Научная статья
Original article

Оценка безопасности информационных систем с помощью тестирования на проникновение

Шкрадюк А.Д.
Уральский государственный экономический университет, г. Екатеринбург, Россия
Автор-корреспондент: aleksey.shkradyuk@mail.ru

Аннотация: Подход к защите конфиденциальной информации от внешних угроз требует современных методов, ведь злоумышленники не стоят на месте и идут в ногу с развитием информационных технологий, а некоторые даже на шаг впереди. В данной статье рассмотрен современный метод защиты компьютерной информации от угроз путём имитации атаки злоумышленника извне на компьютерную систему или сеть, называемый в простонародье «Пентест».

Ключевые слова: Информационная безопасность, компьютерные системы, тестирование на проникновение, CTF.

Для цитирования: Шкрадюк А.Д. Оценка безопасности информационных систем путем тестирования на проникновение. Умная цифровая экономика. 2022. Т.2, №4, с. 18-30

Information systems security assessment using penetration testing

Shkradyuk A.D.
Ural State University of Economics, Yekaterinburg, Russia
Corresponding author: aleksey.shkradyuk@mail.ru

Abstract: The approach to protecting confidential information from external threats requires modern methods, because attackers do not stand still and keep pace with the development of information technology, and some are even one step ahead. This article discusses a modern method of protecting computer information from threats by simulating an attacker's attack from outside on a computer system or network, called in the common people "Pentest".

Keywords: Information security, computer systems, penetration testing, CTF.

For citation: Shkradyuk A.D. Information systems security assessment by penetration testing. Smart digital economy. 2022. Т.2, №4, pp. 18-30

Введение

В современном мире изо дня в день организации теряют большое количество ресурсов по причине атак злоумышленников на их информационные системы. Из-за подобных атак, как минимум, может нарушиться работоспособность сервисов компании, могут быть украдены персональные данные с серверов и выложены в общий доступ. Не все компании сообщают о подобных инцидентах, так как не хотят портить свою репутацию, но тем не менее, любая организация, использующая, например, облачные сервисы для хранения своих данных или

клиентских баз, может быть подвержена цифровой атаке. Именно поэтому все организации, которые могут себе позволить, привлекают сторонних специалистов для тестирования своих IT-инфраструктур, веб-приложений и продуктов перед их выпуском в дальнейший оборот.

Главная цель тестирования – выявить любые слабые места в системе или сети, которыми могут воспользоваться злоумышленники. В любом приложении или сети чаще всего есть недостатки, которые злоумышленник может использовать для нарушения конфиденциальности, целостности или доступности данных, а также для дальнейшего эксплуатирования системы.

Получить практический опыт в данной области можно благодаря CTF соревнованиям.

CTF (Capture The Flag) – командные соревнования по компьютерной безопасности, цель которых захватить «флаг» – набор символов или произвольная фраза. Существует два формата проведения соревнований:

- task-based (jeopardy) – игрокам предоставляется набор заданий, к которым требуется найти ответ и отправить его. Каждое задание оценивается различным количеством очков, в зависимости от сложности. Обычно выделяются следующие категории: admin – задачи на администрирование; joy – различные развлекательные задачи вроде коллективной фотографии или мини-игры; ctb – задачи на аудит удалённых машин (crack the box); reverse – исследование программ без исходного кода (реверс-инжиниринг); stegano – стеганография; ppc – задачи на программирование; crypto – криптография; web – задачи на веб-уязвимости (SQL Injection, XSS и др.).

- classic – в классической схеме каждая команда получает выделенный сервер или небольшую сеть для поддержания её функционирования и защиты. Во время игры команды получают очки за корректную работу сервисов своего сервере и за украденную информацию (флаги) с серверов противников [3].

Целью статьи является разработка методики поиска уязвимостей в веб-приложениях с использованием программного обеспечения GNU/Kali-Linux на примере типовой задачи категории ctb форма CTF соревнования.

Основные понятия тестирования на проникновения

Разберёмся, что из себя представляет тестирование на проникновение. Тестирование на проникновение (жарг. Пентест) – метод оценки безопасности компьютерных систем или сетей средствами моделирования атаки злоумышленника [6, 7]. При тестировании на проникновение эксперты создают условия, при которых имитируется реальная атака на информационные системы компании. В общем, специалисты повторяют действия настоящих хакеров, но на легальной основе.

Специалисты по тестированию на проникновение должны уметь то же, что и хакеры, чтобы имитировать атаки на информационные системы. Они могут называть себя «белыми» хакерами, этичными хакерами или же белошляпниками (White hat hacker), действия которых не нарушают закон и даже идут на пользу. Также существуют черношляпники и серошляпники. Черношляпники – люди, действия которых нарушают закон и несут вредительный характер, а серошляпники – это люди, чья активность постоянно скачет на грани законности. Этичные хакеры отличаются от информационных вредителей тем, что не используют тестируемые системы для своих личных целей, а лишь проверяют, может ли она

быть успешно атакована, если да, то как, и что нужно исправить, чтобы этого избежать. После проверки специалистом системы на уязвимости, создаётся отчёт и отправляется заказчику для дальнейшего усиления слабых мест.

Тестирование на проникновение является разносторонним методом проверки защиты компьютерных систем. Всё зависит от заказчика: что ему нужно проверить в своей системе. Это может быть упор на сети, приложения и ПО, социальную инженерию, устройства, а также и на физические системы, или же на всё сразу. Тестируя сети, специалисты ищут слабые узлы, неправильно настроенные протоколы, используют открытые порты, которые, по-хорошему, не должны быть открытыми. В локальных или сетевых приложениях, а также на крупных сайтах, пентестеры подделывают запросы, пытаются получить доступ к базе данных, встраивают в код вредоносные скрипты и мешают работе сеансов, и это только часть возможных действий. В социальной инженерии важен человеческий фактор: могут ли сотрудники случайно или намеренно сломать систему, поддаться на провокации злоумышленников, например, посредством фишинга – самым известным способом атаки с привлечением сотрудников. Тестируя устройства, этичные хакеры находят программные и аппаратные уязвимости, слабые места сети, к которой подключено устройство, пытаются получить пароли с помощью брутфорса. Физическими системами может являться дата-центр или любое другое охраняемое помещение. Кроме IT-инфраструктуры тестируется возможность взломать физический замок, обойти или вывести из строя камеры видеонаблюдения и различные датчики.

Существуют различные методики тестирования:

1. Внешнее тестирование – проводится от лица, атакующего с внешней стороны. Человек, не относящийся к компании, определяет возможные пути получения доступа к системе дистанционно.
2. Внутреннее тестирование – тестирование от лица пользователя со стандартными правами сотрудника компании, во время которого отыскивается возможность навредить системе изнутри случайным или специальным образом.
3. Белый ящик – у пентестера имеются знания о системе.
4. Черный ящик (слепое тестирование) – тестировщик не имеет предварительной информации о системе и ведёт себя как настоящий злоумышленник. Может опираться лишь на данные, имеющиеся в открытом доступе.
5. Серый ящик – имеются частичные знания о системе.
6. Двойное слепое тестирование – тестирование, находящееся в секрете (в том числе и от службы безопасности), о котором знают буквально 1-2 человека в компании. Данный метод помогает выявить уязвимости, которые нельзя обнаружить с помощью предыдущих методов. В случае такого метода тестирования, необходимо иметь при себе документы, подтверждающие, что тестировщик работает легально. Иначе могут быть проблемы со службой безопасности и законом.

Методика тестирования уязвимостей с помощью инструментальных средств Kali-Linux

Самый сильный и наиболее популярный инструмент любого пентестера – это дистрибутив Kali-Linux, основанный на базе Debian. Он включает в себя набор программ для тестирования уязвимостей и угроз безопасности прямо «из коробки». Именно этот инструмент будет использоваться в практической части данной статьи.

Для примера возьмём одну из задач CTF формата task-based категории ctb. В этом нам поможет сайт «TryHackMe», на котором в бесплатном доступе предоставляется множество виртуальных машин различных уровней сложности, но также имеется и платная подписка, расширяющая возможности пользователя. Для начала регистрируемся на сайте и скачиваем конфигурационный openvpn-файл, чтобы подключиться к сети «TryHackMe» и иметь доступ к тестируемым машинам. Для этого переходим во вкладку «Access», нажав на аватар своего профиля, скачиваем конфигурационный openvpn-файл и запускаем его в GNU/Linux (рисунки 1-3).

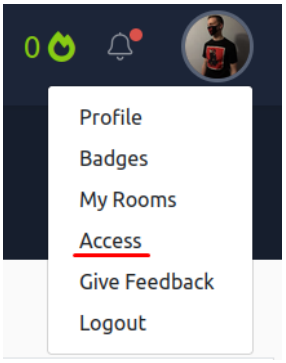

Рисунок 1. Вкладка «Access»

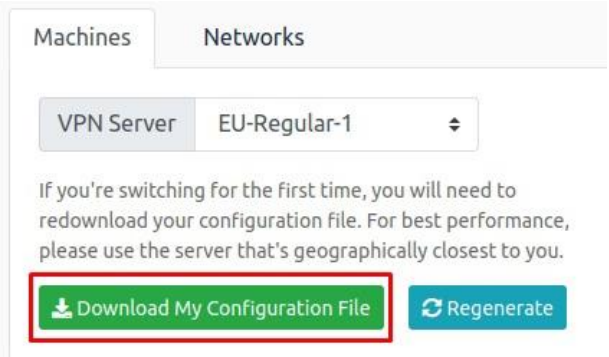

Рисунок 2. Загрузка конфигурационного файла

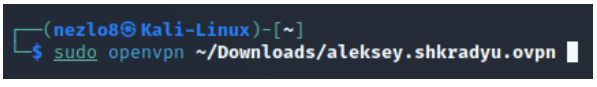

Рисунок 3. Запуск конфигурационного файла

Теперь можем приступать к выбору виртуальной машины.

Поищем несложную машину для дальнейшей попытки её взлома. Для этого переходим во вкладку «Learn», находящейся в «шапке» сайта в левом верхнем углу и выбираем «Search» (рисунок 4).

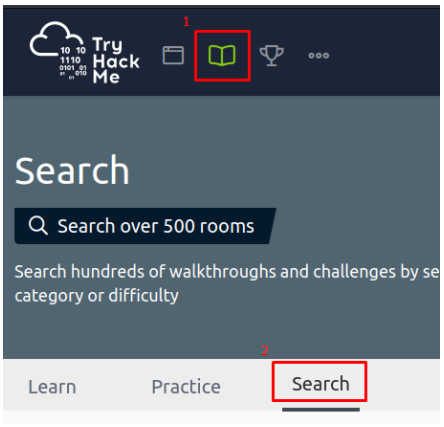

Рисунок 4. Поиск виртуальной машины

Выбираем виртуальную машину под названием «Simple CTF» (рисунок 5).

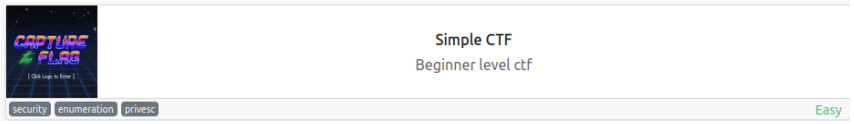

Рисунок 5. Выбор виртуальной машины

Перед нами раскрывается список вопросов, на которые в ходе выполнения задания нужно будет дать ответы (рисунок 6).

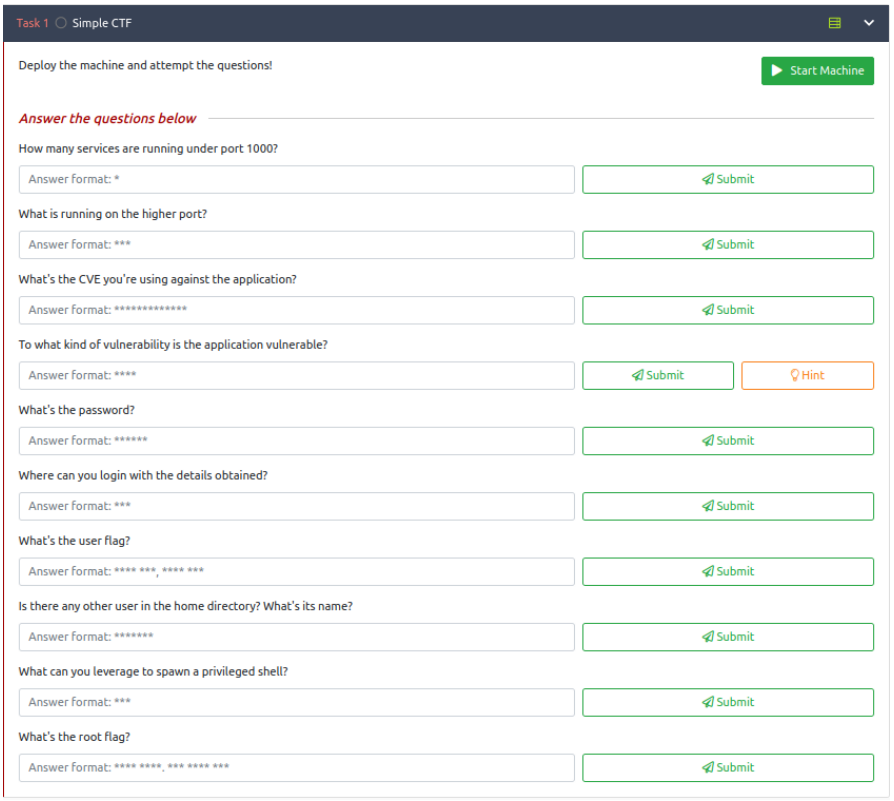

Рисунок 6. Список вопросов

Приступим к разбору данного задания. Запускаем виртуальную машину и ждём появления информации о ней (рисунок 7).

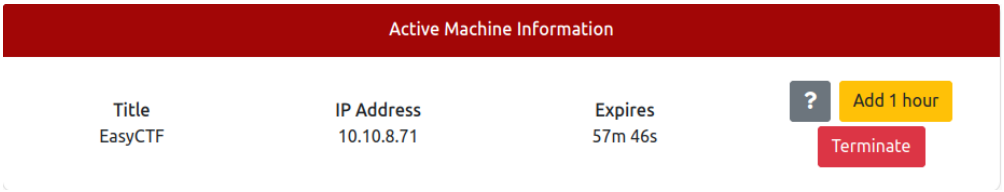

Рисунок 7 – Информация о запущенной виртуальной машине

Видим название машины, её IP-адрес и срок действия, который можно продлевать, если времени осталось менее часа. Первым делом пингуем полученный IP-адрес для проверки связи между нами и машиной (рисунок 8).

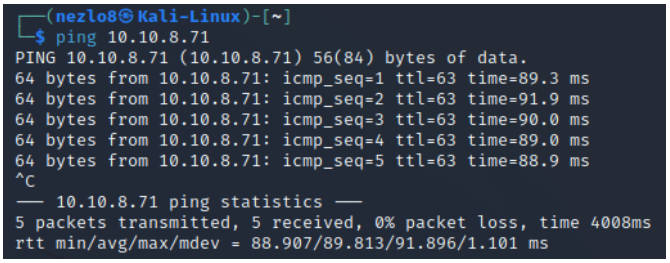

Рисунок 8. Проверка связи с машиной

Машина отвечает на наши запросы. Приступаем к ответам на вопросы. Вопрос 1: сколько сервисов запущено под портом 1000? Для ответа на данный вопрос воспользуемся утилитой nmap и просканируем порты.

```
┌──(nezlo8㉿Kali-Linux)-[~]
└─$ nmap 10.10.8.71 -vv
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-21 11:37 EST
Initiating Ping Scan at 11:37
Scanning 10.10.8.71 [2 ports]
Completed Ping Scan at 11:37, 0.09s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:37
Completed Parallel DNS resolution of 1 host. at 11:37, 0.00s elapsed
Initiating Connect Scan at 11:37
Scanning 10.10.8.71 [1000 ports]
Discovered open port 21/tcp on 10.10.8.71
Discovered open port 80/tcp on 10.10.8.71
Discovered open port 2222/tcp on 10.10.8.71
Completed Connect Scan at 11:37, 7.19s elapsed (1000 total ports)
Nmap scan report for 10.10.8.71
Host is up, received syn-ack (0.087s latency).
Scanned at 2022-11-21 11:37:02 EST for 7s
Not shown: 997 filtered tcp ports (no-response)
PORT     STATE SERVICE      REASON
21/tcp   open  ftp          syn-ack
80/tcp   open  http         syn-ack
2222/tcp open  EtherNetIP-1 syn-ack

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 7.31 seconds
```

Рисунок 9. Список открытых портов сервера

Можем заметить, что у нас есть два подходящих активных порта (рисунок 9). Порт 2222 не учитывается, т.к. в вопросе спрашивается про порты ниже, чем 1000. Отправляем ответ на сайте.

Рисунок 10. Ответ на первый вопрос

Ответ верный (рисунок 10). Приступаем к следующему вопросу: что работает на верхнем порту? Наш верхний порт – это 2222, он отвечает за соединение по протоколу «ssh». Отправляем ответ и убеждаемся, что он тоже верный (рисунок 11).

Рисунок 11. Ответ на второй вопрос

Далее нас спрашивают, какой CVE мы будем использовать для атаки на приложение. Здесь явный намёк на то, что мы должны использовать эксплойт. Эксплойт – компьютерная программа, фрагмент программного кода или последовательность команд, использующие уязвимости в программном обеспечении и применяемые для проведения атаки на вычислительную систему. У каждого эксплойта есть свой код CVE – что-то вроде идентификатора. Для поиска подходящего нам эксплойта нужно узнать название и версию

используемого веб-приложения. Чтобы получить эти данные, используем программу «dirsearch» для поиска скрытых директорий.

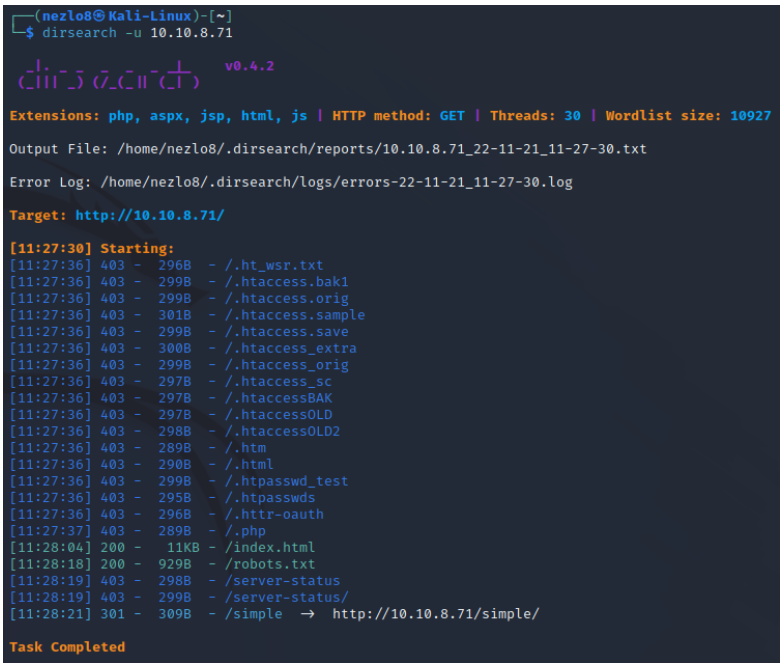

Рисунок 12. Список скрытых директорий

Видим, что нашлась одна скрытая директория (рисунок 12). Переходим на http://10.10.8.71/simple/ и смотрим, что мы имеем.

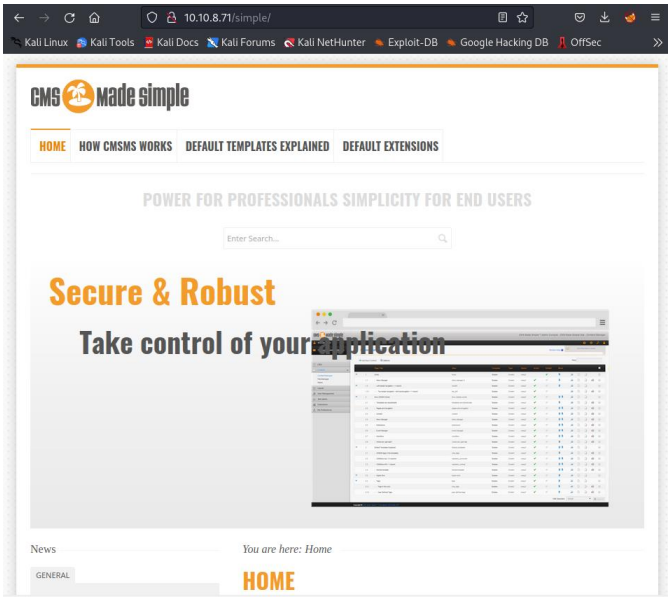

Рисунок 13. CMS Made Simple

В результате у нас есть название CMS (система создания и управления сайтом) – «Made Simple» (рисунок 13). На этой же странице определяем версию CMS системы (рисунок 14).

Рисунок 14. Версия CMS Made Simple

Выполняем поиск эксплойтов с имеющимися у нас данными в терминале Linux, но можно и воспользоваться сайтом «www.exploit-db.com», базы данных одни и те же. Здесь уже дело вкуса, кому как удобнее, но лучше привыкать работать в терминале.

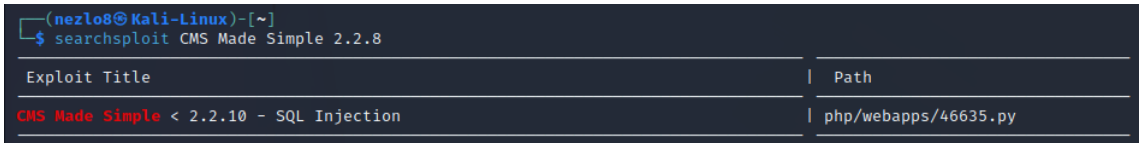

Рисунок 15. Найденный эксплойт

Найден подходящий эксплойт (рисунок 15), в основе которого лежит SQL-инъекция – уязвимость, которая возникает, когда у злоумышленника появляется возможность модифицировать SQL-запрос в приложении. Чтобы ответить на вопрос, нам нужно узнать CVE данного эксплойта. Внутри эксплойта всегда находится его описание, в том числе и CVE. Выводим текст скрипта (рисунок 16).

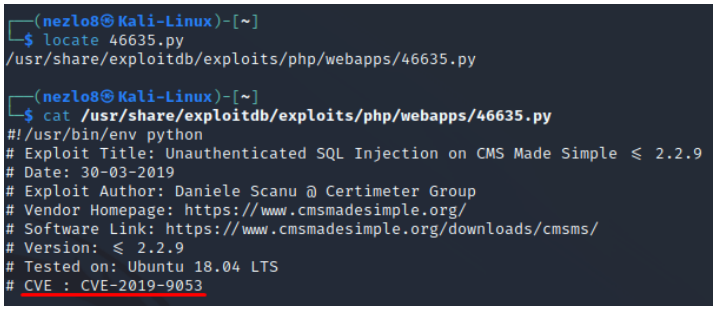

Рисунок 16. CVE эксплойта

Можем отправить ответ сразу на два вопроса (рисунок 17).

Рисунок 17. Ответы на третий и четвертый вопросы

Дальше начинается самое интересное: нам предстоит добыть пароль и получить доступ к машине. Для этого применяем найденный эксплойт. Чтобы понять, как он работает, снова просматриваем скрипт, находим описание применения атрибутов «-u», «-w» и «-c» (рисунок 18).

```
parser = optparse.OptionParser()
parser.add_option('-u', '--url', action="store", dest="url", help="Base target uri (ex. http://10.10.10.100/cms)")
parser.add_option('-w', '--wordlist', action="store", dest="wordlist", help="Wordlist for crack admin password")
parser.add_option('-c', '--crack', action="store_true", dest="cracking", help="Crack password with wordlist", defau
```

Рисунок 18. Описание применения атрибутов «-u», «-w» и «-c»

Запускаем эксплойт с применением вышеприведенных атрибутов (рисунок 19).

```
┌──(nezlo8㉿Kali-Linux)-[~]
└─$ python2 /usr/share/exploitdb/exploits/php/webapps/46635.py -u http://10.10.8.71/simple/ -c -w /usr/share/wordlis
ts/rockyou.txt
```

Рисунок 19. Запуск эксплойта

Спустя некоторое время получаем данные пользователя (рисунок 20).

```
[+] Salt for password found: 1dac0d92e9fa6bb2
[+] Username found: mitch
[+] Email found: admin@admin.com
[+] Password found: 0c01f4468bd75d7a84c7eb73846e8d96
[+] Password cracked: secret
```

Рисунок 20. Полученные данные пользователя

Отправляем ответ на пятый вопрос (рисунок 21).

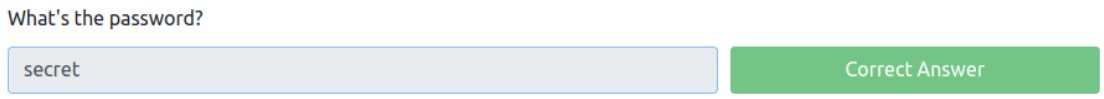

Рисунок 21. Ответ на пятый вопрос

Приступаем к следующему вопросу: где можно войти с полученными данными? Вспоминаем, что у нас имеется открытый порт 2222, который отвечает за ssh-соединение. Очевидно, что с полученными данными мы можем войти по протоколу «ssh». Отправляем ответ (рисунок 22).

Рисунок 22. Ответ на шестой вопрос

Подключаемся к серверу по протоколу «ssh» и находим в каталоге пользователя файл user.txt, выводим его содержимое в консоль (рисунок 23).

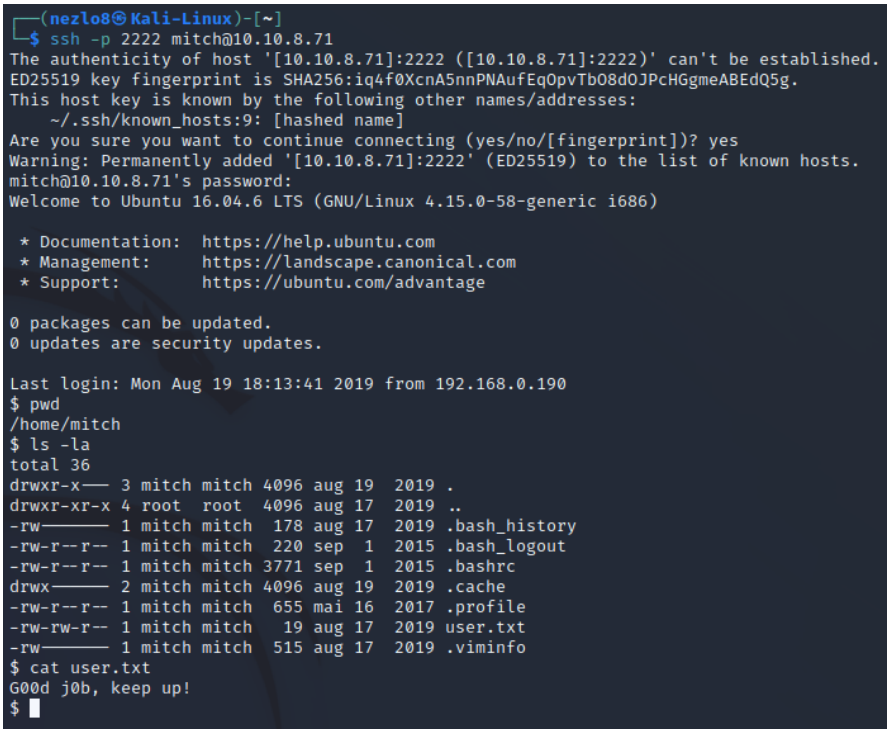

Рисунок 23. Флаг пользователя

Отправляем найденный флаг на сайт (рисунок 24).

Рисунок 24. Ответ на седьмой вопрос

Мы получили доступ к серверу, используя удалённое подключение по протоколу «ssh». Следующее, что от нас требуется – найти имя ещё одного пользователя. Для этого смотрим содержимое директории /home и видим ещё одного пользователя под именем «sunbath» (рисунок 25). Это и есть наш ответ. Отправляем на сайт (рисунок 26).

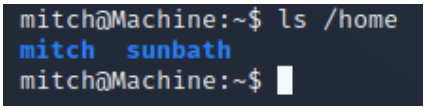

Рисунок 25. Второй пользователь

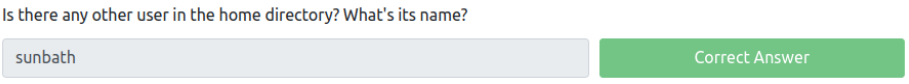

Рисунок 26. Ответ на восьмой вопрос

Далее нас спрашивают, что мы можем сделать для получения привилегированной оболочки. Посмотрим, какие команды можно выполнить с root-правами без ввода root-пароля, введя команду «sudo -l» (рисунок 27).

```
mitch@Machine:~$ sudo -l
User mitch may run the following commands on Machine:
    (root) NOPASSWD: /usr/bin/vim
mitch@Machine:~$
```

Рисунок 27. Команды, разрешенные на выполнение с root-правами без ввода root-пароля

Видим, что доступна команда «vim». Фиксируем данную информацию на сайте (рисунок 28).

Рисунок 28. Ответ на девятый вопрос

Мы стали ещё ближе к получению root-прав. Используем команду «vim» с root-правами для получения привилегированной оболочки и находим флаг root-пользователя (рисунок 29).

```
mitch@Machine:~$ sudo vim -c ':!/bin/bash'

root@Machine:~# pwd
/home/mitch
root@Machine:~# cd /root
root@Machine:/root# ls
root.txt
root@Machine:/root# cat root.txt
W3ll d0n3. You made it!
root@Machine:/root#
```

Рисунок 29. Получение root-прав и флага root-пользователя

Отправляем ответ и завершаем задание (рисунок 30).

Рисунок 30. Ответ на десятый вопрос

Задание успешно выполнено. Мы получили root-права и ответили на все вопросы в задании.

Подводя итоги данного исследования, можно сделать вывод, что в современное время почти любая информационная система имеет те или иные уязвимости, и одним из лучших способов их пресечь является привлечение специалистов по тестированию на проникновение для обнаружения слабых мест и дальнейшего их усиления. В статье были рассмотрены современный метод оценки безопасности компьютерных систем или сетей средствами моделирования атаки злоумышленника и способ получения практических навыков в этой сфере. Также было приведено решение одного из типовых кейсов для демонстрации практической стороны данного метода.

Список литературы

1. Википедия. Испытание на проникновение. [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Испытание_на_проникновение (дата обращения: 21.11.2022).

2. SkillFactory.Блог. Пентест. [Электронный ресурс]. URL: https://blog.skillfactory.ru/glossary/pentest/ (дата обращения: 20.11.2022).

3. Курс молодого бойца CTF. [Электронный ресурс]. URL: https://kmb.cybber.ru/ (дата обращения: 21.11.2022).

4. Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution. [Электронный ресурс]. URL: https://www.kali.org/ (дата обращения: 21.11.2022).

5. TryHackMe. [Электронный ресурс]. URL: https://tryhackme.com/ (дата обращения: 21.11.2022).

6. Georgia Weidman. Penetration Testing. A Hands-On Introduction to Hacking. San Francisco, 2014. 495 c.

7. Kanimozhi V.R and Shanmugapriya N., 2019, Ethical Hacking: the Need for Cyber Security. Int J Recent Sci Res. 10(10), pp. 35339-35341.