

BÁO CÁO THỰC HÀNH

Môn học: Bảo mật Web và Ứng dụng

Tên chủ đề: Lab 1 – Tổng quan các lỗ hổng bảo mật web thường gặp

GVHD: Ngô Đức Hoàng Sơn

Nhóm: 14

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT230.O21.ANTT.1

STT	Họ và tên	MSSV	Email
1	Vũ Tuấn Sơn	21521389	21521389@gm.uit.edu.vn
2	Bùi Đức Anh Tú	21522735	21522735@gm.uit.edu.vn
3	Lê Huy Hiệp	21522067	21522067@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Nội dung	Tình trạng	Trang
1	Yêu cầu 1	90%	1 - 3
2	Yêu cầu 2	50%	3 - 5
3
Điểm tự đánh giá			?/10

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

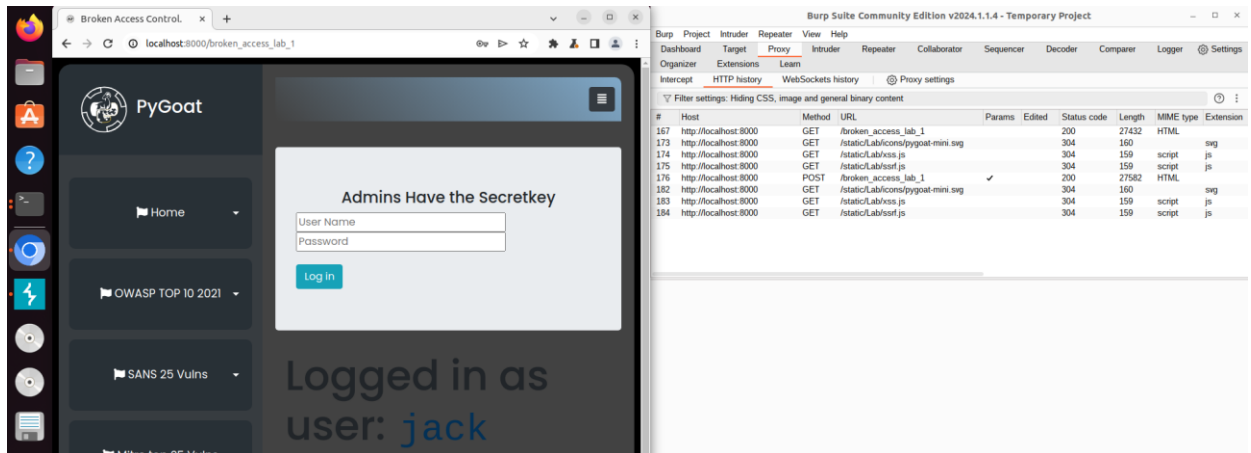
¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

Bài tập 1:

Sử dụng repeater để thực hành bài tập trên.

Có thể thấy tất cả các request và response khi đi qua Burp Suite đều được ghi lại thông tin ở phần Proxy → HTTP History:



Tại lần đăng nhập thứ 2 trở đi, Cookie của request gửi đi sẽ có thêm trường admin=0 để xác định đây không phải là admin. Ta nhấp chuột phải vào các request này và chọn Send to repeater:

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension
194	http://localhost:8000	POST	/broken_access_lab_1		✓	200	27432	HTML	
198	http://localhost:8000	GET	/static/Lab/icons/pygoat-mini.svg			304	160	image/svg+xml	svg
201	http://localhost:8000	GET	/static/Lab/ssrf.js			304	159	script/javascript	js
202	http://localhost:8000	GET	/static/Lab/xss.js			304	159	script/javascript	js

Request

```

AppletWebkit/537.36 (KHTML, like Gecko)
Chrome/121.0.6167.160 Safari/537.36
12 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
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://localhost:8000/broken_access_lab_1
18 Accept-Encoding: gzip, deflate, br
19 Accept-Language: en-US,en;q=0.9
20 Cookie: csrftoken=
FhAYryusF0cCm3zsZ2U9dPljHUogqwlRPVJPzah8obZZLOawCTiXz
15iDpOhATpA; sessionId=
lii8e4k8ecyqlu93oqjlqohppauox5n7; admin=0
21 Connection: close

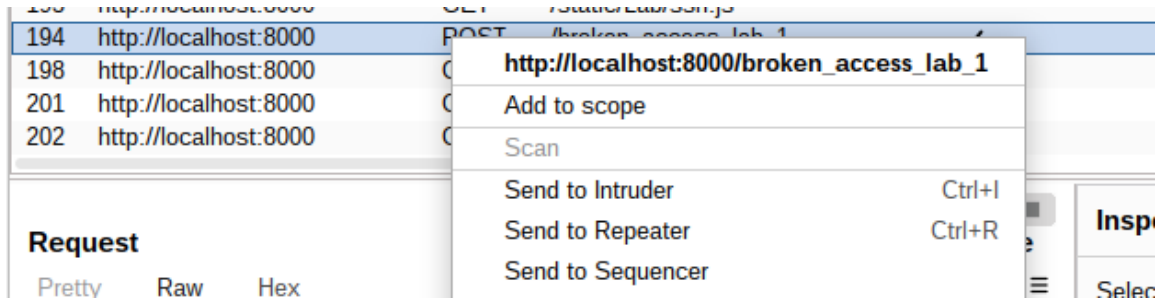
```

Response

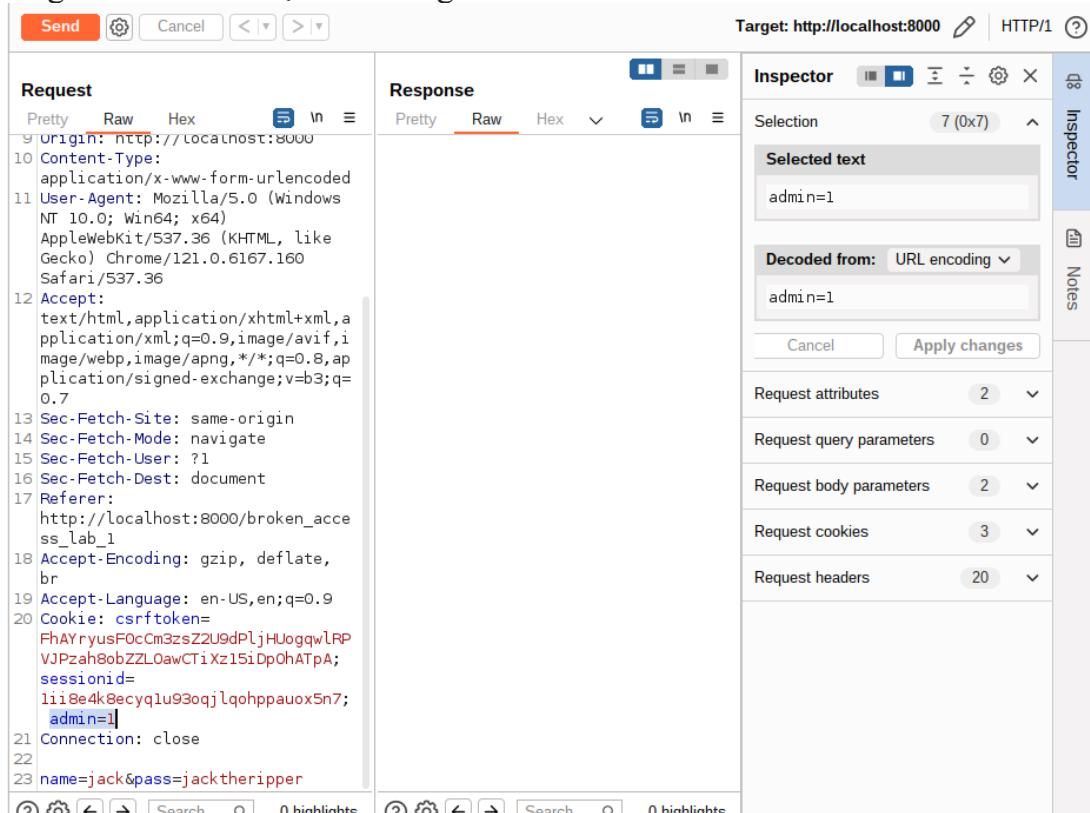
```

1 HTTP/1.1
1 200
OK
2 Server:
gunicorn
3 Date:
Sat, 09
Mar
2024
09:28:5
7 GMT
4 Connection:
close
5 Content-Type:
text/html

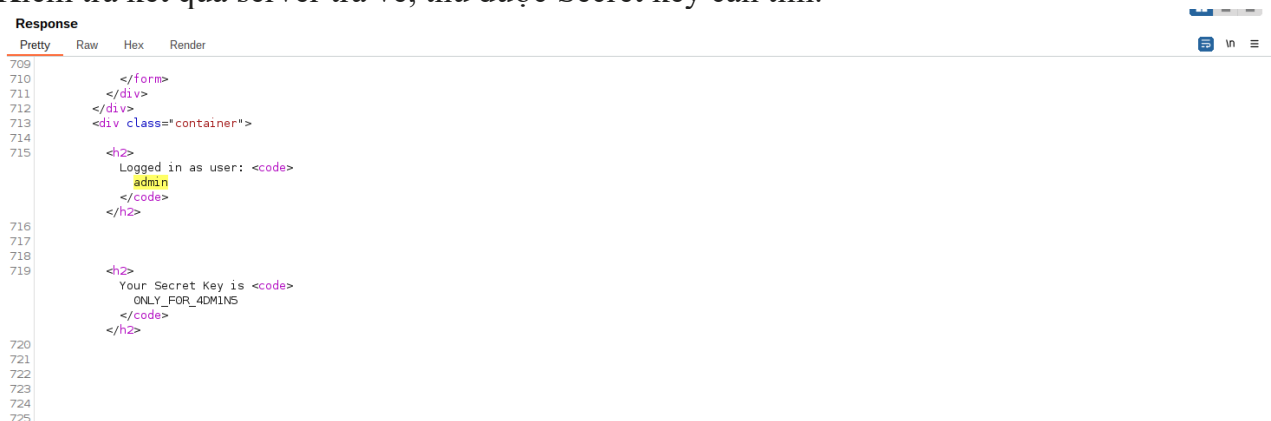
```



Sửa trường admin=1 và chọn send để gửi đi:



Kiểm tra kết quả server trả về, thu được Secret key cần tìm:



Bài tập 2:

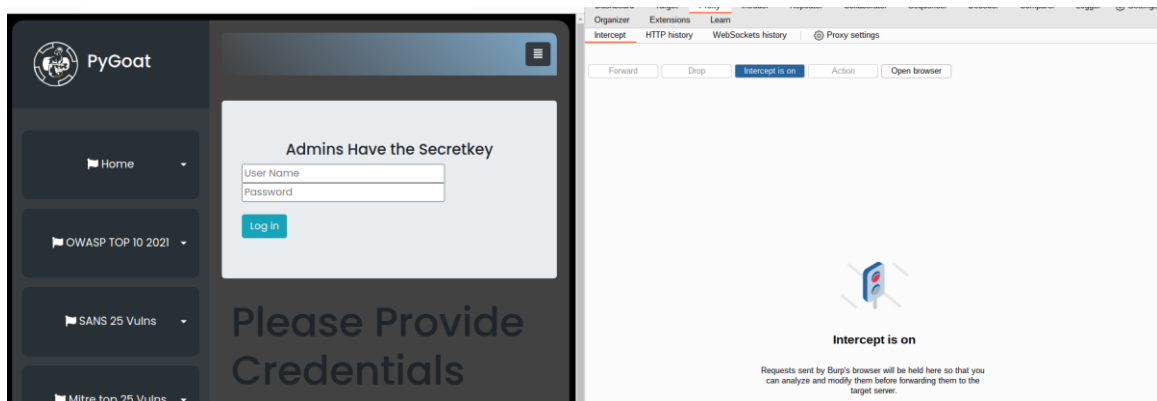
#Tiêu đề: Broken Access Control – information, data

#Mô tả lỗ hổng: Broken Access Control là một lỗ hổng bảo mật phổ biến xảy ra khi kiểm soát truy cập không được triển khai đúng, cho phép kẻ tấn công truy cập vào các tài nguyên hoặc chức năng mà họ không được phép. Trong phạm vi của bài Lab, ta sẽ sử dụng PyGoat, mục A01: Broken Access Control để thực hiện tìm hiểu về lỗ hổng này.

Tóm tắt: Ta dùng Burp Suite để chặn request gửi đi, quan sát chúng và thay đổi các trường thích hợp để đăng nhập dưới danh nghĩa là admin.

Các bước để thực hiện lại và bằng chứng:

1. Bước 1: Ta tiến hành truy cập vào bài Lab bằng trình duyệt của Burp Suite và bật chế độ Intercept để các request khi được trình duyệt này gửi đi sẽ được Burp Suite bắt lại và ta có thể đọc, thay đổi nội dung của chúng:



2. Bước 2 : Tiến hành đăng nhập với tài khoản được cho trước: jack/jacktheripper và quan sát request được gửi đi:

```
20 Cookie: csrftoken=FhAYryuSF0cCm3zsZ2U9dPljHUogqwlRPVJPzah8obZZLOawCTiXz15iDp0hATpA;
    sessionid=lii8e4k8ecyqlu93oqlqohppaux5n7
21 Connection: close
22
23 name=jack&pass=jacktheripper
```

Request này khi được gửi đi có kèm theo cookie của user và các thông tin và người dùng nhập vào để gửi đến server. Lúc này server trả về 1 response có nội dung cần chú ý sau:

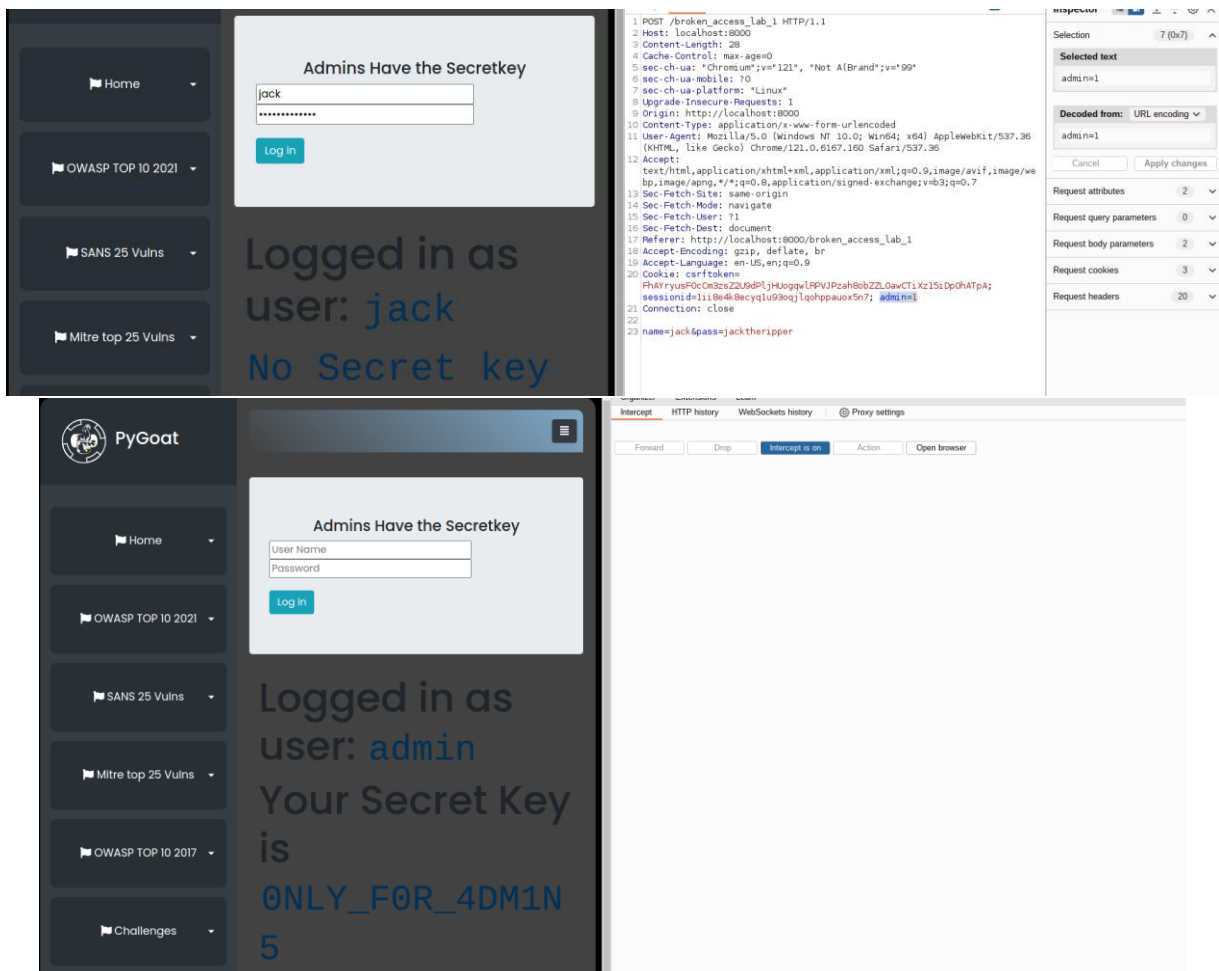
```
10 Referer-Policy: same-origin
11 Cross-Origin-Opener-Policy: same-origin
12 Set-Cookie: admin=0; expires=Sat, 09 Mar 2024 09:00:05 GMT; Max-Age=200; Path=/
13
14 <!DOCTYPE html>
```

Có vẻ như sau khi đăng nhập lần đầu tiên, server sẽ xác minh người dùng có phải admin hay không và gán thêm 1 trường admin này để xác minh người dùng.

3. Bước 3: Từ dữ kiện của bước 2, ta tiến hành đăng nhập lại 1 lần nữa với tài khoản được cho trước:

```
20 Cookie: csrfToken=FhAYrYusFOcCm3zsZ2U9dPljHUogqwlRPVJPzah8obZZLOawCTiXz15iDp0hATpA; sessionId=1ii8e4k8ecyqlu93oqjlqohppaux5n7; admin=0
21 Connection: close
22
23 name=jack&pass=jacktheripper
```

Lúc này thật sự đã có trường admin trong phần Cookie, ta thay đổi admin=1 và forward request này đi:



Như vậy ta đã có thể đăng nhập dưới danh nghĩa admin bằng tài khoản của jack.

Nhận được key là: ONLY_F0R_4DM1N5

#Mức độ ảnh hưởng của lỗ hổng: CAO

#Khuyến cáo khắc phục: Cần thay đổi cơ chế xác thực người dùng, thay vì chỉ dùng 1 trường cookie để xác thực thì cần xác thực xem tên đăng nhập cũng như mật khẩu có tương đồng với admin=1 tại cookie hay không.

Bài tập 3:

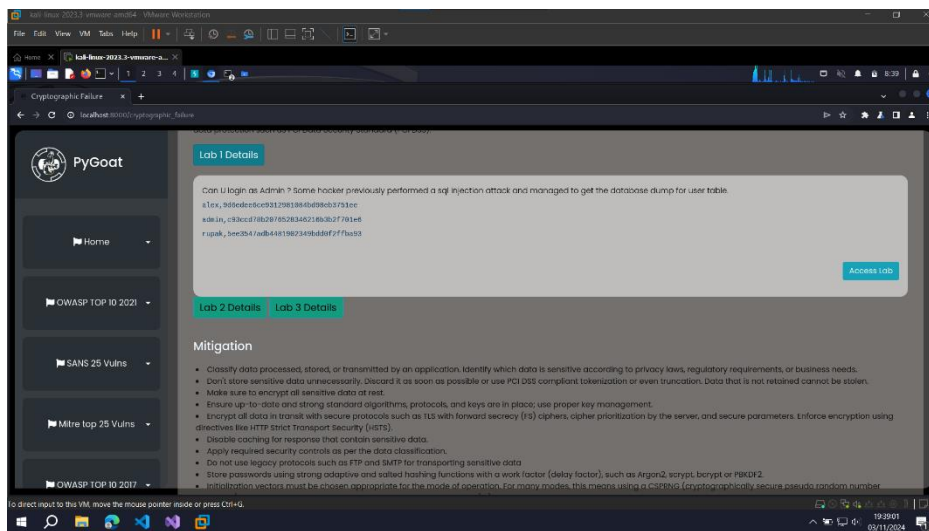
#Tiêu đề: Cryptographic Failures

#Mô tả lỗ hổng: nơi kẻ tấn công thường nhắm mục tiêu vào dữ liệu nhạy cảm khi bạn không bảo vệ chúng đúng cách Trong phạm vi của bài Lab, ta sẽ sử dụng PyGoat, mục A02:2021 – Cryptographic Failures để thực hiện tìm hiểu về lỗ hổng này.

Tóm tắt: kẻ tấn công trước đó đã thực thi thành công lỗi SQL injection và lấy được bảng thông tin đăng nhập của người dùng với username và 1 đoạn chuỗi ký tự. Ta sẽ dùng các trang web online để tìm ra mật khẩu đã được hash bằng MD5

Các bước để thực hiện lại và bằng chứng:

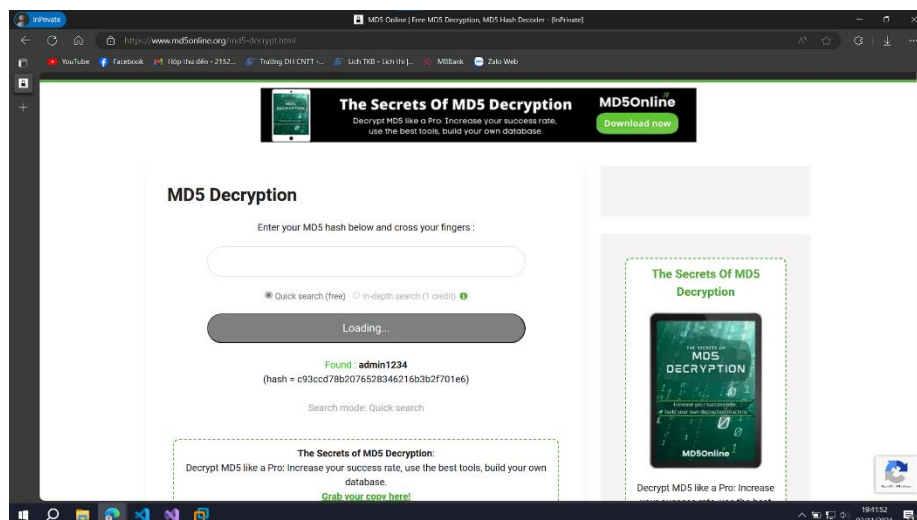
1. Bước 1



Ở đây chúng ta thấy được thông tin đăng nhập của user bao gồm username và password đã được băm

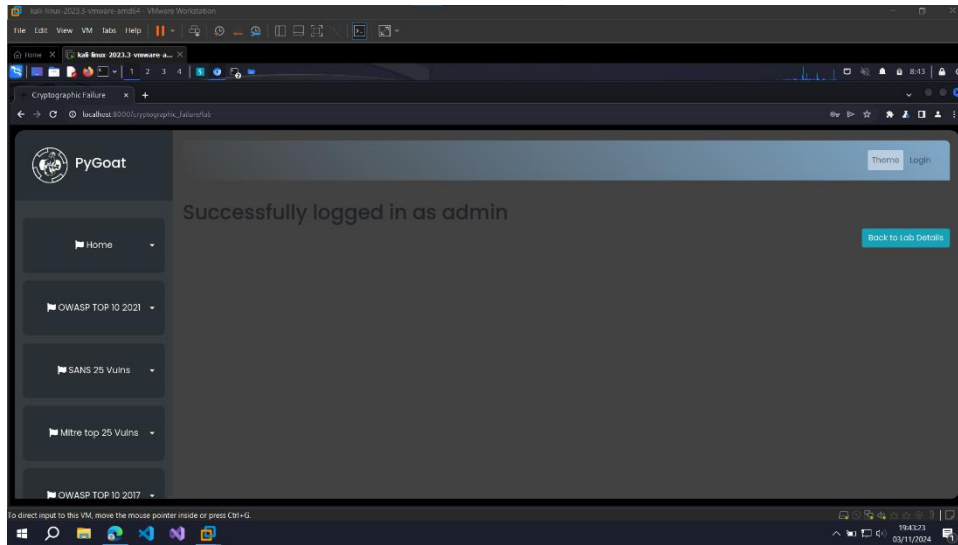
2. Bước 2

Chúng ta sẽ dùng web online decrypt đoạn mã hash md5 để lấy mật khẩu



3. Bước 3

Đăng nhập bằng username và password tìm ra



Thành công vào được tài khoản admin

Tài liệu hỗ trợ và tham khảo: [A02 Cryptographic Failures - OWASP Top 10:2021](#)

#Mức độ ảnh hưởng của lỗ hổng: CAO

Lỗ hổng mã hóa có thể dẫn đến việc tiết lộ dữ liệu nhạy cảm, ảnh hưởng đến tính bảo mật của hệ thống.

#Khuyến cáo khắc phục:

- Sử dụng thuật toán mã hóa mạnh mẽ.
- Quản lý khóa đúng cách.
- Áp dụng mã hóa đúng cách cho dữ liệu nhạy cảm.

Bài tập 4:

#Tiêu đề: SQL injection

#Mô tả lỗ hổng: một loại tấn công injection giúp thực thi các câu lệnh SQL độc hại, có thể ảnh hưởng đến bất kỳ trang web hoặc ứng dụng web nào sử dụng cơ sở dữ liệu SQL. Trong phạm vi của bài Lab, ta sẽ sử dụng PyGoat, mục A03:2021 – SQL Injection để thực hiện tìm hiểu về lỗ hổng này

Tóm tắt: Lỗi tiêm SQL có thể được nhận ra thông qua một vài thủ thuật như tiêm một ký tự ' vào bất kỳ trường nào. Nếu kết quả là một lỗi SQL thì lỗi tiêm SQL có thể đã xảy ra

Các bước để thực hiện lại và bằng chứng:

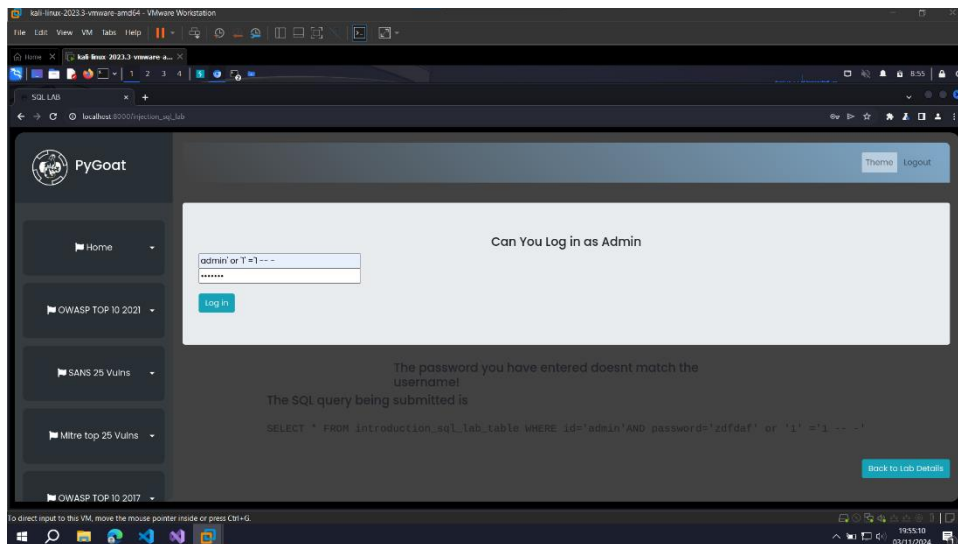
1. Bước 1

Theo như hướng dẫn thì câu truy vấn tài khoản sẽ là

"SELECT * FROM introduction_login WHERE user='"+name+"'AND password='"+password+"'"

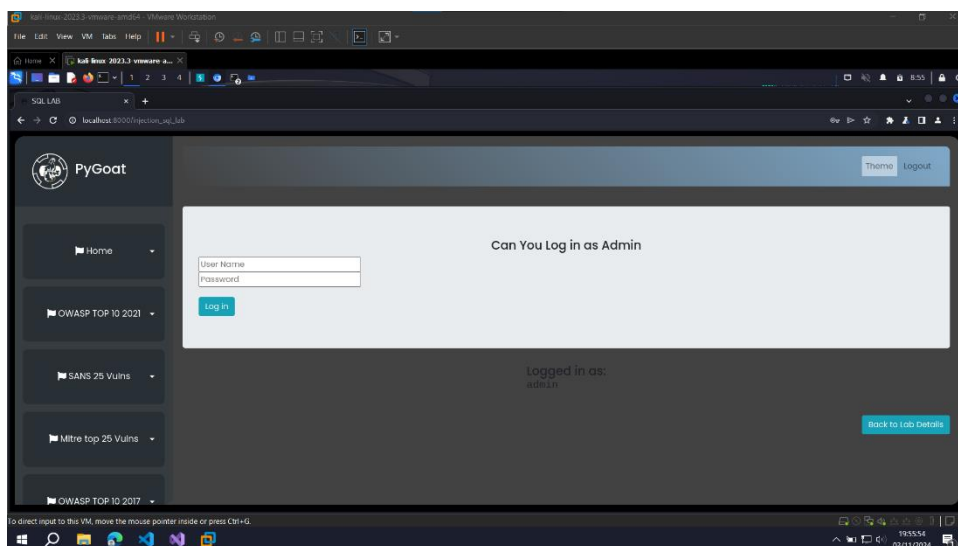
2. Bước 2

Chúng ta tiến hành tiêm sql vào bằng cách thêm or '1'='1' để câu truy vấn luôn đúng



Ở đây chúng ta thêm vào ở dòng user name

3. Bước 3



Thành công vào được tài khoản admin

Tài liệu hỗ trợ và tham khảo:

[What is SQL Injection \(SQLi\) and How to Prevent Attacks \(acunetix.com\)](https://www.acunetix.com/what-is-sql-injection/)

[A01 Broken Access Control - OWASP Top 10:2021](#)

#Mức độ ảnh hưởng của lỗ hổng: CAO

Kẻ tấn công có thể sử dụng lỗ hổng SQL Injection để vượt qua các biện pháp bảo mật ứng dụng. Họ có thể đi xung quanh xác thực và ủy quyền của một trang web hoặc ứng dụng web và truy xuất nội dung của toàn bộ cơ sở dữ liệu SQL. Tội phạm có thể sử dụng nó để truy cập trái phép vào dữ liệu nhạy cảm.

#Khuyến cáo khắc phục: Cách ngăn chặn các cuộc tấn công SQL Injection

Xác thực đầu vào và các truy vấn tham số bao gồm các câu lệnh đã chuẩn bị

Xác minh và kiểm tra dữ liệu đầu vào từ người dùng

Đặt quyền truy cập cơ sở dữ liệu sao cho người dùng chỉ có quyền truy cập vào những bảng và trường cần thiết.

Kiểm tra bảo mật thường xuyên

Bài tập 5:

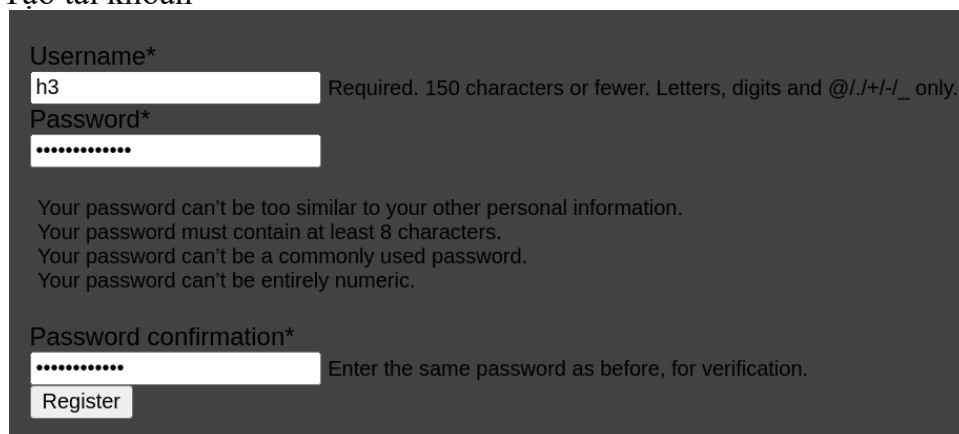
Tiêu đề: Insecure Design - ảnh hưởng đến tài khoản và thông tin

Mô tả lỗ hổng: Là một danh mục rộng lớn bao gồm các lỗ hổng khác nhau được diễn giải như “thiếu hoặc không hiệu quả của kiểm soát thiết kế” từ đó kẻ xấu có thể lợi dụng. Thiết kế không an toàn và các khuyết điểm trong triển khai có nguyên nhân và biện pháp khắc phục khác nhau.

#Tóm tắt: Lỗ hổng Insecure Design của bài tập này là không có cơ chế đảm bảo tính duy nhất của người dùng. Tài khoản có thể được tạo một cách liên tục từ 1 người dùng.

Các bước thực hiện và để lại bằng chứng:

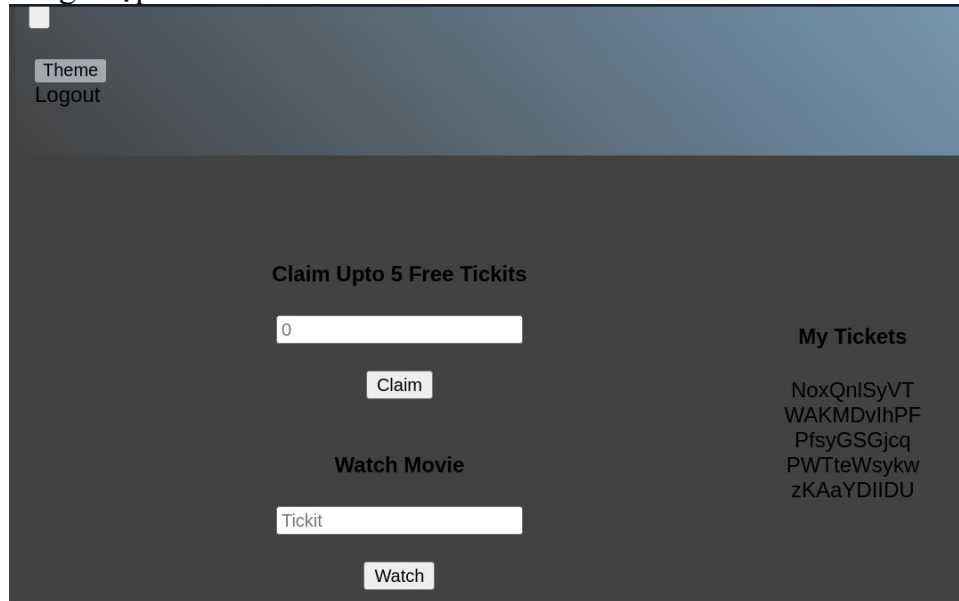
1. Bước 1: Tạo tài khoản



The screenshot shows a registration form with the following fields and text:

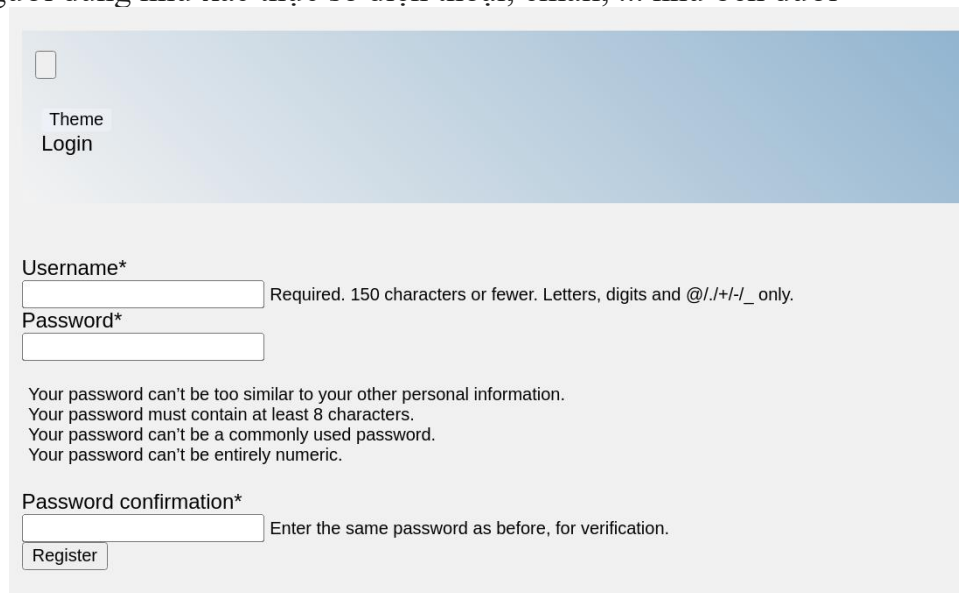
- Username***: Input field containing 'h3'. To the right, it says: "Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only."
- Password***: Input field with masked characters (dots).
- Below the password field, there are four lines of feedback text:
 - Your password can't be too similar to your other personal information.
 - Your password must contain at least 8 characters.
 - Your password can't be a commonly used password.
 - Your password can't be entirely numeric.
- Password confirmation***: Input field with masked characters. To the right, it says: "Enter the same password as before, for verification."
- At the bottom left is a **Register** button.

2. Bước 2: Đăng nhập vào và tiến hành claim 5 vé



The screenshot shows a web application interface with a dark theme. At the top left, there are links for 'Theme' and 'Logout'. The main content area has a heading 'Claim Upto 5 Free Tickits'. Below this, there is a text input field containing the number '0', a 'Claim' button, and a 'Watch Movie' button. Below the 'Watch Movie' button is another text input field labeled 'Tickit' and a 'Watch' button. On the right side, there is a section titled 'My Tickets' which lists five alphanumeric strings: NoxQnISyVT, WAKMDvIhPF, PfsyGSGjcq, PWTteWsykw, and zKAaYDIIDU.

3. Bước 3: Trong quá trình tạo ta thấy việc tạo tài khoản không có cơ chế xác thực tính duy nhất của người dùng như xác thực số điện thoại, email, ... như bên dưới

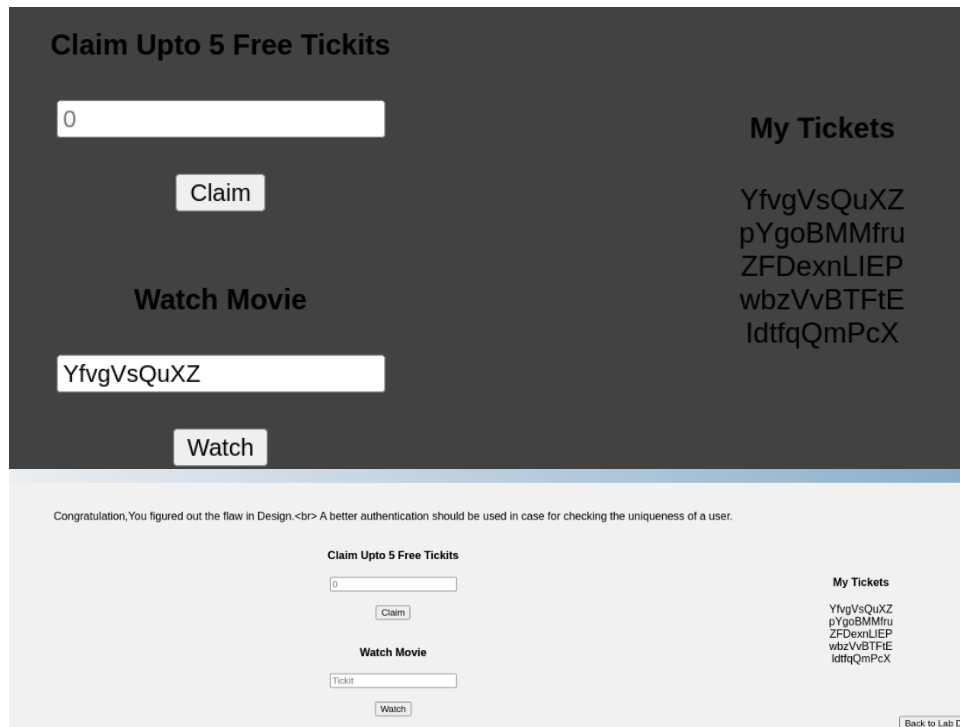


The screenshot shows a web application interface with a light blue header. At the top left, there are links for 'Theme' and 'Login'. The main content area has a 'Register' button. Below the button, there are two text input fields: 'Username*' and 'Password*'. The 'Username*' field has a hint: 'Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.' The 'Password*' field has a hint: 'Your password can't be too similar to your other personal information. Your password must contain at least 8 characters. Your password can't be a commonly used password. Your password can't be entirely numeric.' Below the 'Password*' field is a 'Password confirmation*' field with a hint: 'Enter the same password as before, for verification.' At the bottom, there is a 'Register' button.

=> 1 người dùng có thể tạo nhiều tài khoản để lấy hết vé

Khi nhấn claim thấy còn lại 55 vé nghĩa là cần thêm 11 tài khoản (Mỗi cái 5 vé). Tiến hành tạo thêm 11 tài khoản.

4. Bước 4: Xem phim khi đã lấy hết số vé



Tài liệu hỗ trợ và tham khảo:

[A04 Insecure Design - OWASP Top 10:2021](#)

[Insecure Design Vulnerability: Explanation and Examples | QAwerk](#)

Mức độ ảnh hưởng của lỗ hổng: Rất lớn

Thiết kế không an toàn có thể cho phép kẻ tấn công:

- Vượt qua các cơ chế xác thực được sử dụng bởi một ứng dụng web.
- Sửa đổi một số tham số URL thông qua các kênh không được ủy quyền.
- Truy cập vào các hệ thống để khai thác thông tin nhạy cảm.
- Giả định các tài khoản người dùng hợp lệ và đạt được truy cập không được ủy quyền vào các nguồn tài nguyên được bảo vệ bằng mật khẩu để khai thác hệ thống thêm nữa.
- Lấy quyền truy cập vào bất kỳ môi trường nào và mở rộng phạm vi của cuộc tấn công sang các môi trường khác.
- Giả mạo hệ thống mục tiêu để quá tải máy chủ và mạng với nhiều yêu cầu để làm sập chúng.
- Gửi các truy vấn được chỉ định để trích xuất thông tin về các lỗ hổng hệ thống có thể tạo điều kiện cho một cuộc tấn công.
- Tiếp quản tài khoản hoàn toàn.
- Thực hiện các cuộc tấn công khác như cross-site scripting, SQLi, LDAP injection, cross-site request forgery và path transversal.

Khuyến cáo khắc phục: Thêm cơ chế xác thực để định danh người dùng duy nhất (số điện thoại, gmail...). Sử dụng phương thức truyền tin HTTPS thay vì HTTP để tránh để lộ thông tin tài khoản người dùng khi truyền qua mạng. Thiết lập mô hình các mối đe dọa. Áp dụng các cơ chế mã cho dữ liệu khi lưu trữ và khi truyền. Áp dụng chiến lược phân loại dữ liệu (dữ liệu công khai, riêng tư, dữ liệu bị hạn chế, dữ liệu có rủi ro cao) nếu ứng dụng có xử lý dữ liệu nhạy cảm.

Bài tập 6:

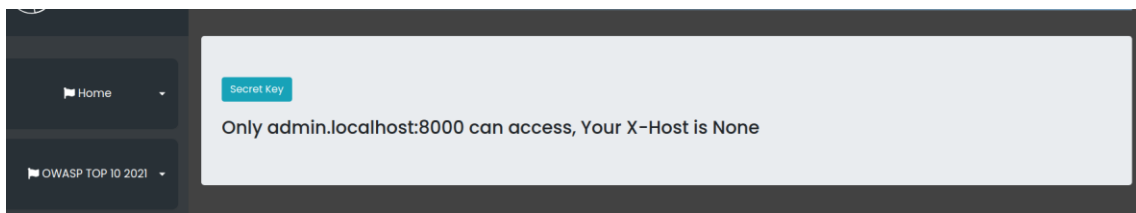
#Tiêu đề: Security Misconfiguration – Information, data

#Mô tả lỗ hổng: Security Misconfiguration là lỗ hổng xảy ra khi các cài đặt, cấu hình và môi trường không được thiết lập đúng cách, tạo ra điểm yếu và cho phép kẻ tấn công tiến hành khai thác.

Tóm tắt: Dựa vào thông báo “X-host is None” ta sẽ thêm thông tin trường này trong request đến server dưới danh nghĩa là admin.localhost:8000 để lấy secret.

Các bước để thực hiện lại và bằng chứng:

1. Bước 1: Thử click button để get Secret key nhưng không được:

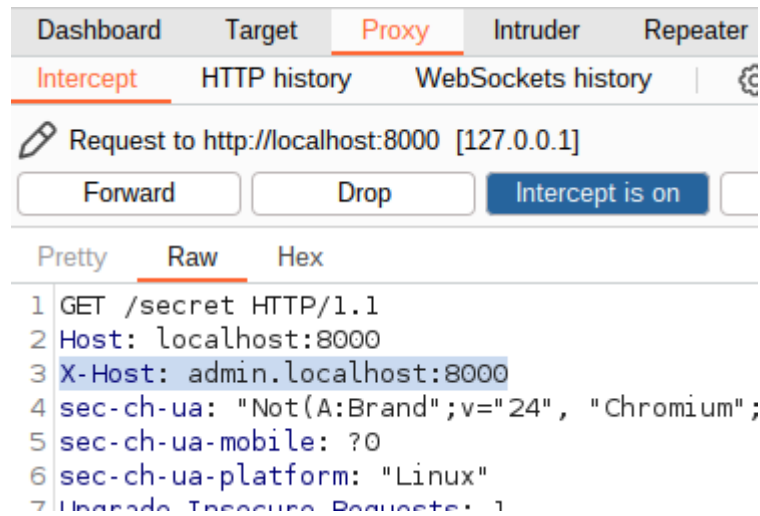


Nhận thấy thông báo X-Host is None. Có vẻ như trường này chưa có thông tin trong request. Kiểm tra để xác minh lại:

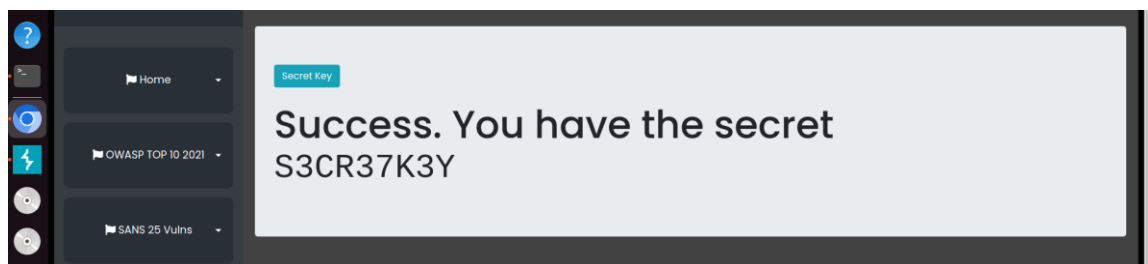
Request

```
Pretty  Raw  Hex
1 GET /secret HTTP/1.1
2 Host: localhost:8000
3 sec-ch-ua: "Not(A:Brand";v="24", "Chromi
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: "Linux"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0;
  Gecko) Chrome/122.0.6261.95 Safari/537.36
8 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
9 Sec-Fetch-Site: same-origin
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Referer: http://localhost:8000/sec_mis_l
14 Accept-Encoding: gzip, deflate, br
15 Accept-Language: en-US,en;q=0.9
16 Cookie: csrfToken=FhAYryusF0cCm3zsZ2U9dP
  sessionId=lii8e4k8ecyqlu93oqjlqohppauox5r
17 Connection: close
18
19
```

2. Bước 2 : Xác minh được cấu hình bị thiếu, tiến hành chặn request gửi đi và thêm trường “X-Host: admin.localhost:8000” vào request:



3. Bước 3: Forward request đi và nhận được kết quả:



Như vậy, secret cần tìm là: S3CR37K3Y

Tài liệu hỗ trợ và tham khảo:

#Mức độ ảnh hưởng của lỗ hổng: CAO

#Khuyến cáo khắc phục: Khi cấu hình không nên có các thành phần dư hoặc thiếu khi truy vấn, lọc repeater vì có thể tạo lỗ hổng cho hacker lợi dụng. Ngoài ra, khi tiến hành thiết kế hệ thống cần phải có 1 quy trình đủ tốt để phát triển ứng dụng.