

Kỹ thuật điều tra phân tích tấn công web

Mục lục

1. Giới thiệu về ứng dụng web.....	3
1.1. Các kiến thức cơ bản về ứng dụng web	3
a. Khái niệm về ứng dụng web	3
b. Cấu trúc của một ứng dụng web	3
c. Mô tả hoạt động của ứng dụng web	4
d. Khái niệm về Hacker.....	4
e. HTTP Request & HTTP Response.....	4
f. Phiên làm việc & Cookies	6
1.2. Một số kỹ thuật tấn công web phổ biến	7
a. Tấn công thu thập thông tin.....	7
b. Tấn công dựa trên lỗi cấu hình.....	8
c. Tấn công quá trình xác thực	8
d. Tấn công phiên làm việc	8
e. Tấn công lợi dụng thiếu sót trong việc kiểm tra dữ liệu đầu vào hợp lệ.....	8
2. Điều tra số & điều tra tấn công web.....	9
2.1. Điều tra số	9
a. Khái niệm.....	9
b. Các giai đoạn.....	9
c. Phân loại điều tra số	9
2.2. Điều tra và phân tích tấn công Web.....	11
3. Kỹ thuật điều tra và phân tích phía người dùng.....	12
3.1. Các khái niệm liên quan.....	12
a. Người dùng.....	12
b. Kỹ thuật điều tra và phân tích phía người dùng.....	12
c. Trình duyệt.....	12
3.2. Các công cụ hỗ trợ	15
4. Các kỹ thuật điều tra và phân tích phía máy chủ	17
4.1. Phân tích luồng dữ liệu	17

a. Khái niệm và một số công cụ.....	17
b. Ứng dụng trong phân tích tấn công ứng dụng web cơ bản	18
4.2. Phân tích tập tin nhật ký.....	23
a. Khái niệm liên quan	23
b. Ứng dụng Regular expression trong phân tích tập tin nhật ký tự động.....	24

1. Giới thiệu về ứng dụng web

1.1. Các kiến thức cơ bản về ứng dụng web

a. Khái niệm về ứng dụng web

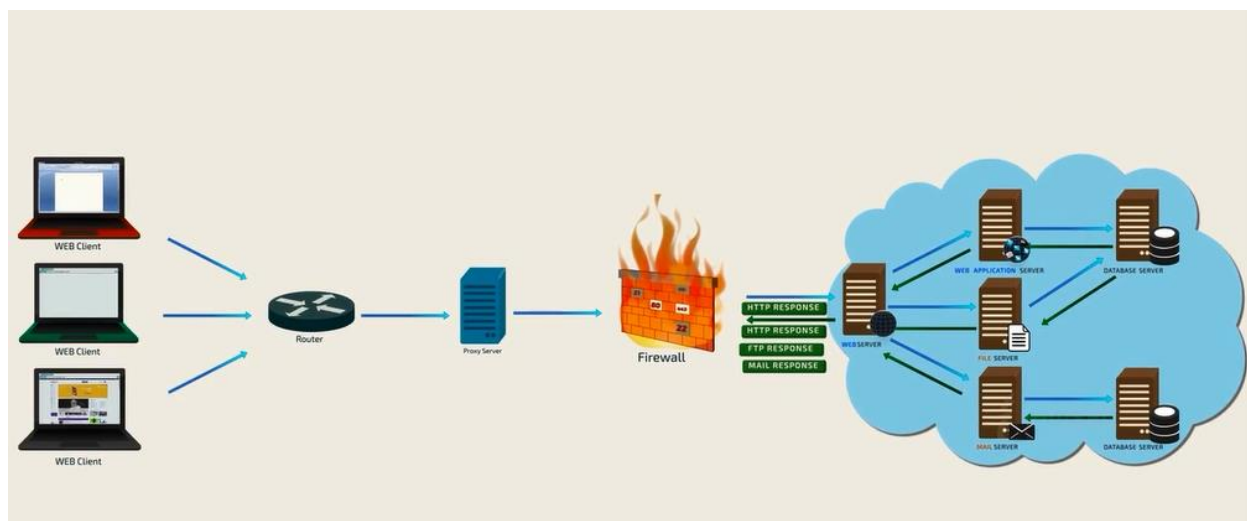
Ứng dụng web là một ứng dụng khách chủ sử dụng giao thức HTTP để tương tác với người dùng hay hệ thống khác.

Trình khách dành cho người dùng thường là một trình duyệt web như Internet Explorer, Firefox hay Google Chrome. Người dùng gửi và nhận các thông tin từ trình chủ thông qua việc tác động vào các trang Web. Các chương trình có thể là các trang trao đổi mua bán, các diễn đàn, gửi nhận email...

Tốc độ phát triển các kỹ thuật xây dựng ứng dụng Web cũng phát triển rất nhanh. Trước đây những ứng dụng Web thường được xây dựng bằng CGI (Common Gateway Interface) được chạy trên các trình chủ Web và có thể kết nối vào các cơ sở dữ liệu đơn giản trên cùng một máy chủ. Ngày nay ứng dụng Web được viết bằng Java (các ngôn ngữ tương tự) và chạy trên máy chủ phân tán, kết nối đến nhiều nguồn dữ liệu khác nhau.

b. Cấu trúc của một ứng dụng web

Một ứng dụng web thường có kiến trúc gồm:



Hình 1: Cấu trúc của ứng dụng Web

Trong đó:

- Trình khách (hay còn gọi là trình duyệt): IE, Firefox, Google Chrome
- Trình chủ: Apache, IIS,...
- Cơ sở dữ liệu: SQL server, MySQL,...
- Tường lửa: Lớp rào chắn bên ngoài một hệ thống mạng, vai trò kiểm soát luồng thông tin giữa các máy tính
- Proxy xác định những yêu cầu từ trình khách và quyết định đáp ứng yêu cầu hay không, Proxy đóng vai trò cầu nối trung gian giữa máy chủ và máy khách

c. Mô tả hoạt động của ứng dụng web

Đầu tiên trình duyệt sẽ gửi một yêu cầu (request) đến trình chủ Web thông qua các phương thức cơ bản GET, POST,... của giao thức HTTP. Trình chủ lúc này có thể cho thực thi một chương trình được xây dựng từ nhiều ngôn ngữ như Perl, C/C++,... hoặc trình chủ yêu cầu bộ diễn dịch thực thi các trang ASP, PHP, JSP,... theo yêu cầu của trình khách.

Tùy theo các tác vụ của chương trình được cài đặt mà nó xử lý, tính toán, kết nối đến cơ sở dữ liệu, lưu các thông tin do trình khách gửi đến... và từ đó trả về cho trình khách một luồng dữ liệu có định dạng theo giao thức HTTP, gồm hai phần:

- Header mô tả các thông tin về gói dữ liệu và các thuộc tính, trạng thái trao đổi giữa trình duyệt và máy chủ.
- Body là phần nội dung dữ liệu mà máy chủ gửi về máy trạm, nó có thể là một tập tin HTML, một hình ảnh, một đoạn phim hay một văn bản bất kỳ

d. Khái niệm về Hacker

Hacker là những người am hiểu về hệ điều hành, hệ quản trị cơ sở dữ liệu, các ngôn ngữ lập trình... Họ sử dụng kiến thức của mình trong việc tìm tòi và khai thác các lỗ hổng của hệ thống mạng. Một số hacker chỉ dừng lại việc phát hiện và thông báo lỗi tìm được cho các nhà bảo mật hay người phát triển chương trình, họ được coi như là WhiteHat (Hacker mũ trắng). Một số hacker dựa vào những lỗ hổng thực việc khai thác trái phép nhằm mục đích phá hoại hay mưu lợi riêng, nhưng người này bị xem như là BlackHat (Hacker mũ đen).

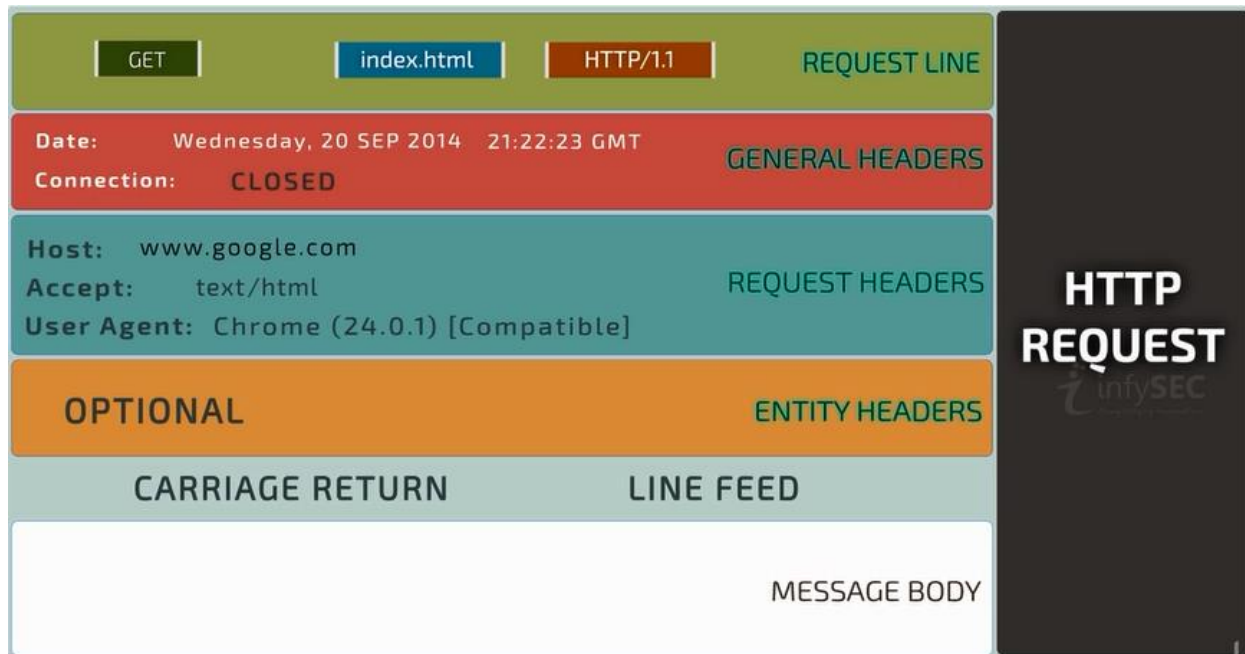


Hình 2: Phân loại Hacker

e. HTTP Request & HTTP Response

HTTP header là phần đầu của thông tin mà trình khách và trình chủ gửi cho nhau. Những thông tin trình khách gửi cho trình chủ được gọi là HTTP requests (yêu cầu) còn trình chủ gửi cho trình khách là HTTP responses (phản hồi). Thông thường một HTTP header gồm nhiều dòng, mỗi dòng chứa tên tham số và

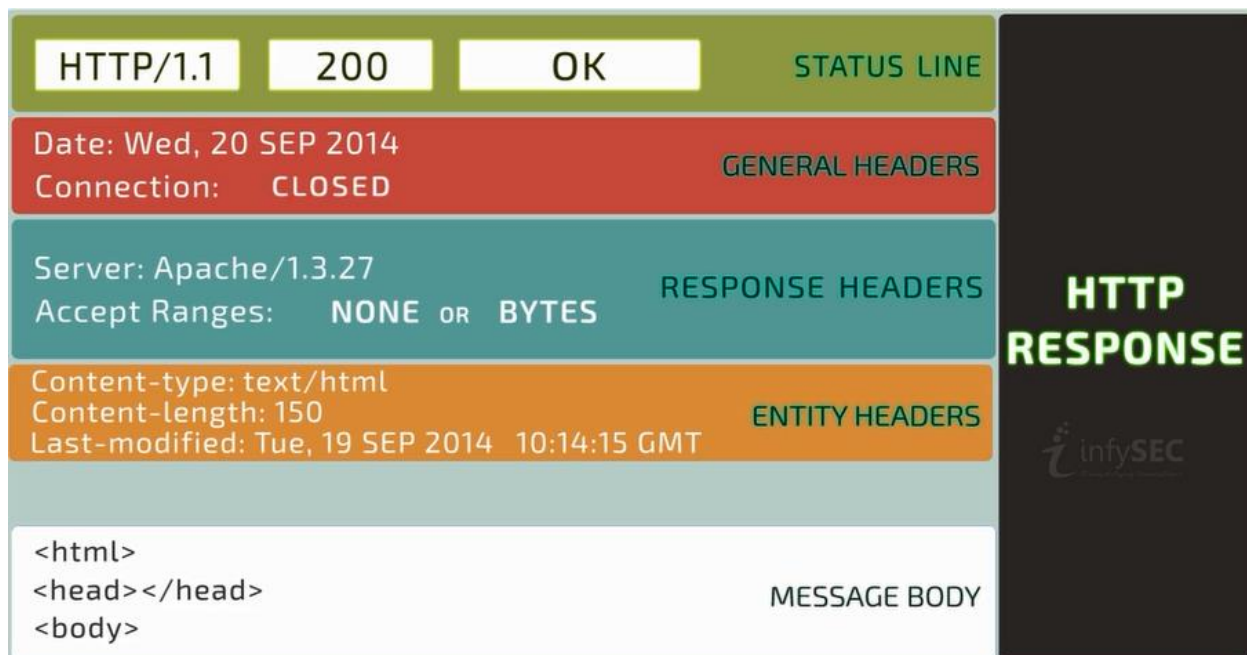
giá trị. Một số tham số có thể được dùng trong cả Header yêu cầu và Header trả lời, còn số khác thì chỉ được dùng riêng trong từng loại.



Hình 3: HTTP Request

HTTP yêu cầu:

- Dòng đầu của HTTP Request là dòng Request-Line bao gồm các thông tin về phương thức mà HTTP request này sử dụng (POST, GET, HEAD, TRACE,...). URI là địa chỉ định danh của tài nguyên. HTTP version là phiên bản HTTP đang sử dụng
- Tiếp theo là các trường Header thông dụng như
 - Accept: Loại nội dung có thể nhận được từ thông điệp phản hồi. Ví dụ: text/plain, text/html,...
 - Accept-Encoding: Các kiểu nén được chấp nhận. ví dụ: gzip, xz,...
 - User-Agent: Thông tin về trình duyệt của người dùng
 - Connection: Tùy chọn cho kết nối hiện tại. Ví dụ: closed, keep-alive, update,...
 - Cookie: Thông tin HTTP Cookie từ máy chủ
- Header của HTTP request sẽ kết thúc bằng một dòng trống



Hình 4: HTTP phản hồi

Cấu trúc của HTTP phản hồi gần giống với HTTP yêu cầu, chỉ khác nhau là thay vì Request-Line thì HTTP phản hồi có Status-Line.

HTTP phản hồi:

- Status-Line có ba phần chính như sau: HTTP-version là phiên bản HTTP cao nhất mà máy chủ đang hỗ trợ, Status-Code: mã kết quả trả về, Reason-Phrase: mô tả về Status-Code
- Tiếp theo là các tham số và kèm một dòng trống để báo hiệu kết thúc header
- Cuối cùng là phần thân của HTTP response

Để hiểu rõ hơn về HTTP Request và HTTP Response, chúng ta có thể tham khảo đường dẫn dưới đây:

- <https://tools.ietf.org/html/rfc2616>

f. Phiên làm việc & Cookies

HTTP là giao thức hướng đối tượng tức quất và phi trạng thái, nghĩa là HTTP không lưu trữ trạng thái làm việc giữa trình duyệt với trình chủ. Sự thiếu sót này gây khó khăn cho một số ứng dụng web, bởi vì trình chủ không biết được trước đó trình duyệt đã có những trạng thái nào. Vì thế, để giải quyết vấn đề này, ứng dụng Web đưa ra một khái niệm phiên làm việc (Session). Còn SessionID là một chuỗi để chứng thực phiên làm việc. Một số trình chủ sẽ cung cấp một Session ID cho người dùng khi họ xem trang web trên trình chủ. Để duy trì phiên làm việc, Session ID thường được lưu vào:

- Biến trên URL
- Biến ẩn
- Cookies

Phiên làm việc chỉ tồn tại trong một thời gian cho phép, thời gian này được cấu hình quy định tại trình chủ hoặc bởi ứng dụng thực thi.

Cookie là những phần dữ liệu nhỏ, có cấu trúc và được chia sẻ giữa trình chủ và trình duyệt của người dùng. Các cookie được lưu dưới những file dữ liệu nhỏ dạng text, được ứng dụng tạo ra để lưu trữ/ truy tìm/ nhận biết các thông tin về người dùng đã ghé thăm trang Web và những vùng mà họ đã truy cập qua trong trang web. Những thông tin này có thể bao gồm tên/ định danh người dùng, mật khẩu, sở thích, thói quen. Ở những lần truy cập sau đến trang web đó, ứng dụng có thể sử dụng lại những thông tin lưu trong cookie.

Cookie được phân làm hai loại secure/non-secure và persisten/non-persistent, do vậy ta tổng hợp được bốn kiểu cookie là:

- Persistent & Secure
- Persistent & Non-Secure
- Non-Persistent và Non-Secure
- Non-Persistent và Secure

Persistent cookies được lưu dưới dạng tập tin .txt trên máy khách trong một khoảng thời gian nhất định. Non-Persistent cookie thì được lưu trên bộ nhớ RAM của máy khách và sẽ bị hủy khi đóng trang web hay nhận được lệnh hủy từ trang web. Secure cookies chỉ có thể được gửi thông qua HTTPSs, Non-Secure cookie có thể gửi được bằng cả hai giao thức HTTPS hay HTTP.

Các thành phần của một cookie gồm:

- Domain: Tên miền của trang web đã tạo cookie
- Flag: Mang giá trị True/False - xác định các máy khác với cùng tên miền có được truy xuất đến cookie hay không
- Path: Phạm vi các địa chỉ có thể truy xuất cookie
- Secure: Mang giá trị True/False, tương ứng với Secure cookie và Non-Secure cookie
- Expiration: Thời gian hết hạn của cookie. Nếu giá trị này không được thiết lập thì trình duyệt sẽ hiểu đây là non-persistent cookie và chỉ lưu trong bộ nhớ RAM và sẽ xóa nó khi trình duyệt bị đóng
- Name: Tên biến
- Value: Giá trị của biến

Kích thước tối đa của cookie là 4kb. Số cookie tối đa cho một tên miền là 20 cookie.

1.2. Một số kỹ thuật tấn công web phổ biến

a. Tấn công thu thập thông tin

Những tập tin và ứng dụng trên hệ thống chứa những thông tin quan trọng như mã nguồn trang web, tập tin chứa mật khẩu của người dùng trên hệ thống luôn là những mục tiêu hàng đầu cho hacker

Một số phương pháp chính:

- Tấn công quét công
- Tấn công dò quét thư mục
- Thu thập thông tin từ Internet

b. Tấn công dựa trên lỗi cấu hình

Các tập tin cấu hình, ứng dụng luôn luôn tồn tại các lỗ hổng chưa được khám phá hoặc các lỗ hổng cũ, do sự không cảnh giác của người quản trị website nên vẫn tồn tại, nhờ vào đặc điểm này, Hacker có thể dễ dàng tìm kiếm các đoạn mã khai thác trên Internet hoặc tự phát triển các mã khai thác để khai thác điểm yếu của cấu hình. Một trong số các tấn công điển hình là: Misconfiguration Attack và 0-day Attack.

c. Tấn công quá trình xác thực

Do nhiều yếu tố nên quá trình xác thực của các trang web luôn tồn tại các lỗ hổng hoặc điểm yếu, nơi các Hacker luôn có nhiều phương pháp để tiến hành tấn công nhằm chiếm được Username/Password của quản trị viên hay người dùng.

Một số phương pháp:

- Tấn công dò quét mật khẩu
- Tấn công từ điển
- SQL injection

d. Tấn công phiên làm việc

Đây là kỹ thuật tấn công cho phép Hacker mạo danh người dùng hợp lệ bằng cách nghe trộm khi người dùng đăng nhập vào hệ thống, sau đó Hacker sẽ dùng lại Session ID của người dùng hợp lệ để tiến hành xâm nhập hoặc chuộc lợi. Hoặc bằng cách giải mã Session ID của người dùng hợp lệ để tiên đoán và tạo ra các session ID hợp lệ khác,...

Một số phương pháp:

- Session Hijacking
- Brute Force Session ID
- Session Fixation Attack

e. Tấn công lợi dụng thiếu sót trong việc kiểm tra dữ liệu đầu vào hợp lệ

Hacker lợi dụng những ô nhập dữ liệu, các tham số đầu vào để gửi đi một đoạn ký tự bất kỳ khiến cho hệ thống phải thực thi đoạn lệnh hay bị phá vỡ hoàn toàn.

Một số phương pháp:

- Chèn mã lệnh thực thi trên trình duyệt - Cross-Site Scripting
- Tiêm mã truy vấn cơ sở dữ liệu - SQL Injection
- Thêm câu lệnh hệ thống - OS Command Injection
- Vượt đường dẫn - Path traversal
- Tràn bộ đệm - Buffer Over Flow

2. Điều tra số & điều tra tấn công web

2.1. Điều tra số

a. Khái niệm

Theo wikipedia, điều tra số (Pháp y số) là một nhánh của khoa học pháp y, bao gồm việc phục hồi và điều tra tài liệu tìm thấy trong các thiết bị kỹ thuật số, vấn đề này liên quan trực tiếp tới Hacker. Thuật ngữ điều tra số ban đầu được sử dụng như một từ đồng nghĩa cho điều tra máy tính nhưng đã được mở rộng để bao gồm điều tra tất cả các thiết bị có khả năng lưu trữ dữ liệu số.

Điều tra pháp y kỹ thuật số có nhiều ứng dụng. Phổ biến nhất là hỗ trợ hoặc bác bỏ giả thuyết trước tòa án hình sự hoặc dân sự. Các vụ án hình sự liên quan đến việc vi phạm các luật được định nghĩa bởi luật pháp và được thi hành bởi cảnh sát và bị truy tố bởi nhà nước, chẳng hạn như giết người, trộm cắp và hành hung chống lại người thi hành công vụ. Các vụ kiện dân sự về việc bảo vệ quyền và tài sản của cá nhân (thường liên quan đến tranh chấp gia đình) nhưng cũng có thể liên quan đến các tranh chấp hợp đồng giữa các tập đoàn thương mại.

b. Các giai đoạn

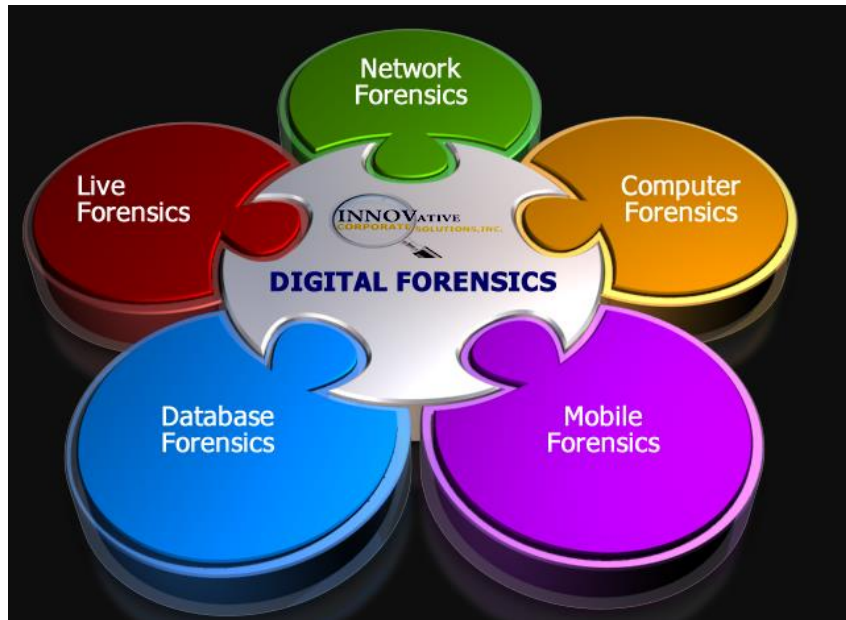
Các giai đoạn của điều tra số:

- Thu thập thông tin từ hiện trường
- Phân tích
- Báo cáo

Việc thu thập thông tin từ hiện trường là một bước tiền đề cũng như quan trọng nhất trong quá trình điều tra số, bao gồm các phương pháp phổ biến như "Memory dump", tạo các bản sao của media (sử dụng các phương pháp sao chép, lưu trữ và hàm băm), sử dụng các thiết bị & phương pháp chặn ghi để bảo toàn dữ liệu gốc,...

c. Phân loại điều tra số

Tùy thuộc vào loại thiết bị, phương tiện hoặc hiện vật, điều tra pháp y kỹ thuật số được phân thành nhiều loại khác nhau.



Hình 5: Phân loại điều tra số

Computer forensic:

Mục đích của điều tra máy tính là giải thích hiện trạng của một tạo tác kỹ thuật số; chẳng hạn như hệ thống máy tính, phương tiện lưu trữ hoặc tài liệu điện tử. Điểm mấu chốt của loại điều tra này là việc phân tích và báo cáo một loạt các thông tin; từ nhật ký (chẳng hạn như lịch sử internet) đến các tệp thực trên ổ đĩa.

Ví dụ: Trong năm 2007 các công tố viên đã sử dụng một bảng tính được phục hồi từ máy tính của Joseph E. Duncan, để cáo buộc y bản án tử hình. Sát thủ giết Sharon Lopatka đã được xác định vào năm 2006 sau khi các tin nhắn email của y nêu lên chi tiết những hình ảnh tra và hành hình cô gái.

Mobile forensic:

Điều tra thiết bị di động là một nhánh phụ của pháp y kỹ thuật số liên quan đến việc thu hồi bằng chứng kỹ thuật số hoặc dữ liệu từ một thiết bị di động. Các cuộc điều tra thường tập trung vào dữ liệu đơn giản như dữ liệu cuộc gọi và thông tin liên lạc (SMS / Email) thay vì phục hồi sâu dữ liệu đã xóa.

Thiết bị di động cũng hữu ích cho việc cung cấp thông tin vị trí; hoặc từ theo dõi vị trí / GPS sẵn có hoặc qua nhật ký trang web trên thiết bị di động, theo dõi các thiết bị trong phạm vi của chúng.

Network forensic:

Điều tra mạng máy tính có liên quan đến việc theo dõi và phân tích lưu lượng mạng máy tính, cả mạng cục bộ và WAN / internet, với mục đích thu thập thông tin, thu thập bằng chứng, hoặc phát hiện xâm nhập. Lưu lượng truy cập thường bị chặn ở cấp gói và được lưu trữ để phân tích sau hoặc được lọc theo thời gian thực.

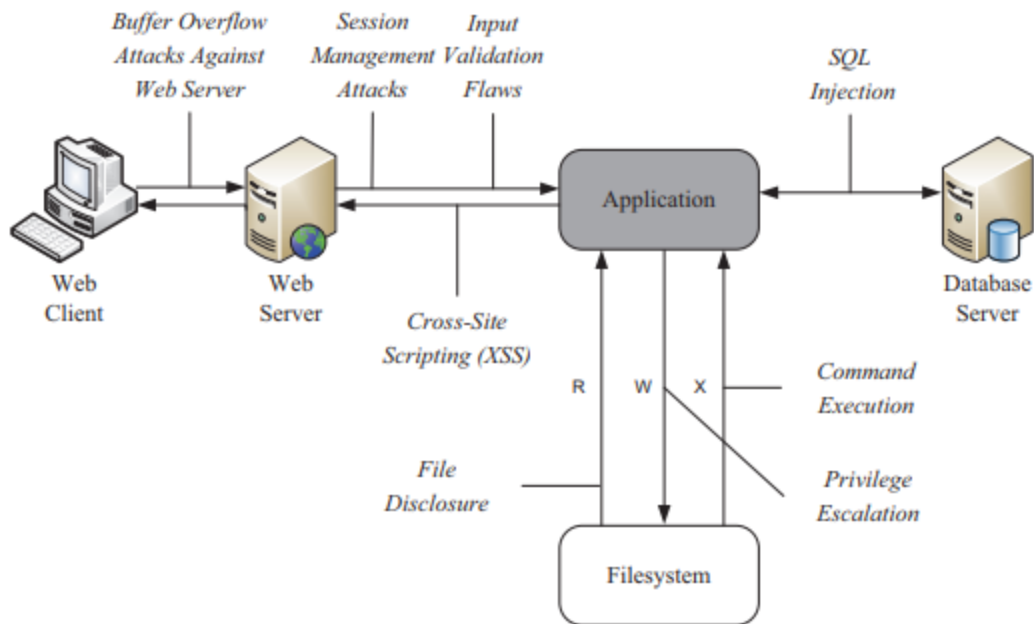
Database forensic:

Điều tra cơ sở dữ liệu là một nhánh của điều tra số liên quan đến nghiên cứu & phân tích về cơ sở dữ liệu và siêu dữ liệu của chúng. Điều tra loại này sử dụng nội dung cơ sở dữ liệu, tệp nhật ký và dữ liệu trong RAM để tạo dòng thời gian hoặc khôi phục thông tin có liên quan.

Live forensic:

Đây là một nhánh của pháp y kỹ thuật số. Nó kiểm tra dữ liệu có cấu trúc với mục đích khám phá và phân tích các mẫu hoạt động gian lận do Hacker gây ra.

2.2. Điều tra và phân tích tấn công Web



Hình 6: Cấu trúc và các phương pháp tấn công Web

Hình 6 trên vừa cho chúng ta thấy lại được cấu trúc của Web và các phương pháp tấn công, khai thác phổ biến vào từng thành phần của cấu trúc web, hơn nữa, các phương pháp tấn công ngày càng đa dạng, phức tạp và hiệu quả, chính vì vậy, điều tra và phân tích tấn công web là một đề tài khá thú vị cần quan tâm.

Sau quá trình tìm hiểu và nghiên cứu, bài báo cáo sẽ đưa ra hai phương pháp chính trong việc điều tra và phân tích tấn công Web, đó là:

- Kỹ thuật điều tra và phân tích phía người dùng
- Các kỹ thuật điều tra và phân tích phía máy chủ

3. Kỹ thuật điều tra và phân tích phía người dùng

3.1. Các khái niệm liên quan

a. Người dùng

Người dùng ứng dụng web là những khách hàng, người dùng mạng máy tính, quản trị viên hoặc các kẻ tấn công có nhu cầu kết nối tới trang web để thực hiện các hành động theo nhu cầu và mong muốn của bản thân.

Phân loại:

- Người dùng thông thường
- Kẻ tấn công

b. Kỹ thuật điều tra và phân tích phía người dùng

Như đã thấy ở trên, người dùng có hai loại, vì thế, kỹ thuật điều tra và phân tích phía người dùng ra đời với mục tiêu:

- Xác định xem người dùng có phải là nạn nhân
- Xác định xem người dùng có phải là kẻ tấn công

Vì sao lại như vậy? Như đã biết có rất nhiều kiểu, phương pháp tấn công phía client side ví dụ như: XSS, Phishing,... Nếu chúng ta không có những chứng cứ số hoặc không được tiếp cận các thiết bị truy cập website của người dùng, thì việc điều tra tấn công là rất khó khăn, vấn đề này rất cần thiết với người dùng hợp lệ & các nạn nhân của các cuộc tấn công gián tiếp hoặc trực tiếp qua website.

Đối với kẻ tấn công, để xác nhận đúng một người có phải là kẻ tấn công hay không, ngoài chứng cứ, bằng chứng trên Server side, ta cũng cần các chứng cứ hay bằng chứng trực tiếp trên thiết bị truy cập website của người dùng nhằm đưa ra một quyết định vững chắc rằng họ vi phạm hoặc phạm tội.

Các kỹ thuật chính:

- Điều tra và phân tích dữ liệu trên Hệ điều hành (Vì thời lượng và giới hạn của bài báo cáo, nên Forensic OS sẽ không được đề cập tới, người đọc có thể tham khảo link sau đây để tìm hiểu thêm về vấn đề này:
<https://www.google.com.vn/search?q=os+forensic&oq=os+forensic&aqs=chrome..69i57j015.3855j0j7&sourceid=chrome&ie=UTF-8>)
- Điều tra và phân tích thông tin Trình duyệt
 - Email
 - Các trang đã truy cập
 - Các thông tin tìm kiếm trên mạng

c. Trình duyệt

Khái niệm trình duyệt:

Trình duyệt web là công cụ để thực hiện các hoạt động khác nhau trên Internet của người dùng, người dùng sử dụng trình duyệt cho nhiều chức năng như: tìm kiếm thông tin, truy cập vào tài khoản email, giao dịch thương mại điện tử, nhắn tin,... Trình duyệt cũng ghi lại nhiều dữ liệu liên quan đến hoạt động của

người dùng, các thông tin như: URLs được truy cập bởi người dùng, cookie, tệp bộ nhớ cache, thời gian truy cập & thời gian sử dụng trình duyệt,...



Hình 7: Một số trình duyệt Web nổi tiếng

Việc kiểm tra các bằng chứng nói trên là một trong các điểm chủ chốt của quá trình "Browser forensic". Các trình duyệt lưu trữ các tập tin quan trọng này ở nhiều phần khác nhau trên hệ điều hành, ngoài ra như ta đã thấy, có rất nhiều trình duyệt khác nhau, đồng nghĩa với nó đó chính là dữ liệu hoặc địa điểm lưu trữ các tập tin cũng khác nhau. Dưới đây là bảng tổng hợp các bản ghi Cache, các bản ghi Lịch sử, Cookie registry và các tập tin đã tải xuống ở các trình duyệt nổi tiếng, để dễ dàng hơn trong quá trình truy vết và điều tra.

Web Browser	Operating System	File Path
Internet Explorer	Windows 95/98	C:\Temporary Internet Files\Content.ie5 C:\Cookies C:\History\History.ie5
	Windows 2000/XP	C:\Documents and Settings\%username%\Local Settings\Temporary Internet Files\Content.ie5 C:\Documents and Settings\%username%\Cookies C:\Documents and Settings\%username%\Local Settings\History\history.ie5
	Windows Vista, 7 and latest version	C:\Users\%username%\AppData\Local\Microsoft\Windows\Temporary Internet Files\ C:\Users\%username%\AppData\Local\Microsoft\Windows\Temporary Internet Files\Low\
Firefox	Linux	/home/\$USER/.mozilla/firefox/\$PROFILE.default/places.sqlite
	MacOS-X	/Users/\$USER/Library/Application Support/Firefox/Profiles/\$PROFILE.default/places.sqlite
	Windows XP	C:\Documents and Settings\%username%\Application Data\Mozilla\Firefox\Profiles\%PROFILE%.default\places.sqlite
	Windows Vista, 7 and latest version	C:\Users\%USERNAME%\AppData\Roaming\Mozilla\Firefox\Profiles\%PROFILE%.default\places.sqlite
Safari	MacOS-X	/Users/\$USER/Library/Safari/ /Users/\$USER/Library/Caches/com.apple.Safari/
	Windows XP	C:\Documents and Settings\%username%\Application Data\Apple Computer\Safari\ C:\Documents and Settings\%username%\Local Settings\Application Data\Apple Computer\Safari\
	Windows 7	C:\Users\%username%\AppData\Roaming\Apple Computer\Safari\ C:\Users\%username%\AppData\Local\Apple Computer\Safari\
Opera	Linux	/home/\$USER/.opera/
	MacOS-X	/Users/\$USER/Library/Opera/
	Windows XP	C:\Documents and Settings\%username%\Application Data\Opera\Opera\ C:\Users\%username%\AppData\Roaming\Opera\Opera\ C:\Users\%username%\AppData\Local\Opera\Opera\
Google Chrome	Linux	/home/\$USER/.config/google-chrome/Default/Preferences
	MacOS-X	/Users/\$USER/Library/Application Support/Google/Chrome/Default/Preferences
	Windows XP	C:\Documents and Settings\%username%\Local Settings\Application Data\Google\Chrome\User Data\Default\Preferences C:\Users\%username%\AppData\Local\Google\Chrome\User Data\Default\Preferences
	Windows Vista, 7 and latest version	C:\Users\%username%\AppData\Local\Google\Chrome\User Data\Default\Preferences

Hình 8: Bảng tổng hợp các bản ghi của một số trình duyệt nổi tiếng

Internet Explorer là trình duyệt web mà người dùng máy tính thường hay sử dụng, các hoạt động sẽ được lưu cho từng người dùng riêng tương ứng với thư mục người dùng của họ, dữ liệu được lưu trong Cookie, Cache, lịch sử và lịch sử tải xuống (tham khảo thêm ở hình 8). Ngoài ra dữ liệu cũng có thể được lưu trong tập tin cơ sở dữ liệu như index.dat hay container.dat và dữ liệu trong hai tập tin này được lưu dưới dạng nhị phân. Cũng lưu dữ liệu trong tập tin cơ sở dữ liệu dưới dạng nhị phân đó là trình duyệt Safari, tuy nhiên safari đặt tên tập tin lưu trữ là history.plist, ở đây lưu trữ các thông tin như địa chỉ URLs, ngày tháng truy cập, lượng truy cập ở mỗi website. Firefox sử dụng định dạng dữ liệu SQLite để lưu trữ các thông tin, chúng được đặt tên là places.sqlite. Opera thì lưu trữ các thông tin trên ở các tập tin .dat khác nhau như: cookies4.dat, download.dat, global_history.dat. Google chrome cho phép lưu trữ dữ liệu trong tập tùy chọn, tùy thuộc vào lựa chọn của người dùng.

Dưới đây là bảng cung cấp địa chỉ, nơi dùng để xóa các bản ghi của từng loại trình duyệt.

Web Browser	Delete Options Path
Internet Explorer	Settings/ Internet Options/ / Deletes
Firefox	Settings /Privacy/History about:preferences#privacy
Google Chrome	Settings /History/Search Data chrome://settings/clearBrowserData
Safari	Settings / Privacy / Delete All Web Site Data Settings/History
Opera	Settings /History/Privacy and Security/Delete All Search Data opera://settings/clearBrowserData

Hình 9: Địa chỉ xóa bản ghi dữ liệu của các trình duyệt

3.2. Các công cụ hỗ trợ

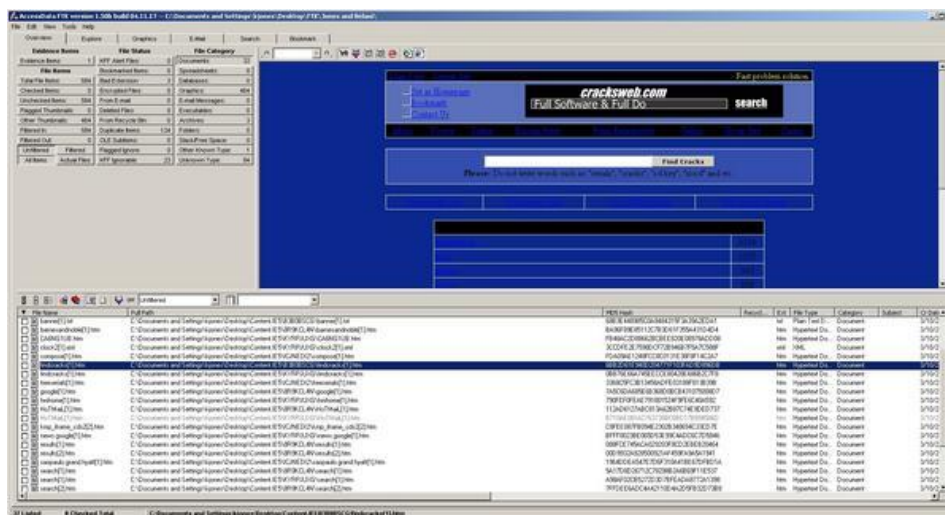
Các chương trình phổ biến như NetAnalysis, Browser history, FTK và Encase là các phần mềm nguồn mở & miễn phí, khá hiệu quả trong quá trình điều tra số trình duyệt.

Net Analysis:

NetAnalysis là một công cụ được cấp phép do công ty Digital Detective phát triển để điều tra số các trình duyệt web, hỗ trợ Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Apple Safari và Opera browsers. Nó cho phép kiểm tra lịch sử Internet, bộ nhớ cache, cookie và các thành phần khác. công cụ này cho phép thu thập nhanh bằng chứng theo hành vi của người dùng. Phần mềm này cũng có các công cụ phân tích hiệu quả để giải mã và hiểu dữ liệu. Đồng thời, nó có khả năng sử dụng các truy vấn SQL để xác định bằng chứng liên quan. Ngoài ra nó có thể được sử dụng để phục hồi các thành phần trình duyệt web đã xóa.

FTK:

FTK là một trong những công cụ được phát triển để phân tích toàn bộ hệ thống. Nó cho phép phân tích dữ liệu trình duyệt web với tính năng, đặc điểm. Lịch sử trình duyệt web được ảo hóa chi tiết. Internet Explorer, Firefox, Chrome, Safari và Opera là trình duyệt được hỗ trợ. Ngoài ra, dữ liệu trình duyệt web đã xóa có thể được phục hồi bởi FTK. Phần mềm này cũng có tính năng báo cáo kết quả phân tích.



Hình 10: Giao diện của FTK

Browser History Examiner:

Browser History Examiner là một công cụ được cấp phép phát triển bởi Foxton Forensics Company, có chức năng trích xuất và phân tích lịch sử web. Nó hỗ trợ các trình duyệt web Chrome, Firefox, Internet Explorer và Edge. Và nó có thể phân tích nhiều loại dữ liệu dưới dạng tải xuống, dữ liệu bộ nhớ cache và tệp URL đã truy cập.

Encase:

Encase là một công cụ phân tích được phát triển để kiểm tra toàn bộ hệ thống. Nó cho phép kiểm tra trình duyệt web, dữ liệu với các tính năng của trình duyệt. Với sự trợ giúp của một tập lệnh đơn giản, tất cả các lịch sử trình duyệt, cookie và tệp bộ nhớ cache được sao chép vào một tệp bằng cách sử dụng phần mềm của bên thứ ba. Nó cũng cho phép phục hồi các thành phần internet đã bị xóa. Dữ liệu thu được có thể được phân tích bằng cách lọc theo các thông số từ và thời gian chính.

4. Các kỹ thuật điều tra và phân tích phía máy chủ

Hiện nay, có rất nhiều các thiết bị, công cụ hỗ trợ điều tra & phân tích tấn công một cách dễ dàng, ví dụ như các hệ thống: IDS/IPS, honey pot, honey net,... Tuy nhiên trong bài viết này sẽ đưa ra hai phương pháp chính hỗ trợ điều tra và phân tích tấn công web phía máy chủ, với trường hợp máy chủ Linux Apache & không hỗ trợ các hệ thống phát hiện xâm nhập hay phân tích dữ liệu hiện đại, chủ yếu dựa trên các công cụ mã nguồn mở miễn phí.

Hai phương pháp chính:

- Phân tích luồng dữ liệu
- Phân tích tập tin nhật ký

So sánh hai phương pháp:

Phương pháp	Điểm mạnh	Điểm yếu
Phân tích luồng dữ liệu	Có thể phân tích tất cả các thông tin	Dữ liệu cần phải được chặn bắt Dữ liệu có thể cần được lắp ráp, chống phân mảnh, chuẩn hóa (Các gói tin IP, IP fragments,...) Rất khó để chặn bắt và giải mã dữ liệu trên đường chuyển đã mã hóa (Encrypted traffic, High Traffic load,...)
Phân tích tập tin nhật ký	Dữ liệu có sẵn trong các tập tin	Các tập tin nhật ký thường chỉ chứa một phần nhỏ của toàn bộ dữ liệu (ví dụ: thiếu các tham số trong gói POST HTTP)

4.1. Phân tích luồng dữ liệu

a. Khái niệm và một số công cụ

Luồng dữ liệu:

Luồng dữ liệu (RFC3679) là một chuỗi các gói tin được gửi từ một nguồn cụ thể tới một đích hoặc nhiều đích, trong đó nguồn gán nhãn cho chuỗi các gói tin này là một luồng riêng.

Một số dấu hiệu cần chú ý:

- Địa chỉ IP nguồn, đích
- Cổng
- Giao thức và cờ hiệu
- Hướng luồng dữ liệu
- Khối lượng dữ liệu được truyền

Quan hệ giữa các địa chỉ IP:

- One to many: Spam, Scan port trên 1 dải mạng,...
- Many to one: DDOS attack, máy chủ syslog,...
- Many to many: Đồng bộ dữ liệu, phát tán virus,...
- One to one: Tấn công có mục tiêu, truyền tin,...

Phân tích luồng dữ liệu thực hiện việc thanh tra một chuỗi các gói tin có liên quan đến nhau nhằm xác định các hành vi nghi ngờ, trích xuất dữ liệu hay phân tích các giao thức trong luồng.

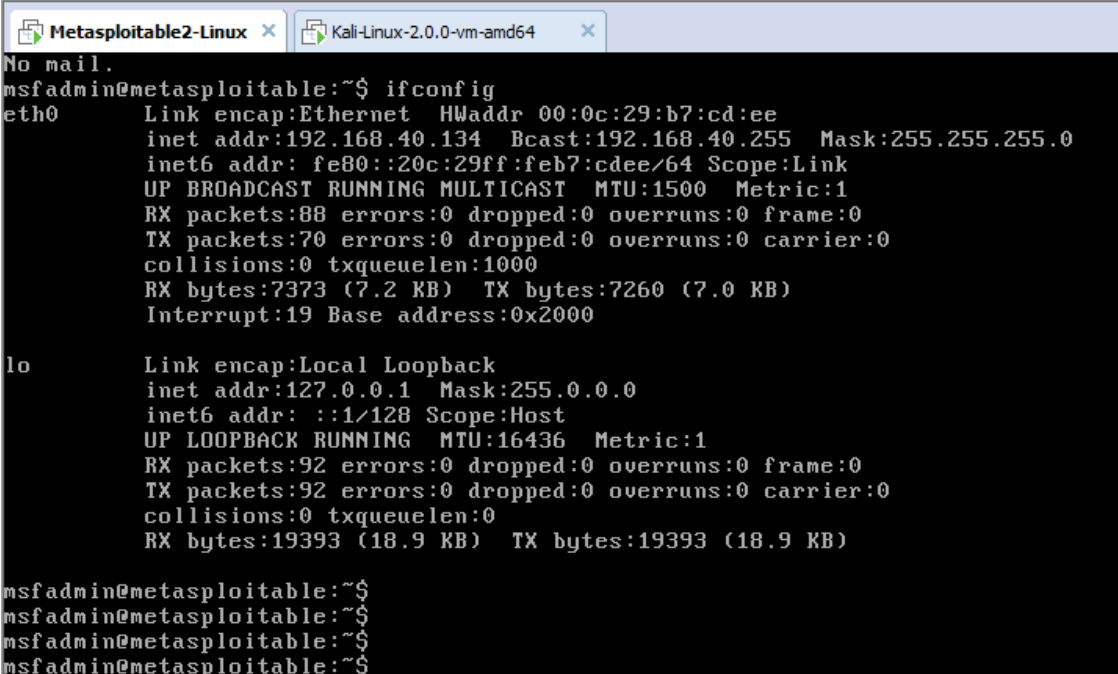
Một số công cụ nổi tiếng sử dụng trong quá trình phân tích luồng dữ liệu:

- Wireshark
- Tshark
- TCP dump

b. Ứng dụng trong phân tích tấn công ứng dụng web cơ bản

Mô hình thực hiện:

Môi trường được xây dựng trên VMware và sử dụng chế độ card mạng NAT. Dải địa chỉ tương ứng cho card mạng NAT là: 192.168.40.0/24. Máy mục tiêu là Metasploitable 2 với địa chỉ: 192.168.40.134, Metasploitable 2 là một hệ điều hành miễn phí, mã nguồn mở, chứa các lỗ hổng để thực hiện mô phỏng tấn công web một cách hợp pháp. Địa chỉ 192.168.40.131 được cấp cho máy tấn công Kali. Quá trình phân tích sử dụng công cụ Wireshark. Luồng dữ liệu được phân tích chính là luồng dữ liệu HTTP từ địa chỉ 192.168.40.131 đến địa chỉ 192.168.40.134.

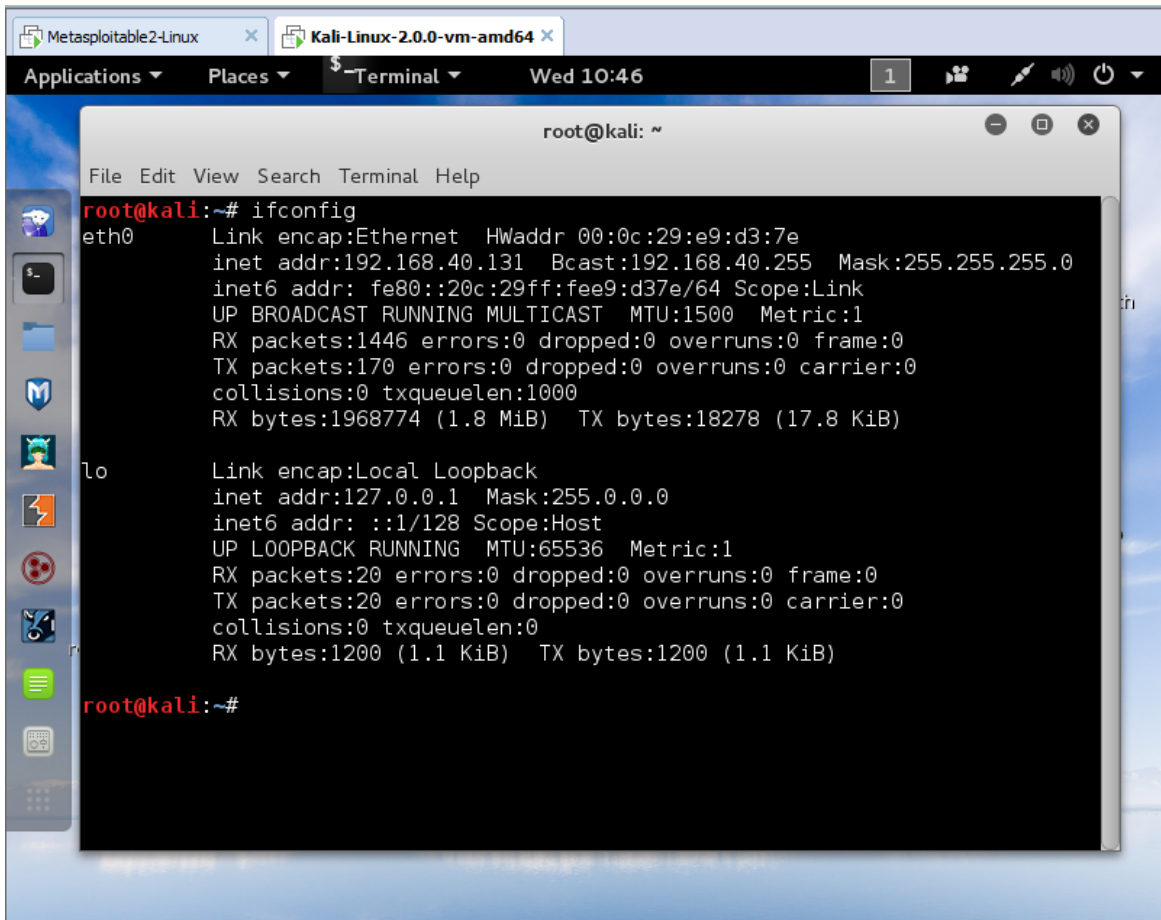


```
Metasploitable2-Linux x Kali-Linux-2.0.0-vm-amd64 x
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:b7:cd:ee
          inet addr:192.168.40.134  Bcast:192.168.40.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:feb7:cdee/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:88  errors:0  dropped:0  overruns:0  frame:0
          TX packets:70  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7373 (7.2 KB)  TX bytes:7260 (7.0 KB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:92  errors:0  dropped:0  overruns:0  frame:0
          TX packets:92  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19393 (18.9 KB)  TX bytes:19393 (18.9 KB)

msfadmin@metasploitable:~$
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$
```

Hình 11: Máy Metasploitable 2



Hình 12: Máy Kali

Sau khi sử dụng trình duyệt web từ máy tấn công để truy cập vào địa chỉ ip của mục tiêu, sử dụng Wireshark chúng ta sẽ thấy được rất nhiều luồng dữ liệu từ các nguồn khác nhau, các giao thức khác nhau,... Thật may mắn khi Wireshark có chức năng bộ lọc (filter) để giúp người dùng có thể thỏa mãn nhu cầu, yêu cầu đặt ra trong bài báo cáo đó chính là theo dõi luồng dữ liệu từ máy kali tới máy mục tiêu, giao thức HTTP v1. Chúng ta có thể sử dụng câu lệnh filter dưới đây để thực hiện yêu cầu trên Wireshark:

http && ip.src_host=="192.168.40.131" && ip.dst_host == "192.168.40.134"

No.	Time	Source	Destination	Protocol	Length	Info
564	1392.889621	54.246.133.196	192.168.40.131	TCP	54	443 → 48161 [ACK] Seq=4828 Ack=347 Win=64240 Len=0
565	1393.602490	54.246.133.196	192.168.40.131	TLSv1	113	Change Cipher Spec, Encrypted Handshake Message
566	1393.603077	192.168.40.131	54.246.133.196	TCP	60	48161 → 443 [ACK] Seq=347 Ack=4887 Win=37960 Len=0
567	1393.605911	192.168.40.131	54.246.133.196	TLSv1	315	Application Data
568	1393.605913	54.246.133.196	192.168.40.131	TCP	54	443 → 48161 [ACK] Seq=4887 Ack=608 Win=64240 Len=0
569	1393.884751	54.246.133.196	192.168.40.131	TLSv1	859	Application Data
570	1393.885095	54.246.133.196	192.168.40.131	TCP	54	443 → 48161 [FIN, PSH, ACK] Seq=5692 Ack=608 Win=64240 Len=0
571	1393.887972	192.168.40.131	54.246.133.196	TLSv1	91	Encrypted Alert
572	1393.887974	54.246.133.196	192.168.40.131	TCP	54	443 → 48161 [ACK] Seq=5693 Ack=645 Win=64240 Len=0
573	1393.887975	192.168.40.131	54.246.133.196	TCP	60	48161 → 443 [FIN, ACK] Seq=645 Ack=5693 Win=40880 Len=0
574	1393.887976	54.246.133.196	192.168.40.131	TCP	54	443 → 48161 [ACK] Seq=5693 Ack=646 Win=64239 Len=0
575	1394.976170	Vmware_e9:d3:7e	Broadcast	ARP	60	Who has 192.168.40.134? Tell 192.168.40.131
576	1394.976301	Vmware_b7:cd:ee	Vmware_e9:d3:7e	ARP	42	192.168.40.134 is at 00:0c:29:b7:cd:ee
577	1394.981366	192.168.40.131	192.168.40.134	TCP	74	42522 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=506613 TSecr=0 WS=1024
578	1394.981367	192.168.40.134	192.168.40.131	TCP	74	80 → 42522 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=256338 TSecr=506613 WS=32
579	1394.981367	192.168.40.131	192.168.40.134	TCP	66	42522 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=506613 TSecr=256338
580	1394.981367	192.168.40.131	192.168.40.134	HTTP	367	GET / HTTP/1.1

Hình 13: Các luồng khi chưa lọc

No.	Time	Source	Destination	Protocol	Length	Info
580	1394.981367	192.168.40.131	192.168.40.134	HTTP	367	GET / HTTP/1.1
596	1395.884228	192.168.40.131	192.168.40.134	HTTP	348	GET /favicon.ico HTTP/1.1
606	1395.932405	192.168.40.131	192.168.40.134	HTTP	378	GET /favicon.ico HTTP/1.1
648	1431.996010	192.168.40.131	192.168.40.134	HTTP	371	OPTIONS / HTTP/1.1
676	1459.794247	192.168.40.131	192.168.40.134	HTTP	263	OPTIONS /mutillidae/ HTTP/1.1
855	1471.975164	192.168.40.131	192.168.40.134	HTTP	142	OPTIONS /mutillidae/ HTTP/1.1
1211	1563.645993	192.168.40.131	192.168.40.134	HTTP	411	GET /mutillidae/ HTTP/1.1
1264	1563.815690	192.168.40.131	192.168.40.134	HTTP	453	GET /mutillidae/styles/global-styles.css HTTP/1.1
1311	1563.872732	192.168.40.131	192.168.40.134	HTTP	451	GET /mutillidae/javascript/ddsmoothmenu/jquery.min.js HTTP/1.1
1314	1563.875745	192.168.40.131	192.168.40.134	HTTP	453	GET /mutillidae/javascript/ddsmoothmenu/ddsmoothmenu.js HTTP/1.1
1316	1563.875746	192.168.40.131	192.168.40.134	HTTP	467	GET /mutillidae/styles/ddsmoothmenu/ddsmoothmenu-v.css HTTP/1.1
1318	1563.875748	192.168.40.131	192.168.40.134	HTTP	441	GET /mutillidae/favicon.ico HTTP/1.1
1320	1563.875748	192.168.40.131	192.168.40.134	HTTP	441	GET /mutillidae/javascript/bookmark-site.js HTTP/1.1
1321	1563.875748	192.168.40.131	192.168.40.134	HTTP	465	GET /mutillidae/styles/ddsmoothmenu/ddsmoothmenu.css HTTP/1.1
1464	1565.535612	192.168.40.131	192.168.40.134	HTTP	468	GET /mutillidae/images/coykillericon.png HTTP/1.1
1476	1565.544132	192.168.40.131	192.168.40.134	HTTP	473	GET /mutillidae/images/owasp-logo-400-300.png HTTP/1.1
1481	1565.554529	192.168.40.131	192.168.40.134	HTTP	462	GET /mutillidae/images/twitter.gif HTTP/1.1

Hình 14: Luồng nhận được sau khi lọc

RFC 2616 định nghĩa ra 8 phương thức cho HTTP 1.1. Các phương thức này là: GET, POST, HEAD, PUT, DELETE, TRACE, OPTIONS và CONNECT. Cần chú ý rằng, các phương thức kể trên không chỉ giúp lập trình viên dễ dàng chỉnh sửa, thiết kế lên ứng dụng web của mình, mà một số kẻ tấn công có thể lợi dụng các đặc điểm của phương thức để tiến hành khai thác và tấn công.

Khi sử dụng OPTIONS, client sẽ hỏi server: “Hey Server, bạn hỗ trợ các phương thức nào trong 8 phương thức, hãy kể cho tôi nha!!!”. Điều này cung cấp thông tin cho Hacker để xác định phương thức đang khả dụng & các phương pháp tấn công nào có thể sử dụng.

Phương thức TRACE cho phép client nhìn thấy những yêu cầu của bản thân khi các yêu cầu này được nhận và phản hồi từ phía server, lợi dụng điều này Hacker có thể tấn công Cross-site Tracing (XST) - https://www.owasp.org/index.php/Cross_Site_Tracing

Phương thức PUT và DELETE là hai phương thức nguy hiểm nhất khi chúng có thể gây ra các rủi ro bảo mật lớn cho ứng dụng. PUT cho phép tải lên bất kỳ loại dữ liệu độc hại nào lên máy chủ. DELETE thì ngược lại, cho phép loại bỏ bất kỳ tài nguyên nào từ ứng dụng web, ví dụ như xóa các tập tin cấu hình (Web configuration file).

Trong luồng nhận được sau khi lọc, chúng ta thấy OPTIONS method và TRACE method đang được yêu cầu từ phía máy Kali.

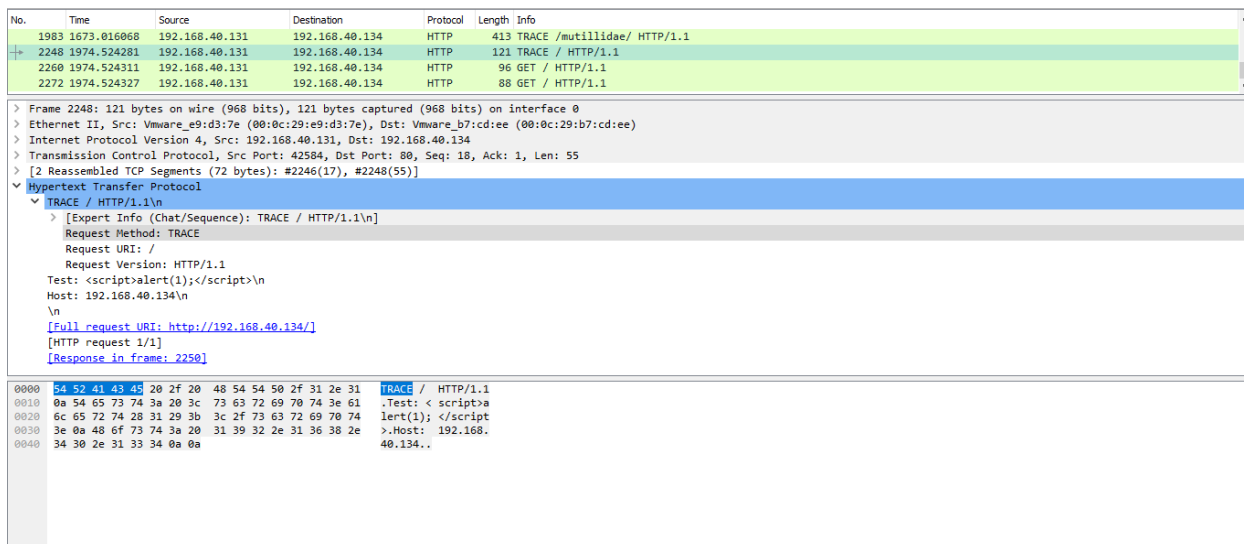
No.	Time	Source	Destination	Protocol	Length	Info
580	1394.981367	192.168.40.131	192.168.40.134	HTTP	367	GET / HTTP/1.1
596	1395.884228	192.168.40.131	192.168.40.134	HTTP	348	GET /favicon.ico HTTP/1.1
606	1395.932405	192.168.40.131	192.168.40.134	HTTP	378	GET /favicon.ico HTTP/1.1
648	1431.996010	192.168.40.131	192.168.40.134	HTTP	371	OPTIONS / HTTP/1.1

```

> Frame 648: 371 bytes on wire (2968 bits), 371 bytes captured (2968 bits) on interface 0
> Ethernet II, Src: Vmware_e9:d3:7e (00:0c:29:e9:d3:7e), Dst: Vmware_b7:cd:ee (00:0c:29:b7:cd:ee)
> Internet Protocol Version 4, Src: 192.168.40.131, Dst: 192.168.40.134
> Transmission Control Protocol, Src Port: 42528, Dst Port: 80, Seq: 1, Ack: 1, Len: 305
  Hypertext Transfer Protocol
    OPTIONS / HTTP/1.1\r\n
      [Expert Info (Chat/Sequence): OPTIONS / HTTP/1.1\r\n]
      Request Method: OPTIONS
      Request URI: /
      Request Version: HTTP/1.1
      Host: 192.168.40.134\r\n
      User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:31.0) Gecko/20100101 Firefox/31.0 Iceweasel/31.8.0\r\n
      Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
      Accept-Language: en-US,en;q=0.5\r\n
      Accept-Encoding: gzip, deflate\r\n
      Connection: keep-alive\r\n
      \r\n
      [Full request URI: http://192.168.40.134/]
      [HTTP request 1/1]
      [Response in frame: 650]
  
```

Offset	Hex	ASCII
0040	f7 a0 4f 50 54 49 4f 4e 53 20 2f 20 48 54 54 50	..OPTION S / HTTP
0050	2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 31 39 32 2e	/1.1..Ho st: 192.
0060	31 36 38 2e 34 30 2e 31 33 34 0d 0a 55 73 65 72	168.40.1 34..User
0070	2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f	-Agent: Mozilla/
0080	35 2e 30 20 2d 58 31 31 3b 20 4c 69 6e 75 78 20	5.0 (X11 ; Linux
0090	78 38 36 5f 36 34 3b 20 72 76 3a 33 31 2e 30 29	x86_64; rv:31.0)
00a0	20 47 65 63 6b 6f 2f 32 30 31 30 30 31 30 31 20	Gecko/2 0100101
00b0	46 69 72 65 66 6f 78 2f 33 31 2e 30 20 49 63 65	Firefox/ 31.0 Ice
00c0	77 65 61 73 65 6c 2f 33 31 2e 38 2e 30 0d 0a 41	weasel/3 1.8.0..A
00d0	63 65 67 70 74 3a 20 74 65 78 74 2f 68 74 6d 6c	cept: t ext/html

Hình 15: OPTIONS từ wireshark

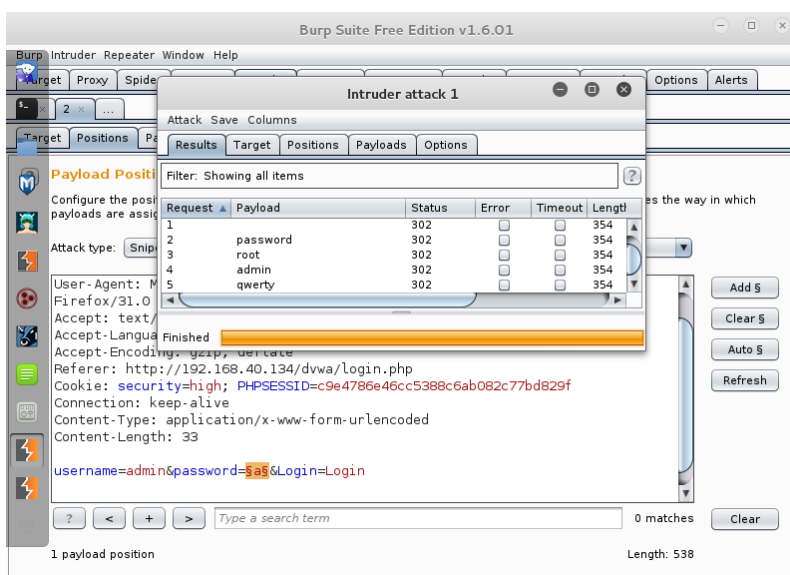


Hình 16: TRACE từ Wireshark

Rất nhiều website sử dụng chức năng xác thực cơ bản như đăng nhập, đây cũng là một trong những chức năng mà Hacker rất thích để tấn công khai thác cơ chế đăng nhập. Về mặt cơ bản, kẻ tấn công sẽ thử các username và password cho đến khi đăng nhập thành công và chiếm được thành công tài khoản. Hầu hết các cuộc tấn công kiểu này đều kết hợp hai dạng: tấn công vét cạn và tấn công từ điển bằng các công cụ tự động & từ điển.

Trong trường hợp này, máy tấn công Kali đã sử dụng Burpsuite và một số tham số cơ bản để tấn công mật khẩu như sau:

- Admin – password, Admin – a , Admin – #rỗng
- Admin – root
- Admin – admin
- Admin – qwerty



Hình 17: Kali tấn công mật khẩu

Với wireshark, ta sẽ thấy 6 gói tin POST chứa các tham số tấn công vét cạn từ Kali

No.	Time	Source	Destination	Protocol	Length	Info
7673	14908.161986	192.168.40.131	192.168.40.134	HTTP	597	POST /dwa/login.php HTTP/1.1 (application/x-www-form-urlencoded)
7683	14908.330014	192.168.40.131	192.168.40.134	HTTP	596	POST /dwa/login.php HTTP/1.1 (application/x-www-form-urlencoded)
7693	14908.536308	192.168.40.131	192.168.40.134	HTTP	604	POST /dwa/login.php HTTP/1.1 (application/x-www-form-urlencoded)
7703	14908.794609	192.168.40.131	192.168.40.134	HTTP	600	POST /dwa/login.php HTTP/1.1 (application/x-www-form-urlencoded)
7714	14909.150799	192.168.40.131	192.168.40.134	HTTP	601	POST /dwa/login.php HTTP/1.1 (application/x-www-form-urlencoded)
7724	14909.563459	192.168.40.131	192.168.40.134	HTTP	602	POST /dwa/login.php HTTP/1.1 (application/x-www-form-urlencoded)

> Frame 7673: 597 bytes on wire (4776 bits), 597 bytes captured (4776 bits) on interface 0
 > Ethernet II, Src: Vmware_e9:d3:7e (00:0c:29:e9:d3:7e), Dst: Vmware_b7:cd:ee (00:0c:29:b7:cd:ee)
 > Internet Protocol Version 4, Src: 192.168.40.131, Dst: 192.168.40.134
 > Transmission Control Protocol, Src Port: 42620, Dst Port: 80, Seq: 1, Ack: 1, Len: 531
 > Hypertext Transfer Protocol
 > HTML Form URL Encoded: application/x-www-form-urlencoded
 > Form item: "username" = "admin"
 > Form item: "password" = "a"
 > Form item: "Login" = "Login"

Hình 18: Kết quả của tấn công mật khẩu từ Wireshark

Vì sao chúng ta có thể khẳng định đây là tấn công vét cạn từ các công cụ tự động? Vì trong kết quả từ wireshark, chúng ta thấy có 6 gói tin POST chứa các thông số username và password khác nhau và được yêu cầu đến trang login trong khoảng thời gian dưới 0,5s.

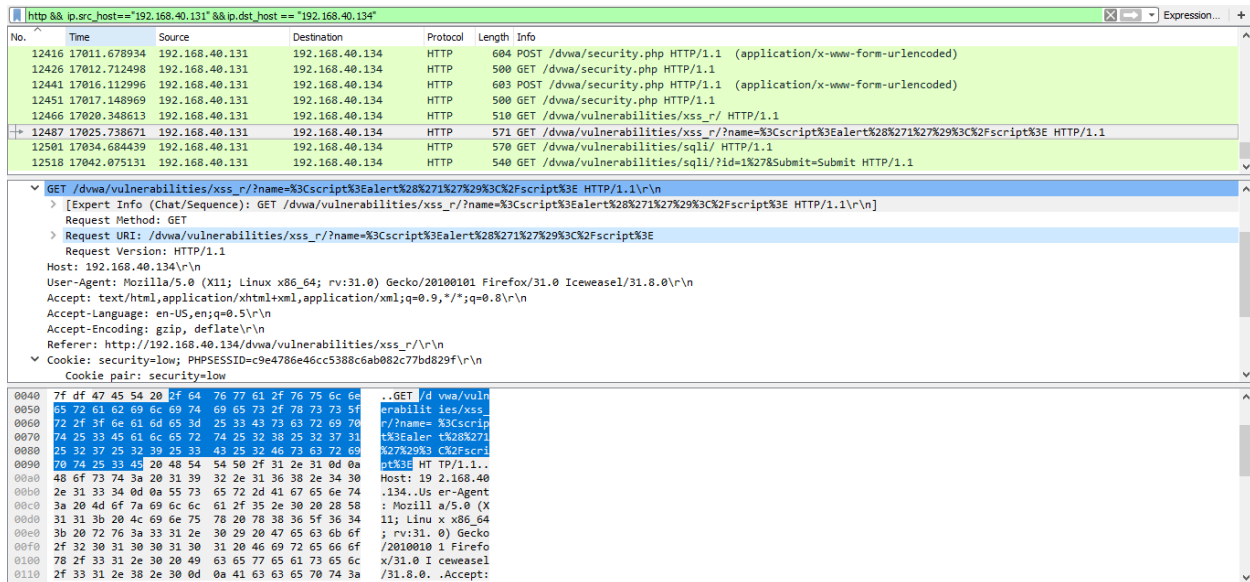
Khi Hacker nhắm tới một trang web, việc thu thập thông tin và nắm được cấu trúc của ứng dụng là điều thiết yếu. Tất nhiên, những kẻ tấn công sẽ không sử dụng các kỹ thuật duyệt thủ công để thống kê nội dung, mà họ sử dụng các kỹ thuật Spidering tự động để phục vụ bản thân.

No.	Time	Source	Destination	Protocol	Length	Info
8100	15986.887877	192.168.40.131	192.168.40.134	HTTP	352	GET /dav/ HTTP/1.1
8113	15986.910175	192.168.40.131	192.168.40.134	HTTP	352	GET /dav/ HTTP/1.1
8116	15986.910198	192.168.40.131	192.168.40.134	HTTP	322	GET /eb6de07 HTTP/1.1
8132	15986.915888	192.168.40.131	192.168.40.134	HTTP	325	GET /robots.txt HTTP/1.1
8142	15986.916227	192.168.40.131	192.168.40.134	HTTP	333	GET /mutillidae/styles/ HTTP/1.1
8152	15986.917655	192.168.40.131	192.168.40.134	HTTP	394	GET /mutillidae/styles/global-styles.css HTTP/1.1
8167	15986.930393	192.168.40.131	192.168.40.134	HTTP	346	GET /mutillidae/styles/ddsmoothmenu/ HTTP/1.1
8178	15986.930415	192.168.40.131	192.168.40.134	HTTP	406	GET /mutillidae/styles/ddsmoothmenu/ddsmoothmenu.css HTTP/1.1
8212	15986.930450	192.168.40.131	192.168.40.134	HTTP	337	GET /mutillidae/javascript/ HTTP/1.1
8215	15986.930451	192.168.40.131	192.168.40.134	HTTP	397	GET /mutillidae/javascript/bookmark-site.js HTTP/1.1
8233	15986.930476	192.168.40.131	192.168.40.134	HTTP	409	GET /mutillidae/javascript/ddsmoothmenu/ddsmoothmenu.js HTTP/1.1
8244	15986.930488	192.168.40.131	192.168.40.134	HTTP	350	GET /mutillidae/javascript/ddsmoothmenu/ HTTP/1.1
8260	15986.934633	192.168.40.131	192.168.40.134	HTTP	407	GET /mutillidae/javascript/ddsmoothmenu/jquery.min.js HTTP/1.1
8320	15986.936390	192.168.40.131	192.168.40.134	HTTP	333	GET /mutillidae/images/ HTTP/1.1
8334	15986.941987	192.168.40.131	192.168.40.134	HTTP	408	GET /mutillidae/styles/ddsmoothmenu/ddsmoothmenu-v.css HTTP/1.1
8346	15986.942031	192.168.40.131	192.168.40.134	HTTP	379	GET /mutillidae/index.php HTTP/1.1
8348	15986.942032	192.168.40.131	192.168.40.134	HTTP	393	GET /mutillidae/index.php?page=home.php HTTP/1.1
8351	15986.942032	192.168.40.131	192.168.40.134	HTTP	394	GET /mutillidae/index.php?page=login.php HTTP/1.1
8392	15987.066381	192.168.40.131	192.168.40.134	HTTP	409	GET /mutillidae/index.php?do=toggle-hints&page=home.php HTTP/1.1
8433	15987.110695	192.168.40.131	192.168.40.134	HTTP	412	GET /mutillidae/index.php?do=toggle-security&page=home.php HTTP/1.1
8639	15987.160705	192.168.40.131	192.168.40.134	HTTP	389	GET /mutillidae/set-up-database.php HTTP/1.1
8650	15987.209205	192.168.40.131	192.168.40.134	HTTP	410	GET /mutillidae/index.php?page=show-log.php HTTP/1.1
8664	15987.294299	192.168.40.131	192.168.40.134	HTTP	415	GET /mutillidae/index.php?page=captured-data.php HTTP/1.1
8884	15987.349670	192.168.40.131	192.168.40.134	HTTP	409	GET /mutillidae/index.php?page=credits.php HTTP/1.1
8985	15987.558513	192.168.40.131	192.168.40.134	HTTP	411	GET /mutillidae/index.php?page=user-info.php HTTP/1.1
8987	15987.558596	192.168.40.131	192.168.40.134	HTTP	420	GET /mutillidae/index.php?page=view-someones-blog.php HTTP/1.1
8989	15987.558598	192.168.40.131	192.168.40.134	HTTP	410	GET /mutillidae/index.php?page=register.php HTTP/1.1
9066	15987.627654	192.168.40.131	192.168.40.134	HTTP	409	GET /mutillidae/?page=add-to-your-blog.php HTTP/1.1

Hình 19: Phát hiện tấn công Spider từ Wireshark

Tương tự với cách phát hiện tấn công mật khẩu, ta thấy rằng trong khoảng thời gian 1s, đã có rất nhiều GET request tới các đường dẫn của website Metasploitable 2, điều mà một người thực hiện thủ công không thể làm được!

Cross Site Scripting (XSS) hay còn được biết tới cái tên “script injection”. Cho phép kẻ tấn công chèn các đoạn javascript độc hại tới ứng dụng web nhằm cho trình duyệt chạy nó. Có hai loại XSS đó là: stored và reflected. Để khám phá lỗi XSS, cách đơn giản nhất, Hacker sẽ nhập vào đầu vào đoạn mã như sau `<script>alert('1')</script>` vào một trường ngẫu nhiên và quan sát nếu trình duyệt thực thi và xuất hiện popup thông báo “1”, nghĩa là website đã bị lỗi XSS.



Hình 20: Phát hiện tấn công XSS từ wireshark

Ta thấy thông qua GET request, payload `<script>alert('1')</script>` đã được url encode thành `%3Cscript%3Ealert(%27%27)%3C%2Fscript%3E`, đây là một dấu hiệu để phân tích và phát hiện tấn công XSS. Ngoài ra, các payload độc hại cũng có thể được gửi qua POST request, tùy thuộc vào từng trang web. Tương tự với XSS, ta cũng có thể phát hiện các trường hợp tấn công injection tương tự như: SQL injection, path traversal,... thông qua wireshark.

4.2. Phân tích tập tin nhật ký

a. Khái niệm liên quan

Tập nhật ký máy chủ web (web server log file)

Các web server chuẩn như Apache và IIS tạo thông điệp ghi nhật ký theo một chuẩn chung (CLF – common log format). Tập nhật ký CLF chứa các dòng thông điệp cho mỗi một gói HTTP request, cấu tạo như sau:

Host Ident Authuser Date Request Status Bytes

Trong đó:

- Host: Tên miền đầy đủ của client hoặc IP
- Ident: Nếu chỉ thị IdentityCheck được kích hoạt và client chạy identd, thì đây là thông tin nhận dạng được client báo cáo
- Authuser: Nếu URL yêu cầu xác thực HTTP thì tên người dùng là giá trị của mã thông báo này
- Date: Ngày và giờ yêu cầu
- Request: Dòng yêu cầu của client, được đặt trong dấu ngoặc kép (“”)
- Status: Mã trạng thái (gồm ba chữ số)
- Bytes: số bytes trong đối tượng trả về cho client, ngoại trừ các HTTP header
- Mỗi yêu cầu có thể chứa các dữ liệu bổ sung như đường liên kết hoặc chuỗi ký tự của người dùng.

Nếu mã thông báo không có giá trị, thì mã thông báo được biểu thị bằng một dấu gạch ngang (-).

Ví dụ:

```
192.168.40.131 - - [08/May/2018:08:43:52 -0400] "GET /dvwa/login.php HTTP/1.1" 200 1289 "-"
"Mozilla/5.0 (X11; Linux x86_64; rv:31.0) Gecko/20100101 Firefox/31.0 Iceweasel/31.8.0"
```

Lợi ích lớn nhất của tập tin nhật ký là tính sẵn có tương đối đơn giản và phân tích nội dung của chúng. Máy chủ web như Apache mặc định phải cho phép ghi nhật ký. Các ứng dụng thường thực hiện ghi nhật ký để đảm bảo truy xuất nguồn gốc của các hành động của chúng. Trong khi lưu lượng mạng đầy đủ cung cấp các thông tin bổ sung, chi phí mua lại và xử lý của nó thường lớn hơn lợi ích của nó. Việc thu thập lưu lượng mạng yêu cầu: trong suốt với gói tin và thường là phần cứng bổ sung. Quan sát lưu lượng có thể đạt được với hubs, các công SPAN, vòi hoặc thiết bị nội tuyến. Mọi thiết bị đều phải mua, cài đặt và được hỗ trợ. Một khi dữ liệu đã được thu thập thì sẽ được phân tích ngay lập tức. Hiện tại, lưu lượng truy cập mạng được thu thập có cùng dạng với tệp nhật ký và sẵn sàng để được phân tích. Cuối cùng, các tệp nhật ký cung cấp khả năng dễ dàng và dễ xử lý để theo dõi bảo mật.

b. Ứng dụng Regular expression trong phân tích tập tin nhật ký tự động

Các phương pháp phân tích tập tin nhật ký thủ công & phát hiện tấn công theo dấu hiệu luôn là các phương pháp hiệu quả về mặt kết quả, tuy nhiên sẽ mất rất nhiều thời gian và công sức để phân tích log file, vì log file thường chứa rất nhiều dòng nhật ký. Vì vậy Regular expression là lựa chọn của bài báo cáo này.

Regular là gì? Regex cho phép xử lý các chuỗi ký tự linh hoạt, hiệu quả và mạnh mẽ. Regex cho phép bạn mô tả và phân tích chuỗi ký tự với các bản mẫu tương tự như một ngôn ngữ lập trình nhỏ. Regex có trong nhiều dạng công cụ, nhưng sức mạnh của nó chỉ được thể hiện tối đa khi là 1 phần của một ngôn ngữ lập trình.

Dưới đây là đoạn code viết bằng python, sử dụng Regular Expression trong việc phân tích tập tin nhật ký web phát hiện tấn công XSS.

```
import os, sys, re
from collections import Counter
from subprocess import call

PATH = sys.argv[1]
TYPE = sys.argv[2]

if TYPE == 'access':
    log = 'access.log'

elif TYPE == 'error':
    log = 'error.log'

f = open(PATH+log, 'r')
ipList = []
```

```

xss_match='(.(POST\s+|GET\s+|HEAD\s+|PUT\s+|OPTION\s+).+?=.+?(S|s)(C|c)(R|r)(I|i)(P|p)(T|t)(S|s)(E|e)(L|l)(F|f)(A|a)(L|l)(E|e)(R|r)(T|t).+?HTTP/[0-9]\.[0-9].+)'
time_regex = re.compile("([0-9]{2}:[0-9]{2}:[0-9]{2}\s+)")
date_regex = re.compile("(\d{2}|\d{4})/(\d{2}|\w{3})/(\d{2}|\d{4})(?:\:\s+)"")
ip_regex = "(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})"
ip_regexsearch = re.compile(ip_regex)
xss_payload_regex=re.compile("((POST\s+|GET\s+|HEAD\s+|PUT\s+|OPTION\s+).+?=.+?(S|s)(C|c)(R|r)(I|i)(P|p)(T|t)(S|s)(E|e)(L|l)(F|f)(A|a)(L|l)(E|e)(R|r)(T|t)(J|j)(A|a)(V|v)(A|a)(S|s)(C|c)(R|r)(I|i)(P|p)(T|t)):\:(X|x)(S|S)(S|s).+?HTTP/[0-9]\.[0-9].+)"")

for line in f.read().split('\n'):
    if re.match(xss_match, line):
        dateData = date_regex.search(line)
        timeData = time_regex.search(line)
        ipData = ip_regexsearch.search(line)
        if re.match(xss_match, line):
            payloadType = "XSS"
            payloadData = xss_payload_regex.search(line)
print "["+payloadType+"] "+dateData.group(0)+" | "+timeData.group(0)+"|"+ ipData.group(0)+" | "+payloadData.group(0)

```

Ứng dụng tool vừa viết với Access.log trích xuất từ Metasploitable 2 trong phần trên ta có kết quả tương tự khi sử dụng wireshark để phát hiện tấn công XSS.

```

access.log
~/
REExample.py x
access.log x
192.168.40.131 - - [08/May/2018:08:43:51 -0400] "GET /dvwa HTTP/1.1" 301 320 "-"
"Mozilla/5.0 (X11; Linux x86_64; rv:31.0) Gecko/20100101 Firefox/31.0
Iceweasel/31.8.0"
192.168.40.131 - - [08/May/2018:08:43:51 -0400] "GET /dvwa/ HTTP/1.1" 302 - "-"
"Mozilla/5.0 (X11; Linux x86_64; rv:31.0) Gecko/20100101 Firefox/31.0
Iceweasel/31.8.0"
192.168.40.131 - - [08/May/2018:08:43:52 -0400] "GET /dvwa/login.php HTTP/1.1" 200
1289 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:31.0) Gecko/20100101 Firefox/31.0
Iceweasel/31.8.0"
192.168.40.131 - - [08/May/2018:08:43:52 -0400] "GET /dvwa/dvwa/css/login.css
HTTP/1.1" 200 608 "http://192.168.40.134/dvwa/login.php" "Mozilla/5.0 (X11; Linux
x86_64; rv:31.0) Gecko/20100101 Firefox/31.0 Iceweasel/31.8.0"
192.168.40.131 - - [08/May/2018:08:43:53 -0400] "GET /dvwa/dvwa/images/login_logo.png
HTTP/1.1" 200 12875 "http://192.168.40.134/dvwa/login.php" "Mozilla/5.0 (X11; Linux
x86_64; rv:31.0) Gecko/20100101 Firefox/31.0 Iceweasel/31.8.0"
192.168.40.131 - - [08/May/2018:08:43:56 -0400] "POST /dvwa/login.php HTTP/1.1" 302 -
"http://192.168.40.134/dvwa/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:31.0)
Gecko/20100101 Firefox/31.0 Iceweasel/31.8.0"
192.168.40.131 - - [08/May/2018:08:43:57 -0400] "GET /dvwa/login.php HTTP/1.1" 200
1328 "http://192.168.40.134/dvwa/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:31.0)
Gecko/20100101 Firefox/31.0 Iceweasel/31.8.0"
192.168.40.131 - - [08/May/2018:08:56:48 -0400] "POST /dvwa/login.php HTTP/1.1" 302 -
"http://192.168.40.134/dvwa/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:31.0)
Gecko/20100101 Firefox/31.0 Iceweasel/31.8.0"
192.168.40.131 - - [08/May/2018:08:56:48 -0400] "POST /dvwa/login.php HTTP/1.1" 302 -
"http://192.168.40.134/dvwa/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:31.0)

```

Hình 21: Access.log từ máy Metasploitable 2

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# python REExample.py /root/ access
[XSS] 08/May/2018: | 09:31:43 |192.168.40.131 | GET /dwa/vulnerabilities/xss_r/?name=%
3Cscript%3Ealert%28%271%27%29%3C%2Fscript%3E HTTP/1.1" 200 4361 "http://192.168.40.134/
dwa/vulnerabilities/xss_r/" "Mozilla/5.0 (X11; Linux x86_64; rv:31.0) Gecko/20100101 F
irefox/31.0 Iceweasel/31.8.0"
```

Hình 22: Kết quả sau khi ứng dụng Regex

References:

<https://www.sans.org/reading-room/whitepapers/logging/detecting-attacks-web-applications-log-files-2074>

<https://www.amazon.com/Web-Application-Hackers-Handbook-Exploiting/dp/1118026470>

<https://docs.python.org/2/library/re.html>

About authors:

I am ManhNho (AKA Manh Pham Tien), a very young researcher passionate in penetration testing, web security / exploit, cryptography & network security