Security and Resilience
of Web and Cloud Systems
Practicum

A. Gorbenko, O. Tarasyuk
Edited by V.S. Kharchenko

Resilient Web and Cloud Architecting

Vulnerability Study of Web and Computer Systems

Time-Probabilistic Modeling of secure and resilient Cloud systems

Security and Resilience of Web and Cloud Systems. Practicum

Tempus 2013 SEREIN 2016

PRACTICUM

SECURITY AND RESILIENCE OF WEB AND CLOUD SYSTEMS

2017

Co-funded by the Tempus Programme of the European Union

Министерство образования и науки Украины
Национальный аэрокосмический университет
им. Н.Е. Жуковского «ХАИ»

О.М. Тарасюк, А.В. Горбенко

Безопасность и устойчивость Веб- и облачных систем

Security and Resilience of Web- and Cloud-Systems

Практикум

Под редакцией В.С. Харченко

Проект
SEREIN 543968-TEMPUS-1-2013-1-EE-TEMPUS-JPCR
Modernization of Postgraduate Studies on Security and Resilience for Human and Industry Related Domains

2017

УДК 004.052

Авторы: О.М. Тарасюк, А.В. Горбенко. **Безопасность и устойчивость Веб- и облачных систем. Практикум /** Под ред. Харченко В.С. – Министерство образования и науки Украины, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», 2017. – 40 с.

ISBN 978-966-96770-6-8

Изложены материалы практической части тренинг-курса «Безопасность и устойчивость Веб- и облачных систем» (Security and Resilience of Web- and Cloud-Systems), подготовленного в рамках проекта TEMPUS SEREIN «Modernization of Postgraduate Studies on Security and Resilience for Human and Industry Related Domains» (543968-TEMPUS-1-2013-1-EE-TEMPUS-JPCR).

Приведены описания практические работ, которые предназначены для ознакомления с технологиями и средствами анализа и обеспечения устойчивости и информационной безопасности Веб- и облачных систем, а также исследования уязвимостей и реализация механизмов защиты от них. Предоставляется учебная программа курса и описание лабораторных работ и тренингов.

Предназначено для инженеров, занимающихся созданием и обеспечением информационной безопасности веб-приложений и систем Cloud Computing, для веб- разработчиков и специалистов по оценке качества и безопасности веб- и облачных систем, для магистров и аспирантов университетов, обучающихся по направлениям информационной безопасности, компьютерных наук, компьютерной и программной инженерии, а также для преподавателей соответствующих курсов.

Библ. – 17 наименований, рисунков – 4, таблиц – 2.

Рецензенты:

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

− Dr. Peter Popov, City University of London (London, UK).

Книга рекомендована к изданию ученым советом Национального аэрокосмического университета имени Н.Е. Жуковского «Харьковский авиационный институт» (протокол № 6 от 22 февраля 2017 г.).

ПРЕДИСЛОВИЕ

Данное пособие является частью учебно-методического обеспечения модуля «Безопасность и устойчивость Веб- и облачных систем» (Security and Resilience of Web- and Cloud-Systems), подготовленного в рамках проекта TEMPUS SEREIN «Modernization of Postgraduate Studies on Security and Resilience for Human and Industry Related Domains»1 (543968-TEMPUS-1-2013-1-EE-TEMPUS-JPCR) и служит дополнением к лекционному материалу, изложенному в [1, 2].

Устойчивость компьютерных систем определяется их способностью обнаруживать и парировать отказы и сбои, часть из которых не была предусмотрена на этапе проектирования этих систем. Особенно остро такая задача стоит при построении Веб- и облачных систем, которые характеризуются глобальной распределённостью компонентов, их гетерогенностью, высокой сложностью. При построении этих систем практически невозможно предусмотреть все возможные исключительные ситуации (ошибки, отказы и сбои) и реализовать методы их парирования.

Вторым немаловажным аспектом создания и эксплуатации Веб- и облачных систем является их потенциальная уязвимость к кибер-атакам и информационным вторжениям, целью которых является нарушение доступности предоставляемых услуг, целостности или же конфиденциальности данных.

В связи с этим актуальным является исследование механизмов обнаружения исключительных ситуаций, их диагностирование, а также выявление и анализ уязвимостей программного обеспечения Веб- и облачных систем.

В пособии приводится описание лабораторных работ и семинаров, методические рекомендации по самостоятельному изучению материала курса, в приложении дана учебная программа курса.

Практическая часть курса включает лабораторные работы и семинарские занятия, посвященные анализу, разработке и исследованию:

– механизма исключительных ситуаций Веб- и облачных систем (лабораторная работа №1);

– уязвимостей программного обеспечения Веб- и облачных систем (лабораторная работа №2);

– механизмов использования уязвимостей программного обеспечения для реализации информационных атак (лабораторная работа №3);

Каждая из лабораторных работ включает: цель, учебные, практические и исследовательские задачи; программу подготовки; краткий теоретический материал; программу проведения разработок и исследований; требования к содержанию отчету; варианты заданий; контрольные вопросы.

Семинарские занятия проводятся по перспективным направлениям развития современных Веб- и облачных систем. Описание семинарских занятий включает тему, цель, методические указания по подготовке и проведению.

Пособие предназначено для для инженеров, занимающихся созданием и обеспечением информационной безопасности веб-приложений и систем Cloud Computing, для веб- разработчиков и специалистов по оценке качества и безопасности веб- и облачных систем, для магистров и аспирантов университетов, обучающихся по направлениям информационной безопасности, компьютерных наук, компьютерной и программной инженерии, а также для преподавателей соответствующих курсов.

Авторы выражают благодарность рецензентам, коллегам по проекту, кафедрам университетов за ценную информацию, методическую помощь и конструктивные предложения, которые высказывались в процессе обсуждения практической части данного курса.

[1] *Этот проект финансируется при поддержке Европейской комиссии. Эта публикация (сообщение) отражает мнения только авторов, и Комиссия не может нести ответственность за любое использование содержащейся в нем информации.*

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.

1 ЛАБОРАТОРНЫЕ РАБОТЫ

1.1 Исследование механизмов обнаружения исключительных ситуаций в веб- и облачных системах

Цель и задачи работы

Целью работы является экспериментальное исследование механизма обнаружения исключительных ситуаций в веб- и облачных системах, построенных на основе различных платформ и технологий веб-сервисов.

Учебные задачи:

– изучение механизма обработки исключительных ситуаций (exception handling) в технологиях веб-сервисов;

– изучение технологий засева ошибок (fault injection) в распределенные веб- и облачные системы.

Практические задачи:

– разработка тестовых веб-сервисов;

– засев/симуляция различных отказов и сбоев, характерных для веб- и облачных систем;

–

Исследовательская задача:

– исследование способности различных технологий веб-сервисов распознавать разные виды исключительных ситуаций

– исследование механизмов распространение исключительных ситуаций (exception propagation) через стек технологий и протоколов веб-сервисов;

– исследование скорости и точности диагностирования исключительных ситуаций в веб-системах.

Подготовка к лабораторной работе

При подготовке к лабораторной работе необходимо:

– уяснить цели и задачи исследований;

– изучить теоретический материал, приведенный в описании;

– изучить и проанализировать особенности обработки исключений в веб- и облачных системах;

– ознакомиться с порядком работ и уточнить программу исследований.

Теоретический материал

Концепция сервис-ориентированной архитектуры (Service-Oriented Architecture, SOA) была предложена для решения проблем обеспечения эффективного, надежного и безопасного взаимодействия сложных распределенных систем. SOA предполагает, что современные веб-системы должны строиться на основе слабо связанных программных модулей (служб), которые имеют общедоступные интерфейсы (описанные с помощью языка описания веб-служб WSDL – Web Service Description Language) и специальный механизм взаимодействия (с помощью протокола SOAP – Simple Object Access Protocol). Описания этих модулей могут быть обнаружены другими программными системами в специальных реестрах UDDI (Universal Description, Discovery, and Integration), после чего модули могут быть вызваны с помощью SOAP-сообщений на основе языка XML, передаваемых посредством стандартных интернет-протоколов, таких как HTTP, SMTP, FTP и др.

Достижение высокой надежности и устойчивости сервис-ориентированной архитектуры имеет решающее значение для ряда критических и бизнес-критических областей, таких как телекоммуникационные системы, системы электронной коммерции, банкинга, интернета вещей и др. Знание точных причин и источников возникновения исключений, возникающих во время работы веб-службы позволяет разработчикам применять наиболее подходящие методы обеспечения отказоустойчивости[1] и восстановления после ошибок [2].

Например, в работе [3] обсуждаются два механизма отказоустойчивости, активно применяемые в сервис-ориентированных веб-системах: 1) backward error recovery (возврат системы к предыдущему безошибочному состоянию) и 2) forward error recovery (переход системы в одно их последующих безошибочных состояний). Последний, обычно, зависит от логики приложения и использует механизмы обработки исключений. Поскольку возврат к предыдущему безошибочному состоянию не

всегда применимо к веб-службам из-за того, что большинство из них не хранят своего состояния (т.е. являются stateless), обработка исключений становится наиболее популярным методом обеспечения отказоустойчивости и восстановления после ошибок веб-служб.

В лабораторной работе предлагается выполнить экспериментальный анализ механизмов распространения исключений в веб-системах и проанализировать различия в обработке ошибок и задержках распространения при использовании разных технологий реализации веб-сервисов (например, IBM WebSphere SDK, Apach Axis, Microsoft .Net и др.). Для выполнения такого анализа предлагается использовать методику засева ошибок/дефектов (fault injection).

Засев дефектов является хорошо зарекомендовавшим себя методом оценки надежности и отказоустойчивости вычислительных систем. Несмотря на то, что применение данного метода для исследования распределённых систем в целом является достаточно хорошо изученным (см., например [4, 5]), существует определённый дефицит работ в области его применения к веб-сервисам и сервис-ориентированным системам. В работах [6, 7] представлен практический подход к анализу и оценке надежности Web-сервисов. В частности, авторы [7] описывают методику тестирования робастности (т.е. устойчивости к некорректным входным параметрам вызова), которые были использованы для оценки надежности веб-сервисов.

В работах [8, 9] представлены специализированные онтологии, используемые для систематической генерации тестовых примеров для внедрения/засева дефектов, атак и временных задержек, а также описаны методы обнаружения сбоев в SOA, возникающие при потере или искажении сетевых пакетов. Тем не менее, описанные выше работы не учитывают процесс распространения сообщений о возникновении исключительных ситуфций (exception propagation), а также не исследуют влияние используемой платформы или технологии создания веб-услуг на быстроту оповещения о возникшей исключительной ситуации и точность диагностирования.

Ошибки, дефекты, отказы и сбои веб-служб. Общепринятая концепция надежности компьютерных систем [10] определяет следующие угрозы их функционированию: ошибки (errors), дефекты (faults, bugs) и отказы/сбои (failures).

Дефект в компьютерной системе, например, дефект в программном обеспечении, допущенный вследствие ошибки программиста, может привести к отказу или сбою системы. Отказ/сбой системы диагностируется, когда ошибочный результат распространяется за пределы интерфейса системы. Дефект – это гипотетическая причина, приведшая к получению ошибочного результата. Обычно различают три группы дефектов компьютерных систем [10]: дефекты проектирования и разработки, физические неисправности и ошибки взаимодействия.

Основными этапами взаимодействия веб-сервисов [11] являются (см. рис. 1):

1) привязка (binding) веб-службы;

2) вызов (invocation) веб-службы;

3) обмен сообщениями SOAP между клиентом и веб-сдужбой;

4) обработка веб-сдужбой запросов клиентов и подготовка результата.

Предварительно был составлен список из 18 различных ошибок/дефектов, специфичных для сервис-ориентированных систем, возникающих на всех указанных этапах (см. Таблицу 1), засев и анализ проявления которых предлагается выполнить в процессе выполнения лабораторной работы. Эти ошибки могут быть условно разделены на три категории: 1) сбои сети и отказы удаленной веб-службы; 2) внутренние ошибки и сбои веб-службы и 3) ошибки привязки на стороне клиента. Указанные ошибки/дефекты являются общими (не специфичными для конкретного приложения) и могут проявляться в любой веб-службе во время её работы.

По своему усмотрению слушатели могут дополнить представленный список ошибок, дефектов и сбоев.

Сообщения о возникших исключительных ситуаций (exception messages), генерируемые программной системой, являются проявлением симптомов возникновения/активации ошибки, сбоя или дефекта [15].

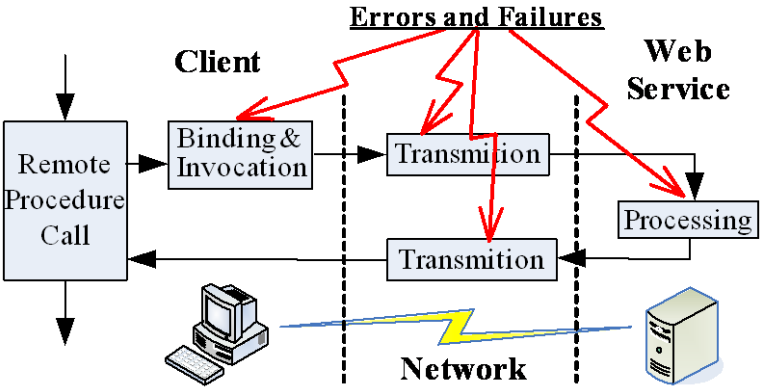

Рис. 1. Потенциальные места возникновения ошибок, отказов и сбоев сервис-ориентированных веб-систем

Таблица 1. Перечень ошибок, дефектов и сбоев для их засева в тестовую веб-службу

№	Type of error/failure	Error/failure domain
1.	Network connection break-off	Network and system failures
2.	Domain Name System is down	
3.	Loss of request/response packet	
4.	Remote host unavailable	
5.	Application Server is down	
6.	Suspension of WS during transaction	Service errors and failures
7.	System run-time error	
8.	Application run-time error	
9.	Error causing user-defined exception	
10.	Error in Target Name Space	Client-side binding errors
11.	Error in Web Service name	
12.	Error in service port name	
13.	Error in service operation’s name	
14.	Output parameter type mismatch	
15.	Input parameter type mismatch	
16.	Error in name of input parameter	
17.	Mismatching of number of input params	
18.	WS style mismatching (“Rpc” or “Doc”)	

К традиционным сетевым отказам и сбоям относятся недоступность службы DNS или же потери и искажения сетевых пакетов. Кроме того, безотказная работа веб-службы зависит от безотказного функционирования системного программного обеспечения, такого как веб-сервер, сервер приложений и система управления базами данных. В работе предлагается проанализировали сбои, возникающие при принудительном и неожиданном прекращении работы сервера приложений (WebSphere, Apache Tomcat и IIS).

Ошибки могут возникать и на стороне клиента при раннем связывании или вызове динамического интерфейса (Dynamic Invocation Interface). Например, «Ошибка в пространстве имен целей», «Ошибка в имени веб-службы» и т.д. gроисходят из-за изменений параметров вызова и/или несоответствий между WSDL описанием веб-службы и фактическим интерфейсом вызова. Наконец, сбои и отказы самих веб-служб могут быть связаны с программными и системными ошибками времени выполнения (run-time errors), которые генерируют пользовательские или системные исключения.

Ошибки времени выполнения, такие как «переполнение стека» или «нехватка памяти», приводят к исключениям на уровне системы в целом. Исключение, возникающее вследствие выполнения операции типа «Деление на ноль» также перехватывается и генерируется на системном уровне. Однако, такую исключительную ситуацию гораздо проще сымитировать нежели, например, системные исключения, связанные в переполнением стека.

Типичными примерами ошибок времени выполнения приложений являются «Несоответствие типа операнда» или «Выход индекса за пределы массива».

В таблицу 1 включены ошибки несоответствие типа операнда, зацикливание процесса, а также ошибка, вызывающая определяемое пользователем исключение (исключение определяется программистом во время разработки веб-службы).

Прикладные отказы (6, 7, 8) могут быть симулированы с помощью внедрения соответствующих ошибок в программный код сервиса.

Ошибки связывания (10-18) являются, по сути, набором тестов робастности и реализуются на стороне клиента с помощью передачи веб-сервису ошибочных значений параметров вызова.

Сетевые отказы и сбои могут быть сымитированы путем фильтрации межсетевым экраном сообщений DNS-сервера, принудительного завершения работы сервера приложений и закрытия сетевых подключений на стороне клиента и веб-службы.

Трассировка и документирование исключительных ситуаций. Для генерации, документирования и трассировки исключительных ситуаций предлагается воспользоваться стандартным инструментарием используемого языка разработки (Java или C#).

На рисунке 2 показан фрагмент Java кода, который следует за критическим разделом программы и служит для перехвата возможных исключительных ситуаций, а также отображения результата трассировки стека исключений в случае возникновения ошибок. Такой код должен быть реализован как на стороне клиента, так и на стороне веб-сервиса.

Например, результат трассировка стека исключений, соответствующего ошибке «Несовпадение типов операндов», обнаруженной веб-службой, приведен на рисунке 3.

Представленная цепочка распространения исключений имеет всего четыре вложенных вызова (начинающихся с предлога “at”). Для сравнения, трассировка стека исключений при возникновении аналогичной ошибки при взаимодействии с веб-сервисом, реализованным с помощью IBM WSDK, имеет 63 вложенных вызова.

```
...
catch (Exception e) {
  e.printStackTrace();
}
```

Рис. 2. Пример программного кода Java для перехвата и трассировки стека исключений

```
java.rmi.ServerException: JAXRPC.TIE.04:
Internal Server Error (JAXRPCTIE01: java.lang.
NumberFormatException: For input string:
"578ER")
   at com.sun.xml.rpc.client.dii.BasicCall.
invoke(BasicCall.java:497)
   at ai.c1.xai12.wstest.InvoceWS.invoce
(InvoceWS.java:125)
   at ai.c1.xai12.wstest.InvoceWS.
invoceByVector(InvoceWS.java:75)
at wstest.Main.main(Main.java:42)
```

Рис. 3. Пример трассировки стека исключений, соответствующего перехвату ошибки №8 («Несовпадение типов операндов») на стороне клиента, использующего реализацию JAX-RPC от Sun Microsystems

Программа разработок и исследований

В лабораторной работе необходимо исследовать механизм генерации и распространения исключительных ситуаций в технологиях веб-услуг.

В качестве технологий реализации веб-сервисов (конкретной реализации прикладного программного интерфейса для создания веб-служб, а также сервера приложений для их развертывания) предлагается использовать:

– IBM WSDK + WebSphere;
– Apach Axis + Tomcat;
– Apach Axis + Glassfish;
– Microsoft .Net + IIS.

По своему желанию слушатели могут расширить список исследуемых технологий.

В качестве интегрированной среды разработки могут быть использованы Eclipse IDE, Netbeans IDE (для Java веб-сервисов), а также Microsoft Visual Studio (для.Net веб-сервисов).

В качестве основы для создания тестового веб-сервиса может быть использован; например, класс «Калькулятор», реализующий методы для выполнения арифметических операций: сложить, вычесть, умножить, разделить и т.п. (см. рис. 4).

```
package wscalc;
public class WSCalc implements WSCalcSEI {
   public String getMul (int a, int b) {
      return new Integer(a * b).toString();
   }
   ...
}
```

Рис. 4. Пример реализации тестового класса для создания веб-сервиса

Этапы выполнения работы

Общий алгоритм выполнения лабораторной работы представлен на рис. 5.

Он включает в себя шесть последовательных операций:

1) разработка Java/C# класса, в качестве основы для создания тестового веб-сервиса;

2) диверсная реализация и развёртывание двух или более тестовых веб-сервисов с использованием разных технологий (IBM WSDK, Apach Axis, Microsoft .Net и др.);

3) анализ и спецификация потенциально-возможных ошибок и дефектов/отказов, характерных для распределенных веб- и облачных систем;

4) последовательный «засев» специфицированных ошибок и дефектов/отказов;

5) документирование и проведение сравнительного анализа сообщений об исключительных ситуациях, сгенерированных диверсными реализациями веб-служб; анализ точности диагностирования ошибок и сбоев;

6) трассировка распространения сообщений о возникновении исключительных ситуаций через стек технологий веб-сервисов и оценка быстроты обнаружения ошибок и сбоев.

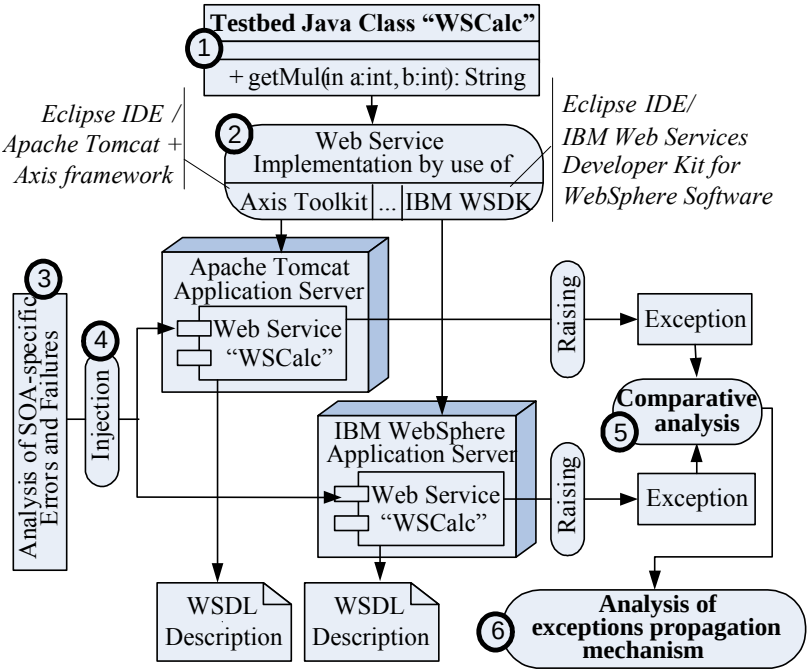

Рис. 5. Программа проведения исследований

Требования к содержанию отчета

Отчет должен включать:

– цели и программу проведения разработок и исследований;

– информацию с описанием исследуемых тестовых веб-сервисах и технологиях их реализации;

– особенности и применённые способы «засевы» ошибок и дефектов;

– отчет о результатах перехвата исключительных ситуаций по каждому из сервисов, представленный в виде таблицы, которая включает:

а) наименование и номер ошибки;

б) сообщение об исключительной ситуации (или её отсутствие) верхнего уровня;

в) количество вложенных вызовов исключений на основе трассировки стека исключений;

г) среднее время генерации сообщения верхнего уровня об исключительной ситуации.

– результаты анализа и сравнения точности и быстроты диагностирования исключительных ситуаций для разных реализаций тестового веб-сервиса;

В приложение к отчету включить результаты трассировки стека исключений.

Контрольные вопросы

В результате выполнения лабораторной работы предлагается ответить на следующие теоретические и практические вопросы:

1. В чем отличие между ошибкой, дефектом, отказом и сбоем? Как эти понятия связаны между собой?

2. Какие ошибки и дефекты являются наиболее характерными для веб-систем и веб-сервисов? По какой причине они возникают и каковы их последствия?

3. В чем заключается методика засева ошибок? Каким образом она реализуется практически для сервис-ориентированных систем?

4. Является ли промежуточное программное обеспечение (конкретная технология реализации) веб-сервисов устойчивым к ошибкам в параметрах вызова?

5. Что такое исключительная ситуация? Для чего используется этот механизм?

6. Какие конструкции используются для оповещения об исключительных ситуациях при разработке программного обеспечения?

7. Позволяет ли генерируемое сообщение об исключительной ситуации однозначно диагностировать причину его возникновения?

8. Позволяет ли трассировки стека исключений более точно идентифицировать первопричину (root case failure)?

9. Зависит ли длина цепочки распространения исключений и скорость диагностирования ошибки от конкретной технологии реализации и развертывания веб-сервисов?

10. Какие существуют отличия в реализации механизма исключительных ситуаций в разных технологиях веб-сервисов?

1.2 Исследование уязвимостей программного обеспечения веб- и облачных систем

Цель и задачи работы

Целью работы является изучение уязвимостей программного обеспечение, а также получение практических навыков работы с базами данных уязвимостей.

Учебные задачи:

– изучение причин возникновения и последствий уязвимостей;

– анализ особенностей и характеристик уязвимостей.

Практические задачи:

– получение навыков практической работы с базами данных уязвимостей (CVE, NVD, OSVDB, …);

Исследовательские задачи:

– анализ жизненного цикла уязвимостей;

– оценка риска программного обеспечения.

Подготовка к лабораторной работе

При подготовке к лабораторной работе необходимо:

– уяснить цели и задачи исследований;

– изучить теоретический материал, приведенный в описании, а также лекционной части курса;

– ознакомиться с порядком работ и уточнить программу исследований в соответствии с заданным вариантом.

Теоретический материал

Информацию об уязвимостях можно получить из общедоступных источников – специализированных баз данных уязвимостей (БДУ), которые, как правило, предоставляют информацию об уязвимостях программного обеспечения в XML формате:

- Open Source Vulnerability Database (www.osvdb.org) – предоставляет информацию как в XML-формате, так и в виде SQL-дампов;
- Common Vulnerabilities and Exposures (www.cve.mitre.org) – является поставщиком единого общего словаря уязвимостей CVE; информация поставляется в виде XML-файлов;

- National Vulnerability Database (www.nvd.nist.gov) – является наиболее подробной БДУ; включает метрики для оценки степени критичности уязвимостей, а также предоставляет XML-словарь для идентификации программного обеспечения и системной конфигурации, содержащих уязвимость.

Программа разработок и исследований

1. Загрузить базы данных уязвимостей NVD и CVE.
2. Выполнить поиск уязвимостей за последний год в базе данных NVD для программного обеспечения Web-сервера, конфигурация которого определяется вариантом задания.
3. Проанализировать характеристики отобранных уязвимостей.
4. Используя идентификаторы CVE уязвимостей, отобранных из базы данных NVD, сделать выборку информации об этих уязвимостях по базе данных CVE.
5. Выполнить анализ дат публикации информации об эквивалентных уязвимостях в CVE и NVD базах данных.
6. Построить кумулятивные графики обнаружения уязвимостей в программном обеспечении с учетом дат их публикации в базах данных CVE и NVD.
7. Выполнить поиск информации о подтверждении уязвимости и её исправлении (выходе патча) на web-ресурсах разработчика программного обеспечения.
8. Построить графики текущего количества уязвимостей в программном обеспечении с учетом дат их обнаружения (публикации в базе CVE) и исправления.
9. Выполнить интегральную оценку уязвимости Web-системы на основе методики, предложенной в [16-17].

Варианты заданий

Варианты заданий представлены в табл. 1.

Требования к содержанию отчета

Отчет должен включать:

– цели и программу проведения исследований;

– отчеты по обнаруженным уязвимостям программных продуктов, представленные в виде таблиц и графиков (кумулятивное число уязвимостей; текущее кол-во уязвимостей);

– интегральную оценку уязвимости конфигурации Web-системы, представленная в виде графика;

– сравнение количества уязвимостей по группам CWE;

– результаты анализа и выводы по работе.

Таблица 1. Варианты заданий

№ варианта	Конфигурация Web-системы			
	Операционная система	Web-сервер	Сервер приложений	Сервер базы данных
1	Novel Linux	Apache	JBoss	MySQL
2	Oracle Solaris	iPlanet	Oracle WebLogic	Oracle DB
3	Free BSD	nginx	GlassFish	Postgre SQL
4	RedHat Linux	lighttpd	Resin	MySQL
5	IBM AIX	IBM HTTP Server	IBM Web Sphere	IBM DB2
6	Aple MacOS Server	Apache	WebObjects	MySQL
7	Windows Server	IIS	IIS	MS SQL

Контрольные вопросы

1. Что такое уязвимость программного обеспечения?
2. Какие базы данных предоставляют информацию об уязвимостях? Какая информация храниться в этих базах?
3. Опишите жизненный цикл уязвимости.
4. По каким критериям оценивается степень критичности уязвимостей в базе данных NVD?
5. Что такое CVE, CPE, CVSS, CWE?
6. Назовите наиболее распространенные виды уязвимостей?

7. Какие существуют способы защиты от уязвимостей?

8. Что такое эксплойт? Каким образом наличие эксплойта влияет на степень критичности уязвимости?

9. На чем основан механизм функционирования сканеров уязвимостей?

10. Каким образом соотносятся понятия «уязвимость», «вирус», «сетевая атака»?

2 СЕМИНАРЫ

2.1 Общие методические рекомендации по подготовке к семинарам

Тематика семинаров: современные технологии моделирования, оценки, разработки, верификации, экспертизы и применения гарантоспособных (надежных, живучих и безопасных) аппаратных, программных и сетевых компонент, компьютерных систем и сетей для критических и бизнес-критических приложений.

Цель семинара: приобретение знаний и практических навыков по подготовке и презентации выполненного проекта (реферата, аналитического обзора, разработки) по вопросам гарантоспособности компьютерных систем и сетей для критических и бизнес-критических приложений.

Подготовка к семинару

1. Получение (определение) темы работы (реферата аналитического обзора, разработки) и уточнение задач.

Темы работ могут формироваться студентами самостоятельно и согласовываться с руководителями исходя из следующей цепочки ключевых слов:

– принципы, методы, средства, технологии… (principles, methods, techniques, tools, technologies);

– моделирование, оценка, обеспечение (повышение), проектирование, разработка, реализация, испытания, обслуживание, верификация и валидация, экспертиза, аудит, сертификация, …(*modeling, simulation, ensuring; guaranteeing, designing, development, testing, maintenance, verification and validation, expertise, audit, certification, inspection*…);

– гарантоспособность (надежность), безотказность, ремонтопригодность, долговечность, сохраняемость, готовность, отказоустойчивость, живучесть, функциональная и информационная безопасность, достоверность и др. (*dependability, reliability, reparability, durability, preservability, availability, fault-*

tolerance, survivability, safety, security, high confidence, trustworthiness…);

– программные средства (системные, прикладные, технологические), аппаратные (технические) средства, ранее разработанные компоненты, СБИС, микропроцессоры, ПЛИС, компьютеры, компьютерные системы, компьютерные сети, веб-сервисы, сервис-ориентированные архитектуры, распределенные вычисления, метакомпьютинг, грид-системы, технологии Cloud Computing (*software, hardware, OTS (Off-The-Shelf) and COTS (Commercial-Off-The-Shelf)-components, VLSI, microprocessors, PLD-devices (CPLD, FPGA, ASIC), computers, computing systems, computer-based systems, networks, web-services, SOA, distributed systems (computing), metacomputing, GRID-systems, Cloud Computing Technologies: IaaS, PaaS, SaaS*),…;

– коммерческие приложения; бизнес-критические приложения: банковские технологии, е-коммерция, е-наука, е-обучение…; критические приложения: авиационные, ракетно-космические, военные (оборонные), транспортные, телекоммуникационные, нефтегазовые, энергетические системы, АЭС (*commercial applications; business-critical applications: banking, e-commerce, e-science, e-learning, …; critical (safety-critical and mission-critical) applications: aviation, rocket-space, military (defense), transport, telecommunication, oil-gas, energy systems, NPP,…*

Примеры тем реферата:

– технологии создания SaaS-приложений и примеры их практического использования;

– методы обеспечения отказоустойчивости компьютерных сетей и сервис-ориентированных систем;

– анализ видов, причин и последствий отказов SaaS-приложений.

2. Разработка плана работ и распределение ответственности между участниками целевой группы. План работ может быть представлен в виде диаграммы Ганта, включающей основные мероприятия, сроки и распределение ответственности между участниками целевой группы.

Целевая группа состоит из 3 человек. Примерный ресурс времени на подготовку 9х3=27 часов (+ 15 мин презентации). Распределение ответственности определяют участники группы.

Пример распределения ответственности: менеджер, осуществляющий планирование и координацию работ, а также представляющий идеологию работ на семинаре (1 часть общего доклада – постановку задачи), *разработчик* теоретической части – системный аналитик (2 часть доклада), разработчик прикладной части – программист, электронщик (3 часть доклада + дизайн).

3. Поиск информации по теме работы (библиотека, Интернет, Информационно-ресурсный центр критического компьютинга и его филиалы - www.mastac.irc.com) и ее предварительный анализ.

Англоязычные термины даны для удобства поиска информации в Интернет. Возможно представление реферата и презентации на английском языке, что повысит оценку за семинар. Поиск информации осуществляется по ключевым словам, приведенным в пункте 1.

Методические указания и список рекомендуемой литературы к рефератам выдаются индивидуально (по группам).

4. Разработка плана отчета и презентации проекта. План отчета (и презентации) включает подготовку следующих разделов:

– введение (актуальность, вызовы практики, краткий анализ состояния вопроса – литературы, цель и основные задачи реферата, структура и характеристика содержания, план работ и распределение ответственности);

– систематизированное изложение основных частей реферата (классификационные схемы, характеристика моделей, методов, средств, технологий по группам, выбор показателей и критериев для оценки, сравнительный анализ);

– выводы (констатация достижения поставленной цели, основные теоретические и практические результаты, их значимость, направления дальнейших работ);

– список литературы;

– приложения.

5. Написание отчета. Отчет имеет объем 15-20 страниц формата А4 (шрифт 14, 1,5 инт., поля 2 см), включая титульный лист, содержание, основной текст, литература, приложения.

Рефераты, подготовленные путем простой компиляции Интернет-материалов, без тщательного структурирования, с некорректной терминологией и без выводов не рассматриваются.

Обязательным приложением к реферату является план работ и распределение ответственности (диаграмма Ганта), презентационные слайды и электронный вариант всех материалов.

6. Подготовка презентации. Презентация разрабатывается в PowerPoint и соответствует плану реферата (10-15 слайдов) исходя из времени на презентацию – 10 мин.

Презентация должна включать следующие слайды:

– титульный слайд (с указанием ВУЗа, кафедры, дисциплины, темы доклада, автора, даты презентации);

– содержание (структура) доклада;

– актуальность рассматриваемых вопросов, цель и задачи доклада исходя из этого анализа;

– слайды с раскрытием содержания поставленных задач;

– выводы по докладу;

– список использованных источников.

Каждый из слайдов должен содержать колонтитул с указанием темы и авторов доклада.

Содержание слайдов не должно представлять собой части текста из отчета, а включать ключевые слова, рисунки, формулы.

Подача информации может быть динамической.

Защита работы

Защита работы осуществляется на семинаре, занимает 15 мин и включает собственно доклад с презентацией (10 мин) и обсуждение (5 мин).

Оценка работы

Оценка выполненной работы учитывает качество текста отчета (форма и содержание), презентации (содержание и дизайн), собственно доклад (структура, содержание и выводы), полноту, глубину и правильность ответов на вопросы.

Оценка за выполненную работу выставляется каждому студенту из группы авторов доклада индивидуально в соответствии с результатами и распределением ответственности.

2.2 Особенности подготовки к семинарам по дисциплине «Безопасность и устойчивость Веб- и облачных систем»

При изучении данной дисциплины семинарские занятия проводятся по следующим направлениям:

– современные технологии создания распределенных информационных систем. Парадигмы SOA, Saas, PaaS, IaaS, Cloud Computing;

– методы моделирования работы компьютерных сетей и распределенных сервис-ориентированных систем (марковские графы, сети Петри, системы массового обслуживания) и их практическое применение;

– исследование гарантоспособности компьютерных сетей и сервис-ориентированных систем. Анализ видов, причин и последствий отказов;

– методы обеспечения отказоустойчивости программных средств. Технологии Backward Error Recovery, Forward Error Recovery, Compensation;

– технологии обеспечения информационной безопасности компьютерных сетей и Web-сервисов. Технологии обнаружения и парирования сетевых атак.

3 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО САМОСТОЯТЕЛЬНОЙ РАБОТЕ

3.1 Пояснения к учебной программе

Самостоятельную работу над дисциплиной «Безопасность и устойчивость Веб- и облачных систем» следует начинать с изучения учебной программы, которая приведена в Приложении. Эта программа включает следующие элементы.

Эта программа включает следующие элементы.

Объект изучения – безопасные и устойчивые веб- и облачные системы.

Предмет изучения – принципы, методы, технологии и средства разработки и развертывания безопасных и устойчивых сервис-ориентированных веб- и облачных систем.

Требования к исходным знаниям и навыкам, которые необходимо иметь перед началом изучения:

– принципы и методы системного анализа;

– теория вероятностей и математической статистики;

– теория надежности;

– базовые знания в области современных компьютерных сетей и сетевых информационных технологий;

– знания и практические навыки разработки программного обеспечения.

Целью дисциплины является изучение современных методов и подходов к проектированию, оценке и реализации безопасности и устойчивости информационных систем, компьютерных сетей и Web-сервисов для бизнес-критического применения.

В результате ее изучения обучаемые должны научиться:

– анализировать и синтезировать информацию;

– задавать и отвечать на поставленные вопросы, мыслить креативно и критически;

– предпринимать исследовательские действия и оценивать получаемые результаты с использованием качественных и количественных показателей;

– формулировать практические решения проблем, эффективно использовать время и доступные ресурсы для достижения целей дисциплины;

– демонстрировать гибкость, адаптируемость, самомотивацию и инициативу, умение выразить свое мнение.

Структура и содержание модулей. Дисциплина включает четыре модуля:

– модуль 1 – Устойчивые интернет и облачные системы, в котором рассмотрена парадигма и принципы создания устойчивых (resilient) веб- и облачных систем;

– модуль 2 – Уязвимость веб- и облачных систем. Модуль включает описание методики анализа уязвимости программного обеспечения современных веб- и облачных систем;

– модуль 3 – Создание безопасных и устойчивых веб- и облачных систем; в данном модуле изучаются основные принципы и подходы к обеспечению устойчивости указанных систем к информационным вторжениям, а также минимизации риска эксплуатации имеющихся уязвимостей;

– модуль 4 – Стандарты и средства для обеспечения безопасности веб- и облачных систем, включает описание имеющихся утилит и стандартов, соблюдение которых позволит обеспечить и подтвердить безопасность веб- и облачных систем.

По каждому из модулей предусмотрены лабораторные занятия и семинары и дан список рекомендуемой литературы.

Отчетность по дисциплине включает отчеты по каждой из лабораторных работ и семинарам, а также экзамен, который включает типовые вопросы и задачи.

3.2 Подготовка к занятиям и экзамену

Подготовка к занятиям по дисциплине детально описана в разделах 2 и 3.

При подготовке к лабораторным работам следует обратить внимание на уяснение целей и задач (учебных или теоретических, практических и исследовательских) и знаний, которые нужны для их выполнения. При выполнении разработок и исследований необходимо строго руководствоваться описанием и попытаться найти ответы на вопросы, приведенные в конце каждой работы. Особое внимание следует уделить формулировке выводов по результатам исследований при оформлении отчета.

При подготовке к семинарам важно правильно спланировать свою работу в составе группы проекта, организовать отбор и анализ необходимой литературы, подготовку качественной презентации и подготовку к ответам на возможные вопросы.

Своевременная и основательная подготовка к лабораторным работам – гарантия успешной сдачи экзамена.

Важным элементом изучения дисциплины является формирование полной и корректной системы используемых терминов и ключевых понятий (энергоэффективность, производительность, пропускная способность и т.д.). Целесообразно по каждому из модулей сформировать их множество и установить логические святи между соответствующими понятиями.

Кроме того, следует обратить внимание на вопросы, вынесенные на самостоятельное изучение, которые приводятся в программе и уточняются преподавателем.

ЛИТЕРАТУРА

1. Chan, Pat. P.W., Lyu, M.R., Malek, M. 2006. Making Services Fault Tolerant. *In D. Penkler, M. Reitenspiess, and F. Tam (Eds.): Service Availability, International Service Availability Symposium, LNCS 4328*, Berlin, Heidelberg: Springer-Verlag, 43–61.

2. Managing Exceptions in Web Services Environments. 2003. An AmberPoint Whitepaper (http://www.amberpoint.com).

3. Tartanoglu, F., Issarny, V., Romanovsky, A., Levy, N. 2003. Coordinated Forward Error Recovery for Composite Web Services. In Proceedings of the 22nd Symposium on Reliable Distributed Systems (SRDS), Florence, Italy, 167-176.

4. Marsden, E., Fabre, J.-C., Arlat, J. 2002. Dependability of CORBA Systems: Service Characterization by Fault Injection. In Proceedings of the Symposium on Reliable Distributed Systems, Osaka, Japan.

5. Brambilla, M., Tziviskou, C. 2005. Fundamentals of Exception Handling Within Workflow-Based Web Applications. Journal of Web Engineering (JWE), Vol. 4, Issue 1, 38-56.

6. Vieira, M., Laranjeiro, N., Madeira, H. 2007. Assessing Robustness of Web-services Infrastructures. In Proceedings of the 2007 Int. Conf. On Dependable Systems and Networks (DSN'2007), 131–136.

7. Duraes, J., Vieira, M., Madeira, H. 2004. Dependability Benchmarking of Web-Servers. In M. Heisel et al. (Eds.): SAFECOMP 2004, LNCS 3219, 297–310.

8. Looker, N., Gwynne, B., Xu, J., Munro, M. 2005. An Ontology-Based Approach for Determining the Dependability of Service-Oriented Architectures. In Proceedings of the 10th IEEE International Workshop on Object-oriented Real-time Dependable Systems, USA.

9. Looker, N., Munro, M., Xu, J. 2005. Simulating Errors in Web Services. International Journal of Simulation Systems, Science & Technology, vol. 5.

10. Avizienis, A., Laprie, J.-C., Randell, B., Landwehr, C. 2004. Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing, Vol. 1, No. 1, 11–33.

11.W3C, Web Services Architecture. 2004. http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

12.Gorbenko, A., Mikhaylichenko, A., Kharchenko, V., Romanovsky, A. 2007. Experimenting With Exception Handling Mechanisms Of Web Services Implemented Using Different Development Kits. Technical report CS-TR 1010: http://www.cs.ncl.ac.uk/research/pubs/trs/papers/1010.pdf, Newcastle University

13.Looker, N., Munro, M., Xu, J. 2004. Testing Web Services. In Proceedings of the 16th IFIP International Conference on Testing of Communicating Systems, Oxford.

14.Gorbenko, A., Kharchenko, V., Furmanov, A., Tarasyuk, O. 2006. F(I)MEA-Technique of Web Services Analysis and Dependability Ensuring. In M. Butler et al. (Eds.): Rigorous Development of Complex Fault-Tolerant Systems (LNCS 4157), Berlin, Heidelberg: Springer-Verlag, 153–167.

15.Cristian, F. 1995. Exception Handling and Tolerance of Software Faults. In *Software Fault Tolerance*, M. Lyu, ed., 81-107.

16.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.

17.Gorbenko, A., Romanovsky, A., Tarasyuk O. and Biloborodov O. Study of Vulnerabilities of Enterprise Operating Systems, Proc. Int. Symp. on Software Reliability Engineering (ISSRE'2017), Toulouse (France), October 23-26, 2017

АНОТАЦІЯ ТА ЗМІСТ

УДК 004.052

Автори: О.М. Тарасюк, А.В. Горбенко. **Безпека та стійкість Веб- та хмарних систем. Практикум /** Під ред. Харченка В.С. – Міністерство освіти та науки України, Національний аерокосмічний університет ім. М.Є. Жуковського «ХАІ», 2017. – 40 с.

ISBN 978-966-96770-6-8

У посібнику викладені матеріали практичної частини тренінг модулю «Безпека та стійкість Веб- та хмарних систем» (Security and Resilience of Web- and Cloud-Systems), підготовленого для аспірантів в рамках проекту TEMPUS SEREIN «Modernization of Postgraduate Studies on Security and Resilience for Human and Industry Related Domains» (543968-TEMPUS-1-2013-1-EE-TEMPUS-JPCR).

Представлено опис практичних робіт, які призначені для ознайомлення з технологіями та засобами аналізу та забезпечення інформаційної безпеки Веб- та хмарних систем, а також дослідження вразливостей та реалізації механізмів захисту від них. Представлена навчальна програма курсу та опис лабораторних робіт й тренінгів.

Книга призначена для інженерів, що займаються створенням та забезпеченням інформаційної безпеки веб-додатків та систем Cloud Computing, для веб- розробників та спеціалістів з оцінки якості та безпеки веб- и хмарних систем, для магістрів та аспірантів університетів, що навчаються за напрямками інформаційної безпеки, комп'ютерних наук, комп'ютерній і програмній інженерії, а також буде корисна для викладачів відповідних навчальних курсів.

Бібл. – 17 найменувань, рисунків – 4, таблиць – 2.

ЗМІСТ

ABSTRACT AND CONTENT

UDC 004.052

Authors: Tarasyuk O., Gorbenko A. **Security and Resilience of Web- and Cloud-Systems /** Kharchenko V. (edit.). – Department of Education and Science of Ukraine, National Aerospace University named after N. Zhukovsky "KhAI", 2017. – 40 p.

ISBN 978-966-96770-6-8

Practical materials of study course the "Security and Resilience of Web- and Cloud-Systems" are expounded in this training textbook prepared for PhD-students within the framework of project TEMPUS SEREIN «Modernization of Postgraduate Studies on Security and Resilience for Human and Industry Related Domains» (543968-TEMPUS-1-2013-1-EE-TEMPUS-JPCR).

Course curriculum, description of laboratory works, practical trainings and methodical recommendations for self-sufficient study are given.

The book is intended for university master and PhD students learning cybersecurity, computer sciences, computer and software engineering as well as for teachers lecturing respective courses.

Ref. – 17 items, figures – 4, tables – 2.

CONTENT

ПРИЛОЖЕНИЕ. УЧЕБНАЯ ПРОГРАММА

TEACHING PROGRAM

TITLE OF THE MODULE	Code
Security and Resilience of Web- and Cloud-Systems	**TM**

Teacher(s)	Department
Coordinating: Dr. Olga Tarasyuk **Others:** Prof. Anatoliy Gorbenko	Computer Systems and Networks

Study cycle	Level of the module	Type of the module
Engineer	A	Full-time tuition. Compulsory

Form of delivery	Duration	Language(s)
Full-time tuition	One semester	English

Prerequisites	
Prerequisites: Computer Systems and System Analysis; Computer Networks; Web programming and design	**Co-requisites (if necessary):** Foundations of Dependability and Security; System and Network Security and Resilience

Credits of the module	Total student workload	Contact hours	Individual work hours
4	108	36	72

Aim of the module (course unit): competences foreseen by the study programme
The aim of module is to create a knowledge base for multidisciplinary research on web- and cloud-systems resilience and security. It includes overview of a concept of computer system resilience, methods and approaches to analyse and exploit vulnerability of web- and cloud-systems, and exception handling techniques.

Learning outcomes of module (course unit)	Teaching/learning methods	Assessment methods
At the end of course, the successful student will be able to: 1. Understand a concept of resilient web- and cloud-systems and implement it in practice.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
2. Analyse vulnerability of web- and cloud-systems and use ethical hacking skills to investigate them.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
3. Implement intrusion-tolerance and intrusion avoidance techniques to enhance security of web- and cloud-systems.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire

Themes	Contact work hours							Time and tasks for individual work	
	Lectures	Consultations	Seminars	Practiacl work	Laboratory work	Placements	**Total contact work**	**Individual work**	**Tasks**
1. RESILIENT INTERNET AND CLOUD COMPUTING SYSTEMS	6				4		**10**	**18**	1.5. Web- and Cloud Technologies 1.6. Ubiquitous Computing Systems 1.6. Antifragile Computing Systems
1.1. Dependability Retrospective of Web and Cloud systems									
1.2. Resilience of Ubiquitous Computing Systems									
1.3. The Threat of Uncertainty									

1.4. Resilience Principles and Resilient System Architectures									
2. VULNERABILITY OF WEB- AND CLOUD COMPUTER SYSTEMS	4				4		**8**	**18**	2.5.Vulnerability Discovery 2.5. Vulnerability Types
2.1. Security and Vulnerability of Multilevel Computing Architectures									
2.2. Software Vulnerability Lifecycle									
2.3. Vulnerability Databases and Datasets									
2.4. Vulnerability Analysis and Vulnerability Metrics									
3. BUILDING SECURE AND RESILIENT WEB- AND CLOUD SYSTEMS	4		6				**10**	**18**	3.5. Reading research papers on secure and resilient web- and cloud systems 3.6. Prepa-ration of material for seminars according to individual tasks.
3.1. Resilience Models for the Internet and Cloud Computing Systems									
3.2. Redundancy and Diversity Impact on System Security									
3.3. Intrusion Tolerance Techniques									
3.4. Intrusion Avoidance Techniques									
4. STANDARDS AND TOOLS FOR WEB- AND CLOUD SYSTEMS SECURITY	4				4		**8**	**18**	4.5. Penetration testing tools
4.1. Open Web Application Security Project (OWASP)									
4.2. Kali Linux Project for Digital Forensics and Penetration Testing									
4.3. The Metasploit Project									

4.4. Vulnerability Scanners and Security Content Automation Protocol (SCAP)									
Total	**#**		**6**		**12**		**36**	**72**	

Author	**Year of issue**	**Title**	**No of periodical or volume**	**Place of printing. Printing house or internet link**
Compulsory literature				
В. С. Харченко и др.	2017	Secure and Resilient Computing for Industry and Human Domains. Vol. 1.		Харьков: Нац. аэрокосм. ун-т им. Н. Е. Жуковского "ХАИ"
В. С. Харченко и др.	2017	Secure and Resilient Computing for Industry and Human Domains. Vol. 2.		Харьков: Нац. аэрокосм. ун-т им. Н. Е. Жуковского "ХАИ"
A. Gorbenko, A. Romanovsky, O. Tarasyuk and O. Biloborodov	2017	Study of Vulnerabilities of Enterprise Operating Systems		Int. Symp. on Software Reliability Engineering (ISSRE'2017), Toulouse (France), October 23-26, 2017
A. Avizienis, J. Laprie, B. Randell and C. Landwehr	2004	Basic Concepts and Taxonomy of Dependable and Secure Computing	vol. 1, no. 1, pp. 11-33	IEEE Transactions on Dependable and Secure Computing
J. Laprie	2005	Resilience for the Scalability of		4th IEEE International

		Dependability		Symposium on Network Computing and Applications (NCA'05)
J. Laprie	2008	From dependability to resilience		IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'08)
K. Trivedi, D. Kim and R. Ghosh,	2009	Resilience in computer systems and networks		International Conference on Computer-Aided Design
L. Strigini	2012	Resilience: What Is It, and How Much Do We Want?	vol. 10, no. 3, pp. 72-75	IEEE Security and Privacy Magazine
E. Hollnagel, D. Woods and N. Leveson	2006	Resilience Engineering, Concepts And Precepts		Ashgate Publishing
Additional literature				
S. Frei, M. May, U. Fiedler and etc.	2006	Large-scale vulnerability analysis s		SIGCOMM Workshop on Large-Scale Attack Defense
M. Garcia, A. Bessani, I. Gashi and etc.	2011	OS Diversity for Intrusion Tolerance: Myth or Reality?		IEEE/IFIP 41st Int. Conf. on Dependable Systems & Networks (DSN'2011)
A. Gorbenko, O. Tarasyuk, V. Kharchenko and A. Romanovsky	2011	Using Diversity in Cloud-Based Deployment Environment to Avoid Intrusions	LNCS 6968, p. 145–155	Software Engineering for Resilient Systems, Springer: LNCS

СОДЕРЖАНИЕ

БЕЗОПАСНОСТЬ И УСТОЙЧИВОСТЬ ВЕБ- И ОБЛАЧНЫХ СИСТЕМ

Практикум

Под редакцией Харченко В.С.

Авторы: О.М. Тарасюк, А.В. Горбенко

Компьютерная верстка
Тарасюк О.М.

Оригинал-макет изготовлен на кафедре компьютерных систем и сетей
Национального аэрокосмического университета им. Н.Е.Жуковского
“Харьковский авиационный институт”

Подписан к печати 22.02.17
Формат 60×84/16 Бумага офс. №2.
Усл. печ. л. 3 Уч.-изд. л. 4 Тираж 100 экз.
Заказ Цена свободная

Адрес редакции:

ХАИ кафедра 503
Украина, 61070, Харьков-70, ул. Чкалова, 17

Отпечатано ФЛП Лысенко И.Б.
61070, Харьков-70, ул. Чкалова, 17,
моторный корпус, к. 147, т.707-44-76

Свидетельство о внесении субъекта издательского дела
в государственный реестр издателей, изготовителей и распространителей
издательской продукции ДК №2607 от 11.09.06 г.