application requirements and objectives for selection of communication profiles; - ISO/TS 17427:2014 Intelligent transport systems - Cooperative systems - Roles and responsibilities in the context of cooperative ITS based on architecture(s) for cooperative systems; - ISO/TR 17465-1:2014 Intelligent transport systems - Cooperative ITS - Part 1: Terms and definitions; - ISO/TS 19321:2015 Intelligent transport systems - Cooperative ITS - Dictionary of in-vehicle information (IVI) data structures; - ISO 21213:2008 Intelligent transport systems - Communications access for land mobiles (CALM) - 3G Cellular systems; - ISO 24978:2009 Intelligent transport systems - ITS Safety and emergency messages using any available wireless media - Data registry procedures.
<p>Standards in the on-board interfaces</p>	<p>SAE Standards and Recommended Practices from the SAE Safety and Human Factors Committee:</p> <ul style="list-style-type: none"> - SAE J2364 Navigation Function Accessibility While Driving;

	<ul style="list-style-type: none"> - SAE J2365 Calculation of the Time to Complete In-Vehicle Navigation Tasks; - SAE J2399 Adaptive Cruise Control (Acc) Operating Characteristics and User Interface; - SAE J2400 Forward Collision Warning Systems: Operating Characteristics and User Interface; - SAE J2802 Blind Spot Monitoring System (BSMS): Operating Characteristics and User Interface - SAE J2808 Road/Lane Departure Warning Systems: Human Interface; - SAE J2831 Design and Engineering for In-Vehicle Alphanumeric Messages.
Security	<ul style="list-style-type: none"> - ISO/IEC 27000:2014 Information technology. Security techniques. Information security management systems. Overview and vocabulary; - ISO/IEC 27001:2013 Information technology. Security techniques. Information security management systems. Requirements; - ISO/IEC 27002:2013 Information technology. Security techniques. Code of practice for information security management; - ISO/IEC 27003:2010 Information Technology. Security Techniques. Information Security Management Systems Implementation Guidance

Fig. 31.3 shows possible profile-forming database of standards for the choice of methods and processes for design and evaluation of HMI.

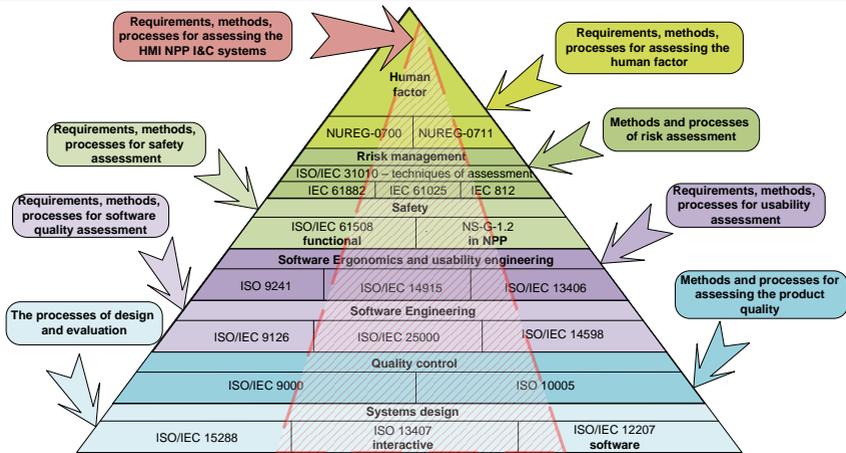


Figure 31.3 – Profile-forming base of standards

31.6 Design principles of human-machine interfaces of resilient systems in nuclear power plants

Basic design principles and requirements for HMI of NPP I&Cs are given in [12]. The same principles can be used as criteria for assessing the quality of I&Cs HMI.

Nevertheless all these principles are important for proper HMI design, some of them may contradict with another, so a compromise between different principles should be reached to ensure effective system design. That is why it is important to identify the relative weight of the principles in comparison with other principles.

Results of the expert analysis and ranking of these principles/criteria are given below.

Personnel safety - this principle is ambiguous. In the broad sense, PS is a consequence of the implementation of its main purpose – to provide the safety of NPP. In this sense, it is an integral characteristic, which is inapplicable as a basic design principle. In a narrow sense - as an independent criterion - this principle can be attributed to the safety of I&Cs HMI only, which depends mostly on hardware components of the HMI and cannot deviate significantly under condition that I&Cs is built on modern technical means (for example a workstation monitor can affect user's vision, but all modern LCD monitors are rather similar from this point of view), so relative weight of this principle is rather low in comparison with other principles.

Cognitive compatibility and physiological compatibility - these principles require physiological and psychological capabilities of the operator and the level of his training to be taken into account when designing HMI. As main criteria, these principles allow us to estimate the quality of information, as well as ease of its perception, analysis and understanding. This is very important criteria for the human factor.

Consistency is among high priority principles/criteria. Only mutual coherence feedback to the operator through different channels of information can allow him to make right decisions. Hierarchy of priorities of the informational sources must be clearly defined in case of conflicting data.

Situation awareness is one of the most important principles because it describes the ability of HMI to perform its basic function - to provide an understanding of the situation by the operator by providing him accurate information on the status of the systems.

Task compatibility indicates that the system should meet users' requirement. This feature also is one of the most important, because the system must conform to its destination.

Error tolerance and control – priority of this principle depends on the class of the System. For systems important to safety, this characteristic has very high priority, because it can directly affect the safety of NPP.

Organization of HSI elements - this principle ensures the provision of the information to personnel in accordance with the distribution of roles in the power unit control, the most important information relating to security should be available to all operational staff. This principle is important enough, but not critical.

The low-priority design principles include:

Cognitive Workload – information should be fast perceived and understood. System must minimize requirements for in-mind calculations and conversions, and use some hints. The background data must be presented in a convenient form;

User Model Compatibility – all aspects of the system should be compatible with the mental users' models;

Timeliness– system design must take into consideration users' cognitive capabilities and time limits in connection with the process. The speed of the Informational stream and performance monitoring requirements, which are too fast or too slow, may lead to productivity decline;

Logical Structure – all aspects of the system (formats, terminology, sequencing, grouping, and user decision-support aids) should reflect an obvious logic based on task requirements or some other non-arbitrary rationale. The relationship of each display, control, and data-processing aid to the overall task/function should be clear. The structure of the interface and its associated navigation aids should make it easy for users to recognize where

they are in the data space and should enable them to get rapid access to data not currently visible (e.g., on other display pages). The way the system works and is structured should be clear to the user;

Flexibility – the system should give the user multiple means to carry out actions and permit displays and controls to be formatted in a configuration most convenient for the task;

Feedback – the system should provide useful information on system status, permissible operations, errors and error recovery, dangerous operations, and validity of data.

Simplicity of design – the HMI should represent the simplest design consistent with functional and task requirements.

All of these principles should be considered when designing HMI of the I&Cs, however, because the real HMI is a solution based on a compromise, which doesn't satisfy the above criteria completely, the greatest attention should be given to the high priority principles.

There are some results of the safe HMI design principles ranking on Fig.31.4.

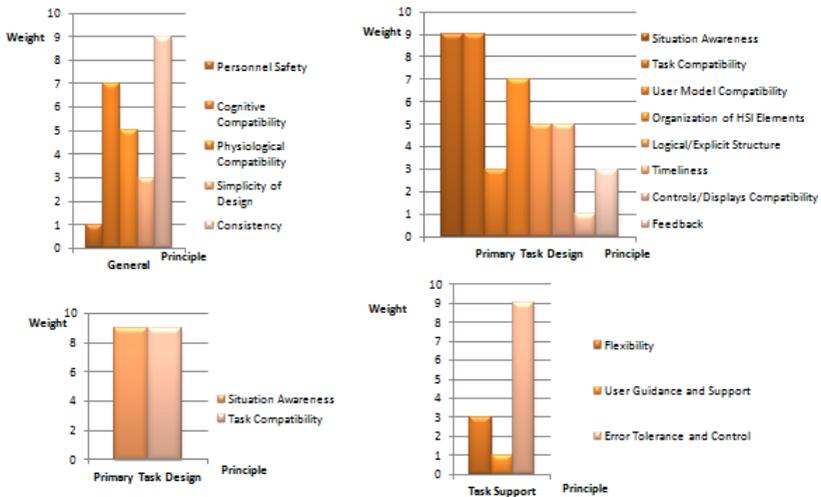


Figure 31.4 – Design principles ranking

31.7 Requirements to human-machine interfaces for intelligent transport systems

The main design manuals regarding the HMI for vehicles are European Statement of Principles on Human Machine Interface [13], JAMA – Japan Automobile Manufacturers Association Guidelines for InVehicle Display Systems [14] and Alliance of Automobile Manufacturers (AAM) [15]. These manuals summarize the key aspects of safety applicable for the human-machine interfaces of the automobile and communication systems.

The parameters and requirements for the CHMI for the ITSs identified as a result of the analysis into the standards, recommendations and the context of the use are given in the table 31.2.

Table 31.2 – Requirements for the HMI for the ITSs

Parameter	Requirement description
Usability	<ul style="list-style-type: none"> – the feedback between the system and the driver should be timely and recognizable; – the driver should be given the information about the current state of the system and any system malfunction; – visual information should be displayed in a way that the driver can assess special details within few sights
	The driver should anytime have the possibility to keep at least one hand on the steering wheel when interacting with the system
	The system should not hinder the driver’s field of vision
Safety	<ul style="list-style-type: none"> – the system should help the driver and should prevent the possible dangerous behaviour of the driver or other road users; – the system should not distract the driver and draw his attention that should be focused on monitoring the road situation; – the system should not provide the driver with the information that can cause the dangerous behaviour of the driver or other road users; – the system should provide the driver with high-priority information rather than the information related mostly to the safety;

	– the system should not hide the vehicle control elements and the displays purposed for driving primarily
Simplicity	The system instructions should be simple, correct and easy to understand
	The visual information should be given piece by piece to ensure the step-by-step control of the system
Cognitive compatibility	The interface should not cause the driver's mixed reaction. The result of the drivers' actions should not be different from what he expects
Other requirements	– brightness, contrast, colours and other parameters of the display should not blind the driver in the night; – the system producing sounds with the volume that can not be adjusted by the driver should not block the sound messages inside and outside the vehicle

31.8 Principles of cybersecurity usability

Usability and cybersecurity complement one another. A well-designed system needs to make it easy for the user to do the right thing, hard to do the wrong thing, and easy to recover when the wrong things happen anyway. In [16] the principles and requirements to ensuring the cybersecurity usability are analysed.

Table 31.3 – Principles of cybersecurity usability

№	Principles	Requirements
1	Accommodate all types of users	Cybersecurity functionality should be designed such that it is flexible and accommodating to novice and expert users
2	Give informative feedback	Feedback should be clear, informative, sufficient, not too technical and where appropriate, give suggestions for going forward
3	Provide help, advice and documentation	Users should be able to easily locate and view help and advice manuals and

		system documentation for cybersecurity functions
4	Error prevention, handling and recovery/Undo	Systems should be designed such that they anticipate user errors and prevent them. If errors do occur however, they should be handled gracefully, be presented in informative prompts and outline steps for recovery. Cybersecurity interface designs support undo and quick exit functionalities for when users make mistakes and enter unwanted application states. Users should be able to rely on and not feel at a loss within the application
5	Allow for visibility of system state	Users should be made aware of the current security state of the system
6	Make security functionality visible and accessible	Security should be visible and easily accessed
7	Reduce cognitive load associated with system activities	Cybersecurity interfaces should be designed to minimise a user's cognitive load whilst using the system
8	Give guidance on what tasks users need to perform and where necessary, provide recommendations support	Systems need to make users aware of and where necessary, supply them with guidance on the cybersecurity tasks they need to perform
9	Emphasise a positive system experience and good levels of user satisfaction	Cybersecurity interfaces should aim to provide users with a positive and satisfactory experience
10	Aesthetic and minimalistic design	Designers should aim to keep interfaces simple, reduce likelihood of information overload, and avoid awkward interface setups
11	Design for learnability	Cybersecurity interfaces should be easy to learn
12	Reduce use of technical and security-specific terms and jargon	To use security features, users have to be able to understand what they mean. Designers should use technical and

		security-specific terms sparingly and where they are used, consider giving descriptions
13	Facilitate the creation of an accurate mental model	Designers should attempt to define systems that consider a user’s mental model, and therefore foster the creation of models that accurately represent the cybersecurity interface and functionality
14	Design such that security does not reduce performance	Designers should utilise efficient algorithms and careful design to ensure that security features can be efficiently used within the software application and system

31.9 Human-machine interface quality models

The variety of software quality models were developed within the framework of program and usability engineering. Most of the models are hierarchical [17, 18]. The table 31.4 contains the most well-known models, which are applied to assess the quality of developed software and its user interface.

Table 31.4 – Existing Models Classification

Types	Models																						
	McCall's (1977)	Gould (1988)	Booth (1989)	Hix et al. (1993)	Wixson (1997)	Dix et al. (1998)	ISO 9241-11(1998)	Lecerov et al. (1998)	Thomas (1998)	Kengeri et al. (1999)	Battleson et al. (2001)	Donyace et al. (2001)	ISO 9126-1 (2001)	ISO 9126-2 (2001)	ISO 9126-3 (2001)	ISO 9126-4 (2001)	Campbell et al. (2003)	Shneiderman (2005)	QUIM (2006)	Sauro et al. (2009)	ISO 25010 (2011)		
Integrated																							
Standardized							+					+	+	+	+	+						+	
Not standardized	+	+	+	+	+	+		+	+	+	+						+	+	+	+			

The HMI model in set-theoretic can be set as a cortege of the following elements:

$$QM_{HMI\&C} = \langle G, MSC, MM, W, DATA, ART, CONT \rangle, \tag{31.1}$$

where G the interface purpose; MSC the set of characteristics; MM the set of metrics; W the set of characteristics and metrics ranks; $DATA$ the set of data for metric measurements; ART the artefacts set; $CONT$ conditions of use.

$$MSC = MFA \cup MCR, \tag{31.2}$$

where MFA the set of factors; MSC the set of criteria.

$$CONT = \langle MUS, MTA, EN, EQ \rangle, \tag{31.3}$$

where MUS the set of users; MTA the set of tasks; EN the environment in which HMI operates (temperature, humidity etc); EQ HMI equipment.

Structure of HMI quality model is presented in figure 31.5.

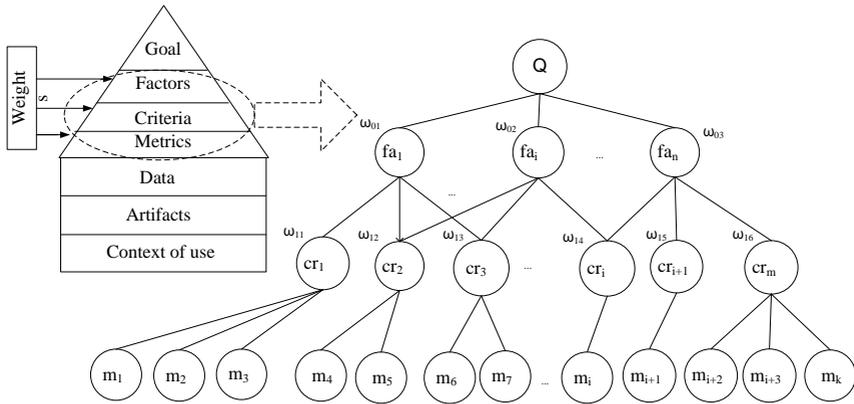


Fig. 31.5 – Structure of HMI quality model

The problem of HMI quality model development requires the consideration of new factors and criteria, based on principles. The development of new criteria is based on the introduction of new metrics, which must reflect the most important aspects of measuring attributes, and have to be rather easy.

An advantage of this model is in taking into consideration most factors that have any influence on HMI quality. It helps to avoid conflicts between different quality requirements.

31.10 Metrics

31.10.1 Safety issue

The realization of the safety requirements for the critical systems is one of the main issues in HMI development. These characteristics determine the system's ability to reach an acceptable level of risk for people's health, their business, software, property or environment in a given context of use.

Quality model analysis has shown that safety metrics are not sufficiently developed. The most widely-known quantitative metrics for safety rating in ISO quality models are following [17]:

– *Safety of the user and of his/her health.* Health troubles may include: injuries from muscular tension, tiredness, headache and so on.

$$X = 1 - A / B , \quad (31.4)$$

where A the number of users, which have reported the problems; B the total number of users.

– *Safety of the people involved in using the system.*

$$X = 1 - C / D , \quad (31.5)$$

where C number of people under risk, D number of people using the product.

The situations with the risk of economical detriment can be also considered.

– *Economical detriment.*

$$X = 1 - E / F , \quad (31.6)$$

where E number of economical detriment cases; F total number of cases, when the system was used.

Situations with a risk of other software damage can be consideration too.

– *Other software damage.*

$$X = 1 - M / F , \quad (31.7)$$

where M number of other software damage cases; F total number of cases, when the system is being used.

The metric (7) can be also calculated as (8):

$$X = N / T , \quad (31.8)$$

where N total cost of damaged software; T time of use.

Thus, an existing set of metrics can't fully characterize the safety of I&Cs and its HMI. Based on this the introduction of new metrics is required.

31.10.2 Security issue

Ensuring and assessment of the security of visual user interface for automated information processing and management systems is an urgent task. In this domain area, it is clearly required to form and develop the conceptual basis.

The security is used in the standards as: the capability of the software to protect information so that unauthorized persons or processes are not able to read or modify it but authorized users and processes are provided with the access to it. It is underlined in the standards that this requirement refers to the transmitted data as well.

The security metrics for HMI are not well developed at the moment. The most popular external quantitative metrics for information safety assessment in ISO quality models are as follows:

– *Access tracing*

$$X = A / B, \quad (31.9)$$

where A is the number of “the facts when users accessed the system and data” registered in the system protocol; B is the number of “the facts when users accessed the system and data” during the assessment.

The metric is experimental. It is recommended to use “penetration tests” to emulate the attacks on the system.

– *Access controllability*. It is required to check the ability of the system to detect the facts of unauthorized access if the system functions are applied in a wrong way.

$$X = A / B, \quad (31.10)$$

where A is the number of unauthorized access types detected; B is the number of unauthorized access types provided in the specification.

– *Data corruption prevention*. It is required to check the correctness of the system working in terms of its functions applied in a wrong way. It is required to determine the effect of data corruption incidents.

$$X = I - A / N, \quad (31.11)$$

where A is the number of the facts of critical data corruption; N is the number of test types used to initiate the fact of data corruption.

$$Y = I - B / N, \quad (31.12)$$

where B is the number of the facts of non-critical data corruption;

$$X = A / T, \quad (31.13)$$

$$X = B / T, \quad (31.14)$$

where T is the time of operation.

In order to calculate the external metrics, the data available outside the system should be used.

Ensuring the HMI information safety includes the integrated consideration of information, functional, psychophysiological and environmental aspects of the safety.

The possible dangers in the area of data security for interaction between users and computers include:

- data misinterpretation due to the environmental effect on user's working place;
- data loss and misinterpretation due to inconsistent data representation;

– misperception of the actual managed object’s state due to indirect information influence.

The countermeasures are directly related to considering the quantitative and qualitative indicators of information flows between the computer and the user as well as characteristics of the user.

The key characteristic of such information flow is the information load on the user that determines the user’s state, work conditions and consistency of the human and technical components.

We can expand the existing quality models by introducing new characteristics, consistent with the principals of HMI I&Cs design in the field of nuclear energy.

The table 31.5 presents an example of quality metrics for “cognitive capability”.

Table 31.5 – Quality metrics for cognitive compatibility

Metric name	Scale	Scale type	Mode of application	Artifacts	The difficulty of obtaining
Purposefulness of tasks	Great, sufficient, satisfactory, unacceptable	Ordinal	User’s testing	User’s monitoring recording	Med.
Awareness of object’s current state	Great, sufficient, satisfactory, unacceptable	Ordinal	User’s testing	User’s monitoring recording	Med.
Workload level	High, medium, low	Ordinal	User’s testing	User’s monitoring recording	Low
Operator’s productivity	High, medium, low	Ordinal	User’s testing	User’s monitoring recording	Low
Vigilance level	High, medium, low	Ordinal	User’s testing	User’s monitoring recording	Med.

31.10.3 Adaptability

The adaptability is additional HMI factor. Property of adaptability in HMI is presented in several forms: change of the given informational input, maintaining the dialog, distribution of the problems between human and machine, adaptation speed.

The flexibility is one of the adaptability criteria. Flexible HMI must provide user with several ways to commit the action, display and control have to be the most suitable for the problem.

Based on a definition of the criterion of HMI flexibility, the following set of metrics can be proposed (Table 31.6).

Table 31.6 – Flexibility metrics

Name of metric	Method	Data	Scale	Way of using	Artifacts
Number of ways to solve problems	$X = A/B$	A - Number of ways to solve problems B – Max. number of ways to solve problems	1 – perfect – stable software flexibility, always implies availability of alternative way of use $X < 1$ - not flexible enough $X > 1$ - too many ways of use, may be a big load B – sets an expert in scientific field	Development testing	Testing reports recording
Level of comfort of displayed data and managing elements provision			High, medium, low	User's testing	User's monitoring recording
System's flexibility efficiency			High, medium, low, no	User's testing	User's monitoring recording
Level of negative load for operator during the use of system's flexibility			Very high, high, medium, low, very low	User's testing	User's monitoring recording

31.11 Ensurance of HMI safety based on Safety Case methodology

Safety Case methodology includes a formal presentation of evidence, arguments and assumptions aimed at providing assurance that the HMI meets safety requirements, and safety requirements are adequate. At the same time attention should be paid to the logical arguments that will be used to demonstrate that the system is safe to use. Purpose, which can be interpreted as testing requirement, is divided into sub-goals until one can identify tools, confirming that the sub-goal is achieved (Fig 31.6). Then these tools are used to verify the safety during development of the system.

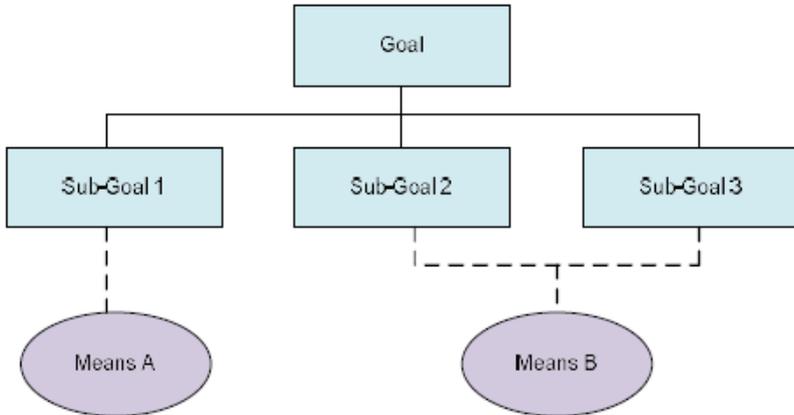


Figure 31.6 – Structure of the objectives

It is important to plan for a Safety Case at the very beginning of the design process. Firstly, this will determine which evidence is necessary to collect and secondly, what should be used to support them in various stages of the life cycle. One problem is the choice of the depth and rigor of evidence. Some items of evidence may be more persuasive than others, and it must be considered when evaluating the effectiveness of the safety case as a whole.

Safety Case Report should contain all necessary information to assess the safety of HMI. The higher safety requirements the more details are required. Good quality Safety Case provides information to the extent and form that make the work of the expert comfortable in terms of reliability, availability, and ease of use. Typical content of the Safety Case includes:

System Description - defines the purpose of the evaluation, describes the system under consideration (the objectives, functions, structure, components, context of use) and its interaction with other systems.

Quality Management Report - gives evidence that the requirements for the process of quality assurance have been met.

Safety management report suggests that an actions, defined in the safety plan, had been implemented. It should include the results of the various analyses, as well as a list of all identified hazards (Journal of Hazards).

Technical Safety Report – it explains technical principles, which provide safety. It should include reports to verify each component, including HMI.

Related Safety Cases – a document that contains references to any Safety Cases for other vital systems, related to the system under consideration.

Findings should be presented in the form of analysis of activities carried out by the developer, and why system attributes are sufficient.

Figure 31.7 shows a conceptual model of the system safety assessment of HMI of I&Cs.

The solution of the safety assessment problems of HMI of I&Cs is complex and directly related to the modeling and analysis of the design process, specification requirements, the context of use and design.

The HMI safety model is constructed by analysis (profiling) of the regulatory framework. The choice of assessment methods directly depends on the safety profile and the stage of the life cycle of the HMI. Before using of different assessment methods, it is important to formalize the process of the upcoming evaluation. This will help to determine the best approach to effectively assess and select the most appropriate method or methods. Selecting of assessment methods should be preferred to those methods which have tool support. Evaluation results have a direct impact on improving of the safety of HMI of I&Cs.

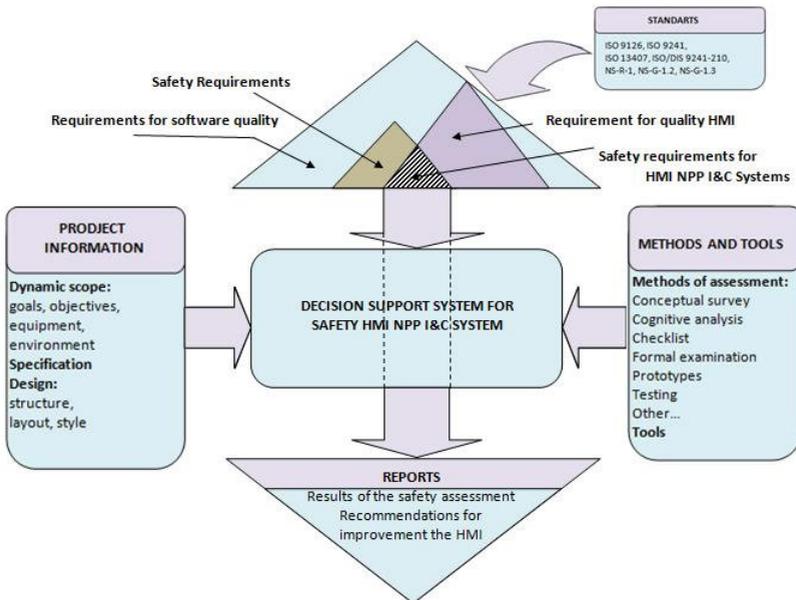


Figure 31.7– Conceptual model of the safety assessment HMI I&C system

General procedure of the Safety Case-oriented assessment is the following. At the first stage HMI safety requirement profile is developed (specified). The profile includes international and industry standards, and regulatory documents developed for various industry domains. The next stage is to determine the goals, objectives and characteristics for the HMI safety evaluation. There is an analysis and a choice of methods of an assessment which directly depends on a design stage, and also from earlier formulated purposes and problems of estimation. The most exact and reliable assessment can be obtained by applying several methods at the same time. The next stage is evaluation of HMI by tools implementing the chosen method. Finally, in the final stage we obtain the results of the evaluation in the form of certain reports and recommendations to improve the HMI. For this an expert combines the results obtained by different methods at the different stages of evaluation. The end result is highlighted in the safety case document, prepared for the evaluated system and HMI.

31.12 Choice of methods

To date, the task of choosing methods for safety assessment in the Safety Case was complicated by the large number of techniques of varying degrees of formality, complexity, ability to use of the life cycle stages, etc.

Since we discuss HMI software only, one can significantly limit the range of the analyzed approaches and methods. As part of UCD-design process of user-centered interactive systems, there is large number of methods relevant to usability [19].

We believe these methods are the most effective at the pre-design gathering stage, at the stage of analysis of the use context (task analysis), as well as at the stage of verification and validation of the finished product (usability testing). Processes and methods of safety HMI evaluation, developed within a software engineering, are mainly focused on the metric evaluation of the finished product.

Methods of risk assessment are given in [20]. Risk assessment can be carried out with varying degrees of depth and detail. The use of one or more methods is possible. When selecting methods, the rationale for their suitability should be presented.

Methods must have the following features:

- to be scientifically sound;
- conform to the system under study;
- to give an understanding of nature and the nature of risk, how to control and process.

Method selection can be implemented based on the following factors:

- purpose of the evaluation;
- system development ;
- type of system;
- resources and opportunities;
- nature and degree of uncertainty;
- complexity of methods;
- ability to obtain quantitative data output;
- the applicability of the method;
- availability and accessibility of information for the system;
- needs of decision makers.

Table 31.7 shows the results of a comparative analysis of several method-candidates for Safety Case. Recommendations and the applicability of a specific technique throughout the risk assessment process of HMI have been considered when selecting methods.

Table 31.7 – A comparative analysis of risk assessment methods

Type of risk assessment methods	Relevance of influencing factors			Possibility of the use of the HMI
	Resources, and capability	Nature and degree of uncertainty	Complexity	
Checklists	Low	Low	Low	+
Preliminary analysis of the hazards	Low	High	Average	–
Scenario Analysis	Average	High	Average	–
Fault tree analysis (FTA)	High	High	Average	–
Analysis of the "tree" of events	Average	Average	Average	–
Analysis of the causes and consequences	High	Average	High	–
The analysis of types and the consequences of failures (FMEA and FMECA)	Average	Average	Average	+
Hazard and Operability Study	Average	High	High	+

(HAZOP)				
Reliability assessment of the operator (HRA)	Average	Average	Average	+
Multi-criteria decision analysis (MCDA)	Low	High	Average	+
“+” - applicable; “-” - no data				

A possible profile of methods for Safety Case and the process of integrated assessment of HMI of resilient systems at all stages of the life cycle is shown on Fig. 31.8.

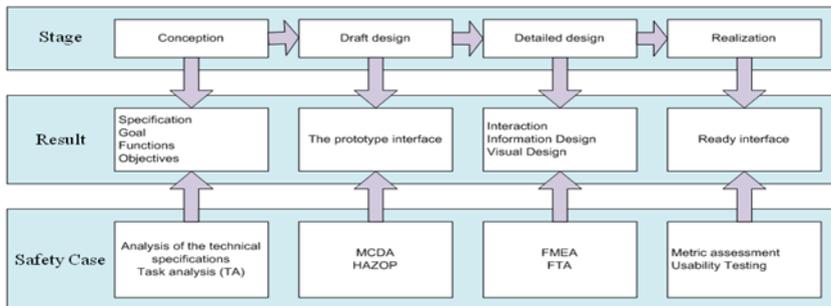


Figure 31.8 – Profile of methods

Conclusion and self-control questions

Human-machine interfaces are an important part of information and control systems for commercial and critical domains. The ability to control and manage systems and objects, for instance reactor, aircraft or medical equipment, the efficiency and reliability of the human and I&Cs as a whole, depend on the HMI quality.

The HMI development process is based on variety of modern technologies

Scalable and flexible interfaces of the operator’s panels allow to integrate into the HMI different systems monitoring and control functions.

HMI for the automobile informational systems improves the traffic safety by decreasing the driver’s informational overload, and thus minimizing the distractions.

Here's what's relevant now in the HMI field: human factor studies in order to reduce the likelihood of errors; analysis of the reliability of operator's actions associated with the risk assessment and taking into account the possible consequences; development of techniques for evaluation of safety.

HMI will be one of the major topics to which investigations in the field of transport safety are going to be devoted in the nearest future.

Safety assessment of the HMI is based on the Safety Case methodology, which allows us to improve the completeness and reliability of the integrated assessment at all stages of the life cycle from concept to finished product.

Rationale and methods selection is done by multidisciplinary profile-forming regulatory framework, which let us to combine the Safety Case methods in software engineering, risk assessment, human factor engineering and usability.

The variety of software quality models were developed within the framework of program and usability engineering. The modern standards do not strictly define the requirements for new interfaces. The problem of HMI quality model development requires the consideration of new factors and criteria, based on technology new principles. The development of new criteria is based on the introduction of new metrics, which must reflect the most important aspects of measuring attributes, and have to be rather easy.

The realization of the safety and security requirements for the critical systems is one of the main issues in HMI development. Quality model analysis has shown that safety and security metrics are not sufficiently developed.

Self-control questions and tasks

1. Please define the human-machine interface.
2. What are the key characteristics of HMI?
3. What principles of ensuring the cybersecurity usability do you know?
4. Please specify the requirements to the HMI for the ITSs.
5. What is the environmental interface?
6. What is the quality model?
7. Please provide the examples of standardized quality models of software user interfaces.
8. What safety metrics do you know?
9. What security metrics do you know?
10. What does the term "cybersecurity usability" stand for?
11. What is the user compatibility model?
12. What does the characteristic of "cognitive compatibility" stand for?
13. Please name the qualitative metrics of cognitive compatibility.
14. How does the HMI adaptivity express itself?

15. What quantitative flexibility metrics of HMI do you know?
16. Why is it required to rank the quality characteristics?
17. What is the key element of data representation in HMI for ITS?
18. What information does the operator receive from video cards and how is it presented?
19. What is the essence of Safety Case methodology?
20. What does Safety Case Report include?

References

1. Green U-Home Terminal. [Electronic resource] - Access mode: <http://pacificcontrols.net/products/green-u-home-terminal.html>.
2. The Personal Adaptive In-Car HMI: Integration of External Applications for Personalized Use. [Electronic resource] - Access mode: http://link.springer.com/chapter/10.1007%2F978-3-642-28509-7_5#.
3. Anokhin A. Designing interfaces / A. Anokhin, N. Nazarenko // *Biotechnosphere*. – 2010. – № 2 (8). – P. 21-27.
4. Anokhin A. Adaptive human-system interface for control of complex systems (in application to nuclear power plant / A.N Anokhin and E.C Marshall // *Book of abstracts of the 21st European Meeting on Cybernetics and System Researches, EMCSR 2012, Vienna, Austria, 10-13 April*. – 2012. – P. 185-188.
5. Orekhova A. Human-machine interface quality assessment techniques: Green and safety issues / A. Orekhova, V. Kharchenko, A. Orekhov // *Proceeding of 10th IEEE International Conference on Digital Technologies "DT 2014"*, Zilina, Slovakia, 2014. – P. 259–264.
6. Orekhova A. Analysis of the requirements for interfaces NPP I&Cs / A. Orekhova, V. Kharchenko // *Bulletin KNTU Named after P. Vasilenko. Engineering. Issue 102. "The problems of energy and Saving energy in agriculture of Ukraine."* - Kharkov: KhNTUA. – 2010. – P.109-111.
7. What technical innovations has Volvo implemented for last 10 years? [www.autoconsulting.com.ua/news]. – 2011.
8. Cars In The Future : Human Machine Interface [<http://www.rospa.com/roadsafety/policy/carsinthefuture/human-machine-interface.aspx>]
9. Orekhova A. Safety case-oriented assessment of human-machine interface for NPP I&C system / A. Orekhova, V. Kharchenko, V. Tilinskiy // *Reliability: Theory & Applications*. – 2012. Vol. 3 (26). – P. 27 – 38.
10. Bauer E. PRORETA 3: An Integrated Approach to Collision Avoidance and Vehicle Automation / E. Bauer, F. Lotz, M. Pfromm // *At - Automatisierungstechnik*. – 2012. – № 12. – P. 755-765.

11. Orekhova A. The Cooperative Human-Machine Interfaces for Cloud-Based Advanced Driver Assistance Systems: Dynamic Analysis and Assurance of Vehicle Safety / A. Orekhova, V. Kharchenko, E. Brezhnev, A. Orekhov, V. Manulik // Proceedings of IEEE East-West Design&Test Symposium (EWDTS'2014). – IEEE Kyiv, Ukraine, 2014. – P. 82–86.

12. Human-System Interface Design Review Guidelines, NUREG-0700, U.S. Nuclear Regulatory Commission Office of Nuclear Regulatory Research Washington. – 2002. – 659 p.

13. European Statement of Principles, European Statement of Principles for in-vehicle information and communication systems. – 1999.

14. JAMA - Japan Automobile Manufacturers Association Guidelines for In-Vehicle Display Systems. – 2004.

15. Alliance of Automobile Manufacturers (AAM) Statement of Principles, Criteria and Verification Procedures on Driver Interactions with Advanced In-Vehicle Information and Communication Systems. – 2006.

16. Jason R. C. Nurse, Guidelines for Usable Cybersecurity: Past and Present / Jason R. C. Nurse, Sadie Creese, Michael Goldsmith, Koen Lamberts // [Electronic resource] - Accessed at: http://www.cs.ox.ac.uk/files/6638/CSS2011_NCGLaauthorsfinal.pdf

17. International Standard ISO/IEC 9126-1:2001 Software engineering – Product quality – Part 1: Quality model [Electronic resource] / ISO. – Accessed at: http://www.iso.org/iso/home/store/catalogue_tc.htm.

18. International Standard ISO/IEC 25010:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models [Electronic resource] / ISO. – Access mode: http://www.iso.org/iso/home/store/catalogue_tc.htm. – 2011.

19. Orekhova A. Analysis of the criteria and methods for the design of safe interfaces information and control systems / ISTC "ICTM-2010": Abstracts. Volume 2. - Kharkiv National Aerospace University "Kharkiv Aviation Institute". – 2010. – P. 219.

20. International Standard Risk management – Risk assessment techniques: ISO/IEC 31010. – 2000.

PART 8. HUMAN-MACHINE ENGINEERING FOR SECURITY CRITICAL AND RESILIENT SYSTEMS

CHAPTER 32 RESILIENT COOPERATIVE HUMAN-MACHINE SYSTEM

32.1 Problem of transport infrastructure safety

According to forecasts of the World Health Organization by 2030 number of victims of road accidents can reach more than two and a half million people per year [1]. Active application of information and communication technologies (IT) can be considered as a strategy to improve the safety of transport infrastructure, reduce accidents, improve service quality and reduce its negative impact on the environment. All of these ITs are fully considered within the framework of unified intelligent transportation system (ITS).

ITS includes a variety of applications, such as traffic management systems, information systems of vehicles, advanced driver assistance systems in motion (ADASs - Advanced Driver Assistance Systems), as well as cooperative applications based on the exchange of information between ITS stations and transport infrastructure.

Different vendors on IT market offer the advanced driver assistance systems [2, 3]. Such systems as a collision warning system, parking assistant, are designed for improvement of safety during the driving and reducing the driver's strain. [4].

One of the development lines of such systems is the improvement of the interaction between the driver and the vehicle control system "human-machine" (Human-Machine Interaction) and the provision information about the current situation on the road in real time for driver (Real-Time Traffic and Travel Information (RTTI)).

The provision this sort of information leads to an increase of situational awareness of vehicle driver. Awareness implies existence of operational information about the vehicle state and road conditions. Sufficient level of situational awareness is required for risk assessment and hazard analysis, planning, goal-setting, etc.

Traditionally, situational awareness includes three levels: (1) the level of perception of the situation, which is provided by monitoring the status of various objects around the vehicle; (2) the level of conclusions, which determines the ability of vehicles to integrate various sources of information and to make assessments of situations on this basis (given level is provided by the decision-making about the current dangers and risks for the vehicle); (3)

the level of prediction, on which the forecast of dangerous situation risks is carried out.

Undoubtedly, increasing of situational awareness leads to overall risk lowering (collisions, overturning, etc.), since it is possible to detect and predict hazardous situations, determine precautions for their reducing in real time. This way, for example, a prediction of great number of unsafe trajectories neighboring vehicles is performed, as well as dynamic risk zones, zones of "comfort" of the vehicle, etc.

Great importance for enhancing of situational awareness has issues for construction of secure dynamic human-machine interfaces (HMI) [5, 6].

At the same time, the point is that are two sides of the safe HMI: firstly, the development and evaluation of interfaces according to the requirements of the normative documents and safety standards, and secondly, reporting succinct information about objects in the area of the vehicle movement to the driver, which can threat him (area of potential hazard (APH)). It is also necessary to take into account the ability of an HMI to adapt to the situation on the road, to take into account the state of the driver, its features, driving experience, behavior peculiarities in critical situations, habits, etc., i.e. increasing of its adaptability.

The high amount of data used in the ITS, leads to the necessity of improvement of information access for all traffic participants. Improvement of situational awareness, risk assessment in the real-life improvement requires the use of large computing facilities for the storage, processing and analysis of data. These facilities are not always available, even for modern on-board computing equipment of vehicle.

Reliability of on-board software is also an additional safety factor in the ITS. It is necessary to consider additional precautions to enhance safety, including the possibility of using modern cloud computing for information processing in the framework of the ITS.

32.2 Methods of safety analysis for intelligent transport systems

One of the main features of the ITS will be an ability to predict risks and improve the safety of the vehicle. This ability will be provided through the use of various types of models and methods of dynamic safety analysis.

The input data for these models and methods will be the data from the ITS stations of other vehicles or infrastructure on the whole. On-board software of ITS station must address problems of risk assessment, detection and forecasting of hazards, improving situational awareness of the driver in real time.

Currently, the dynamic risk assessment uses a variety of methods. The main ones, used in risk analysis of vehicles are:

- The traditional methods used in the safety analysis of complex systems. These include FMECA, ETA, FTA, HAZOP and their extensions for dynamic analysis;

- Methods based on the theory of Systems-Theoretic Accident Modeling and Processes (STAMP) [7]. Their adaptation to the vehicle safety analysis is based on the assumption that accidents occur due to inadequate control by, for example, advanced driver assistance systems. Categories of inadequacy in this case are: inconsistency, command tardiness, etc.;

- Methods based on the use of Bayesian Belief networks (BBN) [8]. Usage of BBN allows to take into account many factors that affect vehicle safety, for example, the characteristics of the road surface, traffic, climate (weather) conditions, the state of the driver (experience, age, physical condition, level of intoxication, etc.), type of vehicle, etc.; for dynamic safety analysis can be used dynamic BAN;

- Methods based on multi-agent simulation, where ITS is considered as a system formed by multiple interacting intelligent agents that have goals, objectives, strategies, behavior, etc. [9];

- Methods of artificial intelligence. This group of methods is used primarily for autonomous vehicles (without driver) [10].

Dynamic criticality matrices, which allow prediction of the risks in terms of likelihood and severity with the aspect of crossing zones of vehicle "comfort" can also be leveraged in the safety analysis [11].

The application of these techniques in real-time for risk analysis can improve the driver's situational awareness, predict the situation, provide support for decision-making under conditions of high dynamics of the traffic situation and exchange of data within the ITS between distributed stations. It is obvious that the vehicle drivers have different awareness of the current situation. Thus, the exchange of information between ITS stations would allow a substantially increase of security and collective awareness of all road users (all about everyone else).

32.3 Driver assistance systems

Deployment of these systems creates prerequisites for the further vehicle intellectualization based on the newest computer technologies, satellite navigation and wireless technologies. These systems are capable of warning the drivers about dangers in motion. They incorporate the systems that provide for the connection and information interchange between vehicles (V2V – vehicle-to-vehicle), between vehicle and infrastructure (V2I - vehicle-to-infrastructure) and between different parts of an intellectual transport infrastructure (I2I – infrastructure-to-infrastructure) fig 32.1 – 32.2.

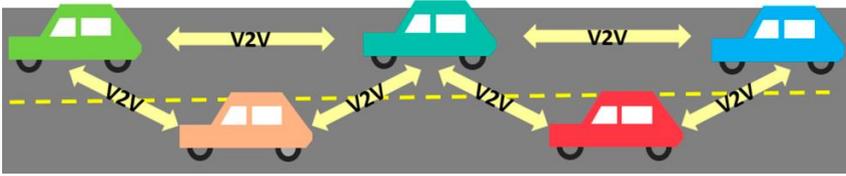


Figure 32.1 – Communication of V2V type

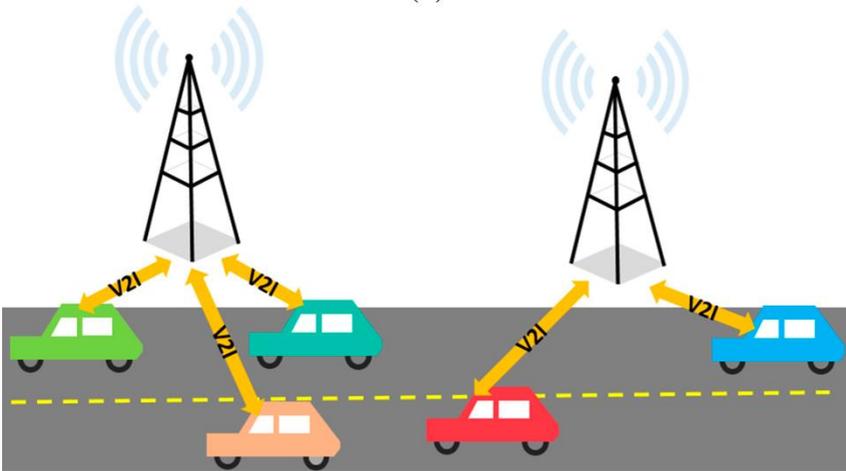


Figure 32.2 – Communication of V2I type

Intelligent transport infrastructure includes complex of equipment that secures acquisition of the almost full information about the road situation and the possibility of a quick response to the changing conditions. If necessary, these systems are complemented with the Global Navigation Satellite System (GNSS).

The infrastructure of the ITSs includes:

- the road complex of all subsystems, among them are the technical monitoring tools, tools for analysis and decision making according to the functional tasks of the subsystems, the control function implementation tools;
- the situational and operations control centers;– wire traffic support tools the purposed to execute the functional tasks of the subsystems;
- information and telecommunication means that ensure the secure interaction with the outside information systems.

Much attention is given to the issues of human factor and HMI in the ITS [3, 12]. HMI will be one of the major topics to which investigations in the field

of transport safety are going to be devoted in the nearest future, as marked in [12]

Motorcar companies offer a whole set of advanced driver assisting systems, for example:

- collision warning system;
- pedestrian detection system
- blind spot information system;
- lane departure warning system;
- driver fatigue monitoring system;
- driver hypo-vigilance system;
- speed alert system;
- drunk driving prevention system.

The table 32.1 provides the examples of the implementation of the advanced driver help systems.

Table 32.1 – Driver help systems implementation examples

System	Manufacturer
Collision warning system with Auto Brake	Volvo
Pre-collision System	Toyota
Adaptive cruise control	Volvo
Lane departure warning system	Volvo
Automated Highway Driving Assist System	Toyota

To use such systems effectively one needs an HMI that maintains human-vehicle interaction and mitigates the negative errors impact on the safety, allows avoiding misinterpretation of the information that the system provides.

32.4 Development of human-machine interfaces

In the European declaration on the principles of HMI functioning [13] In-Vehicle Information Systems (IVIS) designing foundations are offered. The systems must not distract a driver, and the information they convey to the driver has to be predictable and controlled. It is important that interaction with the informational systems neither burdens the driver of the vehicle nor distracts

him. The systems must give the information in a concise and comprehensive way.

Systems that need to communicate to the driver must be easy to use and always keep the driver doing the principal task that consists in driving a vehicle safely. Good HMI system reduces the informational strain on the driver helping to select the most relevant and important information.

As noted in the documents of the European Commission, safe HMI design must take into account the need to integrate nomadic devices and ensure the safety of vulnerable traffic participants (e.g. aged people). Nomadic devices include information and communication equipment such as mobile phone, navigation system, PDA, etc. All these devices are typical examples of the vehicle information systems.

Using nomadic devices may be not matched with a car, especially if their HMI is designed poorly. It should be noted that in the future we should expect to see an increasing number of new systems with different haptic, visual and auditory methods of communication with drivers. Therefore, all the risks associated with the use of such systems should be estimated.

Recommendations on the design of safe HMI of the IVIS suggested in the project Human Machine Interface and the Safety of Traffic in Europe (HASTE) [14].

Among the objectives of the research program within HASTE are the following:

- identify and explore scenarios in which safety issues are most important and relevant;
- explore the connection between the load and the risk in the context of these scenarios;
- conceive the mechanisms of risk increasing in terms of distraction and the driver situational awareness reducing;
- determine risk rates;
- apply existing risk assessment methods to real vehicles;
- consider possible causes of information systems threats related to safety and reliability.

32.5 Cloud-based intelligent transportation system

Cloud computing (CC) is used to receive or transmit data over the Internet via a wireless connection. The idea of using cloud services in ITS is just beginning to gain popularity [15]. The facilities of "clouds" can also affect the increase of transport safety.

Re-engineering the vehicle to a cloud-based technology is discussed in [16]. Vehicles with a global positioning system that are connected to the "cloud", will always "know" their location and the road conditions. Today,

some of the features for vehicle are designed with innovative technology, using the Internet, for example, the communication function V2V (vehicle-to-vehicle) and V2R (vehicle-to-road).

The Volvo Car Group company is working on new car projects - exchange of information about the dangers on the road through the "cloud" and control of the driver's state [17]. The data about the slippery road parts generated basing on the vehicle sensors is passed to the Volvo Cars data base via mobile network on a real-time basis (fig. 32.3). A warning is passed to other vehicles reaching this road part instantly, thus making it possible for the driver to take prompt measures in order to prevent the critical situation.

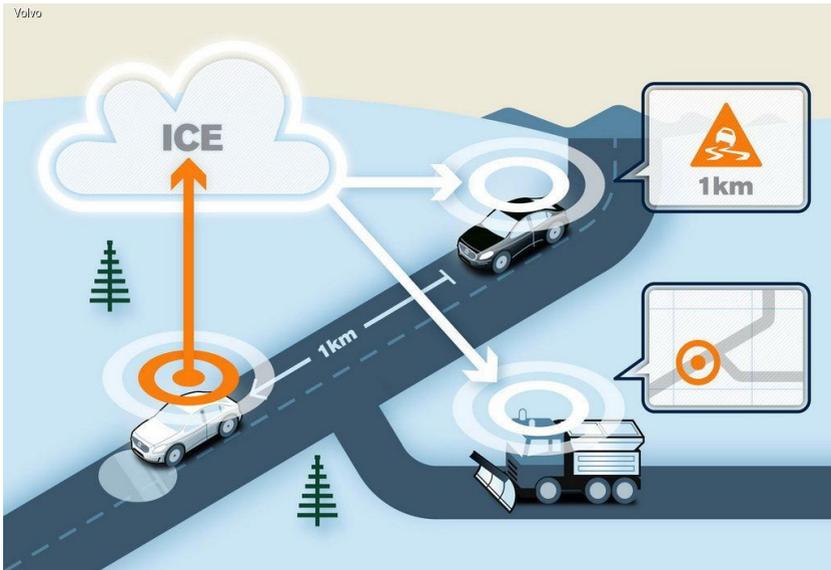


Figure 32.3 – Volvo Cloud service

Other possible application of this technology is the remote diagnostics. Data can be transferred in advance, thus eliminating the problem in real-time [18].

Toyota Motor Corp. and Panasonic jointly develop a service that will connect cars and home appliances through the "cloud" [19].

The review of the systems for driver state analysis is given in the table 32.2.

Driver's exhaustion is assessed by processing multiple parameters:

- vehicle movement (speed, forward and side acceleration, rate of yaw);

Chapter 32 Resilient cooperative human-machine system

- biometric indicators (heart rate, respiration rate, skin temperature);
- driver’s vision (eyes opening rate and vision line);
- driver’s actions (turning angle of the steering wheel, position of the foot and brake throttles);
- road condition (traffic density, road covering).

Table 32.2 – Driver state analyzing systems

System	Sensors used	Implementation examples
DAS (Driver Attention Support) – driver’s exhaustion detection and preventing sleeping at the wheel	<ul style="list-style-type: none"> – IR sensor behind the steering wheel that controls the face temperature – piezoelectric sensor in the safety belt that monitors the breathing rate – patches at the rim of the steering wheel that measure the pulse – IR sensor behind the steering wheel that measure the temperature of the palms 	<ul style="list-style-type: none"> 1) Attention Assist (Mercedes-Benz) 2) Driver Alert Control (Volvo) 3) Seeing Machines (General Motors)
Physical state assessment systems Assessment of the critical health	<ul style="list-style-type: none"> – heart rate sensors installed in the seat – sensors at the rim of the steering wheel: electrodes that monitor the heart rhythm and optical sensors that assess palms 	<ul style="list-style-type: none"> 1) Driver load assessment system (Ford) 2) Aged driver’s state control system

indicators: – pulse; – breathing rate; – skin capacity; – blood sugar level	capacity	3) Vital indicators control system (Toyota) 4) Warning technology for the diabetic drivers (BMW)
--	----------	--

32.6 Cooperative human-machine interfaces

As noted above, the cooperative systems are such systems that wirelessly communicate with other cars. Therefore, under the term of a cooperative HMI we will consider an interface system, distributed among several vehicles. An additional monitor is installed on each vehicle or a compact unit is embedded into the existing HMI to provide information about safety in APH, which gives the information about the safety level. This information (risk matrix) is formed and dynamically adjusted basing on the overall situation for each car (the state of the vehicle, driver and road conditions), which is in the danger zone.

It is clear that these must be adaptive HMI, which reflect not only information about the condition of the car, but of the driver as well. If a driver starts to doze off or falls asleep, it is necessary to wake him up and inform the drivers of motor vehicles that are nearby.

The property of adaptability in the HMI becomes apparent in several forms: changes in the content of the information provided, dialogue, sharing of tasks between man and machine, the speed of adaptation [20].

The project PRORETA is a reserch in the area of the coopertaive HMIs. The research object is the prototype of the cooperative automobile HMI that implements the scenarios of preventing collisions at the cross-roads. The PRORETA HMI system implements a huge number of use scenarios, it does not complicate or irritate and ensures the multimode support. The HMI provides 4 support levels – informaton messages, warnings, actions recommendations, automatic intervention.

One of the variants of cooperative HMI construction - is to use the technology of CC. Fig. 32.4 shows the proposed architecture of the cooperative HMI.

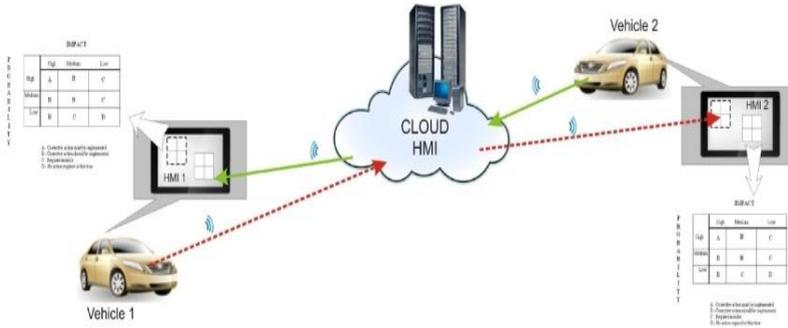


Figure 32.4 – Architecture of the cooperative HMI based on cloud computing

Cooperative HMI provides the measure values of the parameters of vehicle and driver state in real time via the Internet into the "cloud." Here, the data from all the cars is dynamically processed and transmitted to motoring public.

Information from the HMI of one vehicle (shown as a red dashed line, Figure 32.4) passes through the "cloud" and is displayed on the HMI of another vehicle. In turn, the information from the HMI of another vehicle (shown as a green line, Figure 32.4), is also transferred to the HMI of the first vehicle. This information is taken into account when the risk analysis of each vehicle is performed.

There are important issues in developing of HMI: optimization of the information necessary for driver for the safe driving mode; determination of the information views, which stimulate the driver; control and prevention of the driver's distraction.

32.7 Prototype of cooperative human-machine interface

The system consists of three projects combined in a single solution:

- server end – the decision support system (DSS);
- client end – the user HMI;
- Core-project that includes data models for the communication protocol and the common utility functions.

The server end is the web-application, the core of which is the DSS. The web-application is managed by the Apache Tomcat server that supports the HTTP protocol. The protocol allows the interaction between the client and the server. The client end is implemented for the Android platform and it stands for the user interface. The ground map is the key element. The data exchange is performed wirelessly using the data types specified in the general Core-

project. Java serves as the platform for creating the system in question. The figure 32.5 shows the architecture of the system.

The client and the server cooperate wirelessly through the module for communication. The general convenience functions and data models for packetizing can be found in the Core-project that is used by the both sides. Since the communication protocol should provide equal rights for the client and the server, it has been agreed to implement the communication protocol based on TCP from the specification Java EE – WebSocket.

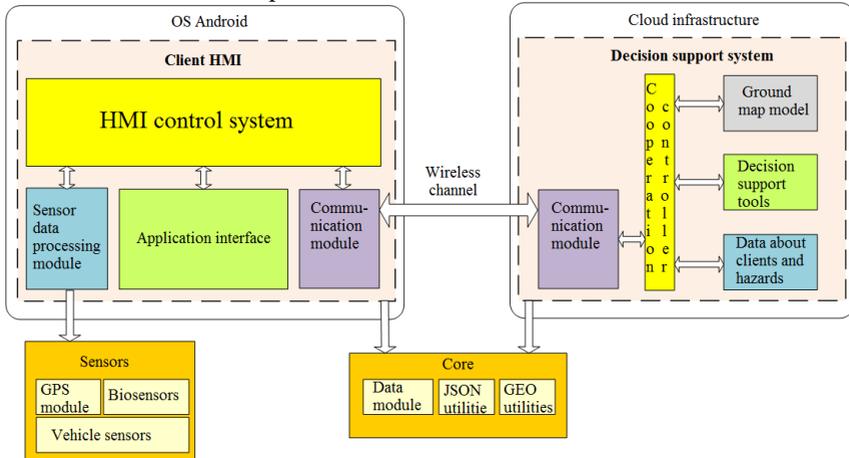


Figure 32.5 – Architecture of the system

The protocol ensures the free data exchange: two equal participants exchange data, each one working independently and sending data to the other one when necessary (fig. 32.6).

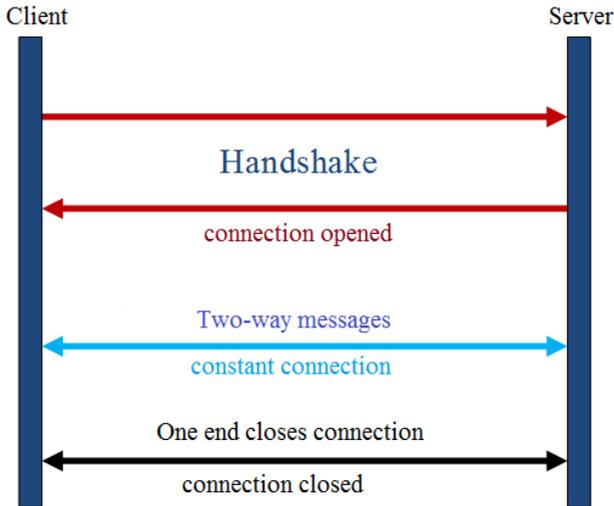


Figure 32.6 – Work principle of the communication protocol WebSocket

The data is packetized. The packets stand for the data type from the core-project in the JSON format. The packets are formed and parsed on both sides in the communication module.

The human-machine interface provides the driver with the information about the road situation, the driver’s state and the vehicle’s state. At the first start of the client application the registration form is displayed (fig. 32.7) where the driver needs to enter his personal data (nickname, age, sex).

Nick name

Age

Sex Male Female

Figure 32.7 – Registration form

The working area on the display is covered with the ground map (fig. 32.8). The current state and the direction of the vehicle is marked on the map with the help of the special arrow indicator. The map is to be centered

according to the current position. The position of other vehicles is displayed by means of arrows having different colours.

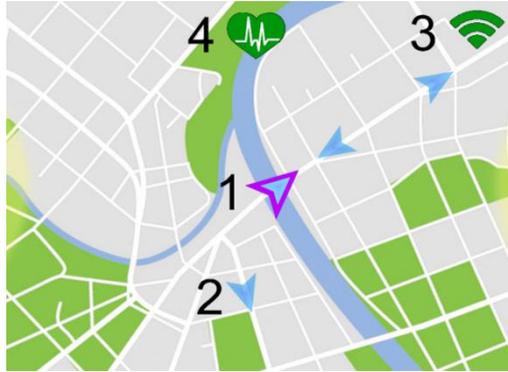


Figure 32.8 – Visual interface

1 – current position; 2 – other vehicle; 3 – server connection indicator;
4 – driver’s state indicator

The connection to the server is displayed by a special indicator. The indicator icon depending on whether there is a connection is shown at the figure 32.9.



Figure 32.9 – Server connection indicator:

1 - connection established, 2- no connection

The driver’s state is displayed by a special indicator. The indicator icon depending on the driver’s state is given at the figure 32.10.



Figure 32.10 – Driver’s state:

1 – good, 2 - poor

The HMI provides for the feature of manual signals to other drivers about the dangerous road stretch by pressing a button with the schematic representation of hazards types on a special board (fig. 32.11).



Figure 32.11 – Hazards menu

The speech recognition has been adopted in the HMI for the voice hazard signal transfer. The command for signal transfer consists of two fields: 1 - key phrase, 2 - hazard type. The key phrase should be brief and easy to pronounce. The possible key phrases are: “OK, motor”, “Go, machine” or simply “Danger”, “Danger ahead”. According to the survey results, the majority of the drivers prefer to set their own key phrases for the control commands

Hazards are indicated on the map with markers displaying the hazard type (table 32.3). The marker is coloured according to the hazard level (low – yellow, middle – orange, high - red).

Table 32.3 – Hazard level

Hazard type	Poor road	Ice condition	Fog	Caving	Reconditioning work	Poor driver’s state	Aggressive driver
Marker							

Map scale should be set according to the range of the lowest hazard level. When the hazard description is queried, an informative message with the enlarged hazard marker and the distance to the hazard object is displayed (fig. 32.12).



Figure 32.12 – Displaying of the markers and informative messages

A new hazard occurred is accompanied by the short voice signals. If the hazard level is high, the driver is informed by the voice messages communicating the hazard, for example: «Aggressive driver ahead, distance one hundred fifty meters, speed 90 kmph», «Fog in a hundred meters». Voice messages should repeat at a 10 second interval. The driver is provided with the possibility to query the hazard description using voice commands like “Voice the hazard”, “Describe the hazard”. The map scale can be configured, the voice and sound messages and volume level can be set or disabled (fig. 32.13).

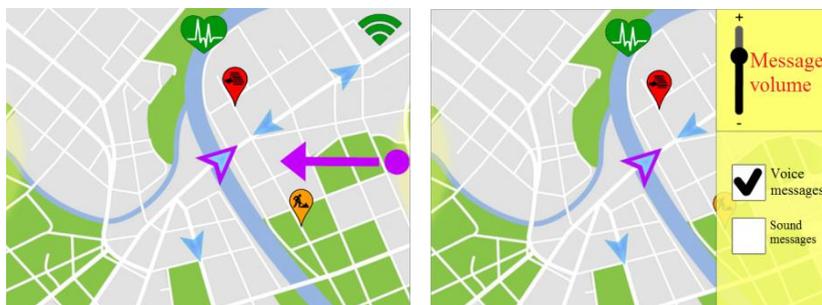


Figure 32.13 – Signal parameters setting

Display brightness and contrast should be adjusted to the daytime. The voice messages volume level is to be adjusted to the noise level in the car.

The overall picture of the road is at the driver's disposal. He can see the ground map, monitor other vehicles moving on a real-time basis. The driver's awareness is improved as the position of the cars undetected through the glass or by the mirror can be obtained. The blind spot issue is resolved. Due to the voice description of the hazards the cognitive load is reduced, the probability of the driver's distraction of the display is lowered.

The client HMI subsystem is implemented on OS Android. It consists of several modules that interact via the HMI control system (fig. 32.14).

The vehicle's sensors data is obtained from the board computer through the wired interfaces. The requests to the biosensors can be done through wireless interfaces. The current coordinates are obtained from the GPS receiver through the wire communication channel.

The module for communication is responsible for receiving and transmitting the messages to the server. The packets are formed by the client subsystem using the data models from the Core-project.

The interface of the application includes the following modules:

1. Visual interface responsible for displaying the following elements on the monitor:

- registration page (personal data filling);
- ground map, current position and position of other participants, hazard objects on the ground map;
- indicators of the driver's state and server connection;
- hazards panel;
- signals setup control panel.

2. Speech synthesizer responsible for voice warnings generation.

3. Speech recognition responsible for voice commands recognition.

4. Sounds management responsible for sound warnings.

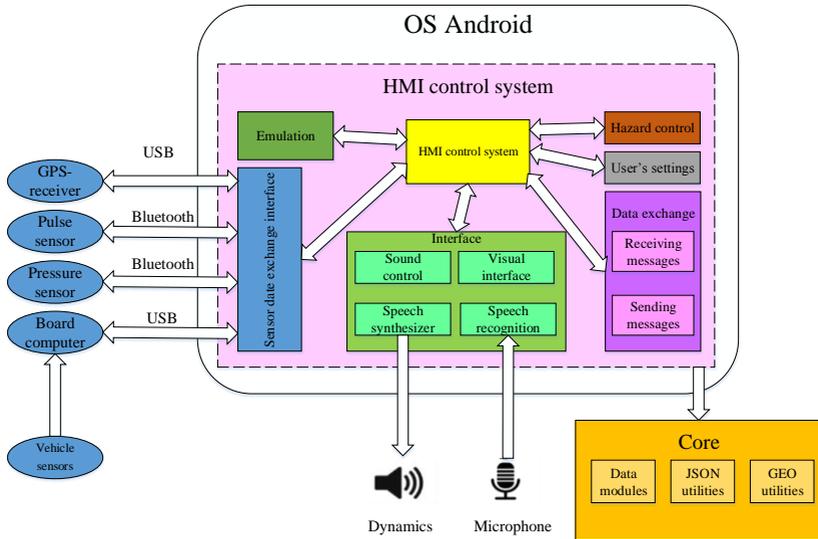


Figure 32.14 – Client HMI architecture

The HMI control system is responsible for the interaction with other modules in the system. It obtains the data from the sensor interfaces and transmits it to the communication module where packetizing takes place and the packets are sent to the server.

The user setup control module is responsible for the configuring and storing the personal data and parameters of the signals. The hazard control module is responsible for the refreshing of the hazards list provided by the server.

The emulation module is responsible for the emulation of the vehicle movement and the data obtained from the biosensors.

Figure 32.15 stands for the connection between the client, the server and the core modules.

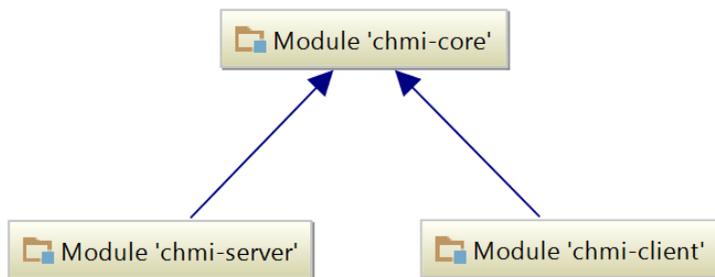


Figure 3.15 – Module structure of the system

The common classes for the client and the server applications are specified in the Core-project. It consists of the following packets:

1. Utils packet that contains 2 classes:
 - GeoUtils – the utilities for the geodata processing;
 - JsonUtils – the utilities for the work with JSON objects.
2. dto packet (Data Transfer Object) that includes the data models used to form the data transmitting packets.

The interface displays the road situation using the clear images. The driver can monitor the movement of the road users on the ground map (their current position, speed, direction). The ground map is associated with the environment, while the vehicles markers are associated with the real vehicles and the hazard markers are associated with the hazard objects. The colour of the markers allows the driver to identify the most dangerous objects. Using the voice warnings about the hazards and the voice commands for transmitting the signal the cognitive load is reduced as well as the driver's distraction of the monitor from the road is eliminated which in total reduces the risk of an accident on the road.

32.8 Human-machine interface assessment

Let us specify the operators to do all the interaction tasks for the HMI and build the model to assess it basing on the classical GOMS method and the assessment method for the automobile navigation systems according to the J2365 standard. The client HMI time indicators assessment model is given in the table 32.4. The table 32.5 displays the code of each operator and the duration for the young and elderly drivers.

Table 32.4 – GOMS method adaptation for the HMI

Operator	Classical GOMS method	GOMS method according to J2365, young / elderly	GOMS for CHMI Code, time
Mental psych-up	Mental operation M = 1.35 s	Mental operation M = 1.50 / 2.55 s	Ment M = 1.50 / 2.55 s
Compare the hazard description with the position on the map			
Assess the most dangerous objects on the map			
Reach the monitor with the hand	Movement D = 0.4 s	Reach far Rf = 0.45 / 0.77 s	Reach D = 0.45 / 0.77 s
Mark the position on the monitor with the finger	Mark Y = 1.1 s	Cursor once s1 = 0.80 / 1.36 s	Mark Y = 0.80 / 1.36 s
Find the marker on the map	-	Search S = 2.30 / 3.91 s	Search P = 2.30 / 3.91 s
Find the control element on the monitor	-	Search S = 2.30 / 3.91 s	
Press the object on the map	Press the key K = 0.2 s	Enter E = 1.2 / 2.04 s	Press H = 1.2 / 2.04 s
Press the button	Press the key K = 0.2 s	Enter E = 1.2 / 2.04 s	
Wait for a response from the	System response R	Response time of system-new menu Rm = 0.50 s	Interface response O = 0.1 s

Chapter 32 Resilient cooperative human-machine system

interface			
Say the key voice command aloud	-	-	Voice command G = 1.5 s
Say the signal voice command aloud	-	-	
Listen to the message about the hazard	-	-	Listen s = 3.5 s
React to the situation	-	-	Rection PE

Table 32.5 – Summary table of the operators

	Mental operation	Reach	Mark	Search	Press	Response	Voice command	Listen	Reaction
	M	D	Y	P	H	O	G	s	PE
Young	1.50	0.45	0.80	2.30	1.2	0.1	1.50	3.5	-
Elderly	2.55	0.77	1.36	3.91	2.04			-	

Identification of the tasks for the work with the human-machine interface:

- 1) Pass the signal about the dangerous road section via the monitor.
- 2) Pass the signal about the dangerous road section using the voice command.
- 3) Disable the sound signals.

- 4) Request the hazard description.
- 5) Learn the road situation.
- 6) Listen to the message about the hazard.

Models development and the calculation of tasks execution.

Task 1. Pass the signal about the dangerous road section via the monitor.

- 1) M – Mental psych-up – 1.50 / 2.55.
- 2) D – Reach the monitor with the hand – 0.45 / 0.77.
- 3) Y – Move the hand (finger) to the left hazard panel – 0.80 / 1.36.
- 4) H – Get the hazard panel out – 1.2 / 2.04.
- 5) Y – Move the finger to the button of interest – 0.80 / 1.36.
- 6) H – Press the hazard button – 1.2 / 2.04.
- 7) O – Wait for a response from the system – 0.1.

The sequence:

$$M + D + Y + H + P + Y + H + O$$

Time for the young people:

$$t_1 = 1.50 + 0.45 + 0.80 + 1.2 + 1.2 + 0.1 = 5.25 \text{ s.}$$

Time for the elderly people:

$$t_2 = 2.55 + 0.77 + 1.36 + 2.04 + 2.04 + 0.1 = 8.86 \text{ s.}$$

Task 2. Pass the signal about the dangerous road section using the voice command.

The sequence:

$$M + G + M + G + O$$

$$t_1 = 1.50 + 1.50 + 1.50 + 0.1 = 4.6 \text{ s.}$$

$$t_2 = 2.55 + 1.50 + 2.55 + 0.1 = 6.7 \text{ s.}$$

Task 3. Disable the sound signals.

The sequence:

$$M + D + Y + H + Y + H + O$$

$$t_1 = 1.50 + 0.45 + 0.80 + 1.2 + 0.8 + 1.2 + 0.1 = 6.05 \text{ s}$$

$$t_2 = 2.55 + 0.77 + 1.36 + 2.04 + 1.36 + 2.04 + 0.1 = 10.22 \text{ s}$$

The quantitative evaluation of the HMI shows that the hazard signal is passed more effectively using the voice commands. The execution of the signal setting task should be optimized through the voice commands, and thus less time will be spent by the driver. Additionally, we can conclude that the voice description requires more time compared to the situation of the driver executing the task of the ground map assessment. However, in this case the driver pays his attention to the map far more quickly which allows him to react to the situation faster.

32.9 Experimental research of cooperative human-machine interfaces

The experiment has been conducted in laboratory conditions basing on the emulation of the vehicles movement on the road from one point to another one. Experiment. Driver's state indication.

1) Starting state of the HMI. The identifier shows that the driver's state is good (fig. 32.16).

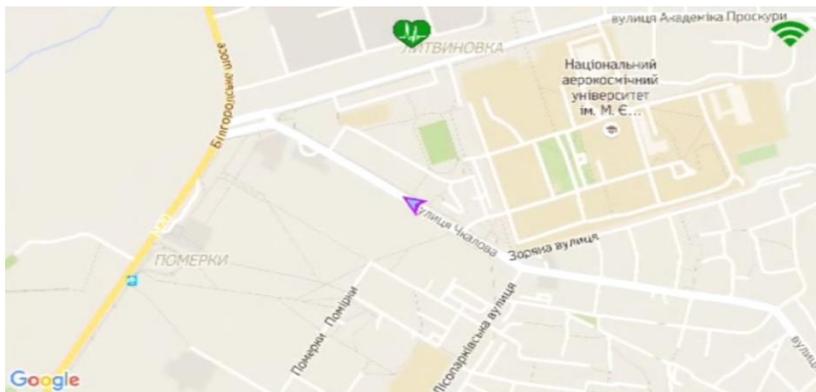


Figure 32.16 – Starting state of the HMI

2) Set the parameters of the driver. Experimental profile 1 (fig. 32.17):

Age = 25 years;

Pulse = 60;

Upper hypertension = 100;

Low hypertension = 50.

The expected result is poor driver's state, and the indicator coloured in red.

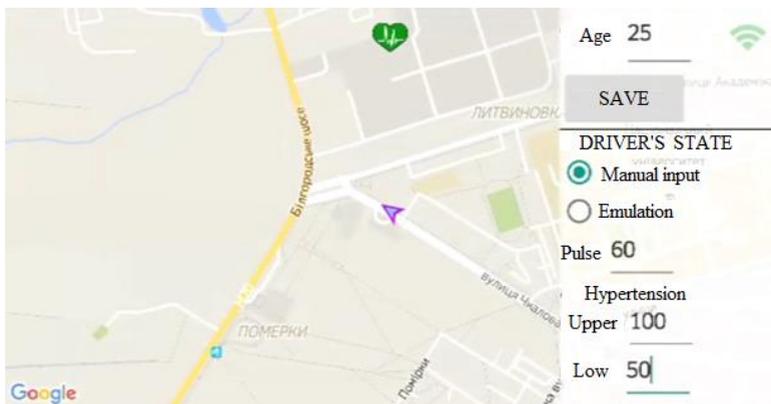


Figure 32.17 – Set the parameters of the driver. Profile 1

3) The result of setting the parameters (fig.32.18).
The indicator shows that the driver has poor state.

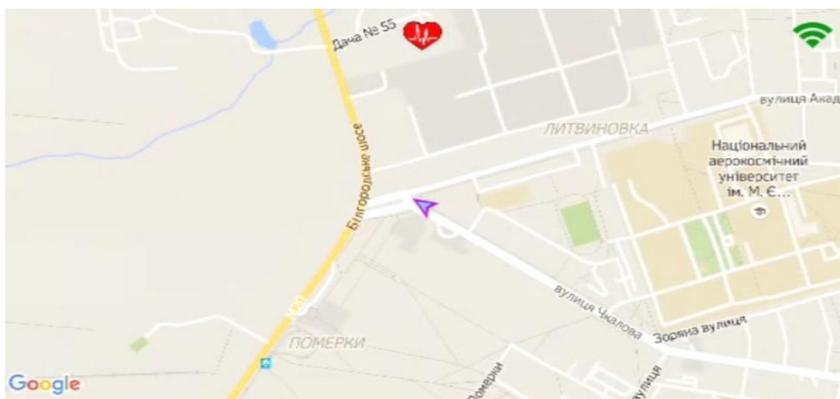


Figure 32.18 – Indication of the poor driver's state

4) Set the parameters of the driver. Experimental profile 2 (fig. 32.19):
Age = 25 years;
Pulse = 80;
Upper hypertension = 130;
Low hypertension = 80.

The expected result is the change of the driver's state indicator, the driver's state is good, and the state indicator gets coloured in green.

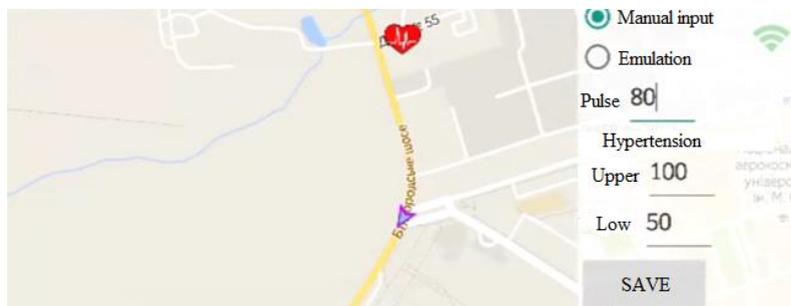


Figure 32.19 – Set the driver's parameters. Profile 2

- 5) The result of setting the parameters (fig.32.20).
The indicator shows that the driver has good state.

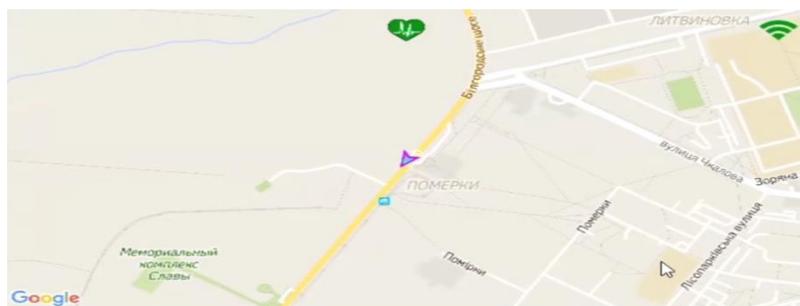


Figure 32.20 – Indication of good driver's state

Conclusion and self-control questions

Different vendors on IT market offer the advanced driver assistance systems. Such systems as a collision warning system, parking assistant, are designed for improvement of safety during the driving and reducing the driver's strain. One of the development lines of such systems is the improvement of the interaction between the driver and the vehicle control system "human-machine" and the provision information about the current situation on the road in real time for driver.

The provision this sort of information leads to an increase of situational awareness of vehicle driver. Great importance for enhancing of situational

awareness has issues for construction of secure dynamic human-machine interfaces .

The high amount of data used in the ITS, leads to the necessity of improvement of information access for all traffic participants. Improvement of situational awareness, risk assessment in the real-life improvement requires the use of large computing facilities for the storage, processing and analysis of data. These facilities are not always available, even for modern on-board computing equipment of vehicle.

Intelligent transport infrastructure includes complex of equipment that secures acquisition of the almost full information about the road situation and the possibility of a quick response to the changing conditions. To use such systems effectively one needs an HMI that maintains human-vehicle interaction and mitigates the negative errors impact on the safety, allows avoiding misinterpretation of the information that the system provides.

The idea of using cloud services in ITS is just beginning to gain popularity. The facilities of "clouds" can also affect the increase of transport safety.

The cooperative transport systems are of the utmost interest nowadays, and the cooperative HMI stand for one of the trends in this area.

Cooperative HMI provides the measure values of the parameters of vehicle and driver state in real time via the Internet into the "cloud." Here, the data from all the cars is dynamically processed and transmitted to motoring public.

The project PRORETA is a reserch in the area of the coopertaive HMIs. The research object is the prototype of the cooperative automobile HMI that implements the scenarios of preventing collisions at the cross-roads. One of the variants of cooperative HMI construction - is to use the technology of cloud computing.

Self-control questions and tasks

1. What does intelligent transport system stand for?
2. What are the trends in driver assistance systems?
3. Please give examples of driver assistance systems.
4. How can the driver's situation awareness be increased?
5. What is the effect of increased driver's situation awareness?
6. Please name the methods used for dynamical risk assessment.
7. What are the types of connection between the transport systems and the infrastructure?
8. What does intelligent transport infrastructure stand for?
9. What are the components of the intelligent transport system infrastructure?

References

1. Stepanov V. Organization of traffic. Intelligent transport and two main troubles”/ V. Stepanov // Haulier. – 2009. №9 (108) [Electronic resource]. Mode of access: [ap-st.ru/ru/filling/y-2009.n-9.oid-425.html].
2. Opel and project URBAN: improvement of safety and cost effectiveness on moving in cities [Electronic resource]. Mode of access: [www.opel.ru/experience/ob-opel/novosti opel]. – 2014.
3. What technical innovations has Volvo implemented for last 10 years? [Electronic resource]. Mode of access: [www.autoconsulting.com.ua/news]. – 2011.
4. Toyota Motor Corporation» presents new systems of vehicle safety [Electronic resource]. Mode of access: [www.major-toyota.ru/news.html]. – 2013.
5. Orekhova A. Information technology of I&C systems human machine interfaces safety assessment / A. Orekhova // Information processing systems. 2013. Vol. 1 (108). – P. 267-271.
6. Orekhova A. Safety case-oriented assessment of human-machine interface for NPP I&C system / A. Orekhova, V. Kharchenko, V. Tilinskiy // Reliability: Theory & Applications. – 2012. – Vol. 3 (26). – P. 27 – 38.
7. Levenson N. Systems-Theoretic Accident Modeling and Processes (STAMP) / N. Levenson // Safety Science. – 2008. – Vol. 4. – P. 237-270.
8. Simoncic M. A Bayesian Network Model of Two-Car Accidents / M. Simoncic // Journal of Transportation and Statistics. – 2004. – Vol. 7, №. 2,3. – P. 13-27.
9. Monteil J. Cooperative highway traffic: multi-agent modeling and robustness assessment to local perturbations / J. Monteil, etc. // Proceedings of 92 Annual Meeting of the Transportation Research Board. – 2013.
10. Charissis V. Artificial Intelligence Rationalefor Autonomous Vehicle Agents Behaviour in Driving Simulation Environment / V. Charissis, etc., // Advances in Robotics, Automation and Control. – 2010. – P. 472.
11. Brezhnev E. Dynamical and Hierarchical Criticality Matrixes-Based Analysis of Power Grid Safety / E. Brezhnev, V. Kharchenko, etc. // Proceedings of ANS PSA International Topical Meeting on Probabilistic Safety Assessment and Analysis, Wilmington. – 2011. – P.1137-1149.
12. Cars In The Future: Human Machine Interface [Electronic resource]. Mode of access: [<http://www.rospa.com/roadsafety/policy/carsinthefuture/human-machine-interface.aspx>].
13. European Statement of Principles, European Statement of Principles for in-vehicle information and communication systems. - 1999.
14. HASTE, Deliverable 4 –Recommended Methodology for a preliminary safety analysis of the HMI of an IVIS. – 2005.

15. Stoltzfus J. Stoltzfus, Cloud Computing for Vehicles: Tomorrow's High-Tech Car / J. Stoltzfus, [Electronic resource]. Mode of access: [http://www.techopedia.com/2/28137/trends/cloud-computing/cloud-computing-for-vehicles-tomorrows-high-tech-car]. - 2012.

16. Cloud Computing can Reengineer the Car Interiors [Electronic resource]. Mode of access: [http://www.cbrdigital.com/2012/01/16/cloud-computing-can-reengineer-the-car-interiors.html] 2012

17. Lynn Walford. Volvo New Connected Car Features-Magnets, Real-Time Cloud Road Data & Driver Sensing [Electronic resource]. Mode of access: <http://www.autoconnectedcar.com/2014/03/volvo-new-connected-car-features-magnets-real-time-cloud-road-data-driver-sensing/>, 2014.

18. Michael Sheehan. Cloud Computing Cars and Mobile Devices [Electronic resource]. Mode of access: [http://scoop.intel.com/cloud-computing-cars-and-mobile-devices/], 2011.

19. Toyota and Panasonic develop cloud service to connect cars and household appliances [Electronic resource]. Mode of access: [http://panasonic.ru/press_center/news/detail/464204] 2014.

20. Anokhin A. Adaptive human-system interface for control of complex systems (in application to nuclear power plant) / A. Anokhin, E. Marshall // Book of abstracts of the 21st European Meeting on Cybernetics and System Researches, EMCSR 2012, Vienna, Austria, 10-13 April. – 2012. – P. 185-188.

33 HUMAN AUTHENTICATION AND BIOMETRY IDENTIFICATION FOR SECURITY

Manager of mission critical system don't know what his employees are doing at working time. At a minimum, they're probably goofing off watching YouTube videos. At worst, they could be steering system to damage or company toward financial ruin. In this chapter we'll see how to keep an eye on person (employee) use informational and technical resources and monitor just about everything else they do with their working PCs.

We can already hear the groans of disgruntled readers as we type these words (and if somebody worried about privacy at work, he has to remember that sometimes human life depends on results of operators actions). But gone are the days when PC monitoring was an optional, draconian security measure practiced only by especially vigilant organizations. Today, more than three-quarters of U.S. companies monitor employee Internet use. If some business is in the remaining quarter that doesn't do so, they're probably overdue for a policy change.

The reason to monitor mission critical system operator's work is evident to investigate potential accidents if the would happen [1]. Everything operators team does on company time and on company resources matters. Also time spent on frivolous Websites can seriously hamper productivity, and visiting objectionable sites on company PCs can subject work-flow problems on mission-critical system and can cause serious legal and security risks, including costly harassment suits from staffers who may be exposed to offensive content or download some trojans, viruses, exploits, rootkits or similar dangerous code.

If we talk about business or military organization. Other consequences may be far worse than mere productivity loss or a little legal hot water. Either unintentionally or maliciously, employees can reveal proprietary information, jeopardizing business strategy, customer confidentiality, data integrity, and more.

In chapter 31 Human-machine interface models were mentioned. We can model operators work [2] to customize and adopt system to its specific needs. Modeling allows to predict and prevent some types of human factors in computer security. To do so we need an internal representation of the system's operator. System's operator modeling is the subdivision of human-computer interaction which describes the process of building up and modifying a conceptual understanding of the user. Another common purpose is modeling specific kinds of system's operator, including modeling of their skills and declarative knowledge, for use in automatic software-tests. In our case we use models to predict user behavior based on his or her skills and declarative

knowledge. According to [2] there are two types of models: descriptive and predictive.

Predictive models, sometimes called engineering models or performance models, are widely used in many disciplines. In human-computer interaction, predictive models allow metrics of human performance to be determined analytically without undertaking time-consuming and resource-intensive experiments. Predictions so generated are a priori: they allow a design scenario to be explored hypothetically without implementing a real system and gathering the same performance metrics through direct observation on real users.

First example of predictive model is the Hick-Hyman law for choice reaction time. This law takes the form of a prediction equation. Given a set of n stimuli, associated one-for-one with n responses, the time to react (RT) to the onset of a stimulus and make the appropriate response is given by:

$$RT = a + b \cdot \log_2 n \quad (33.1)$$

where a and b are empirically determined constants. The Hick-Hyman law has surfaced in a few contexts in interactive systems. One of examples is of a telephone operator selecting among ten buttons when the light behind a button comes on. More recently, we have found the Hick-Hyman law useful in predicting text entry rates on soft keyboards with non-Qwerty layouts. For non-Qwerty layouts, users must visually scan the keyboard to find the desired letter. The act of finding the desired letter among a set of randomly positioned letters is appropriately modeled by the relationship in Equation 33.1.

More close to system's operator's work is predictive model, especially keystroke-level model (KLM). This model was developed as a practical design tool, the goal being to predict the time to accomplish a task on a computer system. The model predicts expert error-free task completion times, given the following input parameters:

- a task or series of sub-tasks;
- method used;
- command language of the system;
- motor skill parameters of the user;
- response time parameters of the system.

A KLM prediction is the sum of the sub-task times and the required overhead. The model includes four motor-control operators (K = key stroking, P = pointing, H = homing, D = drawing), one mental operator (M), and one system response operator (R):

$$T_{EXECUTE} = t_K + t_P + t_H + t_D + t_M + t_R \quad (33.2)$$

Some of the operations above are omitted or repeated, depending on the task. For example, if a task requires n keystrokes, t_K becomes $n \times t_K$. Each t_K operation is assigned a value according to the skill of the user, with values ranging from $t_K = 0.08$ for highly skilled typists to $t_K = 1.20$ s for a typist working with an unfamiliar keyboard. The pointing operator, t_P , is based on Fitts' law. As we can see KLM allows predicting all types of operator's tasks in complicated HCI.

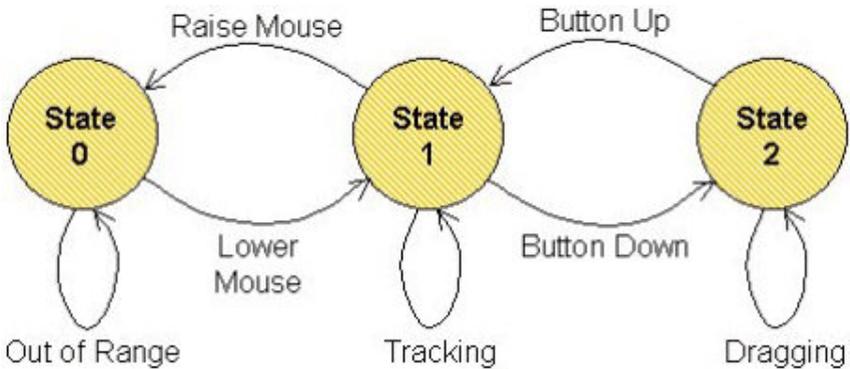


Figure 33.1 - Buxton's 3-state model of graphical input

Descriptive models provide a framework or context for thinking about or describing a problem or situation. Often the framework is little more than a verbal or graphic articulation of categories or identifiable features in an interface. The most bright example of descriptive model is describing keyboard with key-action model (KAM) where keyboard keys are categorized as either symbol keys (letters, numbers, or punctuation symbols), executive keys (ENTER, F1, or ESC), or modifier keys (SHIFT or ALT). According to this model we can describe/design any keyboard or operators manual board.

Another example of descriptive model is Buxton's 3-state model for graphical input devices. This model allows describe user actions with input

devices and was successfully used by Apple, IBM and Toshiba while designing new input devices for their laptops.

Most widely used of these two model types are: predictive model - Fitts' model of the information processing capability of the human motor system and descriptive model - Guiard's model of bimanual control. Fitts' model is a mathematical expression emerging from the rigors of probability theory. It is a predictive model at the mathematical end of the continuum, to be sure, yet when applied as a model of human movement it has characteristics of a metaphor. Guiard's model emerged from a detailed analysis of how human's use their hands in everyday tasks, such as writing, drawing, playing a sport, or manipulating objects. It is a descriptive model, lacking in mathematical rigor but rich in expressive power. Today, both models are commonly used in the research and development of interactive systems. The field combines work in other disciplines, most notably psychology, cognitive science, and sociology. Fitts' and Guiard's models emerged from basic research in an area within experimental psychology known as psychomotor behaviour or, simply, motor control. Also there are more complicated stochastic models [3] to model user behavior. But as shown above [2] we can completely describe and predict system's operator work in normal circumstances.

33.1.1 Endpoint security

Employee monitoring is just one facet of a larger discipline known as endpoint security, which includes everything from malware protection to policy enforcement and asset tracking. Large enterprise computing environments demand comprehensive endpoint-security systems, consisting of server software coupled with client software on each user's machine, that can handle many of these functions at once. These systems can be complex enough to require the experienced IT security expert. But also they can be simpler and designed for smaller organizations or distributed company departments.

For a small business there are several good ways to achieve endpoint security. We can install a Web-hosted system that combines software on the PC with remote monitoring services to protect your computers and enforce compliance with organization policies. We can combine a few complementary tools, such as a desktop security suite and professional tracking software (Trend Micro Worry-Free or Awareness Technologies, InterGuard Sonar). Or, if our organization or budget is tight we can adopt free (and opensource) tools like ActivTrak.

Functionality of endpoint security software (or employee monitoring software) records and controls all operator's computer activity, web filtering

solution blocks any category of website or remote resources, identifies and blocks dangerous activities, block executions of some software and etc.

System's operator monitoring ought to be just one small component in a comprehensive strategy to protect information subsystem of mission-critical system. Once we've made the choice to monitor, we should follow these best practices in endpoint security.

Be straight and clear with operators: Nobody likes being spied on unwittingly. Unless we think someone on your team poses a serious threat that requires covert monitoring, it's best to be up front with staffers about what is tracked and why. Many organizations accomplish this with a simple statement in the operator's handbook telling workers plainly that everything they do on company computers, including individual keystrokes, can and will be tracked. Letting operators know that their behavior is being monitored can serve as a powerful deterrent against unwanted on-line activity.

Filter proactively: Most good endpoint-security tools include Web and e-mail content filters that can block inappropriate sites and prevent users from sending or receiving files that can jeopardize work flow. By limiting the ways staffers can get into trouble, we can prevent problems up front.

Check reports regularly: There's no sense in generating usage reports if no one is going to look at them. The reports that monitoring software generates allows identify potential problems early and take remedial action.

33.1.2 Information security and types of human factor errors

According to the 2014 IBM Chief Information Security Officer Report, 95 percent of information security incidents caused by human factor errors. Human factor errors is a key factor in mission critical systems, such as aviation accidents and in medical errors.

Human factor errors can be defined as circumstances in which planned actions, decisions or behaviors reduce — or have the potential to reduce — quality, safety and security. Human factor errors usually are results of ignoring formal security policy and involved in information security include the following:

- system misconfiguration;
- poor patch management;
- usage of default logins and passwords or easy-to-guess passwords;
- lost devices;
- connecting computers to the Internet through an insecure wireless network;
- disclosure of information via an incorrect email address;

- double-clicking on an unsafe URL or attachment;
- sharing passwords with others;
- reusing the same password and logins on different websites;
- leaving computers unattended when outside the workplace;
- using personally owned mobile devices that connect to the organization's network.

And the most dangerous – possibility of susceptibility to social-engineering attacks [4]. For example the most famous hacker Kevin Mitnick covers social engineering in his book «The Art of Deception». Part of the book is composed of real stories, and examples of how social engineering can be combined with hacking.

Human-factor engineers in aviation assume that serious incidents are not caused by just one human error, but by an unfortunate alignment of several individual events. Incidents happen when a series of minor events occur consecutively and/or concurrently.

Organizations apply a variety of strategies to secure mission-critical systems. Many of these are based on formal security policies rules and some additional meanings. Some well-known examples include the following:

- prevention strategy approaches to support someone in the correct execution of tasks, such as checklists, awareness campaigns, procedures, disciplinary measures, training and retraining;
- eliminating strategies that make it impossible for system users to make a mistake, e.g. usage of automated safeguards such as cryptography, password management, identity and access management, network access rules and automatic standby locks;
- mitigation strategy to mitigate the consequences of errors by making sure detection mechanisms are in place to correct situations before they become an incident, e.g. audits, internal control, breach and intrusion detection solutions, system monitoring and surveillance.

Aviation and health care industries support a holistic error prevention approach to change conditions in the organization, the environment and the systems that people work with. These systemic (socio-technical) strategies could be of great benefit to information security and mission-critical systems.

Among them is crew resource management (CRM) is a training program developed for airline crews to learn how to manage and behave during an incident. CRM training encompasses communication, situational awareness, problem-solving, decision-making and teamwork. The application of CRM in health care and aviation has proven to significantly reduce errors. When applying this method to information security, it is important to recognize that

humans are strongest links in times of crisis. Security incidents will happen, and staff should be trained to recognize and contain them.

Decades' worth of data from aviation incident reporting systems have been effectively used to redesign aircraft, air traffic control systems, airports and pilot training. Information security specialists should also keep analyzing security incidents and near misses. Without such analysis, there is no way to uncover recurring errors. Investigations should target the people involved, the team, the workplace, the organization, third parties and the information and communications technology systems. The important issue is not who blundered, but how and why the incident occurred.

It is human to make errors, and they can never be 100 percent prevented. A mixture of strategies may help to prevent human errors from turning into security incidents. Successes in human error reduction in aviation give hope, while studies of medical errors provide valuable insight [5].

33.1.3 Threats in human-machine interaction

In previous chapter were described most often operators errors which can cause real security **threats**. According to «Federal Information Processing Standards (FIPS) 200, Minimum Security Requirements for Federal Information and Information Systems» by NIST of United States of America, threat definition is «Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service. Also, the potential for a threat-source to successfully exploit a particular information system vulnerability». According to MSDN [6] human plays great role in security threats (fig. 33.2)

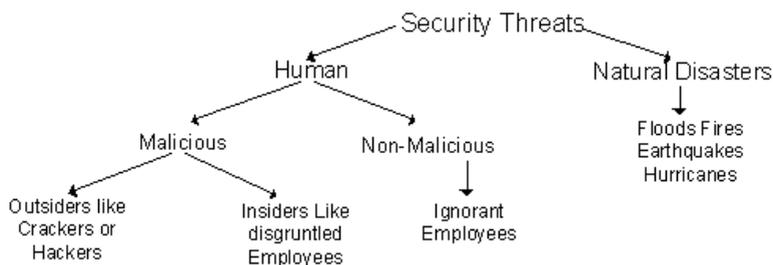


Figure 33.2 - Human threats in security threats classification

As we can see attackers are not the only ones who can harm an organization. The most dangerous attackers are usually inside users (operators), because they know many of the inside information, like passwords and security measures that are already in place. Sometimes insiders can have specific goals and objectives, and have legitimate access to the system. Also system's operators are the peoples most familiar with the organization's computers and applications, and they are most likely to know what actions are the most harmful and might cause the most damage (plant viruses, Trojan horses, worms, rootkits and they can freely browse through the file system.

The insider attack can affect all components of computer security. By browsing through a system, confidential information could be revealed. Trojan horses are a threat to both the integrity and confidentiality of information in the system. Insider attacks can affect availability by overloading the system's processing or storage capacity, or by causing the system to crash.

The primary threat to data integrity comes from authorized users who are not aware of the actions they are performing. Errors and omissions can cause valuable data to be lost, damaged, or altered. Non-malicious threats usually come from employees who are untrained in computers and are unaware of security threats and vulnerabilities. Users, data entry clerks, system operators, and programmers frequently make unintentional errors that contribute to security problems, directly and indirectly. Sometimes the error is the threat, such as a data entry error or a programming error that crashes a system. In other cases, errors create vulnerabilities. Errors can occur in all phases of the system life cycle.

For example in 1996, a laptop computer was stolen from an employee of Visa International that contained 314,000 credit card accounts. The total cost to Visa for just canceling the numbers and replacing the cards was \$6 million.

33.2 Basic principles of authentication, authorization and accounting in information systems. Access control in IS.

Authentication, authorization, and accounting (AAA) are terms for a framework for access control to computer system and information resources. These terms include policies, audit and provide the information necessary to analyze operators activity (partially provide endpoint security) or to bill for services in service providers networks. These processes are very important for safe and resilient work of computer system. They prevent unauthorized access.

Authentication provides a way of user identification by using one of authentication method. Usually by having the user enter a valid user name and valid password before access is granted. It is the simplest method which does

not guarantee operator's presence. The process of authentication is based on each operator having a unique set of criteria for gaining access. The AAA server compares a operator's authentication credentials with other operators credentials stored in a database. If the credentials match, the operator is granted access to the network. If the credentials are not valid, authentication fails and system access is denied.

After authentication, an operator must gain **authorization** for carrying tasks in system. The authorization process determines whether the operator has the authority to carrying such commands. Authorization is the process of enforcing policies: determining what types or qualities of activities, resources, or services an operator is permitted. Authorization occurs within the context of authentication: once an operator is authenticated, he may be authorized for different types of access or activity. Also an operator can have additional policies for additional tasks through additional authentication and authorization for some extra activities, e.g. command «sudo» in Linux or «run as...» in Windows.

Accounting measures the resources a user consumes or activities which have been done during access. Accounting of operator's work is carried out by logging mechanism and is a part of endpoint security which mentioned above.

Authentication, authorization, and accounting services are often provided by a dedicated AAA server with integrated endpoint security subsystem. In billing systems which account user's consumed resources often used a standard by which network access servers interface with the AAA server is the Remote Authentication Dial-In User Service (RADIUS).

AAA is a part of **access control system** and provide security technique that can be used to regulate who/what can view/use resources in a computing environment. Access control subdivided into physical access control (limits access to buildings, rooms and physical assets) and logical access control (limits usage of computer systems, networks, RDBMS, files and data). Usually access control system are based on access lists, attributes (attribute based access control) or on roles (role-based access control). Role-based access control (RBAC) is used by the majority of organizations with more than 500 employees and can implement mandatory access control (MAC) or discretionary access control (DAC). These mechanisms are provided by operating system policies (e.g. SELinux, grsecurity, Windows Security Policies).

33.2.1 Human factors in user authentication

Theoretically [1] in mission-critical systems human factor errors are not allowed, but in deed these errors cause main part of all accidents.

So after incident there are investigations to find a scapegoat for every mishap, the failure depends from a faulty action made by a liable person. In this way, mistakes tend to be covered-up, everyone keeps relevant information about safety for him/herself and inevitably an accident will happen.

In the systemic approach, organizations are aware that to err is human. Thus, they try to limit the scope and the severity of errors via a thorough analysis of incidents/accidents, disseminating all the useful information about threats and implementing departments and areas entirely dedicated to safety with the task to monitor even minor events. In this safety conception, the human contribution is essential and represents the main resource to ensure a high safety level.

So, the human judgment remains the last barrier against accidents, the sole “device” that is able to adapt the rule to the operation in progress or even to deviate from it, having deemed the adaption and the deviation safer than the blind execution of the standard task. At the moment, machines haven't this level of judgment.

Resilience Engineering [1] is a new complex way to approaching safety and it is based on the following paradigm: incidents, crashes and fatalities occur out of the same reasons why we can predict many factors and can theoretically guarantee a good level of safety. We cannot underestimate the importance of weak signals emerging from the daily activities to better understand how to improve all the safety levels in high complexity mission-critical systems. The main feature of complex systems is their dynamic stability/instability equilibrium; in short, we must move to be balanced. Resilience is what makes these systems robust and, at the same time, flexible. The capability of the organization to create adequate risk models and to correctly use the resources in a proactive manner, creating a good synergy, is what makes Resilience Engineering able to face any disturbing input to the normal operations and properly manage the economical resources.

According to the main thought of Resilience Engineering, incidents and accidents do not come from system flaws or individual mistakes, but from the lack of ability of the complex system to adapt its framework to the changed complex environment.

On figure 33.3 are shown basic resilient system components which influence on human-machine interaction: mission-critical system operator's good behavior comes from a base of mental/physical health on which technical/non-technical skills and positive attitude are built (e.g. for nuclear power station's main operator these are the minimum requirements they have to conform everyday at work in order to safely operate with another team members and proactively prevents incidents/accidents).

As we see very important to diagnose good behavior and psychological state of responsible person, because it can influence the outcome of critical operations and the future of the resilient system. It is very important not only authenticate user but test readiness for carrying out his tasks.

Nowadays authentication task can be solved in three ways:

- a person's possession of some object: smart-card, RFID-card or another type of e-token;
- a person's knowledge of a piece of information: password or PIN-code;
- a person's unique physiological attributes or subconscious activities (signature processing, speech, keyboard blind-typing).

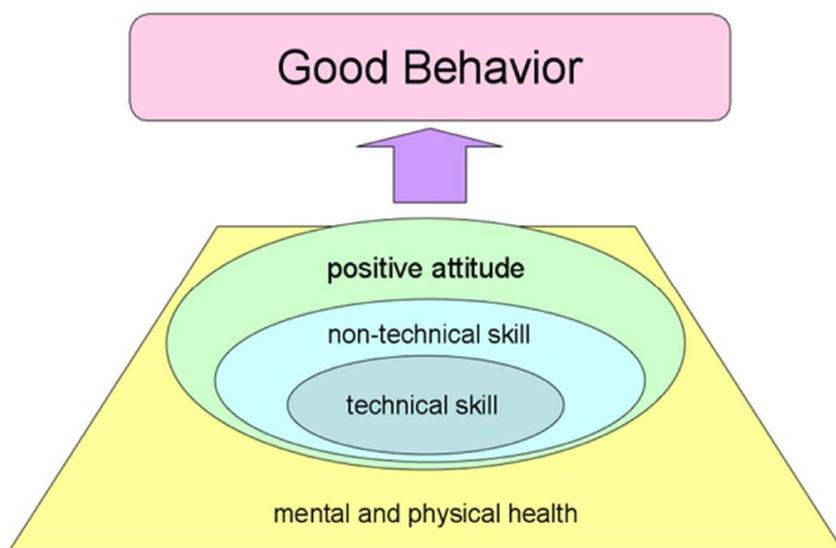


Figure 33.3 - Basic resilient system components in human aspect ■

First and second approaches can be exploited and once the person has lost control of their identifying possession then an unauthorized person can use it for fraudulent activities. Third approach is biometrics and its primary advantage is that it cannot be stolen, misplaced or forgotten. This method based on physiological or behavioral person's characteristics. And biometric authentication based on behavioral characteristics allows to measure good behavior.

Also mission-critical system environment makes difficult for a operator to deal with all these adjustments and accomplish his/her tasks. The solution to it lies in the Ergonomics, which assist the operator in his/her adaptation to unfriendly environments. Ergonomics is the branch of science that searches the best way to:

- build user-friendly environments, where operators have to work;
- modify tools that operators have to manipulate in order to carry out correct assignments.

So biometry authentication system have to conform ergonomic requirements, be acceptable and allows to identify good behavior.

It is essential that biometric technologies, are not the panacea to security and identification issues. To obtain the highest level of security, biometric technologies need to be part of a broader and complete system that incorporates multiple security technologies.

33.2.2 Unique identifying characteristics to authenticate user

In context of biometry authentication few tasks are arose: the registration, storage, protection, issuance, and assurance of a user's personal identifier(s) and privilege(s) in an electronic environment in a secure, efficient, and cost-effective manner. All these tasks are belong to subject area of identity management (fig. 33.4).

At higher level, biometric systems are pattern recognition systems that use different types of sensors, cameras or scanners (image-acquisition devices) to measure physiological or behavior characteristic of a person. Type of such devices depends on application area: scanners or cameras in the case of fingerprint, retina or iris recognition; microphones in the case of voice recognition; tablet or touch-screen in the case of signature recognition. All these devices allow to obtain the biometric patterns or characteristics. The acquired biometric characteristics considered as distinctive between different users and stable for each user. «Biometric passwords» are extracted and encoded into a biometric reference that is a mathematical representation of a person's «biometric password» (like a hash-code of stored password). General scheme of authentication system is presented on figure 33.5, detailed scheme shown on figure 33.6.

Each biometric device and biometry authentication system have their own operating methodology, there are some generalizations that can be made as to what typically happens within a biometric system implementation.

Before verification of a person's identity via a biometry authentication system, must be created a biometric template or «biometric password». This «biometric password» used to compare with pattern provided by person during

verification. For some biometric technologies, a number of «biometric passwords» are created to guarantee high level of accuracy. The template is then referenced with an identifier (PIN, password or another «biometric password»). The successful person's identification during the enrollment procedure and quality of the resultant «biometric passwords» are critical factors in the overall success of a biometric application. A poor quality «biometric passwords» can cause problems for the authenticated person, and often resulting in re-enrollment.

These «biometric passwords» are stored in a file or a database, on a smart card or other token. Biometric systems are automated by hardware and software, allowing for fast, real-time decision making in identification situations.

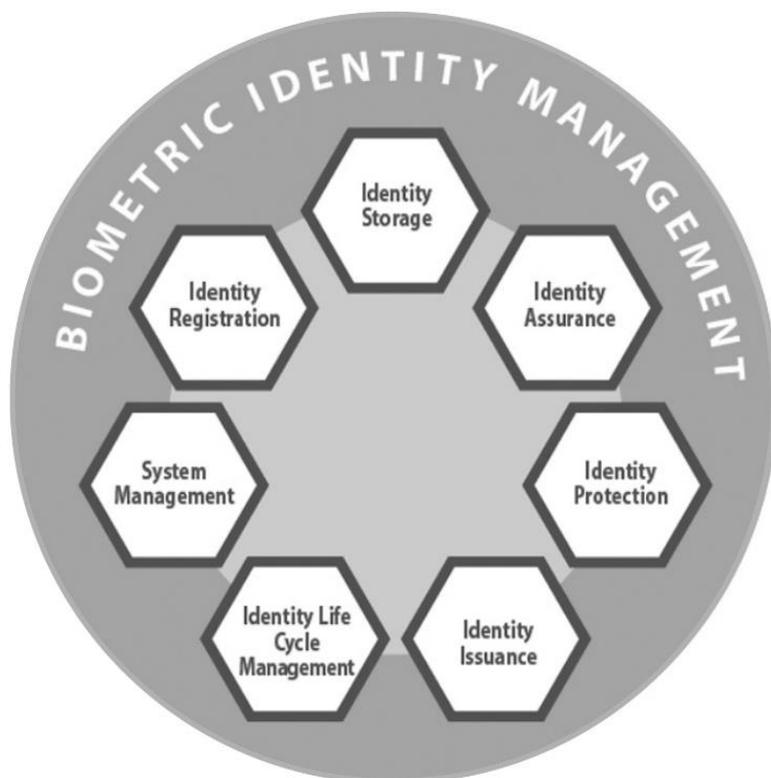


Figure 33.4 - A complete biometry identity management system ■■■

Possible «biometric passwords» storage options include:

1. Store the template within the biometric reader device or PC.
2. Store the template remotely in a central repository.
3. Store the template on a portable token or media, such as a smart card.

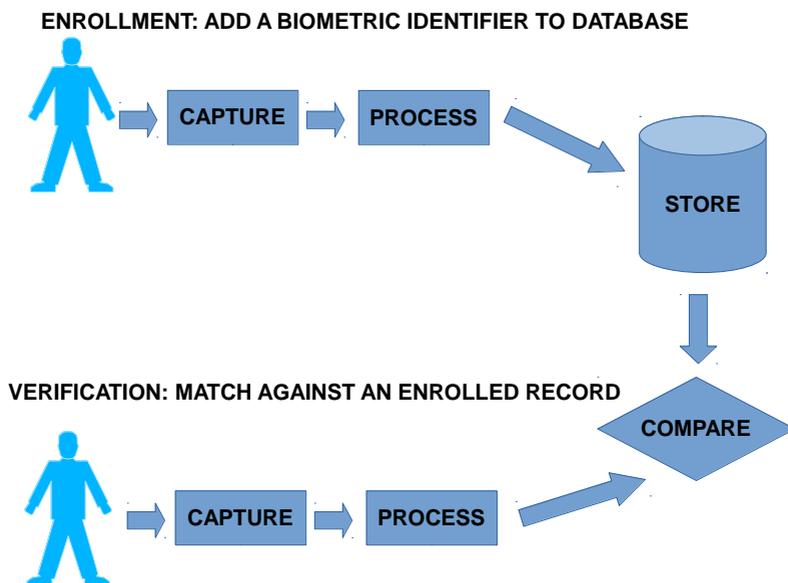


Figure 33.5 - Processes in biometric authentication system

While implementing biometric authentication system owner must take into account personal biometric characteristics criteria. Any human biological or behavioral characteristics can become a biometric identifier and have to conform the following properties:

Universality: Every person should have this characteristic. Exceptions to this rule are: mute people, people without fingers, persons with injured eyes. These exceptions must be taken into account through “work-arounds” such as conventional non-biometric authentication processes. Biometric device have to allow a secure override if a physical property is not available and allows to enter a password, PIN, secure token or enter another "biometric password".

Distinctiveness: Two people must not have identical biometric characteristics. For example monozygotic twins, cannot be distinguished by

face recognition and/or DNA-analysis systems, but they can be distinguished by fingerprints or iris patterns.

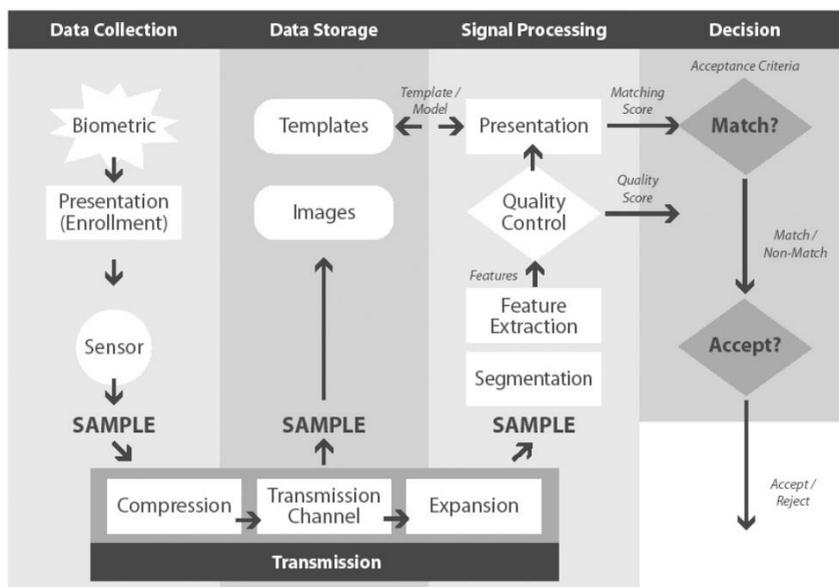


Figure 33.6 - Detailed scheme of a biometric-based system

Permanence: The characteristics should not vary or change with time. A person's face changes significantly with aging and a person's signature and its dynamics may change as well. In this case it requires periodic re-enrollment.

Collectible: Obtaining and measuring the «biometric password» should be easy, non-intrusive, reliable, and robust, as well as cost effective for the application.

33.2.3 Authentication, authorization and accounting operators activity in modern information systems

Biometric authentication system have to conform the following criteria:

Performance: includes the accuracy, resources, and environmental conditions required to achieve the desired results.

Fraud: how difficult it is to fool the system by fraudulent means. An automated access control system that can be easily fooled with a fingerprint

prosthetic or a person's retina photography stuck on a glasses does not provide enough security level.

Acceptability: Person should accept this method of authentication (social acceptance). For example fingerprint authentication associated with crime investigations.

33.3 Biometry authentication techniques

Biometric technologies are the science of detecting and recognizing human characteristics by measuring and analyzing biological data using various electronic technologies. Biometric technologies allow to reduce influence of human factor errors in authentication.

There are different types of biometric technologies integrated in inexpensive consumer devices: Apple iPhone, Samsung Galaxy, Laptops, Tablets etc. Using biometric technologies is a quick and efficient way to log in without the need to remember a password and it also identify person presence. Also id-cards (smart, RFID) can get stolen or lost, passwords and PIN numbers forgotten or shared, but biometry characteristics of a person is hard to fraud and/or stole. Biometric Technologies reduce threats to privacy, security and personal safety. Buildings, airports, schools, universities and mission-critical systems are using biometrics to ensure only authorized personnel are able to gain access. Biometric systems use a variety of physical and behavioral characteristics obtained from an individual to establish identity.

Some systems are multi-modal and using more than one biometric characteristic to detect person's identity. It can improve the accuracy of the biometric system.

Biometrics is based on the measurements of biological and/or behavioral distinctive characteristics. Physical biometrics directly read/measures characteristics of the human body: fingerprint, face, iris, hand, retina, face vein pattern, picture of pinna, DNA.

Behavioral biometric techniques are based on voice recognition, on-line signatures, keyboard strokes. Behavioral human body characteristics are measured indirectly.

Biometric systems are composed with endpoint security components. The biometric system involves the exchange of biometric data and information and comprises of 2 phases:

Phase 1: Enrollment phase and identification: the biometric characteristics of an individual are scanned by the biometric reader to create a feature set or template («biometric password»), often during the registration process.

Phase 2a: Identification: involves checking the individual's biometric against a larger database or watch list of individuals (1:1 matching).

Phase 2b: Verification: compares the individual with a template already stored on a system (1:n matching).

Popularity of biometry authentication systems cause grows of manufacturers and systems. To unify biometric authentication system's developed a number of International, US, and European Standards [8,9], e.g.:

CEN/TS 16428:2012 – 2012-10-24 Biometrics Interoperability profiles - Best Practices for slap tenprint captures

CEN/TS 16634:2014 – 2014-04-09 Personal identification - Recommendations for using biometrics in European Automated Border Control

CEN/TS 16920:2016 – 2016-03-30 Environmental influence testing methodology for operational deployments of European ABC systems

CEN/TS 16921:2016 – 2016-03-30 Personal identification - Borders and law enforcement application profiles for mobile biometric identification systems.

For example International standard ISO/IEC JTC 1/SC 37 [9] was created in 2002 and main scope of it is to support interoperability and data interchange among applications and systems. Generic human biometric standards include: common file frameworks; biometric application programming interfaces; biometric data interchange formats; related biometric profiles; application of evaluation criteria to biometric technologies; methodologies for performance testing and reporting and cross jurisdictional and societal aspects.

For purposes of computer engineering (in software and hardware systems development) very important is implementation of BioAPI. BioAPI v2.0 (ISO/IEC 19784-1), developed by the Bio-API Consortium and released in May of 2006, was designed to produce a standard biometric API aiding developers and consumers. In general BioAPI (Biometric Application Programming Interface) is a key part of the International Standards that support systems that perform biometric enrollment and verification (or identification). It defines interfaces between modules that enable software from multiple vendors to be integrated together to provide a biometrics application within a system, or between one or more systems using a defined Biometric Interworking Protocol (BIP).

33.3.1 Biometric Error Rates

Biometry authentication system deployment depends on different factors which must be taken into account. Key characteristic of biometric systems is accuracy. In biometry authentication scores (or weights) are used to evaluate

the similarity between a scanned pattern and a biometric template in database. The higher the score, the greater similarity between two. Access to a system is granted only if the score is higher than a certain threshold. Biometric system accuracy is measured in terms of decision error, matching error and image acquisition error rates. Decision error rates include:

False Acceptance Rate (FAR) refers to the acceptance of a forger into a system. This parameter estimates the probability (in %) – failing to reject a forger (FAR is a synonym to «Type II error rate»). It is stated as follows:

$$FAR = \frac{NFA}{NIIA} \quad (33.3)$$

or another case:

$$FAR = \frac{NFA}{NIVA} \quad (33.4)$$

Where:

- NFA is the number of false acceptances;
- NIIA is the number of impostor identification attempts;
- NIVA is the number of impostor verification attempts.

False Rejection Rate (FRR) refers to the rejection of a registered user. It estimates the probability (in %) – failing to accept a legitimate user (FRR is a synonym to «Type I error rate»). It is stated as follows:

$$FRR = \frac{NFR}{NEIA} \quad (33.5)$$

or another case:

$$FRR = \frac{NFR}{NEVA} \quad (33.6)$$

Where:

- NFR is the number of false rejections;
- NEIA is the number of enrollee identification attempts;
- NEVA is the number of enrollee verification attempts.

Equal Error Rate (EER) is the point where the FAR and FRR are identical. The EER gives a threshold-independent performance measure. The lower the EER, the better the system's accuracy. Synonym to ERR is Crossover Error Rate (CER).

Matching error rates include:

False Match Rate (FMR) – the probability that a sample will be falsely matched against a ‘non-self’ template;

False Non-Match Rate (FNMR) – the probability that a sample will not match a template of the same user.

Image acquisition error rates include:

The Failure To Enroll Rate (FTER) - the percentage of the population for whom the system is unable to generate repeatable templates (e.g person who is unable to register a fingerprint due to a severe injury).

The Failure To Acquire Rate (FTAR) refers to the proportion of the population for which the system is unable to capture an image of sufficient quality.

Also we have to remember that basic criteria of choosing biometry authentication system should be: difficulty to forge, usability, culturally acceptance, appropriate to environment of their usage and capable of either 1:1 or 1:n matching. Recent news, recommendations, best practices and use cases of biometric authentication technologies are covered on Planet Biometrics [10] and different organizations resources [11].

33.3.2 Biometry authentication methods

Nowadays a person identification by physical and behavioral characteristics became more natural and accepted by people all over the world. One of the most popular biometric technologies are based on static physical characteristics of a person. Widely used for person identification fingerprint recognition, hand and finger geometry, face recognition, iris pattern, retina and vein pattern recognition. A key aspect of this biometric technologies is that they are stable and shouldn't change significantly over a period of time.

Table 33.1 – Person's biometric characteristic comparison

Biometric characteristic	Universal	Unique	Permanence	Collectability	Performance	Acceptability	Circumvention
Face	High	Low	Medium	High	Low	High	Low
Fingerprint	Medium	High	High	Medium	High	Medium	High
Signature	Low	Low	Low	High	Low	High	Low

Another class of biometric authentication systems based on person's behavioral biometry characteristics. Behavioral biometrics based on measurable unique habits of an individual: on-line signature (dynamic signature), gait recognition, keystroke dynamics and voice authentication. On-line signature (fig. 33.7) authentication systems deals with the distinct characteristics of an individual's signature: shape, speed, stroke, pen pressure and timing information. Most often this technology used in applications in the financial sector for authenticating transactions and insurance transactions (e.g. PrivatBank (Ukraine) uses this technology). Keystroke dynamics based on measures of the speed and timing information every time a user presses a key on a computer keyboard. This technology can be applied only to blind-typing and suited to IT security-related tasks (e.g. PC log on).

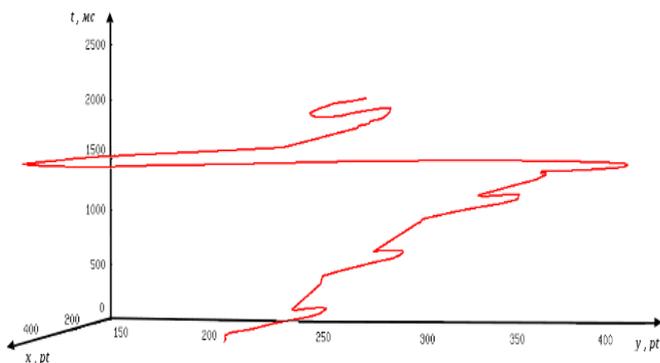


Figure 33.7 - On-line signature in 3D: coordinates x and y and time

Voice authentication based on a voice pitch and speaking style. There are few subtypes of this technology: text dependent (require an individual to say a pre-determined word or phrase), text prompted (the user says random words or phrases from a pre-enrolled set) and text independent (allow the user to speak freely). Also there are gait recognition systems which based on each person's unique way of walking. However this technology remains at the research stage and not used in production.

Main advantage of authentication systems based on static physical characteristics is stability of these characteristics and also main disadvantage is ability to create a mock. These systems cannot guarantee that person is alive.

Main disadvantage of behavior characteristics is ability to change by influence of emotions, illness or some other factors. But main advantage of these characteristics is inability to forge them. Accuracy of authentication systems based on static physical characteristics is more precise in sense of false accept and false reject rates than based on behavior characteristics.

33.3.3 Biometry authentication by fingerprinting

Fingerprint/Fingerprinting are the «traces» of minute ridges and valleys found on the finger of an individual. In the fingers and thumbs, these ridges form basic patterns such as loops, whorls, and arches(fig. 33.8), and also have finer level of details, such as ridge bifurcation and endings, pore placement on the ridge, and feathering of ridge boundaries.

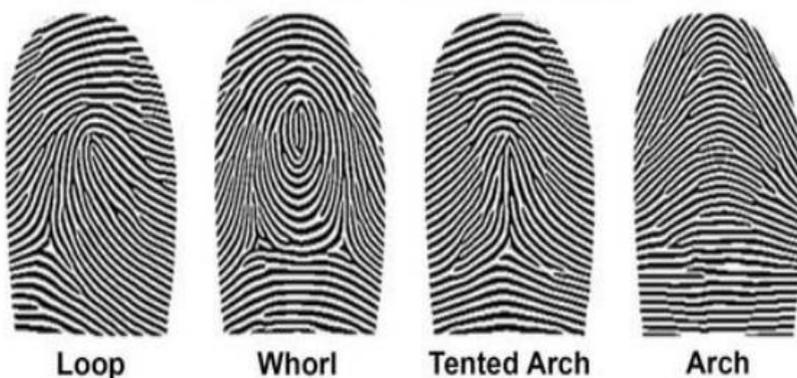


Figure 33.8 - Examples of various fingerprint patterns

Acquisition of a person's fingerprint characteristics for identifying purposes carried out by fingerprint scanning with a fingerprint sensor. Automated Fingerprint Identification System (AFIS) [11, 12] is a specialized biometric system that compares a single finger image with a database of fingerprint images. In law enforcement, AFIS is used to collect fingerprints from criminal suspects and crime scenes. In civilian life, fingerprint scanners are used to identify employees, protect sensitive data, integrated in smartphones etc. It is estimated that the number of possible fingerprint patterns is 10 to the 48th power. Fingerprint technology can be used effectively in both verification (1:1) and identification (1:N) applications.

Finger Image is a two-dimensional picture of the patterns found in the tip of the finger (fig. 33.8). There are four types of fingerprint-based authentication systems: direct correlation techniques, optical comparison, spectral ridge-pattern matching (ridge or global structure analysis) and the most popular by technology vendors' algorithm - minutiae-based matching (local structure analysis). Fingerprint patterns are captured by the system and grouped into several categories: left and right loops; whorls; arches and others.

Fingerprint patterns are stable throughout individual's lifetime, unique, easily analyzed and compared. Fingerprint systems (fig. 33.9) are easy to use, in most cases requiring the user to simply touch a platen with his/her forefinger. In addition to being secure, most fingerprint systems are relatively inexpensive.

Fingerprint devices have high accuracy levels but can suffer from usage errors when users are not properly trained and/or motivated to cooperate when placing their finger(s) on the reader. Conditions must be adequate for accurate authentication; for example, wet fingers, cuts on fingers, or dirt can cause authentication errors. Also, a sensor must be touched by multiple people, and some people feel uncomfortable with touching something that other people have touched repeatedly before them.



Figure 33.9 - Various fingerprint sensors

33.3.4 Biometry authentication by on-line signature

On-line (dynamic) signature authentication [13,14] belongs to the behavioral biometric class of authentication system.

On-line signature authentication identify individual by measuring and analyzing handwritten signatures. The difference between on-line and off-line signature is the following:

- off-line signature or traditional – signature for which there is only a static visual two dimensional record (image);

- on-line signature – a signature during production of which the pen trajectory or dynamics is captured (Fig. 33.10).

Off-line signature doesn't allow to get dynamic (on-line) characteristics and input sequence from it. On-line signature – is discrete signal, which consists of the set of dots, every of which has at least three dimensions: x-coordinate, y-coordinate and coordinate of time t . While it may be easy to duplicate the visual appearance of a static traditional signature, it is difficult to duplicate the behavioral characteristics when someone draws his/her signature.

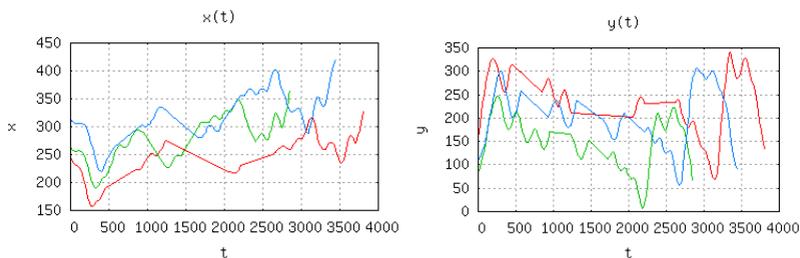


Figure 33.10 - Three on-line signatures $x(t)$ and $y(t)$ of an individual

By means of the graphic tablet, positioned pen, video camera, the stylus of the pocket PC, etc. on-line signature can be captured. Some input devices can increase information characteristics of signature – they allow to capture the slope angle (tilt) and the pressure.

To start the data acquisition phase of registration, the individual must sign his/her name multiple times on the graphical tablet or on the same input device. After data acquisition the on-line signature verification system extracts individual's behavioral characteristics: time of signing; the pressure applied; the velocity in signing the signature or its strokes; size of the signature; number of dots, strokes and order of them.

On-line signature authentication is acceptable and a non-invasive technology because people are currently accustomed to providing a signature to authorize transactions. As a result, there could be a high level of acceptance on the part of the end-user for this technology. Signatures widely used for commerce, so there is no violation of the right to privacy.

There are about 5% of individuals with unstable handwriting. Usually this unstable handwriting caused by some diseases and/or psychological dysfunctions. Also on-line signature authentication systems has limitations:

1. A signature cannot be too short (one curve, cross, dots) or too long (some handwritten text). If on-line signature is too long arise difficulty for the

on-line signature authentication system to identify consistent and unique data points. Too short on-line signature can cause simplicity of forging - a higher false accept rate.

2. The registration and authentication processes must be completed in the same type of environment and conditions. Because environment and conditions can influence on signature.

3. On-line signature can change over time and this fact can cause increasing the level of error rates over time.

Despite user acceptability, long history, and lack of invasiveness, signature verification is not a market leader like other biometric technologies (retina, fingerprint). On-line signature authentication has applications in financial establishments: Chase Manhattan Bank (the first known bank to adopt signature verification technology); IRS (tax returns that have been filed on-line); Charles Schwab & Company; PrivatBank Privat24. It is evident that growth of touchscreen gadgets market and broad usage of electronic documents have the biggest market application for on-line signature authentication systems.

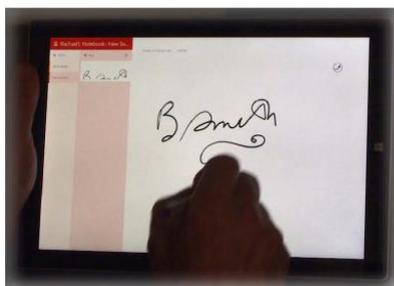


Figure 33.11 - Various devices with handwriting input

33.4 Safe Work Practices and Permit-to-Work Systems

Another very useful feature of authentication system is ability to test operator's psychological readiness to carry out complex and responsible tasks. In this case multifactor authentication system that combine both biometric authentication systems may be designed: by static physical characteristics and by behavior characteristics. First type of systems guarantee high precision and another one prevent operator from work in altered state of consciousness: mental disorder, under influence of alcohol or drugs or another undesirable

psychological state.

As known [14] analysis of handwriting can use as a tool to understand the emotional state of person can be implicated during psychiatric assessment (e.g. antidepressants can influence on signature). In this chapter use of biometric on-line signature authentication system as tool for evaluation the ability to do critical work is presented.

Graphology is a science of handwriting analysis that approved by many government organizations and scientists. Automation of graphology handwriting analysis is very complex and hard to solve problem. There are two intersected subject areas in evaluation of the ability to do critical work:

1) Handwriting recognition and analysis that deals with static images of handwriting, recognition of letters. Various writing features are analyzed: baseline, slant, size, margin, pressure (as a width of handwriting line), speed, spacing, zones, type of writing (cursive or print), types of strokes connections and few more. Speed and pressure measures are not accurate, because taken from 2D image. This task oriented to characterize person and an individual presence is not guaranteed.

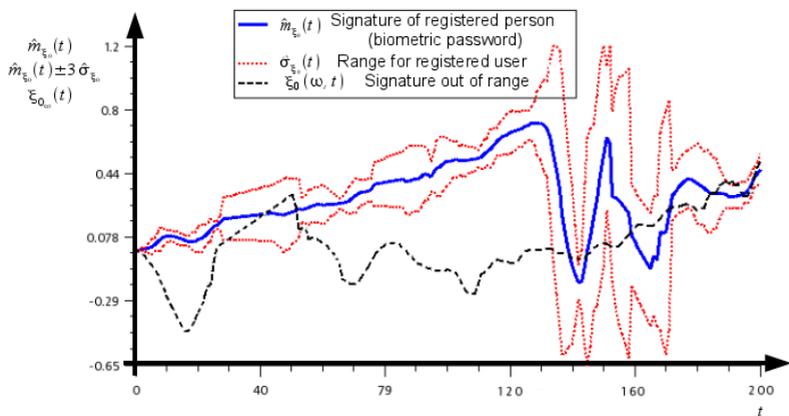
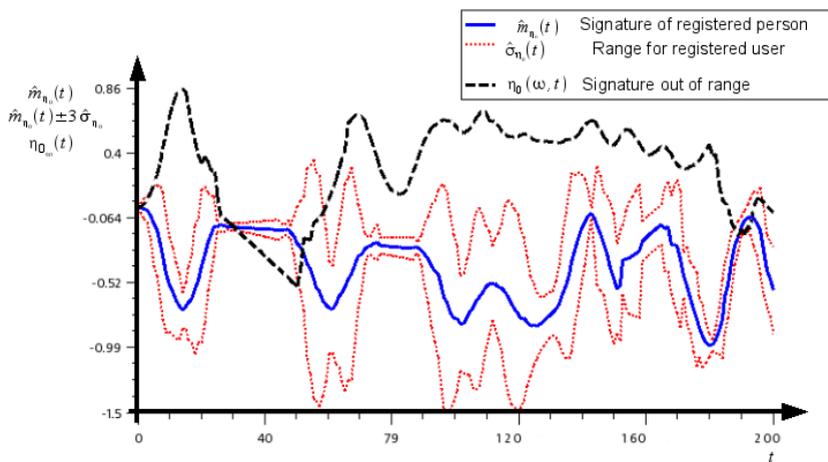
2) Authentication deals with static images of signature or dynamic signature. Main objective is only verification with acceptable precision: similarity between entered signature and stored in the database.

Evaluation of a person's readiness to do critical work combines these two subject areas, except this system mustn't recognize letters or handwriting but simply check for deviation compared with the sample. In contrast to the image recognition system, the system uses on-line signature and measures of pressure, writing speed, angle of a pen will be taken into account objectively.

On figure 33.14 general scheme of registration and authentication combined with permit to work evaluation is shown. On the schemes steps 1, 2, 5 (a) and 4 (b) are similar and described in [15]. Steps 6(a) and 7(a) are standard for authentication systems. On a step 7(a) authentication system log quantity of false rejects and if this number is greater than some threshold (fig.33.12 and fig.33.13) user suggested to update his on-line signature. Step 4(a), 5(b) and 3(b) are very similar except level of threshold.

According statistics there are about 5% of individuals with unstable signature (it can be caused by some mental illness). To detect these individuals authentication system based on statistical characteristics after few attempts of registration recommend to use another authentication method – step 4(a). Evaluation of a person's readiness to do critical work based on comparison with a threshold value (step 3(b)).

Threshold value can be set empirically after analysis of large number of experiments (also neural network usage is possible). Authentication (step 5(b)) based on threshold too.

Figure 33.12 - On-line signature ($x(t)$) in the range and forger signatureFigure 33.13 - On-line signature ($y(t)$) in the range and forger signature

Biometry authentication based on on-line signature mathematical models which are described in [15,16]. Development of authentication system and permit to work system must conform best-practices on biometrics implementation and usage [17].

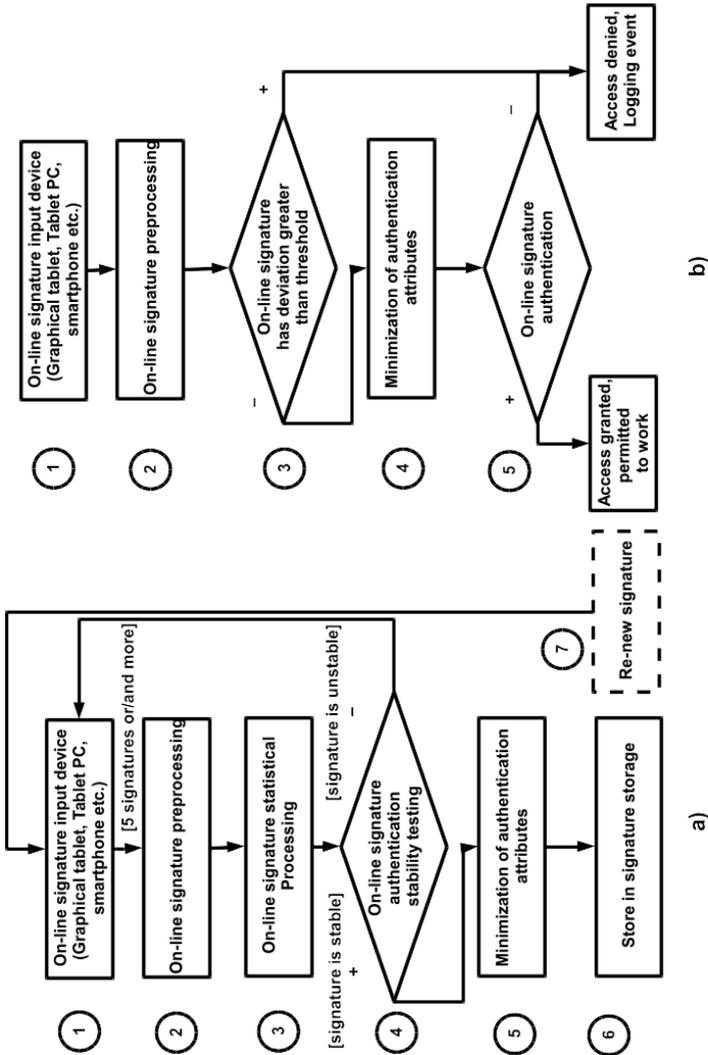


Figure 33.14 - General scheme of registration

Conclusion and self-control questions

We can model operators work [2] to customize and adopt system to its specific needs. Modeling allows to predict and prevent some types of human factors in computer security. Keystroke-level model allows predicting all types of operator's tasks in complicated HCI. Guiard's model emerged from a detailed analysis of how human's use their hands in everyday tasks. It is a descriptive model, lacking in mathematical rigor but rich in expressive power.

Endpoint security is a subject area which includes all aspects of operator's workspace security from malware protection to policy enforcement and asset tracking. Once organisation made the choice to monitor, it should follow best practices in endpoint security.

Human factor errors can be defined as circumstances in which planned actions, decisions or behaviors reduce — or have the potential to reduce — quality, safety and security. 95% of information security incidents caused by human factor errors.

Authentication, authorization, and accounting (AAA) are terms for a framework for access control to computer system and information resources. These terms include policies, audit and provide the information necessary to analyze operators activity or to bill for services in service providers networks. Authentication provides a way of user identification by using one of authentication method. After authentication, an operator must gain authorization for carrying tasks in system. The authorization process determines whether the operator has the authority to carrying such commands. Accounting measures the resources a user consumes or activities which have been done during access. AAA is a part of access control system and provide security technique that can be used to regulate who/what can view/use resources in a computing environment.

It is very important to diagnose good behavior and psychological state of responsible person, because it can influence the outcome of critical operations and the future of the resilient system. It is very important not only authenticate user but test readiness for carrying out his tasks.

Biometric authentication systems based on person's physical and behavioral characteristics. Biometric authentication system have to conform the following criteria: performance, fraud protection, acceptability. Biometric system accuracy is measured in terms of decision error, matching error and image acquisition error rates. Decision error rates include: False Acceptance Rate, False Rejection Rate, Equal Error Rate, False Match Rate, False Non-Match Rate, The Failure To Enroll Rate, The Failure To Acquire Rate. biometric technologies are based on static physical characteristics of a person. Widely used biometric authentication methods based on static physical

characteristics of a person: fingerprint recognition, hand and finger geometry, face recognition, iris pattern, retina and vein pattern recognition. A key aspect of this biometric technologies is that they are stable and shouldn't change significantly over a period of time. Another class of biometric authentication systems based on person's behavioral biometry characteristics: on-line signature (dynamic signature), gait recognition, keystroke dynamics and voice authentication. Authentication system can be used to test operator's psychological readiness to carry out complex and responsible tasks.

Self-control questions and tasks

1. Why and how operator's activity can be modeled?
2. Please give examples of operator's activity predictive and descriptive models.
3. What is endpoint security and why we should use it?
4. What is difference between the identification and the authentication?
5. What is difference between the authorization and the authentication?
6. What is the meaning of term AAA?
6. What are the most often human factor errors?
7. What is social engineering attack?
8. Please specify and describe human security threats.
9. What is the reason of incidents and accidents according to resilience engineering?
10. What are basic resilient system components in human aspect?
11. What components belong to biometric complete identity management system?
12. Describe main processes in biometric authentication system.
13. What properties have to conform any human biological or behavioral characteristics to become a biometric identifier?
14. Please specify the requirements to the authentication system.
15. What is difference between the verification and identification in biometric authentication systems?
16. What is the reason to use standards in biometric authentication and what standards or specification do you know?
17. Please specify types of errors and errors in biometry authentication system.
18. What is difference between the decision, matching and image acquisition error rates?
19. What International and European Person Authentication Standards do you know? What subject areas they cover?
20. What International and European Biometric Standards do you know?

21. What methods of biometric authentication do you know?
22. What pros and cons of fingerprint authentication do you know?
23. What does the term behavior biometry authentication stands for?
24. What pros and cons of on-line signature authentication do you know?
25. How operator's readiness to work can be measure?
26. How you can describe usable authentication system?
27. Which biometric characteristic has High Acceptability but Low Permanence? Which biometric characteristic has Low Performance and is Universally High?

References

1. Gerardo De Maria. Human-Machine Interaction in Aviation: A Future Threat or Resource. American Journal of Science and Technology. Vol. 3, No. 1, 2016, pp.25-42.
2. MacKenzie, I. S. Motor behaviour models for human-computer interaction. In J. M. Carroll (Ed.) HCI models, theories, and frameworks: Toward a multidisciplinary science, pp. 27-54. San Francisco: Morgan Kaufmann. 2003. [Electronic resource]. – Accessed at: http://www.yorku.ca/mack/mackenzie_chapter.html
3. Manavoglu, E., Pavlov, D., Giles, C.L.: Probabilistic user behavior models. In: Proceedings of ICDM. 2003. [Electronic resource]. – Accessed at: <https://cgliles.ist.psu.edu/papers/ICDM-2003-probabilistic-user.pdf>
4. Mitnick K. The Art of Deception / Kevin D. Mitnick, William L. Simon, Steve Wozniak // John Wiley & Sons, 2002. - 304p.
5. Deursen N. How to Reduce Human Error in Information Security Incidents / Nicole van Deursen, 2015 [Electronic resource]. – Accessed at: <https://securityintelligence.com/how-to-reduce-human-error-in-information-security-incidents/>
6. Security Threats. Best Practices for Enterprise Security. Microsoft Developer Network. [Electronic resource]. – Accessed at: <https://msdn.microsoft.com/en-us/library/cc723507.aspx>
7. Parsons K. Human Factors and Information Security: Individual, Culture and Security Environment / Kathryn Parsons, Agata McCormac, Marcus Butavicius, Lael Ferguson // Command, Control, Communications and Intelligence Division Defence Science and Technology Organisation. Australian Government Department of Defence. 2010. [Electronic resource]. – Accessed at: <http://www.dtic.mil/dtic/tr/fulltext/u2/a535944.pdf>

8. European Standard CEN/TC 224 - Personal identification and related personal devices with secure element, systems, operations and privacy in a multi sectorial environment / European Committee for Standardization [Electronic resource]. – Accessed at: https://standards.cen.eu/dyn/www/f?p=204:29:0:::FSP_ORG_ID,FSP_LANG_ID:6205,25&cs=1F65610F4AF62E7CCBB59BE2FC6ABF7CB#1
9. International Standard ISO/IEC JTC 1/SC 37 Biometrics [Electronic resource]. – Accessed at: <https://www.iso.org/committee/313770.html>
10. Biometric Essentials / Planet Biometrics [Electronic resource]. – Accessed at: <http://www.planetbiometrics.com/biometric-essentials/>
11. Biometric Technology Application Manual. Volume One: Biometric Basics / National Biometric Security Project. 2008. 403 p. [Electronic resource]. – Accessed at: http://www.planetbiometrics.com/creo_files/upload/article-files/btamvollupdate.pdf
12. Gashi, I., Mason, S., Lugini, L., Marasco, E. & Cukic, B. (2014). Interoperability between Fingerprint Biometric Systems: An Empirical Study. Paper presented at the IEEE International Conference on Dependable Systems and Networks, 23rd - 26th June 2014, Atlanta, GA, USA.
13. Seifer M. The Telltale Hand: How Writing Reveals the Damaged Brain / Marc J. Seifer // The Dana Foundation. Your gateway to responsible information about the brain. October, 01, 2002 [Electronic resource]. – Accessed at: <http://www.dana.org/Cerebrum/Default.aspx?id=39304>
14. Singh G. Handwriting change as a psychiatric symptom. International Journal of Medical and Dental / Singh G.H, Mehta R.J., Shah N.D., Mehta R.Y. // Sciences 2016, №5(1): 1075-1078.
15. Lutskiv A. Biometry authentication system based on on-line signature verification // Proceedings of the international conference “Computer science and information technologies”(CSIT 2006). - Lviv, 2006. - P.43-48.
16. McKeague I. W. A Statistical Model for Signature Verification // Journal of the American Statistical Association. - 2005. - P.231–241.
17. Jain, A., Flynn, P. & Ross, A. 2008. Handbook of biometrics. 1st edition. Springer. USA.

PART 8 HUMAN-MACHINE ENGINEERING FOR SECURITY CRITICAL AND RESILIENT SYSTEMS

CHAPTER 34 GROUP DECISION MAKING AND HUMAN ASPECTS IN CYBER SECURITY AND EMERGENCY MANAGEMENT

CONTENTS

34.1 Groups and teams: the fundamental concepts	2
34.1.1 Important differences between groups and teams	2
34.1.2 Team failures.....	4
34.2 Metrics of the human-system interaction	6
34.3 Basic principles of high team performance	7
34.3.1 Basics of a successful team process and collaboration between humans	8
34.3.2 Basics of successful team-like cooperation between humans and machines.....	9
34.4 Group decision making and human aspects of emergency management	12
34.4.1 GDM in emergency management.....	13
34.4.2 Factors affecting human group decision making during emergencies.....	15
34.5 DGM and human aspects in cybersecurity	17
34.5.1 A teamwork on cyber security and resilience.....	17
34.5.2 Cyber security collaboration and response	18
34.6 The human factors evaluation and decision making under multiple and conflicting goals	20
34.6.1 Statement of the problem of decision-making under competition....	20
34.6.2 The problem of decision-making using DS belief structures	21
34.6.3 The procedure of group decision making	24
Conclusion	29
Self-control questions and tasks	30
References	30

34 GROUP DECISION MAKING AND HUMAN ASPECTS IN CYBER SECURITY AND EMERGENCY MANAGEMENT

34.1 Groups and teams: the fundamental concepts

There are many common goals across a wide range of work environments, including business, academic, military, medical, and other critical areas, that are either too physically or cognitively complicated to be achieved by individuals working alone. To meet high demand targets, tasks must be performed in real time by people working together as a group or a team. Cybersecurity is like that. Effective cybersecurity requires that every individual, and every part of the organization, to work together as a team.

To meet the need for better system defending and more effective training of the groups, it is necessary to understand their potential, capabilities, and limitations. This knowledge helps to apply the methods of task analysis; perform function allocation in complex human-machine systems and human-technological systems; understand core principles of a successful team process; understand the rules and current limitations of human-machine cooperation; understand channels of processing and resource group competition.

34.1.1 Important differences between groups and teams

In mathematics, a group is represented by algebraic structure composed of a set of elements contains an operation that combines any two elements to form a third element, and that satisfies four axioms, namely closure, associativity, identity, and invertibility.

Social concept of the group can be formulated as a collection of individuals who have relations to one another that make them interdependent to some significant degree or other words a group of people is two or more persons who are connected to one another by social relationships. These definitions relate together three essential elements: the number of individuals involved; connection, and relationship. Furthermore, the group can interact simultaneously (i.e., pooled-interdependent mode) or make individual decisions separately and then collectively confront and discuss the results (i.e., sequential-interdependent mode).

The Fig. 34.1 shows task interdependence and coordination requirements starting at the bottom with the least degree of coordination required (pooled) up to the most degree of coordination required (comprehensive) [1].

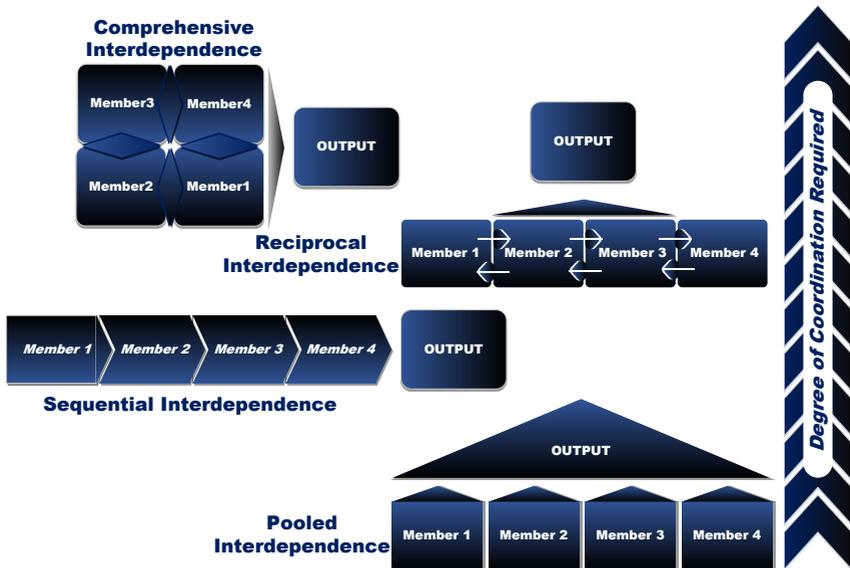


Figure 34.1 – Task interdependence and coordination requirements

In contrast with group, a team is a set of two or more individuals who interact and adapt to achieve shared and valued goals. Traditionally, the team is defined as an interdependent group of two or more people who work together for a fixed amount of time to achieve a common goal. All teams are groups, but not all groups are teams. Compared with an individual, teams represent increased cognitive resources which can contribute a substantial amount of information, situational models, and proposed courses of action.

Teams often are difficult to form. It takes time for members to learn how to work together. The main distinguishing criteria for groups and teams are represented in Table 34.1.

Besides this, there are at least two characteristics distinguish teams from groups:

- Rate team-working
- Presence of a common specific goal or objective

Other points of distinction between a team and a group are that task completion requires:

- (a) dynamic exchange of information,
- (b) coordination of such activities as active communication, situation monitoring, backup behaviors, etc.,

- (c) adjustments to task demands,
- (d) some structure to the members.

In safety-critical systems teams are used in several cases:

- when errors lead to severe consequences;
- when the task complexity exceeds the capacity of an individual;
- when the task environment is ill-defined, ambiguous, and stressful;
- when multiple and quick decisions are needed;
- when the lives of others depend on the collective insight of individual

members.

Table 34.1 – Criteria and differences between groups and teams

Criteria	Team	Group
Goals	Mutually agreed, clarified	Shared interests
Commitments	High, to team and goals	Low
Relationships	Interdependent, coordinated	Interact, shared
Contributions	Synergetic	Individual
Synergy	Positive	Neutral or negative
Accountability	Individual and shared	Individual
Skills	Random, varied	Complementary
Identity	Clearly defined	Shared, maybe
Culture	Shared, stable	Diverse
Example	Baseball team	Baseball fans

Teams take a variety of forms, from teams of teams to human-robot teams. As the complexity of the workplace continues to grow, organizations increasingly depend on teams and the cost of team failures grows.

For security and cyber security problems, all forms of teams might be considered. Especially when we are talking about the emergency management, computer supported collaborative work, team cognition in high-risk situation or cyber wars. True cybersecurity teamwork requires that every group – not just every individual – within the organization assess and address its respective impact [2].

34.1.2 Team failures

As it mentioned in Chapter 33, the “human error” is involved in more than 95 percent of the security incidents. And for most of socio-technical systems, up to 80 % of the accidents were not caused by technological failures but instead were the result of inadequacies in problem solving, faulty decision-making, and substandard or nonexistent [3].

A practical approach to quantifying human error within the accident process is discussed in [4]. To model the likelihood (P) of occurrence of a human error event a following mathematical relationship is proposed:

$$P_{\text{human error}} = \left[\left(1 - \frac{1}{\# \text{options}} \right) \cdot \text{feedback} \cdot \text{adjuster} \cdot \text{redundancy} \right],$$

where

#Options are the choices faced by an individual increase, so does the opportunity for, and likelihood of, error.

Feedback: visual feedback (e.g. the ability to actually see an action performed) will reduce the likelihood of human error.

Adjusters (external or internal): these cover the environment experienced by the operator e including temperature, humidity, clothing, mental and physical capabilities, and training.

Redundancy: this is defined as a real-time repeat of the investigation of whether a human error is occurring.

This relationship fits for measuring the individual human error. But a set of questions are still open and need to be considered: How to measure the team errors and evaluate a team factor? If it is really exists? How it can influence the complex systems?

Individual errors happen when a person either works alone or in a team but isolated from others. Unlike individual errors, team errors (see Table 34.2) happen when there is a minimum of two people collaborating and interacting, such as exchanging information or working together on the same task [5].

Table 34.2 – Some types of team errors

Error type	Overview
Communication error	Failure to communicate information Partial reports and partial orders
Vigilance error	Failure to intercept and prevent errors of other team members
Interpretation error	Incorrect or needlessly delayed diagnosis (decision) based on available information
Management error	Loss of track of progress for a multistep procedure

In contrast to individual’s errors, little is known about errors that are unique to teamwork. Previous work offers a preliminary theoretical framework for understanding team errors [6, 7], but does not explain how or why team errors happen.

One of the consistently found reasons for poor teamwork is the lack of a shared understanding about necessity and forms of teamwork. As a result, emerging conflicts among team members and a breakdown in communication can impair collaboration and result in an underutilization of available resources and the creation of new problems. Besides, team members may not share the same situational model and may be reluctant to question actions of teammates even when serious concerns. To better understand team failures and team capabilities, it is necessary to have a tool for estimation their effectiveness within the context of interaction teams, machines and surrounding systems.

34.2 Metrics of the human-system interaction

In the Chapter 31 some sets of metrics to examination specific requirements for the critical systems are discussed. Much of them include traditional human factors such as reaction time, error rates, and so forth, however, when we are talking about the system level such metrics fail to capture the effectiveness of the human-system interaction and do not diagnose the cause of problems they expose.

To measure system effectiveness, the following five metrics classes can be applied [8]. These metrics are suitable to assess of both individual components and holistic systems.

(1) Autonomous platform behavior efficiency (e.g., usability, adequacy, autonomy, learnability, errors, user satisfaction, automation speed, accuracy and reliability, neglect time).

(2) Mission effectiveness (e.g., key mission-performance parameters relating to the human-automation system).

(3) Human behavior efficiency – operators perform multiple tasks such as monitoring autonomous platform health and status, identifying critical exogenous events, and communicating with others as needed. How humans sequence and prioritize these multiple tasks provides valuable insights into system design effectiveness.

- Information processing efficiency (e.g., decision making)

- Attention allocation efficiency (e.g., scan patterns, prioritization)

(4) Human behavior precursors – the underlying cognitive processes that lead to specific operator behavior, as compared with the human behavior metric class that captures explicit behavior.

- Cognitive precursors (e.g., situation awareness, mental workload, emotional state)

- Physiological precursors (e.g., physical comfort, fatigue)

(5) Collaboration metrics or Team-level metrics.

- Human-automation collaboration

- Automation-automation collaboration
- Human-human collaboration

The final class is the collaboration metrics it addresses the degree to which the collaborators are aware of one another and can adjust their mutual behavior.

The human-automation collaboration metric revolves around measures of team cognition and trust. Evaluation of these parameters can inform system design requirements as well as the development of training material. Objective measurement of trust, a difficult task, is important when system reliability and a culture where different knowledge domains exist in distinct silos could create trust barriers.

In the automation-automation collaboration subclass, the quality and efficiency of the collaboration among the machines can be measured through metrics such as speed of data sharing and decision making among automated agents, quality of the system response to unexpected events, and the ability of the system to handle network disruptions.

The last collaboration metric subclass is human-human collaboration or team collaboration. In networked settings, a human team necessarily works together to perform collaborative tasks, so performance should be measured at the holistic level rather than by aggregating team members' individual performance [9]. Because team members must consistently exchange information, reconcile inconsistencies, and coordinate their actions, one way to measure holistic team performance is through human-human coordination, which includes written, oral, and gestural interactions.

34.3 Basic principles of high team performance

A close relationship exists between good teamwork and successful performance in a high-stakes environment, whereas poor teamwork and communication have emerged as key factors responsible for the occurrence of critical errors.

Further, we refer to two different team formations – human-human and human-machine and discuss some requirements to the efficient and effective team performance and integration of humans with complex machines in terms of 3C: Communication, Coordination, and Collaboration.

Communication refers to the amount of information sharing employed by teams. Coordination refers to the ability of a team to match their behavior to complete interdependent tasks and cooperation is represented by the desire of individuals to provide assistance to team mates. We will use 3C behaviors as indicators of effective team performance.

34.3.1 Basics of a successful team process and collaboration between humans

It is well known that a collection of experts does not create an expert team and not guarantee the good team performance. And the fact that every member of a team understands and accomplishes their individual task well does not mean that the team will perform successfully.

According to [10], a team has to cross the stages such as Work group, Pseudo team, Potential team, and Real team to reach high performance. The inverse process is possible too. When a group develops into an expert team the team is able to identify the task work requirements necessary for them to maintain high levels of performance. And one of the most interesting phenomena here is a collaboration between elements (or agents) where humans or machines play the role of these elements and they are taken as equal in terms of their interaction.

Case 1: Good collaboration

If the interaction of elements (e_1, e_2, \dots, e_n) in the system is orchestrated and consistent its capability (C) is growing and becomes much more than ordinary sum of the capacity their elements that means high performance. The entropy of the system S_H is less than the sum of entropies of their elements S_{Hi} .

$$C(A) \gg (C(e_1) + C(e_2) + \dots + C(e_n)),$$

$$S_H < \sum S_{Hi}.$$

Case 2: Neutral collaboration

Neutral collaboration exists when the degree of interconnectedness and level of coherence in the system doesn't deliver credible component interaction. In this case there is a composition of C instead of emergent properties.

$$C(A) = (C(e_1) + C(e_2) + \dots + C(e_n)),$$

$$S_H = \sum S_{Hi}.$$

Case 3: Weak collaboration

Weak collaboration results in two cases where there is losing control or antagonism common to pseudo systems.

Case 3.1: The absence of control

Under conditions the absence of control a system capability reduces to the capability of its average element:

$$C(A) = (C(a_1) + C(a_2) + \dots + C(a_n)) / n .$$

Case 3.2: Antagonism

Under the competitive antagonism the worst comes to the worst and system capability drops to the minimum.

$$C(A) < \min(C(a_1) + C(a_2) + \dots + C(a_n)) ,$$

$$S_H > (S_H(a_1) + S_H(a_2) + \dots + S_H(a_n)) .$$

The entropy of the system is greater than the sum of entropies of elements of the system, implying that some irreversible process prevents the system operation. It means that if basic principles of a team process are neglected, or if teams get under stress, internal team dynamics may develop which will lead to a lower performance or degradation.

In such a case the following occurs: Team members tend to conform their opinion to the majority in the team. Legitimate concerns are not articulated, and criticism is withheld. The misunderstanding may result from the use of ambiguous and nontechnical terminology as well as from relational problems, so the collaboration is one of the essential guarantors of good team performance but not only one. Thus, in [11] author distinguished four requirements to create a successful, wise, and high performance group:

- (1) to provide diversity of opinion: each person should have private information on the case discussed;
- (2) to ensue independence: people's opinions must not be determined by the opinions of those around them;
- (3) to realize decentralization: people must be able to specialize and draw on local knowledge;
- (4) to promote aggregation: some 'mechanism' has to be used for turning private judgments into a collective decision.

34.3.2 Basics of successful team-like cooperation between humans and machines

Other important sources of potential errors are interactions between humans and technical systems.

The field of Human-Computer Interaction (HC) has coined a principle called WYSIWYG [12]: *What You See Is What You Get*

WYSIWIG describes an interface that allows the user to view something very similar to the end result while creating a document or an image.

Accident analysis seems to work with a similar principle, which can be called WYLFIFYF: *What You Look For Is What You Find*

WYLFIFYF means that an accident investigation usually finds what it looks for. In other words, the assumptions about the nature of an accident constrain the analysis.

To this can be added the principle of WYFIWYF: *What You Find Is What You Fix*

To achieve the state where HC interaction is really effective, it is not enough for people to understand their machine teammates; computers should understand aspects of humans and their goals as well. Computers need to model people in ways that capture their expectations, commands, and constraints and also be able to understand what people “say” (in whatever language – formal or natural – they are using). What does the human expect the computer to do? What is the human telling the computer to do? In this case we are talking about the team-like cooperation. In the Table 34.3 the requirements for effective team-like cooperation between humans and machines is represented.

Table 34.3 Requirements for team-like interactions among humans and automation (adapted from [13])

Requirements	Explanation (Condition)
<p>Mutual predictability of teammates</p>	<p>To be a team player, an intelligent agent like a human must be reasonably predictable and reasonably able to predict others’ actions [14].</p> <p>It should be both observable itself, and it should be able to observe and correctly predict future behavior of its teammates.</p> <p>Making automation more adaptable can change its behavior less predictable. To make actions predictable enough, targets, states, capacities, intentions, changes, and upcoming actions should be visible to the people and automation components that supervise and coordinate with them. This requirement contradicts to the advice sometimes given to automation developers to create systems that have just noticeable difference.</p>
<p>Establishment and</p>	<p>Common ground is the most important basis for</p>

<p>maintenance of common ground</p>	<p>interpredictability [15], it refers to the proper mutual knowledge, mutual beliefs, and mutual assumptions that support interdependent actions in a joint activity.</p> <p>Common ground refers to the process of communicating, testing, updating, tailoring, and repairing mutual understandings and permits people to use abbreviated forms of communication, such as head-nods (or an automation analogy) and still be reasonably confident that potentially ambiguous messages and signals will be understood. It also includes what parties know about each other prior to engagement—for example, the others’ background and training, habits, and ways of working.</p>
<p>Ability to redirect and adapt to one another</p>	<p>Directability refers to deliberate attempts to modify the actions of the teammates as conditions and priorities change. For example, during coordinated activity, team members must exert oneself to evaluate what each other needs to notice, within the context of the task and the current situation. It up-ranges the limits of technology to allow the automation to communicate virtually as if it were part of a well-coordinated human team working in an open, visible environment. The machine will have to signal when it is having trouble and when it is taking extreme action or moving toward the extreme end of its range of authority. [16].</p>

The human-automation interaction should be designed so as to [17]:

- information processing is accessible to the operator;
- communication functions and features enable operators to obtain the information;
- human can assess the credibility of the results;
- interruptions are minimized;
- the level of detail can be controlled;
- the machines monitors the performance of the humans, to alert them to potential errors;
- experts can provide input to the processing and direct its activities.

The focus of this particular research area is the efficient and effective integration of humans with complex machines. Examples of such systems include aircraft cockpits, air traffic control, chemical processing, and the power industry. One of the recent topics is the development of such systems for large robotic teams (10 or more robots). This area focuses on the human

factors and software engineering aspects of integrating such teams while also incorporating artificial intelligence techniques to provide intelligent information and interaction capabilities.

This work requires the multimodal interaction between human and machine and development of interfaces that rely on artificial intelligence and provide intuitive interaction capabilities and flexible human-machine interaction beyond the standard graphical user interface, computer keyboard, and mouse. Examples of multimodal interaction capabilities include eyeglass displays, speech, gesture recognition, and other novel interfaces.

34.4 Group decision making and human aspects of emergency management

Emergencies and pre-emergencies are the situations where group decision support is of vital importance. A risk is not static it develops over time, together with the activities that are performed, the implementation of initiatives, learning from incidents, accidents, and success, use of new technology, development of work processes, and updating of procedures and guiding rules. Risk-informed decisions imply that one has to know whether the decision foundation is sufficient and to evaluate the need and the possibility to reduce the uncertainty further before a decision is made [18]. Emergency management (EM) requires a collaboration of different stakeholders and decision makers, either to prepare to respond to the emergency or to make decisions and allocate resources during the emergency itself [19].

Hardware (input/output device), software (user interface), people (group member and facilitator) and procedures (methods used in meetings) are the four components of group decision support systems (GDSS) [20] (see Figure 34.2). Concerning hardware and software, GDSS offer various levels of computational aid to remove communication obstacles (level 1), provide techniques for structuring decision analysis (level 2) and systematically directing the pattern, timing, or content of the discussion (level 3) [21].

GDSS researchers have identified three main tasks involving different cognitive processes: eliciting information (brainstorming or divergent thinking), exploring courses of action (convergent thinking), and evaluating situations (convergent thinking) [21].

In general case teams can influence the quantity and quality of complex systems performance by affecting the following variables:

- Structure and processes;
- Equipment and technologies;
- Human resource management;
- Teamwork and leadership;

- Communication;
- Organizational culture.

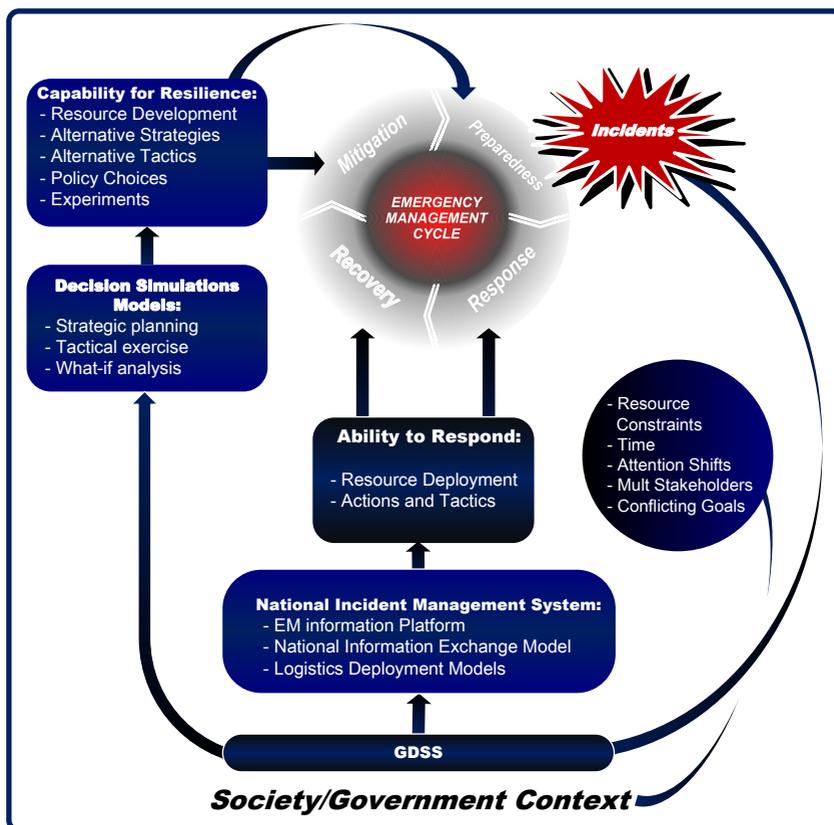


Figure 34.2 – GDSS and the Emergency Management Cycle [22]

34.4.1 GDM in emergency management

Based on the essential characteristics of an emergency, the process of the group decision-making in emergency situations is entirely different from the common decision-making process. The differences are listed as follows:

- Decision makers should respond emergency events more rapidly and take decision schemes more quickly;

- Any emergency events are so complicated that many groups, including departments, enterprises, and other individuals, will all participate in the decision-making process. Therefore, group decision-making requires a considerable level of cooperation;
 - An emergency is highly unpredictable and changeable. Therefore, decision makers should track the emergency events unceasingly, test and correct decision schemes without delay.
- To fit the problems connecting with emergency management the system approach should be applied. The primary enablers of individual and collaborative system thinking are listed in Table 34.3.

Table 34.3 – Enablers of individual and collaborative system thinking

Enablers of individual system thinking	Enablers of collaborative systems thinking	
Individual characteristics	Team characteristics and Norms	Consensus decision making
		Real-time group interactions
Supportive environment	Supportive environment	Overall creative environment
		A realistic schedule
Experiential learning	Experiential learning	Systems experience (Past and concurrent)

Two functions are required to support the generation of group decision making (GDM). The first is the representation of the data collected about a situation. During an emergency, people often only imprecisely or ambiguously know a situation and use the uncertain information to present it, for examples, 'very high,' 'high' and 'very low.' The second is the approaches or tools for assessing the situation. As GDM has to be generated through aggregating these imprecise and inaccuracy information and opinions, fuzzy information processing techniques, particularly fuzzy sets based linguistic term process approaches, are suitable.

This kind of decision-making problems can be handled by human teams but they are too difficult for machines to handle. The Dempster-Shafer theory of evidence enables us to integrate heterogeneous information from multiple sources to obtain collaborative inferences for a given problem. As an example, in Paragraph 34.6 the group decision support technique for handling under multiple and conflicting goals is considered.

34.4.2 Factors affecting human group decision making during emergencies

Groups tend to centralize information flow and decision-making when external pressure arises. A collective decision-making process can be defined as a decision situation in which (i) there are two or more persons, each of them characterized by their own perceptions, attitudes, motivations, and personalities, (ii) who recognize the existence of a common problem, and (iii) attempt to reach a collective decision.

Teamwork activity includes:

- Information exchange;
- Communication;
- Supporting behavior;
- Team initiative / leadership;
- Team training.

One of the most important things here is an information exchange means that team members share information and create a shared mental model (or common operating picture). Information exchange allows the team to develop and maintain a common understanding of the situation as each member communicates critical information. This process allows to provide situation awareness and to draw a ‘big picture’ of what is happening. If some decision is to be made, because the people will have the similar understanding, they may not have conflict. Moreover, by sharing information, people create a pool of knowledge which helps them in creating the more and more clear picture of the incident. It results in effective decision making and turns effective emergency management.

Team communication is strictly related to team performance, and the similarity of knowledge structures between two team members can improve the quality of team performance. Effective communication comes with four essential conditions that team members should meet:

(5) using the uniform communication terminology to pass large amounts of information very quickly;

(6) providing complete internal and external reports to minimize ambiguity associated with communicating

(7) minimizing unnecessary communications (e.g., chatter) focusing only on the essentials of interaction necessary for team performance;

(8) make sure that their communications are clear and audible to minimizing the misinterpretations and misunderstandings of communications as well as reducing the communication-related workload involved in clarifying communications of initially low quality.

The supporting behavior includes two specific teamwork activities. First, team members should correct the errors of other team members. This practice reduces the number of errors and helps to develop the skill levels of team members as they receive feedback on poor performance. Second, team members should provide and request assistance and backup when it is needed. This involves team members monitoring each other's performance, identifying when their team members need assistance or they themselves need assistance, and stepping in to resolve the unbalanced workload situation.

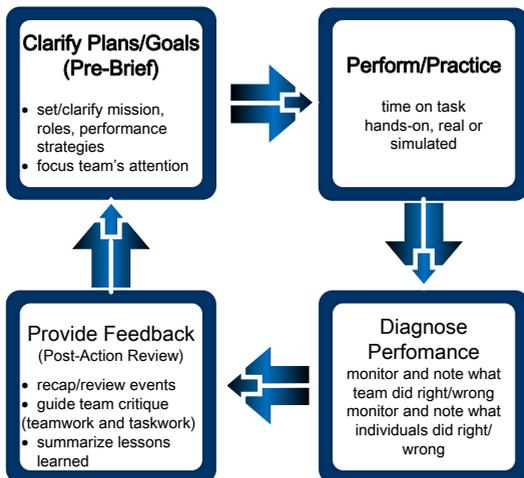


Fig. 34.3 – Team training cycle diagram

Team learning behavior includes such activities as continuously seeking improvement, lessons learned, mutual performance monitoring, feedback, communication, co-ordination, and decision making.

However, as shown in Figure 34.4, different factors such as availability of time, nature of the incident, an experience of team leaders, tendency to be in own comfort zone, confidentiality, a primacy of information with one agency and use of different languages and terminologies affect information sharing and hence group decision making.

Teams must be encouraged and trained to handle emergencies. Emergencies often differ from situations operators normally are trained in, and often the solutions they are trained to take do not fit the actions and decisions they need to take in an emergency. Therefore, training initiatives in collaboration and decision-making will be an important tool in risk and emergency situations [18].

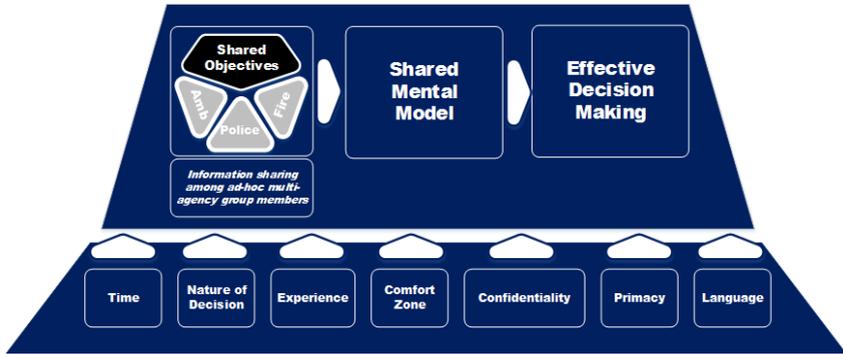


Fig. 34.4 – Factors affecting group decision making (adapted from [23])

As we consider emerging technologies in the context of human-machine interactions and/or unmanned systems and how to design and coordinate these technologies for emergency response, we have identified a new challenge: How can human expertise be integrated into cyber-physical systems to adapt and manage multiple, large scale, time critical processes? Understanding the interaction between computational processes and the physical world with the overarching influence of humans will result in new and challenging problems in control.

34.5 DGM and human aspects of cyber security

A typical organization contains a large number of computing systems such as desktop computers, laptops, servers, networking devices, and more that produce large amounts of data in the form of system logs, network traffic data and sensor data (alerts from intrusion detection systems). High-performing IT departments can establish effective defensive and protective policies and processes, and of course, can provide highly sophisticated security measures. But it takes only one mistake – at the wrong place, at the wrong time – to give a cyber opponent the opening needed to cause a possible, potentially disastrous breach. [2].

34.5.1 A team work on cyber security and resilience

In practice, a system is only as secure as its weakest element, i.e., the easiest way in. Identifying which are the weakest aspects of a system, i.e., the easiest ways of attacking it, is thus a highly relevant component of system security assessment, though obviously, it does not provide all of the answers

[24]. Cyber attacks and the resulting security breaches are part of a rapidly expanding international cyber threat that costs companies billions of dollars each year in lost information and response costs.

The human factor may be a systems weakest link, but may also be a powerful resource to detect and mitigate developing threats. In this context situation awareness (SA) can be considered as a phenomenon that refers to extract environmental information, integrate it with previous knowledge to direct further perception and anticipate future events. Since SA is regarded as a dynamic and collaborative process, it is often required in a team.

Team SA is commonly used in the human-computer interaction community where the concerns are to design computer interfaces so that a human operator can achieve SA in a timely fashion. Within large organizations, the investigation and resolution of cyber incidents rest upon the Cyber Security Incident Response Team. The primary responsibility of this team is to review information from a variety of sources (e.g., intrusion detection systems, automated queries, user reports, notifications from other cyber professionals) to identify evidence of potential cyber threats. The corresponding tasks rely on general knowledge of computer and network systems and domain-specific knowledge of the local infrastructure, and an appreciation of adversary tactics and techniques. There is an emphasis on cognitive processes that enable inferential reasoning, pattern recognition, procedural memory, and communication. The main activities of cyber security teams include threat hunting and threat intelligence, monitoring, detection and resolution of incidents.

Team cognition, which is defined as cognitive processes such as decision-making and learning, occurs at the team level and has a significant effect on team performance [25].

34.5.2 Cyber security collaboration and response

With cyber defense analysis being a complex task, it is sometimes performed by cyber security defense analysts as a large group, with each analyst working on different levels of the task with specific domain knowledge and experience.

Cyber security defense analysts have to monitor and fuse large amounts of data in order to identify patterns that may correspond to potential cyber attacks [26]. For example, analysts usually start from a suspicious set of intrusion alerts, filter network level data pertinent for those intrusion alerts, find associated system level logs, find intelligence reports relating to the situation, and then using their experience and training analyze the data collected to decipher if their network is being attacked or not.

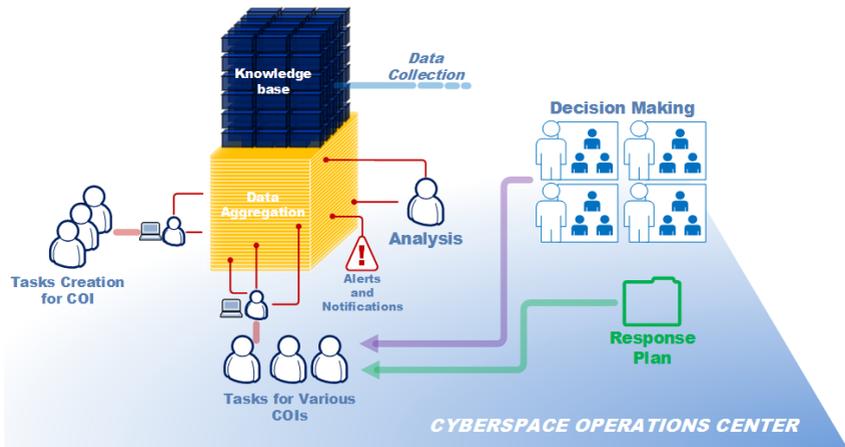


Figure 34.5 – The core of cyber security collaboration and response

Once the analysts suspect there is an ongoing attack, the analysts start collecting data as evidence to support their suspicion and to eventually report the findings to higher authorities. Finally, the analyst must assess the adversaries' intentions and capabilities to take the appropriate response. These tasks are mostly conducted manually using command level interfaces or graphical interfaces.

The cyber response team is liable for developing the written cyber incident response plan and for investigating and responding to cyber attacks in accordance with that plan.

Work teams consist of highly trained individuals with special expert knowledge. Each person is responsible for one specific area of the complex system. Specifically, the cyber response team should [27]:

- Develop the cyber incident response plan.
- Identify and classify cyber attack scenarios.
- Determine the tools and techniques used to detect and prevent attacks.
- Secure the company's computer network.
- Develop a checklist for handling initial investigations of cyber attacks.
- Determine the scope of an internal investigation once an attack has occurred.
- Conduct any studies within the determined scope.
- Promote cyber security awareness within the company.
- Address data breach issues, including notification requirements.

– Conduct follow up reviews on the effectiveness of the company’s response to an actual attack.

If a cyber attack has occurred, the response team should follow the investigation checklist set out in the cyber incident response plan to conduct the initial inquiry.

The initial response varies depending on the type of attack and level of seriousness. However, the response team should stop the cyber intrusions from spreading further into the company’s computer systems.

34.6 The human factors evaluation and decision making under multiple and conflicting goals

As it mentioned above, the group decision-making is a situation where two or more decision makers are involved in the decision of a joint problem whereas each of them has their own understanding of the problem and the decision consequences (competing hypotheses). Formally, competing hypotheses or conflict set is considered as a set of objects concerning which there is no consensus among at least two experts.

Conceptual model M of a typical situation assessment problem in the presence of competing hypotheses

$$M = \langle A, S, P, D \rangle,$$

where A is a set of possible conclusions about the situation (alternatives), a generalization of logic experts; S is a set of baseline data on the situation which is measured in quantitative and qualitative scales; P are the analytical dependences, which provide formation of conclusions $a \in A$ according to the data S ; and D are the techniques that allow to select the most important information from S .

34.6.1 Statement of the problem of decision-making under competition

Let A be a set of alternatives $\{A_1; A_2; \dots; A_q\}$ whose values describe variants of the decision; S be a set of object states $\{S_1; S_2; \dots; S_q\}$, characterizing the possible scenarios; values $c_{11}; c_{12}; c_{1n}; c_{21}; c_{22}; c_{2n}; c_{n1}; c_{n2}; \dots; c_{ln}$ – are the specific level of effectiveness of the solution corresponding to a specific alternative in a certain situation.

Knowledge of the safety conditions fixed in terms of belief structure m . B_1, \dots, B_r are the focal elements of m and $m(B_k)$ are the associated weights.

The task involves finding the best alternative that delivers the payoff to the decision makers.

Moreover, to solve the problem, consider the following conditions:

- the presence of subjective quality expert information, characterized by a set of competing hypotheses and requiring aggregation;
- form of the matrix of solutions may vary depending on the selected performance indicators;
- the method should provide support for decision-making, in order to lookup minimal losses as well as for the problem of finding maximum efficiency.

The next sections present the theoretical provisions based on the extended Dempster-Shafer belief structure and the method for automated decision support based on evidence-based reasoning applicable for critical IT infrastructure. We have implemented our method on top of decision-support software tool, so it can be easily adopted to the different IT security risk management tasks. Dempster-Shafer theory has unique advantages in handling uncertainty in critical IT-infrastructure analysis, namely, a means to explicitly account for unknown possible causes of observational data and the ability to deal with the lack of prior probabilities for all events and the ability to combine beliefs from multiple sources.

34.6.2 The problem of decision-making using DS belief structures

The Dempster-Shafer (DS) belief structure is defined in the space X consisting of a set of n nonzero subsets B_j , $j=1, \dots, n$, called the focal elements and basic belief assignment m called the mass function or the probability of mass which is denoted as m . It is a mapping function defined as $m: 2^X \rightarrow [0,1]$, satisfying

$$\sum_{j=1}^n m(B_j) = 1, \forall B_j \subseteq X,$$

$$m(A) = 0, \forall A \neq B_j.$$

Model of the belief structure is a distributed evaluation with the levels of beliefs to represent an effectiveness of alternative for the selected criteria.

Suppose that the criterion is evaluated by a full range of possible situations with n estimated classes, $H = \{H_1; H_2; \dots; H_j; \dots; H_n\}$, where H_j is the j -th evaluation class.

Without loss of generality, we may assume that H_n is preferred H_{n+1} . This assessment criterion can be represented by the following distribution

$$S(c) = \{F(H_j, m(B_j))\}, \quad j = 1, \dots, n, \quad (34.1)$$

where $m(B_j) \geq 0$, $\sum_{j=1}^N m(B_j) \leq 1$.

The function (34.1) denotes that the criterion is assessed for the class H_n with the level of confidence $m(B_j)$.

Estimation $S(s)$ is complete if $\sum_{j=1}^N m(B_j) = 1$ and incomplete if. A special case is $\sum_{j=1}^N m(B_j) = 0$ which means a complete disregard for the criterion.

There are two measures associated with the belief structures – plausibility (Pls) and belief (Bel) or similarity [28].

Pls is defined as the measure $Pls: 2^X \rightarrow [0,1]$ such that $Pls(A) = \sum_{A \cap B_j \neq \emptyset} m(B_j)$.

Similarly, the confidence measure is defined as $Bel: 2^X \rightarrow [0,1]$, such that

$$Bel(A) = \sum_{B_j \subseteq A} m(B_j).$$

Bel represents precise support, while Pls is a possible support. Through these measures is possible to submit confidence interval A as $[Bel(A), Pls(A)]$. This interval is considered respectively as the lower and upper levels of trust.

Schafer model defines distinguishing frame, Θ , as the space of all possible solutions.

Dempster rule allows for each set of initial subsets (focal elements) on the entire set of input data to generate the resulting subsets and calculate their confidence level (combined measure of confidence (probability mass)). Dempster's rule for combining hypotheses X and Y is performed by orthogonal summing corresponding confidence measures m_1 and m_2

$$m_{12}(A) = \frac{\sum_{X \cap Y = A} m_1(X)m_2(Y)}{1 - k_{12}} \quad (34.2)$$

where

$$k_{12} = \sum_{X \cap Y = \emptyset} m_1(X)m_2(Y) \quad (34.3)$$

The main problem with this approach in the design of automated decision support systems is the presence of a normalizing factor $(1 - k_{12})$ which completely ignores the conflict. Practically, when k_{12} equal 1, the combination rule of evidence (34.2) is not determined mathematically.

To solve this problem, a number of models combining different hypotheses were developed, among them models of D. Dubois et al., E. Lefevre et al., C. Murphy, P. Smets, R. R. Yager et al.

In this chapter we use calculation rule [29] by selecting $\omega_m(\Theta) = 1$ and $\omega_m(A \neq \Theta) = 0 : m(\emptyset) = 0$

$$m(A) = \sum_{X \cap Y = A} m_1(X)m_2(Y), \quad (34.4)$$

$$m(\Theta) = m_1(\Theta)m_2(\Theta) + \sum_{X \cap Y = \emptyset} m_1(X)m_2(Y) = \omega(\Theta) + \omega(\emptyset),$$

if $\Theta = \emptyset$, where $\forall A \in 2^\Theta, A \neq \emptyset$.

In critical applications (for distributed team decision-making, or under interdisciplinary incomprehension, for example) the individual solutions can be compared to formal aggregation procedures to select a general consensus. The final solution must be obtained from the synthesis of performance degrees of criteria. To this end, the aggregation of information is fundamental.

One of the most common methods of aggregation is a method using the ordered weighted averaging (OWA) operator, introduced by Ronald R. Yager. Since its description, the given operator has been used in a wide range of applications. It provides a parameterized family of operators, including arithmetic mean, geometric mean (the ordered weighted geometric (OWG) operator); harmonic mean (the ordered weighted harmonic (OWH) operator); a set of nonadditive integrals (Sugeno integral, Choquet integral); weighted minimum; weighted maximum, as well as enhanced operators of ordered weighted average. Here, we consider two orders of the OWA operator – ascending and descending, as well as some of the main results of their use in the decision-making model.

Assume that X is a set of information sources, $f(x_i)$ is a value supplied $x_i(c)$, σ and s are the permutations such that $a_{\sigma(i)} \geq a_{\sigma(i+1)}$, $a_{s(i)} \leq a_{s(i+1)}$.

Then, according to [30], the OWA operator in ascending order is calculated by (34.5), descending one by (6).

$$OWA_{\sigma} = \sum_{i=1}^n w_i a_{\sigma(i)} \quad (34.5)$$

$$OWA_s = \sum_{i=1}^n w_i a_{s(i)} \quad (34.6)$$

where w is a weight vector such that: $w_i \in [0,1]$, $\sum_{i=1}^n w_i = 1$. If the characteristics of the individual values represent the relative values of the dynamics, for example, describe the average growth rate, it is advisable to use the geometric mean. In this case, the operators of the weighted geometric mean in ascending and descending order are calculated by (34.7) and (34.8), respectively.

$$OWG_{\sigma} = \prod_{i=1}^n a_{\sigma(i)}^{w_i} \quad (34.7)$$

$$OWG_s = \prod_{i=1}^n a_{s(i)}^{w_i} \quad (34.8)$$

34.6.3 The procedure of group decision making

To get the best alternative in the group decision-making, the following steps are involved:

Step 1. Formation of a decision matrix

Depending on the type of the problem, the matrix of possible solutions can be represented as a payoff matrix including performance indicators, or in the form of a risk matrix consists of financial loss indexes. It corresponds to certain combinations of alternatives to decision-making and possible scenarios

	s_1	s_2	...	s_n
A_1	c_{11}	c_{12}	...	c_{1n}
A_2	c_{21}	c_{22}	...	c_{2n}
\vdots	\vdots	\vdots	\vdots	\vdots
A_l	c_{l1}	c_{l2}	...	c_{ln}

Step 2. Definition of a set focal elements $B \subseteq \Theta$ and the appointment of the main mass of probability to subsets

$$B^1 = (B_1^1, B_2^1, \dots, B_i^1, \dots, B_q^1),$$

$$B^2 = (B_1^2, B_2^2, \dots, B_j^2, \dots, B_r^2).$$

Step 3. Calculation of belief function for the combined sets using (34.4)

$$m(B_k) = \sum_{B^1 \cap B^2 = B} m_1(B_i^1) m_2(B_j^2).$$

Step 4. Determination of the weight coefficients collection used in the aggregation functions for the individual sets of focal elements:

$$w = (w_1, w_2, \dots, w_n) \text{ such that } w_j \in [0, 1]; \sum_{j=1}^n w_j = 1.$$

Each weight can be obtained by

$$w_j = Q\left(\frac{j}{n}\right) - Q\left(\frac{j-1}{n}\right) \quad (34.9)$$

Where Q is a function of fuzzy linguistic quantifiers defined as

$$Q(r) = \begin{cases} 0, & \text{if } r < \alpha, \\ \frac{r-\alpha}{\beta-\alpha}, & \text{if } \alpha \leq r \leq \beta, \\ 1, & \text{if } r > \beta. \end{cases} \quad (34.10)$$

$$Q(0) = 0, \quad Q(1) = 1;$$

$$r < t \Rightarrow Q(r) \leq Q(t);$$

$$3) \sum_{j=1}^n w_j = \sum_{j=1}^n \left(Q\left(\frac{j}{n}\right) - Q\left(\frac{j-1}{n}\right) \right) = Q(1) - Q(0) = 1.$$

Quantifier Q in (34.10) is defined as a linear membership function for all $\alpha, \beta, r \in [0, 1]$.

The values α, β are determined depending on the linguistic meaning of the quantifier.

Step 5. Calculating a set N_{ik} , which is formed when the i -th alternative has selected and k -th focal element, $\forall i, k : N_{ik} = \{c_{ij} \mid s_j \in B_k\}$.

Step 6. Ordering N_{ik} sets for each of the criteria

$$OWA_{\sigma}, OWG_{\sigma} : s_1 > s_2 > \dots, > s_j > \dots, > s_{n-1} > s_n,$$

$$OWA_s, OWG_s : s_1 < s_2 < \dots, < s_j < \dots, < s_{n-1} < s_n,$$

$$\forall s_j \in N_{ik}, \quad j = 1, \dots, n.$$

Step 7. Calculation of aggregated values M_{ik}

$$M_{ik} = \sum_{j=1}^n w_j \cdot s_j. \quad (34.11)$$

Step 8. Calculation of the expected value of the overall index for each alternative

$$c_i = \sum_{k=1}^r M_{ik} \cdot m(B_k) \quad (34.12)$$

Step 9. Ordering and selection of an alternative in accordance with the objectives and the current rules.

To illustrate this approach, let us examine a problem selecting strategies to mitigate targeted cyber intrusions. This problem can be solved by combining subjective threat judgment information received from the decision makers based on their professional experience.

Planning team has to identify the best mitigation actions that can be readily implemented but because of funding, technical support, and other causes may not be immediately available for every action. Therefore, it is necessary to prioritize the most suitable mitigation actions to implement in the target system.

1. Assume that the decision problem has four mitigation strategies (alternatives A_1, A_2, A_3, A_4). To each strategy, we attribute generalized metrics, which allow assessing the mitigation actions in four process areas: s_1 - vulnerability management, s_2 - patch management, s_3 - configuration management, and s_4 - incident management as a base for analysis.

	s_1	s_2	s_3	s_4
A_1	10	40	20	30
A_2	15	20	25	30
A_3	40	30	10	20
A_4	40	50	10	30

2. Assume further that there are two groups of experts, each of that defined its own judgment concerning the best mitigation actions and used the model of the belief structure for each alternative on each criterion as follows.

Group 1: $(\{s_1, s_2, s_4\}, 0, 8; \{s_2, s_3, s_4\}, 0, 1; \{s_2, s_4\}, 0, 1)$.

Group 2: $(\{s_1, s_2, s_4\}, 0, 5; \{s_2, s_3, s_4\}, 0, 4; \{s_2, s_4\}, 0, 1)$.

Then the set of focal elements to merging sets can be represented as follows:

	$\{s_1, s_2, s_4\}$ 0,8	$\{s_2, s_3, s_4\}$ 0,1	$\{s_2, s_4\}$ 0,1
$\{s_1, s_2, s_4\}$ 0,5	$\{s_1, s_2, s_4\}$ 0,4	$\{s_2, s_4\}$ 0,05	$\{s_2, s_4\}$ 0,05
$\{s_2, s_3, s_4\}$ 0,4	$\{s_2, s_4\}$ 0,32	$\{s_2, s_3, s_4\}$ 0,04	$\{s_2, s_4\}$ 0,04
$\{s_2, s_4\}$ 0,1	$\{s_2, s_4\}$ 0,08	$\{s_2, s_4\}$ 0,01	$\{s_2, s_4\}$ 0,01

3. Calculation of the belief function carried out by (4):

$B1$	$\{s_1, s_2, s_4\}$	0,4
$B2$	$\{s_2, s_3, s_4\}$	0,04
$B3$	$\{s_2, s_4\}$	0,56

4. Determination of weight coefficients w , which are used for aggregation functions for the individual sets of focal elements. Let $w_1 = (0,4; 0,6)$, $w_2 = (0,3; 0,4; 0,4)$.

5. Defining sets N_{ik} .

6. Ordering sets $N_{ik} : a_{\sigma(i)} \geq a_{\sigma(i+1)}$, and $a_{s(i)} \leq a_{s(i+1)}$.

7. Calculation of aggregated values M_{ik} performed by (34.11), the results are presented in Table 34.4.

Table 34.4. The results of the calculation of aggregate values

Operat or	Aggregate value											
	M_{11}	M_{12}	M_{13}	M_{21}	M_{22}	M_{23}	M_{31}	M_{32}	M_{33}	M_{41}	M_{42}	M_{43}
OWA_s	31	34	36	$24,5$	28	26	34	23	26	45	35	42
OWA_σ	28	32	34	23	27	24	32	21	24	43	31	38
OWG_s	$24,2$	$29,8$	$35,6$	$21,6$	$25,1$	$25,5$	$29,8$	$19,1$	$25,5$	$40,1$	$26,5$	$40,7$
OWG_σ	$21,1$	$27,8$	$33,6$	$20,1$	$24,1$	$23,5$	$27,8$	$17,1$	$23,5$	$38,1$	$22,5$	$36,8$

8. Calculation of the overall index is performed by (34.12). The results are summarized in Table 34.5.

Table 34.5. The results of the calculation of the generalized index

	OWA_s	OWA_σ	OWG_s	OWG_σ
A_1	33,92	31,52	30,81	28,37
A_2	25,48	23,72	23,92	22,16
A_3	29,08	27,08	26,96	24,96
A_4	42,92	39,72	39,89	37,87

9. The choice of an alternative is performed in accordance with the preference rule presented in Table 34.6.

Table 34.6. The results ordering alternatives and prioritizing the most suitable mitigation actions in the target system

Operators	The order of preference alternatives
OWA_s, OWG_s	$A_2 > A_3 > A_1 > A_4$
OWA_σ, OWG_σ	$A_4 > A_1 > A_3 > A_2$

If the main goal of intrusion mitigation programs formulated for improving system security or increasing confidence of security we will use descending order of operators, otherwise for example for risk reduction, ordered weighted operators in ascending order will applicable. For OWA_σ , OWG_σ operators, as the best solution chosen A_4 because it gives the highest expected value. For OWA_s and OWG_s operators selected variant A_2 , since in these cases it is believed that the best result is the lowest.

The group decision support technique permits the use of subjective expert information formalized in the form of family of estimations and based on the combination of hypotheses and ordered weighted average operators. The task is formulated in terms of the belief structures and allows evaluate the minimum and maximum objectives. As an example, the problem of prioritization for cyber intrusion mitigation programs is considered. Another interesting issue to consider in the context of safety and cyber security of critical IT-infrastructures is the group decision support in following areas:

- Prediction the situation change trends, when mitigation actions have not undertaken.

- Prediction safety/security trends in case the decisions were not taken.
- Selection factors that have maximum impact on the critical IT-infrastructure attributes (safety, security, reliability, etc.).
- Evaluation of the impact of individual measures or groups of measures on the critical IT-infrastructure attributes.
- Factoring intrusion-sensitive activity.
- Failure effects evaluation and consequence analysis for individual mitigation programs measures or groups of measures, etc.

The proposed method is effective for decision support under competing hypotheses and helps in enhancing cyber security team performance. The formal basis for automated reasoning based on the theory of Dempster-Schafer has been successfully extended and applied to a number of problems including multisensory data fusion and analysis of process data. Where those decisions relate to the involvement of human analyst resources to activities, this technique essentially improves the efficiency of group decision-making in critical environments. Besides, this approach can be useful for designing a human-machine autonomous system that simultaneously aggregates human and machine knowledge to recognize targets in rapid change environments. These systems dynamically aggregate decisions involving uncertainties from both human and autonomous agents.

This part was written specifically for the international conference "Theory and Engineering of Complex Systems and Dependability." For more information, please refer [31].

Conclusion

Teams and groups are ubiquitous. We use teams in aviation, the military, health care, financial sectors, nuclear power plants, engineering problem-solving projects, manufacturing, and many other domains. Teams definitely are forms of groups, but not all work groups are teams.

In different combinations like human-human, human-machine, and machine-machine the groups and teams allow us to come together and increase individual's potential, resources, and experience.

Our purpose here was threefold: (a) to briefly discuss what we already know about groups, teams, teamwork, and team performance; (b) to highlight recent discoveries and developments in human-system interaction, and (c) to motivate researchers for the future studies in this area. This chapter is purely selective. We focus here only on those areas in which we think significant for some aspects of emergency management and cyber security and in which there is a wide field of future activity.

Self-control questions and tasks

1. What are the main differences between groups and teams?
2. Are there any errors unique to teamwork?
3. What specific task can't be solved by individuals?
4. What metrics can be used to measure the effectiveness of human-system interaction?
5. How can collaboration between the humans and machines be measured? What is the difference?
6. What is meant by weak collaboration and how it impacts on system capabilities?
7. State minimal requirements for team-like interactions among humans and automation
8. What variables can teams influence for ensure quantity and quality of systems performance?
9. How is team communication related to team performance?
10. How is decision-making in emergency situations differ from the typical decision making process?
11. What is team situation awareness?
12. Which activities cyber security teams include?
13. What kind of decision-making problems can be difficult for machines to handle?
14. How to integrate heterogeneous information from multiple sources to obtain collaborative inferences?
15. Give an example of the problem of prioritization for cyber intrusion mitigation programs.

References

1. Task interdependence and coordination requirements: Executive summary [Electronic resource]. – Accessed at: <http://olin-weiand-tejeda.weebly.com/executive-summary.html>
2. Sybersecurity training team sport [Electronic resource]. – Accessed at: <https://www.bizlibrary.com/article/cybersecurity-training-team-sport/>
3. St. Pierre M., Hofinger G., Simon R. Crisis Management in Acute Care Settings Human Factors and Team Psychology in a High-Stakes Environment / Springer International Publishing, 2016. – 433 p.
4. Marhavalas P.K., Koulouriotis D., Gemeni V. Risk analysis and assessment methodologies in the work sites: On a review, classification and comparative study of the scientific literature of the period 2000-2009 // Journal of Loss Prevention in the Process Industries. – 2011. – vol. 24. – pp. 477-523.

5. Sarcevic A., Marsic I., Burd R.S. Teamwork Errors in Trauma Resuscitation // ACM Trans. Comput.-Hum. Interact. – Vol. 19(2). –2012. – 30 p. [Electronic resource]. – Accessed at: <http://ischool.drexel.edu/faculty/asarcevic/pub/a13-sarcevic.pdf>
6. Sasoua, K., Reason, J. T. 1999. Team errors: Definition and taxonomy. Reliability Engin. Syst. Safety 65, 1, 1–9.
7. Trepess, D., Stockman, T. A classification and analysis of erroneous actions in computer supported co-operative work environment // Interact. Comput. – 1999. – Vol. 11(5). – pp. 611–622.
8. Pina, P. E., Donmez B., Cummings, M.L. Selecting Metrics to Evaluate Human Supervisory Control Applications // Cambridge, MA: MIT Humans and Automation Laboratory, 2008.
9. Cooke, N. J., Salas E., Kiekel P. A., Bell B.. Advances in measuring team cognition. In Team cognition: Understanding the factors that drive process and performance / E. Salas, S. M. Fiore, and J. A. Cannon-Bowers, eds. Washington, DC: American Psychological Association, 2004.
10. Katzenbach, J.R., Smith, D.K. The Wisdom of Teams: Creating the High Performance Organization. Collins Business Essentials, An Imprint of HarperCollins Publishers, NY, 2005.
11. Surowiecki J. The wisdom of crowds / Doubleday; Anchor, 2004. – 336 p. [Electronic resource]. – Accessed at: <http://www.ascib.ase.ro/mps/TheWisdomOfCrowds-JamesSurowiecki.pdf>
12. Nuclear Safety: A Human Factors Perspective Edited by J. Mitsumi, B. Wilpert, R. Miller. - CRC Press, 1998. – 350 p.
13. Complex Operational Decision Making in Networked Systems of Humans and Machines: A Multidisciplinary Approach // Committee on Integrating Humans, Machines and Networks: A Global Review of Data-to-Decision Technologies; Board on Global Science and Technology; Policy and Global Affairs; National Research Council – [Electronic resource]. – Accessed at: <http://www.jeffreybradshaw.net/publications/18844.pdf>
14. Sycara, K., and M. Lewis. Integrating agents into human teams. In Team Cognition: Understanding the Factors that Drive Process and Performance, E. Salas and S. Fiore, eds. American Psychological Association, 2004.
15. Clark, H., S. Brennan. Grounding in communication // Perspectives on Socially Shared Cognition, L. Resnick, J. Levine, and S. Teasley, eds. Washington, DC: American Psychological Association, pp. 127–149.
16. Christoffersen, K., D. Woods. How to make automated systems team players // Advances in Human Performance and Cognitive Engineering Research, 2002. – vol. 2.
17. Oxstrand, J., Le Blanc K.L., Joe J.C., Whaley A.M., Medema H., O'Hara J., Framework for Human-Automation Collaboration: Conclusions

from Four Studies / INL/EXT-13-30570, Rev. 0, Idaho National Laboratory, 2013.

18. Kaarstad M., Rindahl G. Shared collaboration surfaces to support adequate team decision processes in an integrated operations setting // *Advances in Safety, Reliability and Risk Management*, 2012 Taylor & Francis Group, London. – pp. 240-248.

19. Drabek, T. *Emergency Management: Principles and Practice for Local Government*, International City Management Association, Washington, D.C, 1991.

20. Gray, P. The Nature of Group Decision Support Systems / *Handbook on Decision Support Systems*, F. Burstein and C.W. Holsapple, eds., Springer, 2008, pp. 371–389.

21. Vennix J. A. M., Andersen D. F., Richardson G. P., Rohrbaugh J. Model building for group decision support: Issues and alternatives in Knowledge elicitation // *Modeling for Learning Organizations*, Productivity Press, Portland, OR, 1994.

22. Zhao X. et. all *Group Decision Support Systems for Emergency Management and Resilience* // *Proceedings of the 50th Hawaii International Conference on System Sciences*. – 2017. – pp. 2489-2497.

23. Mishra J.L. Factors affecting group decision making: an insight on information practices by investigating decision making process among tactical Commanders // *Proceedings of ISIC: the information behaviour conference*, Leeds, 2014: Part 1. [Electronic resource]. – Accessed at: <http://www.informationr.net/ir/19-4/isic/isic10.html#Eng01>

24. Miller S., Appleby S. Evolving OWA Operators for Cyber Security Decision Making Problems // *2013 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*. – pp. 15-22.

25. Rajivan P., Champion M., Cooke N. J., Jariwala S., Dube G., Buchanan V. Effects of Teamwork versus Group Work on Signal Detection in Cyber Defense Teams // *D.D. Schmorow and C.M. Fidopiastis (Eds.): AC/HCI 2013, LNAI 8027*, pp. 172–180, 2013.

26. Boyce, M.W., Duma, K.M., Hettinger, L.J., Malone, T.B., Wilson, D.P., LockettReynolds, J.: Human performance in cybersecurity. // *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. – 2011. – pp. 1115–1119.

27. Farhat V., McCarthy B., Raysman R., *Cyber Attacks: Prevention and Proactive Responses* // Holland & Knight LLP, 2011. – 12 p.

28. Shafer, G.: *A mathematical theory of evidence*. Princeton University Press, Princeton, 1976.

29. Yager, R.R. On the Dempster-Shafer framework and new combination rules // *Information Science*. – 1987. – Vol. 41(2) – pp. 93–137.

30. Yager R.R. On ordered weighted averaging aggregation operators in multi-criteria decision making. // IEEE Trans. Systems, Man and Cybernetics. – 1988. – Vol. 18. – pp. 183–190.

31. Skarga-Bandurova I., Nesterov M., Kovalenko Y. A Group Decision Support Technique for Critical IT-infrastructure // Theory and Engineering of Complex Systems and Dependability Advances in Intelligent Systems and Computing. – 2015. – Vol. 365. – pp. 445-454.

Summary

This course is designed to provide students with a fundamental understanding of human factors that must be taken into account in the engineering of complex systems and understanding ways of reducing the potential for human behaviors that play a significant role in breaches of cyber security. The primary focus is the group aspects of cyber-security, human-machine interaction and decision making on security and resilience for human and industry related domains.

When you have read this chapter, you will: learn more about the groups and teams, their characteristics, features of group errors, and how they differ from an individual ones; know how to measure system effectiveness from a human-system interaction's perspective; understand some group-work phenomena and how cooperation and coordination can effect on the target task performance; get to know the requirements for good team-like interactions among humans and automation; find out factors affecting human group decision making during emergencies; learn the cases when human factor can be powerful resource to detect and mitigate developing threats in cyber space.

Part 9. Security Management and Availability Assessment of Smart Building Automation Systems

35. Security Management Systems

35.1. Standards and models of Security Management Systems

35.1.1 Model of information security risk management system

The management of the information security risk in organizations is carried out using the principles and recommendations of international standards ISO 31000 “Risk Management. Principles and Guidelines”, ISO/TR 31004 “Risk management. Recommendations for the implementation of ISO 31000” as well as ISO/Guide 73 “Risk Management. Glossary” [1-3]. This is due to the fact that in ISO/IEC 27005 “Information technology. Methods and means of ensuring security. Managing Information Security Risk” provides a general approach to managing information security risk [4]. Therefore, the terms specified therein are additionally supplemented by the guidelines of the international standard IEC 31010 “Risk Management. Methods for Risk Assessment” [5]. Defined in these legal documents principles and guidelines used to design and propagation of the risk management system of information security. Such a system is developed in view of the functional limits, functions and conditions of the functions defined by the results of functional modeling in graphical notation IDEF0 [6-8].

The review and synthesis of organizational and technical systems by functional modeling notation in IDEF0 is carried out in various areas [8], including information security [9-18]. This is due to the availability of tools for modeling a wide range of processes to ensure its confidentiality, integrity and availability at an organization at any level of detail. The results obtained herein are used as the basis for making decisions on the reconstruction, replacement or development of a new system [8].

Thus, the system of ensuring information security organization displayed graphic notation IDEF0 [9]. The process of functioning of such a system is modeled on an example of a military medical institution with the separation of such functions as entrance control, task execution, monitoring. However, the functional simulation evaluation process security information technology for the “common criteria” and the basic processes of information security management as notation IDEF0, IDEF3 and DFD considered in [10, 11]. This comparative analysis is performed and selected DFD notation given

vysokorivnevist notation IDEF0, IDEF3. This choice is due to research just how data flows in evaluating the security of information technology and system security management information without determining functions and their implementation consistent. For example, in [12] to solve the problem of constructing models analyze functional component of the research object. In this case, the limits of its functioning are outlined, and due to the decomposition of the functional model, its components are analyzed in more detail. Research safeguards based assessment process modeling notation in IDEF0 and IDEF3 is considered in [13].

This allowed to meet the requirements (width, depth, severity) to the process of assessing compliance with the guidelines of the international standard ISO/IEC 15408. Functional modeling of decision support system for the provision of personal data security was performed in [14]. The constructed functional model made it possible to describe the subject area and, as a result, to form the information space for the presentation of knowledge about the protection of personal data. In addition, the IDEF0 functional model built automated control system security through multi-media approach [15]. Among the main functions of this system is the definition of the composition of the multicentric environment, the development of functions and interdependencies between agents, coding agents. Due to this, the data flow of the network is functionally simulated and investigated.

Options for building virtual infrastructure in the health care problem detection and information security risks in IDEF0 modeled in [16]. This approach focuses on analyzing processes and flows, identifying vulnerabilities and disadvantages of the functioning of the information system. Based on this, a set of countermeasures is proposed to reduce the risk of information security. Study of data flow in the information system functional simulation using a notation IDEF0 is made in [17]. The model of a given system shows its functional structure, processes and interaction between them. Whereas the functional simulation evaluation process security information and system resources e-commerce notation IDEF0 singled out are its functions [18]: Building a model e-commerce systems, modeling threats and risk assessment, modeling evaluation of the security of e-commerce systems, modeling report generation and recommendations.

Given the analysis of recent research and publications, the aim of this section is formulated as a synthesis of the risk management system of information security by determining its functional scope and functions by functional modeling in IDEF0.

Functional modeling of the information security risk management system (ISRMS) in the notation IDEF0 vidobrazhannya reduced to its individual functional blocks as shown in Fig.35.1 [1-8]. It affects the function of the upper level of inputs, outputs, constraints, and call arrangements and

formulated purpose and point of view of building a functional model. Due to this the model shows the structure of the information security risk management system, functional boundaries, functions and conditions for their implementation.

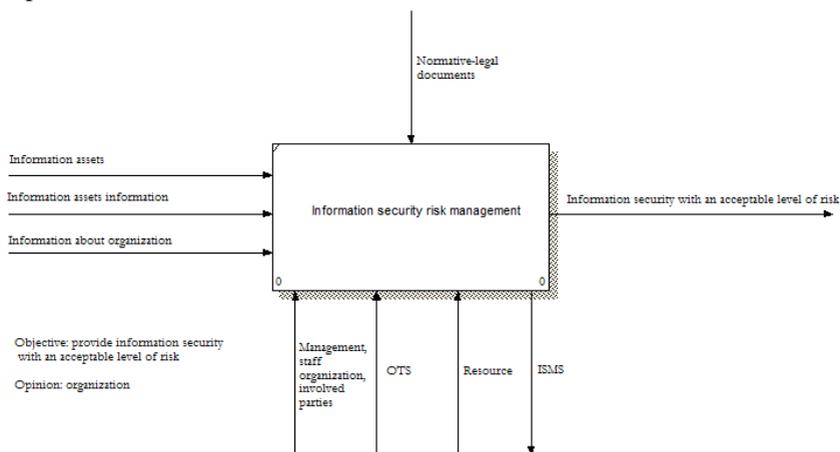


Figure 35.1. Functional model of information security risk management system

Given rice 1, a risk management system is developed and implemented in an organization to provide information security with an acceptable level of risk. This is achieved by establishing information security requirements in an organization and, consequently, by building an effective information security management system [4, 19]. While the risk management system is its fundamental basis and an integral part of all activities related to the provision of information security in the organization [4].

For this purpose, flows of material and information objects, the transformation of which is carried out by the functional block in Fig. 1. In this case, the material objects are the information assets of the organization, which are described by the information flows, namely:

- information about information assets, for example: name, place of surname, last name, first name, patronymic, position of responsible persons;
- information about the organization used to find out the internal and external contexts of the information security risk management system. This determines its scope and limits of implementation, the criteria for risk assessment, exposure criteria and risk acceptance criteria. The impact of this determination is conditioned by the availability of information about the [4]:

- organization's strategy and policy;
- processes in organization, functions and organization structure;
- legal, managerial and contractual requirements in the organization;

- security information policy in the organization;
- location of the organization and its geographical characteristics;
- limitation of the organization's activities;
- expectations of the parties involved, socio-cultural environment.

35.1.2. Standards

However, the activities of the information security risk management limited requirements, guidelines and recommendations relevant legal documents (see [20], Fig. 35.2). Specifically, they determined these activities, imposed restrictions on the processes within it.

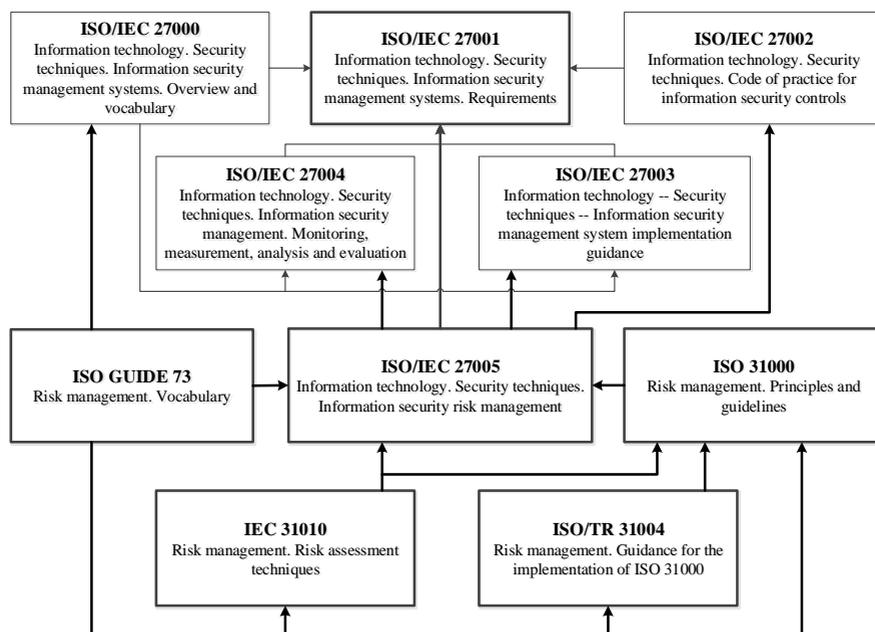


Figure 35.2. The relationship of international standards ISO 27k and ISO 31k

Taking into account the restrictions defined by the regulatory documents, information security risk management is based on information assets and organization through (a) *mechanisms* (see Fig. 35.1 [1-5, 8]):

- management, personnel of the organization and related parties involved in the development and implementation of the information SRMS or related to this activity (the parties involved);

- organizational and technical system (UTS), defined as an organizational structure and a complex of technical means (equipment) for managing the information security risk;

- resources used to manage information security risk, for example [5]: competence, experience, ability and ability of the risk assessment team; restrictions on time and other resources of the organization; available budget in case of attracting external resources.

- and *a call* (see Fig. 1 [1-5, 8]) by system, which is defined to provide a relationship between a security risk management system and a information security management system.

In addition, the basic principle of functional modeling in the IDEF0 notation describes the phenomena and events associated with the operation of the information security risk management system [1-5, 8]. This classification simplifies determining the functional boundaries and contributes to generating uniform approaches and methods of modeling the designated system security information [7, 8]. This is achieved by dividing the functions into two groups: primary and secondary. Within each of these groups, the transformation class classes are determined for their display. As a result, the ratio of hierarchical top-down submission (see Fig. 35.3) is obtained: activity – sub-activity – process – subprocess [3, 7, 8]. Due to it is possible to establish a correspondence between the functions and mechanisms for their implementation. In this case, the mechanism can be interpreted as an organizational and technical structure. At the same time, one of the main principles of functional modeling is the “separation” of the organization from functions [7, 8]. However, there is a correspondence between the hierarchy of functions and the hierarchy of mechanisms, as shown in Fig. 35.4 [8]. In this case, the information security risk management system is modeled without focusing on the organizational and technical system, but with the possibility of establishing the correspondence between the elements of the functional model and the objects of the organizational and technical structure. It is considered as a result of functional modeling of the ISRMS [7, 8].

35.1.3. Functions and structures

The results of functional simulation synthesized ISRMS through its vidobrazhannya separate functional blocks in IDEF0. With this feature set to the upper level system as the designated risk management. This feature defines the activity of ensuring information security with a reasonable level of risk in the interest of the organization on the basis of information about it and its information assets through the mechanisms and call with the limitations imposed by the relevant legal documents.

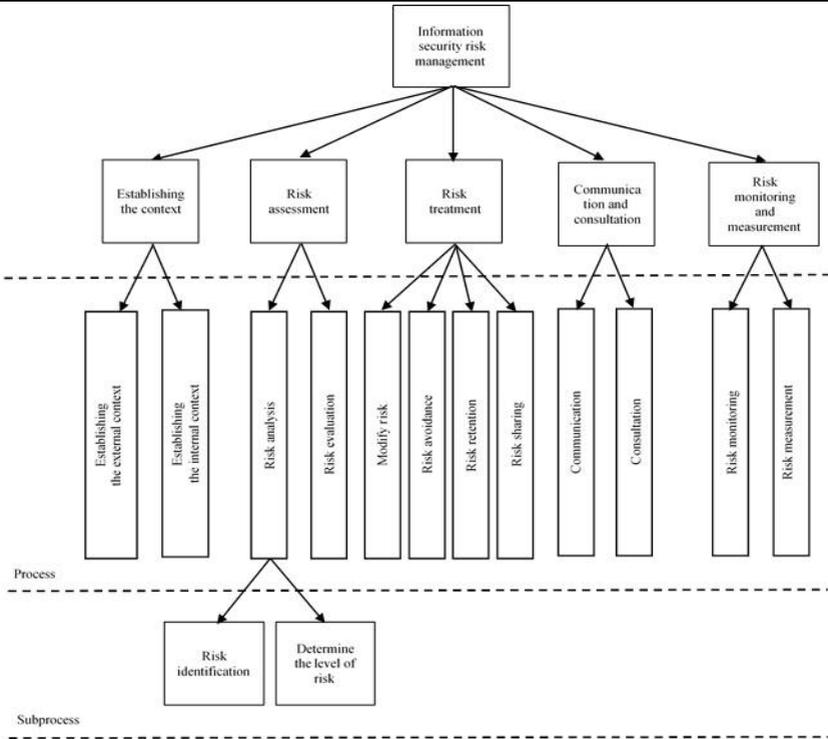


Figure 35.3. Classification of functions of the information security risk management system

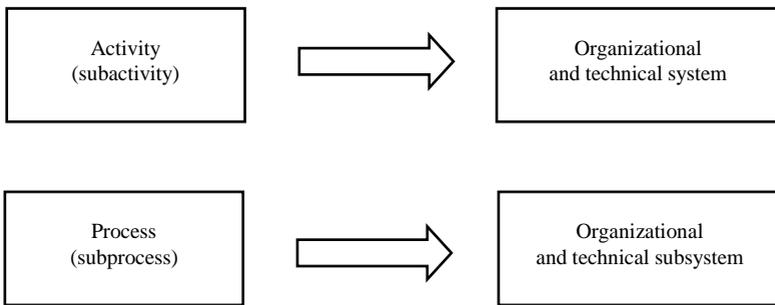


Figure 35.4. Correspondence between functions and organizational and technical structures

To ensure information security, mature companies are implementing the information security management system (ISMS), which are under

construction, as a rule, based on the requirements set out in the international standards ISO/IEC 27k series. In particular, in the standard ISO/IEC 27001:2013 “Information technology. Security techniques. Information Security Management System. Requirements” [21]. We note at once that none of the existing standards contains specific methods for the formation of project requirements for ISMS (that is, for a specific organization). At the same time, often we are talking about some or other aspects of information security (IS), which should be implemented throughout the organization or in relation to a specific business process. In this case, in order to understand what are the most important aspects of what features should be implemented when creating ISMS desirable to have some formal model of this particular ISMS. Investigating the parameters of such a model can give an understanding of which aspects of the IS need to pay close attention, and which aspects are not fundamentally important. In order to understand what kind of formal model the ISMS can have, which of the formal methods are applicable to the ISMS modeling, it is necessary to establish which of the types of formal systems it is similar to. If analogy can identify, then it can be assumed that the formal design techniques known for systems-analogues, will be able to ADAPT Rowan to problems establishing an ISMS.

In order to identify the most common analogies between ISMS and known formal systems, we will examine in more detail the basic qualities of ISMS. According to [21] Information Security Management System – is “that part of the common organization of the control system, which I based on a risk assessment. It, as part of the overall management system, creates, realizes, operates, monitors, revises, accompanies and improves information security”. From this definition it follows that everything and any ISMS can be considered as a class of systems designed to reusable solutions of the same, in a certain sense, tasks. This interpretation suggests the analogy between the ISMS and the Queuing System (QS), in which the requirements for the work performed are manifested in the form of information security events.

Note that in the general case, the sequence of service requirements that have the form of events/information security risks is random, both in terms of the occurrence of events/risks, and in the type of such events / risks. The randomness of the sequence of events/risks served by the ISMS is another aspect of the analogy between ISMS and QS.

According to ISO/IEC 27001:2013, all information security events can be broken down into separate groups depending on which points of Annex A of the standard [21-22] they are implemented. In particular, it can be, for example, events in the IT infrastructure of the organization, the facts of unauthorized crossing of the perimeter of security, personnel problems, non-compliance with certain regulations, emergency incidents, etc. Processing of information security events related to each of these individual groups are

involved, usually a specially trained team of specialists, and sometimes – external organizations, up to the law enforcement agencies. Each of these individual commands can be viewed as a separate event / risk management channel specializing in the events / risks of a certain group, but, in principle, capable of servicing events / risks related to other groups. Thus, we see the availability of processing channels for requirements, and this is the essence of another analogy between ISMS and QS.

In this case, we can say that the ISMS have QS, where the requirements for work performed are shown in the form of occurrence of information security risk, but the essence of work – maintenance of these risks in accordance with the recommendations of ISO/IEC 27k series of standards.

The service is understood in the sense that the ISMS assesses the level of emerging risks and processes those for which the risk assessment exceeds the preset threshold. All other events are documented, but the ISMS does not go into the processing state. In other words, the ISMS simply ignores such events. If, as a separate, let's say zero-event information security, consider also the fact of the absence of any information security events, then it is obvious that such a zero event does not require any processing, that is, it is ignored. Thus, we can state that the risk management mechanism in ISMS should serve all incoming risks, but it assumes two non-overlapping classes of service states: processing and ignoring. In principle the ability to cater for any ISMS risk is even on the bottom of the analogy between ISMS and QS.

We have a number of analogy between ISMS and QS is sufficient for detection of possible interpretations ISMS as QS. In connection with the observed analogies, as the ISMS projections on SMO, we will try to identify analogies in the mapping of SMO to ISMS.

By definition, [23], queuing systems – systems are those in which at random times of the demand for work performed, services. At the same time, the applications received are serviced by means of the available service channels. Consider QS composition, its generalized functional model (see., E.g., [24], Fig. 35.5), as well as the possibility of its interpretation ISMS context. The structure of QS includes applications generator, controller, service node service channels, terminator (failure node applications shredder) and queue.

1.Query generator in QS – an object that generates the application [25-28]. For ISMS generator applications are the exogenous factors (clients, contractors, suppliers, the legislator of the host country, criminals, law enforcement agencies, regulators) and/or endogenous factors (personnel, requirements of internal orders, standards, corporate requirements, interrelations between organizational units, failure or malfunctioning of equipment, etc.) inducing risk, each of which needs maintenance (process or ignore).

2.The queue in the QS – this is a certain mechanism of accumulation of applications, which are built in sequence [25-28]. The rule of forming the sequence of applications is determined by the discipline of the queue.

In the case of an ISMS also provide for a mechanism for the formation of a queue that stores events/risks of information security in line for service. Rule queuing performs the priority application (in the role of priority is the level of risk).

3.Denial of service. It is known [27-28] that QS divided into two classes: QS “refusals” and QS “queue”. In the QS with “denials” the application received at the time when all the service channels are busy is refused, leaves the system and does not participate in the further service process. In the case of ”QS with queue”, the application, which has caught all channels busy, does not leave the system, but goes to the queue and waits until the corresponding channel is freed. The refusal of service in the case of “QM with a queue” can occur, for example, by limiting the length of the queue or the time spent in the queue.

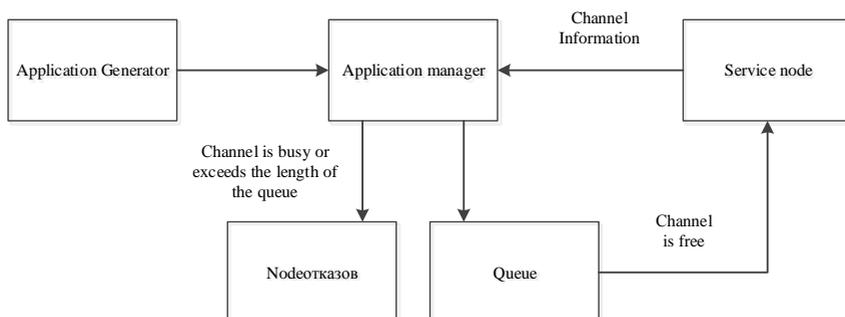


Figure 35.5. Composition and generalized functional model of the QS

The ISMS can not limit the waiting time for servicing, because it is necessary to consider all the risks of information security, which became known to the system [21]. It is unacceptable for ISMS to deny service in connection with the employment of service channels (employment of employees of the IB unit). Thus, we can conclude that the ISMS is a class QS with unlimited queue, then there are no restrictions nor on service time, or the length of the queue.

4.Manager at QS – a decision-making mechanism, required in connection with the service application [6]. Dispatcher in QS:

- accepts applications;
- receives information from the service node about free / busy channels;
- sends applications to the service channels, if there are free channels;

- Forms a queue if the channels are busy;
- monitors the time of the system;
- forms service failures.

In general, the ISMS should be implemented a certain mechanism, the task of which is the reception of applications for maintenance. Depending on the assessment of the level of risk, the dispatcher forms a queue of applications. After that, it distributes the requests between channels (performers, as are the staff that IB units) and controls the temporal characteristics of the service application is a report on the applications received, has to-date information on employment employees that service application.

Moreover, in the international standard ISO/IEC 27035-1:2016 "Information technology. Security method. Management of information security incidents. Part 1. Principles of Incident Management" is defined the need for a Team to respond to information security incidents, which is defined as a team of appropriately qualified and trusted members of the organization that handles information security incidents in the course of their life cycle, [30] and, it is the controller of the ISMS (HelpDesk/IB staff) in the case of the classification of events and how the incident (which is realized or realized the risk of information security), initiate active work began this team.

35.1.4. HelpDesk System

One example of implementing a dispatcher in an ISMS is HelpDesk (Service Desk). HelpDesk (Service Desk) are designed to automate the processing of customer requests. It is more convenient for most customers to receive support via e-mail or on the supplier's website, so most HelpDesk systems are currently online, or they provide a client-side web interface.

The main component of any HelpDesk-solutions (see., e.g. [5], Fig. 35.6) is a query management system (or incident, ticket, bugs).

When the request is received from a client (via phone, e-mail, via the web-site), the system automatically creates a "ticket", which, depending on its content and importance is put in place one of the support staff and/or IB . The employee of the support service and/or IB and he already works with the client to solve the problem. During this process, the ticket status is updated, and the head of support services and and / or IB can monitor both support staff to cope with the load.

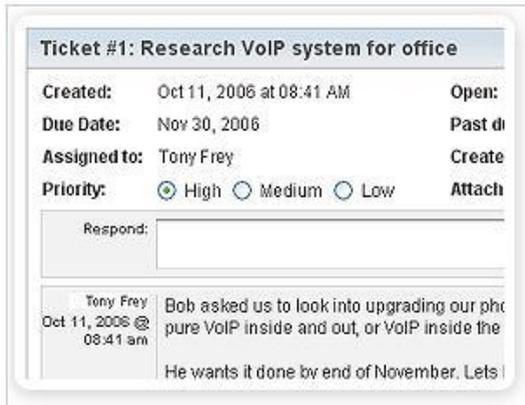


Figure 35.6. An example of using the HelpDesk query management system

In addition to the ticketing system, the HelpDesk system can include the following additional components:

- client base;
- a knowledge base for finding ready solutions;
- web-portal for clients (where they can create queries and monitor their status);
- base of service agreements (SLA = Service Level Agreement);
- base of products.

In addition to customer support, Service Desk systems have been widely used in IT departments of large companies that use Service Desk for managing IT infrastructure (ITSM). Therefore, many Service Desk systems contain specialized functions for IT management, and as a consequence, addressing issues related to IS:

- accounting configurations (catalog of IT resources of the company, their versions and settings);
- accounting problems (problems – are repeated incidents);
- accounting and exert them (for example, versions of software updates) [25].

Thus, the general understanding of the tasks entrusted to the Manager of the ISMS, and analysis of examples of implementation of ISMS Manager in specific systems, gives every reason to believe the existence of a complete analogy between the QS and dispatcherom Manager ISMS both in executing the task and the work sequence.

1.The maintenance node. In the SMO, the service node solves the problem of converting the client’s input request into the result of the client’s wishes. The maintenance node may consist of one serving device (single-channel SMO) or several (multi-channel SMO). If the number of serving units

is greater than one, then the order of their location must be indicated. So, if the servicing units execute in parallel the processing of several requirements simultaneously, then it is a multi-channel QS. If the process of servicing requirements consists of several stages that are executed consecutively one after another on different servicing units, then this system is called multiphase. Each channel has three states: available, busy not work [26, 28].

The ISMS services exist to address the problem of determining the node, whether the event is information security risk if – yes, - evaluating the level of information security risks and taking a further decision on the appropriateness of the treatment of risks in order to bring them to an acceptable value. As you can see, the process of risk processing in ISMS consists of several stages, executed successively one after the other, which is analogous to the operation of multiphase SMO.

Having considered the structure of QS its generalized functional model, and setting a number of structural and functional similarities, discuss the possibilities of interpretation of the main characteristics ISMS in the QS context.

As noted, for example, in the works [26-29], the basic characteristics of any kind of queuing systems are:

1. Incoming flow of requirements (requests) for maintenance.
2. Discipline of the queue.
3. Service mechanism.

For each of these characteristics, consider the analogies between QS and ISMS.

2. Incoming flow of service requirements (requests)

As stated, for example, [28], the study of systems of queuing incoming service requests flow is generally considered a Poisson with intensity λ . This means that requirements are received at random times, and the probability of the appearance of one requirement in the range from $t + \Delta t$ is $\lambda \Delta t$ and does not depend on t , and the probability of occurrence of two or more requirements in this interval is negligible. These assumptions are quite reasonable for many practical cases, in particular for information security events such assumptions are the limiting case dynamicheskog notion of sets of actual threats [31] at $\Delta t \rightarrow 0$.

When this function is as again noted in [26, 28], the length of service of individual applications may be assumed to be random, I exponential distribution law and the average service time $1/\mu$, where μ - intensity of maintenance. This means that the probability of the end of the service of the next application in the interval from t before $t + \Delta t$ does not depend on t and is $\mu \cdot \Delta t$.

Thus, in order for the context of the ISMS creation to be meaningfully interpreted as a queuing problem, it is necessary to determine the probabilistic characteristics of the input risk stream. At the same time, the input flow of risks will be referred to as a random sequence of risks arising at the input of the ISMS due to the occurrence of the corresponding random events of information security.

3. Discipline of the queue

Discipline in the queue queuing system is the set of rules governing the formation, movement and decay queue [26, 28]. QS queue discipline in principle determines, in accordance with which input to the system serving the requirements of connect och before a service routine.

All the principles of organization of the discipline of the queue can be divided into groups:

1. The first group – selection of applications from the queue in order of submission.

2. The second group - the choice of applications on the basis of additional information about the time of the assignment or the application process. Each application received in the system must carry the information about the necessary time for its maintenance.

3.The third group - selection of applications is carried out on the basis of the calculated remaining time of stay in the system.

4.The fourth group - selection of applications from the queue is carried out in a random order.

5.The fifth group implements obsluzh Ivan application interruption, ie application, currently being serviced and located in the channel of service, it can be removed from service, and the channel will be provided to the other application.

Depending on the specific features of the ISMS, any of these disciplines of the queue is possible. But, in addition, other disciplines, in particular, in some types of CDM selection of service requests made to certain priority criteria [26-29]. In ISMS, the level of IS risk is the analogue of priority (the higher the risk level, the higher the priority), and the priority sorting analogy is the process of ranking the IS risks depending on their level, the need for which is defined in clause 6.1.2 e2) of the international ISO/IEC 27001:2013. In extreme cases, when implemented by a threat can cause a lot of damage can be implemented option to interrupt, that is a big part of the forces of the IB division begins to engage in newly emerging risks, putting in question the risk for later.

Thus, for ISMS queue discipline is the same as for the QS, binding characteristic. For queuing discipline used in ISMS group of principles of organization of service applications, which in the case of receipt of the

application at risk of service interruption allowed the service process application (risk) with a higher level (priority).

4. Service mechanism

Service mechanism defined mainly characteristics maintenance procedures.

The characteristics of maintenance procedures include:

- number of service channels (N);
- duration of maintenance procedures (Bp probability distribution Yemeni service requirements);
- number of requirements satisfied as a result of execution of each such procedure (for group applications);
- probability of service channel failure [26].

The results of comparison of the characteristics and service procedures QS ISMS to identify and taxes are shown in Tabl. 35.1.

Table 35.1 – Comparing characteristics SMO and maintenance procedures ISMS

Features SMO maintenance procedures	Features ISMS maintenance procedures	Conclusion
Number of maintenance channel (N)	The number of experts involved in the processing of information security risks	coincides
The duration of maintenance procedures (the probability distribution of the service time requirements)	The processing unit information security risk	coincides
Number requirements satisfied as a result of execution of each such procedure (for group applications)	Applications come discretely, about the risks of information supplied alternately, but every risk can be directed to the violation of one of or more of the properties of the IB. Thus, the number of requests equal to the number of information security properties at risk to Otori entered the service	coincides
The probability of failure of the service channel	The probability of failure Channel Service (illness employee, equipment failure)	coincides

Table of consideration. clear complete coincidence service mechanism QS ISMS and that is reflected in coincidence characteristics maintenance

procedures of these two types of systems, as well as shows complete coincidence and structure of the serving system, which applies different working methods.

It should be noted further that the service time of the application (risk treatment) depends on the nature itself of the application or customer requirements (the amount of risk, the time required for carrying out measures for its handling) and the condition and capabilities serving system (IS units). In some cases, you must also consider the likelihood of failure of the service channel after a certain limited period of time. This characteristic QS can be modeled as a failure stream having priority over all other applications. The same reasoning is fully applicable to the ISMS.

35.2. Risk assessment of Security Management Systems

Construction and use of information security management systems (hereinafter – the ISMS) in modern companies, and especially those whose operation depends on the stable operation of information technology or other critical infrastructure (banking, software development companies, etc.) – is the need of the hour. As the company says, we have already implemented ISMS, and have the experience of its operation more than one year, as well as consulting companies, which are engaged in the provision of services for the construction of the ISMS, the benefits of the organization, which operates the system, significant [32, 33]. These benefits include aspects such as the exclusion of unacceptable risks, optimization of information security costs (IB) through more efficient use of available resources, improving awareness and handling processes to ensure information security. The benefits of the introduction ISMS are also [34]:

- clarity of information assets for the company's management;
- efficient implementation of the security policy (finding and correcting weaknesses in information security system);
- regularly identify security threats and vulnerabilities to the existing business processes;
- calculation of risk and decision based on the business goals of the decisions;
- effective management of enterprise in critical situations;
- demonstration of transparency and purity of the business before the law, due to the relevant standards;
- reduction and optimization of security support costs;
- integration of information security subsystems into a common management system;

- demonstration to customers, partners, business owners its commitment to information security;
- international recognition and enhance the image of the company, both domestically and in foreign markets.

Modern organizations, building at ISMS, guided, as a rule, to the international standard ISO/IEC 27001:2013 “Information technology. – Methods of security. – Information security management systems. – Requirements” [34]. This standard determines the feasibility of using a risk-based approach to the overall management of information security and, in particular, arising from his demands to the construction of the ISMS. In order to clarify the requirements for risk management in the context of building an ISMS within the group of standards ISO/IEC 27k series adopted the international standard ISO/IEC 27005:2011 “Information technology. – Methods and security features. – Information security risk management” [35]. In it, in particular, predetermined, that “the risks must be identified, quantified or qualitatively described and arranged in accordance with the priorities according to the risk evaluation criteria and for relevant organization purposes”.

For the formation of the correct design and construction requirements for an ISMS is an important given in the standard definition of risk: “Risk is a combination of the effects arising from the adverse event and the probability of occurrence of an event. ”In particular, if such a combination takes the form multiplicative, equation for calculating the risk level can be written in the following form:

$$R = H \cdot p,$$

where R – the level (magnitude) of risk, H – estimate of the consequences (damages), resulting from an undesirable event that (we are talking about the consequences of) in the case of information security events take the form of damage, p – the risk of information security events. Sometimes this probability p is the probability of the implementation of information security threats, or simply the probability that a threat.

Obviously, based on the relationship (1) can form a trivial risk ranking criterion. But, moreover, it is possible to assume that based on the relation (1) and the concept of an acceptable risk $R = R_0$ can be determined probabilistic criteria and its value set as a design requirement in the construction ISMS (immediately specify that a probability criterion can not be installed obvious relation $p = R_0/H$, since the value H is unknown). For this purpose the idea of an approach that uses a so-called “risk maps”, which allow “risk owners” to set acceptable levels of risk $R = R_0$ and share the risks on acceptable and

unacceptable spending on “risk maps” lines corresponding to $R = R_0$. This approach is described in ISO/IEC Standard 27005:2011 [36], where in the risk map is represented as a two-dimensional table, wherein the cells at intersections of respective rows and columns comprise respective risk value. At the same time, the risk value is estimated, for example, on a scale from 0 to 8 (see Tabl. 35.2).

Table 35.2 – Example risk scale

The probability of an incident scenario	very low	low	Average	High	Very high
Power exposure					
Very high	4	5	6	7th	8
High	3	4	5	6	7th
Average	2	3	4	5	6
low	1	2	3	4	5
very low	0	1	2	3	4

An example of the implementation of this campaign is observed, in particular, in the Guidelines for the establishment of an ISMS and risk assessment of the National Bank of Ukraine [37].

Thus as the obvious default send assumed that risks are categorized as acceptable, the ISMS should be processed in “automatic” mode and without the use of organizational and administrative measures and/or without additional resources. In such cases, the system operator must act according to the protocol, so to speak, “not including intelligence” – just routine work of technical support team. It is only in cases of events, a risk that exceeds a predetermined acceptable level, or when the manifest risk of accumulation effect, should engage in the work of the risk manager, and sometimes the response team on information security incidents [38] Formed a risk treatment plan to attract additional resources, both human and financial, sometimes outsourced organization.

However, it should be noted that the “risk map” operate on single events manifestations and do not consider their possible re (multiple) displays. Accumulation effects set of events, each of which falls within an acceptable, zone could lead to damage higher than the one that is associated with each of the components of a given level of risk, even without such a phenomenon as the risk of provocation one another. All this leads to the realization that the

level of acceptable risk of a single event can not be used as a valid project requirements for the construction of the ISMS. In other words, the currently existing methods of building ISMS are unable to transform an acceptable level of risk defined by the owner in the correct formal requirements for building ISMS. Even if such requirements are formally nominated, there is no answer to the question how to make sure that the ISMS built from the requirement to ensure a given level of risk, ensure compliance with this requirement.

The thesis set out in the preceding paragraph, the conclusion of the non-constructive design requirements for an ISMS based on the concept “to ensure the level of risk is not higher R_0 . “In our view, the correct design requirements should be worded differently, namely as follows: ISMS should be created to function as a queuing system, which enables the processing flow of risk events, risk level $R \geq R_0$ and a given probability P_0 of occurrence of such events.

To substantiate the correctness of such a requirement is necessary to demonstrate the possibility to determine for a given value of acceptable risk $R = R_0$ value of the probability P_0 with which manifest events associated with risk $R \geq R_0$.

In other words, you need to show solvability of the following problem: for a given acceptable level of risk $R = R_0$ is necessary to estimate the probability of P_0 occurrence of the event risks $R \geq R_0$. The dual formulation of the same problem: for a given level of acceptable risk $R = R_0$ to estimate the probability P_1 with which events can occur with the risks $R < R_0$. It is obvious that $P_0 + P_1 = 1$.

Probability estimates P_1 can be made using the concept of probability and geometrical methods [39]. First of all, we introduce the two-dimensional Cartesian coordinate system, the horizontal axis where we plot the probability values p , and the vertical axis – the value of damage H . Obviously, the probability values range from $p=0$ before $p=1$ and the value of the damage in the range of $H = 0$ up to $H = H_{\max}$. For uniformity, the range of change of probability of damage to the range change, we introduce the normalized magnitude of damage

$$h = \frac{H}{H_{\max}} .$$

Then a normalized value of the damage will vary in the range of $h = 0$ (at $H = 0$) to $h = 1$ at $H = H_{\max}$.

In Cartesian coordinates ($h0p$ “unit square” is defined) $OACE$ (see Fig.35.7), the locus of the points corresponding to any possible risk values normalized r :

$$r = h \cdot p. \tag{2}$$

Where r subject to the condition $0 \leq r \leq 1$ due to the conditions $0 \leq h \leq 1$ and $0 \leq p \leq 1$.

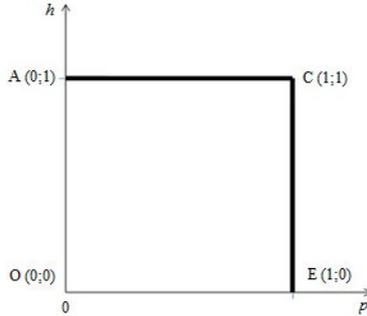


Figure 35.7. The locus of the points of any possible risk of normalized values

$$r = h \cdot p$$

Since the length of each side of the square $OACE$ is equal to one, then the area of the $S_{o\tilde{o}u}$ square $OACE$ is equal to 1:

$$S_{o\tilde{o}u} = 1 \cdot 1 = 1.$$

We define a normalized level of acceptable risk $r = r_0$. From the relation (2) must obviously functional relationship

$$h = r_0 \cdot \frac{1}{p}. \tag{3}$$

graph is a hyperbola $h = (1/p)$, the coefficient is shifted r_0 from the origin (0,0) toward point coordinates (1,1). If we impose a hyperbola $h = (1/p)$ on the unit square $OACE$, the locus of points of all the risks divided into two subsets (see Fig. 35.8), Namely: figure $OABDE$ defines the locus of points of risk values for which the ratio $r < r_0$, while the figure BCD defines the locus of points of the set of values risks for which the relation $r \geq r_0$.

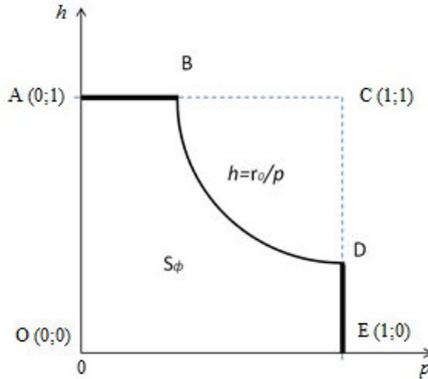


Figure 35.8. The locus of the points of risk values divided hyperbola $h = (1/p)$

In such case the probability P_1 that the value of an arbitrary normalized risk $r = r_0$ will not exceed a predetermined level normalized risk r is determined by area ratio figures $OABDE$ for the area “unit square” $OACE$

$$P_1 = \frac{S_\phi}{S_{o\phi u}}, \tag{4}$$

Where S_ϕ – area of the figure $OABDE$, and $S_{o\phi u}$ – area “of the unit square”. As has previously been shown that $S_{o\phi u} = 1$, the relation (4) becomes:

$$P_1 = S_\phi. \tag{5}$$

Thus, the probability P_1 that an arbitrary risk is the condition $R > R_0$ equal to the square shape $OABDE$. It remains to calculate the area of this figure.

For this figure we will divide $OABDE$ into two parts (see Fig.9.): Part one – the figure $OABG$ with area S_1 and part of the second – the figure $GBDE$ with area S_2 . It's obvious that

$$S_\phi = S_1 + S_2. \tag{6}$$

The area S_1 is calculated as the area of a rectangle with sides OA and AB . Side length OA as previously caused, is equal to 1. A side length AB is determined by the numerical value of the coordinates of probabilistic B . The point B is the intersection point of the line $b = 1$ with the hyperbola defined by (3). Then the numerical value of the coordinates of the probability B can be determined by substituting the value of $h = 1$ the left side of relation (3):

$$1 = r_0 \cdot \frac{1}{p}.$$

From this relation it follows that the numerical value of the probability coordinate $p = p_0$ point B is: $p_0 = r_0$.

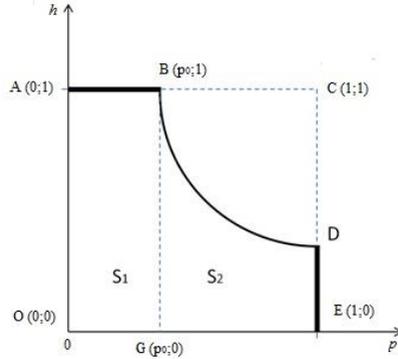


Figure 35.9. Splitting the figures $OABDE$ on the two figures: a rectangle $OABG$ and a figure $GBDE$

Then the area S_1 can be expressed by the following relation:

$$S_1 = 1 \cdot r_0 = r_0. \quad (7)$$

The area of S_2 the second figure $GBDE$ which is formed by a hyperbola given by the relation (3) and three lines: $h=0$, $p=p_0=r_0$ and $p=1$, Definite integral is calculated as the following formula:

$$S_2 = \int_{r_0}^1 \frac{r_0}{p} dp = r_0 \int_{r_0}^1 \frac{1}{p} dp = r_0 \ln p \Big|_{r_0}^1 = r_0 (\ln 1 - \ln r_0).$$

Because the $\ln 1 = 0$, then the formula for calculating the area S_2 takes the following form:

$$S_2 = r_0 (\ln 1 - \ln r_0) = -r_0 \ln r_0. \quad (8)$$

Then, for calculating the square shape $OABDE$ substitute in (6) the values (7) and (8) we obtain:

$$S_{\phi} = S_1 + S_2 = r_0 - r_0 \ln r_0 = r_0 (1 - \ln r_0). \quad (9)$$

Thus, (5), a formula is obtained to estimate the probability P_1 that the normalized values of the possible risks will not exceed a predetermined value acceptable risk r_0 :

$$P_1 = r_0 (1 - \ln r_0). \quad (10)$$

Analyze the obtained relation.

Firstly, since for values of r_0 the condition $0 \leq r_0 \leq 1$ insofar function in equation (10) takes negative values $\ln r_0$. Due to this, the subtracted value $(-r_0 \ln r_0)$ is converted into a positive term in formula (10). To reflect this fact explicitly formula (10) can be represented as follows:

$$P_1 = r_0 (1 + \ln(r_0^{-1})). \tag{11}$$

Example position graph of this function with respect to the line graph $P = r_0$ shown in Fig.35.10.

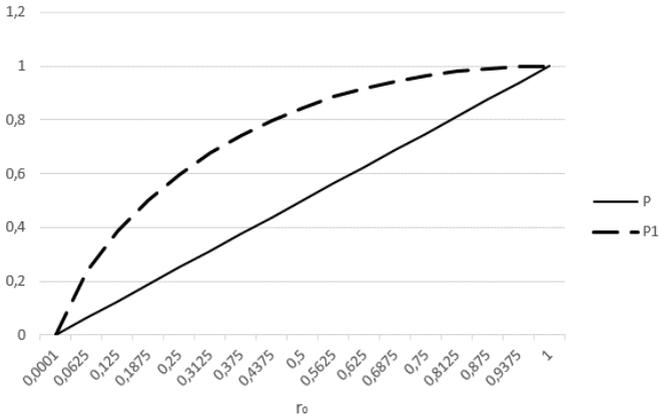


Figure 35.10. Position graph of a function $P_1 = r_0(1 + \ln(r_0^{-1}))$ with respect to the graph of the function $P = r_0$

From (11) it follows that the probability P_1 with which the normalized risks may arise $r < r_0$, almost always exceeds the predetermined value of normalized acceptable risk r_0 , except for a single case $r_0 = 1$. In this extreme case, $\ln r_0 = 0$ the relation (11) becomes

$$P_1 = r_0 (1 + \ln(r_0^{-1})) = 1 \cdot (1 + \ln 1) = 1 \cdot (1 + 0) = 1,$$

and it is a formal reflection of the trivial fact that if the maximum amount of damages $H = H_{\max}$ to set as an acceptable, then the value of all the risks are acceptable.

Secondly, one can determine the maximum error probability replacement P_1 risk r_0 (i.e. probability $P = r_0$), as an aberration function given by equation (11) from the line $P = r_0$ by taking the following difference:

$$P_1 - P = r_0 \left(1 + \ln(r_0^{-1})\right) - r_0 = r_0 \ln(r_0^{-1})$$

Schedule function corresponding to such difference is shown in Fig. 35.11 and from it we can directly get:

1) the maximum value of the probability estimation error slightly exceeds the value of 0.36 (or rather, it is equal to 0.3678) from one of the normalized level of risk;

2) the maximum value of error is achieved in the neighborhood of the normalized values of risk $r_0 = 0.36$;

3) exceeding the level 10% of probability estimation error can be observed on the 80% possible values r_0 ;

4) the error rate exceeding a chance of 36% over 10% all values of r_0 .

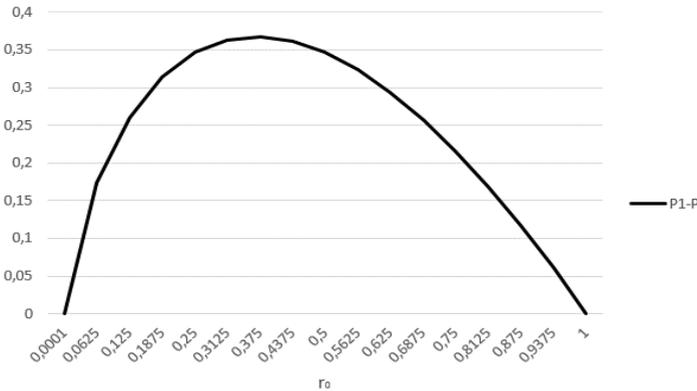


Figure 35.11. Schedule a function of the difference $P_1 = r_0 \left(1 + \ln(r_0^{-1})\right)$ and $P = r_0$

Thus, the use of the geometric approach to the estimation of the probability P_1 that the arbitrary values normalized risk r of information security threats will fall into the zone $r < r_0$, has made it possible to obtain accurate quantification of this probability as a formula (11). As a result, it was found that the probability is P_1 almost always higher than the level r_0 . In most cases, this difference reaches 30%, and over 10% all the cases, the difference is even slightly higher than 36%.

Thus, the use of the geometric approach makes it possible to transform a subjective measure of risk appetite risk of the owner displayed in the form of an acceptable level of risk in the formal probabilistic criterion, based on which we can formulate testable requirements for the establishment of information security management systems.

Currently, information security risk is interpreted as the impact of uncertainty on the achievement of objectives. Achieving goals means ensuring confidentiality, integrity and availability of information. It is necessary to take into account various factors. Given their diversity, it should be noted that uncertainty in these factors is greater than statistical certainty. Therefore, the estimation of uncertainty is more correct, in contrast to the probability of a threat to the security of information. Since entropy is a measure of uncertainty, it is proposed to use entropic approach to information security risk assessment [32].

The idea of using entropy for risk assessment is known. Some of its provisions *ysya* expressed, for example, [33-36]. In particular, entropy risk measures have been investigated in the formation of a portfolio of securities and experimentally established the value of the parameter of measure at which the best effect is achieved [33]; methods of definition of entropy in the estimation of market risks are described [34]; As discussed entropic combination of financial risks as a convex combination entropic risk measures and measures CVaR and conducted analysis of proposed efficiency measures [35]; the overcoming of the problem of forming a portfolio of measures for the modernization of organizations for minimizing economic risks on the basis of their information-ethernet model [5]; the decision support system for managing the portfolio of securities based on entropy risk measures is presented [37].

However, analysis of available sources and materials on the Internet showed that using entropy approach to the definition of “information security risk” was first proposed in [32] and developed in [38].

Suppose that for some objects in *A* a priori known set of *n* threats to information security and a well-ordered set of *m* damage states due to the implementation of these threats:

$$x_1, x_2, \dots, x_i, \dots, x_m, \\ i = \overline{1, m}.$$

Obviously that $n \leq m$. This indicates the existence not identical information security threats that lead to the same loss. An example of this may be due to the threat of CEA ation which losses are zero.

In addition, when ordering it is understood that

$$0 \leq x_1 \leq x_2 \leq \dots \leq x_i \leq \dots \leq x_m \leq x_{\max},$$

where x_{\max} – the maximum loss is equal to the full liquidation of all information assets of *A* for an infinitesimal time interval without any residues. Also consider memo, known probability distribution p_i on plural x_i , namely loss

x_1 arises with probability p_1 , damage x_2 – with probability p_2 , damage x_i – with probability p_i .

Given this, a full set of events $x_1, x_2, \dots, x_i, \dots, x_m$ let's call such a set of damage states that one of them will necessarily occur as a result of the threat of information security. Because the states of damage $x_1, x_2, \dots, x_i, \dots, x_m$ a complete set of events is given with their probabilities

$$p_1, p_2, \dots, p_i, \dots, p_m,$$

$$p_i \geq 0, \sum_{i=1}^m p_i = 1,$$

it is considered a given for intsevu scheme

$$X = \begin{pmatrix} x_1 & x_2 & \dots & x_i & \dots & x_m \\ p_1 & p_2 & \dots & p_i & \dots & p_m \end{pmatrix}.$$

With the help of the final scheme, we describe the state of uncertainty in ensuring the confidentiality, integrity and availability of information. The degree of uncertainty is different for different schemes. Therefore, to estimate the degree of uncertainty of a given finite scheme, an entropy measure is used

$$H_A(p_1, p_2, \dots, p_i, \dots, p_m) = - \sum_{i=1}^m p_i \lg p_i,$$

where $H_A(p_1, p_2, \dots, p_i, \dots, p_m)$ – entropy ultimate scheme (see. Fig. 12), which is proposed to be used for risk assessment of information security facility A [39]. If one of the values and probability equal to unity, then the function $H_A(p_1, p_2, \dots, p_i, \dots, p_m) = 0$. This year ent m corresponds to the case where the head out on time can provide the implementation of information security threats with certainty and, consequently, lack of uncertainty. Then as a fixed one m max uncertainty equally probable ultimate scheme of threats.

In addition, the correct use of entropy approach to information security risk assessment is confirmed by the following properties of entropy [40,41]

1. Size

$$H_A(p_1, p_2, \dots, p_i, \dots, p_m) \geq 0.$$

This means that the information security risk can either be equal to or greater than zero, and therefore can not be negative. This explains the difference security of commercial risks, where there may be a negative risk equivalent income losses versus a positive risk.

2. If the final circuit of two objects A and B

$$X_1 = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1i} & \dots & x_{1m} \\ p_{11} & p_{12} & \dots & p_{1i} & \dots & p_{1m} \end{pmatrix},$$

$$X_2 = \begin{pmatrix} x_{21} & x_{22} & \dots & x_{2j} & \dots & x_{2n} \\ p_{21} & p_{22} & \dots & p_{2i} & \dots & p_{2n} \end{pmatrix}$$

capacious mutually independent,

$$H_{AB}(X_1 X_2) = H_A(X_1) + H_B(X_2).$$

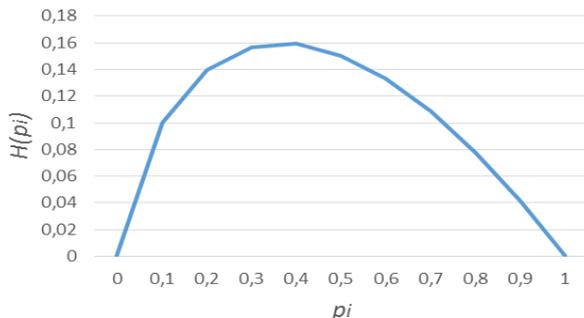


Figure 35.12 – Graphical display depending entropy ultimate scheme

As a consequence, the risk of information security of two objects *A* and *B* is equal to the sum of the risks of each object. This is consistent with an intuitive understanding of the risk of information security.

3.If the final circuit of two objects *A* and *B*

$$X_1 = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1i} & \dots & x_{1m} \\ p_{11} & p_{12} & \dots & p_{1i} & \dots & p_{1m} \end{pmatrix},$$

$$X_2 = \begin{pmatrix} x_{21} & x_{22} & \dots & x_{2j} & \dots & x_{2n} \\ p_{21} & p_{22} & \dots & p_{2i} & \dots & p_{2n} \end{pmatrix}$$

capacious mutually dependent, the

$$H_{AB}(X_1 X_2) = H_A(X_1) + H_B^{X_1}(X_2).$$

As a consequence, the information security risk of the facility *B* decreases if the known result of the realization of the threat of the final circuit of the object *A*.

4. If two objects *A* and *B* have the same distribution of the probability of causing damage as a result of the implementation of threats, the information security risk for such objects is the same.

5. If

$$p_1 = p_2 = \dots = p_i = \dots = p_m = \frac{1}{m},$$

then the value $H_A(p_1, p_2, \dots, p_i, \dots, p_m)$ gets the most value. Interpretation of this property is: if the object was not aware of the probability of threats, the risk of

information security maximum. The use of means and measures to handle it under any threat reduces the likelihood of causing damage if the probability of zero damage improves accordingly. As a consequence, the magnitude $H_A(p_1, p_2, \dots, p_i, \dots, p_m)$ will decrease, which indicates a reduction in information security risk.

6. For two objects A and B the security of information is higher in the object, the risk of information security is less. If $H_A(X_1) > H_B(X_2)$, then the difference $[H(A) - H(B)]$ shows how much the object's security information management system is B better than the object A .

7. If the object's final pattern A complemented by an impossible event

$$H_A(p_1, p_2, \dots, p_i, \dots, p_m, 0) = H_A(p_1, p_2, \dots, p_i, \dots, p_m),$$

then entropy and, consequently, the risk of information security do not change.

Using entropy approach makes it possible to build a valid basis of quantitative risk assessment of information security. This is due to manipulation of form distribution damage, not its specific values. On the other hand, while this advantage is also a disadvantage. Since there is a need to establish forms of probability distribution losses due to the application of information security threats implementations. In addition, the need to form a complete set of implementations threats compounded by the lack of practical statistical loss. Since any negative developments in security risk should be treated immediately with a view unemozhlyvlyuvannya its recurrence in the future, resulting in failure conditions stationary observation.

35.3 Audit of Security Management Systems

35.3.1. Principles

When conducting an audit of a security management system (SMS), several basic principles must be followed. These principles will allow to make the SMS audit a useful and reliable tool for maintaining the policy of management, provide data on the basis of which the organization can improve the SMS. Compliance with these principles is a prerequisite for providing objective and sufficient conclusions based on the audit results and allows auditors working independently of one another to arrive at similar conclusions under the same circumstances.

The guidelines are based on the following six principles.

A) Integrity – is the basis of professionalism.

Auditors and persons managing the audit program should:

- perform their work honestly, diligently and responsibly;
- respect and respect all applicable legislative requirements;

- demonstrate their technical competence in the performance of work;
- do their work impartially, remain honest and unbiased in all their actions;
- Be cautious and not be influenced by any influences that other interested parties may have on their judgments or conclusions.

B) Fair presentation – is the obligation to provide truthful and accurate reports.

Conclusions of audits, audit findings and reports should reflect the audit activity truthfully and accurately. Unresolved problems and disagreements between the audit team and the audited organization should be reflected in the reports. The exchange of information must be truthful, accurate, objective, precise on time, understandable and complete.

C) Due professional care – diligence and ability to make the right decisions when conducting an audit.

The professional care of the auditors is related to the importance of the task performed and confidence from the audit client and other interested parties. An important factor in the performance of auditors in terms of work with professional care is the ability to make informed decisions in any situation during the course of the audit.

D) Confidentiality – confidentiality of information.

Auditors should be cautious when using and ensuring the protection and safety of information obtained by them during the audit. Information obtained during the audit should not be used improperly for personal gain by the auditor or the audit client, or in a way that harms the legitimate interests of the audited organization. Compliance with this principle includes the proper handling of confidential or classified information.

E) Independence – is the basis for the impartiality and objectivity of conclusions based on the audit results.

Auditors should be independent of the audited activity whenever feasible, and always perform their work in such a way as to be free from prejudice and conflict of interest. When conducting internal audits, auditors should be independent of the heads of departments and activities they are checking. Auditors should maintain an objective opinion throughout the entire audit process to ensure that audit findings and conclusions are based only on audit evidence.

For small organizations, it may not be possible to ensure the independence of internal auditors from the activities they audit, but every effort should be made to exclude any interest and ensure an objective examination of the activity being audited.

F) The evidence-based approach – is a reasonable basis for achieving reliable and reproducible audit findings in a systematic audit process.

35.3.2 Objectives and program

An organization that is going to conduct audits should prepare an audit program to determine the effectiveness of the SMS of the organization. The audit program should include audits covering the declared standards for safety management systems that are implemented or is being implemented in the enterprise.

Top management should ensure the development of the objectives of the audit program, in order to guide the planning and performing of audits, and the effective implementation of the audit program. Goals may depend on:

- A) identified requirements to information security;
- B) the requirements of ISO 27001 standard;
- C) quality of functioning of the audited organization, which reflects the occurrences of failures and incidents of information security to measure effectiveness.
- D) information security risks of the organization being audited.

The audit program should include the information and resources necessary to organize audits and their efficient and effective implementation within a set timeframe. It is necessary to monitor and measure the implementation of the audit program to ensure that the goals are achieved. In order to identify possible improvements, the audit program should be analyzed.

An important role in the process of designing an audit is played by the process of developing an audit program, in which it is necessary to identify the persons responsible, their competence, and the scope of the audit program. Also in the program, it is necessary to reflect the identification and evaluation of audit risks and resources. The next step is to implement this program with the subsequent monitoring, analysis and improvement of the previously developed audit program.

When developing the audit program, the person who manages the audit program should: establish the scope of the audit program; Identify and assess the risks associated with the audit program; Identify responsibilities for auditing; Determine the procedures for the audit program; Identify the necessary resources; Ensure the implementation of an audit program, which includes the definition of audit objectives, the scope and criteria of individual audits, the definition of audit methods and the formation of a group of auditors; Ensure the management and preservation of the relevant entries under the audit program; Monitor, analyze and improve the audit program [42].

The person who is responsible for managing the audit program should inform the senior management about the content and status of the audit program and, if necessary, receive its approval. At the same time, it demands

competence of persons responsible for managing the audit program. They should be competent enough for efficient and effective management of the audit program and the risks associated with it. It is necessary for the person responsible for managing the audit program to participate in activities to continually improve his professional level in order to maintain his knowledge and skills necessary to manage the audit program at the proper level.

Also, the person responsible for managing the audit program should determine the scope of the audit program, which may vary depending on the size and nature of the activity of the organization being audited, and also on the nature, functional characteristics, complexity and level of development of the audited SMS and those of its elements that are given the most important. It should be noted that the scope of the audit program can vary, and include following factors that can affect the scope of the audit program:

A) the scope of the SMS, including the total number of employees working for each facility, and relationships with third-party organizations that regularly operate at the assessed site;

B) the number of sites covered by the SMS, as well as the complexity of the SMS (including the number and criticality of processes and activities).

The audit program should focus on setting priorities based on information security risks and the business requirement for SMS areas requiring more detailed study.

When identifying resources for the audit program, the person responsible for managing the audit program should take into account:

A) financial resources necessary for the development, implementation, management and improvement of audit activities;

B) methods / techniques and tools for conducting audits;

C) the availability of auditors and technical experts with the competence required to achieve the specific objectives of the audit program;

D) the scope of the audit program and the risks of the audit;

E) travel time and costs for transport, accommodation and other organizational requirements for conducting the audit;

F) the volume and level of development of information and communication systems.

The person responsible for managing the audit program is also responsible for implementing the audit program. If necessary, the implementation of the audit program should consider the requirements for confidentiality of the audited and other interested parties, including possible legal or contractual requirements.

The basis for each individual audit of the SMS should be documented objectives, scope and criteria for this audit. They should be determined by the person responsible for managing the audit program, and be consistent with the overall objectives of the audit program.

The objectives of the audit include determining what should be done when conducting a specific audit.

The scope of the audit should reflect the information security risks that correspond to the requirements of the business, and the business risks of the audited organization.

In addition, audit objectives may include the following:

A) assessing whether the scope of the SMS is sufficiently defined and whether all information security requirements are taken into account;

B) assessing the relevance of the objectives of the SMS set by management;

C) evaluation of supporting processes and effective improvement of SMS.

The audit team should ensure that the scope and boundaries of the SMS of the audited organization are clearly defined in terms of the characteristics of the organization's activities. Its location, assets and technologies, including details and justification for any non-admission in the scope. The audit team should confirm that the audited organization in the scope of the SMS covers all necessary requirements.

Therefore, auditors must ensure that the assessment and processing of the information security risk of the audited organization properly reflect its activities and are limited to the scope of its activities. Auditors should also ensure that the interaction with services or activities that are not completely within the scope of the SMS are considered within the SMS and is included in the information security risk assessment of the audited organization. An example of such a situation is the collective use of tools (for example, IT systems, databases and telecommunications systems) together with other organizations [43].

The audit criteria are used as a basis for comparison, which determines compliance, and may include applied policies, objectives, procedures, standards, legislative requirements, management system requirements, contractual requirements or codes of rules governing information security activities.

In the event of any changes regarding the objectives, scope and criteria for the audit, if necessary, the audit program should be modified accordingly.

The person responsible for managing the audit program selects and determines methods for effective audit implementation, depending on the objectives, scope and criteria for this audit.

In the event that two or more auditing organizations jointly audit the SMS of one organization, persons responsible for managing various audit programs should agree on the method of this audit and consider issues related to the availability of necessary resources and planning for the activities of this audit.

35.3.3 Team and documentation

The audit team is formed by the person responsible for managing the audit program. It appoints members of the audit team, including the team leader and any technical experts required to audit the SMS.

The audit team should be formed taking into account the competence necessary to achieve the objectives of the SMS audit. If an auditor conducts an audit, he must perform all duties assigned to the head of the audit team. If the level of competence of auditors in the audit team is not sufficient, technical experts may be included in this group to provide the necessary competence. Technical experts should work under the guidance of the auditor, but not act as an auditor.

The responsibility for conducting a specific audit lies with the head of the audit team. This should be done in advance, so that sufficient time remains before the planned audit date to ensure effective planning of this audit.

At the same time, the person responsible for managing the audit program should manage the output of the audit program. Also, this person must ensure the creation, management and maintenance of relevant records in order to demonstrate the implementation of the SMS audit program. Processes should be established to ensure that the required confidentiality is respected for audit records.

The form and volume of information presented in the records should demonstrate that the objectives of the audit program have been achieved. Also, an audit program should be analyzed to assess the extent to which its objectives are met. Conclusions from the analysis of the audit program should be used for the process of continuous improvement [44].

The person responsible for managing the audit program should analyze the overall implementation of the audit program, identify areas for improvement, and, if necessary, amend the SMS audit program.

When the audit is started, the responsibility for conducting the audit remains with the appointed head of the audit team until the completion of this audit.

The team leader must establish initial contact with the auditee to conduct the audit. This contact can be formal or informal. The objectives of the initial contact are: establishment of communication and information transfer channels with representatives of the audited organization; Confirmation of authority to conduct an audit; Providing information related to the scope of audit, audit methods and the composition of the audit team, including technical experts; Obtaining permission to access relevant documents for planning goals and tasks, including records.

To ensure that the stated objectives of the audit can be achieved, you need to determine the possibility of conducting an audit. Before the audit

begins, you should request the audited organization to have SMS records that are not available for audit by the audit team, for example, containing confidential or critical information. The person in charge of managing the audit program should determine whether it is possible to audit the SMS in sufficient measure in the absence of these records. If it is concluded that auditing the SMS is not possible without analyzing the identified records, the responsible person should notify the auditee about the impossibility of performing the audit until appropriate access rights are granted or an alternative is offered.

When preparing for the on-site audit, it is necessary to analyze the SMS documentation of the auditee. The documentation should include, as far as applicable, documents and records of the management system, as well as reports on previous audits. When analyzing the documentation, it is necessary to take into account the size, nature of the activity, the complexity of the organization and its SMS, and the purpose and scope of the audit.

After the analysis, the head of the audit team should prepare an audit plan for the SMS, based on the information contained in the audit program and the documentation provided by the audited organization. The audit plan should consider the impact of the audit, taking into account its impact on the information security of the audited organization and provide the basis for an agreement between the audit client, the audit team and the auditee regarding the audit. This plan should promote the best coordination, consistency and timing of performance of audit work for the most effective achievement of the result.

The amount of information presented in the audit plan should reflect the scope and complexity of the audit, as well as the impact of uncertainties on the achievement of audit objectives. The audit plan can be analyzed and approved by the audit client, and it should be submitted to the auditee for review. Any objections on the part of the audited organization relating to the audit plan should be resolved between the head of the audit team audited by the organization and the audit client.

The head of the audit team of the SMS in the course of consultations with members of the audit team should identify and distribute responsibility between each member of the group for the audit of specific processes, works, functional units or areas of production activity. With such a distribution, the independence and competence of auditors and the effective use of resources should be taken into account, as well as the various roles and responsibilities of auditors, trainees and technical experts.

The head of the audit team should conduct work meetings of the audit team in order to distribute work assignments and address issues related to possible changes. In the course of the audit of the SMS, changes can be made to work assignments or performance of work in order to ensure achievement of

the audit objectives set.

Members of the audit team should collect and analyze information relevant to their area of responsibility and prepare the working documents properly for recording and recording audit evidence.

A preliminary meeting is required to conduct an audit of the SMS on-site. The preliminary meeting is held with the management of the audited organization and, where possible, with those persons who are responsible for the audited units or processes. This meeting provides the opportunity to ask questions.

The amount and degree of information provided should correspond to the level of awareness of the auditee with the audit process. In many cases, for example, when conducting internal audits in small organizations, the preliminary meeting can consist only of announcing that an audit has commenced and explaining the nature or specifics of the audit. In other cases, the preliminary meeting may be of an official nature, in which the registration of persons present at the meeting takes place. The preliminary meeting should be held under the supervision of the head of the audit team.

An important component of the enterprise SMS audit is document analysis. This analysis may be carried out in conjunction with other audit activities and may continue the implementation of audit activities if this does not adversely affect the effectiveness of the audit.

If the necessary documentation can not be provided within the timeframe specified in the audit plan for the SMS, the head of the audit team should be informed by the person responsible for managing the audit program and the organization being audited. Auditors must verify the availability of documentation and its compliance with the requirements of ISO / IEC 27001. Auditors must confirm that the selected measures and controls are related to the outcome of the risk assessment and processing process and can be subsequently tracked to policy and the entire SMS.

During the audit, there may be a need to conclude formal agreements for the exchange of information between the audit team and the audited organization, the audit client and, possibly, external bodies (for example, supervisory authorities), especially where the law contains requirements for mandatory notification About inconsistencies. In the SMS audit group, it is necessary to periodically exchange information, evaluate the audit process and, if necessary, redistribute responsibilities between members of the audit team.

If the existing audit evidence indicates that the audit objectives are not feasible, the supervisor of the SMS audit team should report to the audit client or the organization being inspected the reasons for taking appropriate action. Such measures may include making changes and re-approving an audit plan, changing the objectives or scope of the audit, or terminating the audit.

During the audit of the SMS, accompanying persons and observers (for

example, representatives of the regulatory body or other interested parties) may be present. They should not influence or interfere with the audit. In the event that this can not be guaranteed, the head of the audit team has the right to refuse observers to participate in some audit activities.

For observers, any obligations related to health, safety and confidentiality should be specified and regulated between the audit client and the audited organization. Accompanying persons appointed by the audited organization should assist the audit team and act upon the request of the head of the audit team.

During the audit, information relating to the audit objectives, scope and audit criteria, including information relating to the interaction between units, activities and processes, should be collected by necessary samples and verified. As evidence of the audit of the SMS, only the information that can be verified should be accepted. The audit evidence must be recorded. If during the collection of certificates the audit team becomes aware of any new or changed information security risks, they should be considered and taken appropriate measures.

Collecting information and evidence on the implementation and effectiveness of SMS processes, as well as measures and controls and controls is an important part of the SMS audit. Possible methods of collecting relevant information during the audit include checking information assets and SMS processes, measures and means of control and management, using automated audit tools.

The SMS auditors should ensure that all information received from the audited organization is properly treated in accordance with the agreement between the audited organization and the audit team.

Based on the results of the SMS audit, conclusions are drawn. To obtain audit findings, audit evidence must be compared and evaluated against audit criteria. Audit findings may indicate compliance or non-compliance with audit criteria. In the event that this can not be guaranteed, the head of the audit team has the right to refuse observers to participate in some audit activities.

Inconsistencies in the audit and their supporting evidence must be recorded. Nonconformities can be classified (ranked). They should be analyzed with the audited organization to confirm the objectivity of the audit evidence and to confirm that the identified non-conformities are correctly understood. All possible measures should be taken to resolve any differences of opinion on the evidence and / or audit findings, and unresolved issues should be documented.

The final meeting follows the formation of conclusions, on which they are brought. The conclusions formulated and voiced at the meeting should be clear and recognized by the audited organization. To participate in the final meeting should involve the managers of the organization being audited and,

where appropriate, the staff responsible for the functions or processes that were audited during the audit, as well as the audit client and other parties. If necessary, the head of the audit team should inform the auditee about the situations that occurred during the audit, which may reduce the credibility of the information stated in the audit conclusions. The amount and degree of information provided should correspond to the level of awareness of the audited organization about the audit process. In other cases, such as internal audits, the final meeting is less formal and can consist only of reporting findings and conclusions from the audit.

The prepared report should be approved by the head of the SMS audit team within the agreed timeframe and should be sent to the recipients defined by the audit procedures.

Audit is considered complete if all planned audit activities have been completed. Documents related to the audit should be stored or destroyed by agreement between the parties involved in accordance with the procedures of the audit program and applicable legislative and other requirements.

Conclusions based on the results of the audit may, depending on the purposes of the audit, indicate the need for corrections, corrective and preventive actions or actions to improve the SMS. Such actions, as a rule, are developed and carried out by the audited organization within the agreed time periods. If necessary, the auditee should inform the person responsible for managing the audit program and the group of auditors on the status of implementation of these actions. The implementation and effectiveness of these actions must be verified. Such verification may be part of a subsequent audit.

Trust in the audit process of the SMS and its ability to achieve the objectives depends on the competence of the persons involved in planning and conducting the audit, including auditors and audit team leaders. Competence should be evaluated through a process that takes into account personal qualities and the ability to apply the knowledge and skills acquired through training, production experience, training as an auditor and experience in auditing. This process must take into account the needs of the audit program and its objectives.

There is no need for each auditor in the audit team to have the same level of competence; And it is necessary that the overall competence of the audit team is sufficient to fulfill the audit objectives. The assessment of the competence of auditors should be planned, implemented and documented in accordance with the audit program, including procedures for obtaining an objective, reliable and relevant result.

So the auditor must have such personal qualities as ethics, openness and open-mindedness, diplomacy, observation, receptivity, universality, perseverance, determination, independence, adherence to principles, readiness

for self-improvement, high culture of behavior, ability to cooperate and work with people.

Auditors must have the knowledge and skills necessary to achieve the intended results of audits, which they will be entrusted with. All auditors must have general knowledge and skills, and it is also assumed that they will have some special knowledge and skills in specific disciplines and management branches. Heads of audit teams should have the additional knowledge and skills necessary to ensure proper management of the audit team.

The SMS auditors should have the knowledge and skills in such areas as safety management techniques that will allow the auditor to investigate the SMS and generate appropriate audit findings and recommendations. Knowledge and practical skills in this area should include the terminology of information security, the principles of security management and their application, the methods of information security risk management and their application, the general knowledge of information technology and methods of ensuring information security, existing information security threats, vulnerabilities, measures and controls And management, as well as the main organizational, legal and contractual context of the SMS.

If additional specialized knowledge and/or skills are required in the audit of the SMS, consideration should be given to attracting information security experts (for example, those with competence in a specific field of activity, or competence in IT security or business continuity management). If experts are involved, their competence must be carefully evaluated.

Unlike auditors, audit team leaders need to have additional knowledge and skills to manage and guide the audit to ensure effective and efficient auditing. Audit team leaders conducting SMS audits that include various aspects of security management must understand the requirements of standards for each aspect and must clearly understand the boundaries of their knowledge and skills for each of these aspects.

For an objective assessment of auditors, it is necessary to determine the criteria for this evaluation. Criteria can be qualitative (such as demonstrated personal qualities, knowledge or characteristics of skills in training or in the performance of duties in the workplace) and quantitative (such as work experience and training in years, number of audits conducted, number of hours of training and training in auditing).

Conclusions

In this chapter the standards, principles, functions and feature of synthesis of the security management system have been described. Its functional scope and functions have been presented by modeling in IDEF0.

Based on consideration of the structure of QS and its functional analysis model established structural and functional analogy between the QS

and the ISMS. In particular, the essential elements of these systems are comparable and to solve such problems, in the first approximation of the ISMS can be regarded as a multiphase QS with expectation and unlimited queue. From this analogy implies the possibility of formulating and solving problems associated with the ISMS as the queuing of tasks and SMO can be regarded as a formal model of the ISMS. It is obvious that the study of the parameters of such a model may pave the way to an understanding of those aspects of information security, which is necessary to pay close attention to the life cycle of information security management.

The audit evidence must be verifiable. It is based on samples of available information, since the audit is carried out in a limited period of time and with limited resources. The appropriate use of samples is closely related to the confidence with which they relate to the findings of the audit.

To maintain their competence in the audit of the SMS, auditors and audit team leaders should regularly participate in the SMS audits and strive for a continuous increase in professionalism. Professional growth includes maintaining and improving competence. It can be achieved through additional practical experience, training, internships, self-training, tutoring, attending meetings, seminars and conferences or other activities. Auditors, audit team leaders and employees responsible for managing the audit program must constantly improve and improve their competence.

Questions for self-checking

1. Which standards do requirements define to security management systems?
2. What does it mean security risk and security management risk?
3. What are features of model of information security risk management system?
4. How functional model may be presented using IDEF notation?
5. How security management risks can be evaluated?
6. How is risk scale selected?
7. Please formulate conclusions basing on analysis of figures 35.7-35.9. What comments can be done step by step?
8. What is reason of function optimum on fig.35.11?
9. What are objectives and principles of security management systems audits?
10. What are features of requirements to audit team and documentation?

REFERENCE

- [1]. Mokhor, V. V. & Bogdanov, A. M. (2011), 'Presentment of the standart «ISO 31000:2009. Risk management. Principles and guidelines» in russian', *Das Management*, No. 3, pp. 5-18.
- [2]. Mokhor, V. V. & Bogdanov, A. M. (2011), 'BS 31100:2008. Risks handling : general practical recommendations', *Das Management*, 2011, No. 4, pp. 7-28.
- [3]. Mokhor, V. V., Bohdanov, O. M., Kruk, O. V., Tsurkan, V. V. (2012), 'Localization attempt ISO GUIDE 73:2009 «Risk management – Vocabulary»', *Bezpeka informacii*, Vol. 18, No. 2, pp. 12-22.
- [4]. International Organization for Standardization (2011), ISO/IEC 27005: *Information technology. Security techniques. Information security risk management*, Geneva, 68 p.
- [5]. National standard of Ukraine (2013), DSTU IEC 31010: *Risk management. Risk assessment techniques*, Kyiv, 73 p.
- [6]. International Organization for Standardization (2015), ISO/IEC 15288: *Systems and software engineering. System life cycle processes*, Geneva, 108 p.
- [7]. Tsurkan, V. V. (2013), 'Functional approach to process modeling of information security risk managing', *Proceeding of informatioaln technologies and security. Assessment of international conference*, Problem of registration information Institute of National Academy of Sciences of Ukraine, Kyiv, pp. 193-194.
- [8]. Guidance document (2000), GD IDEF0: *Methodology of functional modeling IDEF0*, Moscow, 75 p.
- [9]. Atiskov, A. I. & Monakhova, T. V. (2006), 'Process automation of system engineering of enterprise information protection by IDEF and UML', *SPIIRAS Proceedings*, Vol. 2, Iss. 3, pp. 115-119.
- [10]. Zaitcev, O. E. (2007), 'Structural modeling approaches of the main components of IT security' in A. V., Liubimov, A. V., Sukhanov, *Proceedings of The theory and technology of programming and data protection. Application of computer technology on international conference*, ITMO University, St. Petersburg, pp. 56-60.
- [11]. Liubimov, A. V., Shustikov, S. V., & Andreeva, N. V. (2008), 'Functional modeling of information security management system of the organization over the ISO/IEC 2700x standards', *Scientific and technical journal of information technologies, mechanics and optics*, № 7 (52), pp. 251-257.
- [12]. Krivolapov, V. G. (2009), 'Complex technique of information security risk modeling open systems', PhD thesis, Moscow Engineering Physics Institute.

- [13]. Komin, D. S. & Potii, A. V. (2010), 'The IDEF models of the security assurance level evaluation', *Bulletin of V. Karazin Kharkiv National University: mathematical modeling, information technology, automated control systems*», No. 925, Iss. 14, pp. 98-105.
- [14]. Vasilev, V. I. & Belkov, N. V. (2011), 'Decision Support System for the assurance of personal data', *Vestnik UGATU*, Vol. 15, No. 5 (45), pp. 54-65.
- [15]. Tcybulin, A. M. (2012), 'Multi-agent approach to the construction of an automated enterprise information security management system', *Izvestiya SFedU: engineering sciences*, No. 12, pp. 111-116.
- [16]. Buldakova, T. I., Suiatinov, S. I. & Mikov, D. A. (2013), 'The analysis of information risks of virtual health infrastructures', *Information society*, viewed 17 February 2016, <http://emag.iis.ru/arc/infosoc/emag.nsf/BPA/87f599404bc9073d44257c2a00476485>
- [17]. Mikov, D. A. (2014), 'Analysis of the study data flow techniques for assessing information security risks', *Prospero*, No. 7, pp. 27-33.
- [18]. Oladko, V. S. (2015), 'Software system to assess security level of e-commerce systems', *University Journal of Control and Computer Science*, No. 4 (33), pp. 46-53.
- [19]. International Organization for Standardization (2013), *ISO/IEC 27001: Information security management systems. Requirements*, Geneva, 23 p.
- [20]. Mokhor, V. V. (2015), 'Normative and legal aspects a information security risk management system designing' in V. V., Tsurkan, O. M., Kruk, *Proceedings of information security of Ukraine on ukrainian conference*, Taras Shevchenko National University, Kyiv, pp. 122-123.
- [21]. International Organization for Standardization (2013), *ISO/IEC 27001 : Information technology. Security techniques. Information security management systems. Requirements*, Geneva, P. 23.
- [22]. «The concept of information security management system», viewed 10 January 2017, <http://globaltrust.ru/ru/uslugi/vnedrenie-sistem-upravleniya-informacionnoi-bezopasnostyu/ponyatie-sistemy-upravleniya-informacionnoi-bezopasnostyu>.
- [23]. «Elements of queuing theory», viewed 10 January 2017, math.immf.ru/lections/206.html.
- [24]. «Queuing Systems», viewed 10 January 2017, http://eos.ibi.spb.ru/umk/11_4/5/5_R0_T6.html.
- [25]. «What is Helpdesk (Service Desk)?», viewed 10 January 2017, <http://www.helpdeski.ru/tags/helpdesk>.
- [26]. Venttsel, E.S. (1988), 'Operations research: objectives, principles, methodology', Nauka, Moscow, 132 p.

- [27]. Venttsel, E.S. (1969), *'Probability theory'*, Nauka, Moscow, 515 p.
- [28]. Gnedenko, B.V., Kovalenko, I.N. (2012), *'Introduction to queuing theory'*, Bukinist, Moscow, 400 p.
- [29]. Korshunov, Iu.M. (1980), *'Mathematical Foundations of Cybernetics'*, Energiia, Moscow, 424 p.
- [30]. International Organization for Standardization (2016), *ISO/IEC 27035-1 : Information technology. Security techniques. Information security incident management. Part 1 : Principles of incident management*, Geneva, P. 21.
- [31]. Mokhor, V.V., Bohdanov, A.M., Kruk, O.N., Tsurkan, V.V. (2010), 'Building a risk assessment of information security based on dynamic set of actual threats', *Collection of scientific works Institute of Modelling Problems in Power Engineering*, iss. 56, P. 87-99.
- [32]. '«Jet Infosystems» company has built ISMS «Eldorado»', viewed 15 June 2016, <http://www.osp.ru/osp-new/public/resources/releases/?rid=7954>.
- [33]. 'ISO 27001 – Information Management Security System', viewed 15 June 2016, <http://www.enhancequality.com/iso-standards/iso-27001-information-security-management-system/>.
- [34]. Dmitriev A. (2007), 'Information security management', viewed 15 June 2016, http://www.comizdat.com/index.php?in=ksks_articles_id&id=568.
- [35]. International Organization for Standardization (2013), *ISO/IEC 27001 : Information technology. Security techniques. Information security management systems. Requirements*, Geneva, 23 p.
- [36]. International Organization for Standardization (2011), *ISO/IEC 27005: Information technology. Security techniques. Information security risk management*, Geneva, 68 p.
- [37]. Guidelines for the implementation of information security management systems and risk assessment methodology in accordance with the standards of the National Bank of Ukraine, viewed 15 June 2016, <http://zakon3.rada.gov.ua/laws/show/v0365500-11/page>.
- [38]. International Organization for Standardization (2011), *ISO/IEC 27035 : Information technology. Security techniques. Information security incident management*, Geneva, 78 p.
- [39]. Kendall M., Moran P. (1972), 'Geometrical probabilities', Nauka, Moscow, 192 p.
- [40]. Yu.Gordienko, S.Stirenko, O.Alienin, K.Skala, Z.Soyat, A.Rojbi, J.R.Lypez Benito, E.Artetxe Gonz6lez, U.Lushchik, L.Sajn, A.Llorente Coto, G.Jervan (2017), *Augmented Coaching Ecosystem for Nonobtrusive Adaptive Personalized Elderly Care on the Basis of*

- Cloud- Fog-Dew Computing Paradigm, Proc. IEEE 40th Int. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 387-392).
- [41] V.V. Mokhor, and V.V. Tsurkan, "The entropy approach to the definition of the "information security risk", in Proc. *XXVIII conf. Modeling*, Kyiv, 2009, p. 22.
- [42] Netkachov, O., Popov, P., & Salako, K. (2014, September). Quantification of the impact of cyber attack in critical infrastructures. In International Conference on Computer Safety, Reliability, and Security (pp. 316-327). Springer International Publishing.
- [43] Aniello L., Bondavalli A., Ceccarelli A., Ciccotelli C., Cinque M., Frattini F., Guzzo A., Pecchia A., Pugliese A., Querzoni L., Russo S. (2014). Big data in critical infrastructures security monitoring: Challenges and opportunities. arXiv preprint arXiv:1405.0325. <https://pdfs.semanticscholar.org/de55/a260f6c4203b9c6853eae89050bb8b84dff6.pdf>
- [44] Balasubramanian, S., Srinivasan, S., Buonopane, F., Subathra, B., Vain, J., & Ramaswamy, S. (2016). Design and verification of Cyber-Physical Systems using TrueTime, evolutionary optimization and UPPAAL. *Microprocessors and Microsystems*, 42, 37-48

36 ASSESSMENT OF SMART BUILDING AUTOMATION SYSTEMS REALIABILITY AND CYBER SECURITY USING ATTACK AND FAULT TREES

As noted in Chapter 35, in several cases maintenance of Building automation system (BAS) architecture components stops at the operation phase. However, due to circumstances, it is impossible to refuse application of such components or they might have low cost. Moreover, when developing specifications for information and control systems of smart buildings to assess the reliability and cyber-security the selection of the non-failure operating probability criterion (NOP) of the system can be justified.

In this Chapter, we discuss the application of the Attack Tree Analysis (ATA) technology to assess the impact of each component of the system architecture on its reliability and cyber security. Using ATA does not take into account recovery and maintenance, but it allows monitoring any attacks on components and assessing the impact of these attacks on the system as a whole. In the second part of the Chapter, strategies of developing Markov models for describing the recovery of system components after an attack or a software failure are discussed. The use of ATA or Markov models is usually justified by the customer's requirements for a specific criterion for assessing the quality of the system.

36.1 A conceptual approach to assessing reliability and cyber-security of smart building information and control systems

In this Chapter, with respect to the BAS, the main requirement of the user (client) is to ensure a given system availability, the second requirement is to ensure the cyber security of the system and information throughout the life cycle.

For the three-level BAS architecture considered in the thesis, the system-wide availability is influenced by the components of all its levels. The failure of the communication level component directly affects the availability of the system, since the impossibility of

transferring the administration commands isolates the lower-level actuators. In addition, the communication level is most accessible for attacks on its components, which reflects its contribution to system-wide cyber security. Components of other levels (management, automation) also affect the availability of BAS; attacks on them can be identified through monitoring and analysis of system performance. Given the distribution of these levels, it is assumed that single failures of their components do not lead to system shutdown in general.

36.1.1 Basic principles

The architectures of information and control systems of smart buildings can be structurally different from each other, depending on the area where they will be applied (hospitals, departmental buildings, etc.). Fig. 36.1 shows the tree of high-level architecture attacks built using the ATA approach.

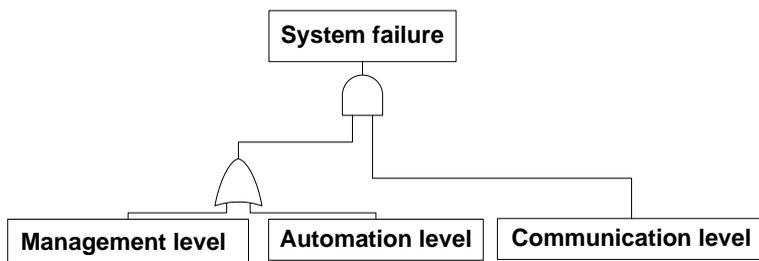


Fig. 36.1 – Presentation of the BAS architecture using the ATA approach

The Attack Tree Analysis is considered as an analytical method in which ways of achieving an undesirable state of the system (in particular, a failure state) are examined. The purpose of the ATA analysis is to assess the reliability and cyber security of the system. This helps architecture developers to understand how the system works with weak points in the project, which can be used by attackers. The ATA analysis shows which requirements for system components need to be increased to ensure cyber security and reliability throughout the life cycle. When using this toolkit, the system is analyzed in the context of the surrounding operating environment to find all possible ways of

failure occurrence. When constructing the model in the form of an event tree, two types of gates are used (AND, OR). The event after gate "AND" occurs with simultaneous manifestation of changes at the input of the gate. The event at the output of the "OR" gate arises if at least one change in the state of the component occurs at its input.

Fig. 36.1 shows the upper level tree of the ATA analysis of the BAS architecture, including three levels. The ATA tree allows to prioritize each level when creating a complex failure event of the system as a whole. Fig. 36.1 shows that the communication level has the highest priority and direct connection via the "OR" gate to the system failure state. The other two levels are connected to each other through the "AND" gate, they cannot independently lead the system to a fault state, and system failure occurs only when faults occur at these levels simultaneously. Nevertheless, the probability of such an event must be taken into account.

When there is a need to analyze the cyber-security of the system, we should choose a specific event – a failure or attack on the system component as a target of the attacker, and then determine the immediate, necessary and sufficient reasons for achieving this goal. Such reasons may not be fundamental to a system-wide failure, but they are the immediate causes for this event. They are considered as sub-goals, or targets of the second level of the attacker. In determining all immediate, necessary and sufficient reasons, a step-by-step analysis of the tree from top to bottom is performed until the ATA model resolution limit is reached, that is, the atomic failure event of the BAS component.

Taking into account all possible targets for attacks that can be directed to the system and its components at each level, then it is necessary to consider the scenarios of cyber-attacks.

36.1.2 General scheme of the dependability analysis

Taking into account the positions of reliability and cyber security allows expanding the list of causes of failures and weaknesses of the system within the framework of a unified dependability concept. In the direction of reliability, hardware and software defects, as well as interaction defects due to operating personnel errors and attacks on the

system are analyzed. On the cyber security aspect, software vulnerabilities, Trojans and backdoors are analyzed (Fig. 36.2).

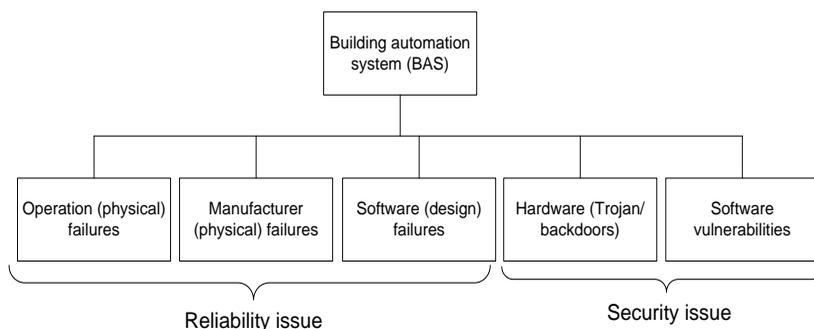


Fig. 36.2 – Causes of failures in BAS components taking into account aspects of reliability and cyber security

36.2 Vulnerability analysis of smart building information and control systems components

According to [1], the BAS architecture has three levels, therefore, vulnerability analysis should be performed for components of these levels. Identifying and assessing the vulnerabilities of these levels helps the developer to manage risks and determine the degree of threat at the design stage of the system. According to the analysis carried out in [2], the main elements of the system architecture that have a high level of threat are FPGA, database, communication. The information obtained in the analysis of vulnerabilities can be used to compile IMECA matrices and forms the basis for designing ATA models.

36.2.1 Analysis of vulnerabilities of FPGA devices

A field-programmable gate array (FPGA) is produced as a ready-to-use electronic device. For application in digital systems, such devices must be programmed. The advantages of FPGA-platforms include simplicity of tuning and cost-effectiveness. In addition, such platforms can be updated during the lifetime, it is simply enough to download a new application code. FPGA-platforms have other advantages, but, nevertheless, their main advantage is the design

flexibility. When analyzing the cyber security of FPGA platforms, it is necessary to take into account all the features of the life cycle of both FPGA chips and information and control systems (I&C) based on FPGAs. Participants of the processes are manufacturers of FPGA chips, designers and developers of I&C systems as well as users of I&C systems based on FPGA. Cyber-security analysis for FPGA technology covers the design and development processes as well as the operation of integrated I&C systems. It should be noted that cyber-security vulnerabilities could be introduced by:

- a manufacturer of FPGA chips in the design, production, setup and testing of FPGA microcircuits;
- a developer of I&C systems at the design, coding and testing stages;
- an I&C operator of the system during operation and maintenance.

36.2.2 Analysis of vulnerabilities in databases

Recently, the number of attacks on databases (DBs) has increased. This is due to the growing demand for data stored in the database and the expansion of access to databases via the global network. The databases in I&C systems of smart buildings contain information that is important for the system and its various levels for controlling executive devices.

When we expand access rights to the stored information for several users, this increases the likelihood of data theft. Therefore, in BASs access to the database must be constantly monitored. An attacker seeks to gain access to important information that he can use to attack or monitor the system. Various types of threats that affect the cybersecurity of databases are given below.

1. Abuse of rights and privileges. The threat arises in a situation where database users have more privileges than it is required to perform functional duties. These privileges can be deliberately or unintentionally transmitted to intruders.

2. Vulnerabilities of operating systems, such as Windows, UNIX, Linux, etc., as well as OS services that interact with databases, can act as a means for unauthorized access. Such vulnerabilities can also be used for denial of service (DoS) attacks. As a rule, they are fixed after installing/updating the operating system security patches.

3. Rootkits (rootkits) of databases. A rootkit is a program or procedure that is hidden inside the management system (DBMS) and provides administrative privileges to access data and disable the Intrusion Prevention Systems (IPS). The rootkit can be installed after using the vulnerabilities of the main operating system. Identification of rootkits is performed using periodic audits; when there are no such audits, the presence of a rootkit in the database can remain unnoticed. To gain credentials for entering the database, attackers can use different strategies (social engineering, direct search of passwords), and they can be successful in case of using weak authentication methods. In the presence of a rootkit, the DBMS assumes that the attacker has the identity of legitimate database users.

4. Weakening the requirements for auditing. The presence of simplifications and weaknesses in the mechanisms of DBMS audit and event logging can become a critical threat for the system, especially in industries with strict regulatory requirements. To restore the history, prior to incidents, the protocols PCI, SOX and HIPAA, which allow for advanced logging, are used. It should be noted that the logging of suspicious or undefined operations in the database must be performed automatically. The audit log is the last line of cybersecurity in the database. The records in it allow detecting an intrusion, which in turn will help to track violations of a particular user at a certain point in time.

36.2.3 Analysis of the vulnerabilities in wireless communications

In the architecture of wireless communications, there are four main components [3]. They include the radio frequency data channel; access points providing connection to the network of the organization; transceivers of end devices (laptops, smartphones, etc.); and programs with a user interface. These components may be vulnerable and subject to attack, which will lead to breach of confidentiality, integrity and availability [4]. The following types of attacks on wireless communications are analyzed.

1. Unintentional association, the type of unauthorized access to the company's wireless networks. When a user turns on the computer and connects to a wireless access point that belongs not to a corporate, but

to the neighboring network, it may not even know that this has happened. Such a violation of cybersecurity can reveal valuable information about the company and create a connection between the company's network and a fake network [5]. The same incident can occur with a laptop connected to a wired network.

2. Peer-to-peer networks. Such networks are often organized to exchange data between two wireless devices. Despite the possibility of using enhanced encryption methods, as a rule, they are neglected when creating peer-to-peer networks [6].

3. "Man-in-the-middle" attack: an attacker creates a program access point (AP), which connects corporate users. After that, the attacker connects to a real access point using another wireless card that provides a constant stream of traffic through a transparent hacker network to the real network [7]. Thus, an attacker can listen to the traffic.

4. Denial-of-service attack (DoS). An attacker organizes a constant load on the target access point or network using dummy requests, error messages, messages about premature successful connections, and/or other commands. Due to this attack, users cannot access the network. These attacks are based on abuse of protocols, such as the Extensible Authentication Protocol (EAP).

36.2.4 Scenarios of cyber-attacks on information and control systems of smart buildings

Cyber-attacks are conducted to disrupt the normal operation of the BAS by stealing, modifying or destroying data, or code. One way to conduct cyber-attacks is to hack personal computer systems or I&C systems of organizations, their infrastructure, computer networks, and/or personal computer devices. Typically, the source of cyber-attack is difficult to detect, since an attacker makes efforts to ensure anonymity. Such attacks can be organized not by individuals, but by whole cyber-campaigns within the framework of cyber war or cyber terrorism. The ways to implement cyber-attacks include installing spyware on a PC, destroying the infrastructure of an organization or even a whole state. Every day, the complexity and danger of cyber-attacks increases.

Like random components failures, cyber-attacks can be directed to hardware channels and BAS software. Since the BAS components are accessed from the global network [8], they are all potential targets of cyber-attacks.

Attacks on hardware can use embedded code or errors made to the chip through the fault of the manufacturer. Therefore, a hardware bookmark, virus or worm can be active for some time. Software attacks can be carried out using various tools for monitoring and reading data, for example, scanning the radio channel of wireless devices for transmitting and receiving data.

Scenarios of cyber-attacks on hardware channels or software can cause a system-wide failure through a hardware failure and errors in the software component.

To analyze the cyber security of BAS, it is necessary to analyze and study all possible attacks on the system, to predict how an attacker will attempt to access the system from the inside. [9] The scenarios of cyber-attacks on the BAS can be divided into three parts:

1. The attacker gets access with the help of special tools for monitoring the network. Access is an intermediate goal. At the initial stage, the attacker's goal is to monitor the network and read the inter-level data exchange.

This type of attack cannot be detected for a long time, since it often has no signs of detection during system operation. The way to counter these kinds of attacks is to enhance the cyber security of the network.

2. In the second part of the scenario, the attacker's goal is to disrupt the system. This can be performed by introducing malicious code (virus, worm) into the system. The recovery time of the system after this attack is different and depends on the level that has been attacked:

- a) if the attacker seeks to capture the automation level and stop one of its components, it is possible to detect a system error and restore the code by changing or updating the system during recovery. Without removing the code, the system can also save partial operability;

- b) if the target of the attacker is the management level, then the recovery process will be difficult, since this level controls all system tasks and it is difficult to conduct maintenance without a complete shutdown of the system. A cyber-attack on the management level causes a long recovery time and high costs for renewal.

3. If an attacker becomes aware of design errors, then cyber attacks can be carried out directly.

The described stages of cyber-attack scenarios are systematized and presented in Fig. 36.3. This scheme can be used to understand the attacker's strategy when he tries to access and attack BAS.

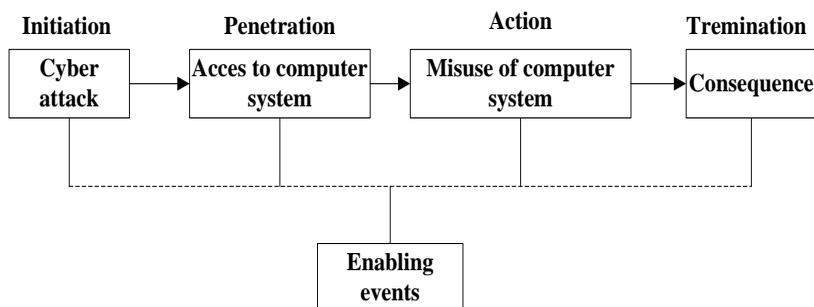


Fig. 36.3 – The main stages of cyber-attack scenarios on BAS

36.3 Development of models for assessing the cyber security of smart building I&CS using FMECA and ATA technologies

The overall goal of attacks can be characterized as a violation of the performance of system functions defined at the design stage. Identification of failures implies the definition of the characteristics of potential mechanisms for their occurrence and an assessment of the probability of failure in real systems during the operational phase. In order to protect the system, developers and users should find answers to the three following questions: "How the system can fail?" "What consequences will the failure have?", and "How much can the system handle?". To answer these questions, FMECA and ATA techniques have been developed, which will be considered further for assessing cyber-attacks on BAS architecture components.

36.3.1 BAS analysis using the FMECA and IMECA methodologies

Failure Modes and Effects Analysis (FMEA) is a technological process that is used to study the potential consequences of failures of

the system on it and its environment [10]. If this takes into account the criticality of failures, then the method is called Failure Modes, Effects and Criticality Analysis (FMECA) [11]. FMEA and FMECA are the most popular tools for finding design defects during the development of the system. They also facilitate the search and elimination of defects during the operation of the system. In this paper, in addition to these methods, the method of assessing the types, consequences and criticality of external influences – IMECA – is also used [12]. Unlike FMEA and FMECA, it considers system failures caused by malicious external actions (intrusions). In accordance with the scenario of cyber-attacks discussed in the previous subsection, we can apply IMECA to analyze the cyber security of a BAS within this scenario and measure the level of failures of system architecture components. According to the analysis of cyber security, the components of the system can be divided into subsets of elements (hardware, software). In this paper, FMEA was used to illustrate the impact of attacks on the operability of the system hardware (Table 36.1). IMECA is used to analyze the software component of the system, as shown in Table 36.2.

Table 36.1 – System FMECA analysis of BAS according to cyber-attack scenarios

Architecture level	Failure type	Failure cause	Failure consequences
Management level	Hardware	Operator errors or design defects	This level is represented as a system control unit; a failure will lead to the system shutdown
Management level	Hardware	Design errors or intrusion into components	System downtime and recovery time will be long and costly, since there is a need to modify the hacked component
Automation level	Hardware	End device shutdown	The system works without downtime and with limited data

			entry. The recovery time will be short, since the hacked sensor can be quickly replaced
--	--	--	---

Table 36.2 – System IMECA analysis of BAS according to cyber-attack scenarios

Architecture level	Component	Types of attack	Cause of failure	Impact on operability	Consequences	
					Cybersecurity	Availability
Communication level	Wi-Fi	Passive	An attacker has access to the wireless network and monitors all transmitted data	Failures	An attacker knows all the transmitted data	Impact on availability is not provided
		Active	After an attack, the access is obtained to enter the network; an attacker breaks the connection between the levels using various tools (viruses, bookmarks)	Denials	The purpose of the attack is to disable the system and completely disable the security system	Full impact on availability, as the system goes into the failure mode until the vulnerability is identified and removed

Management level	DB	Passive	After a successful cyber-attack, an attacker gets access to a database for reading and recording information	Failures	The security of the system is compromised, since an attacker controls the data inside the system	The availability of the system depends on the purpose of the attacker: he can either steal data or damage them and disable the system
------------------	----	---------	--	----------	--	---

36.3.2 Models of components of the BAS architecture in the form of an ATA tree

To begin with, the ATA models presented in Figs. 2.4-2.6 are considered. Increasing the Attack Trees was carried out gradually from below-upwards. Initially, the trees of the components of individual levels were built (examples are given: the ZigBee protocol of the switching level in Fig. 36.5 and the FPGA controllers of the automation level in Fig. 36.4).

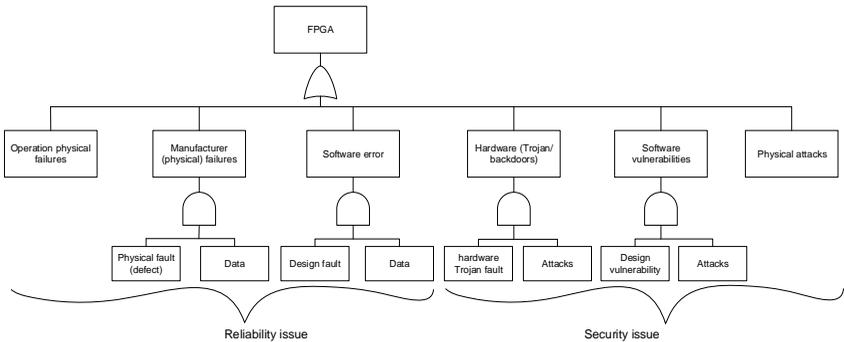


Fig. 36.4 – Attack Tree model of FPGA controllers

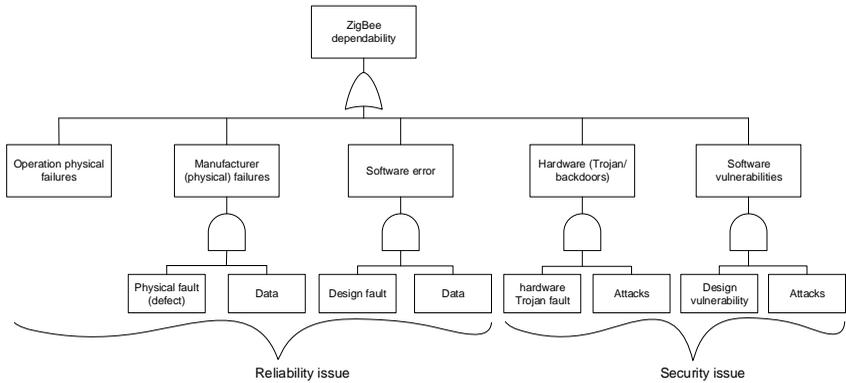


Fig. 36.5 – Attack tree model of ZigBee protocol

Then, an ATA tree was built for the entire BAS system. For this tree, calculations were made of the probability of a failure in a subset of cybersecurity, the results of which are summarized in Table 36.3.

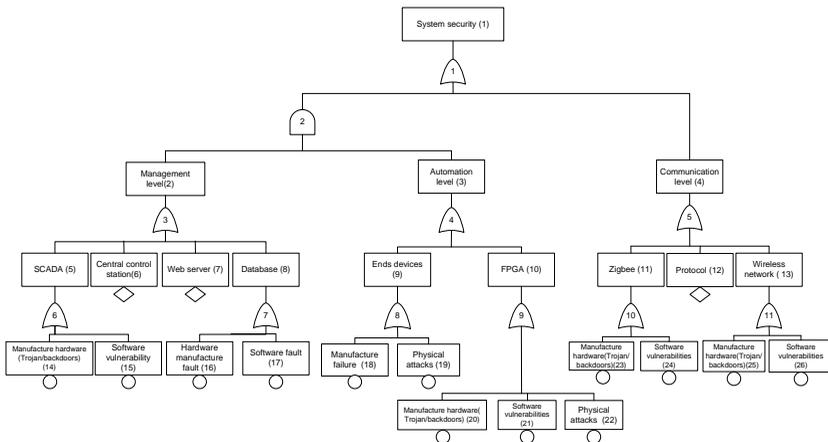


Fig. 36.6 – Attack tree model of BAS components for assessing static indicators of cyber security

Table 36.3 – Calculation of a failure probability of the information and control system in a smart building according to cyber security indicators

Architecture level	Component No	Vulnerability class of BAS component	Probability of successful attack	Probability of failure of BAS as a result of external influences (attacks on vulnerabilities) 0.000281468
Management level	1	Manufacture hardware (Trojan/backdoors) (14)	0.0000842	
	2	Software vulnerability (15)	0.0000458	
	3	Hardware manufacture (20)	0.0000789	
	4	Software fault (21)	0.0000523	
	5	Central control station (6)	0.0000157	
	6	Web server (7)	0.0000791	
Automation level	7	Manufacture failure (16)	0.0000825	
	8	Physical attacks (17)	0.0000423	
	9	Manufacture hardware (Trojan/backdoors) (22)	0.0000373	
	10	Software vulnerability (23)	0.0000656	
	11	Physical attacks (24)	0.0000474	
Communication level	12	Manufacture hardware (Trojan/backdoors) (18)	0.0000063	
	13	Software vulnerability (19)	0.0000888	
	14	Manufacture hardware (Trojan/backdoors) (25)	0.0000764	

	15	Software vulnerability (26)	0.0000678	
	16	Protocol (13)	0.0000421	

36.3.3 Models of BAS architecture in the form of FTA and AvTA trees

The approach proposed in the work allows to identify the causes of failures in a complex multi-level system, which is especially important when analyzing the vulnerabilities of individual components of lower levels. The model considered earlier (Fig. 36.1) needs to be improved for the subsequent combination of two types of failure trees (FTA – Fault Tree Analysis and ATA – Attack Tree Analysis) and accounting for recovery processes (AvTA-Availability Tree Analysis).

The developed BAS models in the form of separate trees (FTA, ATA and AvTA) are presented in Fig. 36.7 ... Fig. 36.9. With the help of the constructed trees, the calculation of the probability of the system failure due to software defects and attacks on vulnerabilities has been made, the results of which are presented in Table 36.4.

Table 36.4 – Calculation of the probability of failure-free operation of the smart building I&C system in terms of reliability and cyber security

Arch. level	Subset	Component	Name of the AvTA input parameter	Value (probability)	
Hardware	Reliability	FPGA	physical operation failure (hardware)	0.0012	Probability of system failure = 0.001590089
			physical operation failure (soft hardware error)	0.002	
			manufacture failure (hardware)	0.25	
		ZigBee	physical operation failure (hardware)	0.0021	
			physical operation failure (soft hardware error)	0.1265	
			manufacture failure	0.15157	

Software	Security		(hardware)	
		Database	physical operation failure (hardware)	0.17664
			physical operation failure (soft hardware error)	0.20171
		Rec/hardware	recovery depending on type of failure	0.8
		FPGA	intrusion failure (severe hardware vulnerability)	0.25185
			intrusion failure (soft hardware vulnerability)	0.27692
	Ahw	attack by intruder (hardware)	0.30199	
	Rec/software	recovery depending on type of failure	0.5	
	Reliability	FPGA	failure caused by design fault (software)	0.005
			failure caused by software design (soft software error)	0.015
			failure caused by ageing (software)	0.025
		ZigBee	failure caused by design fault (software)	0.035
failure caused by software design (soft software error)			0.045	
failure caused by ageing (software)			0.055	
Database		failure caused by design fault (software)	0.065	
		failure caused by software design (soft software error)	0.075	
		failure caused by ageing (software)	0.085	

		Rec/hardware	recovery depending on type of failure	0.8	
	Security	FPGA	intrusion failure (severe software vulnerability)	0.0215	
			intrusion failure (soft software vulnerability)	0.078	
			attack by intruder (software)	0.325	
		Database	intrusion failure (severe software vulnerability)	0.445	
			intrusion failure (soft software vulnerability)	0.59675	
			attack by intruder (software)	0.7485	
		ZigBee	intrusion failure (severe software vulnerability)	0.90025	
			intrusion failure (soft software vulnerability)	0.0252	
			attack by intruder (software)	0.0785	
			Rec/software	recovery depending on type of failure	0.5

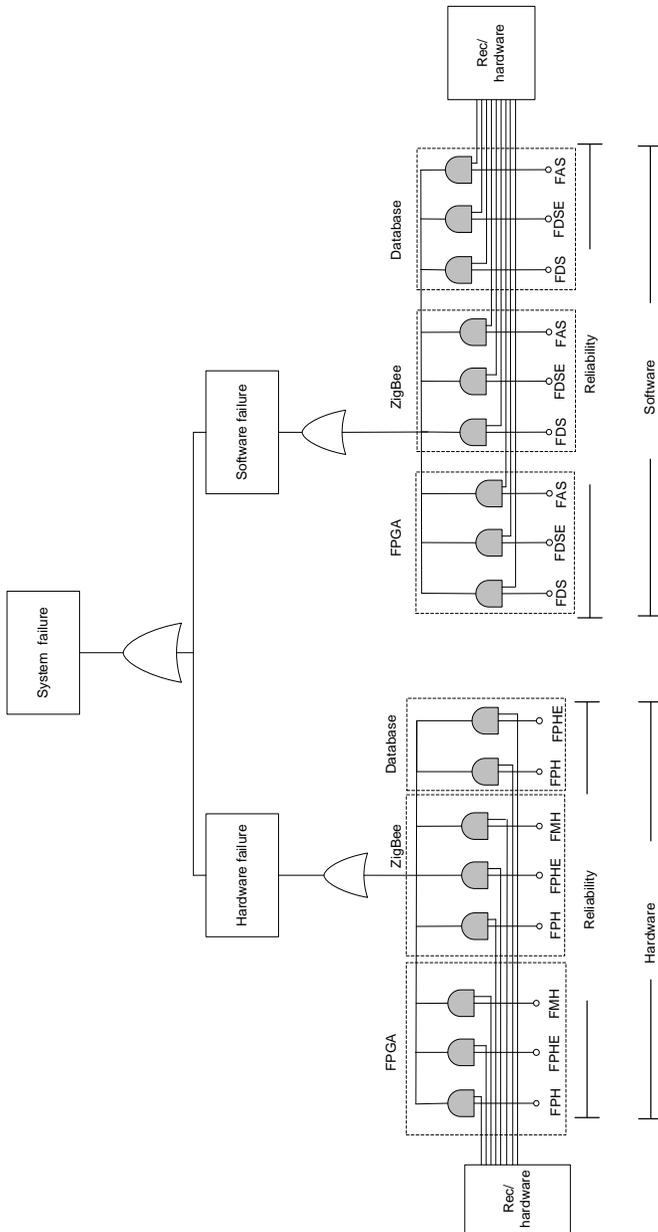


Fig. 36.7 – Fault tree model of BAS components

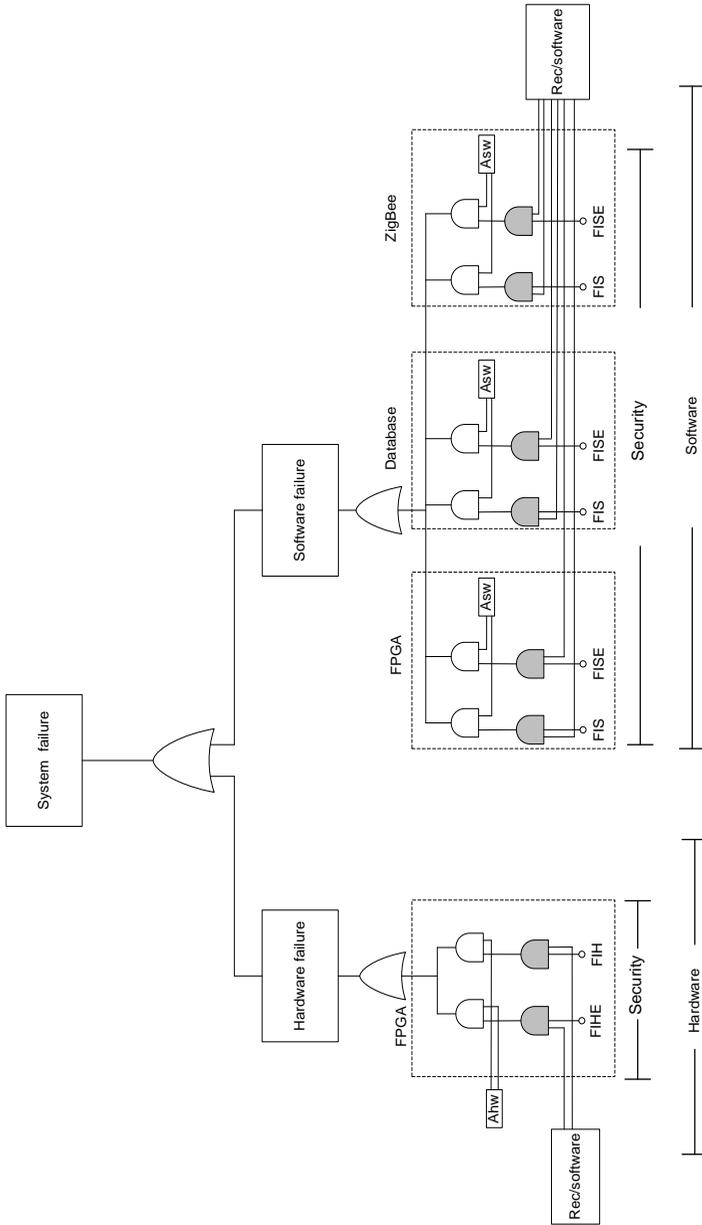


Fig. 36.8 – Attack tree model of BAS components

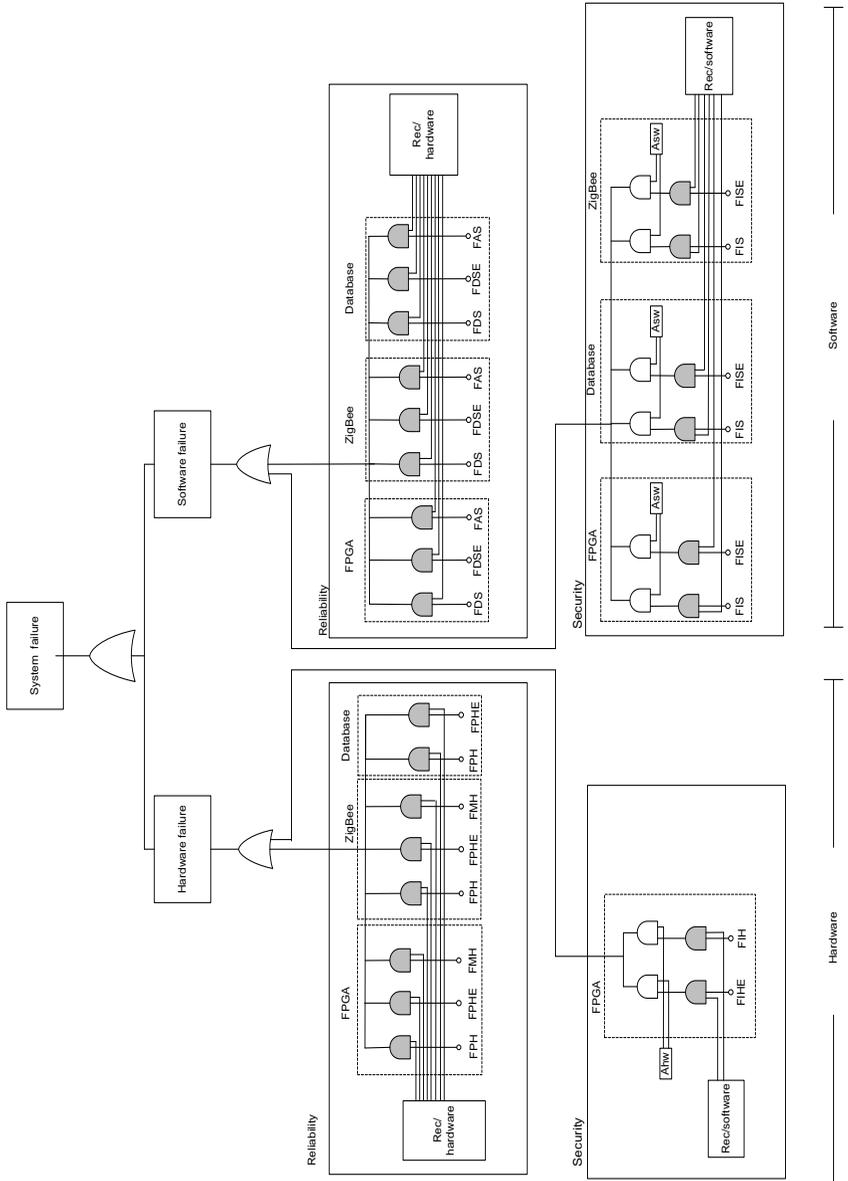


Fig. 36.9 – Availability tree model of BAS components

36.4 Scaling of models for assessing the reliability and cyber security of smart building I&C systems

The project of intellectualization of the university campus buildings presented in Fig. 36.10 provides the installation of sensors and actuators in buildings of different categories. In ordinary residential buildings, the elements of the low-level intelligent building systems linked to the BAS are located, the control level of which is located in a separate data center. The data center is located within the reach of the local network of the communication level. Thus, each zone, denoted as "Area" in Fig. 36.10, due to ensuring the requirements for autonomy of functioning, is considered as a BAS of the first level (Level 1), which is shown in Fig. 36.10. The administrative building in the "Area 1" zone also has intelligent systems, as well as the servers on which the private cloud is deployed (Private Cloud). This cloud provides a management level over the entire campus. To communicate with the cloud, other zones use the resources of the Internet, because the distances between them cannot be limited to the use of the local network.

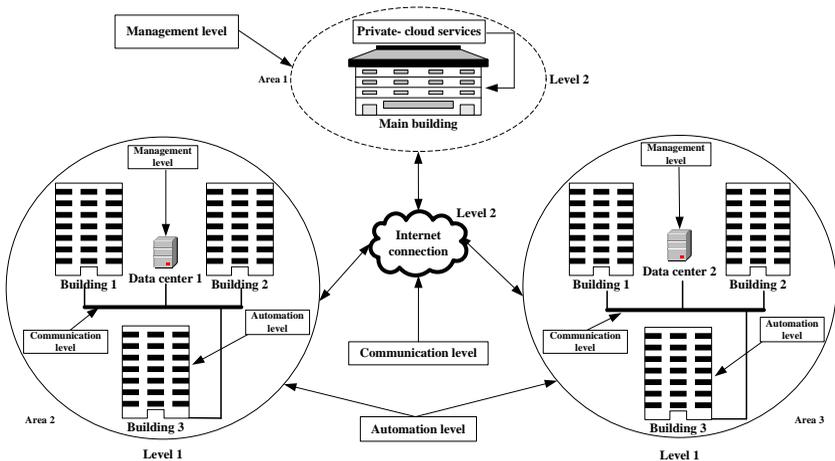


Fig. 36.10 – Design of the architecture of the intellectualization system for the smart university campus

Thus, when scaling tree models of failures and attacks on the university campus according to Fig. 36.10, three levels of architecture are also pointed out. At the management level, Private Cloud servers deployed in the administrative building are considered. The communication level unites all Internet connections between cloud servers and the BAS residential buildings. The automation level is associated with the BAS of residential buildings of the first level.

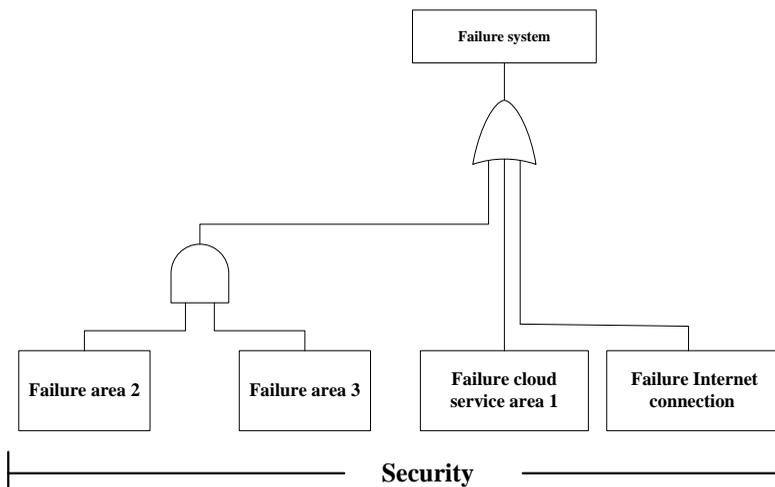


Fig. 36.11 – The tree of attacks (ATA) on components of the university campus intellectual system

When constructing an Attack Tree model for the university campus systems (Fig. 36.11), generalized indicators of the non-failure operating probability of individual zones, cloud servers and the communication level are considered. The last two NOPs were identified in [13,14], and the NOP of the BAS level is determined by the previously developed models of cyber security (Fig. 36.8). The Attack Tree of the university campus is constructed using assumptions about the impossibility of hacking the whole system only by attacking one of the BASs of the first level. This means that attackers in order to transfer the entire system to the failure mode must either crack both BASs of the first level at the same time, or disrupt the cyber security in communication and management levels.

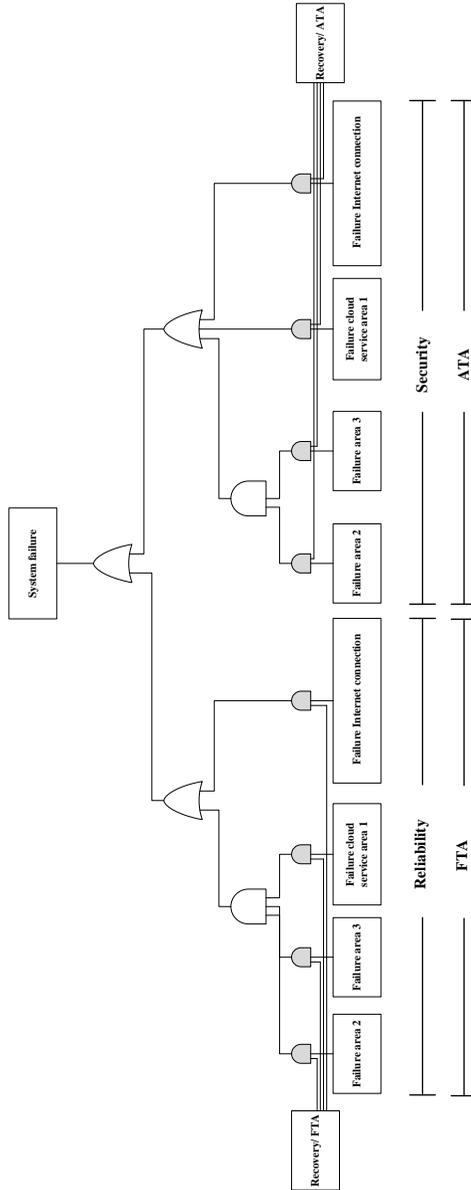


Fig. 36.12 – The tree of fault ant attacks (AvTA) on components of the university campus intellectual system

The Fault Tree model of the university campus intellectual system (Fig. 36.13) also considers the generalized non-failure operating probability indicators of the BAS level obtained with the help of previously developed FTA-models (Fig. 36.7). NOPs of cloud servers and the level of communication were defined in [15]. Due to the autonomy of the operation of systems in different zones, a system-wide failure occurs only if the BASs of these zones simultaneously shutdown, or if the communication level is damaged.

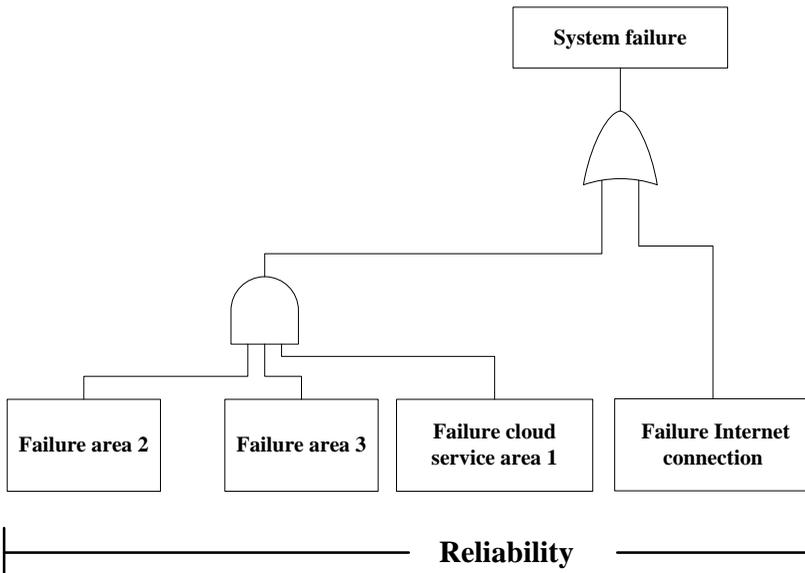


Fig. 36.13 – Fault tree (FTA) model for components of the university campus intellectual systems

Table 36.5 shows the results of calculations of the NOPs for the intellectual system of the university campus, and the AvTA model of the campus is presented in Fig. 36.12.

Table 36.5 – Calculation of the NOP for I&Cs of the smart building according to indicators of reliability and cyber security

Type of	Issues	Parameters	Probability	
---------	--------	------------	-------------	--

Tree				
FTA	Reliability	Failure area 2	0.0012	System probability to failure with recovery=0.006187324
		Failure area 3	0.002	
		Failure – cloud services –area 1	0.25	
		Failure Internet connection	0.0021	
		Recovery /FTA	0.8	
ATA	Security	Failure area 2	0.005	System probability to failure without recovery=0.011139648
		Failure area 3	0.015	
		Failure – cloud services –area 1	0.0025	
		Failure Internet connection	0.0065	
		Recovery /ATA	0.5	

According results of calculations, it is possible to draw a conclusion that accounting factors of recovery and blocking of attacks allows to specify the importance of NOP value for the intellectual system of the university campus by an order of magnitude.

36.5 Development of a conceptual model for the I&Cs functioning of the smart building taking into account recovery and maintenance

In general, the BAS conceptual model should cover a full set of reasons for system shutdown [16]. At the same time, the dimension and complexity of the model cause the search for ways of its decomposition into smaller models describing the mutually independent causes of failures. Thus, for models of hardware and software failures, it is possible to construct both a generalized model and two separate availability models with the subsequent multiplication of their resulting availability coefficients (or functions).

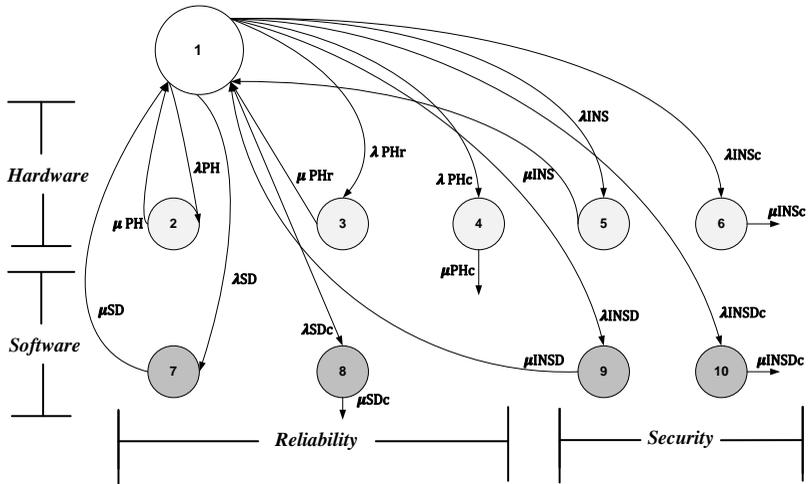


Fig. 36.14 – Conceptual scheme for constructing the general model of BAS functioning taking into account two groups of failure causes

The general concept of building a model with two groups of failure causes (subsets of reliability and cyber security) is presented in Fig. 36.14. The upper level is occupied by the initial working state of the S_1 system. The level below is a subset of the hardware states – the group of states $S_2 \dots S_6$ caused by the manifestations of the faults in hardware. The lower part of the Fig. shows the subset of the states of the software tools $S_7 \dots S_{10}$. Under the condition of changing the parameters of manifestation defects in design and interaction (intrusions), the model will expand in the direction of four vectors from states S_4, S_6, S_8, S_{10} , to final states in which the parameter change stops. Causes and events, which change the parameters of the manifestation of design faults, are described in detail in [17]. Explanations to the definition of the input parameters of the conceptual model are given in Table 36.6.

Table 36.6 – Input parameters of the conceptual model for the I&CS of the smart building

Parameter notation	Detailed description of the input parameter
λ_{PH}	Physical operation failure (hardware)

μ PH	Physical operation failure (hardware/repair)
λ PHr	Physical failure operation (soft error)
μ PHr	Physical operation failure (soft hardware error/restart)
λ PHc	Physical manufacture failure (hardware)
μ PHc	Manufacture failure (hardware/changing design)
λ INS	Intrusion failure (soft hardware vulnerability)
μ INS	Intrusion failure (soft hardware vulnerability /restart)
λ INSc	Intrusion failure (severe hardware vulnerability)
μ INSc	Intrusion failure (severe hardware vulnerability/changing design)
λ SD	Failure caused by design fault (software)
μ SD	Soft error caused by design fault (software/restart)
λ SDc	Failure caused by design fault (software)
μ SDc	Failure caused by design fault (software/changing code)
λ INSD	Intrusion failure (soft software vulnerability)
μ INSD	Intrusion failure (soft software vulnerability/restart)
λ INSDc	Intrusion failure (severe software vulnerability)
μ INSDc	Intrusion failure (severe software vulnerability/changing code)

The logic of the mechanisms for changing the parameters of attacks on the vulnerabilities of the BAS architecture component is as follows. Initially, at the time of putting the system into operation, it contains some set of component vulnerabilities. At the same time, this set contains vulnerabilities known from records in open repositories as well as the so-called "zero day" vulnerabilities (about which there is no information in open repositories).

In the process of functioning, the following events that affect the change in the number of vulnerabilities in the system can take place:

- elimination of single vulnerabilities (both open and "zero day") after attacks of intruders;

- elimination of single vulnerabilities (both open and "zero day") after their detection by users;
- elimination of a group of open vulnerabilities resulting from cyber security maintenance procedures;
- introduction of new vulnerabilities as a result of BAS reconfiguration or software updating.

Fig. 36.15 graphically shows how to resolve single (a) and group (b) vulnerabilities of BAS components.

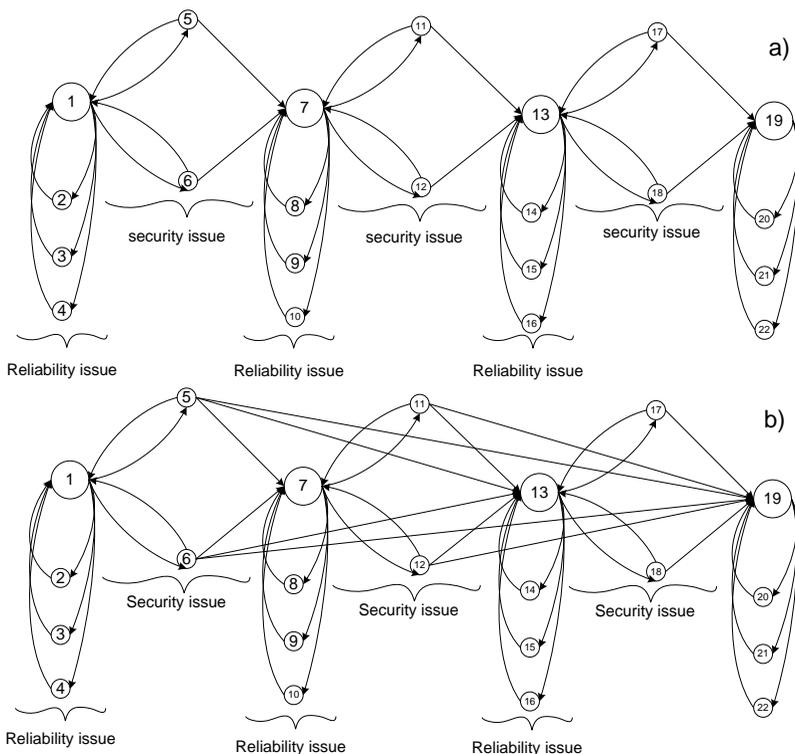


Fig. 36.15 – Dynamics of change in the BAS conceptual model when performing security maintenance procedures with elimination of single (a) and group (b) vulnerabilities

In the interest of further research, it is assumed that the number of failure causes is limited to two subgroups: software defects due to

design errors and attacks on software component vulnerabilities. Taking into account such an assumption, the dimension of the conceptual model decreases, as shown in Fig. 36.16, a. Fig. 36.16, b shows a Markov graph of the conceptual model, taking into account the second assumption about the sequential manifestation of defects and attacks on vulnerabilities. In addition, it is assumed that a defect or vulnerability will be eliminated with probabilities PR (PS).

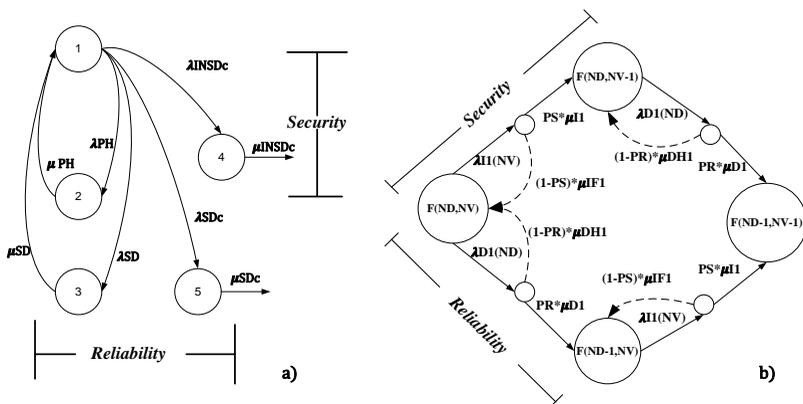


Fig. 36.16 – A simplified graph of the BAS conceptual model (a) and with consideration of the PR (PS) probabilities to eliminate defects and vulnerabilities (b)

In the future, when modeling a system with a number of defects and vulnerabilities more than 1, the dimension of the graph shown in Fig. 36.16, b will increase, but the depicted lozenge will remain the reference fragment of the BAS model.

Conclusions

The chapter presents the existed techniques and conceptual approaches to assessing the reliability and cybersecurity of information and control systems using models in the form of fault and Attack Trees as well as graph models of states and transitions.

The reliability and cyber-security models BASs using AND-OR trees for analysis of failures and attacks has been described. This

allowed taking into account the influence of faults and vulnerabilities of BAS components on the probability of failure.

The Attack Tree models for the BAS components and for the system as a whole are considered as well as Fault Tree Models and combined failure and attack models (AvTA), which allow considering the recovery of operability and blocking of attacks.

From the practical point of view, described models and techniques are important as allowing choice a non-maintenance BAS component, and develop more detailed requirements and techniques for assessing the reliability and cyber security.

Questions to self-checking

1. Please describe the main components of Building automation system (BAS) architecture.

2. Which are the main differences between Attack Tree Analysis (ATA), Fault Tree Analysis (FTA) and Availability Tree Analysis (AvTA)?

3. Which are typical vulnerabilities of FPGA devices?

4. Which are typical vulnerabilities in databases?

5. Which are typical vulnerabilities in wireless communications?

6. Which are probable scenarios of cyber-attacks and their consequences for BAS states?

7. Please, describe the main procedures of FMECA and FTA technologies

8. Please, describe the main issues of IMECA and ATA technologies

9. Which are the main steps of modeling of BAS architecture components by use of the ATA?

10. Which states are possible in conceptual model for the BASs functioning taking into account strategies of recovery and maintenance?

References

1. Farooq, Umer, Marrakchi, Zied, Mehrez, Habib. Tree-Based Heterogeneous FPGA Architectures – New York Springer Science+Business Media. 2012. 188 p.

2. Rie Higuchi. Building automation and control systems. The United Kingdom. A multi client study – BSRIA Limited Old Bracknell Lane West, Bracknell, 2013. – 203 p.
3. Hatambeiki, A. Wireless Network Security – San Francisco, California, 2004. – 132 p.
4. D. Nagamalai, B. Dhinakaran, P. Sasikala, S. Lee and J. Lee, Security Threats and Countermeasures in WLAN, – Technologies for Advanced Heterogeneous Networks. AINTEC 2005. Lecture Notes in Computer Science, – vol 3837, – pp. 168-182, – 2005, doi: 10.1007/11599593_13.
5. Vishali R. Security in Wireless Local Area Networks. – International Journal of Computer Science and Information Technology Research. – 2014. – Vol.2, Issue 2. – P. 472-483.
6. K. Scarfone, D. Dicoi, M. Sexton and C. Tibbs, Guide to securing legacy IEEE 802.11 wireless networks, – NIST Special Publication 800-48, – 2008, doi: 10.6028/NIST.SP.800-48r1.
7. R. Jain, Wireless LAN Security II: WEP Attacks, WPA and WPA2, –Washington: University in Saint Louis, – 2009. – 33 p.
8. Mustafa Qahtan Abdulmunem Al-Sudani, V. S. Kharchenko, D. D. Uzun. Vulnerability analysis of wireless networks: case for smart building automation system – Radioelectronic and computer systems. - 2015. – Vol. 2. - P. 69–76.
9. Kharchenko V., Ponochovnyi Y., Abdulmunem AS.M.Q., Andrashov A. Availability Models and Maintenance Strategies for Smart Building Automation Systems Considering Attacks on Component Vulnerabilities. – Advances in Intelligent Systems and Computing, Vol. 582, 2017, P. 186-195. DOI: 10.1007/978-3-319-59415-6_18
10. Technical manual. TM 5-698-4, Failure Modes, Effects and Criticality Analyses (FMECA) for Command, Control, Communications, Computer, Intelligence, Surveillance, and Reconnaissance (C4ISR) Facilities. – Department of the Army Washington, DC, imp. 29 September 2006. – 75 p.
11. X. Cheng, Z. Xing, Y. Qin, Y. Zhang, S. Pang and J. Xia, Reliability Analysis of Metro Door System Based on FMECA, –Journal of Intelligent Learning Systems and Applications, – vol. 05, no. 04, – pp. 216-220, – 2013, doi: 10.4236/jilsa.2013.54024

12. E. Babeshko, V. Kharchenko and A. Gorbenko, Applying F(I)MEA-technique for SCADA-Based Industrial Control Systems Dependability Assessment and Ensuring, – 2008 Third International Conference on Dependability of Computer Systems DepCoS-RELCOMEX, Szklarska Poreba, – pp. 309-315, – 2008. doi: 10.1109/DepCoS-RELCOMEX.2008.23.

13. T. Novak and A. Treytl, “Functional safety and system security in automation systems - a life cycle model”, – In 2008 IEEE International Conference on Emerging Technologies and Factory Automation, Hamburg, – pp. 311-318, – 2008, doi: 10.1109/ETFA.2008.4638412.

14. Feruza Sattarova, Y. Tao-hoon Kim, IT Security Review: Privacy, Protection, Access Control, Assurance and System Security, – In International Journal of Multimedia and Ubiquitous Engineering, – Vol. 2, No. 2, – pp. 17-31, – 2007.

15. Q. Yu and R. J. Johnson, Smart grid communications equipment: EMI, safety, and environmental compliance testing considerations, – Bell Labs Technical Journal, – vol. 16, no. 3, pp. 109-131, – Dec. 2011, doi: 10.1002/bltj.20525

16. K. S. Trivedi, D. S.fdc Kim, A. Roy and D. Medhi, Dependability and security models, – 7th International Workshop on Design of Reliable Communication Networks, – Washington, DC, – pp. 11-20, – 2009. doi: 10.1109/DRCN.2009.5340029.

17. B. Joshi, D. Pradhan and S. Mohanty, Fault Tolerant Nanocomputing, – Lecture Notes in Electrical Engineering, – pp. 7-27, – 2010, doi: 10.1007/978-90-481-8540-5_2.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ К РАЗДЕЛУ 36

AP – Access point

ATA – Attack Tree Analysis

AvTA – Availability Tree Analysis

BAS – Building automation system

DB – Database

DBMS – Database management system

DoS – Denial of service

EAP – Extensible Authentication Protocol

FMEA – Failure Modes and Effects Analysis

FMECA – Failure Modes, Effects and Criticality Analysis

FPGA – Field-programmable gate array

FTA – Fault Tree Analysis

I&CS – Information and control systems

IMECA – Intrusion Modes, Effects and Criticality Analysis

IPS – Intrusion Prevention System

NOP – Non-failure operating probability

PC – Personal Computer

АННОТАЦИЯ

В разделе представлены модели надежности и кибербезопасности информационно-управляющих систем умных домов с использованием И-ИЛИ деревьев анализа отказов и атак учитывающих влияние дефектов и уязвимостей различных компонент их архитектуры и параметров процессов восстановления работоспособности и блокировки атак, позволяющих рассчитать вероятности отказа систем. Учет надежности и кибербезопасности позволяет расширить перечень причин отказов и слабых мест системы в рамках единой концепции гарантоспособности. По направлению надежности анализируются аппаратные и программные дефекты, а также дефекты взаимодействия вследствие ошибок обслуживающего персонала. По аспектом кибербезопасности анализируются уязвимости программных средств, троянские программы и бэкдоры.

У розділі представлені моделі надійності та кібербезпеки інформаційно-керуючих систем розумних будинків з використанням ТА-АБО дерев аналізу відмов і атак шляхом урахування впливу дефектів і вразливостей різних компонент їх архітектури і параметрів процесів відновлення працездатності і блокування атак, що дозволяє розрахувати ймовірності відмови систем. Врахування позицій надійності та кібербезпеки дозволяє розширити перелік причин відмов та слабких місць системи в рамках єдиної концепції гарантоздатності. За напрямком надійності аналізуються апаратні та програмні дефекти, а також дефекти взаємодії внаслідок помилок обслуговуючого персоналу. За аспектом кібербезпеки аналізуються вразливості програмних засобів, троянські програми та бекдори.

Building automation systems models as failure and attack tree and states graph are discussed in the section. The further development was given to the reliability and cyber security model of information and control systems of smart buildings using AND-OR trees of faults and attacks analysis by taking into account the influence of the defects and vulnerabilities of various components of their architecture and the

parameters of the processes of recovery and blocking of attacks, which allows to calculate the probability of failure of the system. Consideration of the reliability and cyber security positions allows to expand the list of causes of failures and weaknesses in the system within the framework of a single concept of dependability. Hardware and software defects as well as defects in interaction due to operating personnel errors and attacks on the system are analyzed in the direction of reliability. The cyber security aspect analyzes vulnerabilities in software, Trojan programs and backdoors.

37 ASSESSMENT OF SMART BUILDING AUTOMATION SYSTEMS AVAILABILITY AND SECURITY CONSIDERING MAINTENANCE STRATEGY

Modification of software tools of different architecture levels of the smart building BAS due to the elimination of design defects and patching of vulnerabilities leads to a change in the parameters of the failure and recovery flows of the system. As it was shown in the previous Chapters, it is preferable to use the apparatus of Markov and semi-Markov processes to study systems with variable parameters [1,2]. In [3], a systematic approach to the construction of multi fragment models is developed, and in [4], models that take into account reliability and security factors for web systems have been developed. However, in known studies, the influence of different maintenance strategies concerning these factors has not been investigated.

Thus, it is necessary to choose a more acceptable approach for constructing Markov models of BAS availability for common and separate maintenance, taking into account the gradual elimination of software defects and vulnerabilities.

37.1 Formalization of mathematical models for availability of intelligent building I&CS

When studying planning and maintenance procedures of BAS architecture software components, an important step is to obtain quantitative values of the probabilistic components of their availability. The use of the Markov modeling apparatus is associated with a certain set of constraints, which does not allow to construct and apply a single unified model. The output is the construction of a complex of models, in which each model allows to obtain similar result indicators, which are convenient for making comparisons and searching for optimal solutions.

The main aspect of modeling the functioning of BAS architecture software components is accounting for the manifestation and

elimination of limited sets of software defects and vulnerabilities, and these sets are considered as non-overlapping.

The second aspect is maintenance, in the course of which it is possible to identify and eliminate both defects and vulnerabilities. Maintenance procedures can be carried out throughout the BAS lifecycle, or be limited to a certain number of procedures.

The third aspect is the composition of maintenance activities: they can be aimed only at identifying software defects, or only to identify vulnerabilities, or contain a common set of measures to identify both defects and vulnerabilities. A set of basic models is systematized in Table 37.1.

Table 37.1 – Characteristics of the classification for availability models for smart building I&CS

General characteristics of the model	Model specification	Conventional notions
A) Base model without maintenance	-the number of defects 0..Nd - the number of vulnerabilities 0..Nv - the number of maintenances 0	MBAS1
B) Model with common maintenance	- the number of defects 0..Nd - the number of vulnerabilities 0..Nv - the number of maintenances: unlimited during the system whole life cycle - type of maintenance: common	MBAS2.1
	- the number of defects 0..Nd - the number of vulnerabilities 0..Nv - the number of maintenances: 0..Np - type of maintenances: common	MBAS2.2
C) Model with separate	- the number of defects 0..Nd - the number of vulnerabilities	MBAS3.1

maintenance	$0..N_v$ - the number of maintenances: unlimited during the system whole life cycle - type of service: separate	
	- the number of defects $0..N_d$ - the number of vulnerabilities $0..N_v$ - the number of maintenances by defects $0..N_{dp}$, - the number of maintenances by vulnerabilities $0..N_{dv}$ - type of service: separate	MBAS3.2

The time intervals for conducting common and separate maintenances include the periods of testing, elimination of detected defects and vulnerabilities, and verification of the modified software. The procedures for finding defects and vulnerabilities differ both in composition and in duration, and their completeness determines the corresponding probabilities of PCS and PCR.

37.2 Models for availability of information and control systems in smart buildings taking into account reliability and safety procedures

37.2.1 Basic model of availability of BAS architecture taking into account software defects and vulnerabilities (MBAS1)

The basic model describes the processes of manifestation and elimination of software defects and vulnerabilities as separate flows of random events. The initial number of defects (N_d) and vulnerabilities (N_v) are the input parameters of the model. In addition, the input parameters are intensities of random event flows common for all Markov models. In the thesis, an example of the BAS architecture is considered, which at the time of putting into operation contains two software defects and two vulnerabilities. Fig. 37.1 shows its marked graph.

The main assumptions are those about the simplest failure and recovery flows that change the state of the system. After the manifestation of a defect (or vulnerability), the system with the probability PR (PS) stops working until they are completely eliminated. With the probability 1-PR (for defects) or 1-PS (for vulnerabilities) the system returns to the previous operable state through restart of the program. In the course of elimination, new defects and vulnerabilities are not introduced. As defects and vulnerabilities occur, they are gradually eliminated. In the particular case of BAS functioning after the defect or vulnerability manifestations, the system stops until they are completely eliminated (i.e., PR = 1 and PS = 1).

The operable states in Fig. 37.1 are shown in large circles with the number of defects and vulnerabilities in them; Inoperable states are shown in small circles without signatures. In the initial state F(Nd, Nv), the system contains 2 software defects and 2 vulnerabilities.

The manifestation of software defects on the graph is illustrated by diagonal transitions with a downward shift (weighted intensities $\lambda_{Di}(Nd)$), and vulnerabilities – by diagonal transitions with upward shift (weighted intensities $\lambda_{Ij}(Nv)$). After the manifestation of vulnerabilities, they are eliminated with intensities $PS * \mu_{Ij}$, respectively; the elimination of software defects is performed with $PR * \mu_{Di}$ intensities. After all defects and vulnerabilities have been removed, the system goes to the F(0,0) state.

The software restart is illustrated by transitions from inoperable states, weighted intensities $(1-PR) * \mu_{DH_i}$ and $(1-PS) * \mu_{IF_i}$.

The marked state graph and transitions (Fig. 37.2), which includes an endless numbering of states, was constructed using the modified function grPlot_marker. The Kolmogorov SDE is constructed according to the graph of MBAS1 is as follows:

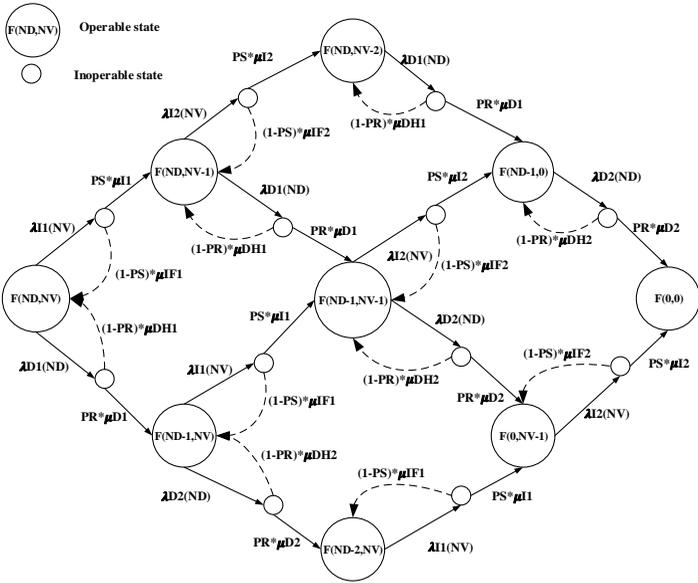


Fig. 37.1 – Marked graph of the base model MBAS1 taking into account the manifestation and elimination of software defects and vulnerabilities (without numbering of states)

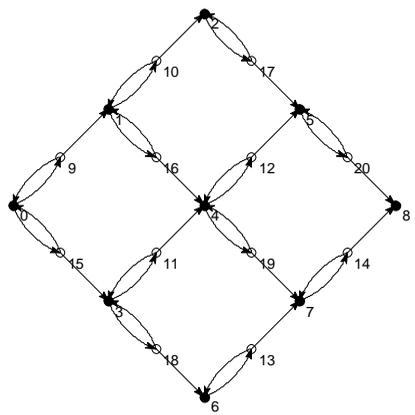


Fig. 37.2 – Marked orgraph of the base model MBAS1 with the numbering of states, built using grPlot_marker

$$\begin{aligned}
dP_0(t)/dt &= -(\lambda I_1 + \lambda D_1)P_0(t) + (1-PS)\mu IF_1 P_9(t) + (1-PR)\mu DH_1 P_{15}(t), \\
dP_1(t)/dt &= -(\lambda I_2 + \lambda D_1)P_1(t) + PS\mu I_1 P_9(t) + (1-PS)\mu IF_2 P_{10}(t) + \\
&\quad + (1-PR)\mu DH_1 P_{16}(t), \\
dP_2(t)/dt &= -\lambda D_1 P_2(t) + PS\mu I_2 P_{10}(t) + (1-PR)\mu DH_1 P_{17}(t), \\
dP_3(t)/dt &= -(\lambda I_1 + \lambda D_2)P_3(t) + (1-PS)\mu IF_1 P_{11}(t) + PR\mu D_1 P_{15}(t) + \\
&\quad + (1-PR)\mu DH_2 P_{18}(t), \\
dP_4(t)/dt &= -(\lambda I_2 + \lambda D_2)P_4(t) + PS\mu I_1 P_{11}(t) + (1-PS)\mu IF_2 P_{12}(t) + \\
&\quad + PR\mu D_1 P_{16}(t) + (1-PR)\mu DH_2 P_{19}(t), \\
dP_5(t)/dt &= -\lambda D_2 P_5(t) + PS\mu I_2 P_{12}(t) + PR\mu D_1 P_{17}(t) + (1-PR)\mu DH_2 P_{20}(t), \\
dP_6(t)/dt &= -\lambda I_1 P_6(t) + (1-PS)\mu IF_1 P_{13}(t) + PR\mu D_2 P_{18}(t), \\
dP_7(t)/dt &= -\lambda I_2 P_7(t) + PS\mu I_1 P_{13}(t) + (1-PS)\mu IF_2 P_{14}(t) + PR\mu D_2 P_{19}(t), \\
dP_8(t)/dt &= PS\mu I_2 P_{14}(t) + PR\mu D_2 P_{20}(t), \\
dP_9(t)/dt &= -((1-PS)\mu IF_1 + PS\mu I_1)P_9(t) + \lambda I_1 P_0(t), \\
dP_{10}(t)/dt &= -((1-PS)\mu IF_2 + PS\mu I_2)P_{10}(t) + \lambda I_2 P_1(t), \\
dP_{11}(t)/dt &= -((1-PS)\mu IF_1 + PS\mu I_1)P_{11}(t) + \lambda I_1 P_3(t), \\
dP_{12}(t)/dt &= -((1-PS)\mu IF_2 + PS\mu I_2)P_{12}(t) + \lambda I_2 P_4(t), \\
dP_{13}(t)/dt &= -((1-PS)\mu IF_1 + PS\mu I_1)P_{13}(t) + \lambda I_1 P_5(t), \\
dP_{14}(t)/dt &= -((1-PS)\mu IF_2 + PS\mu I_2)P_{14}(t) + \lambda I_2 P_6(t), \\
dP_{15}(t)/dt &= -((1-PR)\mu DH_1 + PR\mu D_1)P_{15}(t) + \lambda D_1 P_0(t), \\
dP_{16}(t)/dt &= -((1-PR)\mu DH_2 + PR\mu D_2)P_{16}(t) + \lambda D_2 P_1(t), \\
dP_{17}(t)/dt &= -((1-PR)\mu DH_1 + PR\mu D_1)P_{17}(t) + \lambda D_1 P_2(t), \\
dP_{18}(t)/dt &= -((1-PR)\mu DH_2 + PR\mu D_2)P_{18}(t) + \lambda D_2 P_3(t), \\
dP_{19}(t)/dt &= -((1-PR)\mu DH_1 + PR\mu D_1)P_{19}(t) + \lambda D_1 P_4(t), \\
dP_{20}(t)/dt &= -((1-PR)\mu DH_2 + PR\mu D_2)P_{20}(t) + \lambda D_2 P_5(t), \\
\sum_{i=0}^{20} P_i(t) &= 1; \\
P_0(0) &= 1; \forall i \in [1..20] \Rightarrow P_i(0) = 0.
\end{aligned} \tag{37.1}$$

Table 37.2 – Input parameter values of the MBAS1 model

#	Name	Mathlab-name	Time interval	Value	Measur. Unit
1.	The intensity of the first software defect manifestation $\lambda D1$	laR(1)	5,45 years	5e-4	1/hour
2.	The intensity of the second software defect manifestation $\lambda D2$	laR(2)	6,09 years	4.5e-4	1/hour
3.	The intensity of the first software vulnerability manifestation $\lambda I1$	laS(1)	0,91 year	3e-3	1/hour
4.	The intensity of the second software vulnerability manifestation $\lambda I2$	laS(2)	0,78 year	3.5e-3	1/hour
5.	The intensity of recovery with elimination of the first software defect $\mu D1$	muR(1)	2 hours	0.5	1/hour
6.	The intensity of recovery with elimination of the second software defect $\mu D1$	muR(2)	2,5 hours	0.4	1/hour
7.	The intensity of recovery with elimination of the first software vulnerability $\mu I1$	muS(1)	2,22 hours	0.45	1/hour
8.	The intensity of recovery with elimination of the second software vulnerability $\mu I2$	muS(2)	2,94 hours	0.34	1/hour
9.	The intensity of the restart without elimination of software defects $\mu DH1 = \mu DH2$	muRH	12 minutes	5	1/hour
10.	The intensity of the restart without elimination of software vulnerabilities $\mu IF1 = \mu IF2$	muSF	10 minutes	6	1/hour
11.	The probability of the software defect elimination during recovery	PR		0.9	

12.	The probability of the software vulnerability elimination during recovery	PS		0.9	
13.	The number of software defects in the system	Nd		2	
14.	The number of software vulnerabilities in the system	Nv		2	

To solve the SDE, the method ode15s was used in the Matlab system for the time interval of [0 ... 50000] hours. To construct the matrix of the Kolmogorov-Chapman system of differential equations, we use the matrixA function [4]. To solve the system of differential equations, the built-in solver Matlab ode15s is used. The availability function is defined as:

$$A(t) = \sum_{i=0}^{(Nd+1)(Nv+1)-1} P_i(t) \quad (37.2)$$

The results of the simulation are shown in Fig.37.3. The graph of the model has the following character of the change in the availability function. At the first stage, the availability of the system is reduced to the minimum, and then it asymptotically tends to the established value.

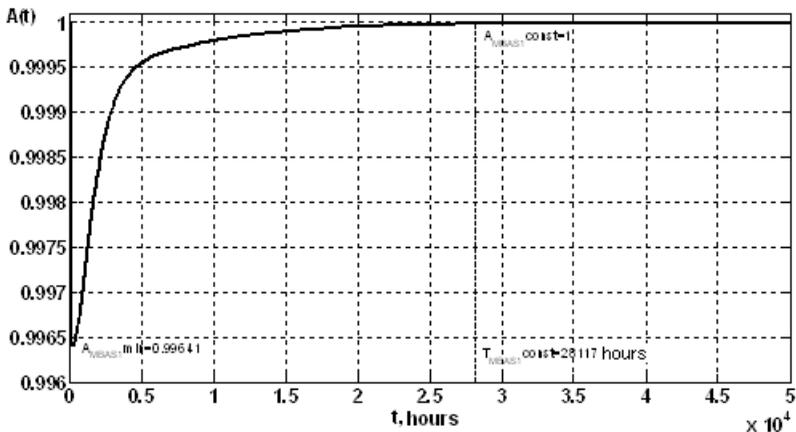


Fig. 37.3 – Results of modeling the availability of the BAS architecture (the resulting indicators are determined with the error of 10^{-5})

Thus, with further analysis of the results, it is necessary to take into account three parameters:

- the minimum value of the availability function $A_{MBAS1min} = 0$;
- the value of the availability function in the steady state $A_{MBAS1const} = 1$;
- the time interval for the transition of the availability function to the steady state $T_{MBAS1const} = 28117$ hours.

In a system without maintenance and provided absence of defects and vulnerabilities, availability asymptotically tends to 1. Therefore, it is of further interest to investigate the impact of individual parameters on the values of the availability function at the minimum point and the time interval for the transition of the availability function to the steady state. For the MBAS1 model, the following parameters were selected (Table 37.3):

Table 37.3 – The boundaries of the variable values of the input data of MBAS1

Name	Mathlab-name	Value row	Measuring unit
The number of software vulnerabilities in the system	Nv	[0..4]	
The probability of the software defect elimination during recovery	PR	[0..1]	
The restart intensity without elimination of software vulnerabilities	muSF	[4..10]	1/hour

The results of modeling in the form of graphical dependencies are shown in Fig.37.4-Fig.37.6.

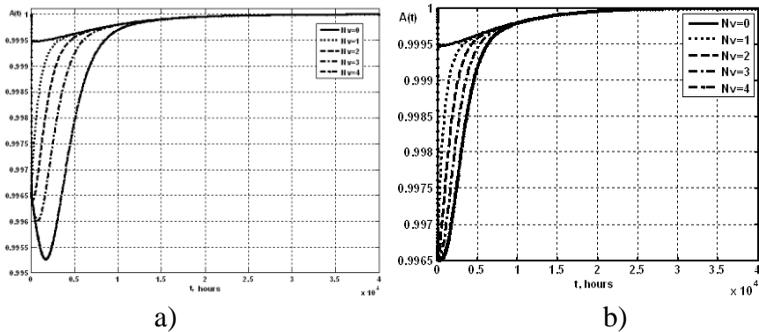


Fig. 37.4 – Graphs of changing the MBAS1 availability model for different numbers of vulnerabilities N_v : (a) – with $\lambda I = \text{var}$, $\mu I = \text{var}$; (b) with $\lambda I = \text{const}$, $\mu I = \text{const}$

The graphs in Fig.37.4 clearly illustrate the behavior of the availability function with different number of vulnerabilities. Obviously, in a system with a large number of vulnerabilities, the latter will be eliminated with a longer time interval. But due to the presence of processes of software defect manifestation and elimination (which is illustrated by the curve with $N_v = 0$), the period of transition of the availability function to the steady state for systems with different number of vulnerabilities remained at the level of $T_{\text{MBAS1const}} = 28117$ hours. Fig.37.4 (a) illustrates the dependence of the minimum of the availability function on the parameter N_v , but this dependence is of an indirect nature, since the increase in N_v contributes to the dynamics of the parameters λI and μI . For the purity of the experiment, additional studies were carried out, during which the parameters λI and μI did not change with the increase in the number of N_v vulnerabilities. The result is shown in Fig.37.4 (b), and it is well illustrated that with the growth of N_v , the minimum of the availability function does not change ($A_{\text{MBAS1min}} = 0.9965$).

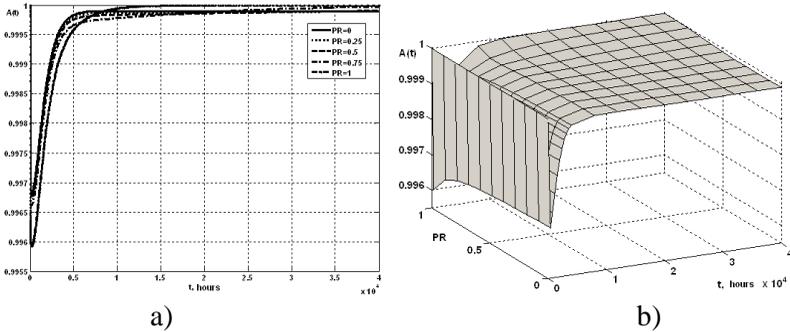


Fig. 37.5 – Two- (a) and three-dimensional (b) graphs of the change in the availability function of the MBAS1 model for different values of the probability of eliminating the software defect during recovery

The analysis of the graphs in Fig. 37.5 (a) showed that with the growth of the parameter PR, the process of transition of the availability function to the steady state is accelerated. It is also obvious that when $PR = 0$, the availability function will never reach a single value ($A(t)=1$ under $t \rightarrow \infty$), since instead of eliminating the defects of the software, the system will be continuously restarted. The three-dimensional graph in Fig. 37.5 (b) gives more visualization of the availability function behavior depending on the PR parameter. The dependence of the minimum of the availability function on the PR parameter is clearly visible: at $PR = 1$, the value of $A_{MBAS1 \min} = 0.996$; with a decrease of PR to zero the value of $A_{MBAS1 \min}$ asymptotically tends to $A_{MBAS1 \min} = 0.9969$.

The analysis of the graph in Fig. 37.6 (b) showed that the value of the μSF parameter (the intensity of the system restart after the manifestation of the vulnerability in the software) will depend on the minimum of the availability function, at $\mu SF = 10$ (1/hour) $A_{MBAS1 \min} = 0.9974$; and under $\mu SF = 4$ (1/hour) $A_{MBAS1 \min} = 0.9957$. This dependence is non-linear, which is well illustrated by the three-dimensional graph. The two-dimensional graphs in Fig. 37.6 (a) show that the parameter μSF does not affect the rate of transition of the availability function to the steady state. This is due to the influence of manifestation and elimination processes of software defects.

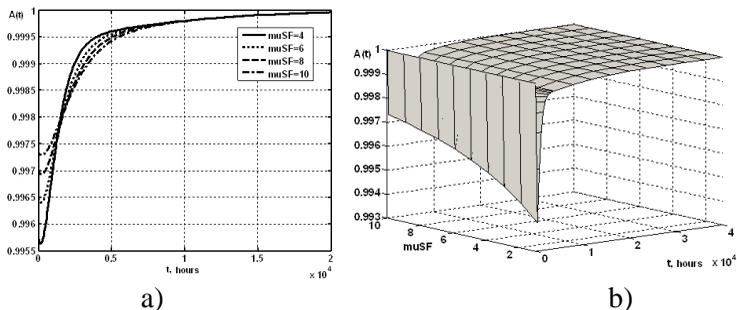


Fig. 37.6 – Two- (a) and three-dimensional (b) graphs of the change in the availability function of the MBAS1 model at different values of the restart intensity without eliminating software vulnerabilities

37.2.2 The BAS availability model taking into account common service (MBAS2.1)

This model is an extension of the basic one and includes additional states that allow modeling of the maintenance procedures. The marked graph of the model is shown in Fig. 37.7. When constructing the graph of the model, to increase the visibility it was assumed that the defect or vulnerability was completely eliminated without restarting the system (i.e., $PR = PS = 1$). However, this assumption concerns only the graphic image in Fig. 37.7 (a); Fig. 37.7 (b); and the subsequent simulation results take into account the restart of the system. In addition to the assumptions listed above, the MBAS2 model assumes that during the common maintenance, it is possible to detect and eliminate one software defect or one vulnerability.

The states simulating common maintenance procedures are shown by shaded circles. The transitions to maintenance states are performed from operational states with a maintenance rate λM_j . In the process of maintenance activities, the detection of a software defect occurs with the PCR probability, the detection of vulnerability – with the PCS probability. Simultaneous detection of the software vulnerability and defect occurs with the probability of $PCR \cdot PCS$. The probability of PF undetectable defects and vulnerabilities complements previous events to the full group:

$$PF+PCS+PCR+PCS*PCR=1. \quad (37.3)$$

Thus, four transitions are possible from the maintenance state:

a) if a vulnerability with a PCS probability is detected, a vertical upward transition is performed, weighted by the $PCS*\mu Ms$ intensity, where μMs is the inverse of the mean detection time and elimination of the vulnerability [5], $\mu Ms = 1 / (TdetV + TremV)$;

b) in case of detection of a software defect with a PCR probability, a vertical downward transition is performed, weighted by the intensity of $PCR*\mu Mr$, where μMr is the inverse value of the mean detection time and elimination of the defect [6], $\mu Mr = 1 / (TdetD + TremD)$;

c) in case of detection of a software defect and a vulnerability with a $PCS*PCR$ probability, a right-hand transition weighted by the $PCS*PCR*\mu Mrs$ intensity is performed, where μMrs is the inverse of the mean detection and elimination time of the defect and vulnerability,

$$\mu Mrs = \frac{\mu Mr \cdot \mu Ms}{\mu Mr + \mu Ms} ; \quad (37.4)$$

d) if the defect and the vulnerability are not detected with PF probability, a return to the previous working state (to the left) weighted by the intensity $PF*\mu Mt$ is performed, where μMt is the inverse of the average maintenance time, $\mu Mt=1/T_M$.

It should be noted that in this model, we consider maintenance operations that do not anticipate the number of defects and vulnerabilities. Therefore, after removing all vulnerabilities, the transitions from the maintenance states simulating the defect detection are weighted by the parameter $(1-PCR)*\mu Mt$.

Similarly, transitions simulating the detection of a vulnerability after the removal of all software defects are weighted by the parameter $(1-PCS)*\mu Mt$. The extreme right state, in which maintenance of the system without defects and vulnerabilities is simulated, has, respectively, a transition weighted by the μMt parameter. The marked orgraph is presented in Fig. 37.7 (b).

To construct the matrix of the Kolmogorov-Chapman system of differential equations, we use the matrixA function [4]. The Kolmogorov SDE solution was performed in the Matlab system using the ode15s method for the time interval [0 ... 50000] hours. The availability function is determined by (37.2). The results of the solution are presented graphically in Fig. 37.8.

Table 37.4 – Values of the input parameters of the MBAS2.1 model

#	Name	Mathlab-name	Time interval	Value	Measur. Unit
1.	The intensity of the first software defect manifestation $\lambda D1$	laR(1)	5,45 years	5e-4	1/hour
2.	The intensity of the second software defect manifestation $\lambda D2$	laR(2)	6,09 years	4.5e-4	1/hour
3.	The intensity of the first software vulnerability manifestation $\lambda I1$	laS(1)	0,91 year	3e-3	1/hour
4.	The intensity of the second software vulnerability manifestation $\lambda I2$	laS(2)	0,78 year	3.5e-3	1/hour
5.	The intensity of recovery with elimination of the first software defect $\mu D1$	muR(1)	2 hours	0.5	1/hour
6.	The intensity of recovery with elimination of the second software defect $\mu D1$	muR(2)	2,5 hours	0.4	1/hour
7.	The intensity of recovery with elimination of the first	muS(1)	2,22 hours	0.45	1/hour

	software vulnerability $\mu I1$				
8.	The intensity of recovery with elimination of the second software vulnerability $\mu I2$	$\mu S(2)$	2,94 hours	0.34	1/hour
9.	The intensity of the restart without elimination of software defects $\mu DH1 = \mu DH2$	μRH	12 minutes	5	1/hour
10.	The intensity of the restart without elimination of software vulnerabilities $\mu IF1 = \mu IF2$	μSF	10 minutes	6	1/hour
11.	The probability of the software defect elimination during recovery	PR		0.9	
12.	The probability of the software vulnerability elimination during recovery	PS		0.9	
13.	The number of software defects in the system	Nd		2	
14.	The number of software vulnerabilities in the system	Nv		2	
15.	The intensity of maintenance common by vulnerabilities and defects λMj	λMj	100 hours	$1e-2$	1/hour
16.	The intensity of common maintenance activities μMt	μMt	2,5 hours	0.4	1/hour
17.	The intensity of detection and elimination of vulnerabilities μMs	μMs	5 hours	0.2	1/hour
18.	The intensity of detection and elimination of defects μMr	μMr	3,33 hours	0.3	1/hour
19.	The probability of vulnerability detection during maintenance procedures	PCS		0.4	
20.	The probability of software	PCR		0.2	

	defect detection during				
	maintenance procedures				

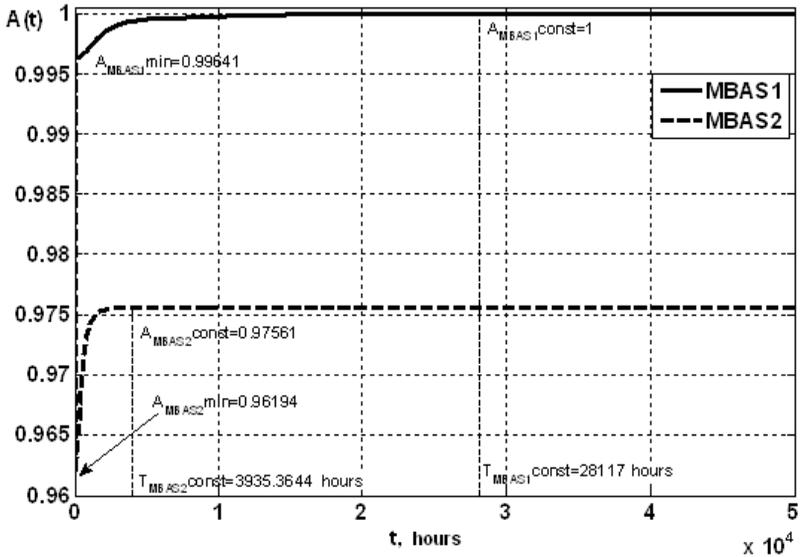


Fig. 37.8 – Graphs of the change in the BAS availability function without maintenance (MBAS1) and with the common maintenance (MBAS2.1) (the resulting indicators are determined with the error of 10^{-5})

The results of the simulation are shown in Fig. 37.8. The graphs of the models have the same nature of the change in the availability function. At the first stage, the availability of the system is reduced to the minimum, then it asymptotically tends to the established value. Thus, with further analysis of the results, it is necessary to take into account three parameters:

- the minimum value of the availability function $A_{MBAS_i;min}$ (for the MBAS1 model – 0.9964, for the MBAS2.1 model – 0.96194);
- the value of the availability function in the steady state $A_{MBAS_i;const}$ (for MBAS1 model – 1, for MBAS2.1 model – 0.97561);
- the time interval for the transition of the availability function to the steady state $T_{MBAS_i;const}$ (for the MBAS1 model – 28117 hours, for the MBAS2.1 model – 3935.36 hours).

As can be seen from the graphs in Fig. 37.8, carrying out maintenance activities reduces both the established value of the availability function and its minimum. The MBAS2.1 model is characterized by a desire for availability to the value determined by the extreme right fragment:

$$A_{MBAS\ 2.1}^{const} = \frac{\mu Mt}{\lambda Mj + \mu Mt}, \quad (37.5)$$

accordingly, the input parameters λMj and μMt will affect the value of $A_{MBAS\ 2}^{const}$.

Therefore, it is of further interest to investigate the impact of individual parameters on the values of the availability function at the minimum point and the time interval for the transition of the availability function to the steady state.

Given the constraint (37.2), in the MBAS2.1 model, the PCS and PCR parameters can simultaneously assume a maximum value of $\sqrt{2}-1 = 0.4142$. Otherwise, given the time limit for services, it is possible to "bias" both the identification of vulnerabilities and the detection of software defects. That is, with $PCR = 1 \rightarrow PCS = 0$ and vice versa, with $PCS = 1 \rightarrow PCR = 0$.

In this regard, there arises a problem of finding the optimal, from the point of view of minimizing the time for eliminating defects and vulnerabilities, distributing measures for their detection in the common maintenance cycle. Let us consider the following statement of the problem. In the system with 6 defects and 2 vulnerabilities, we need to determine the values of PCR and PCS, under which $T_{MBAS\ i}^{const} \rightarrow \min$. In this case, it is necessary to further analyze the indirect impact of parameter selection on the value of $A_{MBAS\ 2.1}^{min}$.

To solve the problem, there is an accepted assumption about the ideality of the measures for identifying defects and vulnerabilities ($PF=0$), but it will be removed in the future. The values of the variable input parameters are presented in Table 37.5.

At $PF = 0$, the value of the PCS parameter is defined as:

$$PCS = \frac{1 - PCR}{1 + PCR} \quad (37.6)$$

Table 37.5 – The boundaries of the variable values of the MBAS2.1 model input data

Name	Mathlab-name	Value row
The number of software defects in the system	Nd	[0..6]
The probability of software defect detection and elimination during common maintenance	PCR	[0..1]

To investigate the impact of these parameters, special cyclic software constructs were developed. The results of modeling in the form of graphical dependencies are shown in Fig. 37.9.

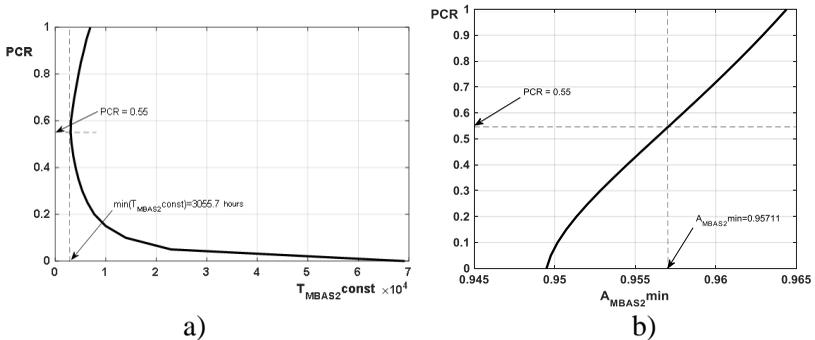


Fig. 37.9 – Graphs of the dependence of the resulting parameters $T_{MBAS\ 2.1\ const}$ (a) and $A_{MBAS\ 2.1\ min}$ (b) model with the common maintenance (MBAS2.1) on the input PCR parameter

The simulation results showed that the minimum achievable time $T_{MBAS\ 2.1\ const} = 3055.7$ hours is achieved with the PCR value of 0.55 (in addition, another parameter is $PCS = 0.29$). However, it should be taken into account that the value of the second result parameter $A_{MBAS\ 2.1\ min} = 0.95711$ is in the middle of the curve in Fig. 37.9, b, i.e., the minimization is performed only by the parameter $T_{MBAS\ i\ const}$.

Based on the studies carried out, the values of the PCR input parameter depend on the initial number of defects under the condition of $T_{MBAS_i;const} \rightarrow \min$.

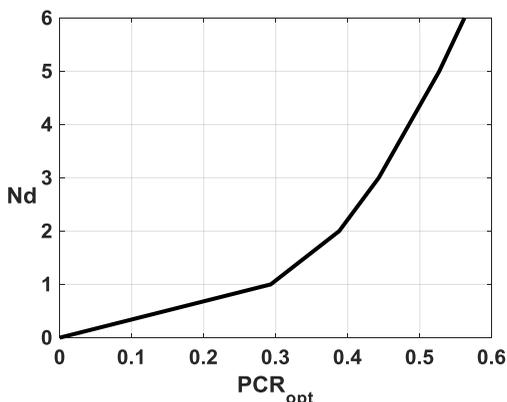


Fig. 37.10 – Graph of the dependence between the optimal PCR parameter (according to the $T_{MBAS_i;const} \rightarrow \min$ criterion) of the common maintenance model (MBAS2.1) and the initial number of defects in the Nd system

The values of PCRopt are tabulated and are presented in Table 37.5. Fig. 37.11 shows the dependence of PCRopt on the input parameters Nd and Nv in three-dimensional space.

Table 37.5 – Tabulated PCRopt values

Nd Nv	0	1	2	3	4	5	6
0	0	1	1	1	1	1	1
1	0	0,357	0,481	0,544	0,585	0,629	0,677
2	0	0,293	0,388	0,443	0,485	0,527	0,562
3	0	0,254	0,320	0,365	0,436	0,466	0,489
4	0	0,214	0,280	0,329	0,380	0,412	0,430
5	0	0,190	0,246	0,292	0,329	0,360	0,406
6	0	0,167	0,224	0,263	0,308	0,340	0,361

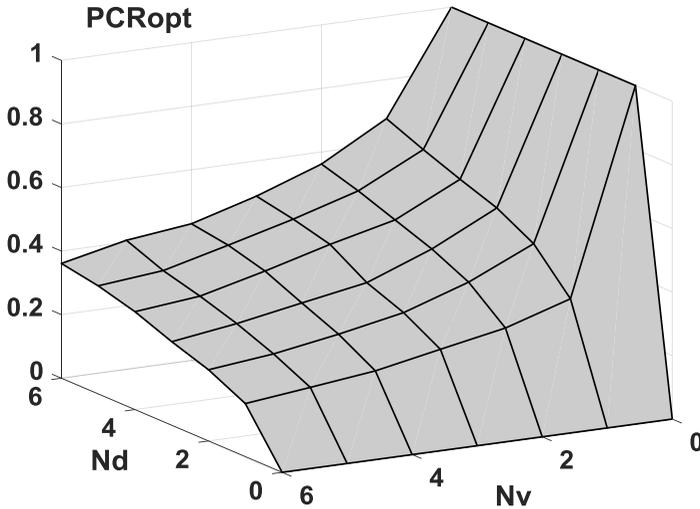


Fig. 37.11 – Three-dimensional graph for the dependence of the optimal PCR parameter (according to the $T_{MBAS;const} \rightarrow \min$ criterion) in the common maintenance model (MBAS2.1) on the initial number of Nd defects and the Nv vulnerabilities in the system

We will further consider the impact of the PF parameter on the values of $A_{MBAS 2.1 \min}$ and $T_{MBAS;const}$. In the process of condition fulfillment, the assumption is made about the uniformity of efforts aimed at identifying defects and vulnerabilities in the common maintenance process ($PCR = PCS$). Under such condition, the probability of undetectability of defects and vulnerabilities in the maintenance process varies from 0 (at $PCR = PCS$) to 1 (at $PCR = PCS = 0$).

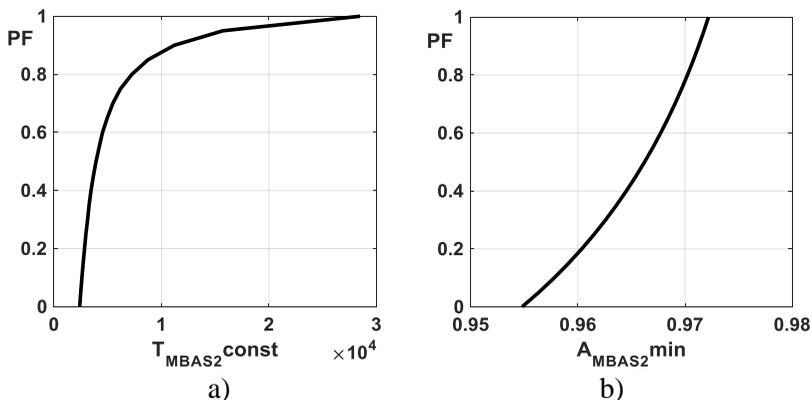


Fig. 37.12 – The graph for the dependence of the resulting parameters $T_{MBAS\ 2.1}const$ (a) and $A_{MBAS\ 2.1}min$ (b) of the model with common maintenance (MBAS2.1) on the input PF parameter

The simulation results (Fig. 37.12) illustrate the fact that the undetection of vulnerabilities and defects in the course of common maintenance delay the time of their elimination (the resulting parameter $T_{MBAS\ 2}const$ increases with the probability PF to 1). In this case, the value of the resulting indicator $A_{MBAS\ 2.1}min$ improves due to the fact that the common maintenance procedures without eliminating defects and vulnerabilities are shorter ($\mu Mt > \mu Ms$, $\mu Mt > \mu Mr$ and $\mu Mt > \mu Mrs$).

37.2.3 The BAS availability model taking into account separate maintenance (MBAS3.1)

The model is also extended with respect to the basic MBAS1 and includes additional states of the separate maintenance procedures. Unlike the previous model, MBAS2.1, the number of maintenance states is doubled, since we consider maintenance procedures, the purpose of which is to identify only software defects, and vice versa, only vulnerabilities. The marked graph of the model is shown in Fig.37.13.

When constructing the graph of the model, to increase the visibility it was assumed that the defect or vulnerability was completely eliminated without restarting the system (i.e., $PR = PS = 1$). But this assumption concerns only the graphic representation in Fig. 37.13 (a), Fig. 37.13 (b) and the subsequent simulation results take into account the restart of the system.

The states that simulate separate maintenance procedures are shown by circles with different strokes. Transitions to maintenance states are performed from operable states: to vulnerability maintenance states – with the maintenance intensity λMs ; to maintenance states for software defects – with the intensity λMr . Since separate maintenance is considered, two complete groups of events are formed: the detection of vulnerability in the maintenance process with the probability of PCS and undetection of vulnerability with probability $(1-PCS)$; detection of a software defect in the maintenance process with a probability of PCR and undetection a defect with probability $(1-PCR)$.

Two transitions are performed from each maintenance state for the vulnerabilities: the first one with the intensity $PCS * \mu Ms$ simulates the identification and elimination of the service vulnerability; the second one with the intensity $(1-PCS) * \mu Mt$ simulates maintenance without revealing vulnerability. If all vulnerabilities are removed, the transition from the maintenance state is weighted by the μMt intensity. Similarly, there is a simulation of transitions from maintenance states to software defects. Transitions with the intensity of $PCR * \mu Mr$ simulate the identification and elimination of a software defect in maintenance; transitions with intensity $(1-PCR) * \mu Mt$ simulate maintenance without detecting defects. If all defects are eliminated, the transitions from the maintenance state are weighted by the μMt intensity. The marked orgraph shown in Fig. 37.13 (b).

To construct the matrix of the Kolmogorov-Chapman system of differential equations, we use the matrixA function [4]. The Kolmogorov SDE solution was performed in the Matlab system using the ode15s method for the time interval of $[0 \dots 50000]$ hours. The availability function is determined by (37.1). The results of the solution are presented graphically in Fig. 37.14.

Table 37.6 – Values of the input parameters of the MBAS3.1 availability model

#	Name	Mathlab-name	Time interval	Value	Measur. Unit
1.	The intensity of the first software defect manifestation $\lambda D1$	laR(1)	5,45 years	5e-4	1/hour
2.	The intensity of the second software defect manifestation $\lambda D2$	laR(2)	6,09 years	4.5e-4	1/hour
3.	The intensity of the first software vulnerability manifestation $\lambda I1$	laS(1)	0,91 year	3e-3	1/hour
4.	The intensity of the second software vulnerability manifestation $\lambda I2$	laS(2)	0,78 year	3.5e-3	1/hour
5.	The intensity of recovery with elimination of the first software defect $\mu D1$	muR(1)	2 hours	0.5	1/hour
6.	The intensity of recovery with elimination of the second software defect $\mu D1$	muR(2)	2,5 hours	0.4	1/hour
7.	The intensity of recovery with elimination of the first software vulnerability $\mu I1$	muS(1)	2,22 hours	0.45	1/hour
8.	The intensity of recovery with elimination of the second software vulnerability $\mu I2$	muS(2)	2,94 hours	0.34	1/hour
9.	The intensity of the restart without elimination of software defects $\mu DH1 = \mu DH2$	muRH	12 minutes	5	1/hour

10.	The intensity of the restart without elimination of software vulnerabilities $\mu IF1 = \mu IF2$	μSF	10 minutes	6	1/hour
11.	The probability of the software defect elimination during recovery	PR		0.9	
12.	The probability of the software vulnerability elimination during recovery	PS		0.9	
13.	The number of software defects in the system	Nd		2	
14.	The number of software vulnerabilities in the system	Nv		2	
15.	The intensity of maintenance common by vulnerabilities and defects λMj	λMj	1000 hours	$1e-3$	1/hour
16.	The intensity of separate maintenance by vulnerabilities λMs	λMj	200 hours	$5e-3$	1/hour
17.	The intensity of separate maintenance by defects λMr	λMr	1000 hours	$1e-3$	1/hour
18.	The intensity of common maintenance performance μMt	μMt	2,5 hours	0.4	1/hour
19.	The intensity of detection and elimination of vulnerabilities μMs	μMs	5 hours	0.2	1/hour
20.	The intensity of detection and elimination	μMr	3,33 hours	0.3	1/hour

	of defects μMr			
21.	The probability of vulnerability detection during maintenance procedures	PCS		0.4
22.	The probability of software defect detection during maintenance procedures	PCR		0.2

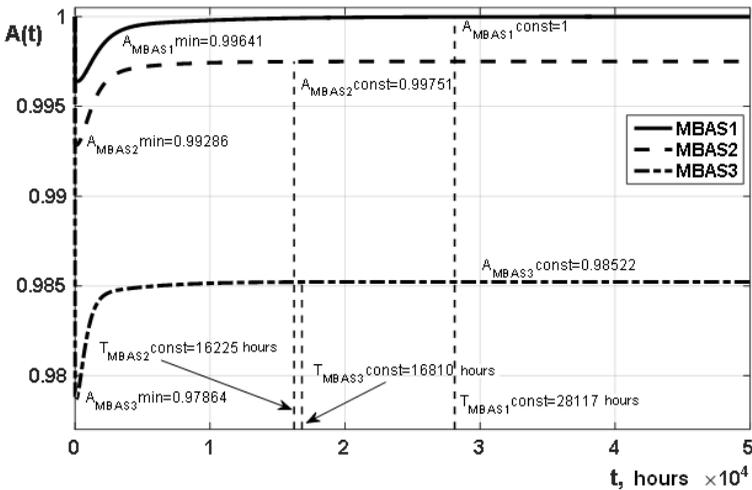


Fig. 37.14 – Graphs of the change in the availability function of the BAS without maintenance (MBAS1), with the common maintenance (MBAS2.1) and separate maintenance (MBAS3.1) (the resulting indicators are determined with the error of 10^{-5})

The simulation results are shown in Fig. 37.14. The graphs of the models have the same nature of the change in the availability function. At the first stage the availability of the system is reduced to the minimum, and then it asymptotically tends to the established value. Thus, with further analysis of the results, it is necessary to take into account three parameters:

- the minimum value of the availability function $A_{MBAS_i,min}$ (for the MBAS1 model – 0.99641, for the MBAS2.1 model – 0.99286, for the MBAS3.1 model – 0.97864);

- the availability value in the steady state $A_{MBAS_i,const}$ (for the MBAS1 – 1 model, for the MBAS2.1 model – 0.9975, for the MBAS3.1 model – 0.9852);

- the time interval for the transition of the availability function to the steady state $T_{MBAS_i,const}$ (for the MBAS1 model – 28117 hours, for the MBAS2.1 model – 16225 hours, for the MBAS3.1 model – 16810 hours).

As can be seen from the graphs in Fig. 37.14, carrying out maintenance activities reduces both the established value of the availability function and its minimum. Due to the accepted assumptions about the gradual elimination of defects and vulnerabilities, the availability of the system without maintenance asymptotically tends to 1.

For models with maintenance, the desire of availability to the value determined by the extreme right fragment is typical, which for the separate maintenance is:

$$A_{MBAS\ 3.1}const = \frac{\mu Mt}{\lambda Mr + \lambda Ms + \mu Mt} \quad (37.7)$$

This can explain the gain of the model with the common maintenance by the indicators of the minimum of the availability function (by 0.0142) and the stationary value of the availability function (by 0.0123).

Carrying out the maintenance allows 1.73 times to speed up the identification and elimination of defects and vulnerabilities. In this case, the difference in $T_{MBAS_i,const}$ indicators for models with common and separate maintenance is insignificant (less than 1%). But here it is necessary to take into account the fact that MBAS2.1 and MBAS3.1 models were given the same probability values for detecting PCS and PCR defects and vulnerabilities. And if in the model MBAS3.1 PCS and PCR can vary in the range of 0.1 simultaneously, then in the MBAS2.1 model the parameters PCS and PCR can simultaneously take the maximum value of 0.4142.

Further, we are interested in the study of the influence of individual parameters on the values of the availability function at the minimum point and the time interval for the transition of the availability function to the steady state.

Unlike MBAS2.1, in the current model, PCS and PCR parameters can simultaneously change the value on the interval [0..1]. It is expected that with better detectability of defects and vulnerabilities (PCS = 1 and PCR = 1), there will be an acceleration of the transition of the availability function to the steady state. Then the interest is the problem of studying the impact of the PCS and PCR parameters on the minimum of the availability function of $A_{\text{MBAS 3.1}}^{\text{min}}$ with different number of defects and vulnerabilities. In addition, the indirect influence of the input parameters on the value of $T_{\text{MBAS 3.1}}^{\text{const}}$ should be further analyzed.

Table 37.7 – The boundaries of the variable values of the MBAS3.1 model input data

Name	Mathlab-name	Value row
The number of software defects in the system	Nd	[0..6]
The number of software vulnerabilities in the system	Nv	[0..6]
The probability of detection and elimination a software defect during separate maintenance	PCR	[0..1]
The probability of detection and elimination a software vulnerability during separate maintenance	PCS	[0..1]

To study the impact of these parameters, special cyclic program constructs were developed. The results of modeling in the form of graphical dependencies are shown in Fig.37.15.

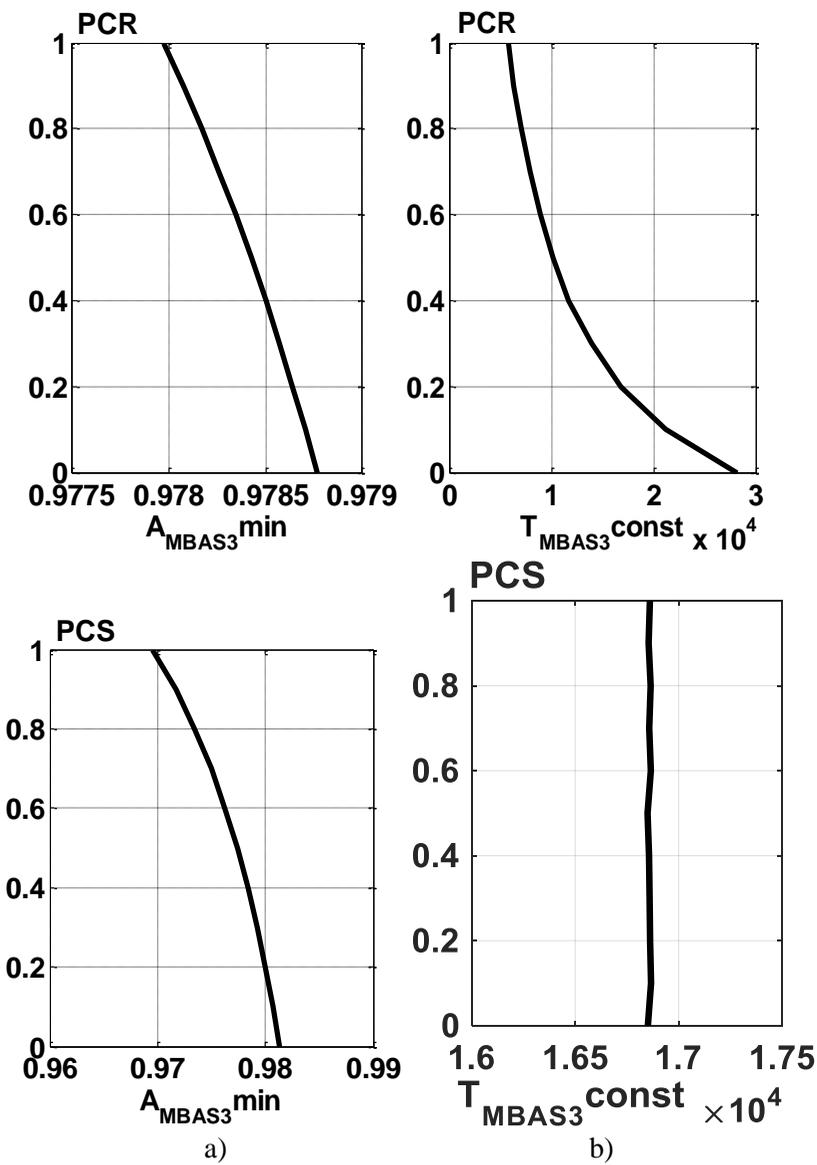


Fig. 37.15 – Graph of the dependence of the resulting parameters $A_{MBAS\ 3.1} \text{ min}$ (a) and $T_{MBAS\ 3.1} \text{ const}$ (b) of the model with separate maintenance (MBAS3) on the input PCS and PCR parameters

The analysis of the graphs in Fig. 37.15 confirms the optimality of the parameter $PCR = 1$ in the MBAS3 model, with the optimality being performed both by the $T_{MBAS\ 3.1const} \rightarrow \min$ criterion and by the $A_{MBAS\ 3.1min} \rightarrow \min$ criterion. At $PCS = 1$, the optimality is observed by the criterion $A_{MBAS\ 3.1min} \rightarrow \min$.

The most interesting were the results of the studying the influence of the PCS parameter values on the resulting indicator $T_{MBAS\ 3const}$. If we look at Fig. 37.15 (d), then it seems that the $T_{MBAS\ 3.1const}$ values vary randomly with the change in the PCS. However, the spread between the obtained values of $T_{MBAS\ 3.1const}$ does not exceed 16 hours, which is $3.4e-5$ relative to the boundaries of the investigated time interval. Therefore, in the received configuration, the values of the input PCS parameter have no impact on the $T_{MBAS\ 3.1const}$ result. This is explained by the fact that the intensity of the maintenance by vulnerabilities is five times greater than the maintenance intensity by defects, therefore, for any PCS, the system will more get in states of maintenance by vulnerabilities.

Further, it is advisable to compare the models with the common and separate maintenance according to the resulting $T_{MBAS\ iconst}$ indicator for the optimal values of the input parameters PCS and PCR.

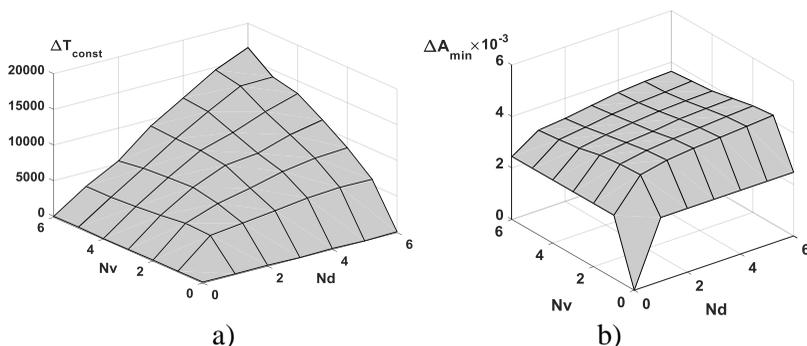


Fig. 37.16 – Dependence of the resultant difference $\Delta T_{MBAS\ iconst}$ (a) and $\Delta A_{MBAS\ imin}$ (b) for models with separate and common service on the input parameters N_d and N_v

During the comparison, the values of the intensities of common and separate maintenance were assumed equal to $\lambda M_j = \lambda M_s = \lambda M_r = 1e-3$ (1/hour). To increase the visibility, the results are shown in the form of the dependence of the difference $\Delta T_{const} = T_{MBAS\ 3.1\ const} - T_{MBAS\ 2.1\ const}$ on the dimension of the sets of input defects and vulnerabilities (N_d and N_v).

If there are no defects ($N_d = 0$) or vulnerabilities ($N_v = 0$) at the initial moment of time or $N_v=0$, models with common and separate maintenance show a commensurate rate of elimination of vulnerabilities ($N_d=0, N_v=[1..6]$) or defects ($N_d=[1..6], N_v=0$): the difference between the indicators $T_{MBAS\ i\ const}$ does not exceed 102 hours. This can be explained by the fact that in the model with common maintenance under such conditions the corresponding optimal parameter $PCR = 1$ ($PCS = 1$) is adopted.

However, if there are defects and vulnerabilities in the system ($N_d > 0, N_v > 0$), the advantage of the model with separate maintenance is evident, where defects and vulnerabilities are eliminated faster. This advantage (illustrated by the difference ΔT_{const}) increases with the initial number of defects and vulnerabilities. In addition, Fig.37.16 (b) illustrates the weak dependence of the difference $\Delta A_{MBAS\ i\ min}$ on the number of defects and vulnerabilities; its dynamics does not exceed 10^{-4} .

37.2.4 BAS availability model with a limited number of common maintenances (MBAS2.2)

This model describes the functioning of the system in the context of common maintenance activities, but unlike the MBAS2.1 model, the number of such activities throughout the life cycle is limited.

The simulation reflects the following principle: at the planning stage of the maintenance procedures, developers can only assume the number of undetected defects and vulnerabilities. In addition, when planning common maintenance, it is impossible to know in advance what will be revealed: a defect, a vulnerability, or both defect and vulnerability. Therefore, it is planned to conduct a certain number of N_p maintenance procedures.

Fig. 37.17 shows a marked graph of the BAS architecture with two defects and two vulnerabilities ($N_d = 2, N_v = 2$), in which six ($N_p = 6$)

common maintenance operations are performed. The parameter N_p corresponds to the number of vertical diagonals of the rhomboid Fig. of orgraph (on which the common maintenance states are located). The logic of model functioning in this case is the following: the first maintenance ($N_p = 1$) is carried out after the system is put into operation and its state has. Next, different paths of transitions over the states of the model are possible, therefore, the second maintenance ($N_p = 2$) has two probable states and is carried out either after the defect is eliminated (transition from the state $F(N_d-1, N_v)$), or after the vulnerability is removed (transition from the state $F(N_d, N_v-1)$) or skipped (if during the first service both the defect and the vulnerability are eliminated). The third maintenance ($N_p = 3$) has already three possible states (with transitions from the states $F(N_d, N_v-2)$, $F(N_d-1, N_v-1)$, $F(N_d-2, N_v)$) and also can be skipped if in the course of the second maintenance both the defect and the vulnerability have been identified and eliminated. The fourth maintenance ($N_p = 4$) has two possible states (with transitions from the states $F(N_d-1, 0)$, $F(0, N_v-1)$); the fifth and sixth maintenances have one probable state (with the transition from the state $F(0, 0)$).

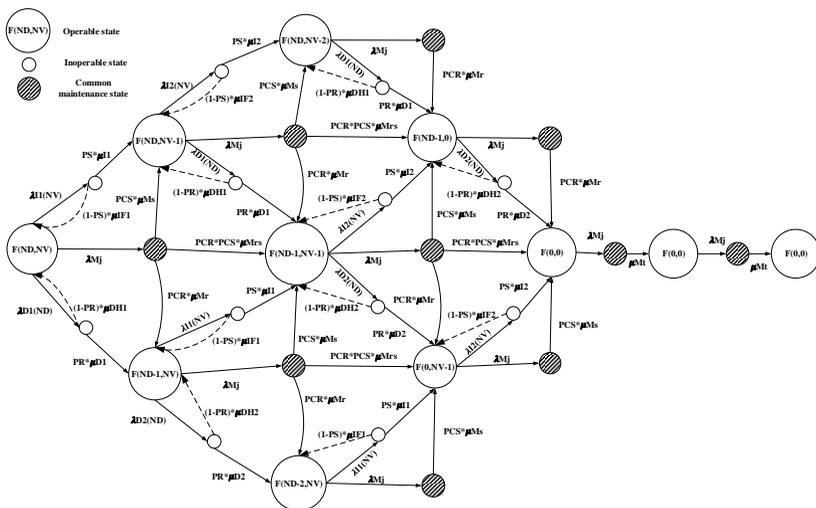


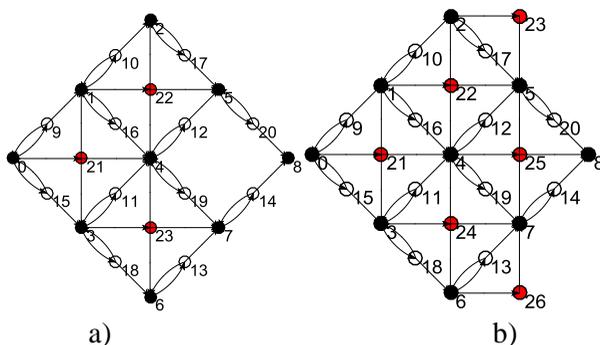
Fig. 37.17 – Marked graph of the MBAS2.2 model taking into account the limited number of common maintenances ($N_p = 6$)

The "indicator" of the termination of common maintenance operations is the counter of their number. However, in the model, such a counter can only be used if the states of the service are passed once, i.e., under the condition of absolute effectiveness of the maintenance operations ($PF = 0$).

When constructing a model, it is necessary to take into account three versions of the forecasts of the number of common maintenance operations:

- a) $N_p < N_d + N_v$;
- b) $N_p = N_d + N_v$;
- c) $N_p > N_d + N_v$.

The marked orgraphs of the models constructed taking into account these variants of the forecasts are shown in Fig. 37.18. Fig. 37.18 a and b show orgraphs of the system with two defects and vulnerabilities, in which the number of scheduled maintenance operations does not exceed 4 (two for Fig. 37.18 a and three for Fig. 37.18b). Fig. 37.18c shows the orgraph of the model, in which the predicted number of maintenance operations ($N_p = 6$) covers all the diagonals and corresponds to the actual number of defects and vulnerabilities in the system. The graph of the model shows that immediately after the elimination of all defects and vulnerabilities, the maintenance procedures are terminated.



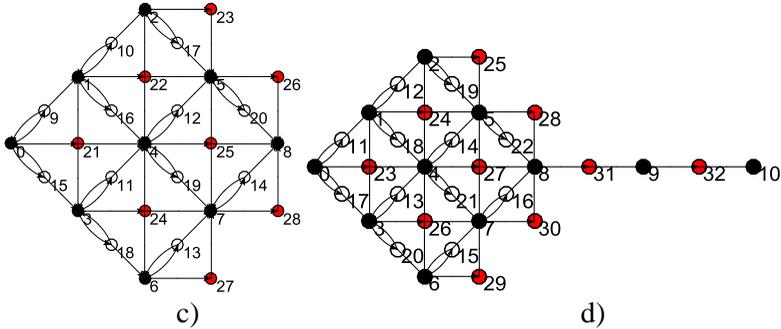


Fig. 37.18 – Marked orgraph of the MBAS2.2 model taking into account the limited number of common maintenance $N_p = 2$ (a), $N_p = 3$ (b), $N_p = 4$ (c), $N_p = 6$ (d).

The orgraph of the model MAS2.2, in which the number of maintenances ($N_p = 6$) exceeds the real number of diagonals in the system ($N_d + N_v = 4$), is shown in Fig. 37.18. As it can be seen from the graph, after the elimination of all defects and vulnerabilities, the common maintenance procedures are carried out for two more periods, and then terminated. In this regard, the availability function covers additional states and is calculated as:

$$A(t) = \sum_{i=0}^{(Nd+1)*(Nv+1)+Np-(Nd+Nv)-1} P_i(t) \quad (37.8)$$

The calculation of the availability indicators is made for the input data from Table 37.7. The values of the PCR parameters are taken from Table 37.5, the parameter PCS is determined from (37.6). To construct the matrix of the Kolmogorov-Chapman system of differential equations, we use the matrixA function [4]. The Kolmogorov SDE solution was performed in the Matlab system using the ode15s method for the time interval [0 ... 50000] hours. The availability function is determined by (37.2). The results of the solution are presented graphically in Fig. 37.19.

Table 37.7 – Values of the input parameters of the MBAS2.2 model

#	Name	Mathlab-name	Time interval	Value	Measur. unit.
1.	The intensity of the first software defect manifestation $\lambda D1$	laR(1)	5,45 years	5e-4	1/year
2.	The intensity of the second software defect manifestation $\lambda D2$	laR(2)	6,09 years	4.5e-4	1/year
3.	The intensity of the first software vulnerability manifestation $\lambda I1$	laS(1)	0,91 year	3e-3	1/year
4.	The intensity of the second software vulnerability manifestation $\lambda I2$	laS(2)	0,78 year	3.5e-3	1/year
5.	The intensity of recovery with elimination of the first software defect $\mu D1$	muR(1)	2 hours	0.5	1/year
6.	The intensity of recovery with elimination of the second defect $\mu D1$	muR(2)	2,5 hours	0.4	1/year
7.	The intensity of recovery with elimination of the first software vulnerability $\mu I1$	muS(1)	2,22 hours	0.45	1/year
8.	The intensity of recovery with elimination of the second software vulnerability $\mu I2$	muS(2)	2,94 hours	0.34	1/year
9.	The intensity of the restart without elimination of software defects $\mu DH1 = \mu DH2$	muRH	12 minutes	5	1/year
10.	The intensity of the restart without elimination of software vulnerabilities $\mu IF1 = \mu IF2$	muSF	10 minutes	6	1/year
11.	The probability of the software defect elimination during recovery	PR		0.9	

12.	The probability of the software vulnerability elimination during recovery	PS		0.9	
13.	The number of software defects in the system	Nd		2	
14.	The number of software vulnerabilities in the system	Nv		2	
15.	The intensity of maintenance common by vulnerabilities and defects λM_j	laMj	100 minutes	1e-2	1/year
16.	The intensity of common maintenance procedures μM_t	muMt	2,5 minutes	0.4	1/year
17.	The intensity of detection and elimination μM_s	muMs	5 minutes	0.2	1/year
18.	The intensity of defect detection and elimination μM_r	muMr	3,33 minutes	0.3	1/year
19.	The probability of vulnerability detection during maintenance procedures	PCS		0.4409	
20.	The probability of defect detection during maintenance procedures	PCR		0.388	
21.	Predicted number of common maintenance	Np		2	

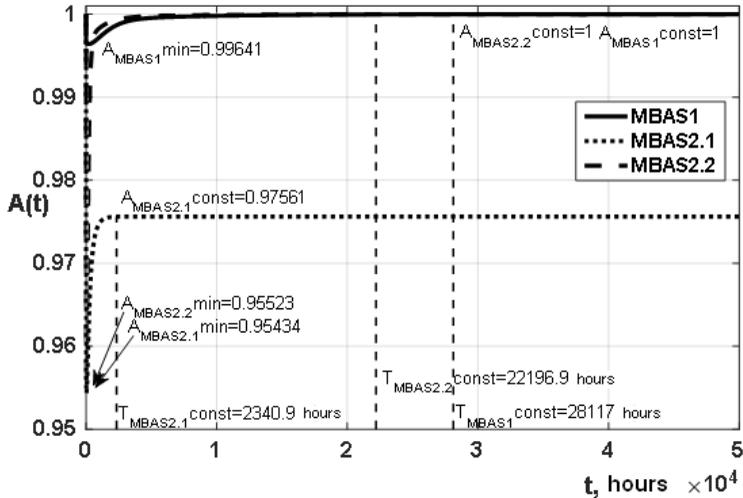


Fig. 37.19 – Graphs of the change in the availability function of the BAS architecture without maintenance (MBAS1), with the common unlimited (MBAS2.1) and limited (MBAS2.2) maintenance (the resulting indicators are determined with the error of 10^{-5})

The analysis of the graphs in Fig. 37.19 showed that the limitation of the number of maintenances in the MBAS2.2 model allows achieving the ideal availability ($A_{MBAS\ 2.2\ const=1}$) in the steady state. At the same time, the value of the availability minimum for models with limited and unlimited maintenance differs insignificantly (by $8.83e-4$). The transition period for the availability function in the MBAS2.2 mode is 9.48 times higher than that of the MBAS2.1 model with unlimited common maintenance; however, the elimination of defects and vulnerabilities in the model with maintenance is faster than in the MBAS1 model (1.27 times).

Since interest is caused by a decrease in the detection and elimination of all defects and vulnerabilities, then further we consider the influence of individual input parameters on the resulting indicator $T_{MBAS\ 2.2\ Const}$ (in addition, their impact on $A_{MBAS\ 2.2\ min}$ is analyzed). In this case, the dimensionality of the model is increased to $N_d=3$, $N_v=3$, the value of the PCR parameter is also taken from Table 37.5.

Table 37.8 – The boundaries of the variable values of the MBAS2.2 model input data

Name	Mathlab-name	Value row	Measur.unit
Predicted number of common maintenances	N_p	[0..10]	
The intensity of maintenance common by vulnerabilities and defects λM_j	laMj	[1e-2..1e-4]	1/hour

To study the impact of these parameters, special cyclic program constructs were developed. The results of simulation in the form of graphical dependencies are shown in Fig. 37.20 – Fig. 37.22.

The results of the studying the forecast accuracy impact (N_p) showed the expected result. If the lack of defects and vulnerabilities is predicted ($N_p = 0$), the MBAS2.2 model degenerates into MBAS1 (Fig. 37.20, a) and has the highest level of $A_{MBAS\ 2.2\ min}$ (Fig. 37.20, c). With the growth in the number of limited N_p maintenances up to $N_p = 6$, the process of identifying and eliminating defects and vulnerabilities as a whole is accelerating. In this case, the graph of the change of $T_{MBAS\ 2.2\ const}$ in Fig. 37.20, d has a specific appearance of a broken curve: up to the limit $N_p \leq N_v + N_d$, it shows a decrease in the resultant index and for $N_p > N_v + N_d$, the value of $T_{MBAS\ 2.2\ const}$ increases with N_p (as unsuccessful maintenance procedures are accumulated). A noticeable explanation in the behavior of $A_{MBAS\ 2.2\ min}(N_p)$ at $N_p = 5$ is given by the fact that with such a number of maintenances the "availability" is provided from the maintenance state of the extreme right operable state S15 (Fig.37.21, a). In this case, in Fig. 37.20, a, it is clear that with the appearance of excessive maintenances ($N_p = 6, N_p = 8$), the minimum of the availability function shifts along the time axis to the right.

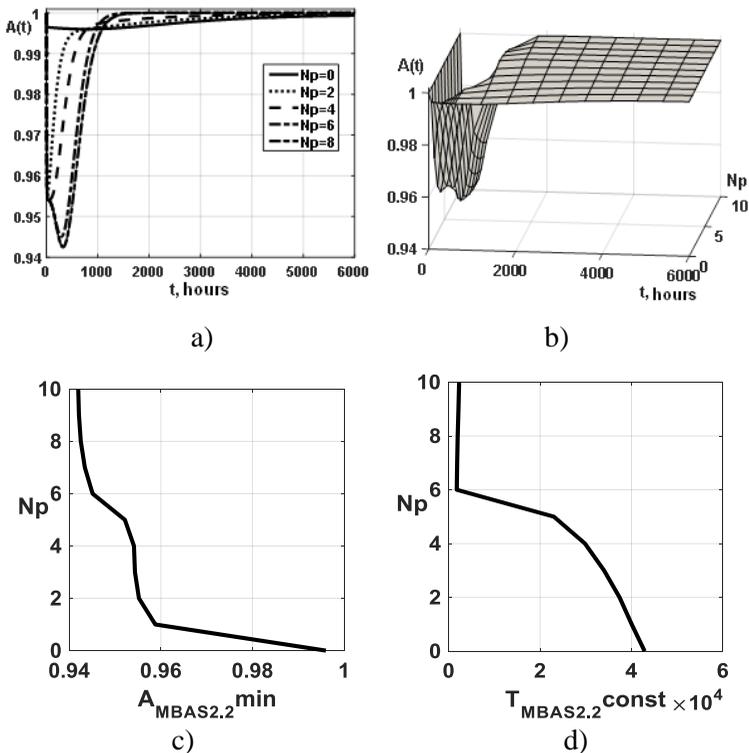


Fig. 37.20 – Graphs of the change in the resulting indicators of the MBAS2.2 model (a, b – availability functions, c – minimum availability function, d – transition period to the steady state with the error of 10^{-5}) with a limited number of common maintenances N_p

In the course of the study, it was determined that the minimum resulting indicators of $T_{\text{MBAS 2.2 const}}$ are achieved with a forecast of $N_p = 6$, the marked graph for this forecast is shown in Fig. 37.21, b.

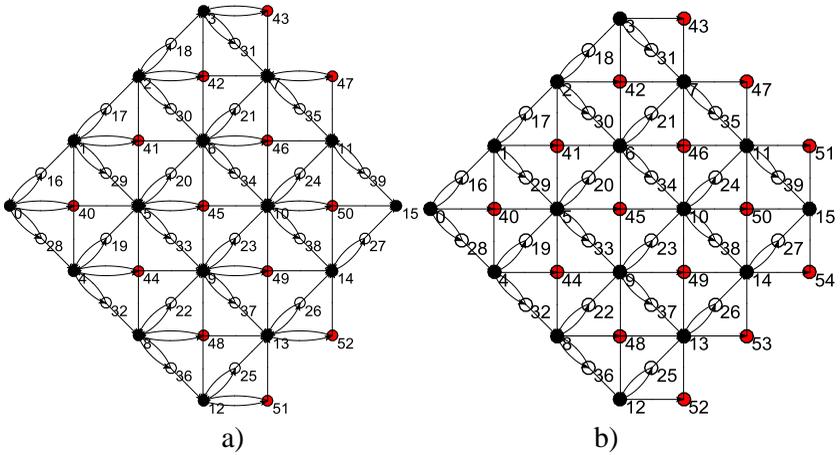
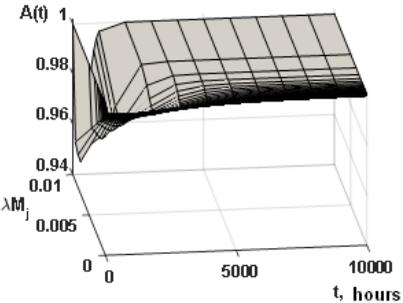
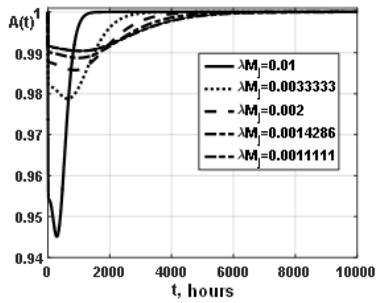


Fig. 37.21. –Orgraphof the BAS architecture, $N_p = 5$ (a) and optimal according to $T_{MBAS\ 2.2const} \rightarrow \min$ criterion of the BAS architecture, $N_p = 6$ (b)

Further, the impact of maintenance intensity, common by the vulnerabilities and defects λM_j , on the resulting parameters of $T_{MBAS\ 2.2const}$ and $A_{MBAS\ 2.2min}$, is considered. When constructing models, the values of the input parameters $N_v = N_d = 3$, $N_p = 6$ were adopted.



a) b)

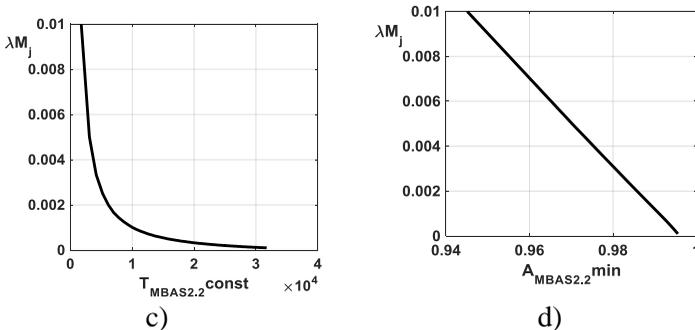


Fig. 37.22 – Graphs of the change in the resulting indicators of the MBAS2.2 model (a, b – availability functions, c – minimum availability function, d – transition period to the steady state with the error of 10^{-5}) from the maintenance intensity λ_{Mj}

The results given in Fig. 37.22 also show the expected result: the more frequent the maintenance procedures are, the faster the defects and vulnerabilities will be identified and corrected. The value of the resulting indicator $A_{MBAS2.2} \text{ min}$ decreases linearly.

37.2.5 The BAS availability model taking into account the limited number of separate maintenance (MBAS3.2)

This model describes system functioning in the context of separate maintenance activities, but unlike the MBAS3.1 model, the number of such activities throughout the life cycle is limited.

Simulation shows the same principle as in the MBAS2.2 model: at the planning stage of the maintenance procedures, developers can only assume the number of undetected defects and vulnerabilities. But unlike the common maintenance model, the MBAS3.2 model knows for sure that only vulnerabilities will be fixed during the maintenance of vulnerabilities, and only defects will be eliminated during defect maintenance. Therefore, in the MBAS3.2 model, the N_{dp} and N_{vp} input parameters determine the planned number of maintenances for defects and vulnerabilities, respectively.

The marked graph of the model is shown in Fig. 37.23. When constructing the graph of the model to increase the visibility, it was

assumed that the defect or vulnerability was completely eliminated without restarting the system (i.e., $PR = PS = 1$). But this assumption concerns only the graphic representation in Fig. 37.23; subsequent simulation results take into account the restart of the system.

The graph in Fig. 37.23 is the BAS model with two defects and two vulnerabilities ($N_d = 2, N_v = 2$), and it additionally describes three maintenances by defects ($N_{dp} = 3$) and one maintenance by vulnerability ($N_{vp} = 1$). Unlike the MBAS2.2 model, the planned number of maintenances (for example, over defects) determines not the number of vertical diagonals of the rhomboid Fig. of the orgraph, but corresponds to inclined lines in the direction of the shift when eliminating defects (right-down). In detecting and eliminating defects, the logic of the functioning of the MBAS3.2 model is the following: the first maintenance ($N_{dp} = 1$) is performed after the system is put into operation and has three probable states (with transitions from the states $F(N_d, N_v), F(N_d, N_v-1), F(N_d, N_v-2)$). After maintenance, the detected defect is eliminated, therefore, the second maintenance ($N_{dp}=2$) also has three probable states (with transitions from the states $F(N_d-1, N_v), F(N_d-1, N_v-1), F(N_d-1, 0)$). Since only two defects were initially present in the system, the third maintenance by defects is redundant and an additional fragment is required for its modeling in the graph (it is shown by a dashed Fig. line). The third maintenance also has three probable states.

Since only one maintenance is planned for the vulnerabilities, it will have four probable states with transitions from the states $F(N_d, N_v), F(N_d-1, N_v), F(N_d-2, N_v), F(N_d-2, N_v)$. The second vulnerability will be eliminated only after its manifestation.

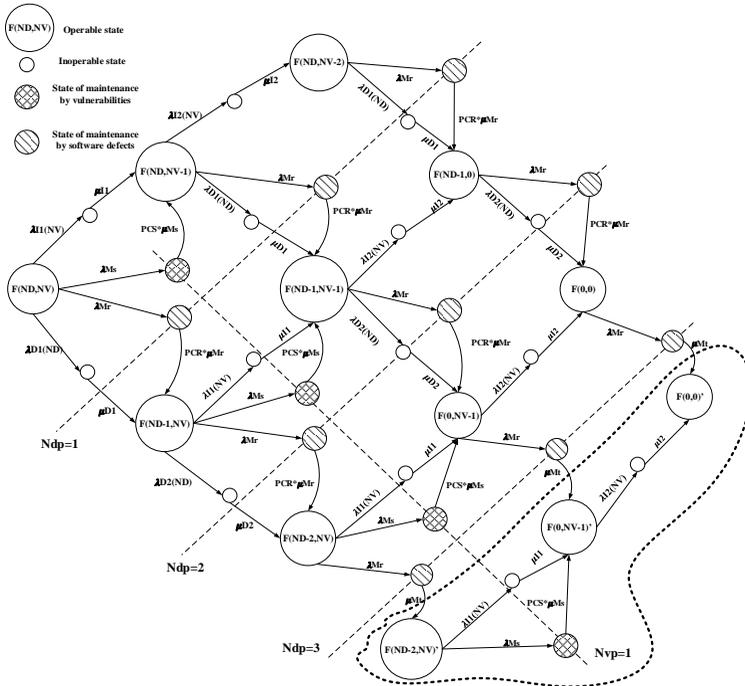


Fig. 37.23 – Marked graph of the MBAS3.2 model taking into account the limited number of separate maintenances by defects ($Ndp = 3$) and vulnerabilities ($Nvp = 1$)

When building the model, it is necessary to take into account four variants of the forecasting the initial number of defects and vulnerabilities:

- a) $(Ndp \leq Nd) \& (Nvp \leq Nv)$
- b) $(Ndp \leq Nd) \& (Nvp > Nv)$;
- c) $(Ndp > Nd) \& (Nvp \leq Nv)$;
- d) $(Ndp > Nd) \& (Nvp > Nv)$.

The marked orgraphs of models constructed with these forecast options are shown in Fig. 37.24. Fig. 37.24, a shows the orgraph of the system with two defects and vulnerabilities, in which the number of maintenances by defects/vulnerabilities does not exceed 2 (two by vulnerabilities and one by defects). To improve the visibility of the state of maintenance over defects are shown in yellow circles, over

vulnerabilities – in green. Fig. 37.24, b shows the orgraph of the model, in which the predicted number of maintenance by vulnerabilities exceeds their number in the system. This causes the occurrence of additional operable (S3, S7, S11, S15) and inoperable (S27, S31, S35, S51) states.

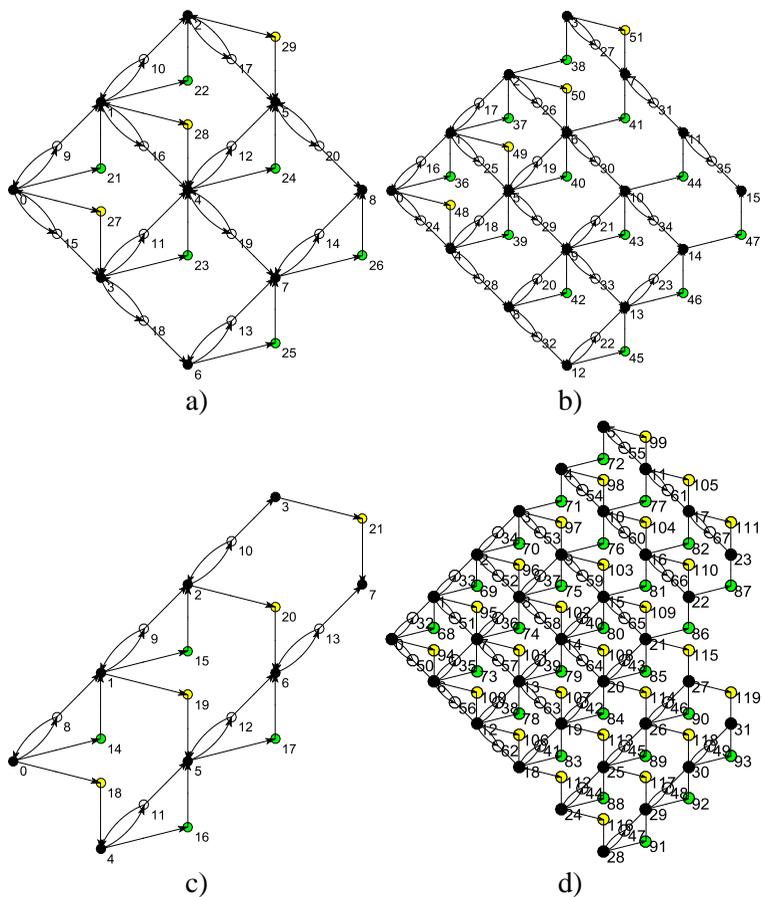


Fig. 37.24 – Marked orgraph of MBAS3.2 model taking into account the limited number of separate maintenances for configurations:
 a) $N_d=2, N_v=2, N_{dp}=1, N_{vp}=2$; 6) $N_d=3, N_v=2, N_{dp}=1, N_{vp}=3$;
 b) $N_d=0, N_v=3, N_{dp}=1, N_{vp}=2$; r) $N_d=3, N_v=3, N_{dp}=5, N_{vp}=5$.

Fig. 37.24, c shows the orgraph of the model, in which defects are absent, but one maintenance is planned to be according to defects. This causes the occurrence of additional operable (S4, S5, S6, S7) and inoperable (S11, S12, S13, S16, S17) states. The orgraph of the MBAS3.2 model, in which the number of planned maintenances by both defects and vulnerabilities (Ndp = 5, Nvp = 5) exceeds their real number in the system (Nd = Nv = 3) and is shown in Fig.37.24. As can be seen from the graph, after the elimination of all defects and vulnerabilities, the maintenance procedures are carried out for two more periods, and then terminated. In this regard, the availability function covers additional states and is calculated as:

$$A(t) = \sum_{i=0}^N P_i(t) \tag{37.9}$$

$$N = (Nd+1) \cdot (Nv+1) + (Nd+1) \times$$

$$\times (\max(Nvp, Nv) - Nv) + (Nv+1) \times$$

$$\times (\max(Ndp, Nd) - Nd)$$

The calculation of the availability indicators is performed for the input data from Table 37.9. For comparison with the MBAS2.2 model, the latter model has the PCR taken from Table 37.5; the PCS parameter is determined by (37.6). To construct the matrix of the Kolmogorov-Chapman system of differential equations, we use the matrixA function [4]. The Kolmogorov CDS solution was performed in the Matlab system using the ode15s method for the time interval of [0 ... 50000] hours. The availability function is determined by (37.9). The results of the solution are presented in the graphical form in Fig. 37.25.

Table 37.9 – Values of the input parameters of the MBAS3.2 model

#	Name	Mathlab-name	Time interval	Value	Measur. unit
1.	The intensity of the first software defect manifestation λD1	laR(1)	5,45 years	5e-4	1/year
2.	The intensity of the second software defect manifestation	laR(2)	6,09 years	4.5e-4	1/year

	$\lambda D2$				
3.	The intensity of the first software vulnerability manifestation $\lambda I1$	laS(1)	0,91 years	3e-3	1/year
4.	The intensity of the second software vulnerability manifestation $\lambda I2$	laS(2)	0,78 years	3.5e-3	1/year
5.	The intensity of recovery with elimination of the first software defect $\mu D1$	muR(1)	2 hours	0.5	1/year
6.	The intensity of recovery with elimination of the second software defect $\mu D1$	muR(2)	2,5 hours	0.4	1/year
7.	The intensity of recovery with elimination of the first software vulnerability $\mu I1$	muS(1)	2,22 hours	0.45	1/year
8.	The intensity of recovery with elimination of the second software vulnerability $\mu I2$	muS(2)	2,94 hours	0.34	1/year
9.	The intensity of the restart without elimination of software defects $\mu DH1 = \mu DH2$	muRH	12 minutes	5	1/year
10.	The intensity of the restart without elimination of software vulnerabilities $\mu F1 = \mu F2$	muSF	10 minutes	6	1/year
11.	The probability of the software defect elimination during recovery	PR		0.9	
12.	The probability of the software vulnerability elimination during recovery	PS		0.9	
13.	The number of software defects in the system	Nd		2	
14.	The number of software vulnerabilities in the system	Nv		2	
15.	The intensity of maintenance common by vulnerabilities and	laMj	1000 hours	1e-3	1/year

	defects λM_j				
16.	The intensity of separate maintenance by vulnerabilities λM_s	laMs	200 hours	5e-3	1/year
17.	The intensity of separate maintenance by defects λM_r	laMr	1000 hours	1e-3	1/year
18.	The intensity of common maintenance performance μM_t	muMt	2,5 hours	0.4	1/year
19.	The intensity of detection and elimination of vulnerabilities μM_s	muMs	5 hours	0.2	1/year
20.	The intensity of defect detection and elimination μM_r	muMr	3,33 hours	0.3	1/year
21.	The probability of vulnerability detection during maintenance procedures in the MBAS3.2 model	PCS		1	
22.	The probability of software defect detection during maintenance procedures in the MBAS3.2 model	PCR		1	
23.	The probability of vulnerability detection during maintenance procedures in the MBAS2.2 model	PCS		0.4409	
24.	The probability of software defect detection during maintenance procedures in the MBAS2.2 model	PCR		0.388	
25.	Predicted number of common maintenances in the MBAS3.2 model	Nvp		2	
26.	Predicted number of common maintenances in the MBAS3.2 model	Ndp		2	
27.	Predicted number of common	Np		4	

maintenances in the MBAS2.2 model				
-----------------------------------	--	--	--	--

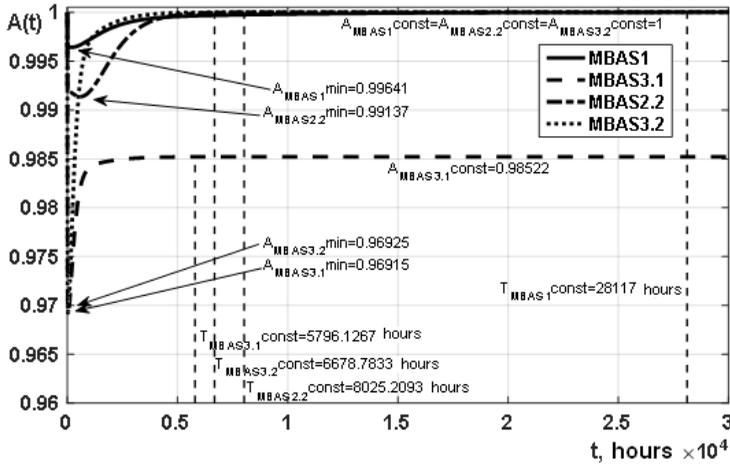


Fig. 37.25 – Graphs of change in the availability function of the BAS architecture without maintenance (MBAS1); with separate unlimited (MBAS3.1), common (MBAS2.2) and separate limited (MBAS3.2) maintenance (the resulting indicators are determined with the error of 10^{-5})

The analysis of the graphs in Fig. 37.25 showed that limiting the number of separate maintenances in the MBAS3.2 model (as in the MBAS2.2 model) allows achieving an ideal availability ($A_{MBAS_{3.2} \text{ Const}=1}$) in the steady. Also as in the previous MBAS2.2 model, the minimum availability value for models with limited and unlimited maintenance differs insignificantly (by $9.73e-5$). However, common maintenance remains an advantageous one according to the $A_{MBAS_i \min}$ (by 0.022) indicator.

If we compare models with limited and unlimited maintenance, then it is clear that the latter (MBAS2.1 in Fig. 37.19 and MBAS3.1 in Fig. 37.25) has a shorter period of transition of the availability function to the steady state. The difference between the resulting $T_{MBAS_i \text{ const}}$ indicators of models MBAS3.1 and MBAS3.2 is 882.6 hours. The transition period for the availability function to the steady state in the

MBAS3.2 model is 1346.4 hours less than in the limited common maintenance MBAS2.2. In addition, eliminating defects and vulnerabilities in the model with maintenance is faster than in the MBAS1 model (4.2 times).

Since interest is caused by a decrease in the detection and elimination of all defects and vulnerabilities, then further we consider the influence of individual input parameters on the resulting indicator $T_{MBAS\ 3.2\ Const}$ (in addition, their impact on $A_{MBAS\ 2.2\ min}$ is analyzed). The dimensionality of the model is increased to $Nd = 3, Nv = 3$.

Table 37.10 – The boundaries of the MBAS3.2 model input values

Name	Mathlab-name	Value row	Measur.unit
Predicted number of separate maintenances	Ndp, Nvp	[0..10]	
The intensity of defect detection and elimination μMr	muMr	[0.1..1]	1/hour

The results of modeling in the form of graphical dependencies are shown in Fig. 37.26 – Fig. 37.27.

Dependence of the resulting indicator $A_{MBAS\ 3.2\ min}$ on the number of separate maintenances is shown in Fig. 37.26, a. Analysis of the three-dimensional graph allows to distinguish the following points. The BAS system without maintenance is optimal according to the criterion $A_{MBAS\ 3.2\ min} \rightarrow \max$ ($Ndp=Nvp=0, A_{MBAS\ 3.2\ min}=0,996$). The system without maintenance by defects ($Ndp = 0, Nvp > 0$) exceeds the system without maintenance by vulnerabilities ($Nvp = 0, Ndp > 0$) by $A_{MBAS\ 3.2\ min}$ by 0.021. In BAS systems with the number of limited separate maintenances greater than the real number of defects and vulnerabilities ($Ndp > 3, Nvp > 3$), the change in $A_{MBAS\ 3.2\ min}$ does not exceed $6.3e-8$.

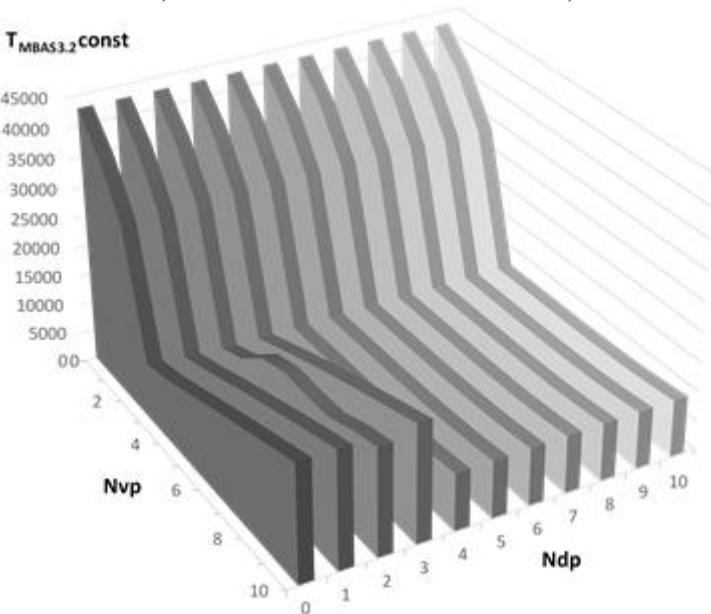
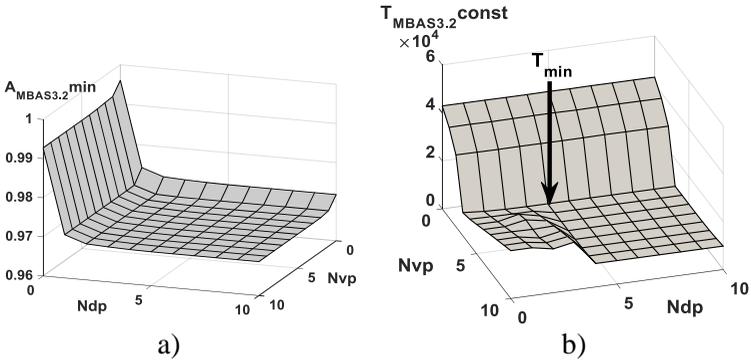


Fig. 37.26 – Graphs of the change in the resulting indicators of the MBAS3.2 model (a – the minimum of the availability function, b – the period of transition to the steady state with the error of 10^{-5}) with a limited number of separate maintenances

Fig. 37.26b shows the dependence of the transition period of the MBAS3.2 availability function in the steady state on the number of separate maintenances. The location of the minimum on the three-

dimensional graph is shown by a special metrics and corresponds to the value $\min(T_{\text{MBAS } 3.2\text{const}})=8496,153$ hours under the configuration of the number of maintenances $N_{\text{vp}} = 3$, $N_{\text{dp}} = 4$. In BAS systems with the number of limited separate maintenances greater than the actual number of defects and vulnerabilities ($N_{\text{dp}} > 3$, $N_{\text{vp}} > 3$), the change in the $T_{\text{MBAS } 3.2\text{const}}$ does not exceed 1256.546489 hours, but there is a growing trend of $T_{\text{MBAS } 3.2\text{const}}$ with an increase in N_{vp} , which is shown in Fig. 37.27.

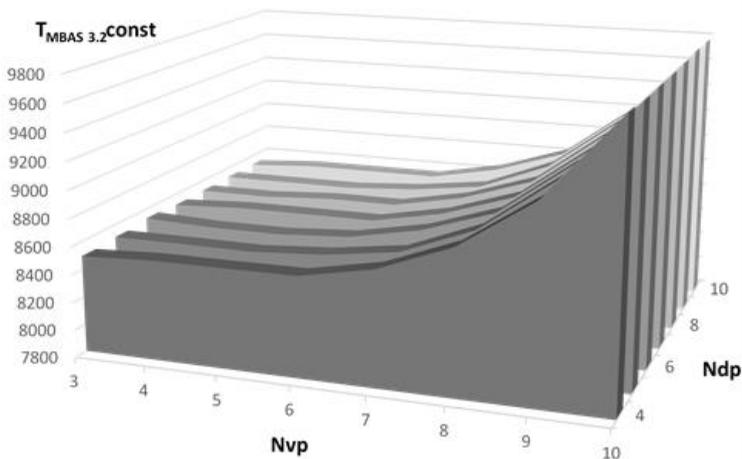


Fig.37.27 – Details of the change of $T_{\text{MBAS } 3.2\text{const}}$ in the MBAS3.2 model on the intervals $N_{\text{dp}} > 3$, $N_{\text{vp}} > 3$

When analyzing the three-dimensional graph in Fig. 37.26, and over $N_{\text{dp}} = \text{const}$, an insignificant chaotic change in the parameter $T_{\text{MBAS } 3.2\text{const}}$ is observed at the intervals $N_{\text{vp}} < 3$ and $N_{\text{vp}} > 3$ under $N_{\text{dp}} > 3$ and for the entire interval $N_{\text{vp}} = [0..10]$ under $N_{\text{dp}} < 3$. This is shown in detail in Fig. 37.28.

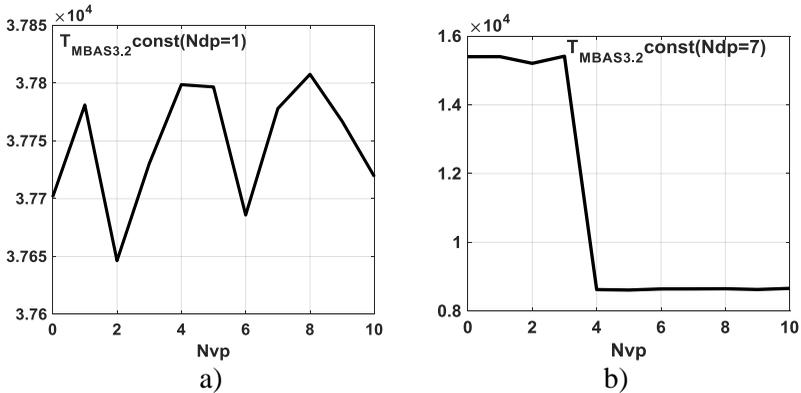
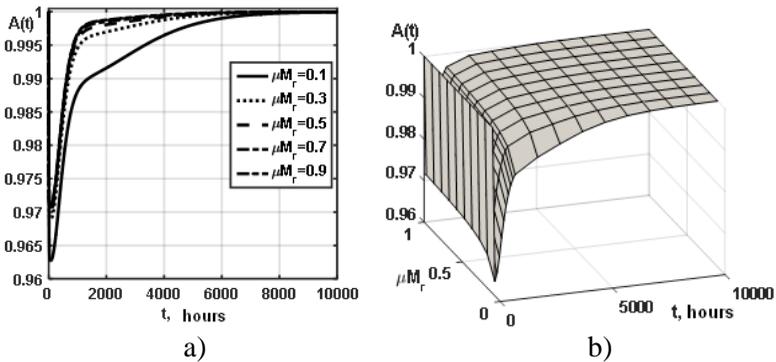


Fig.37.28 – Detailization of the change in $T_{\text{MBAS3.2const}}$ of the model MBAS3.2 on slices $N_{dp} = 1$ (a), $N_{vp} = 7$ (b)

Explanation of this dependence follows from the difference in the input parameters λ_{Ms} and λ_{Mr} – with their accepted values ($\lambda_{Ms} = 5e-3$ and $\lambda_{Mr} = 1e-3$), the transition to the maintenance state by vulnerabilities is performed with greater intensity.

Next, the influence of the intensity of the detecting and eliminating the μ_{Mr} defect on the resulting parameters of $T_{\text{MBAS3.2const}}$ and $A_{\text{MBAS3.2min}}$ is considered. When constructing models, the values of the input parameters $N_v = N_d = 3$, $N_{vp} = 3$, $N_{dp} = 4$ were taken.



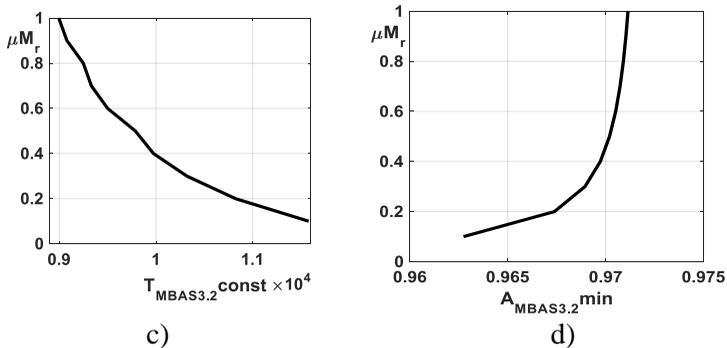


Fig. 37.29 – Graphs of the change in the resulting indicators of the MBAS3.2 model (a, b – availability functions, c – minimum availability function, d – transition period to the steady with the error of 10^{-5}) from the intensity of detection and elimination of the defect μM_r

The results shown in Fig. 37.29 also show the expected result: if the maintenance quickly identifies and corrects defects, then the minimum availability function ($A_{MBAS\ 3.2\ min}$) increases, and the transition period to the steady state decreases. Thus, with a 10-fold acceleration of detection and elimination of defects during maintenance, the value of $A_{MBAS\ 3.2\ min}$ increases by 0.0084, and the period of detection and elimination of all defects and vulnerabilities decreases by 1.2872 times.

37.3 Scaling of availability models for information and control systems of smart buildings

With the expansion of intellectualization systems to the level of the university campus (Fig. 36.7), the number of types of failures and points of cyber-attacks application that determine the state of a system-wide failure potentially increases. Taking into account their step-by-step elimination in the course of security and safety maintenance activities, or after their manifestation, the dimension of the Markov models increases (as the number of model fragments increases). Despite the fact that in this Chapter the typical architecture of BAS for $N_d = 2$ and $N_v = 2$ was considered, the developed models simply scale

to an arbitrary number of defects and vulnerabilities. The increase in the dimensionality of the models was illustrated in Fig. 37.18, Fig. 37.21 and Fig. 37.24; And the results of calculations of models with increased dimensionality, for example, made it possible to construct the dependence of the PCR parameter (according to the $T_{MBAS, const} \rightarrow \min$ criterion) of the common maintenance model (MBAS2.1) on the initial number of defects in the Nd system.

Conclusions

The chapter presents FTA, ATA and Markov models for availability of smart BAS taking into account various variants of recovery and maintenance processes as well as the parameters of software faults and vulnerability attacks.

These models are combined to assess availability, and cyber security, to improve the accuracy of assessing availability indicators and determine the requirements for the coefficient of cyber security and availability (the level of availability of the system in the steady state).

The BAS models and technique considering the different modes and strategies of system maintenance (with and without the elimination of faults and vulnerabilities after their detection, with and without the maintenance procedures, etc.) have been described and analyzed.

Questions to self-checking

1. Please describe the classification for availability models of BASs.
2. Which are the main differences between common and separate maintenance?
3. Which are the main differences between unlimited and limited number of maintenance?
4. Which are the main differences between maintenance by reliability and security?
5. Which are the main steps of base modeling without maintenance MBAS1
6. Which are the main steps of modeling BAS with common unlimited maintenance MBAS2.1?

7. Which are the main steps of modeling BAS with common limited maintenance MBAS2.2?
8. Which are the main steps of modeling BAS with separate unlimited maintenance MBAS3.1?
9. Which are the main steps of modeling BAS with separate limited maintenance MBAS3.2?
10. Please describe the scaling of availability models for BASs.

References

1. K. S. Trivedi, D. S.fdc Kim, A. Roy and D. Medhi, Dependability and security models, – In 7th International Workshop on Design of Reliable Communication Networks, – Washington, DC, – pp. 11-20, – 2009. doi: 10.1109/DRCN.2009.5340029.
2. Q. Yu and R. J. Johnson, Smart grid communications equipment: EMI, safety, and environmental compliance testing considerations, – Bell Labs Technical Journal, – vol. 16, no. 3, – pp. 109-131, – Dec. 2011, doi: 10.1002/bltj.20525.
3. Kharchenko, V., Odarushchenko, O., Odarushchenko, V., Popov, P. Selecting mathematical software for dependability assessment of computer systems described by stiff markov chains. – CCIS, – vol. 1000, – pp. 146–162. – 2013/
4. Kharchenko V., Abdul-Hadi A.M., Boyarchuk A., Ponochovny Y. Web Systems Availability Assessment Considering Attacks on Service Configuration Vulnerabilities. – Advances in Intelligent Systems and Computing, – vol 286. – P.275-284 – 2014. doi: 10.1007/978-3-319-07013-1_26.
5. M. Grottke, H. Sun, R. Fricks and K. Trivedi, Ten Fallacies of Availability and Reliability Analysis, Service Availability. – Lecture Notes in Computer Science, – vol 5017, – pp. 187-206, – 2008, doi: 10.1007/978-3-540-68129-8_15.
6. Kharchenko V., Ponochovnyi Y., Abdulmunem A.S.M.Q., Andrashov A. Availability Models and Maintenance Strategies for Smart Building Automation Systems Considering Attacks on Component Vulnerabilities. – Advances in Intelligent Systems and Computing, Vol. 582, 2017, P. 186-195. DOI: 10.1007/978-3-319-59415-6_18.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ К РАЗДЕЛУ 37

BAS – Building automation system

I&CS – Information and control systems

SDE – System of differential equations

АННОТАЦИЯ

В разделе представлены марковские модели готовности информационно-управляющих систем умных домов, учитывающие различные варианты процессов восстановления и обслуживания, а также параметров проявления программных дефектов и атак на уязвимости, что позволяет повысить точность оценки и определить требования к коэффициенту готовности и средствам киберзащиты. Рассмотрены реализации аналитических моделей готовности информационно-управляющих систем умных домов с учетом отказов и атак на компоненты их архитектуры (MBAS1), с учетом проведения неограниченного количества процедур общего и раздельного обслуживания (MBAS2.1, MBAS3.1) и с учетом проведения ограниченного количества процедур общего и раздельного обслуживания (MBAS2.2, MBAS3.2) по надежности и безопасности.

У розділі представлені марковські моделі готовності інформаційно-керуючих систем розумних будинків шляхом врахування різних варіантів процесів відновлення і обслуговування, а також параметрів прояву програмних дефектів і атак на вразливості, що дозволяє підвищити точність оцінювання та визначити виконання вимог до коефіцієнту готовності та засобів кіберзахисту. Розглянуті реалізації аналітичних моделей готовності інформаційно-керуючих систем розумних будинків з урахуванням відмов і атак на компоненти їх архітектури (MBAS1), з урахуванням проведення необмеженої кількості процедур загального і роздільного обслуговування (MBAS2.1, MBAS3.1) і з урахуванням проведення обмеженої кількості процедур загального і роздільного обслуговування (MBAS2.2, MBAS3.2) по надійності і безпеці.

Building automation systems Markov models are discussed in the section. Markov models for availability of information and control systems of smart buildings have been improved by taking into account different variants of recovery and maintenance processes, as well as parameters of manifestation of software defects and vulnerability attacks, which allows to increase the accuracy of evaluation and to determine the fulfillment of the requirements for the availability factor and means of cyber security. Analytical models for the availability of information and control systems of smart homes, taking into account failures and attacks on their architecture components (MBAS1), have been developed considering the unlimited number of common and separate maintenance procedures (MBAS2.1, MBAS3.1) and the limited number of common and separate maintenance (MBAS2.2, MBAS3.2) procedures for reliability and security are discussed.

V. Sklyar, V. Kharchenko, E. Babeshko, A. Kovalenko, O. Illiashenko, O. Rusin, A. Panarin,
S. Razgonov, D. Ostapiec, I. Zhukovyts'kyi, S. Stirenko, O. Tarasyuk, A. Gorbenko,
A. Romanovsky, O. Biloborodov, I. Skarha-Bandurova, E. Brezhniev, A. Stadnik, A. Orekhov,
T. Lutskiv, V. Mokhor, O. Bakalynskiy, A. Zhylin, V. Tsurkan, M. Q. Al-sudani,
Yu. Ponochovnyi

SECURE AND RESILIENT COMPUTING FOR INDUSTRY AND HUMAN DOMAINS.

Secure and resilient systems, networks and infrastructures

Multi-book, Volume 2

Editor Vyacheslav Kharchenko

National Aerospace University n. a. N. E. Zhukovsky
"Kharkiv Aviation Institute"
17 Chkalova street, Kharkiv, 61070, Ukraine
<http://www.khai.edu>