Đại học Quốc gia Thành phố Hồ Chí Minh

Trường Đại học Công nghệ Thông tin (UIT)

BÁO CÁO BÀI TẬP

Môn học: Bảo mật Web và Ứng dụng

Kỳ báo cáo: Buổi 01 (Session 01)

Tên chủ đề: Intro

GV: Nghi Hoàng Khoa

Ngày báo cáo: 15/03/2023

Nhóm: 7

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT213.N21.ANTN

STT	Họ và tên	MSSV	Email
1	Võ Anh Kiệt	20520605	20520605@gm.uit.edu.vn
2	Hoàng Đình Hiếu	20521317	20521317@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

STT	Công việc	Kết quả tự đánh giá	Người đóng góp
1	Kịch bản 01	100%	
2	Kịch bản 02	100%	
3	Kịch bản 03	100%	
4	Kịch bản 04	100%	
5	Kịch bản 05	100%	

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

1. Kịch bản 01

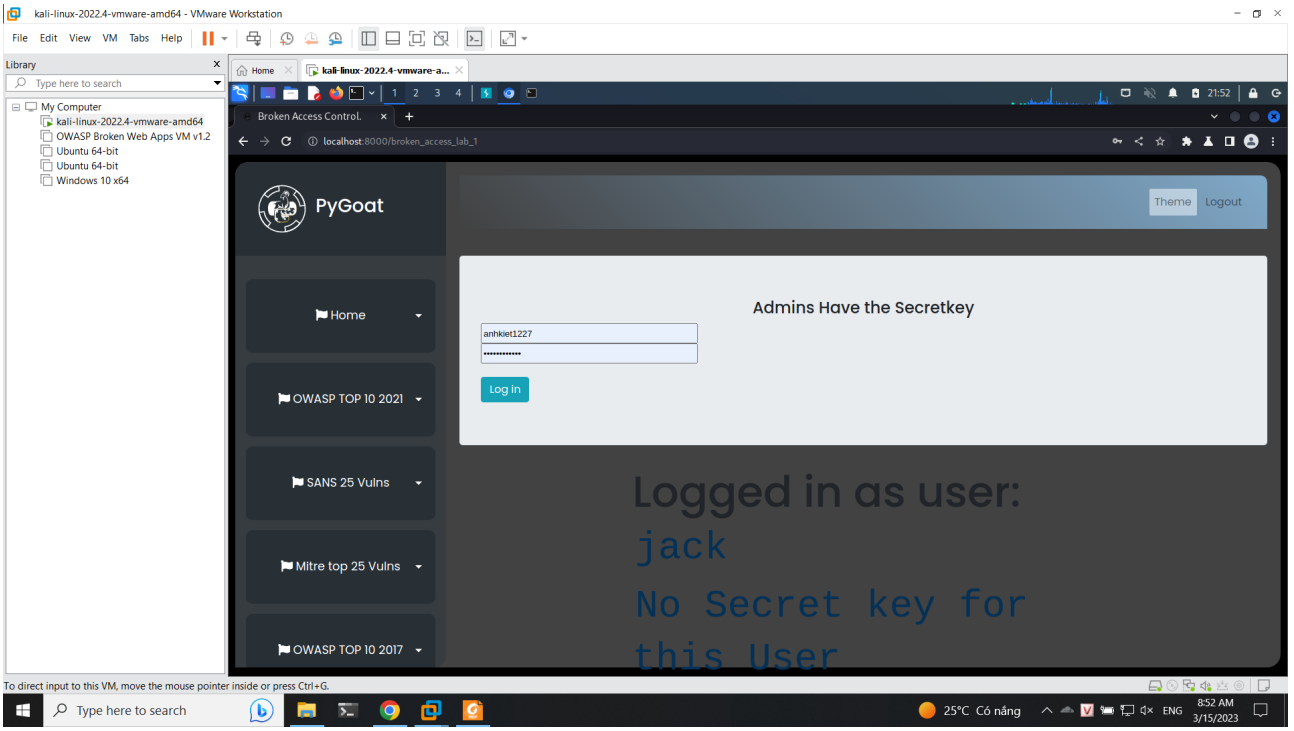
Broken access control - information, data

Mô tả

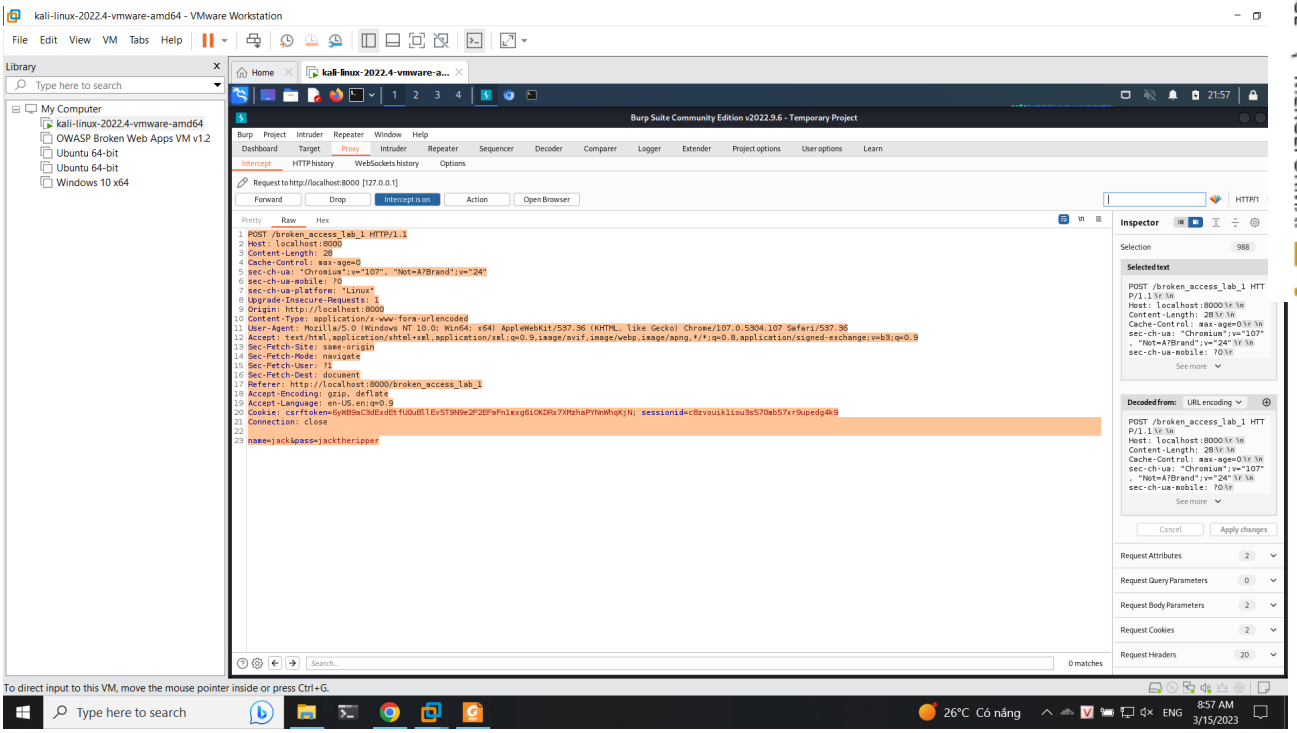
Login gặp vấn đề khi thực hiện sử dụng repeater để gửi thông tin lên trang web

Các bước thực hiện

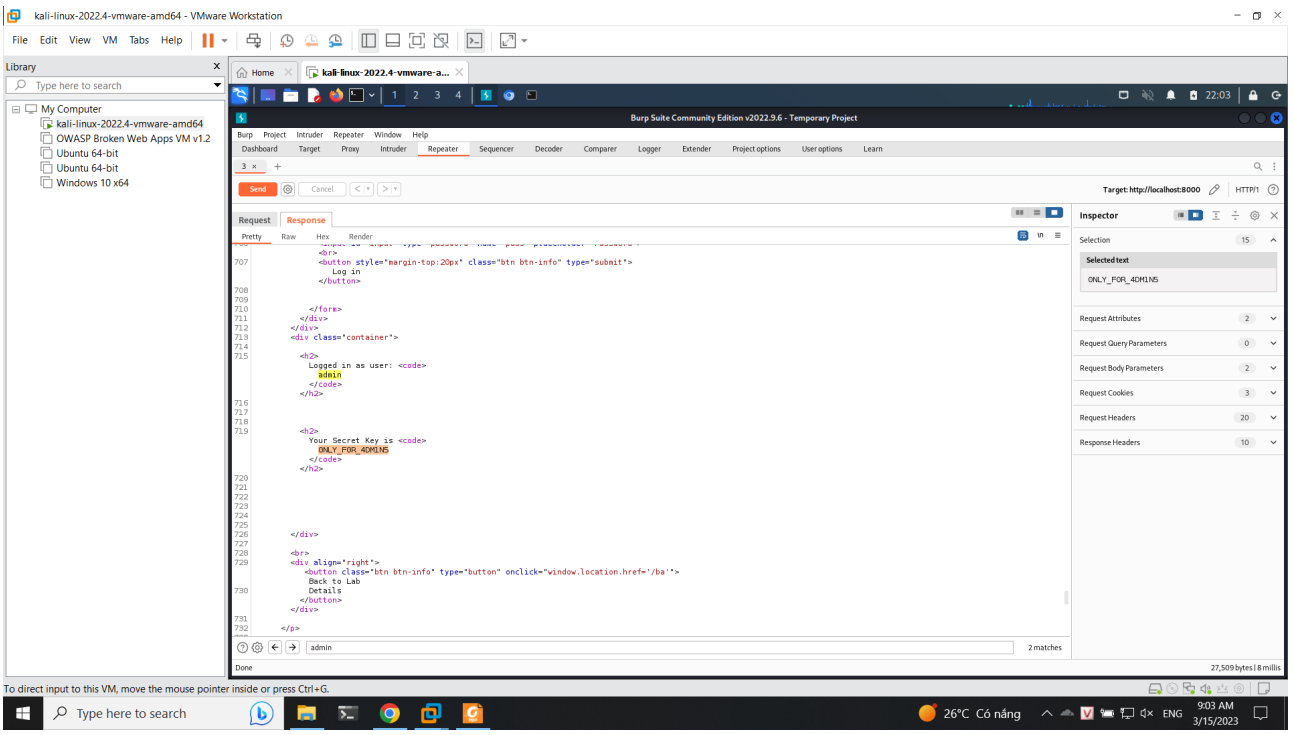
Đầu tiên ta sẽ thực hiện thử đăng nhập bằng tài khoản jack nhưng không có được thông tin nào



Sau đó ta sẽ bắt gói tin khi gửi lên server



Sau đó thực hiện chuyển đến repeater và thay đổi bằng thêm trường admin=1 vào phần cookie vào gửi lên server lại.



Ta có được key là: 0NLY_F0R_4DM1N5

Mức độ: high

Khuyến cáo:

Xác thực quyền truy cập để phòng chống

Nếu kẻ tấn công cố can thiệp vào ứng dụng hoặc cơ sở dữ liệu bằng cách sửa đổi tham chiếu đã cho, thì hệ thống sẽ có thể tắt yêu cầu, xác minh rằng người dùng không có thông tin xác thực phù hợp.

Các ứng dụng web nên dựa vào kiểm soát truy cập phía máy chủ hơn là phía máy khách để kẻ thù không thể giả mạo nó. Ứng dụng nên thực hiện kiểm tra ở nhiều cấp độ, bao gồm cả dữ liệu hoặc đối tượng, để đảm bảo không có lỗ hổng nào trong quy trình.

Tham khảo:

Tham khảo tài liệu lab1

https://www.eccouncil.org/cybersecurity-exchange/web-application-hacking/broken-access-control-vulnerability/#:~:text=Broken%20access%20control%20vulnerability%20is,to%20sensitive%20information%20or%20systems.

2. Kịch bản 02

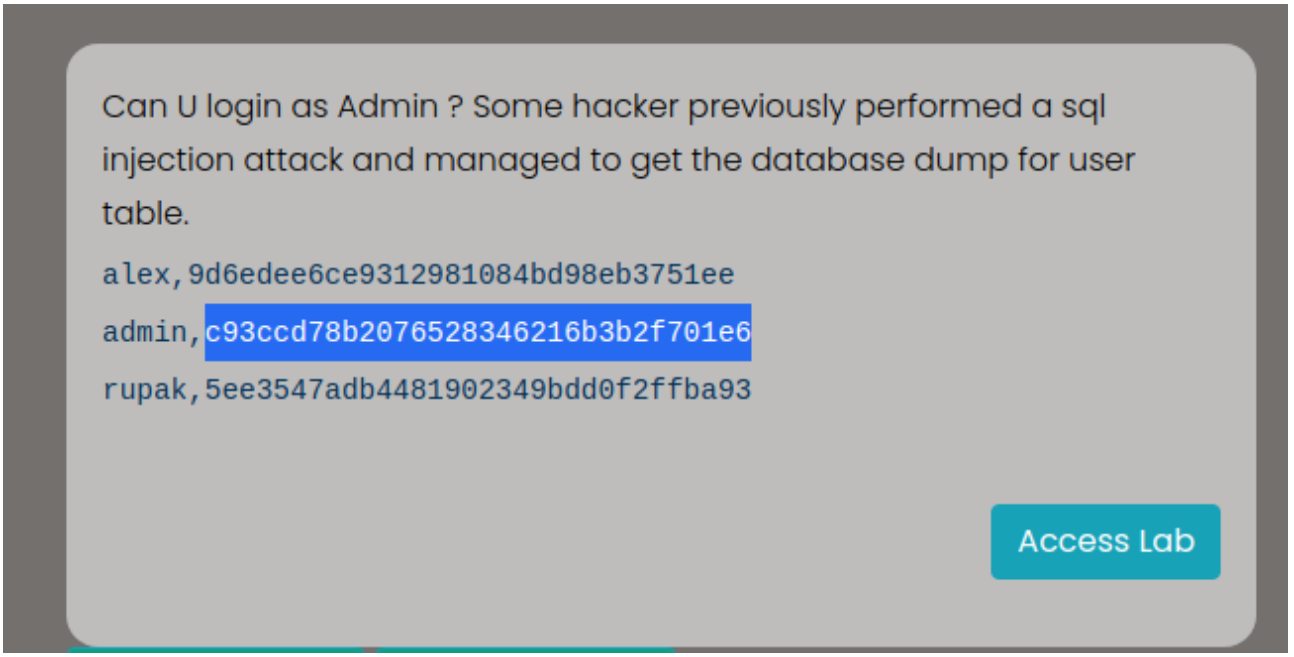
Cryptography failure - Information, data

Mô tả

Lỗi liên quan đến hash killer được map tìm kiếm và tạo table trên internet

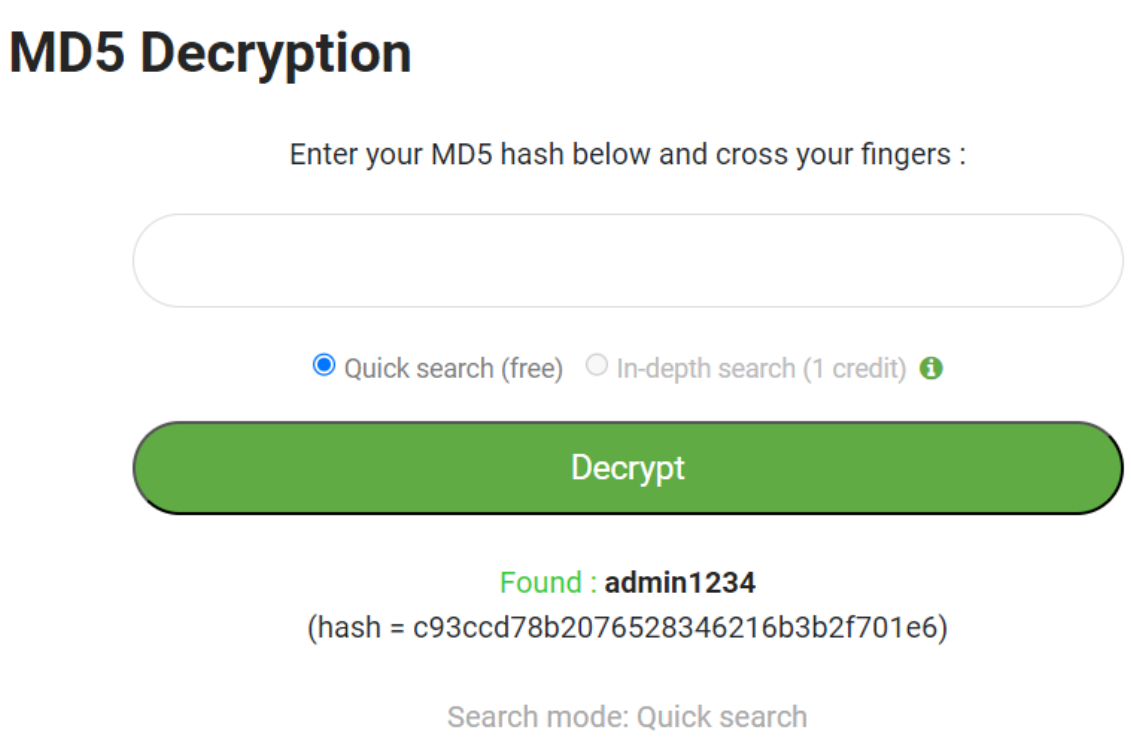
Các bước thực hiện

Đầu tiên ta thấy được là thông tin cung cấp là account và password dưới dạng băm md5

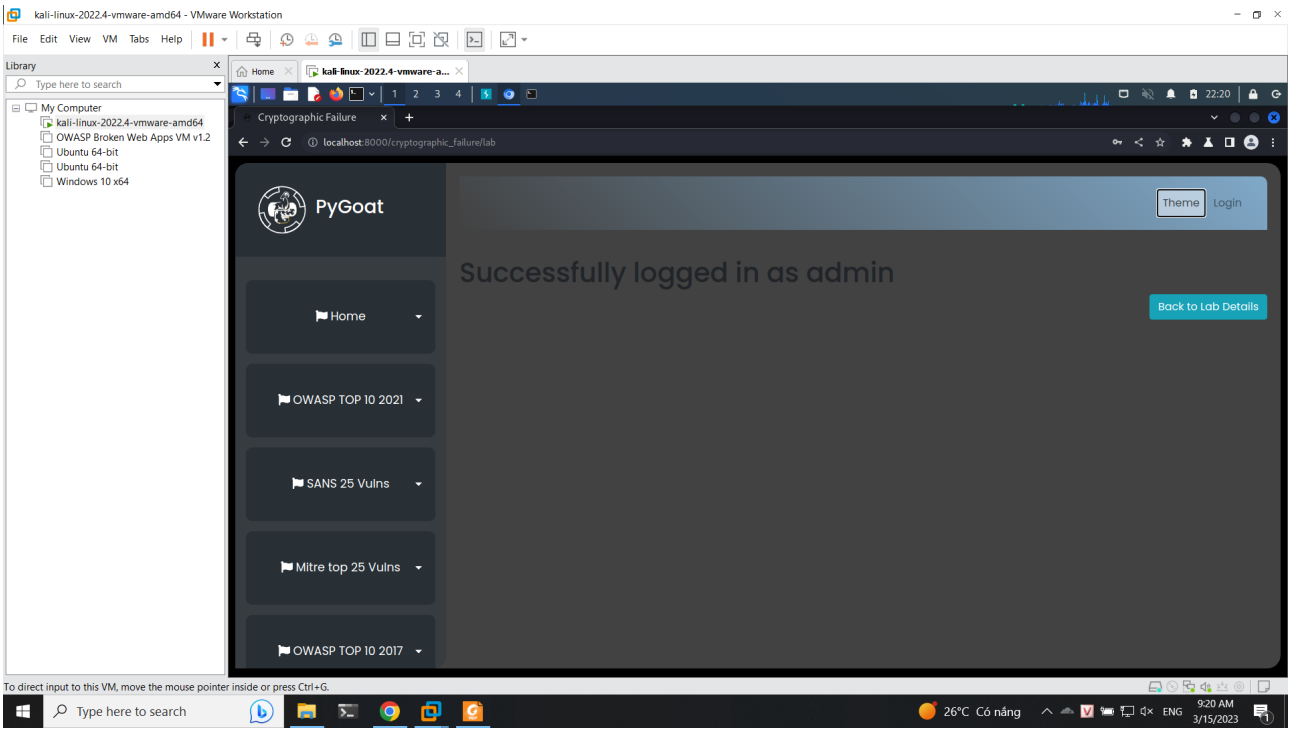


Với băm md5 ta có thể check mật khẩu từ trang:
https://www.md5online.org/md5-decrypt.html

ta thấy được password là admin1234



Thử đăng nhập lại và thành công



Mức độ: high

Khuyến cáo:

Sử dụng hash đời 2 đời 3 hoặc băm mật khẩu, không sử dụng md5 hay sha1 để băm

Tham khảo:

Tài liệu mật mã học thầy Tự

3. Kịch bản 03

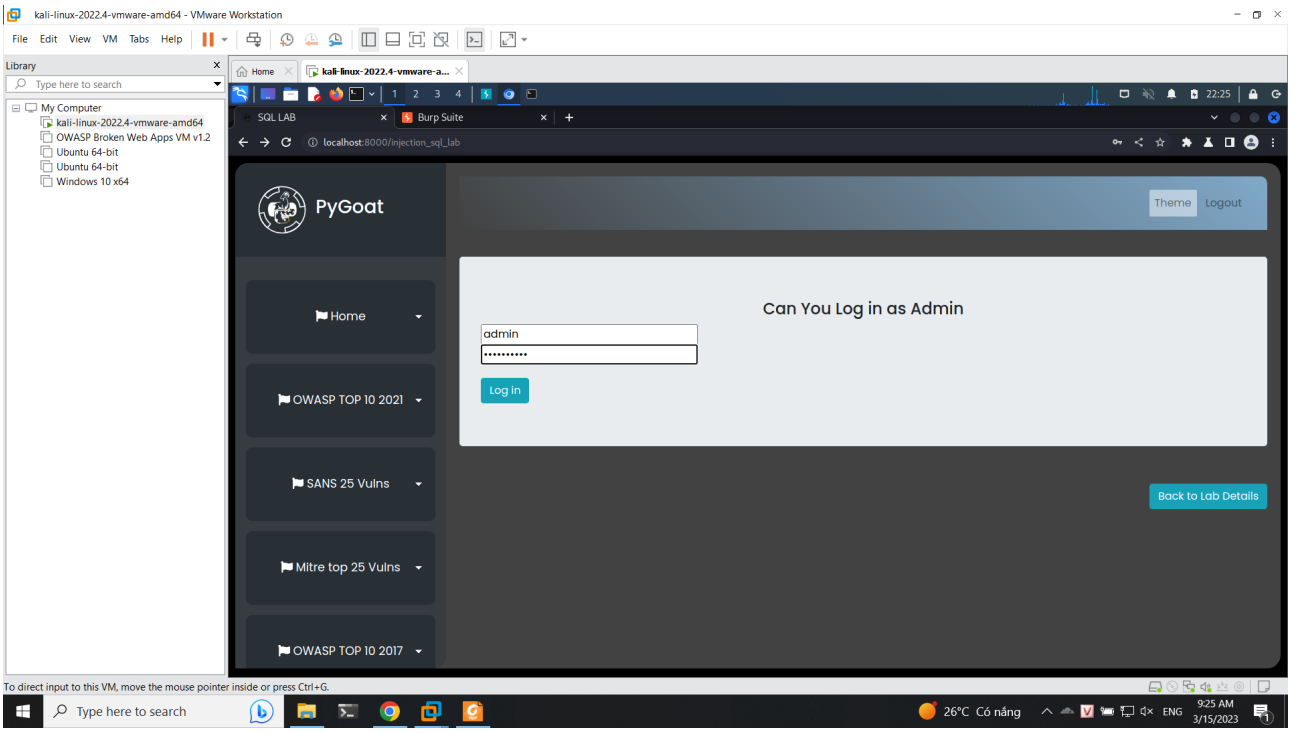
SQL Injection - data, information

Mô tả

Thực hiện truy cập thông qua việc việc truyền câu lệnh sql vào

Thực hiện

Đầu tiên ta biết được rằng là truy cập với account admin, và gợi ý của bài này sql injection ta sẽ thực hiện câu lệnh ‘ or 1=1– để thực hiện đăng nhập



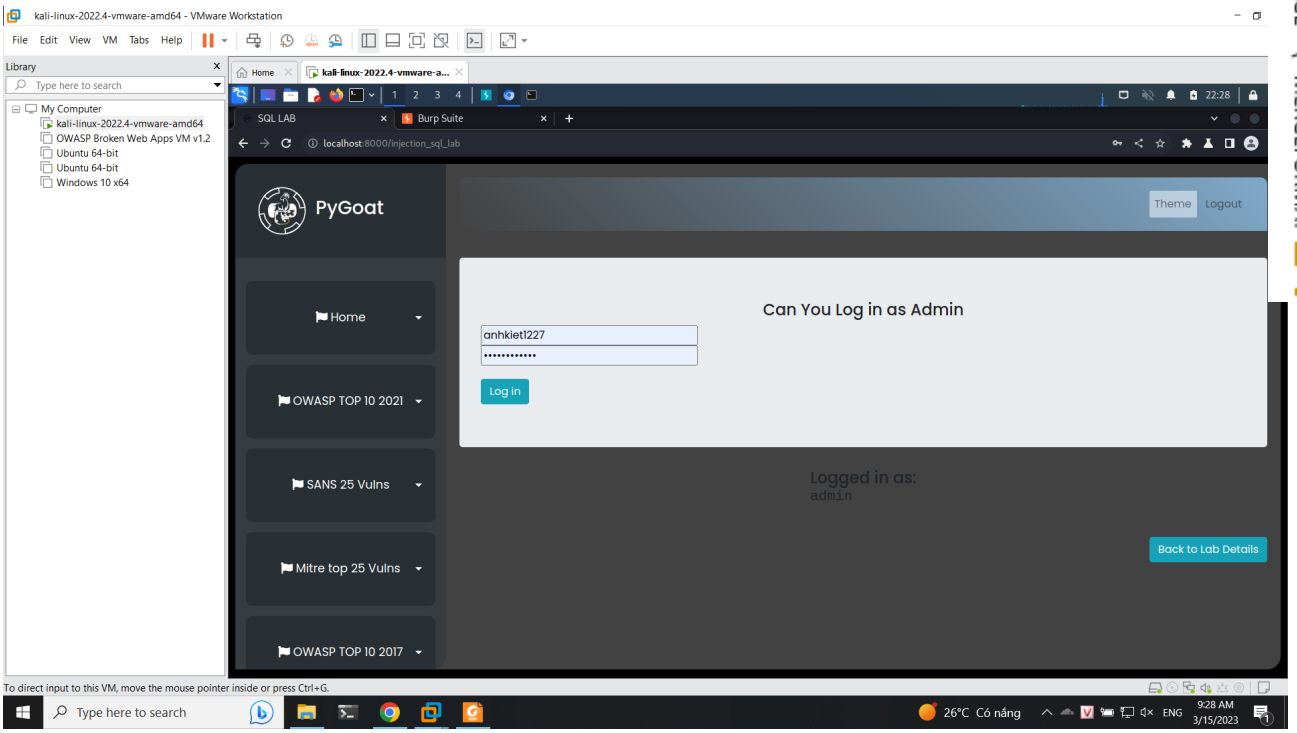
Giải thích

‘ để đóng phần pass truyền vào phía trước

or 1=1 để truyền 1 câu điều kiện luôn đúng vào để chấp nhận truy cập sai

– để comment toàn bộ lệnh phía sau

Ta sẽ thực hiện thử



Ta đã đăng nhập được vào với quyền admin

Mức độ: High

Khuyến cáo:

Bài viết chia sẻ về SQL Injection:
https://www.facebook.com/groups/bht.cnpm.uit/posts/946708845926790/

1/ Tham số hóa câu lệnh

Các ngôn ngữ lập trình khi làm việc sẽ giao tiếp với CSDL bằng cách sử dụng các trình điều khiển CSDL. Các trình điều khiển này cho phép ứng dụng xây dựng và chạy các câu lệnh SQL trên dựa trên CSDL, được trích xuất và thao tác dữ liệu khi cần thiết. Lúc này việc tham số hóa các câu lệnh đảm bảo rằng các giá trị đầu vào được truyền vào các câu lệnh theo 1 cách an toàn hơn.

Sự khác biệt chính là dữ liệu được kết hợp thêm phương thức executeQuery(). Trong trường hợp đầu tiên ta có dễ dàng nhận thấy được chuỗi đã được tham số hóa và các chuỗi này được chuyển đến một “khu vực” riêng biệt, cho phép “điều khiển” một cách chính xác. Trong trường hợp thứ 2, câu lệnh SQL được thiết lập trước khi trình điều khiển được gọi, như vậy chúng có thể bị tấn công. Như vậy việc tham số hóa câu lệnh giúp câu lệnh được chuyển đổi theo dạng tham số giúp cho việc bảo mật trở nên an toàn hơn.

2/ Phương pháp ORM

Bằng cách sử dụng phương pháp ORM, người lập trình sẽ tránh việc phải viết những câu lệnh SQL như vậy giúp cho việc bảo mật được cải thiện hơn.

3/ “Thoát khỏi ký tự đặc biệt”

Bằng cách hạn chế các ký tự đặc biệt (như nháy ‘ hay ngoặc kép “) khi nhập liệu vào sẽ có thể hạn chế được phần nào đó việc bị SQL injection. Nhưng cách này có một số hạn chế nhất định như việc phải hạn chế được tất cả ký tự đặc biệt trong câu lệnh được xây dựng và đôi lúc cuộc tấn công không chỉ dựa trên những ký tự đặc biệt mà còn một số trường hợp đặc biệt.

4/ Lọc đầu vào

Kiểm tra xem các thông tin được nhập vào như địa chỉ email có khớp với một biểu thức bình thường hay không.

Đảm bảo rằng các ký tự bao gồm số và chữ không chứa các ký tự đặc biệt.

Từ chối hoặc loại bỏ khoảng trắng và các ký tự xuống dòng nếu chúng không phù hợp.

Xác thực phía máy khách để cung cấp cho người dùng phản hồi ngay lập tức khi điền vào biểu mẫu, nhưng không có khả năng bảo vệ chống lại một tin tặc nghiêm trọng. Hầu hết các vụ hack được thực hiện bằng cách sử dụng các tập lệnh, thay vì chính trình duyệt.

5/ Phân quyền truy cập

Khi lập trình cần phải đảm bảo rằng việc phân quyền cho người sử dụng chỉ ở hạn mức nào đó tránh việc phân quyền quá nhiều, khi đó hacker có thể sử dụng việc phân quyền đó để tấn công CSDL

6/ “Băm” mật khẩu

Ví dụ về vụ hack dựa trên thực tế là mật khẩu được lưu trữ dưới dạng văn bản thuần túy trong CSDL. Trên thực tế, việc lưu trữ mật khẩu không được mã hóa là một lỗ hổng bảo mật lớn. Các lập trình viên nên viết ứng dụng được lưu trữ mật khẩu người dùng dưới dạng mã băm mạnh, một chiều, tốt nhất là dạng “salted”. Điều này giảm thiểu nguy cơ bị kẻ xấu đánh cắp thông tin đăng nhập hoặc mạo danh người dùng khác.

7/ Xác thực bên thứ 3

Lưu ý cuối cùng, bạn nên xem xét việc thuê ngoài (bên thứ 3) toàn bộ quy trình xác thực của ứng dụng. Facebook, Twitter và Google đều cung cấp các API Oauth hoàn thiện, có thể được sử dụng để cho phép người dùng đăng nhập vào trang web của bạn bằng tài khoản hiện có của họ trên các hệ thống đó. Điều này giúp bạn với tư cách là nhà phát triển ứng dụng tránh khỏi việc triển khai xác thực của riêng bạn và đảm bảo với người dùng của bạn rằng mật khẩu của họ chỉ được lưu trữ ở một vị trí duy nhất.

Tham khảo

Wikipedia

Hacksplaining

4. Kịch bản 04

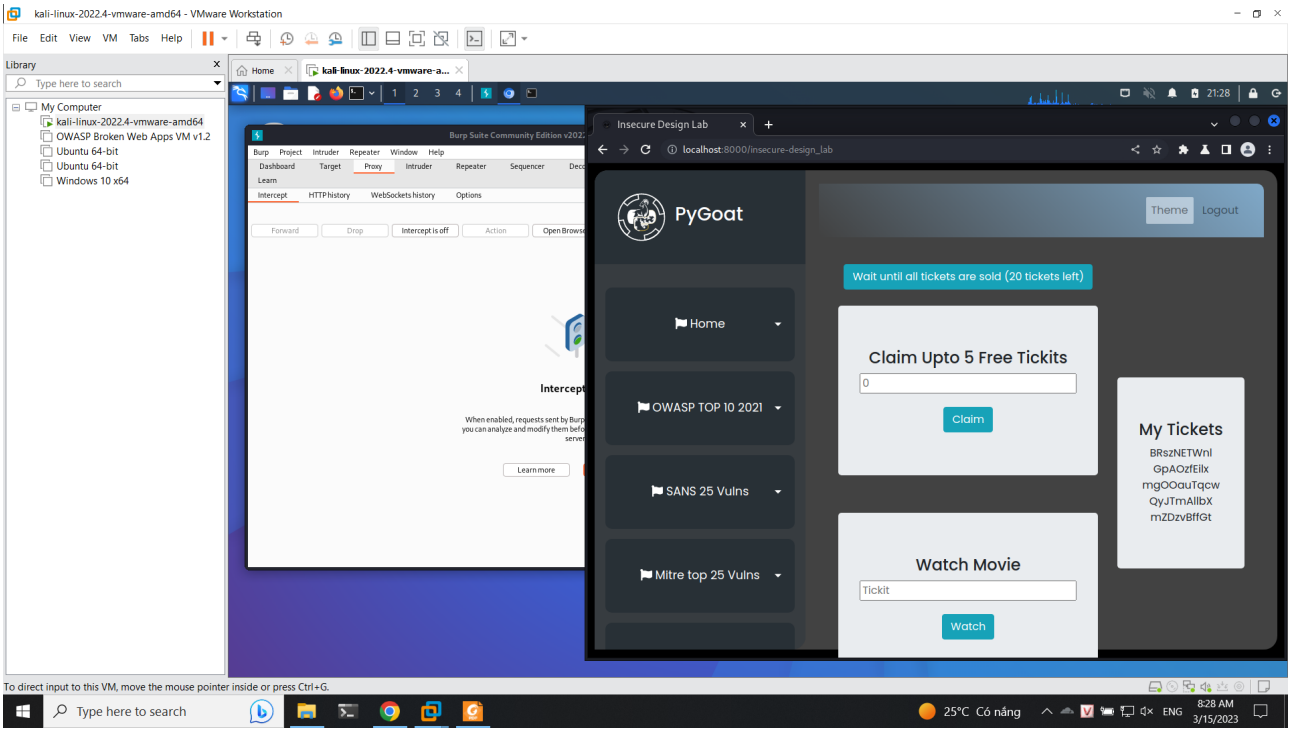
Insecured Design - information, data, profit

Mô tả

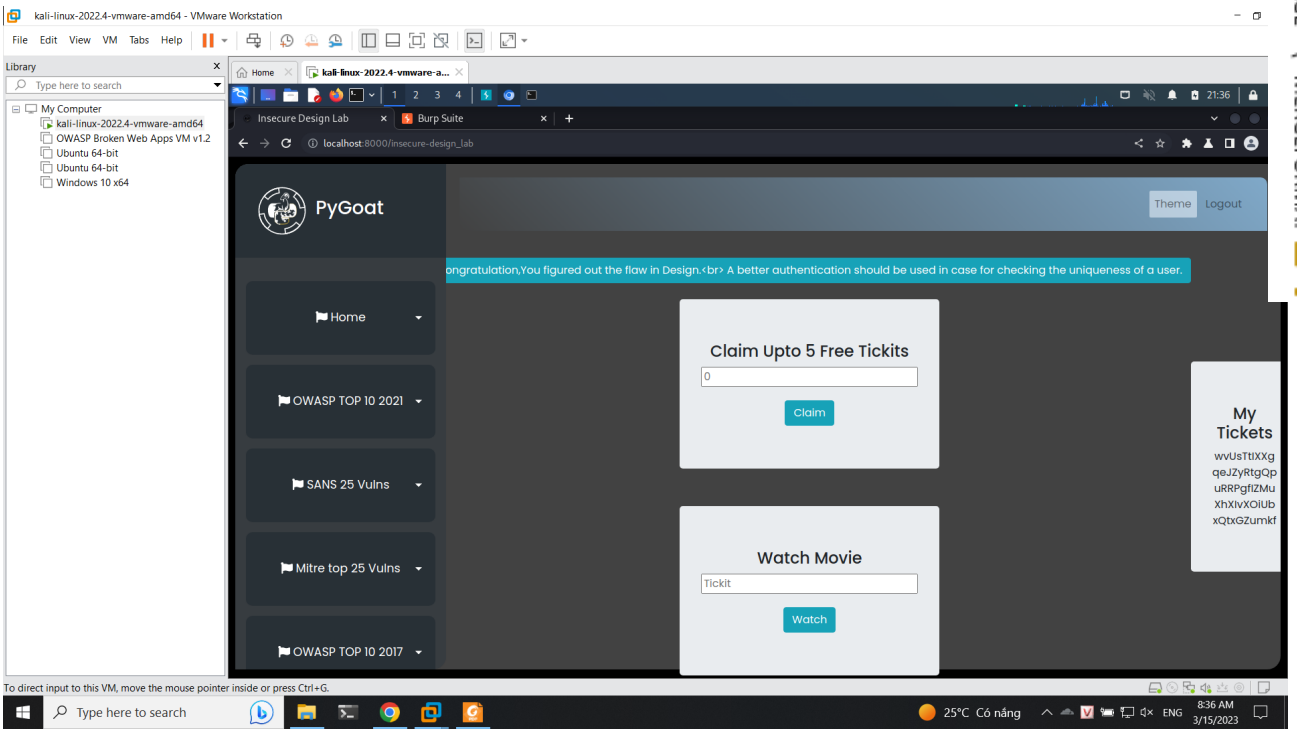
Thiết kế trang web không an toàn khiến bị mất nhiều profit

Thực hiện

Đầu tiên ta sẽ thử tạo tài khoản truy cập vào mua vé, nhưng việc mua vé này không kiểm soát việc ai hay máy nào tạo tài khoản thì việc một người tạo nhiều tài khoản và lấy vé sẽ không kiểm soát được. Vậy ta sẽ thực hiện tạo 12 tài khoản mỗi tài khoản lấy 5 vé để lấy toàn bộ vé



Vậy sau khi mua toàn bộ vé xem và ta sẽ thử xem phim



Ta đã thực hiện được mua hết vé và xem phim

Mức độ: high

Khuyến cáo

Thiết lập và sử dụng vòng đời phát triển an toàn

Thiết lập và sử dụng thư viện các mẫu thiết kế an toàn

Sử dụng mô hình mối đe dọa để xác thực quan trọng, kiểm soát truy cập, logic nghiệp vụ và các luồng chính

Tích hợp ngôn ngữ bảo mật và điều khiển vào user story

Tích hợp kiểm tra tính hợp lý ở mỗi tầng ứng dụng của bạn (từ giao diện người dùng đến phụ trợ)

Viết các bài kiểm tra đơn vị và tích hợp để xác thực rằng tất cả các luồng quan trọng đều có khả năng chống lại mô hình mối đe dọa.

Tách các lớp tầng trên hệ thống và các lớp mạng tùy thuộc vào nhu cầu tiếp xúc và bảo vệ

Tách biệt người thuê một cách mạnh mẽ theo thiết kế trong tất cả các tầng

Hạn chế tiêu thụ tài nguyên theo người dùng hoặc dịch vụ

Tham khảo:

https://owasp.org/Top10/A04_2021-Insecure_Design/

5. Kịch bản 05

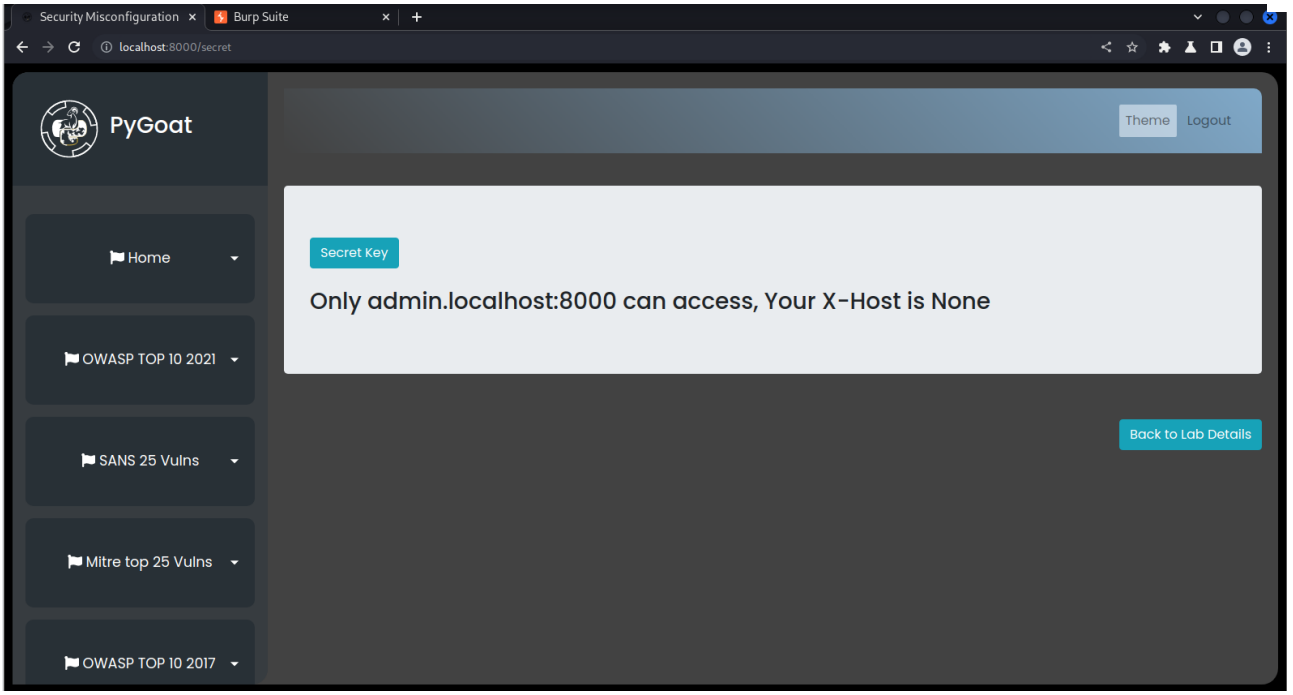
Security Misconfiguration - Data, Information

Mô tả

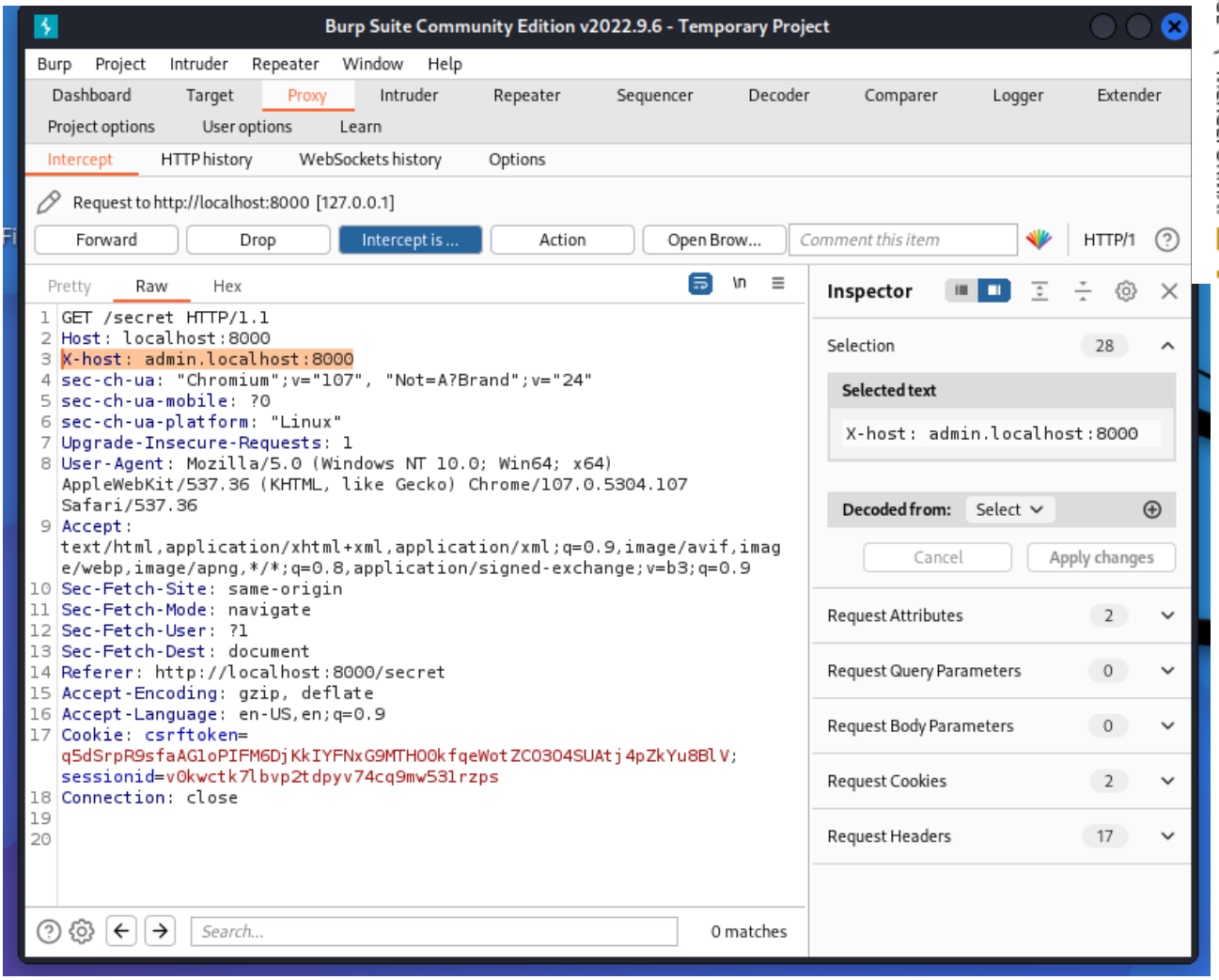
Thêm trường X-Host vào gửi request lên có thể truy cập vào quyền admin

Các bước thực hiện

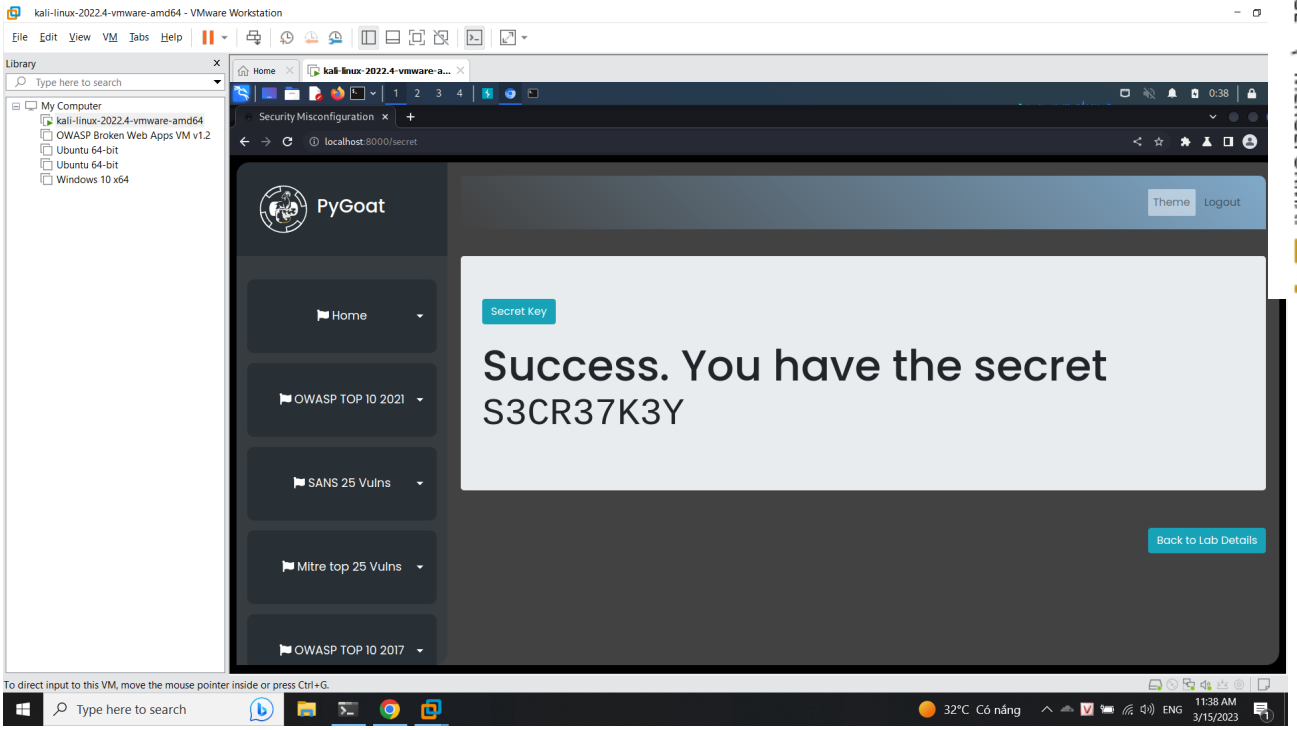
Đầu tiên ta sẽ thực hiện thử get key nhưng không được



Ta sẽ thử chặn gói tin và thêm trường X-Host: admin.localhost:8000



Sau đó ta sẽ thực hiện gửi gói tin này đi đến trang thì ta sẽ truy cập được vào trang admin



Vậy ta có được key là: S3CR37K3Y

Mức độ: High

Khuyến cáo:

Cần thiết kế một quy trình đủ tốt trong phát triển.

Không có thành phần dư thừa khi thực hiện gọi truy vấn, lọc repeater

Cập nhật phiên bản mới nhất trong bảo mật

Xây dựng quy trình tự động để xác minh tính hiệu quả của các cấu hình và cài đặt trong tất cả các môi trường.

Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này

YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hành theo hướng dẫn.
- Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

Báo cáo:

- File .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
- Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
- Đặt tên theo định dạng: [Mã lớp]-SessionX_GroupY. (trong đó X là Thứ tự buổi Thực hành, Y là số thứ tự Nhóm Thực hành/Tên Cá nhân đã đăng ký với GV).

  *Ví dụ: [*NT101.K11.ANTT*]-Session1_Group3.*
- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm.
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá: Sinh viên hiểu và tự thực hiện. Khuyến khích:

- Chuẩn bị tốt.
- Có nội dung mở rộng, ứng dụng trong kịch bản/câu hỏi phức tạp hơn, có đóng góp xây dựng.

Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.

HẾT