**REPORT**

**Mục lục**

1. **Scan host live**
2. **Server web01**
3. **Server web02**
4. **Mail Server – web03**

Danh sách hình ảnh

Hình 1. SCAN------------------------------ đang thiếu

Hình 2. Giao diện của web01 (đang thiếu)

Hình 2.1. Scan ffuf trên web01

Hình 2.2. Scan ffuf đối với wp-content

Hình 2.3. Nội dung của wp-content/upgrade

Hình 2.4. File Readme.md lấy được trong wp-content/upgrade

Hình 2.5. Webshell được tìm thấy trong file zip wp-content/upgrade/plugin

Hình 3.6. Link đến webshell theo đường dẫn như file zip và thực thi webshell

Hình 3.7. Lấy flag dựa trên câu lệnh hướng dẫn trong file README.MD

Hình 3. Giao diện của web02 (đang thiếu)

Hình 3a. Github của TravelLog

Hình 3b. Database lấy được từ Github

Hình 3.1. Dùng ffuf để liệt kê directory

Hình 3.2. Ffuf được một đường dẫn lạ là test.php trong /php

Hình 3.3. Crack đoạn mã bcrypt lấy được từ test.php

Hình 3.4. Test SQLi bằng công cụ sqlmap trên đường dẫn /php/login.php

Hình 3.5. Khai thác lỗi SQLi trên công cụ BurpSuite

Hình 3.6. Crack đoạn mã bcrypt của user "admin"

Hình 3.7. Trang thông tin của web02 sau khi login vào được

Hình 3.8. Ffuf được một directory của /php là uploads

Hình 3.9. Kiểm tra thông tin trên chế độ Inspect của firefox

Hình 3.10. Tìm được điểm khai thác

Hình 3.11. Tạo file php shell dựa trên trang web revshells.com

Hình 3.12. Tiến hành khai thác

Hình 3.13. Upload thành công hình ảnh dưới dạng php shell

Hình 3.14a. File php shell đã không còn lỗi (thiếu ảnh)

Hình 3.14b. Thực thi thành công php shell (thiếu ảnh)

Hình 3.15. Tải netcat về máy web02 để reverse về máy attacker (đang thiếu)

Hình 3.16-17. Chạy script .bat để tìm các host khác

Hình 3.18. Giao diện của FourEye

Hình 3.19. Tiến hành mã hóa 2 lần file payload .exe

Hình 3.20-21. Scan file payload đã mã hóa trên Virustotal

Hình 3.22. Thành công bypass AV để chạy file payload trên máy web02

Hình 3.23. Bắt session meterpreter thành công

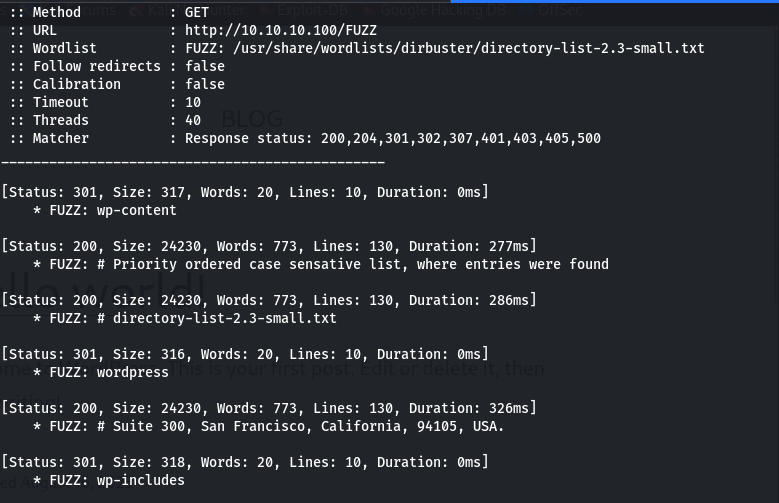
Hình 3.24. Kiểm tra quyền đang được bật

Hình 3.25. Leo quyền thành công lên "NT AUTHORITY\SYSTEM"

Hình 3.26. Thành công lấy được flag

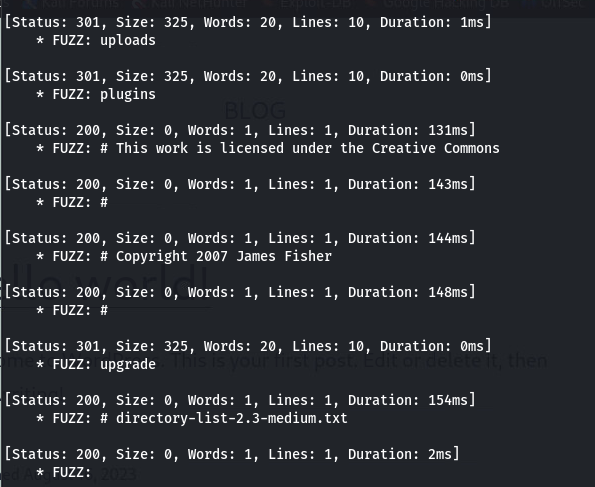
1. **Scan**
2. **Server Web01**

* Sau khi đã xác định host này là web sử dụng wordpress thì việc đầu tiên em làm là em sẽ dùng công cụ ffuf để xem có những directory nào tiềm năng để khai thác
* Sau khi dùng công cụ ffuf để scan thì em thu được các directory sau:
  + Wp-content
  + Wordpress
  + Wp-includes



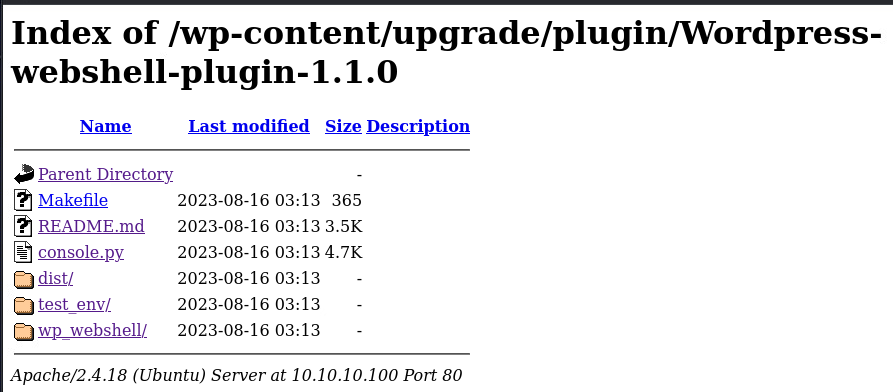
*Hình 2.1* *. Scan ffuf trên web01*

* Sau khi đã xác định được các directory thì em sẽ vào để xem có gì để khai thác không nhưng không thấy gì nên em sẽ tiếp tục dùng ffuf để scan tiếp xem trong wp-content còn gì nữa không
* Sau khi dùng ffuf để tiếp tục scan thì em nhận được các directory sau:
  + Uploads
  + Plugins
  + Upgrade
* Tiếp theo, em tiến hành vào /wp-content/uploads để xem qua 1 lượt nhưng chưa thấy gì khả nghi để khai thác nên em chuyển qua cái khác là wp-content/upgrade



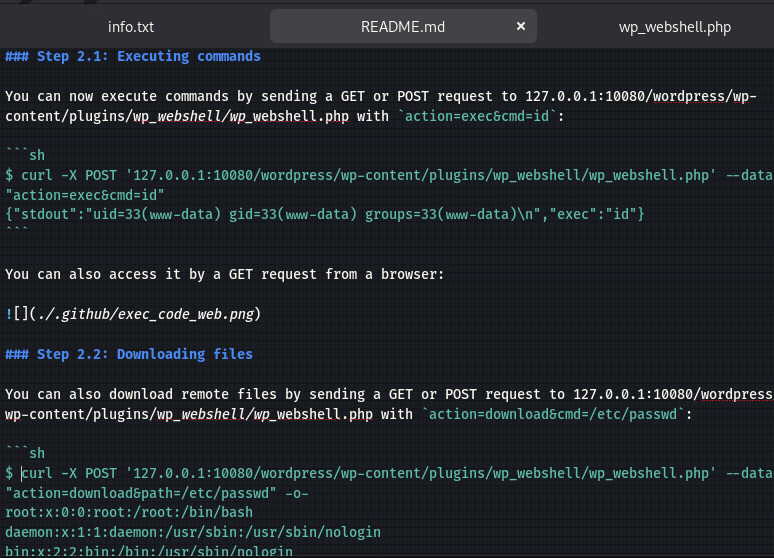
*Hình 2.2* *. Scan ffuf đối với wp-content*

* Khi vào wp-content/upgrade thì em thấy thư mục /plugin/Wordpress-webshell-plugin-1.1.0
* Sau khi vào thì em nhận được các file hữu ích như: README.md, console.py và có 1 thư mục đáng ngờ là wp\_webshell
* Nhưng khi em nhấp vô wp\_webshell thì trang web lại tự động back về /plugin



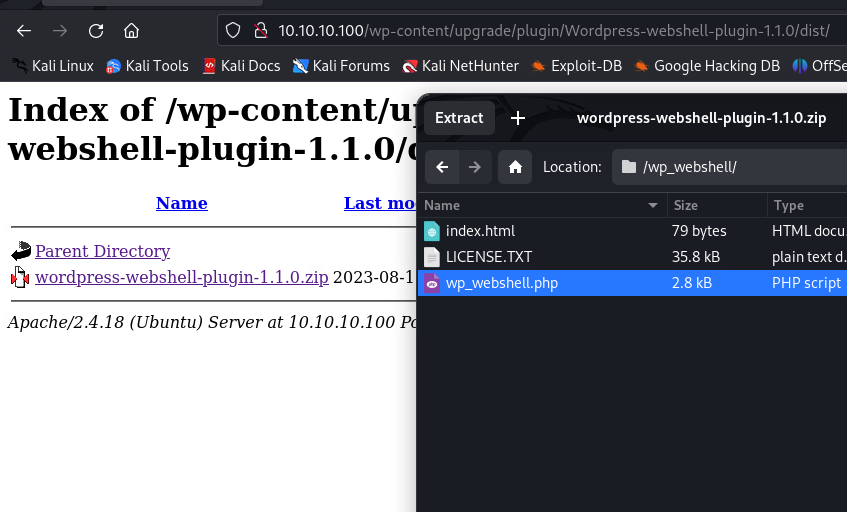
*Hình 2.3* *. Nội dung của wp-content/upgrade*

* Em thử tải file README.md về và mở lên để đọc thì thấy file hướng dẫn cách chạy wp\_shell.php thủ công và cách dùng file console.py để kết nối đến



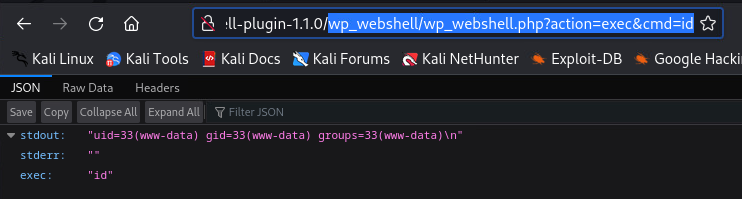
*Hình 2.4* *. File Readme.md lấy được trong wp-content/upgrade*

* Vì khi nãy em nhấp vào wp\_webshell bị trả ngược về /plugin nên em chưa biết khai thác tiếp tục sao đối với thư mục wp\_webshell
* Sau đó em thử vô các thư mục còn lại ngoài cái wp\_webshell là /dist thì em thấy có một file .zip trong đó nên em thử tải về để xem thử
* Sau khi tải về và mở file lên thì em thấy được các thư mục giống với các thư mục trong đường dẫn: wp-content/upgrade/plugin/ Wordpress-webshell-plugin-1.1.0
* Và có thư mục wp\_webshell nên khi em vào thì thấy trong thư mục wp\_webshell có chứa 1 file: wp\_webshell.php
* Từ đây em suy luận ra là trên web theo đường dẫn: wp-content/upgrade/plugin/ Wordpress-webshell-plugin-1.1.0/wp\_webshell mà khi nãy em nhấp vào bị back về /plugin cũng có thể chứa 1 file wp\_webshell.php



*Hình 2.5* *. Webshell được tìm thấy trong file zip wp-content/upgrade/plugin*

* Nên em quyết định nhập đường dẫn trực tiếp lên thay vì cố nhấn vào thư mục wp\_webshell và kèm theo lệnh để thực thi wp\_webshell như hướng dẫn của file README.md là action=exec&cmd=id, đường dẫn đầy đủ là:
  + wp-content/upgrade/plugin/ Wordpress-webshell-plugin-1.1.0/wp\_webshell/wp\_webshell.php?action=exec&cmd=id
* Sau khi em nhập đường dẫn trên thì đã thực thi được file wp\_webshell.php thành công



*Hình 2.6* *. Link đến webshell theo đường dẫn như file zip và thực thi webshell*

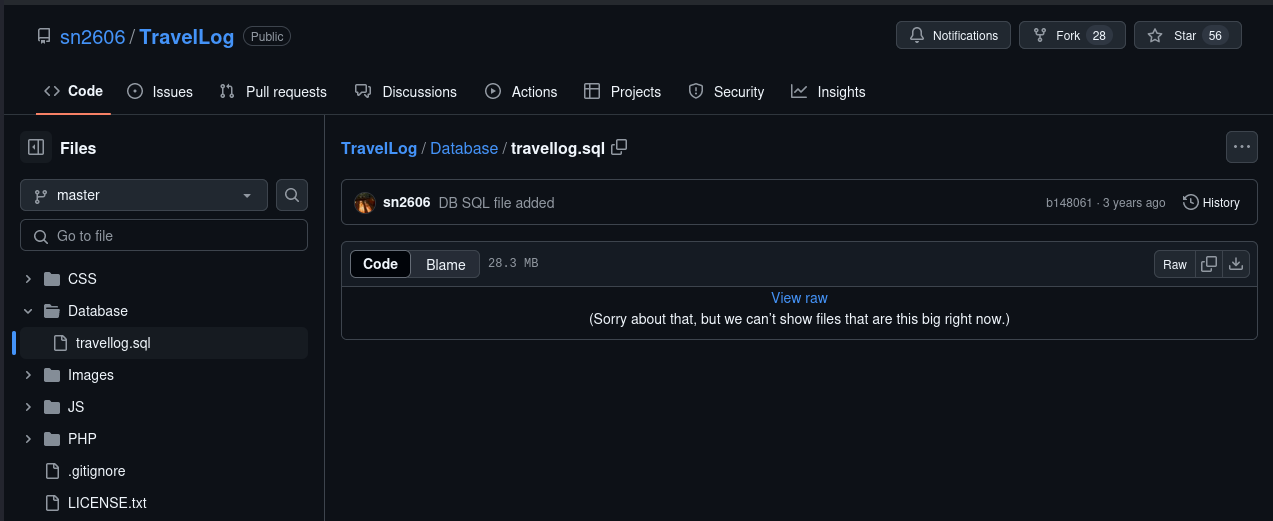
* Sau đó em nhập theo hướng dẫn của file README.md bằng cách dùng công cụ curl và em đã lấy được flag thành công



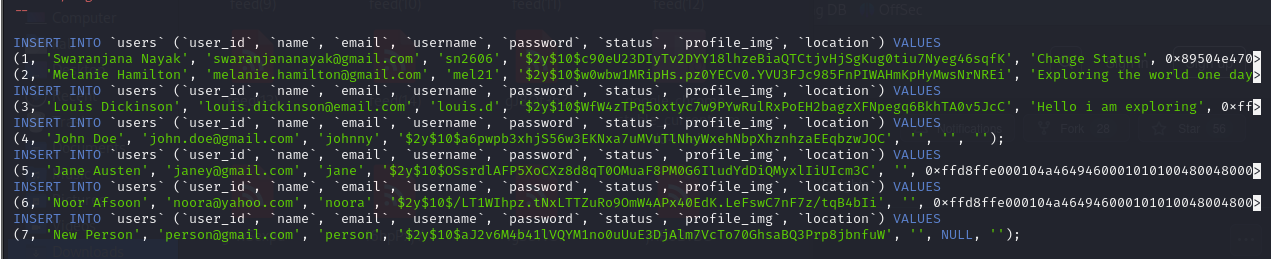
*Hình 2.7* *. Lấy flag dựa trên câu lệnh hướng dẫn trong file README.MD*

1. **Server Web02**

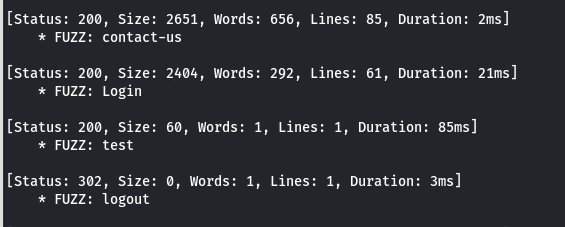
* Ở web02 này em vẫn tiếp tục dùng công cụ ffuf để scan các directory và các đường dẫn .php ẩn, sau đó em thu được các directory sau:
  + Contact-us
  + Login
  + Test
  + Logout
  + ….
* Kèm theo đó là source code và database của trang TravelLog dựa trên đường link giới thiệu trên trang web02



*Hình 3a. Github của TravelLog*

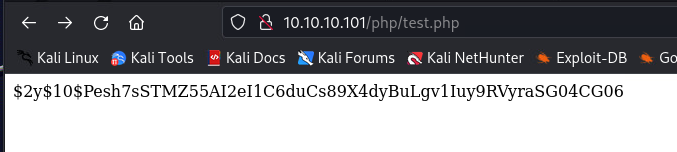
**

*Hình 3b. Database lấy được từ Github*



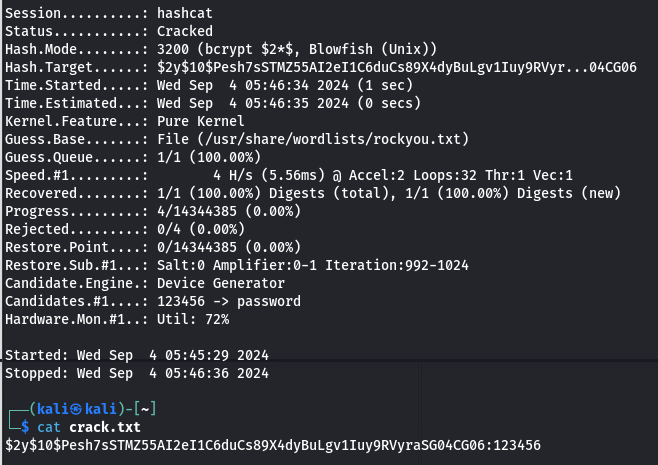
*Hình 3.1. Dùng ffuf để liệt kê directory*

* Em có nhận thấy 1 đường dẫn lạ là test.php nên khi em vào thì nhận được 1 đoạn mã hóa bcrypt



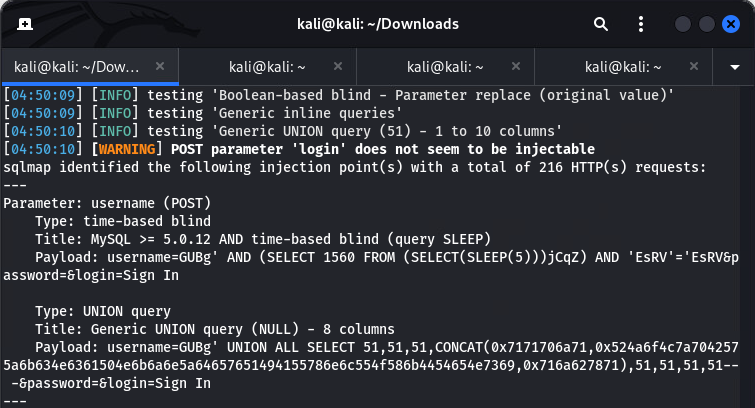
*Hình 3.2. Ffuf được một đường dẫn lạ là test.php trong /php*

* Khi em cố gắng crack nó bằng hashcat thì em nhận được kết quả là: 123456



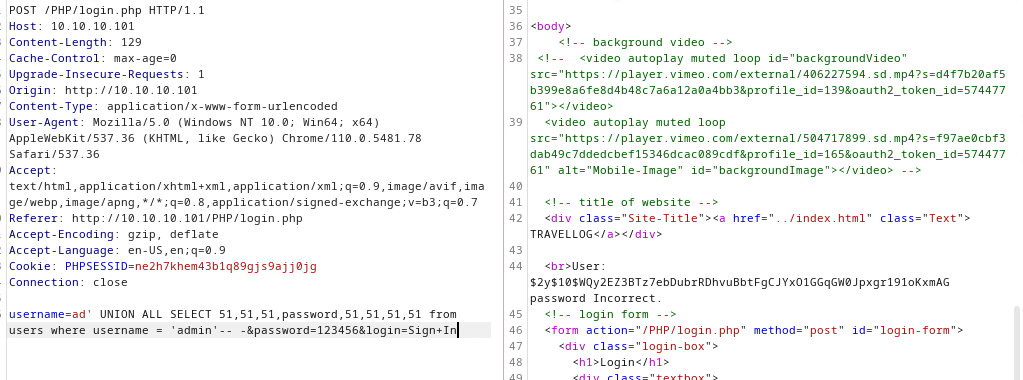
*Hình 3.3. Crack đoạn mã bcrypt lấy được từ test.php*

* Sau khi đã làm đủ mọi cách với đoạn này và em nhấn F5 lại đối với trang /php/test.php thì thấy mã bcrypt thay đổi liên tục nhưng nếu cố crack thì vẫn là “123456” nên em suy ra có thể đây là giả



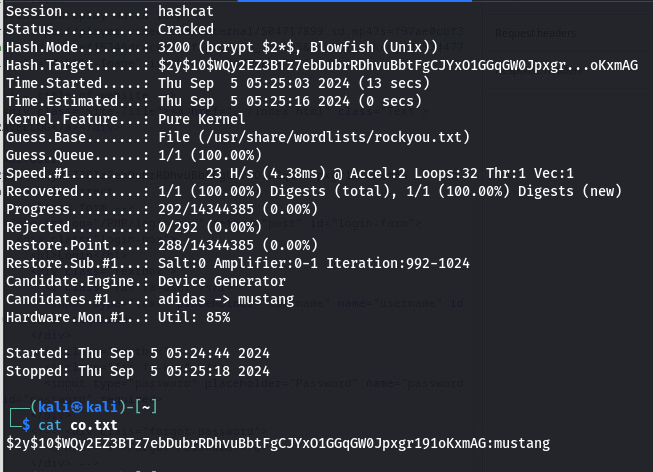
*Hình 3.4. Test SQLi bằng công cụ sqlmap trên đường dẫn /php/login.php*

* Tiếp theo, em dùng các cách của SQLi để cố đăng nhập ví dụ: admin’-- hoặc admin’ or ‘1’=’1 nhưng thất bại nên em thử dùng sqlmap để tự động khai thác lỗi SQLi trên trang login.php
* Khi chạy xong sqlmap thì nó đã phát hiện ra bug SQLi ở tham số “username” và có đưa ra đoạn mã để thực thi
* Em copy đoạn mã đó vào công cụ BurpSuite và sửa đổi đoạn mã cho hợp lí hơn để khai thác
* Vì sqlmap đã hỗ trợ em xác định nhanh được là cột thứ 4 trong 8 cột có thể in ra chữ và em sẽ bắt đầu khai thác từ đây
* Em đã khai thác được toàn bộ username có trong database bằng cách loại trừ từng username mà nó đã in ra trước đó và so sánh với các username có trong database sẵn có mà em thu thập được trên Github, em không thể crack được password từ file .sql trên Github nhưng có 1 user ngoại lệ đó là “admin”
* Nên em suy nghĩ thay vì list username để cố tìm username đáng nghi thì em sẽ list password của riêng user “admin”
* Và em đã thu thập được đoạn mã bcrypt của user “admin”



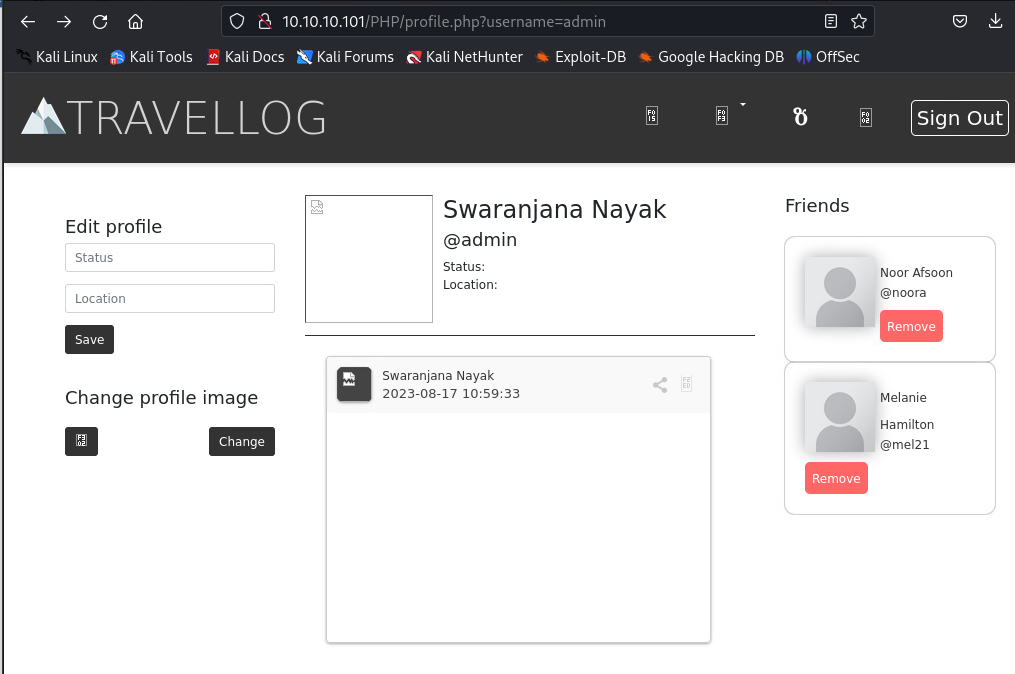
*Hình 3.5. Khai thác lỗi SQLi trên công cụ BurpSuite*

* Sau khi đã có được đoạn mã trên thì em sẽ dùng công cụ hashcat để cố gắng crack nó từ file wordlist rockyou.txt
* Em đã crack thành công đoạn mã trên và nhận được mật khẩu của user “admin” là **mustang**



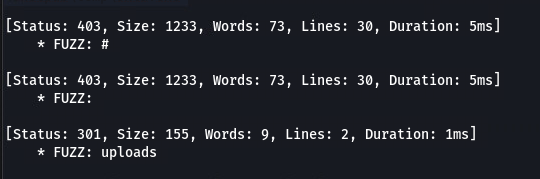
*Hình 3.6. Crack đoạn mã bcrypt của user "admin"*

* Em tiến hành login vào và đây là giao diện của web TravelLog khi login



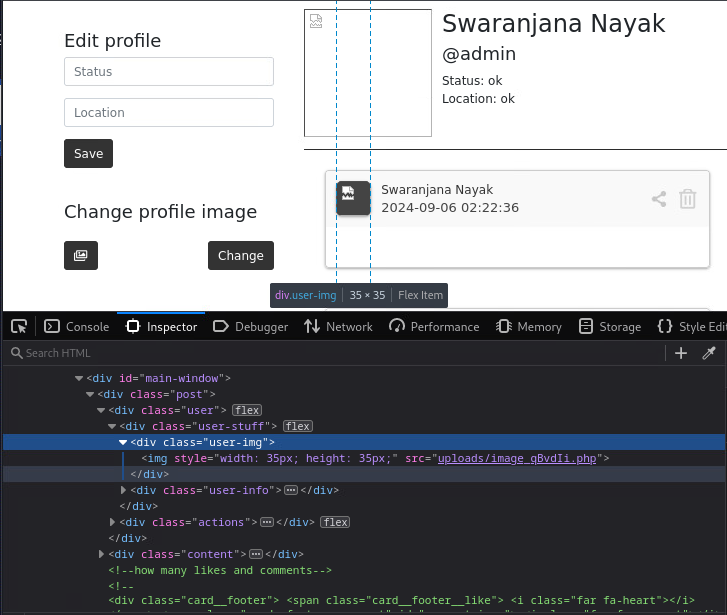
*Hình 3.7. Trang thông tin của web02 sau khi login vào được*

* Trước đó khi em dùng ffuf để scan thì có nhận được 1 thư mục ẩn đáng nghi là /uploads



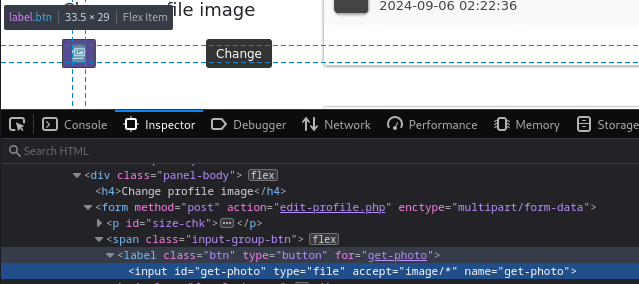
*Hình 3.8. Ffuf được một directory của /php là uploads*

* Em đã login vào được thì em dò các chức năng có trên web để tìm kiếm xem thư mục uploads đó từ đâu và làm sao để có thể thêm được file vào thư mục uploads
* Sau khi đã dò qua các chức năng bằng Inspect của firefox thì em đã thấy được đường dẫn giống như trong ảnh là uploads/image\_xxxxx.png và nó nằm ở **class=user-img**
* Từ đó em có thể suy ra là thay đổi ảnh profile thì ảnh đó sẽ được lưu ở đường dẫn /uploads

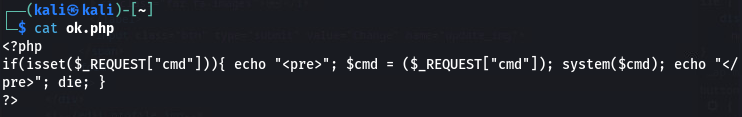


*Hình 3.9. Kiểm tra thông tin trên chế độ Inspect của firefox*

* Khi đã xác định được nơi mong muốn để tải payload lên thì em tiếp tục kiểm tra qua chỗ Upload ảnh profile để em em có thể upload những thể loại file nào thông qua đoạn **accept=”image/\*”**
* Vì là dấu \* nên em nghĩ là up được toàn bộ loại file nên em sẽ tạo 1 file shell php như hình 3.11 để thử up lên

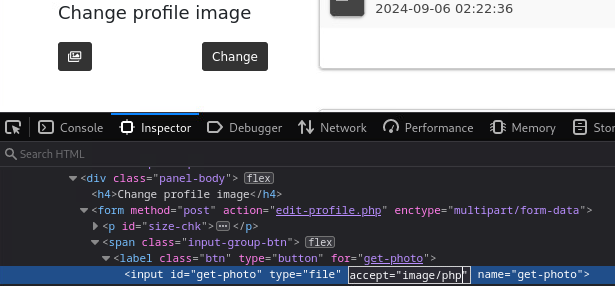


*Hình 3.10. Tìm được điểm khai thác*



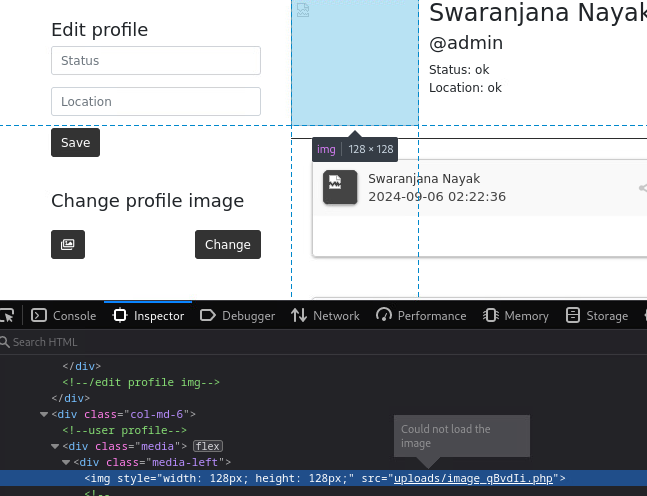
*Hình 3.11. Tạo file php shell dựa trên trang web revshells.com*

* Nhưng khi em upload ảnh dưới dạng **.php** thì em thấy nó báo lỗi như hình 3.14a nên em sẽ thử đổi từ **“image/\*”** thành **“image/php”** xem nó có hoạt động được không



*Hình 3.12. Tiến hành khai thác*

* Sau khi em đã chỉnh về image/php thì em thấy nó đã upload thành công và không còn bị lỗi nữa nên đường dẫn **upload/image\_xxxx.php** cũng không còn bị lỗi như hình 3.14a



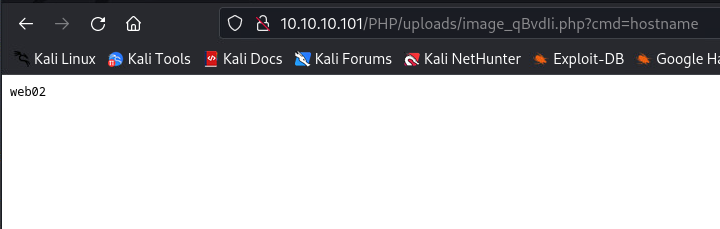
*Hình 3.13. Upload thành công hình ảnh dưới dạng php shell*

Ở đây chèn ảnh bị lỗi khi dùng đường dẫn upload

*Hình 3.14a. Accept = “image/\*”*

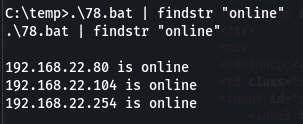
Ở đây chèn ảnh chỉ chạy mỗi file shell không có dòng cmd=hostname **(chèn ảnh này trước)**

*Hình 3.14b. Accept = “image/php”*

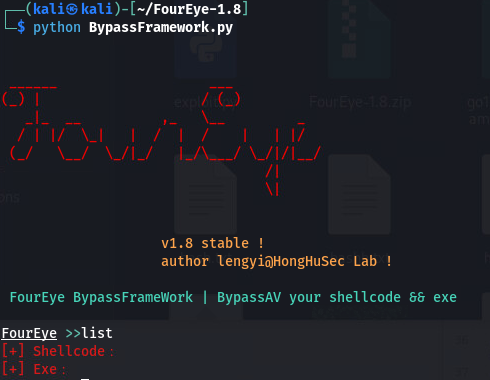


*Hình 3.14c. Thực thi thành công php shell*

* Sau khi đã có reverse shell về máy attack thông qua netcat thì em tải file script .bat lên để scan host live trên 2 dãi mạng còn lại

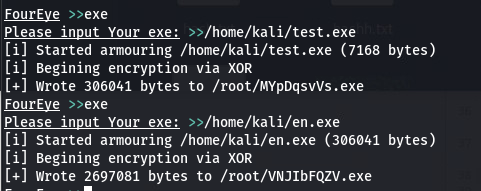
  
*Hình 3.16. Chạy script .bat để tìm các host khác*

* Tiếp theo em muốn bắt được session meterpreter để phục vụ cho việc leo quyền nhưng khi tải file payload lên đều bị xóa mất do AntiVirus đã phát hiện ra file mã độc và tự động xóa nó đi
* Để có thể chạy được file payload trên máy web02 thì em phải tìm cách bypass AV
* Sau khi lên mạng tìm hiểu cách bypass AV thì em tìm được công cụ **FourEye** hỗ trợ mã hóa file exe để có thể giúp bypass AV

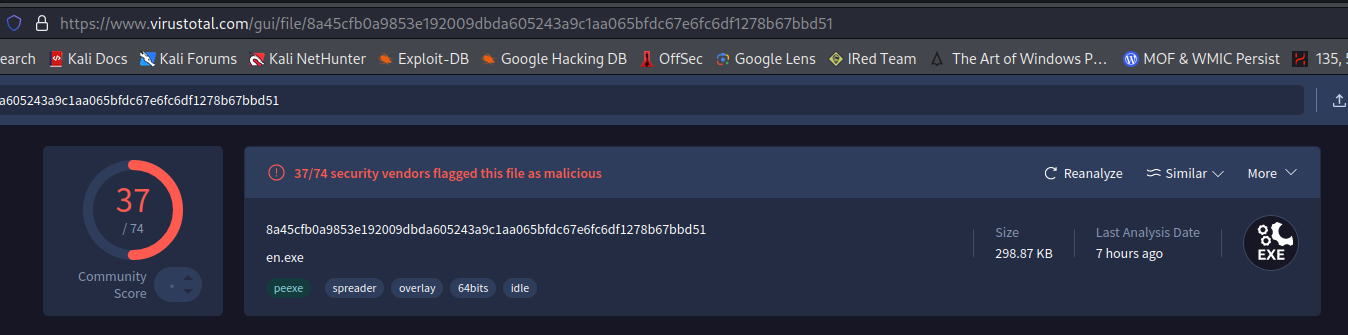


*Hình 3.18. Giao diện của FourEye*

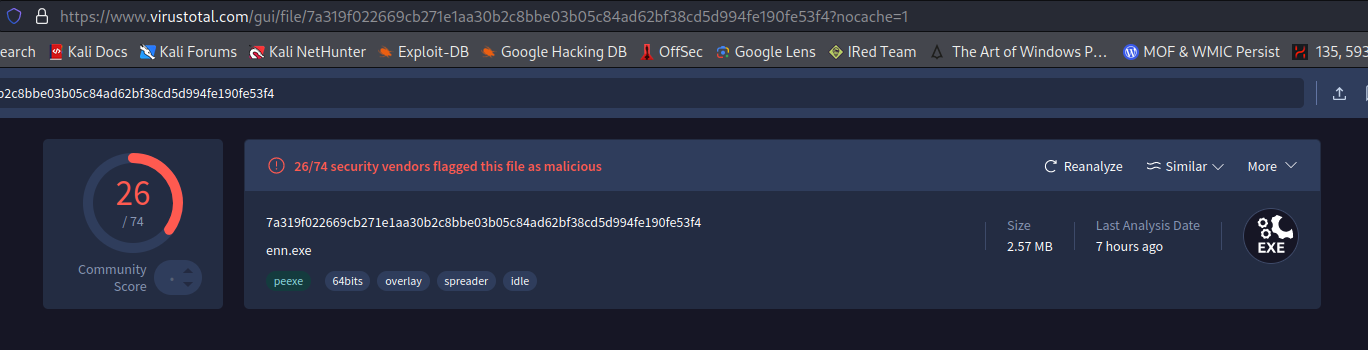
* Em tiến hành tạo file payload bằng msfvenom và đem vào công cụ FourEye để mã hóa
* Trước khi đem vào công cụ FourEye để mã hóa thì file payload được tạo từ msfvenom được trang **Virustotal** phát hiện **55/74**
* Sau khi mã hóa thì em nhận được file payload khác em đã đổi tên thành **en.exe** và scan lại trên **Virustotal** phát hiện **37/74 như hình 3.20**
* Vì vẫn bị AV phát hiện và xóa file nên em quyết định mã hóa thêm 1 lần nữa từ file đã mã hóa trước đó là **en.exe** và đổi tên thành **enn.exe**
* Sau khi đã mã hóa lần 2 thì em lên trang **Virustotal** để scan lại thì chỉ còn **26/74 như hình 3.21** và không còn bị AV phát hiện
* Sau 2 lần mã hóa thì em đã bypass được AV trên máy web02 và thực thi file payload thành công nhận được session meterpreter



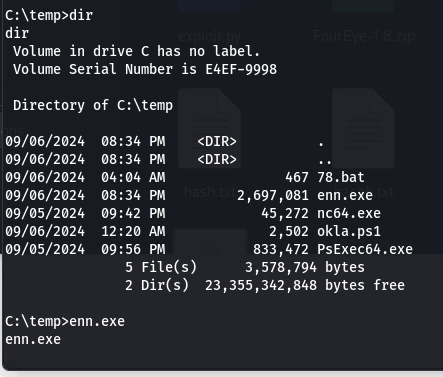
*Hình 3.19. Tiến hành mã hóa 2 lần file payload .exe*



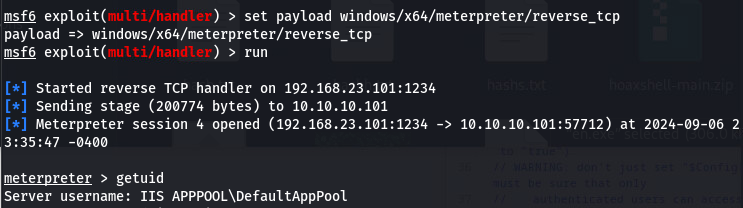
*Hình 3.20. Scan file payload đã mã hóa trên Virustotal lần 1*



*Hình 3.21. Scan file payload đã mã hóa trên Virustotal lần 2*

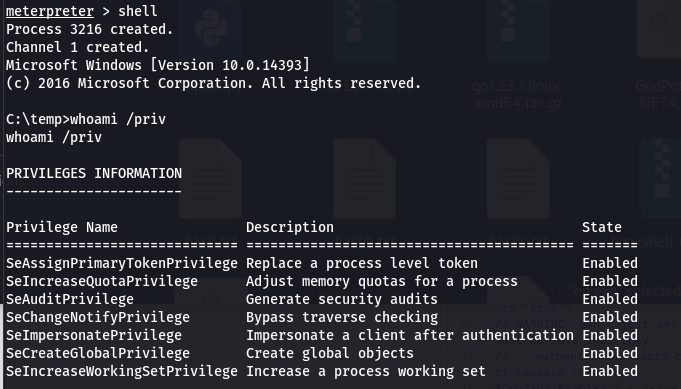


*Hình 3.22. Thành công bypass AV để chạy file payload trên máy web02*

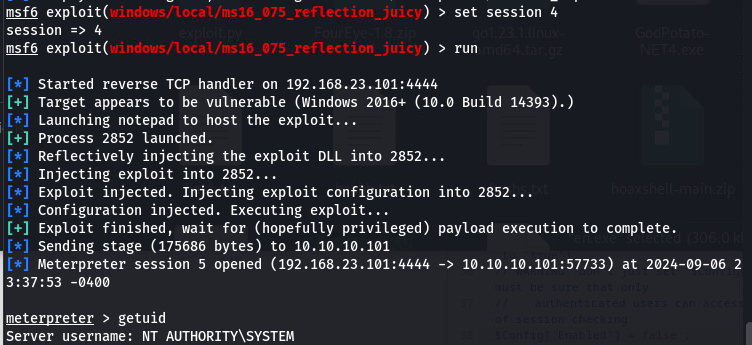


*Hình 3.23. Bắt session meterpreter thành công*

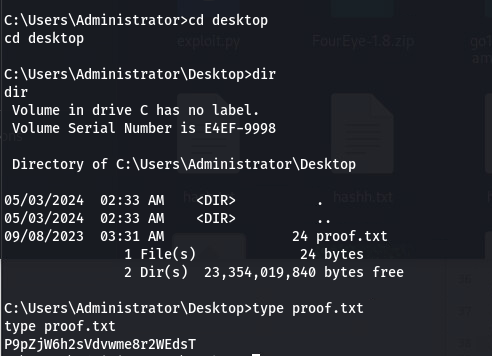
* Khi đã có được session từ meterpreter thì em dùng lệnh whoami /priv để xem có những quyền nào có thể khai thác được
* Em phát hiện có quyền **SeImpersonatePrivilege** đang được bật nên em sẽ dùng lại kiến thức ở module 4 để khai thác
* Em sử dụng module **juicy** của metasploit để khai thác đối với quyền trên và em đã leo quyền thành công trên máy web02 và lấy được flag



*Hình 3.24. Kiểm tra quyền đang được bật*



*Hình 3.25. Leo quyền thành công lên "NT AUTHORITY\SYSTEM"*



*Hình 3.26. Thành công lấy được flag*