# ЛАБОРАТОРНА РОБОТА №8
# АВТОМАТИЗАЦІЯ ВИЯВЛЕННЯ ТА ЕКСПЛУАТАЦІЇ SQL INJECTION ЗА ДОПОМОГОЮ SQLMAP

Мета роботи:

1. Ознайомлення з принципами роботи інструменту sqlmap.
2. Вивчення методів виявлення SQL-ін’єкцій автоматизованими засобами.
3. Освоєння методів отримання даних із баз даних за допомогою sqlmap.
4. Дослідження впливу параметрів sqlmap на ефективність та стабільність атаки.

## Теоретичні відомості

### sqlmap

sqlmap – це високорівневий автоматизований інструмент з відкритим кодом, призначений для проведення тестування на проникнення (penetration testing) веб-додатків з метою виявлення та експлуатації вразливостей класу SQL Injection (SQLi).

Інструмент реалізує повністю автоматизований підхід до аналізу вхідних точок, що взаємодіють із базою даних, та дозволяє не лише ідентифікувати наявність SQL-ін’єкцій, але й виконувати подальшу експлуатацію, включаючи отримання конфіденційної інформації, модифікацію даних, а також взаємодію з операційною системою сервера.

Архітектурно sqlmap складається з модулів:

1. Детекції (detection engine) – визначає наявність вразливостей.
2. Фінгерпринтингу (fingerprinting engine) – встановлює тип СУБД, її версію та особливості.
3. Експлуатації (exploitation engine) – реалізує витяг даних та інші атаки.
4. Обходу захистів (tamper engine) – застосовує техніки обходу WAF/IDS.

Інструмент має широкий спектр функціональних можливостей, що покривають повний цикл атаки. Sqlmap підтримує роботу з більшістю сучасних систем управління базами даних:

1. MySQL/MariaDB.
2. PostgreSQL.
3. Oracle Database.
4. Microsoft SQL Server.
5. SQLite.
6. IBM DB2.
7. Firebird.
8. SAP MaxDB.
9. Sybase.
10. Informix.
11. HSQLDB.

Це забезпечує універсальність інструменту при тестуванні різних технологічних стеків.

Таблиця 1. Підтримка методів SQL-ін’єкцій

| Метод | Опис |
|---|---|
| Boolean-based blind | Аналіз відповіді сервера на логічні умови (true/false) |
| Time-based blind | Використання затримок (SLEEP) для визначення істинності запиту |
| Error-based | Отримання інформації через повідомлення про помилки |
| UNION query-based | Об’єднання запитів для витягу даних |
| Stacked queries | Виконання декількох SQL-запитів за один раз |
| Out-of-band (OOB) | Використання зовнішніх каналів (DNS, HTTP) |

**Автоматизація атак**

Однією із ключових переваг інструменту sqlmap є високий рівень автоматизації процесу тестування та експлуатації SQL-ін’єкцій. На відміну від

ручного підходу, де фахівець самостійно формує payload-и та аналізує відповіді сервера, sqlmap реалізує інтелектуальний механізм, що дозволяє значно скоротити час аналізу та мінімізувати людський фактор.

Зокрема, sqlmap автоматично виконує такі етапи:

1. Аналіз всіх доступних точок введення (GET, POST, cookies, HTTP-заголовки) та ідентифікація тих параметрів, які потенційно взаємодіють із SQL-запитами.

2. Використання великої бази готових payload-ів, адаптованих під різні типи SQL-ін'єкцій і СУБД. Підбір відбувається динамічно з урахуванням поведінки цільового додатку, що підвищує точність детекції.

3. Визначення типу системи управління базами даних (наприклад, MySQL, PostgreSQL, Oracle) шляхом аналізу специфічних особливостей відповіді сервера, синтаксису SQL та поведінки функцій.

4. Вибір найбільш ефективної техніки експлуатації (наприклад, UNION-based замість blind), якщо це можливо. У випадках, коли доступні лише blind SQL-ін'єкції, застосовуються оптимізаційні алгоритми (бінарний пошук, кешування відповідей), що зменшують кількість запитів до сервера.

**Синтаксис sqlmap**

Інструмент запускається через командний рядок (CLI), що забезпечує гнучкість і можливість автоматизації.

Базовий синтаксис:

```
sqlmap [опції]
```

Найпростіший приклад використання:

```
sqlmap -u "http://target.com/page.php?id=1"
```

У цьому випадку sqlmap:

1. Аналізує параметр id.
2. Визначає, чи є він вразливим.
3. Підбирає відповідні техніку атаки.

Для ефективного використання інструменту sqlmap необхідно коректно задати цілі тестування, включаючи адресу ресуру, тип запиту та додаткові

параметри HTTP-взаємодії. Основні параметри, що використовуються для визначення цільового об’єкта, наведено в таблиці 2.

Таблиця 2. Основні параметри sqlmap для задання цілі

| Параметр | Опис | Приклад використання |
|---|---|---|
| **-u** | URL-адреса для тестування | -u "http://site.com?id=1" |
| **-r** | Завантаження HTTP-запиту з файлу | -r request.txt |
| **--data** | Передача POST-даних | --data="id=1&user=admin" |
| **--cookie** | Встановлення cookie | --cookie="PHPSESSID=123" |
| **--headers** | Додавання HTTP-заголовків | --headers="X-Forwarded-For: 127.0.0.1" |
| **--method** | HTTP-метод | --method=POST |

Окрім базового задання цілі, важливу роль відіграють параметри, що визначають глибину та агресивність тестування. Вони впливають на кількість перевірок, складність використовуваних payload-ів та загальну ефективність процесу виявлення вразливостей. Відповідні параметри наведено в таблиці 3.

Таблиця 3. Параметри глибини та агресивності тестування

| Параметр | Діапазон | Опис | Вплив |
|---|---|---|---|
| **--level** | 1-5 | Рівень тестування | Більше значення – більша кількість payload-ів |
| **--risk** | 1-3 | Рівень ризику атак | Більше значення – агресивніші запити |
| **--threads** | 1-10+ | Кількість потоків | Більша швидкість, але більше навантаження |
| **--timeout** | секунди | Таймаут відповіді | Впливає на стабільність |
| **--retries** | число | Кількість повторів | Підвищує надійність |

Перед безпосереднім отриманням даних із бази даних доцільно виконати аналіз середовища СУБД, зокрема визначити поточного користувача, його

привілеї та версію сервера бази даних. Це дозволяє оцінити можливості подальшої експлуатації. Відповідні параметри наведено в таблиці 4.

Таблиця 4. Параметри для отримання службової інформації про СУБД

| Параметр | Опис | Призначення |
|---|---|---|
| --current-user | Отримання поточного користувача БД | Визначення, від імені кого виконуються запити |
| --privileges | Отримання прав користувача | Аналіз рівня доступу |
| --banner | Отримання банера СУБД | Визначення типу та версії БД |

Після успішного виявлення вразливості наступним етапом є отримання інформації з бази даних. Slqmap надає широкий набір інструментів для навігації по структуру бази даних та витягу необхідних даних. Основні параметри для роботи з даними наведено в таблиці 5.

Таблиця 5. Параметри для отримання даних з бази даних

| Параметр | Опис | Приклад |
|---|---|---|
| **--dbs** | Отримати список БД | --dbs |
| **-D** | Вибір бази даних | -D users_db |
| **--tables** | Отримати таблиці | --tables |
| **-T** | Вибір таблиці | -T users |
| **--columns** | Отримати стовпці | --columns |
| **-C** | Вибір стовпців | -C username,password |
| **--dump** | Вивантаження даних | --dump |

У реальних умовах тестування веб-додатки часто захищені додатковими механізмами безпеки, такими як Web Application Firewall (WAF) або системи виявлення вторгнень. Для підвищення ефективності атак sqlmap підтримує параметри маскування та обходу захисту, які наведено в таблиці 6.

Таблиця 6. Параметри обходу захисту та маскування

| Параметр | Опис | Приклад | Вплив |
|---|---|---|---|
| **--tamper** | Використання tamper-скриптів | **--tamper=space2comment** | Обхід WAF |
| **--random-agent** | Випадковий User-Agent | **--random-agent** | Маскування |
| **--proxy** | Використання проксі | **--proxy=http://127.0.0.1:8080** | Перехоплення |
| **--tor** | Використання Tor | **--tor** | Приховування IP |
| **--delay** | Затримка між запитами | **--delay=2** | Менший ризик блокування |

Крім базових можливостей отримання даних, sqlmap підтримує розширені функції, що дозволяють взаємодіяти з файловою системою сервера та операційною системою. Це значно розширює можливості експлуатації вразливостей. Відповідні параметри наведено в таблиці 7.

Таблиця 7. Параметри експлуатації та розширених можливостей

| Параметр | Опис | Приклад |
|---|---|---|
| **--os-shell** | Доступ до shell ОС | --os-shell |
| **--file-read** | Читання файлу | --file-read=/etc/passwd |
| **--file-write** | Запис файлу | --file-write=shell.php |
| **--batch** | Автоматичні відповіді | --batch |
| **--is-dba** | Перевірка прав DBA | --is-dba |

### Приклад використання sqlmap

Базове тестування та отримання БД:

```
sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" --dbs --batch
```

Де:

1. **-u** – задає цільовий URL з параметром cat.

2. **--dbs** – виконує спробу отримання списку баз даних.

3. **--batch** – автоматично погоджується з усіма запитами sqlmap (без взаємодії з користувачем)

Витяг даних з конкретної таблиці:

```
sqlmap -u "http://testphp.vulnweb.com/listproducts.php?cat=1" \
-D acuart -T users -C username,password --dump --batch
```

Де:

1. **-D acuart** – вибір бази даних.
2. **-T users** – вибір таблиці.
3. **-C username,password** – визначення колонок.
4. **--dump** – витяг даних.
5. **--batch** – автоматичний режим.

## Завдання на лабораторну роботу

**Завдання 1.** Кімната SQLMAP на платформі TryHackMe.

1. Увійдіть в обліковий запис на платформі TryHackMe:

https://tryhackme.com

2. Перейдіть до кімнати SQLMAP:

https://tryhackme.com/room/sqlmap

3. Натисність “Join Room” для розблокування завдань:

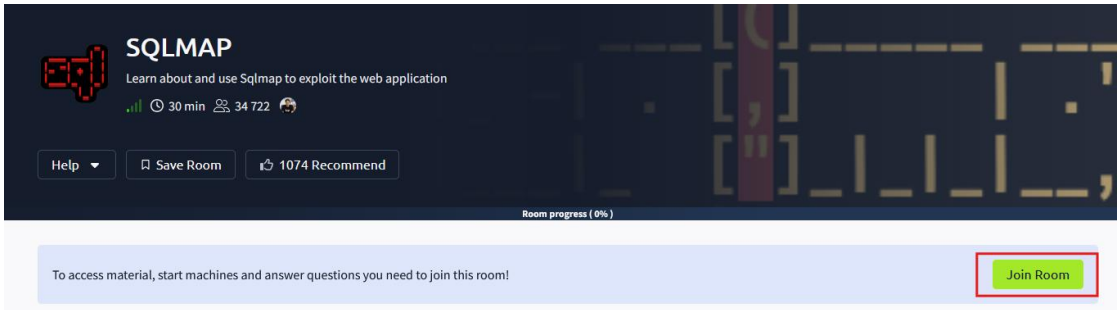


Рисунок 1 – “Join Room” для отримання доступу до завдань

4. Ознайомтеся з теоретичною частиною кімнати.

5. Дайте відповіді на запитання в завданні 2 – Using Sqlmap.

6. У новій вкладці браузера перейдіть до налаштувань облікового запису на платформі TryHackMe:

https://tryhackme.com/manage-account/account-details

7. Перейдіть до розділу “VM and VPN Settings” та натисність “Configuration files”.

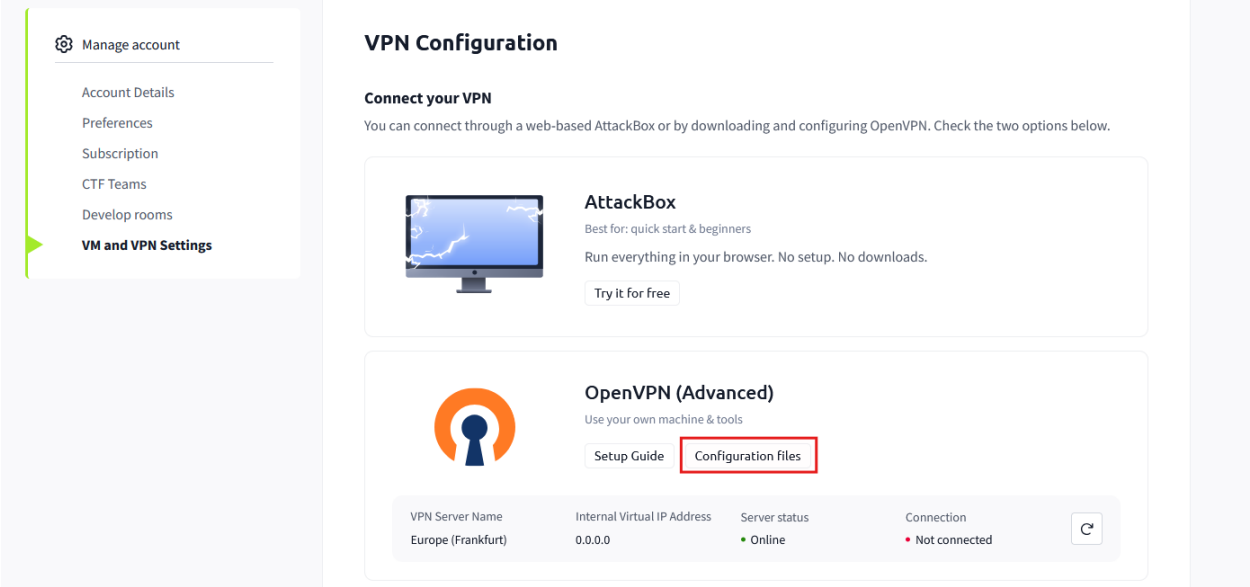


Рисунок 2 – Процес завантаження VPN-налаштувань

8. У вікні “Configuration file” натисність “Download configuration file”.

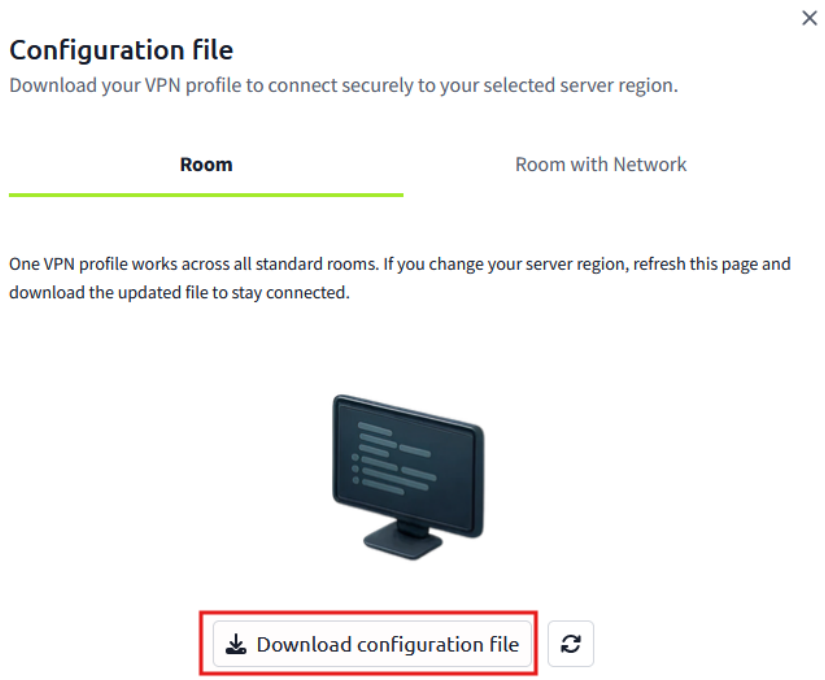


Рисунок 3 - Процес завантаження VPN-налаштувань

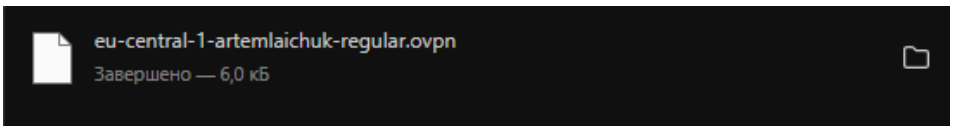


Рисунок 4 – Завантажений OVPN-файл для підключення до VPN платформи TryHackMe

9. Запустіть Kali Linux (збірка від Cisco) та скопіюйте завантажений OVPN-файл на віртуальну машину.

10. Виконайте підключення до VPN за допомогою утиліти openvpn:

*sudo openvpn назва-ovpn-файлу.ovpn*

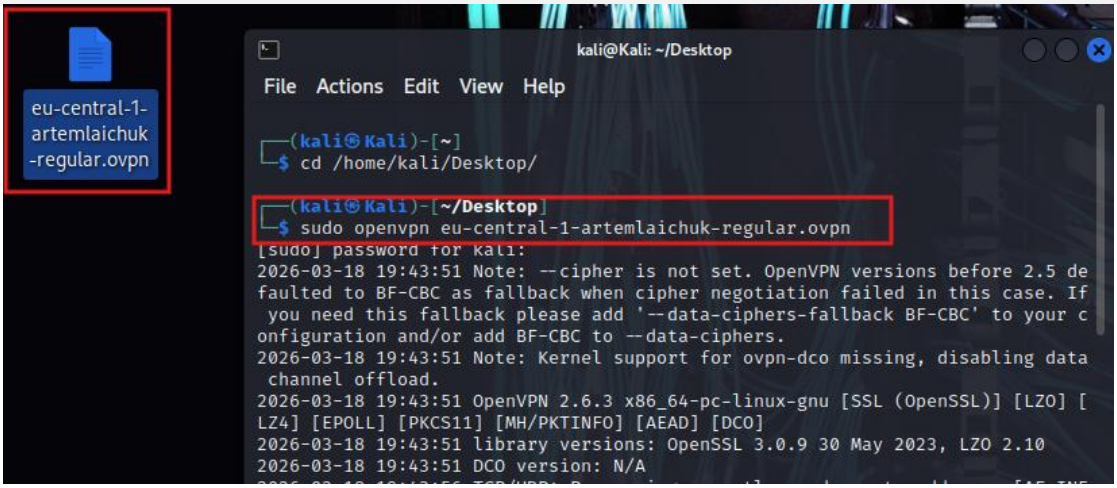


Рисунок 5 – Приклад підключення до VPN платформи TryHackMe

11. Дочекайтеся появи повідомлення “Initialization Sequnce Completed”, яке свідчить про успішне підключення до VPN платформи TryHackMe.

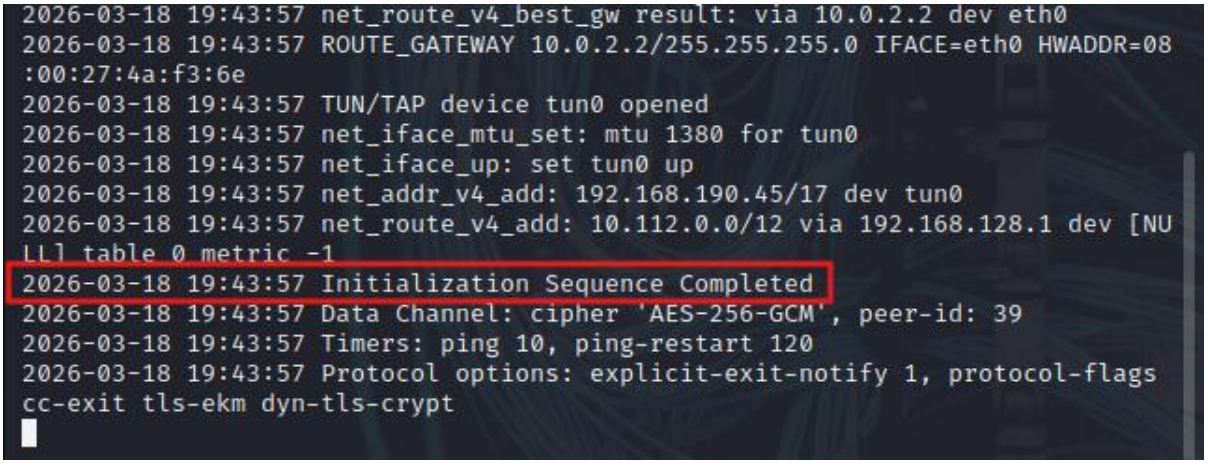


Рисунок 6 – Приклад успішного підключення до VPN

12. Поверніться до кімнати SQLMAP на платформі TryHackMe. Розгорніть завдання №3 та запустіть вразливу машину, натиснувши “Start Machine”.

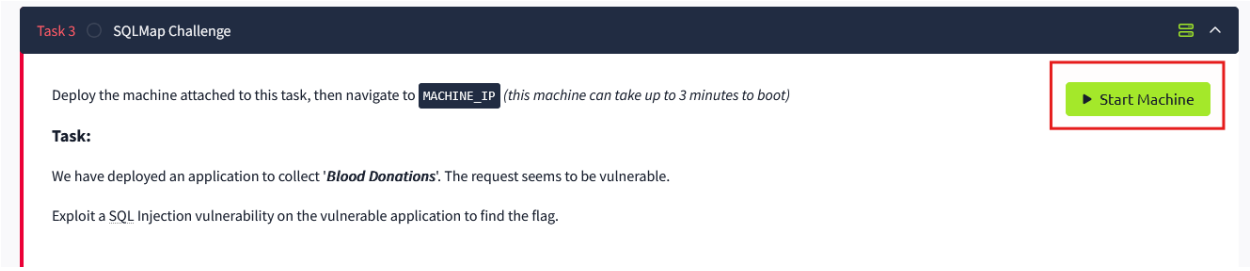


Рисунок 7 – Запуск вразливої машини

13. Дочекайтеся розгортання вразливої машини (1 хв) та скопіюйте її IP-адресу.

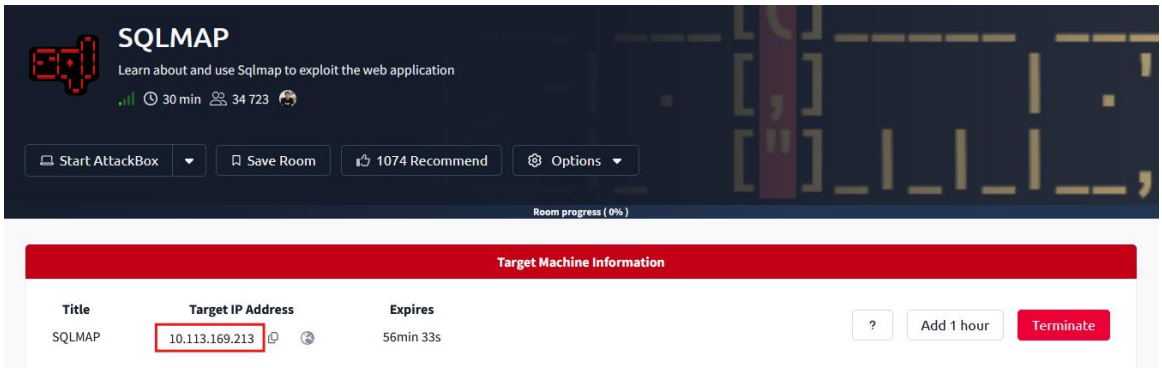


Рисунок 8 – IP-адреса вразливої машини

14. Перевірте доступність вразливої машини шляхом виконання команди ping в терміналі Kali Linux.

15. Перевірте доступність вразливого вебсайту у браузері.

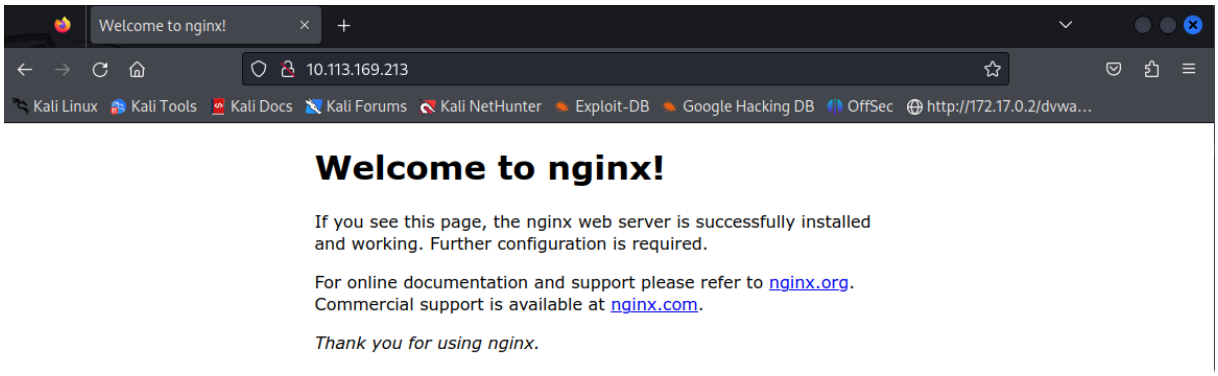


Рисунок 9 – Приклад успішно розгорнутого вебсайту

16. Для відповіді на запитання №1 в завданні №3 – SQLMap Challenge виконайте перебір директорій/шляхів в межах вебсайту за допомогою утиліти **ffuf** із використанням наступного словника:

```
ffuf -u http://IP-адреса-вебсайту/FUZZ -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -s
```

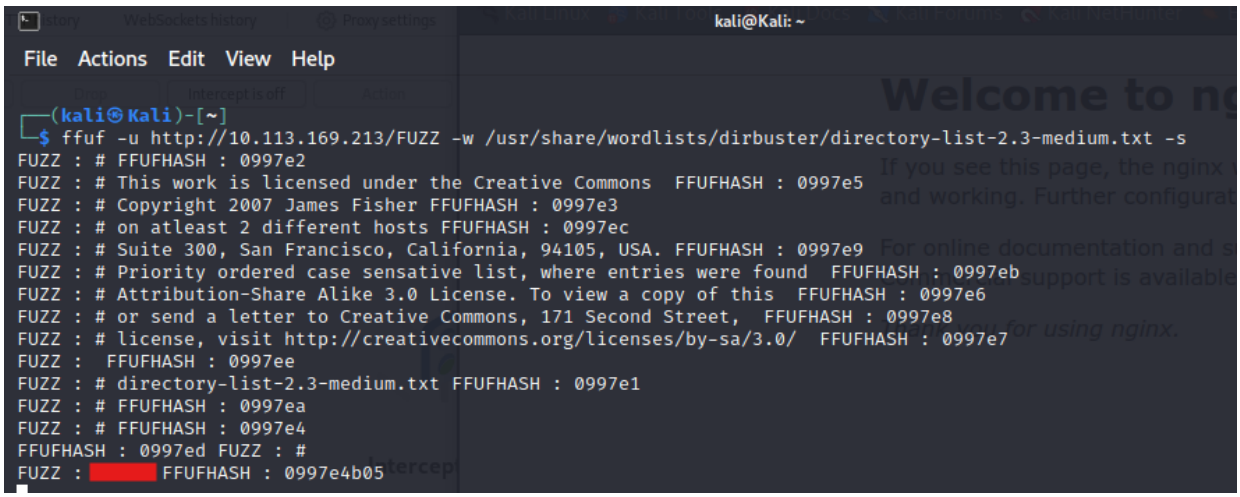


Рисунок 10 – Приклад перебору директорій в межах вебсайту

17. Запустіть інструмент BurpSuite та вбудований в нього браузер Chromium.

18. У вбудованому браузері перевірте доступність знайденої сторінки в результаті виконання п. 16:

*http://IP-адреса/знайдений-шлях/*

19. Для успішної атаки за допомогою sqlmap необхідно використати готовий POST-запит до сторінки. Увімкніть перехоплення запитів в BurpSuite та на вразливому вебсайті натисність “Search”.

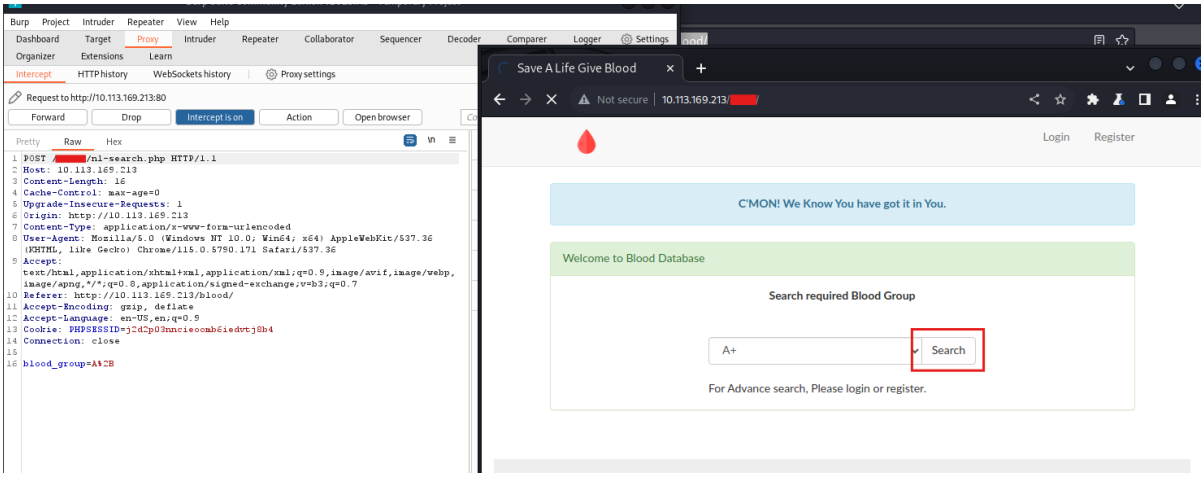


Рисунок 11 – Приклад перехопленого POST-запиту

20. Натисність ПКМ → Copy to file та збережіть POST запит в текстовий файл з назвою **postreq.txt**.

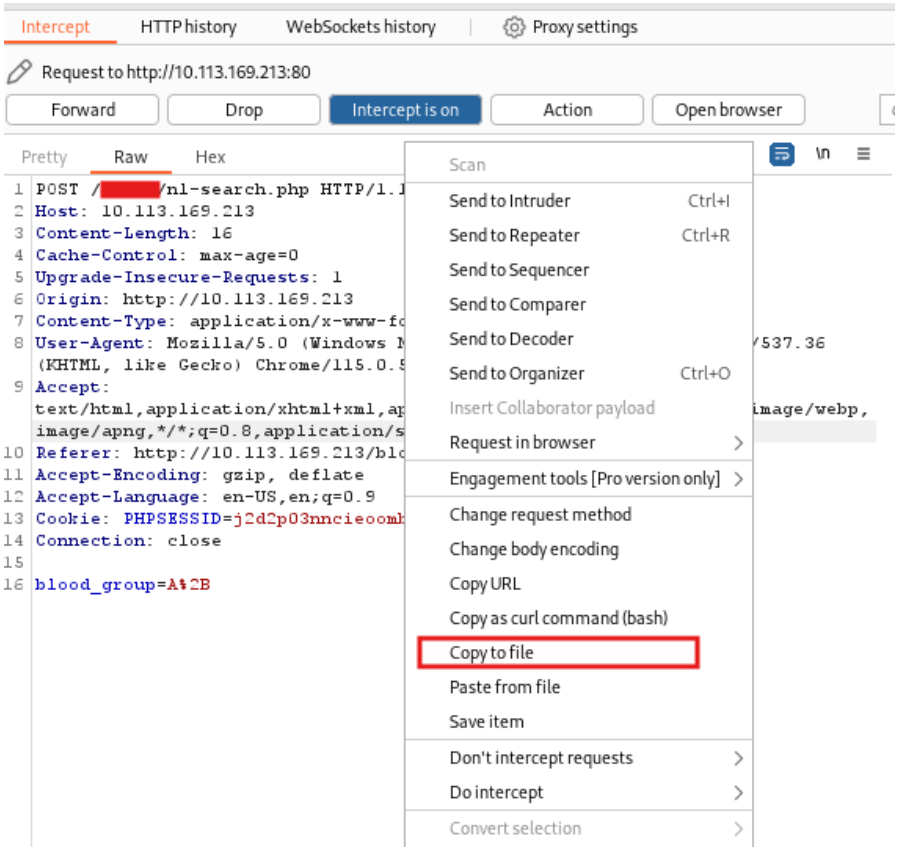


Рисунок 12 – Приклад збереження HTTP запиту в текстовий файл

21. Переконайтеся в успішному збереженні текстового файлу із HTTP-запитом:

```
ls -la postreq.txt
cat postreq.txt
```

22. Виконати експлуатацію SQL Injection за допомогою sqlmap.

Приклад виконання:

```
sqlmap -r postreq.txt -p blood_group --dbs
```

Пояснення:

1. -r postreq.txt – використання збереженого HTTP-запиту.
2. -p blood_group – параметр, який перевіряється на вразливість.
3. --dbs – отримання списку доступних баз даних.


Рисунок 13 – Приклад отримання списку доступних баз даних

23. У ході подальшого дослідження виконайте наступні дії з використанням інструменту sqlmap:

- Отримайте список баз даних на сервері.

- Отримайте список таблиць у базі даних blood.

- Отримайте структуру таблиці flag (визначення назв колонок).

- Отримайте структуру таблиці users.

- Вивантажіть всі дані із таблиці users.

- Вивантажіть всі дані із таблиці flag.

- Вивантажіть лише колонки username та password із таблиці users.

- Виконайте пошук колонок, пов’язаних із паролями, у базі даних.

- Отримайте інформацію про версію та тип системи управління базами даних (banner).

- Визначте поточного користувача бази даних.

- Визначте привілеї користувача бази даних.

24. Використайте отримані результати (п. 23) для відповіді на запитання 2-3 в кімнаті SQLMAP (TryHackMe).

## Контрольні запитання

1. Який параметр sqlmap використовується для вказання цільового URL?
2. Яка команда дозволяє отримати список баз даних?
3. Для чого використовується параметр -r у sqlmap?
4. Який метод SQL Injection передбачає використання логічних умов TRUE/FALSE?
5. Який параметр використовується для отримання таблиць у базі даних?
6. Який параметр використовується для вивантаження даних з таблиці?
7. Який параметр дозволяє визначити поточного користувача бази даних?
8. Який параметр використовується для визначення привілеїв пористувача?
9. Який параметр дозволяє задати рівень тестування sqlmap?
10. Який параметр використовується для вибору конкретної бази даних?
11. Який параметр використовується для вибору конкретної таблиці?
12. Який параметр використовується для передачі POST-даних?
13. Який параметр дозволяє змінити HTTP-метод?
14. Який етап виконується першим при роботі sqlmap?

## Список джерел


1. Sqlmap. *TryHackMe | Cyber Security Training*. URL: https://tryhackme.com/room/sqlmap.
2. sqlmap: automatic SQL injection and database takeover tool. *sqlmap*. URL: https://sqlmap.org/.
3. SQLMap. *Hackviser*. URL: https://hackviser.com/tactics/tools/sqlmap.
4. Тестування безпеки. SQLmap. *QATestLab*. URL: https://training.qatestlab.com/blog/technical-articles/security-testing-sqlmap/.