Đại học Quốc gia Thành phố Hồ Chí Minh
Trường Đại học Công nghệ Thông tin (UIT)

BÁO CÁO THỰC HÀNH

Môn học: Bảo mật web và ứng dụng

Lab 1: Tổng quan các lỗ hổng bảo mật web thường gặp

Ngày báo cáo: 20/03/2024

1. **THÔNG TIN CHUNG:**

Lớp: NT213.O22.ATCL.2

STT	Họ và tên	MSSV	Email
1	Hoàng Gia Bảo	21521848	21521848@gm.uit.edu.vn
2	Phạm Hoàng Phúc	21521295	21521295@gm.uit.edu.vn
3	Nguyễn Đại Nghĩa	21521182	21521182@gm.uit.edu.vn

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

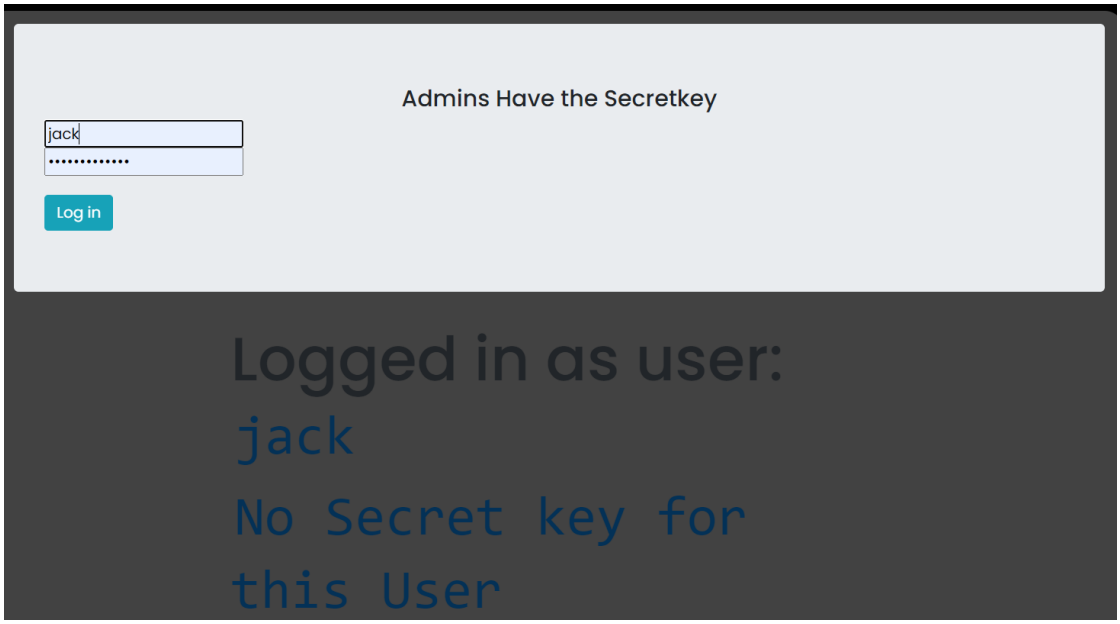
BÁO CÁO CHI TIẾT

Câu 1-2: Broken access control - Data

Mô tả: Sử dụng repeater để gửi request lên trang web có thể làm việc log in bị lỗi, dẫn đến lộ Secret key dù đăng nhập bằng tài khoản user

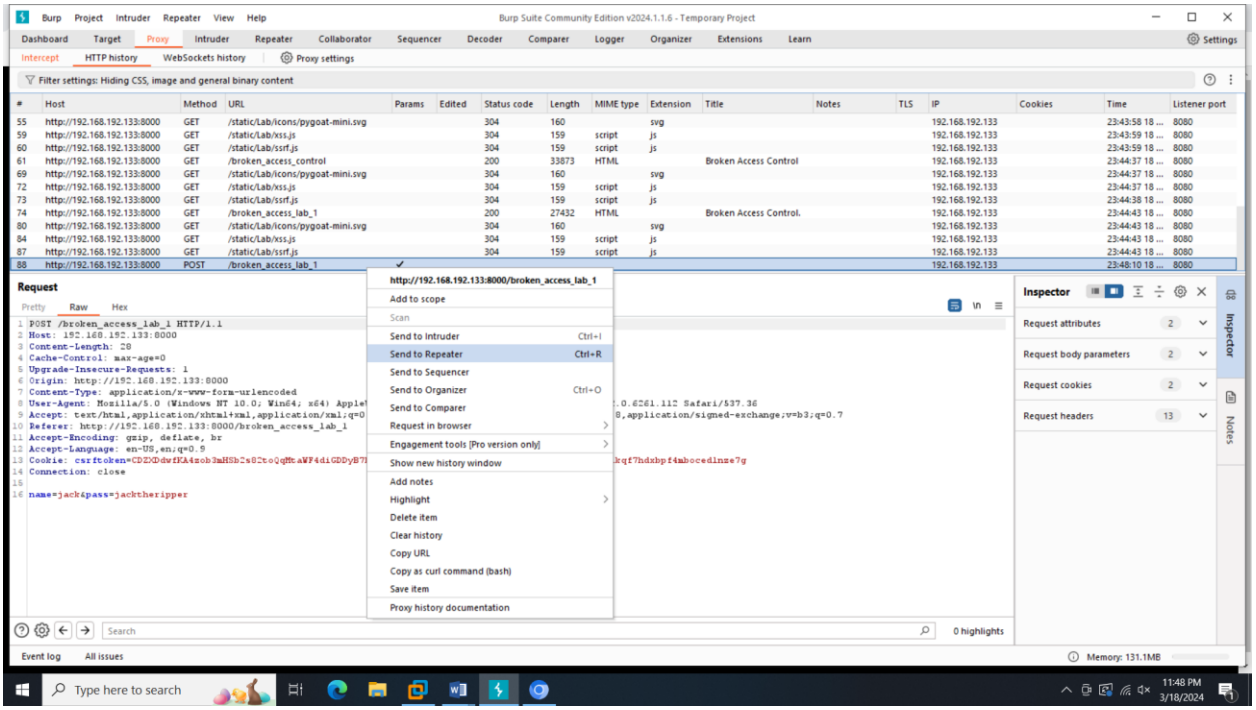
Các bước thực hiện:

Đầu tiên, thử log in vào trang web, vì là tài khoản user nên không có secret key

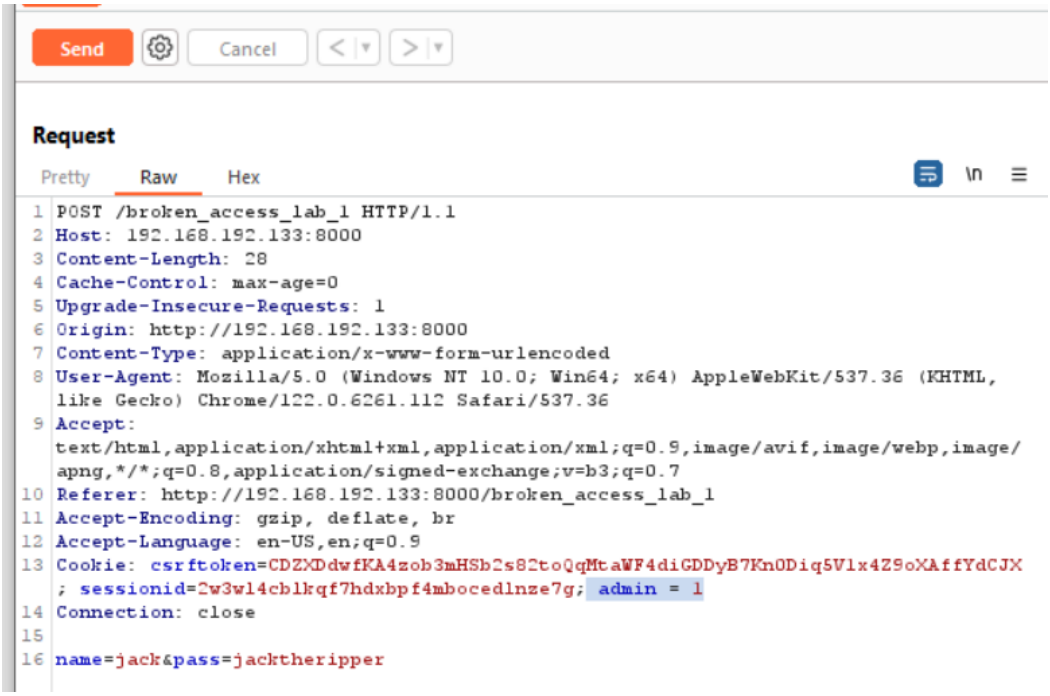


Mở ứng dụng Burp Suite, Chọn **Proxy -> Intercept is on** để bắt và chỉnh sửa gói tin request và tiến hành đăng nhập lại.

Vào HTTP history, chọn gói tin request vừa gửi và send to repeater



Tại mục Repeater, thêm trường **admin =1** vào cookie và gửi lại lên server



Send Cancel

Request

Pretty Raw Hex

```
1  POST /broken_access_lab_1 HTTP/1.1
2  Host: 192.168.192.133:8000
3  Content-Length: 28
4  Cache-Control: max-age=0
5  Upgrade-Insecure-Requests: 1
6  Origin: http://192.168.192.133:8000
7  Content-Type: application/x-www-form-urlencoded
8  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
   like Gecko) Chrome/122.0.6261.112 Safari/537.36
9  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
10 Referer: http://192.168.192.133:8000/broken_access_lab_1
11 Accept-Encoding: gzip, deflate, br
12 Accept-Language: en-US,en;q=0.9
13 Cookie: csrftoken=CDZXDdwfKA4zob3mHSb2s82toQqMtaWF4diGDDyB7Kn0Diq5V1x4Z9oXAffYdCJX
   ; sessionid=2w3wl4cblkqf7hdxbpf4mbocedlnze7g; admin = 1
14 Connection: close
15
16 name=jack&pass=jacktheripper
```

Tìm được secret key là **0NLY_F0R_4DM1N5**

Mức độ ảnh hưởng: Rất cao. Vì khi có thể truy cập bằng tài khoản admin, kẻ tấn công sẽ có toàn quyền thao tác với data (xem, thêm, xóa, chỉnh sửa).

Khuyến cáo khắc phục:

Nên có cơ chế kiểm tra truy cập ở phía Server để đảm bảo rằng Server thực hiện kiểm tra truy cập hợp lệ đối với mọi request, không chỉ dựa trên thông tin có sẵn trong cookie. Sử dụng các phương pháp xác thực bảo mật như JWT hoặc session được lưu trữ an toàn để xác thực và kiểm tra quyền truy cập.

Đừng quá tin tưởng vào thông tin được gửi từ Client. Luôn xác thực và xác minh dữ liệu được gửi từ Client trước khi tin tưởng và sử dụng nó.

Tham khảo:

https://owasp.org/Top10/A01_2021-Broken_Access_Control/

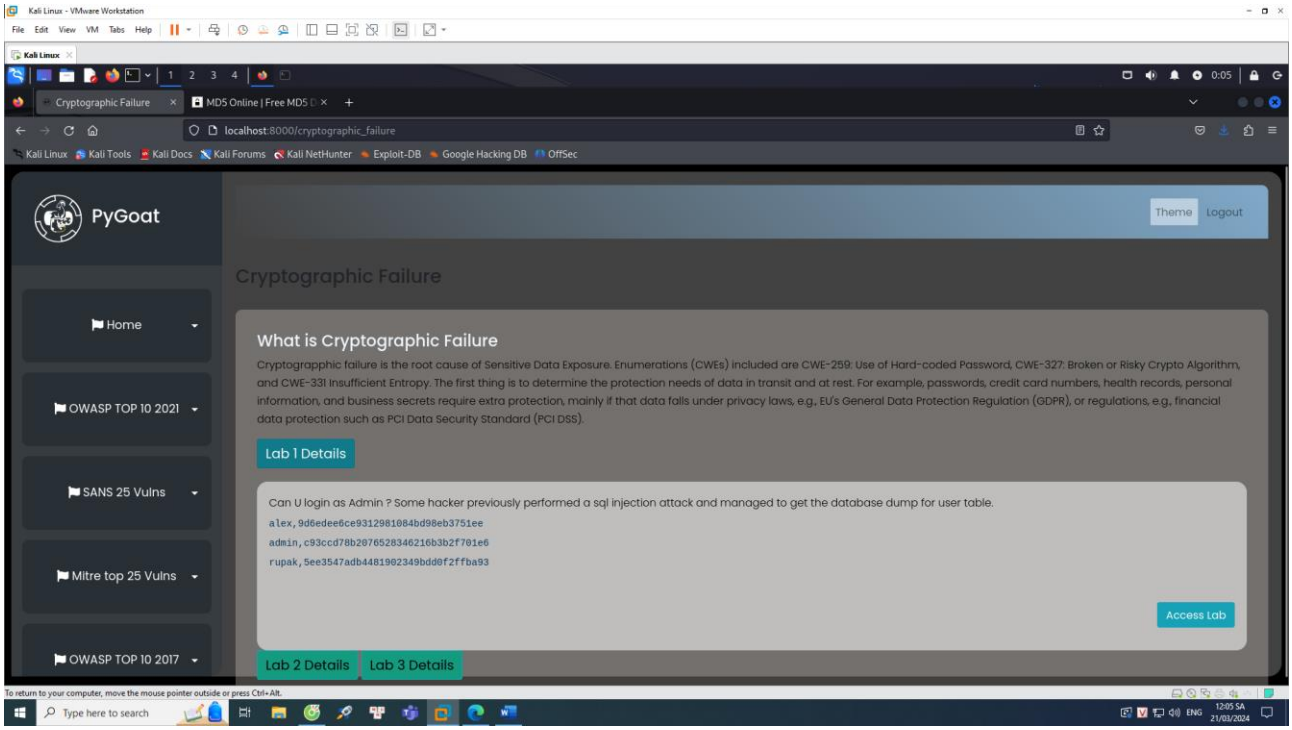
Câu 3: Cryptographic Failures

Tiêu đề: Cryptographic Failures – Password

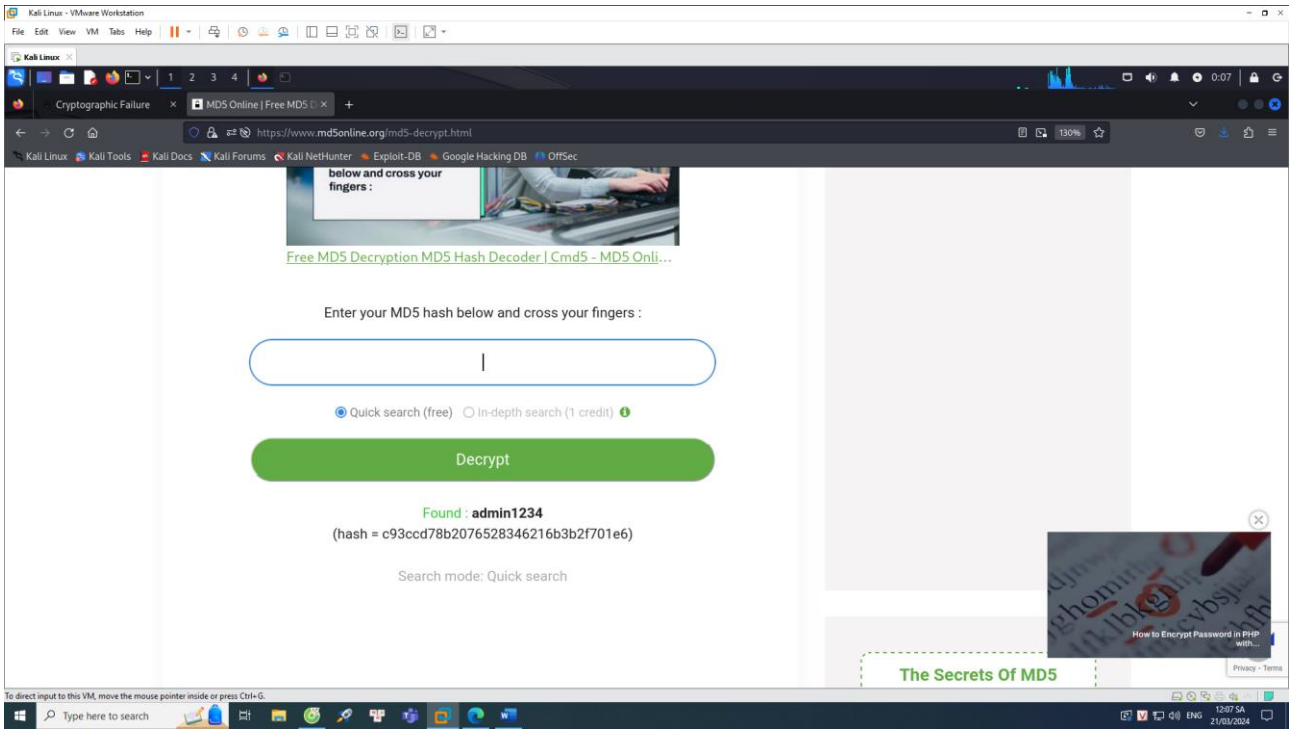
Mô tả lỗ hổng: Trong ứng dụng web này, mật khẩu đã được hash bằng MD5, đây là một hàm băm đã lỗi thời, không được khuyến khích sử dụng nữa do các yếu điểm bảo mật của nó đã được phơi bày. Sau khi mật khẩu đã được hash bị lộ thông qua sql injection thì có thể dễ dàng tìm lại được bản rõ của pass đã hash bằng các trang web online trên mạng.

Các bước thực hiện:

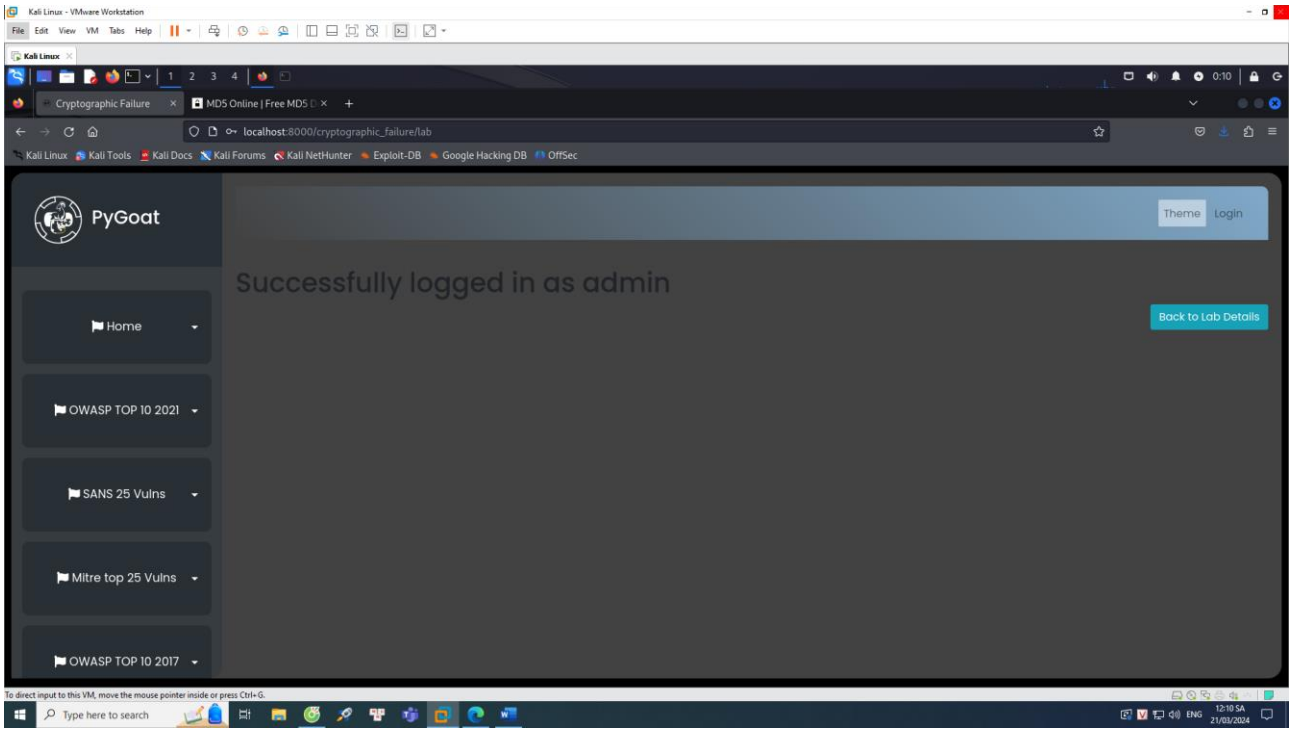
Để có thể đăng nhập với tài khoản admin, thì em sẽ tiến hành decrypt password của admin:



Kết quả nhận được sau khi decrypt:



Có thể thấy được mật khẩu rõ của tài khoản admin là admin1234, em sẽ tiến hành lấy pass này để đăng nhập thử, kết quả như sau:



Vậy là với password là admin1234 thì đã đăng nhập thành công với tài khoản admin.

Mức độ ảnh hưởng của lỗ hổng: Cao

Khuyến cáo khắc phục:

Chuyển sang sử dụng các thuật toán hash an toàn và hiện đại như bcrypt, Argon2, hoặc SHA-256. Các thuật toán này khó bị tấn công hơn và thường đi kèm với các cơ chế làm chậm quá trình hash để làm giảm hiệu quả của các cuộc tấn công brute-force.

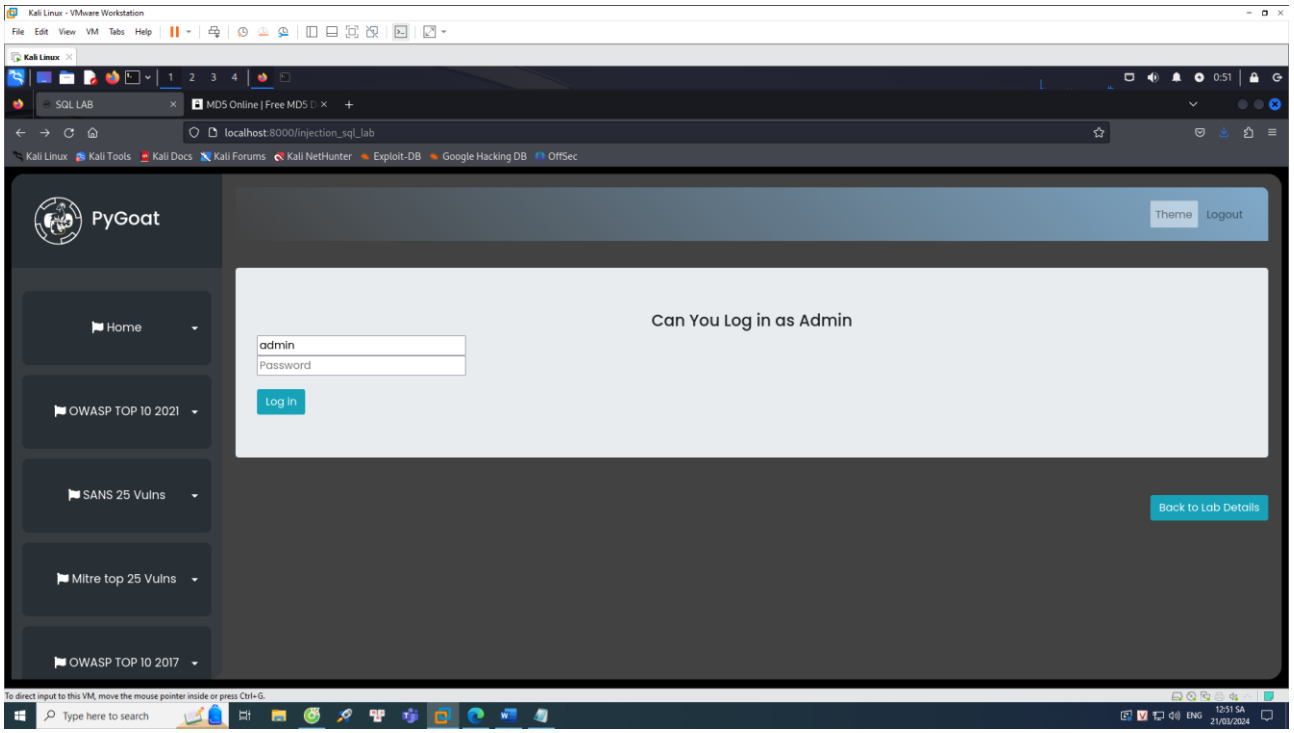
Thêm một chuỗi ký tự ngẫu nhiên, được gọi là 'salt', vào mật khẩu trước khi thực hiện hash. Điều này giúp mỗi hash là duy nhất ngay cả khi hai người dùng có cùng một mật khẩu, và làm giảm hiệu quả của rainbow tables.

Câu 4: Injection

Tiêu đề: SQL Injection – logged in as admin

Mô tả lỗ hổng: Ở ứng dụng web này, người lập trình viên đã không design kĩ, dẫn đến việc bất kì những gì được nhập ở user name và password đều được tiến hành đưa vào xử lí để xác thực mà không kiểm tra xem hay xác minh liệu input đó có thực sự là một input thích hợp trước khi tiến hành xử lí hay không. Cụ thể hơn, ở đây chính là đưa vào một đoạn chuỗi ở ô password để khi nó được đưa thẳng vào query thì sẽ thành một câu truy vấn luôn đúng, từ đó có thể đánh lừa rằng đây là một password hợp lệ và có thể đăng nhập được dưới quyền của admin.

Các bước thực hiện:



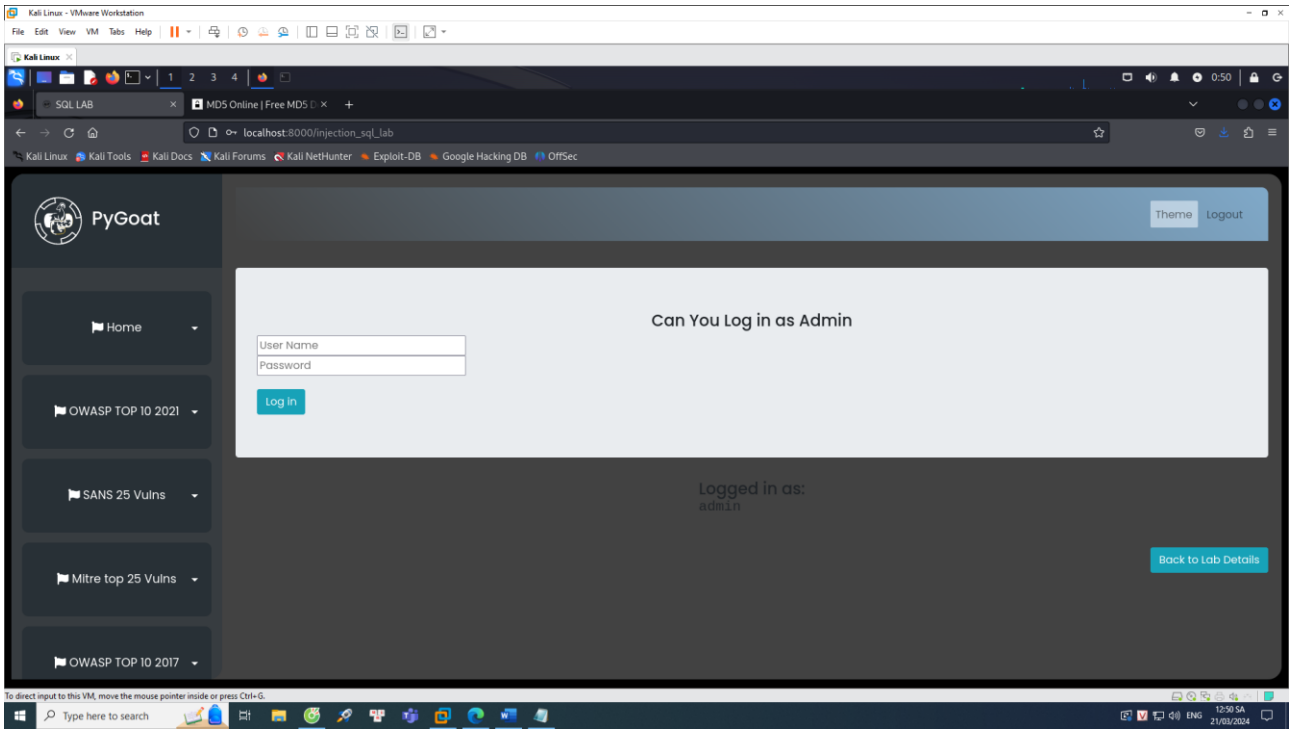
Ở đây em sẽ tiến hành nhập username là admin để đăng nhập vào tài khoản admin, và pass sẽ là chuỗi “anything' OR '1' ='1”.

Để giải thích về input trên thì trước hết cần xem qua câu lệnh truy vấn dùng để so sánh tài khoản, mật khẩu trong database: "SELECT * FROM introduction_login WHERE user='"+name+"'AND password='"+password+"'"

Qua đó có thể thấy được rằng khi nhận được pass trên thì câu lệnh truy vấn sẽ thành: “SELECT * FROM introduction_login WHERE user='admin' AND password='anything' OR '1' ='1'”.

Từ đó sẽ dễ dàng đăng nhập vào với tài khoản admin do '1'='1' sẽ luôn luôn trả kết quả là true.

Kết quả nhận được:



Mức độ ảnh hưởng của lỗ hổng: Cao

Khuyến cáo khắc phục:

Sử dụng Prepared Statements (còn gọi là Parameterized Queries): Đây là một trong những cách hiệu quả nhất để tránh SQL injection. Bằng cách sử dụng các câu lệnh SQL đã được chuẩn bị, chúng ta có thể tách rời dữ liệu đầu vào từ câu lệnh SQL, giúp tránh được việc kẻ tấn công chèn mã độc hại.

Kiểm tra, làm sạch và xác minh tính hợp lệ của tất cả dữ liệu đầu vào từ người dùng trước khi xử lý. Điều này bao gồm việc kiểm tra độ dài, định dạng, và loại dữ liệu.

Sử dụng Web Application Firewall (WAF).

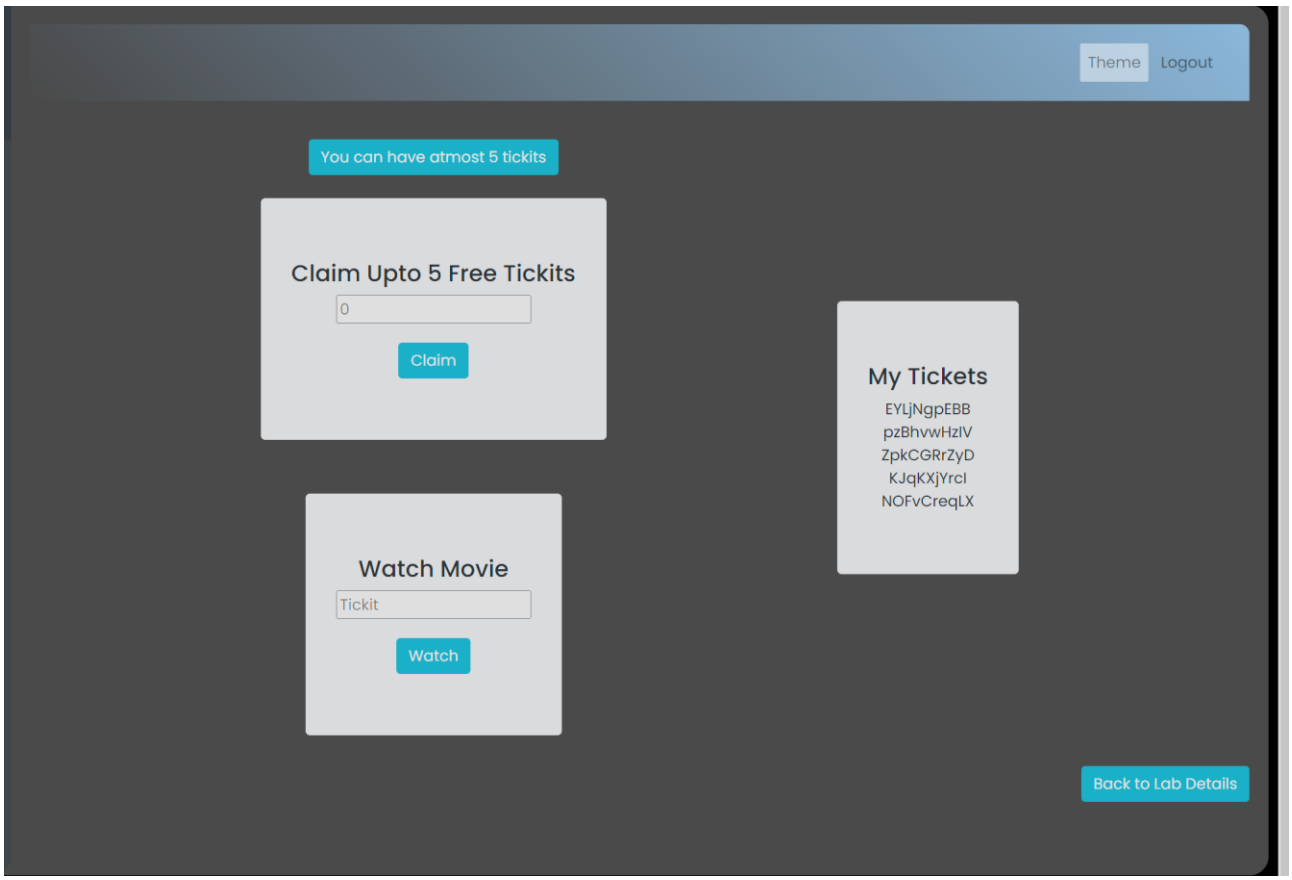
Giới hạn quyền truy cập của người dùng cơ sở dữ liệu, đảm bảo rằng tài khoản cơ sở dữ liệu được ứng dụng web sử dụng chỉ có đủ quyền cần thiết để thực hiện công việc. Không sử dụng tài khoản quản trị viên cơ sở dữ liệu cho các hoạt động bình thường.

Câu 5: Insecure Design

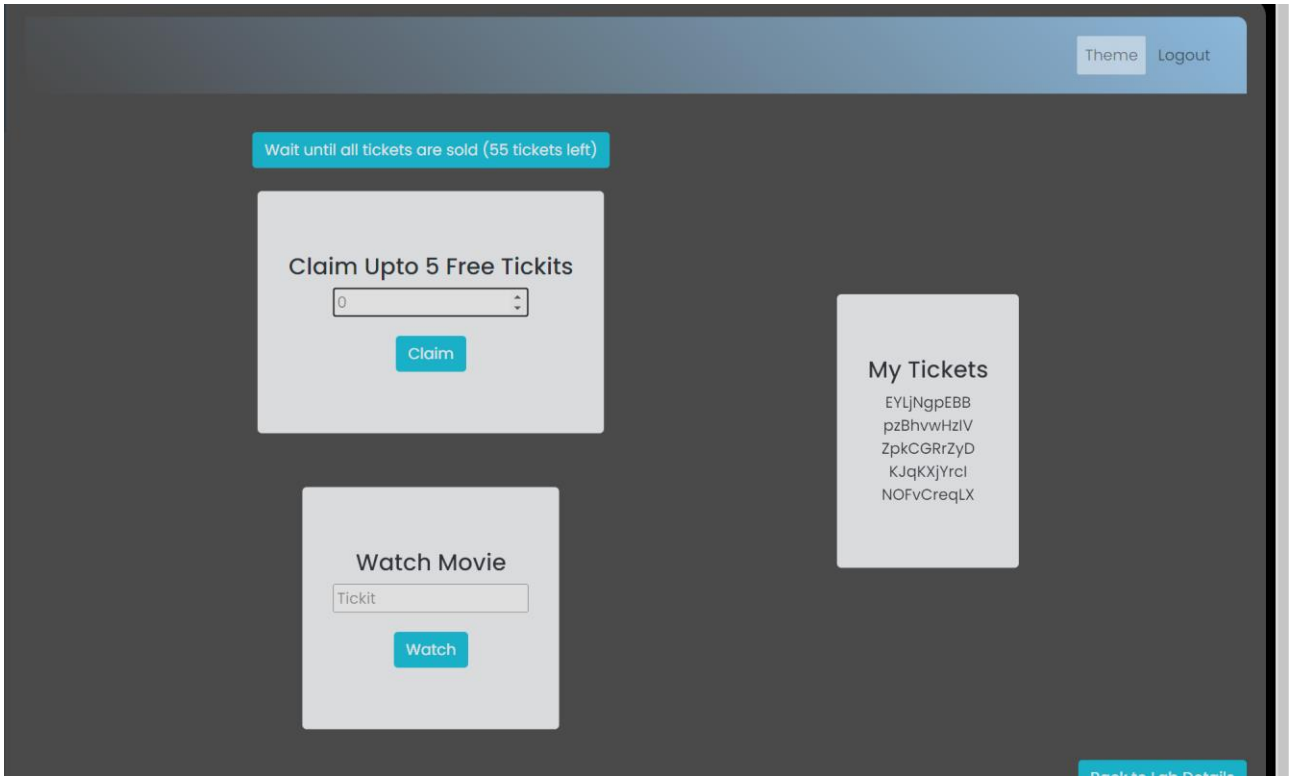
Mô tả : Lấy 5 vé miễn phí

Các bước thực hiện :

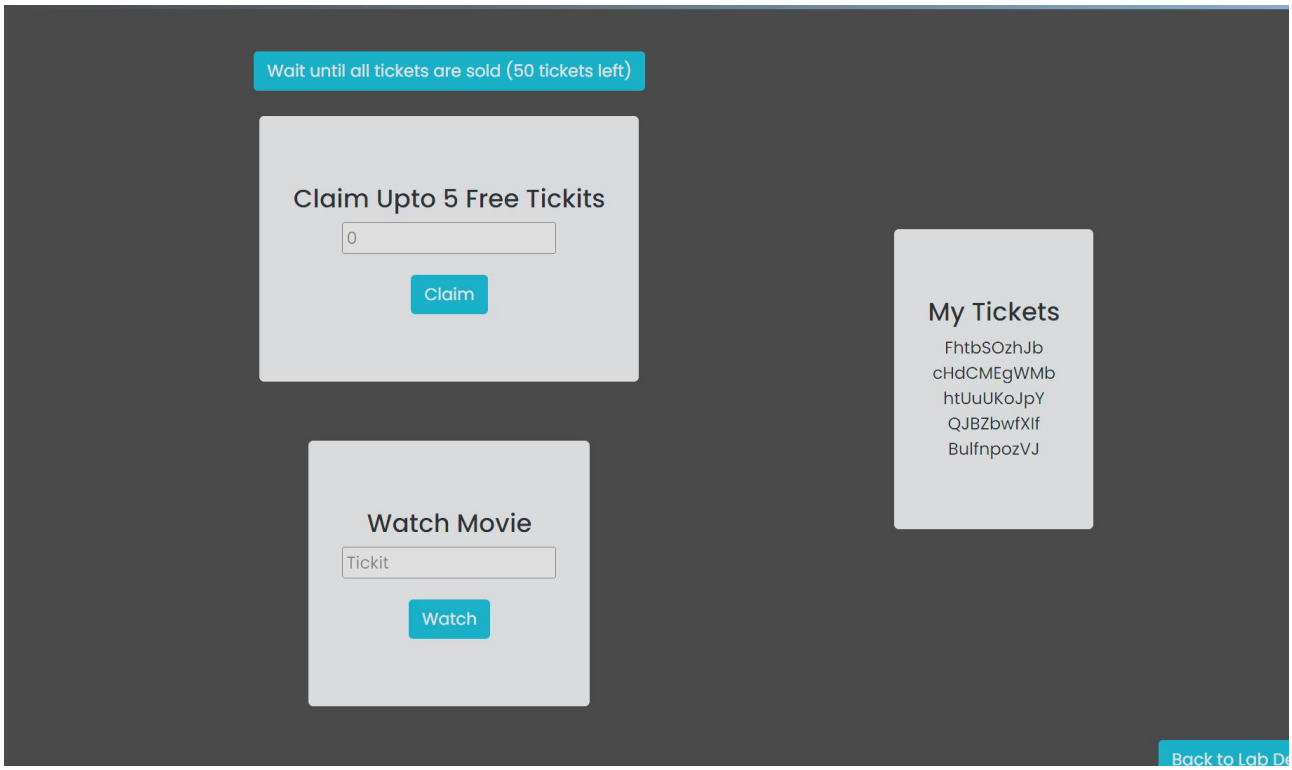
Thử nghiệm lấy vé như bình thường



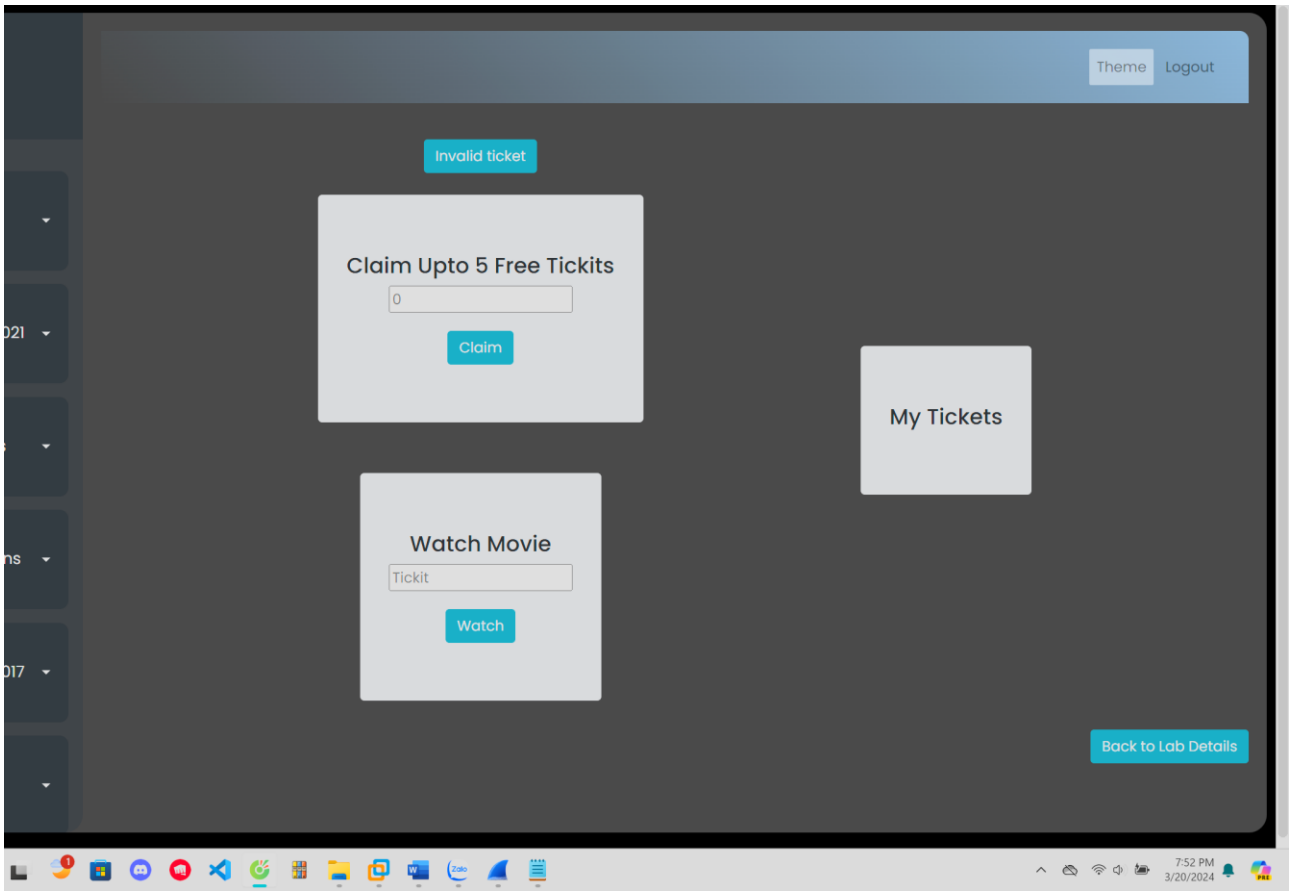
Sau khi lấy 5 vé thì dù có claim nữa thì ta vẫn không nhận được vé, nhưng nếu **claim 0** thì ta sẽ biết được còn bao nhiêu vé

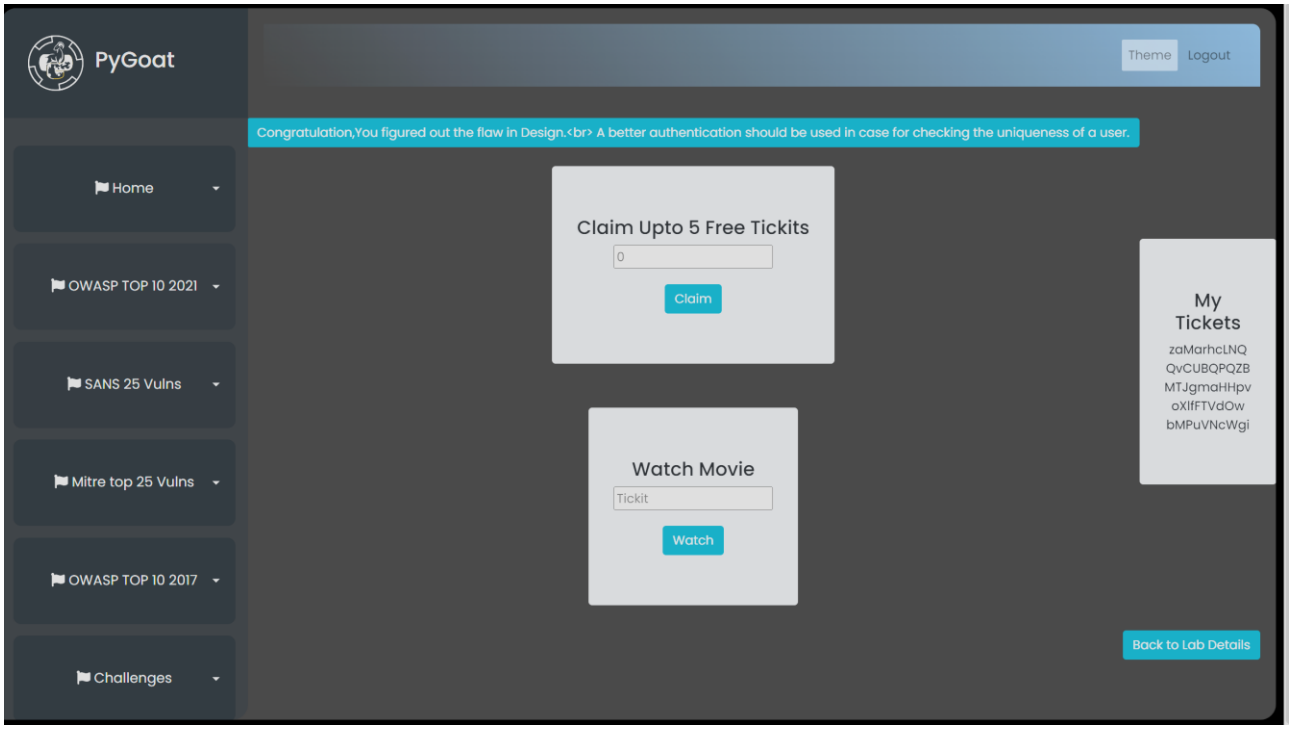


Thử lại việc này với tài khoản khác



Như vậy để lấy được vé miễn phí ta chỉ cần tạo 12 tài khoản.





Sau khi hết vé , bất cứ người dung nào khi đăng nhập vào đều có thể lấy 5 vé và chúng đều xài được.

Mức độ ảnh hưởng: Rất cao

Ở trường hợp này : vé được phát ra liên tục ngay sau khi hết vé, điều này có thể gây tổn thất về tài chính, đồng thời ảnh hưởng đến hình ảnh của nơi bán.

Khuyến cáo khắc phục:

Sửa đổi kiến trúc và thiết kế: Chỉnh sửa kiến trúc và thiết kế của hệ thống để loại bỏ các lỗ hổng và cải thiện tính bảo mật

Thực hiện kiểm soát và theo dõi liên tục: Thiết lập các cơ chế kiểm soát và theo dõi liên tục để phát hiện và ngăn chặn các hoạt động đáng ngờ và tấn công mạng.

- Trong trường hợp này theo em chúng ta nên thiết lập 1 hệ thống kiểm tra số vé được bán ra, và yêu cầu thanh toán online hoàn thành thì mới được nhận vé, lưu lại các mã vé trong hệ thống để thực hiện đối chiếu khi có người sử dụng.

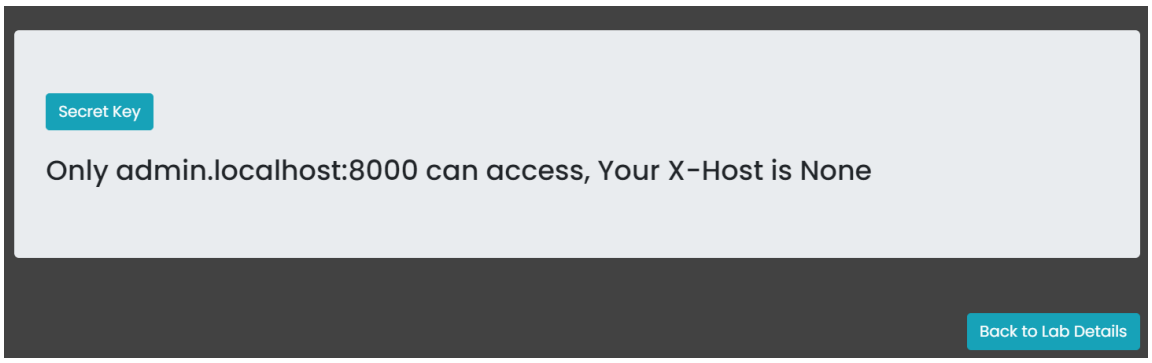
Câu 6: Security Misconfiguration – Data

Mô tả:

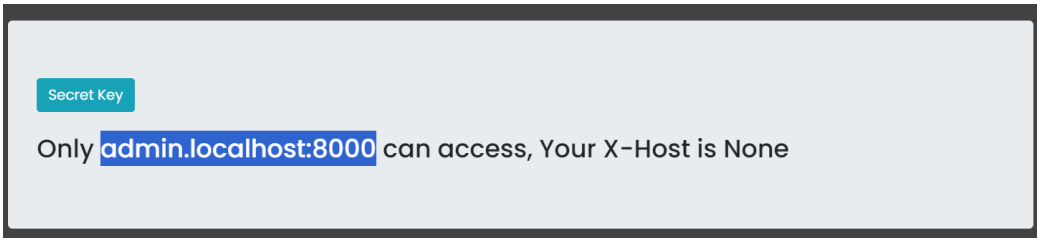
Thêm trường X-host vào request và gửi lên server có thể lấy được secret key

Các bước thực hiện:

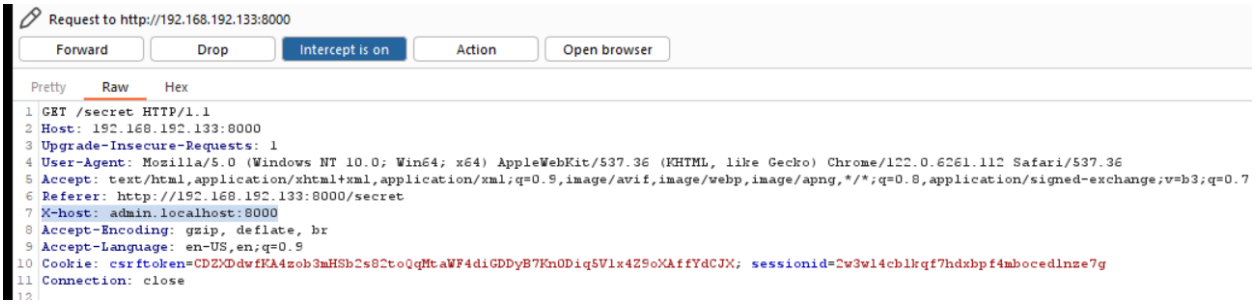
Nhấn thử vào nút Secret Key, vì là tài khoản User nên sẽ không có key



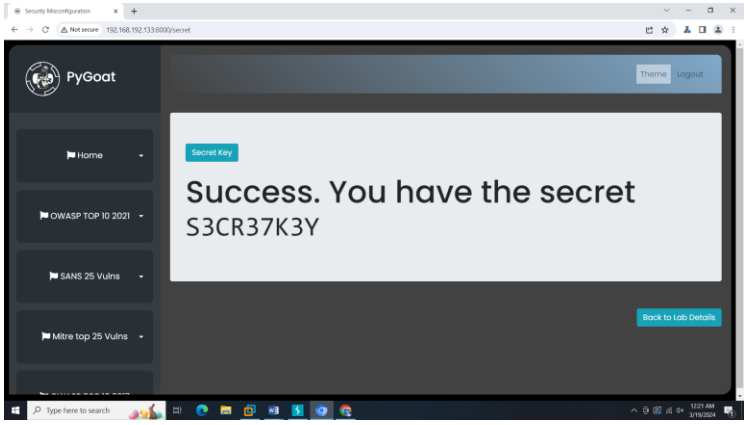
Thông báo này cho ta gợi ý về cách có thể lấy được secret key



Tiến hành chặn gói tin request và thêm vào trường **X-Host: admin.localhost:8000**



Gửi lại gói tin đến server và ta có được secret key là **S3CR37K3Y**



Mức độ ảnh hưởng: Rất cao. Khi có secret key, kẻ tấn công có thể truy cập, xóa hoặc chỉnh sửa dữ liệu

Khuyến cáo khắc phục:

Kiểm tra cấu hình bảo mật, đảm bảo chúng được thiết lập đúng và an toàn. Điều này bao gồm kiểm tra các tệp cấu hình, cấu hình môi trường, cấu hình hệ thống và các cài đặt bảo mật khác

Tuân thủ các nguyên tắc bảo mật, áp dụng các nguyên tắc bảo mật tiêu chuẩn như OWASP. Cập nhật phiên bản mới nhất trong bảo mật

Sử dụng các công cụ kiểm tra tự động để tìm kiếm các lỗ hổng phổ biến, cấu hình không an toàn và khuyết điểm bảo mật.

Tham khảo:

https://owasp.org/Top10/A05_2021-Security_Misconfiguration/