BÁO CÁO BÀI TẬP

**Môn học: Bảo mật web và ứng dụng**

**Kỳ báo cáo: Session 5**

**Tên chủ đề: Ôn tập**

*GVHD: Ngô Khánh Khoa*

*Ngày báo cáo: 22/05/2024*

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT213.O21.ANTN

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Phạm Ngọc Thơ | 21522641 | 21522641@gm.uit.edu.vn |
| 2 | Hà Thị Thu Hiền | 21522056 | 21522056@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-1)**

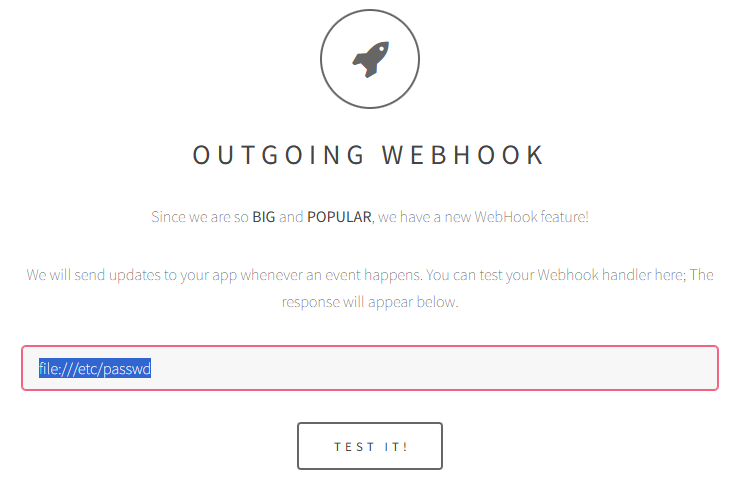
|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** |
| 1 | Kịch bản 01 | 100% |
| 2 | Kịch bản 02 | 100% |
| 3 | Kịch bản 03 | 100% |
| 4 | Kịch bản 04, 05 | 100% |

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

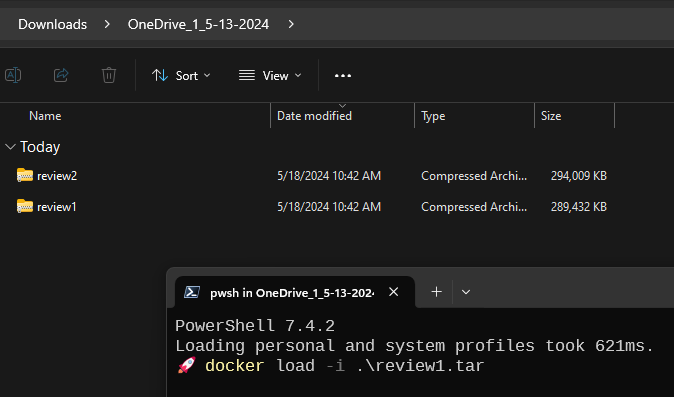
BÁO CÁO CHI TIẾT

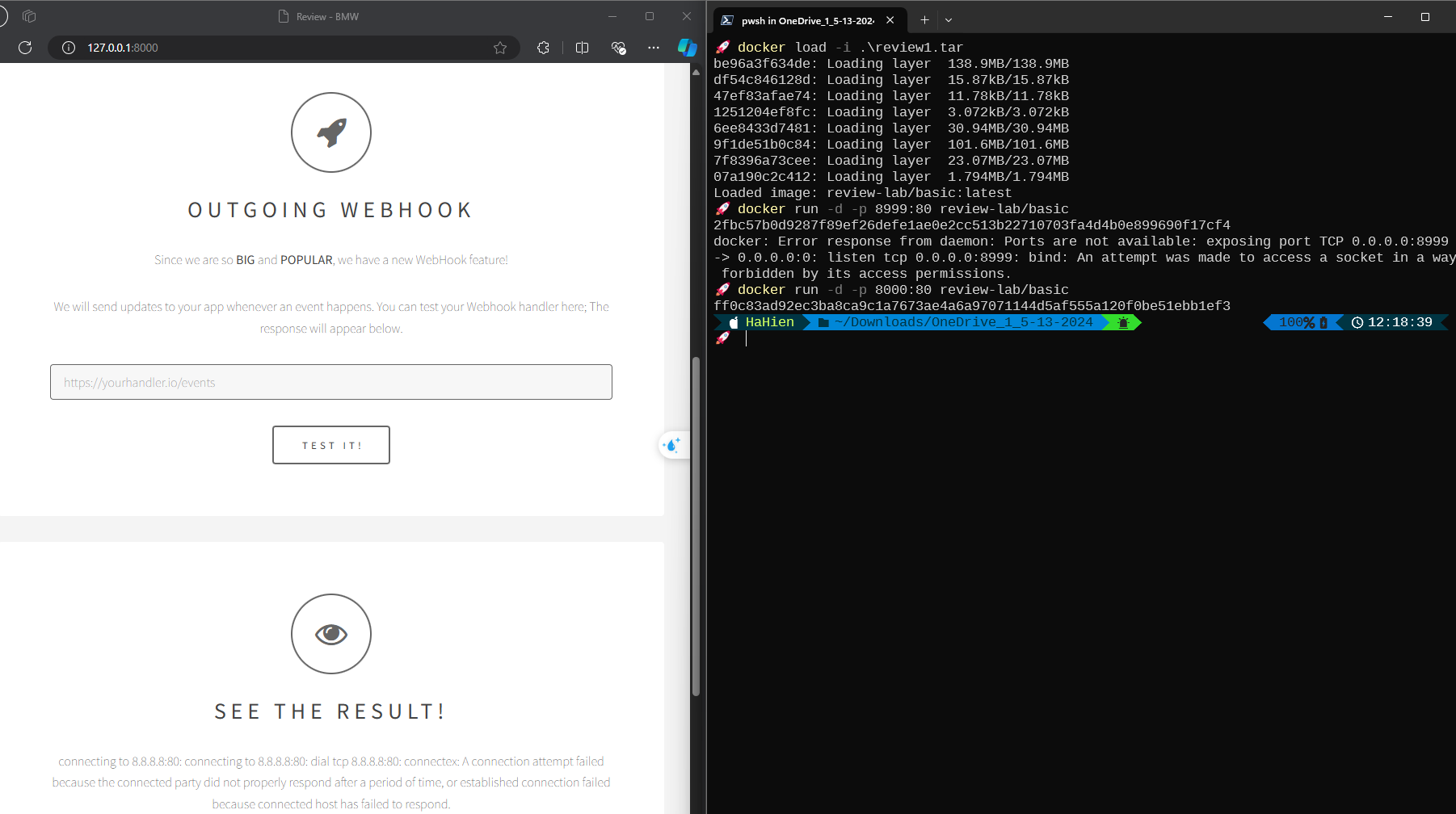
## **Bài tập 1: Báo cáo lỗ hổng tìm thấy. Sử dụng format theo mẫu sau:**

* **Tiêu đề**: Lỗ hổng SSRF - Nguy cơ tấn công từ xa và rò rỉ thông tin nội bộ
  + Lỗ hổng SSRF (Server-Side Request Forgery) có khả năng dẫn đến tấn công từ xa và rủi ro lộ lọt thông tin nội bộ quan trọng. Hệ thống máy chủ và dữ liệu quan trọng là những tài sản có thể bị ảnh hưởng nghiêm trọng.
  + Lỗ hổng SSRF (Server-Side Request Forgery) là một lỗ hổng bảo mật phổ biến trong các ứng dụng web. SSRF xảy ra khi ứng dụng cho phép kẻ tấn công tạo ra và điều khiển các yêu cầu từ máy chủ nội bộ mà không kiểm tra và xác thực đầy đủ. Điều này cho phép kẻ tấn công đọc hoặc tương tác trái phép với các tài nguyên nội bộ như máy chủ cơ sở dữ liệu, dịch vụ nội bộ, hoặc hệ thống quản lý nội bộ. SSRF có thể dẫn đến việc truy cập thông tin nhạy cảm, thực hiện các hành động không mong muốn và gây thiệt hại nghiêm trọng cho hệ thống.
* **Tóm tắt**: Chúng ta sẽ nhập <file:///etc/passwd> theo như hình để lấy password

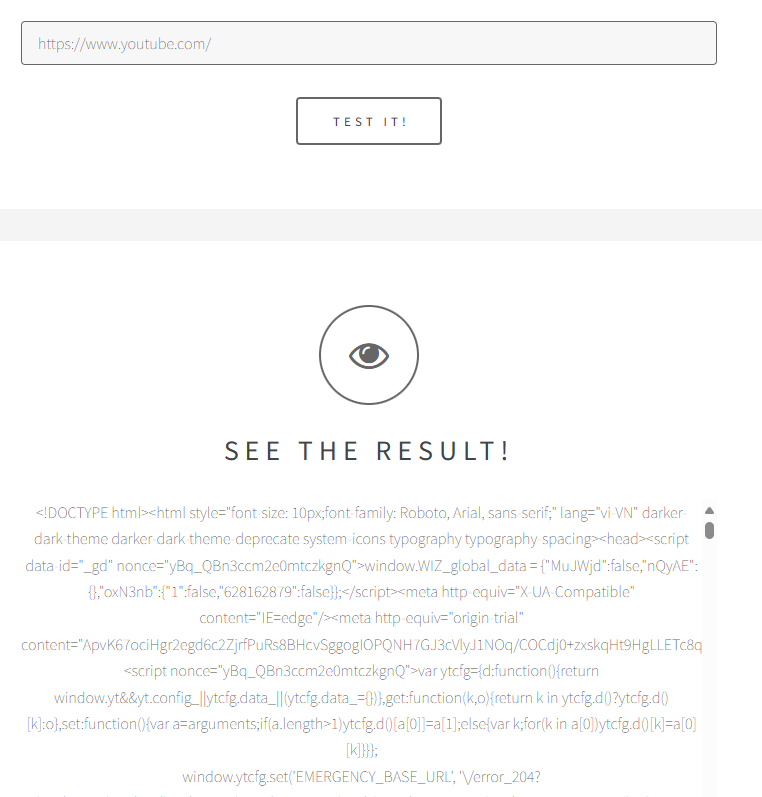


* **Các bước thực hiện**:
  + Chuẩn bị môi trường:

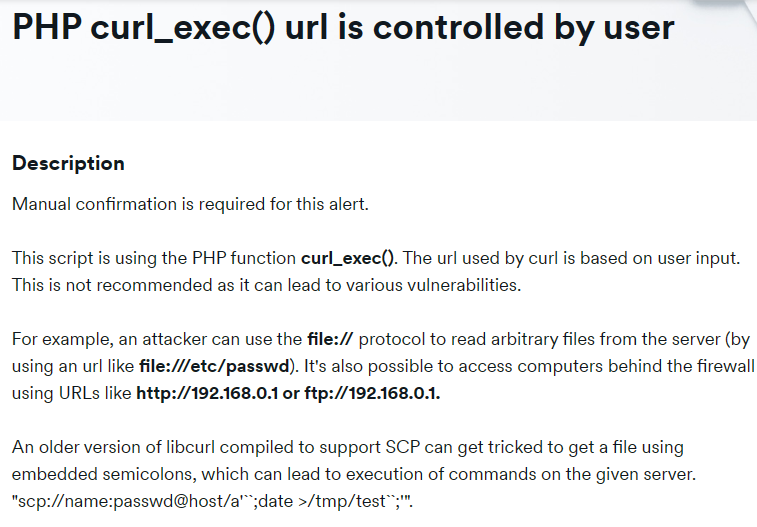




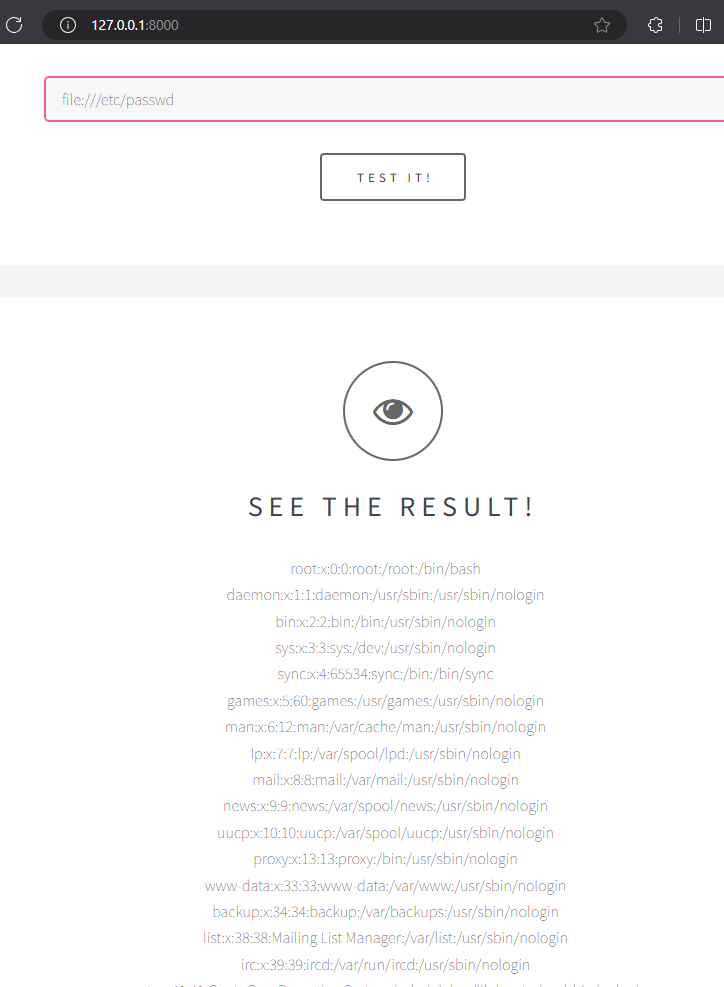
* Dựa vào giao diện web thì ta cần nhâp một chuỗi URL sau đó click button TEST IT ! và trang web sẽ trả về kết quả bên dưới, thử nhập



* Chúng ta có thể dự đoán rằng trang web có thể có lỗ hổng liên quan đến curl\_exec [PHP curl\_exec() url is controlled by user - Vulnerabilities - Acunetix](https://www.acunetix.com/vulnerabilities/web/php-curl_exec-url-is-controlled-by-user/)



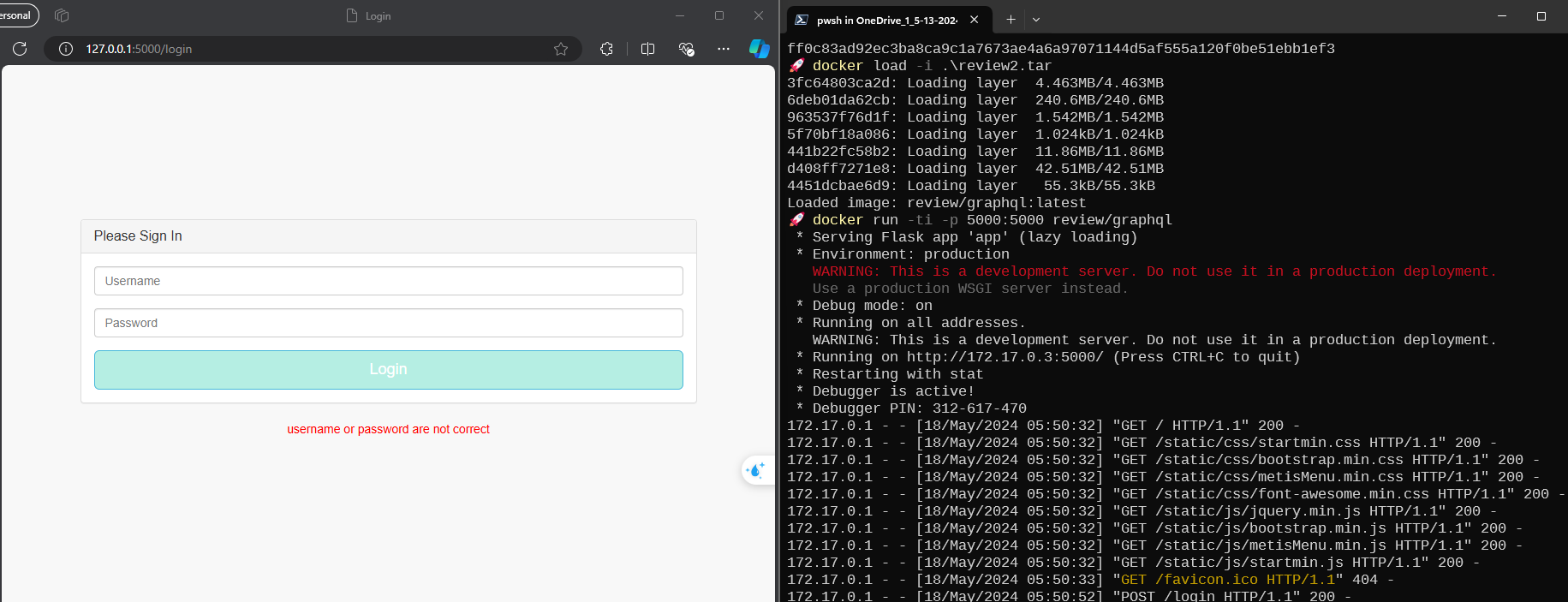
* Chúng ta có thể sử dụng giao thức file:// để đọc các tệp tùy ý từ máy chủ (bằng cách sử dụng url như <file:///etc/passwd>).



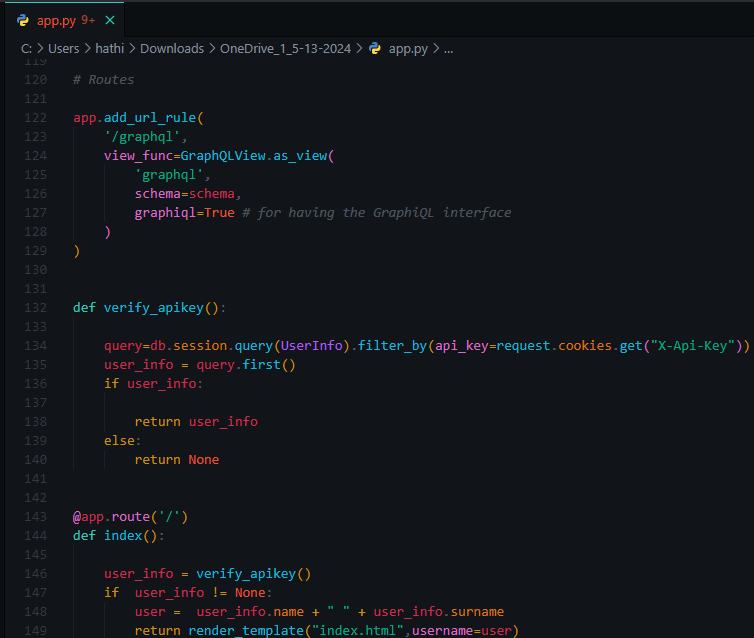
* Khai thác lỗ hổng thành công
* **Mức độ ảnh hưởng của lỗ hổng**: Cao
  + **Lỗ hổng SSRF có mức độ ảnh hưởng cao do khả năng cho phép kẻ tấn công truy cập và tương tác trái phép với các tài nguyên nội bộ, gây ra nguy cơ mất mát dữ liệu quan trọng, làm gián đoạn dịch vụ và tiềm ẩn nhiều nguy cơ bảo mật khác.**
* **Khuyến cáo khắc phục:**
  + Xác thực và lọc đầu vào: Luôn xác thực và lọc các đầu vào từ người dùng để đảm bảo rằng chỉ các URL đáng tin cậy mới được chấp nhận.
  + Giới hạn quyền truy cập: Hạn chế quyền truy cập của ứng dụng đến các tài nguyên nội bộ và bên ngoài, chỉ cho phép truy cập những gì thực sự cần thiết.
  + Sử dụng danh sách trắng: Áp dụng danh sách trắng (whitelist) cho các URL và IP mà ứng dụng có thể truy cập, loại trừ các địa chỉ không tin cậy.
  + Bảo vệ thông qua tường lửa: Sử dụng tường lửa ứng dụng web (WAF) để phát hiện và chặn các yêu cầu bất thường hoặc đáng ngờ.
  + Kiểm tra bảo mật thường xuyên: Thực hiện các bài kiểm tra bảo mật thường xuyên để phát hiện và vá các lỗ hổng mới.
  + Cập nhật và vá lỗi: Luôn cập nhật phần mềm và áp dụng các bản vá bảo mật kịp thời để giảm thiểu rủi ro từ các lỗ hổng đã biết.

1. **Bài tập 2: Báo cáo lỗ hổng tìm thấy, Sử dụng format theo mẫu sau:**

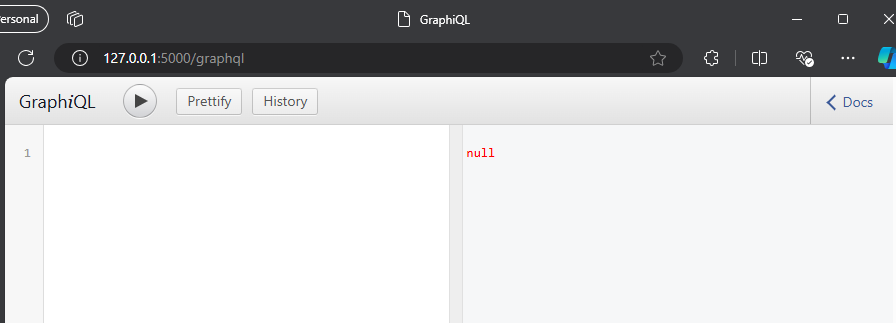
* **Tiêu đề:** Lỗ hổng Insecure Direct Object References (IDOR) là lỗ hổng tham chiếu đối tượng trực tiếp không an toàn, gây nguy cơ lộ thông tin cá nhân, dữ liệu nhạy cảm, và vi phạm quyền truy cập và quyền hạn.
* **Mô tả lỗ hổng:** Lỗ hổng Insecure Direct Object References (IDOR) là một lỗ hổng bảo mật thường gặp trong các ứng dụng web. Lỗ hổng này cho phép người dùng có khả năng truy cập và thay đổi các đối tượng hoặc thông tin mà họ không có quyền truy cập hoặc chỉnh sửa. Điều này có thể dẫn đến lộ thông tin nhạy cảm và vi phạm quyền truy cập, gây ra hậu quả nghiêm trọng cho bảo mật của hệ thống.
* **Tóm tắt**: Thực hiện truy vấn vào trang database và tìm các api để thực hiện truy cập
* **Các bước thực hiện:**
  + Chạy môi trường



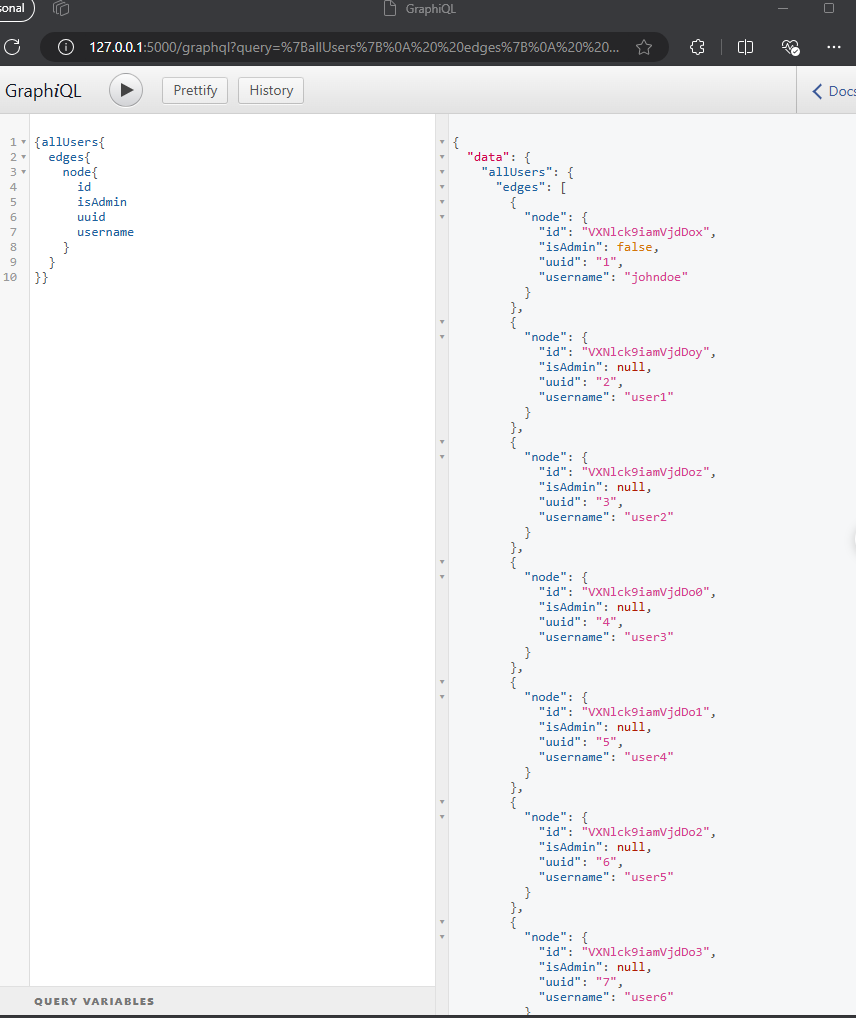
* Mở file app.py đọc code



* Phát hiện có 1 nhánh /graphql, vào thử xem có gì

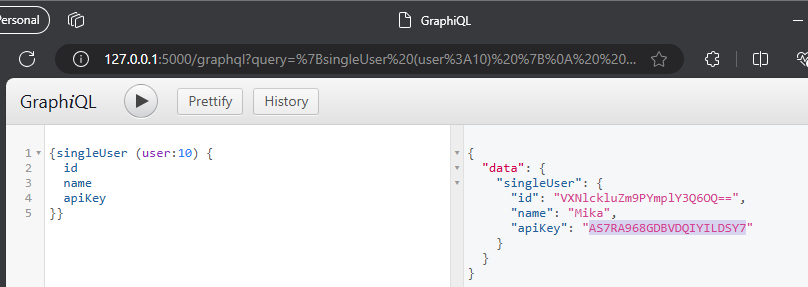


* Thực hiện xem toàn bộ các user và một số thông tin:

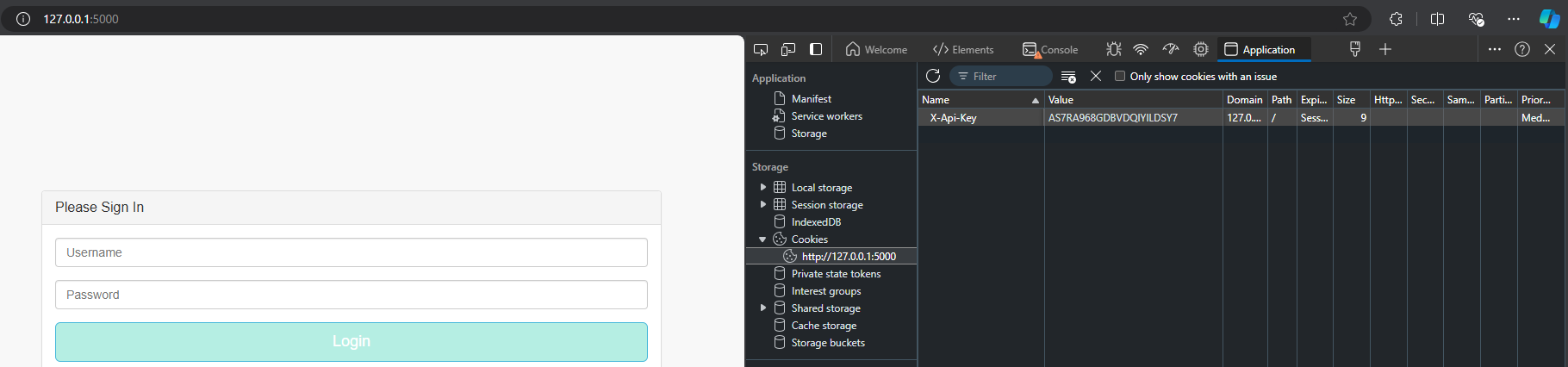
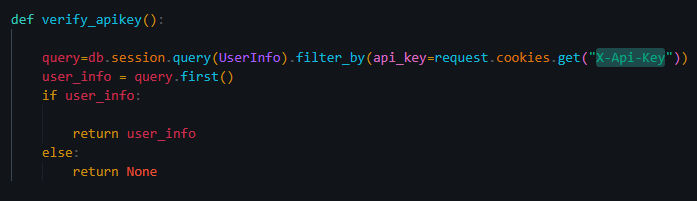




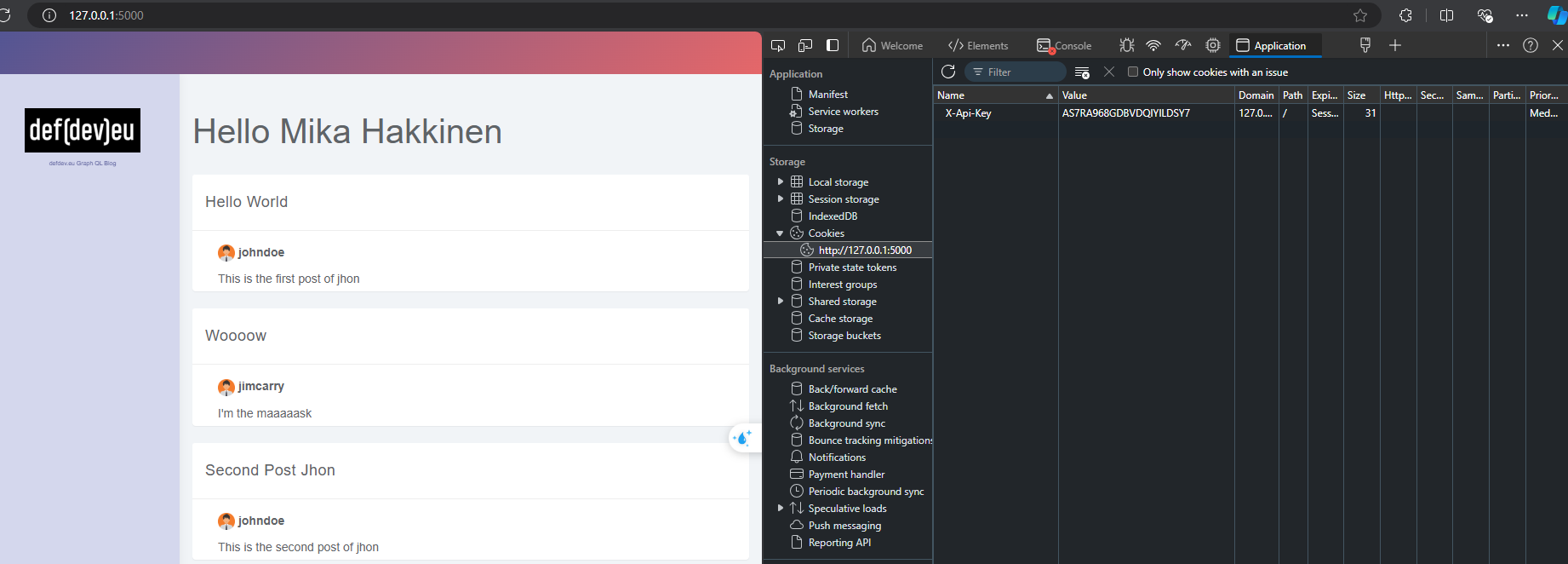
* Ta thấy thông tin cuối cùng “username”: “jimcarry” có “isAdmin”: true, “uuid”: “10”, tiến hành xem thông tin jimcarry với uuid là 10.



* Ta đã lấy được thông tin về apiKey, tiến hành đăng nhập với apiKey này bằng cách truyền apiKey vào application.



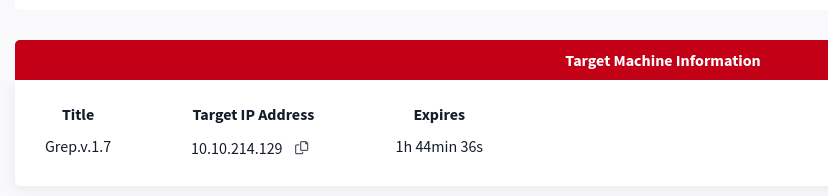
* Refresh lại trang web



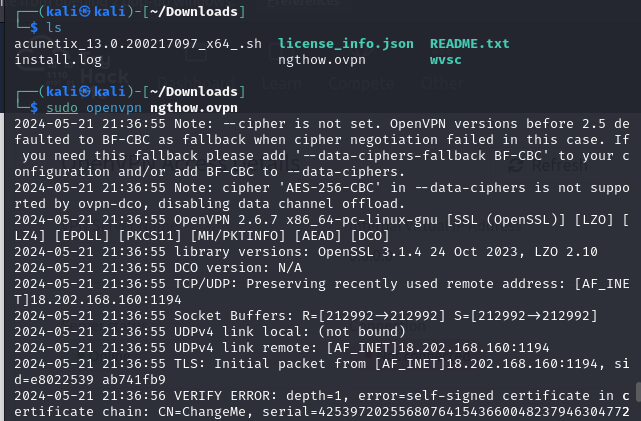
* Đăng nhập vào trang web thành công qua apikey
* **Mức độ ảnh hưởng**: Cao
  + Lỗ hổng Insecure Direct Object References (IDOR) có mức độ ảnh hưởng cao do khả năng lộ thông tin cá nhân, dữ liệu nhạy cảm và vi phạm quyền truy cập và quyền hạn. Sự tồn tại của lỗ hổng này có thể dẫn đến việc tiết lộ thông tin nhạy cảm cho các bên không được ủy quyền và gây thiệt hại nghiêm trọng đến uy tín và bảo mật của hệ thống.
* **Khuyến cáo khắc phục:**
  + Xác thực và ủy quyền đầy đủ: Đảm bảo rằng mọi truy cập đến đối tượng hoặc dữ liệu đều được xác thực và ủy quyền một cách đầy đủ, chỉ cho phép người dùng truy cập và chỉnh sửa các đối tượng mà họ có quyền.
  + Sử dụng mã định danh thay vì tham chiếu trực tiếp: Thay vì sử dụng tham chiếu trực tiếp đến các đối tượng, hãy sử dụng các mã định danh duy nhất để tham chiếu đến chúng, giúp ngăn chặn việc đoán đúng hoặc đoán sai.
  + Giới hạn quyền truy cập: Hạn chế quyền truy cập của người dùng chỉ đến những dữ liệu và đối tượng cụ thể mà họ cần truy cập, tránh việc cung cấp quyền truy cập không cần thiết.
  + Kiểm tra bảo mật thường xuyên: Thực hiện kiểm tra bảo mật định kỳ để phát hiện và vá các lỗ hổng IDOR mới cũng như các lỗ hổng bảo mật khác.
  + Đào tạo và nâng cao nhận thức: Đào tạo nhân viên về nguy cơ và biện pháp phòng ngừa IDOR, đồng thời nâng cao nhận thức về quy trình xác thực và ủy quyền an toàn.
  + Sử dụng công cụ phát hiện lỗ hổng tự động: Sử dụng các công cụ phát hiện lỗ hổng tự động để kiểm tra và phát hiện các lỗ hổng IDOR và các lỗ hổng bảo mật khác trong ứng dụng của bạn.

1. **Bài tập 3: Tìm kiếm thông tin và hoàn thành các thử thách tại** [**https://tryhackme.com/r/room/greprtp**](https://tryhackme.com/r/room/greprtp)**. Viết báo cáo giải thích cách có đáp án các câu hỏi.**

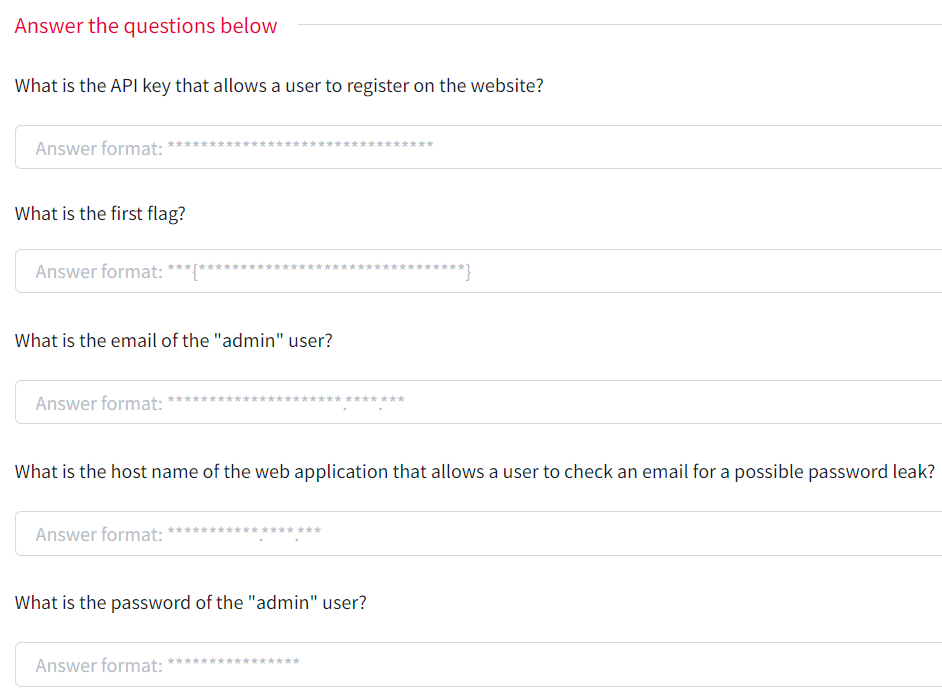
* Thông tin về máy chứa lỗ hổng:



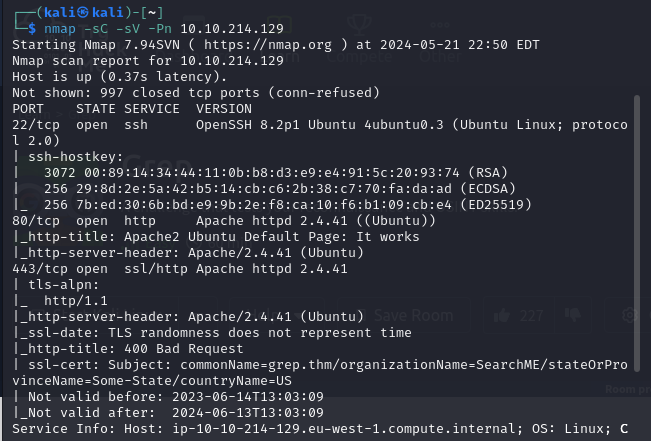
* Sử dụng máy ảo và OpenVPN để kết nối đến được máy tấn công:



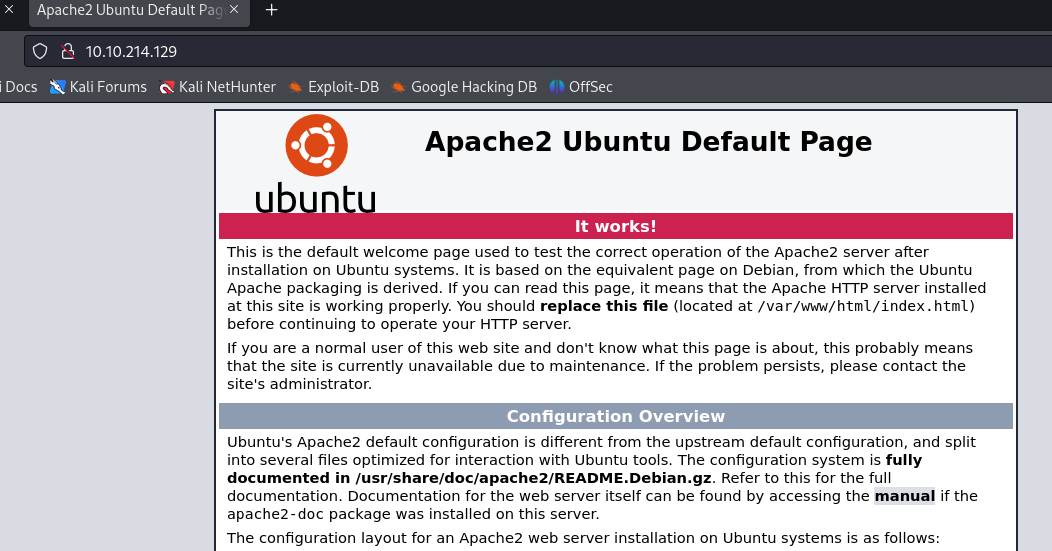
Nhiệm vụ: Trả lời các câu hỏi:



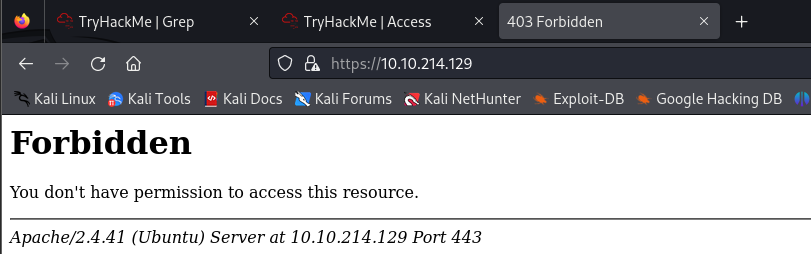
* **What is the API key that allows a user to register on the website?**
* Sử dụng nmap để scan port, có 3 ports đang mở là 22, 80 và 443:



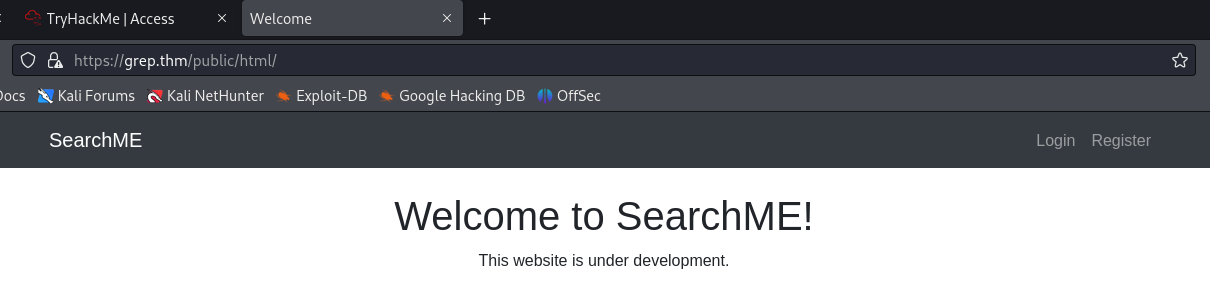
* Truy cập port 80, chỉ hiện ra một trang mặc định:



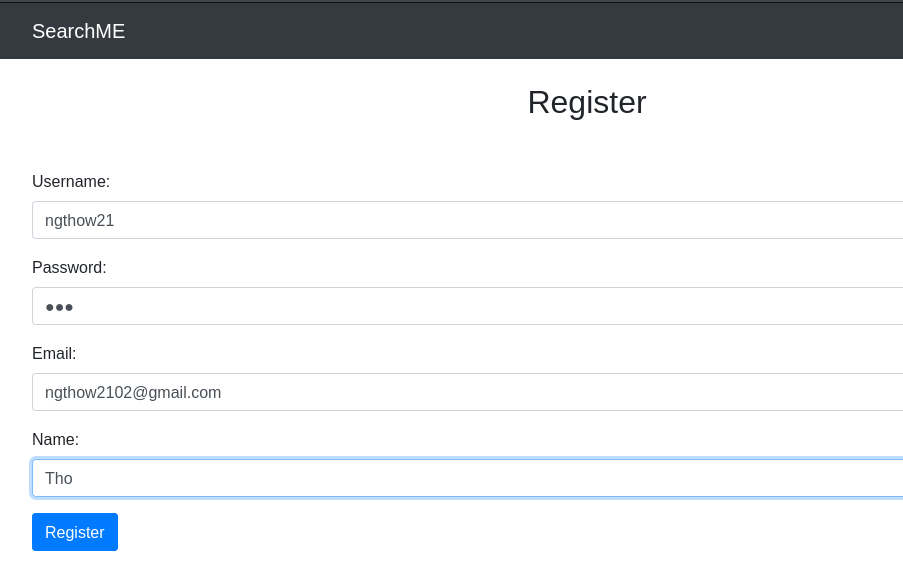
* Truy cập vào port 443 nhưng:



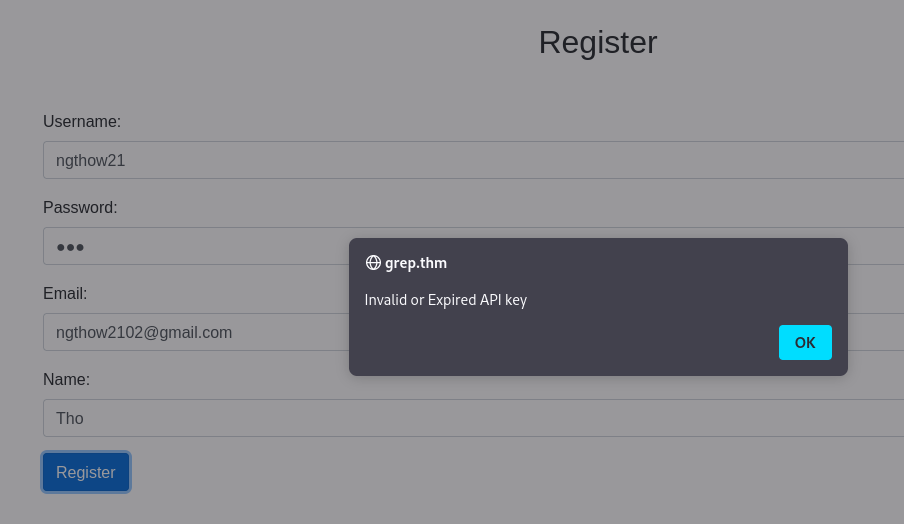
* Bây giờ cần thêm tên miền này vào tệp máy chủ của máy và sau đó thử truy cập URL trên http và https:



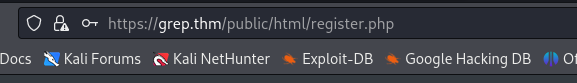
* Tiến hành điền thông tin đăng ký tài khoản:



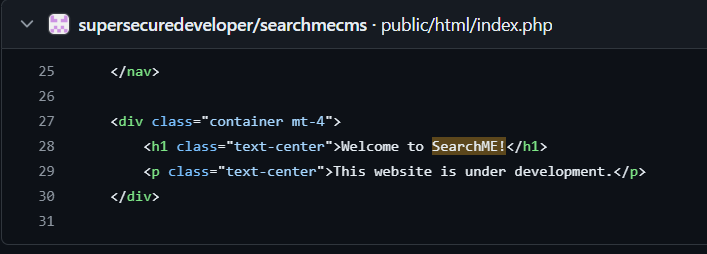
* Nhưng lỗi liên quan đến API key đã hết hạn hoặc không xác định được. Việc tiếp theo cần làm là tìm API key:



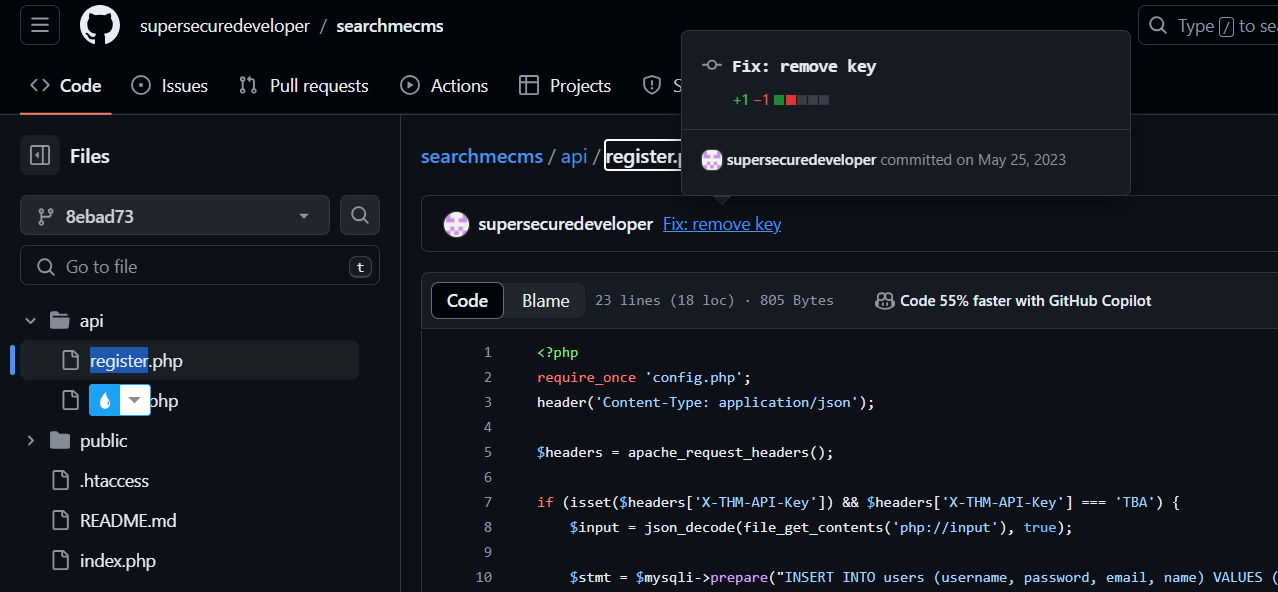
* Trang hiện tại được xây dựng bằng PHP:



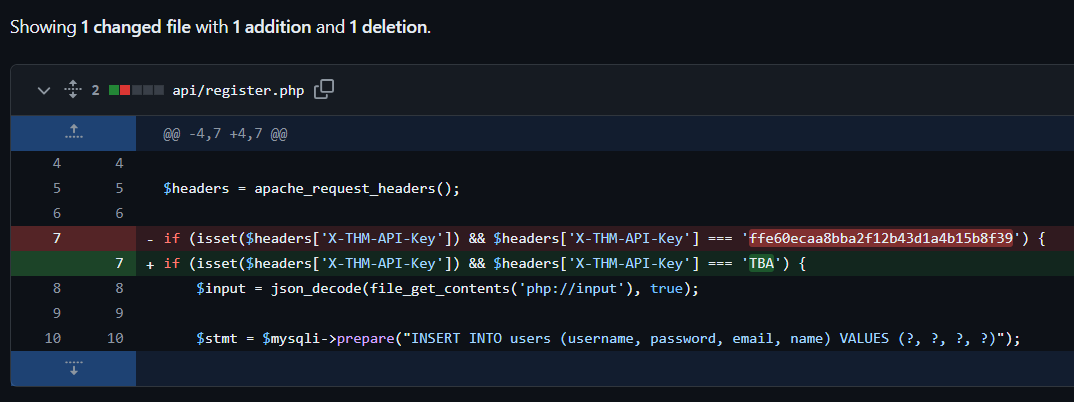
Hiện nay source web có xu hướng được lưu trữ trên GitHub để thuận tiện cho việc sửa đổi và cập nhật phiên bản giữa các thành viên trong nhóm. Tuy nhiên không phải source nào cũng public để người ngoài có thể xem được. Nhưng em vẫn sẽ tìm kiếm thử bằng GitHub với từ khóa “SearchME!”, ưu tiên xem PHP:



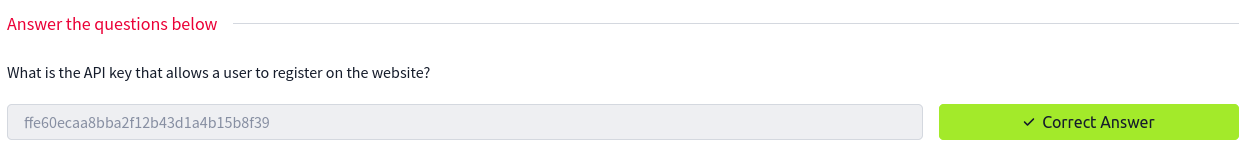
* Tại đây, tìm được commit dùng để fix lỗ hổng để lộ key trong mã nguồn:



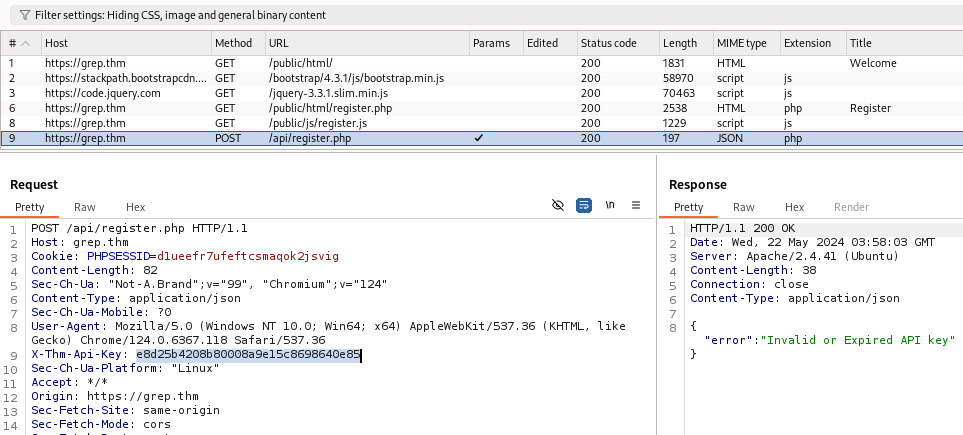
Nhấn vào để xem chỉnh sửa chi tiết, người ta đã xóa API đi và thay bằng *TBA*. API key bị xóa chính là đáp án mà chúng ta cần tìm:

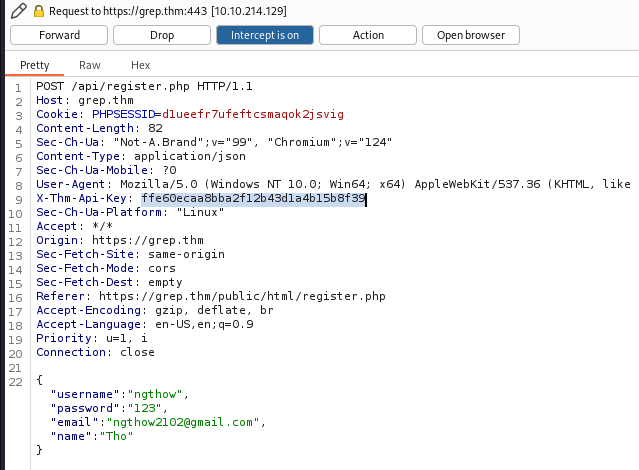


* Kết quả: thành công:

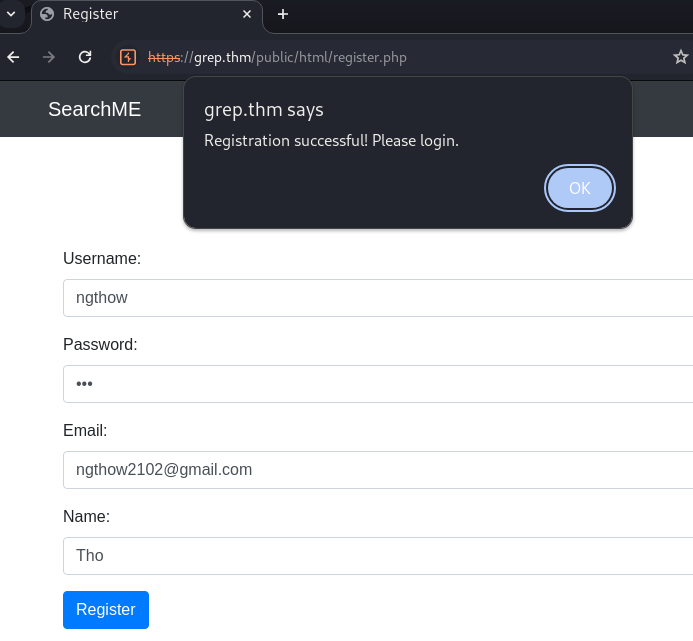


* **What is the first flag?**
  + Tiếp tục quá trình trên, em sẽ dùng Burp Suit để thay đổi API key thành giá trị mà em vừa tìm được:

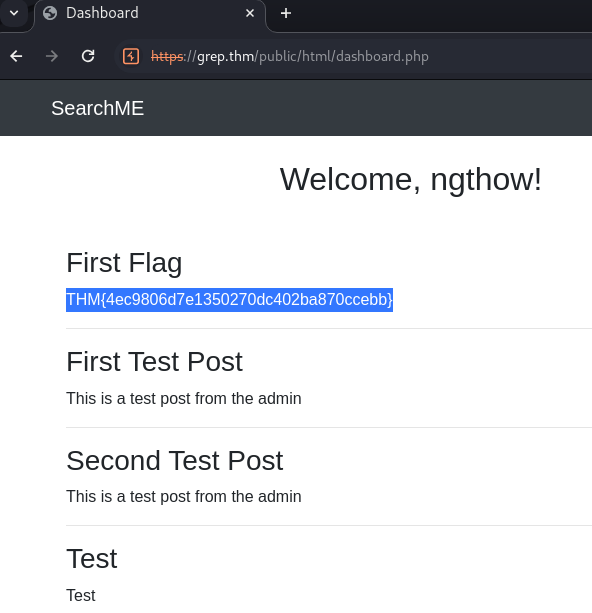


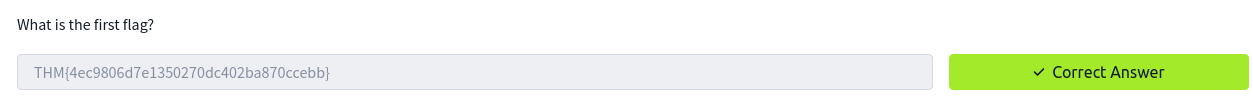


* + Forward và đăng ký thành công:

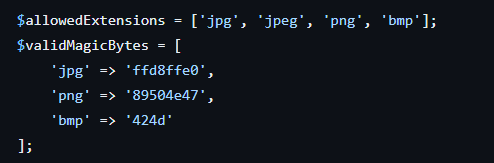


* + Login với tài khoản vừa đăng ký, tại trang dashboard, em tìm được flag:



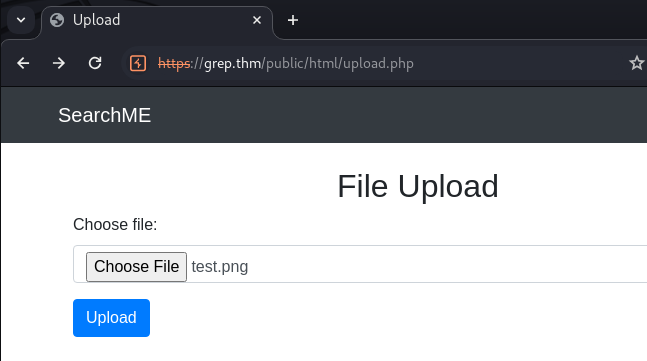


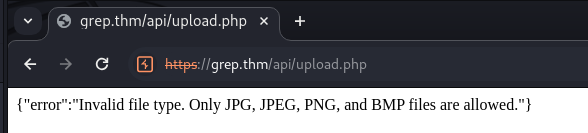
* **What is the email of the "admin" user?**
  + Ở trang trên chỉ có 2 trang là Dashboard và Logout, do đó em nghĩ sẽ không khai thác được nữa. Quay lại với thông tin tìm được trên GitHub, còn một trang chưa được dùng đến là *upload.php*. Quan sát source:



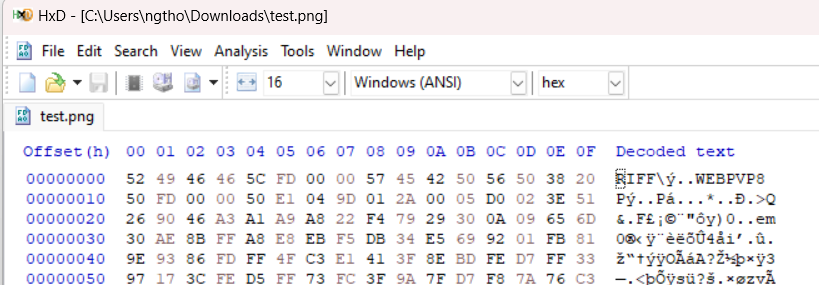
Web nhận vào 3 định dạng file ảnh là jpg, png và bmp với mã hex tương ứng.

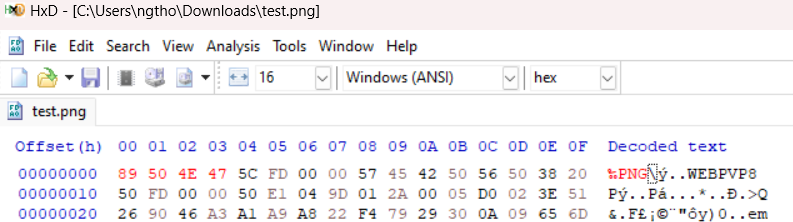
* + Điều hướng đến trang này, em upload một file png ngẫu nhiên nhưng vẫn bị lỗi:



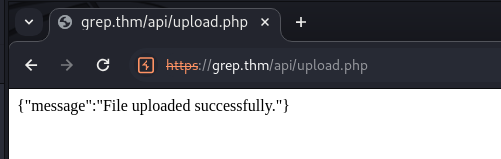


Có thể là file ảnh đã không có mã hex đúng với định dạng được cấu hình trong source nên web không nhận. Do đó, em sẽ tìm cách để chỉnh sửa giá trị hex cho đúng. Em dùng HxD để xem mã hex. Giá trị ban đầu như sau:

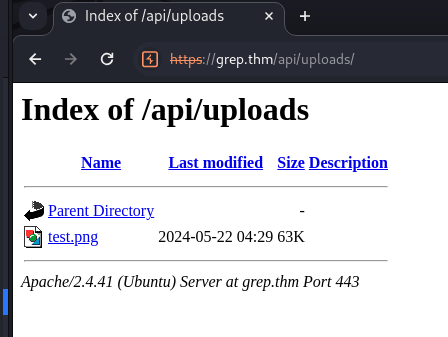
Chỉnh lại:



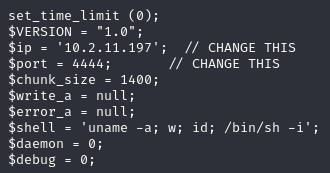
* + Upload lại ảnh vừa chỉnh hex, đã thành công:



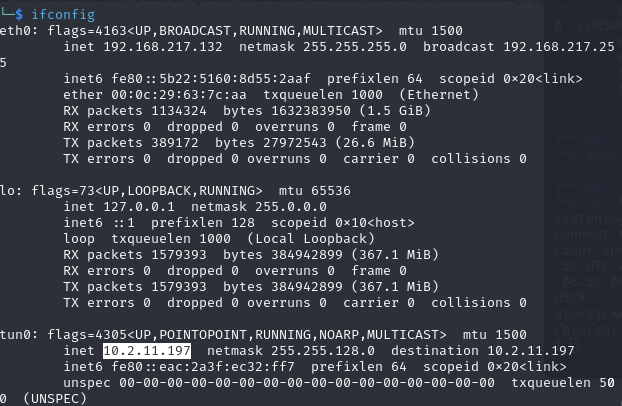
* + Tiếp tục, điều hướng đến thư mục *uploads* và xem ảnh vừa được tải lên:



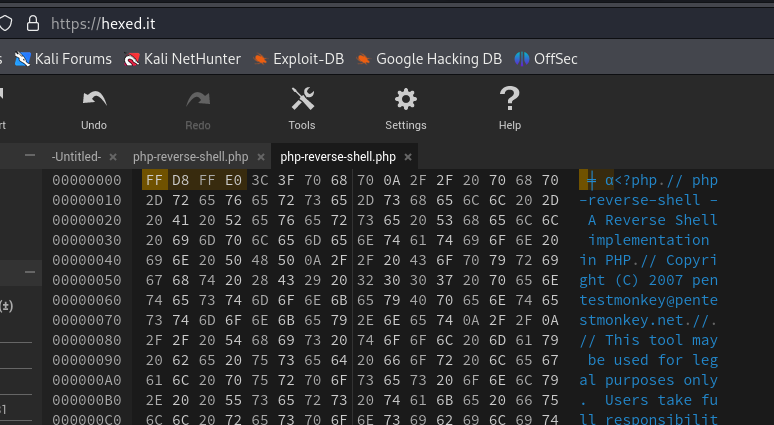
* + Lỗ hổng của web này là việc không thực sự kiểm tra file, chỉ cần có mã hex giống với các định dạng được thiết lập sẵn sẽ bypass được. Em sẽ dùng php reverse shell để chiếm quyền truy cập vào shell của máy nạn nhận: <https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php?source=post_page-----7589bcb72922-------------------------------->



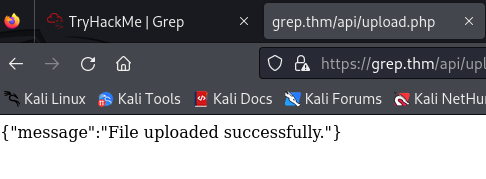
Thay đổi IP thành là IP của máy tấn công:



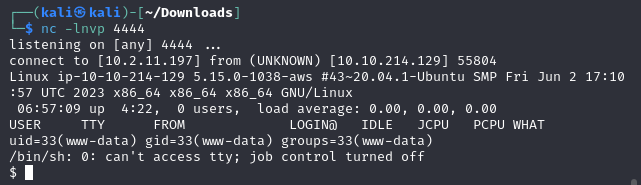
* + Dùng *hexed.it* để chỉnh sửa (vì định dạng lạ nên máy thật Windows của em không cho phép phân tích bằng HxD, em dùng Hex Edit để thay thế). FFD8FFE0 là hex của JPG:



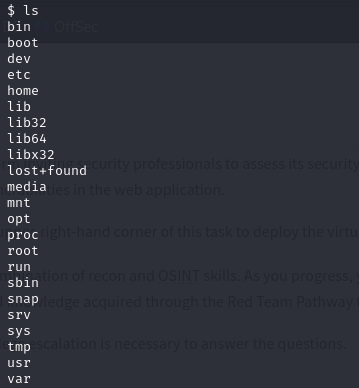
* + Kiểm tra lại thư mục, đã tìm thấy file chứng tỏ upload thành công:



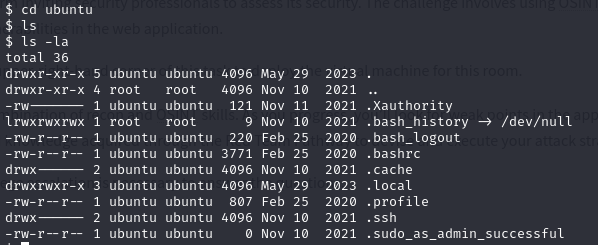
* + Netcat có thể được sử dụng để tạo reverse shell, cho phép mở một shell từ xa trên một máy tính mục tiêu. Do đó, em sẽ dùng netcat để mở shell trên máy nạn nhân. Sau khi upload file, bật netcat và click vào file php vừa upload, ta có thể mở được shell:



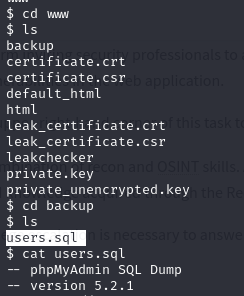
* + Có nhiều thư mục tồn tại trong đây:



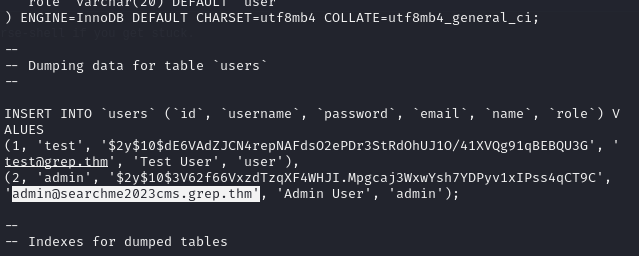
* + Sau khi lần lượt xem các thư mục, tại *home*, em tìm được *ubuntu* và *tryhackme,* nhưng *tryhackme* không có gì cả, em sẽ xem xét cái còn lại, có chút gợi ý liên quan đến “admin”, nhưng mà coi thử thì cũng chưa thấy gì cả:



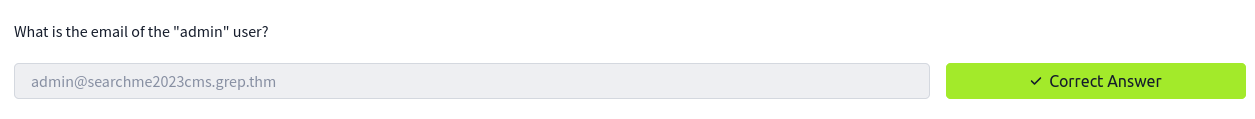
* + Sau khi tìm kiếm thì em tìm được *users.sql:*



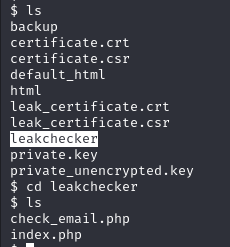
* + Đọc nội dung file, em tìm được email của admin:



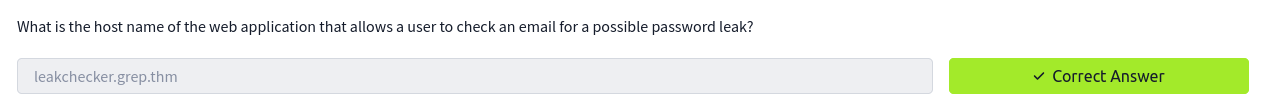
* + Kết quả: chính xác:



* **What is the host name of the web application that allows a user to check an email for a possible password leak?**
  + Yêu cầu của bài này là tìm tên máy chủ của ứng dụng web - cái mà cho phép user kiểm tra email về vấn đề lộ mật khẩu. Tiếp tục bước trên, tại thư mục này, em cũng tìm thấy từ khóa “check” và “leak”:



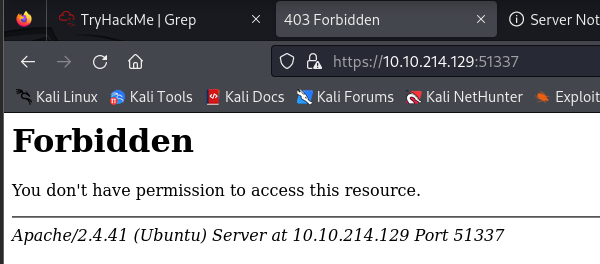
* + 2 file php bên trong *leakchecker* không mở được. Dựa vào answer format của ô điền đáp án, em thấy số lượng ký tự của từ đầu tiên khớp với “leakchecker”, sau khi nhập vào thì đúng thật huhu:



* **What is the password of the "admin" user?**
* Dùng Rustscan để scan port nhanh hơn nmap, tìm được một cổng mở khác trên máy là 51337:

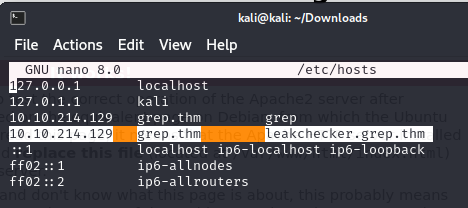


* Truy cập thông qua port này nhưng chưa được:

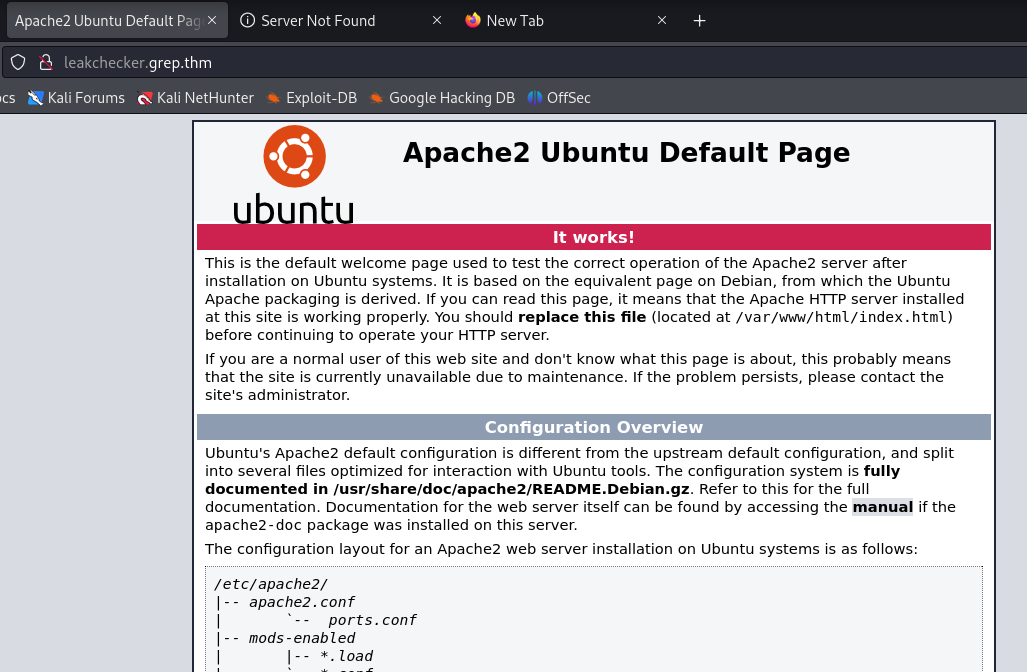


* Bây giờ cần thêm tên miền này vào tệp máy chủ của máy và sau đó thử truy cập URL trên http và https:

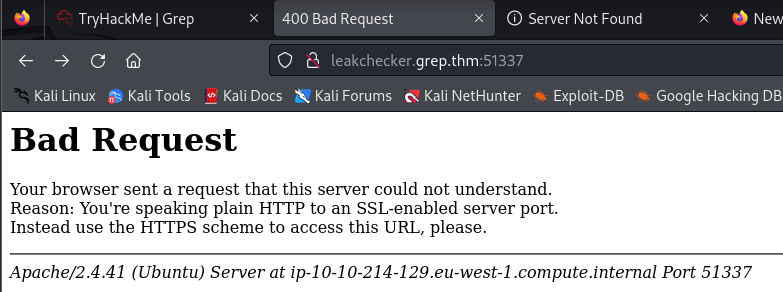




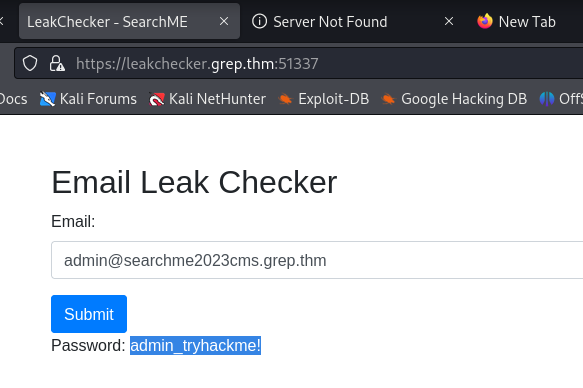
* Hiển thị trang mặc định Apache2 trên http và bị cấm trên https:



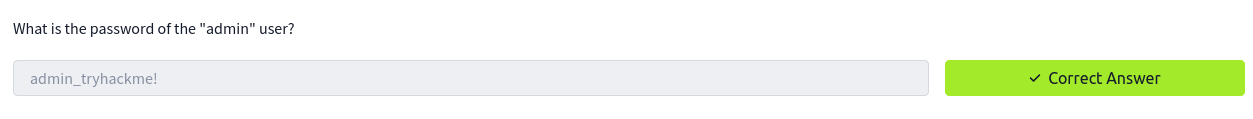
* Mở trên port 51337, http sẽ không hiển thị gì cả. Chuyển sang https sẽ thu được trang cần tìm:



* Đây là trang bị rò rỉ mật khẩu, nên chỉ cần cung cấp email admin, ta sẽ có được password tương ứng:

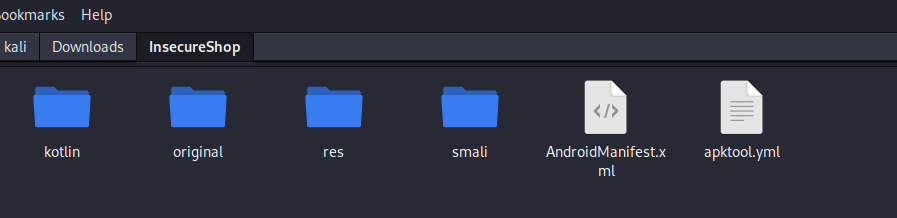


* Kết quả: thành công:

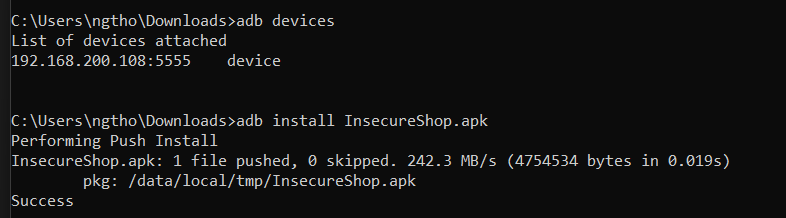


1. **Bài tập 4, 5: Sử dụng ứng dụng android InsecureShop.apk. Báo cáo ít nhất 2 lỗ hổng được tìm thấy.**

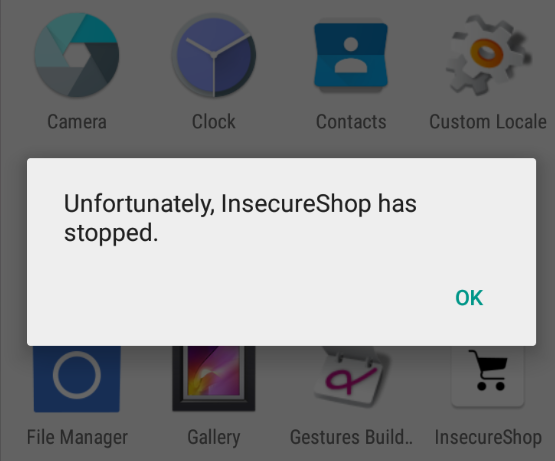
* Sử dụng *apktool* dịch ngược file apk:



* Cài file apk vào thiết bị:



* Nhưng app hiện tại mở không lên được, em cũng không tải lại file apk vì link anh up trên course đã hết hạn:



Do đó, các lỗ hổng em sẽ tìm dựa trên source code.

* **Lỗ hổng 1:**

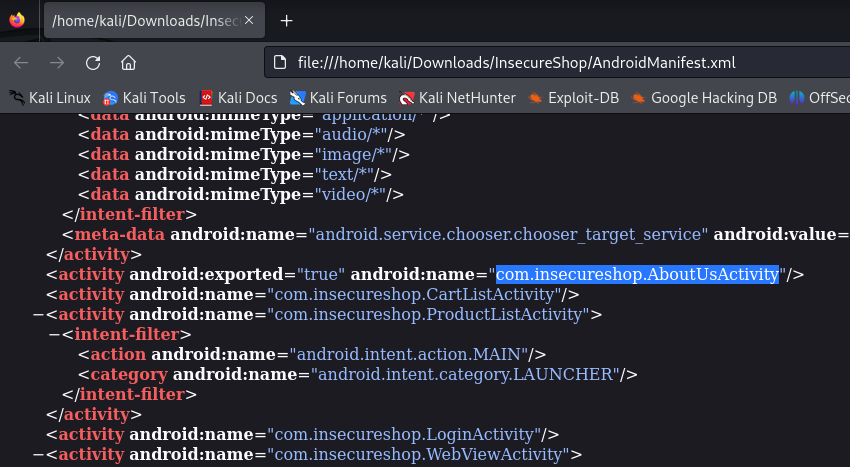
**# Tiêu đề:** Lỗ hổng Activity Exported Attribute trong Android.

**# Mô tả lỗ hổng:**

**### Tóm tắt:** Lỗ hổng này liên quan đến việc cấu hình không chính xác thuộc tính android:exported trong tệp AndroidManifest.xml. Khi thuộc tính android:exported được đặt thành *true*, bất kỳ ứng dụng nào khác trên thiết bị cũng có thể truy cập và khởi chạy Activity đó. Điều này có thể dẫn đến các vấn đề bảo mật nghiêm trọng như truy cập trái phép vào dữ liệu nhạy cảm hoặc thực hiện các hành động không mong muốn trong ứng dụng.

**### Các bước để thực hiện lại và bằng chứng:**

Kiểm tra file *AndroidManifest.xml*:



**### Tài liệu hỗ trợ và tham khảo:**

* <https://cystack.net/blog/vulnerability-management>
* <https://developer.android.com/privacy-and-security/risks/android-exported?hl=vi#:~:text=PLATFORM%3A%20Platform%20Interaction-,Overview,by%20its%20exact%20class%20name>.

**# Mức độ ảnh hưởng của lỗ hổng:**

* Gây rò rỉ thông tin, dữ liệu.
* Ảnh hưởng đến tính toàn vẹn cho thông tin và sẵn sàng cho hệ thống.

**#Khuyến cáo khắc phục:** Đặt thuộc tính android:exported thành *false* cho tất cả các Activity không cần thiết phải được truy cập từ các ứng dụng khác.

* **Lỗ hổng 2:**

**#Tiêu đề:** Lộ thông tin đăng nhập trong mã nguồn chương trình.

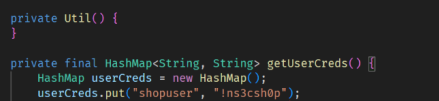
**# Mô tả lỗ hổng:** Lỗ hổng này liên quan đến việc lộ thông tin đăng nhập (bao gồm username và password) trong mã nguồn của ứng dụng. Khi dịch ngược file apk và kiểm tra mã nguồn, thông tin nhạy cảm này có thể được phát hiện trong mã nguồn, dẫn đến nguy cơ bảo mật nghiêm trọng.

**Tóm tắt:**

**### Tóm tắt:** Lộ thông tin đăng nhập trong mã nguồn ứng dụng có thể cho phép kẻ tấn công truy cập trái phép vào các tài nguyên hoặc dịch vụ được bảo vệ bằng các thông tin đăng nhập này. Đây là một vấn đề bảo mật nghiêm trọng cần được khắc phục ngay lập tức. Người ta có thể tận dụng lỗ hổng này để phá hủy hay đánh cắp thông tin, gây ảnh hưởng uy tín doanh nghiệp.

**### Các bước để thực hiện lại và bằng chứng:**

Sau khi dịch ngược và tìm trong source, em tìm được đoạn code chứa thông tin đăng nhập như sau:



**### Tài liệu hỗ trợ và tham khảo:**

* <https://viblo.asia/p/bao-mat-code-android-DXOkRjdVkdZ>

**#Mức độ ảnh hưởng của lỗ hổng:**  nghiêm trọng do có thể dẫn đến leo thang đặc quyền.

**#Khuyến cáo khắc phục:**

* Sử dụng proguard có sẵn trong android.
* Loại bỏ thông tin đăng nhập khỏi mã nguồn ngay lập tức.
* Mã hóa thông tin.
* Sử dụng AWS Secrets Manager, Azure Key Vault để tăng tính an toàn cho việc quản lý cấu hình.

---

***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 .DOCX và .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 đã đăng ký với GVHD-TH).

*Ví dụ: [*NT101.K11.ATCL*]-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 bài Lab.
* 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 được bài thực hành. Khuyến khích:

* Chuẩn bị tốt.
* Có nội dung mở rộng, ứng dụng trong kịch bản phức tạp hơn, có đóng góp xây dựng bài thực hành.

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

**HẾT**

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-1)