Bộ Giáo Dục Và Đào Tạo
Trường Đại Học Ngoại Ngữ - Tin Học Thành Phố Hồ Chí Minh
Khoa Công Nghệ Thông Tin

MÔN HỌC : ĐIỀU TRA TẤN CÔNG

ĐỀ TÀI: Tìm hiểu về NETWORK FORENSICS

Giảng Viên Hướng Dẫn : **Phạm Đình Thắng**

Thành Viên Nhóm:

1. Võ Ngọc Trọng - MSSV: 22DH114790
2. Đào Đức Lương - MSSV: 22DH114621
3. Ngô Thế Đức - MSSV: 22DH114504

Tp. Hồ chí minh, Ngày 11, tháng 10, năm 2024

Nhận xét của giảng viên

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Mục lục

Network Forensics

Danh mục hình ảnh

Lời cảm ơn

Kính gửi Giảng Viên Hướng Dẫn : Phạm Đình Thắng,

Chúng tôi, nhóm sinh viên của lớp 241125011402, xin được gửi lời cảm ơn chân thành đến thầy vì sự hướng dẫn tận tâm và chuyên nghiệp trong quá trình thực hiện đồ án môn đồ án mạng.

Trước khi được thầy hướng dẫn, chúng tôi đã gặp nhiều khó khăn và bối rối trong việc xác định và triển khai các bước cần thiết để hoàn thành đồ án.

Tuy nhiên, sự hỗ trợ tận tâm của thầy đã giúp chúng tôi vượt qua những khó khăn đó một cách hiệu quả.

Thầy đã không chỉ giúp chúng tôi hiểu rõ hơn về quy trình làm đồ án, mà còn truyền đạt những kiến thức quan trọng và kinh nghiệm thực tế từ những dự án đã từng tham gia. Nhờ đó, chúng tôi đã có thể áp dụng những kiến thức đó vào đồ án của mình.

Không chỉ là một giảng viên, thầy còn là một người đồng hành tận tụy và đáng tin cậy trong suốt quá trình thực hiện đồ án. Thầy luôn sẵn lòng lắng nghe và trả lời những câu hỏi của chúng tôi một cách chi tiết và rõ ràng. Thầy đã tạo ra một môi trường học tập tích cực và khích lệ chúng tôi tự tin thể hiện ý kiến và ý tưởng của mình.

Chúng tôi biết rằng những kiến thức và kỹ năng mà chúng tôi đã học được từ thầy sẽ có giá trị lớn trong sự nghiệp và cuộc sống của chúng tôi. Chúng tôi sẽ luôn ghi nhớ những lời khuyên và chỉ dẫn của thầy để ngày càng trở nên giỏi hơn và đóng góp tốt hơn cho ngành nghề của mình.

Một lần nữa, chúng tôi xin chân thành cảm ơn thầy Thắng vì sự hướng dẫn tận tâm và những đóng góp quý báu của thầy trong quá trình thực hiện đồ án. Thầy là một người giảng viên xuất sắc và đáng ngưỡng mộ.

Chúng tôi chúc thầy luôn khỏe mạnh, thành công trong công việc và có thêm nhiều niềm vui trong cuộc sống.

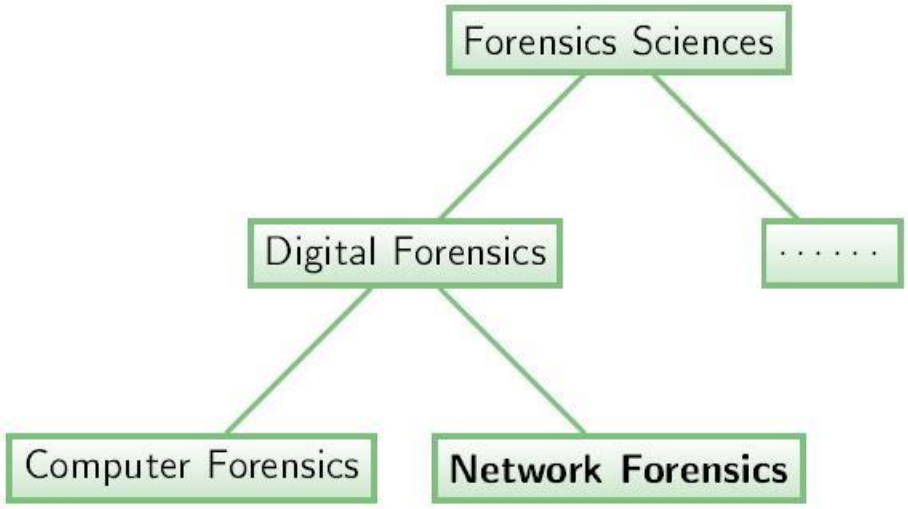
Trân trọng,

Tập thể thành viên trong nhóm.

I. Cơ sở lý thuyết:

1 Tổng quan về Network Forensics:

- Thuật ngữ Network forensics (điều tra mạng) được đưa ra bởi chuyên gia bảo mật máy tính Marcus Ranum vào đầu những năm 90, vay mượn từ các lĩnh vực pháp luật và tội phạm nơi mà "forensics" gắn liền với việc điều tra các hành vi phạm tội.



Hình 1. Network forensics trong Forensics sciences

- Network forensics là một nhánh của digital forensics (điều tra kĩ thuật số) liên quan đến việc giám sát và phân tích lưu lượng mạng máy tính nhằm phục vụ cho việc thu thập thông tin, chứng cứ pháp lý hay phát hiện các xâm nhập. Network forensics cũng được hiểu như Digital Forensics trong môi trường mạng.
- Không giống như các lĩnh vực khác của digital forensics, network forensics xử lý thông tin động và không ổn định. Lưu lượng mạng được truyền đi rồi bị mất, do đó giám định mạng thường là một cuộc điều tra chủ động.

2 Quy trình phân tích:

- **O - Obtain Information (Thu thập thông tin)**
  - Mục tiêu của bước này là thu thập **tất cả các thông tin** liên quan đến hệ thống hoặc mạng mà bạn đang đánh giá. Điều này bao gồm:
  - **Thông tin kỹ thuật**: Các địa chỉ IP, tên miền, cấu hình hệ thống, phiên bản phần mềm.

- **Thông tin từ nguồn mở (OSINT)**: Các thông tin công khai có thể thu thập từ internet, mạng xã hội, hoặc các cơ sở dữ liệu công khai.

❖ **S - Scan for Vulnerabilities (Quét lỗ hổng)**
- Tiến hành **quét hệ thống và mạng** để tìm kiếm các **lỗ hổng bảo mật** tiềm ẩn:
- Sử dụng các công cụ như **Nmap**, **OpenVAS**, hoặc **Nessus** để quét cổng và phát hiện các dịch vụ đang chạy.
- Kiểm tra các lỗ hổng bảo mật trên các ứng dụng web, hệ điều hành và phần mềm sử dụng.

❖ **C - Control Weaknesses (Kiểm soát điểm yếu)**
- Đánh giá các biện pháp kiểm soát bảo mật hiện có và **xác định các điểm yếu**:
- Xác minh **chính sách bảo mật**, **mật khẩu**, **cấu hình tường lửa**, và **quyền truy cập**.
- Kiểm tra tính hiệu quả của **các biện pháp kiểm soát truy cập**, **mã hóa dữ liệu**, và **hệ thống phát hiện xâm nhập (IDS)**.

❖ **A - Analyze the Findings (Phân tích kết quả)**
- Sau khi thu thập dữ liệu và quét các lỗ hổng, tiến hành **phân tích dữ liệu** để hiểu rõ **mức độ nghiêm trọng** của các rủi ro phát hiện được.
- **Ưu tiên** xử lý các lỗ hổng theo mức độ nguy hiểm và khả năng bị khai thác.

❖ **R - Report and Mitigate (Báo cáo và giảm thiểu rủi ro)**
- Viết **báo cáo chi tiết** về những gì đã phát hiện, bao gồm:
- Các lỗ hổng cụ thể và mức độ nghiêm trọng.
- Đề xuất **biện pháp khắc phục** cho từng lỗ hổng.
- Hỗ trợ doanh nghiệp hoặc tổ chức **áp dụng các biện pháp bảo mật** để giảm thiểu rủi ro và tăng cường khả năng phòng thủ.

3 Công dụng của Network Forensics

❖ Network forensics thường có hai công dụng.

- Thực thi pháp luật
  - Phân tích lưu lượng mạng đã thu thập có thể bao gồm các tác vụ như lắp ráp lại các tệp đã chuyển, tìm kiếm từ khóa và phân tích cú pháp giao tiếp của con người như email hoặc phiên trò chuyện.
- Giám sát mạng để tìm lưu lượng bất thường và xác định các cuộc xâm nhập:
  - Kẻ tấn công có thể xóa tất cả các tệp nhật ký trên máy chủ bị xâm phạm; do đó, bằng chứng dựa trên mạng có thể là bằng chứng duy nhất có sẵn để phân tích, giám định.

4 Các dạng hệ thống sử dụng

- "Catch-it-as-you-can" – Đây là nơi tất cả các gói tin đi qua một điểm lưu lượng nhất định được bắt và ghi vào bộ nhớ với việc phân tích được thực hiện sau đó ở chế độ hàng loạt. Phương pháp này đòi hỏi một lượng lớn bộ nhớ.
- "Stop, look and listen" – Đây là nơi mỗi gói được phân tích theo cách thô sơ trong bộ nhớ và chỉ lưu một số thông tin nhất định để phân tích trong tương lai. Cách tiếp cận này yêu cầu bộ xử lý nhanh hơn để theo kịp lưu lượng truy cập đến.

5 Các dạng Network Forensics

5.1 Mạng Ethernet

- Với các công cụ này, các trang web, tệp đính kèm email và lưu lượng mạng khác chỉ có thể được tái tạo nếu chúng được truyền hoặc nhận mà không được mã hóa. Một lợi thế của việc thu thập dữ liệu này là nó được kết nối trực tiếp với máy chủ.
- Ví dụ, nếu biết địa chỉ IP hoặc địa chỉ MAC của máy chủ tại một thời điểm nhất định, tất cả dữ liệu được gửi đến hoặc từ địa chỉ IP hoặc MAC này đều có thể được lọc.
- Để thu thập dữ liệu trên lớp này, card giao diện mạng (NIC) của máy chủ có thể được đưa vào " chế độ hỗn tạp (promiscuous mode) ". Khi làm như vậy, tất cả lưu lượng sẽ được chuyển đến CPU, không chỉ lưu lượng dành cho máy chủ.
- Tuy nhiên, nếu kẻ xâm nhập hoặc kẻ tấn công biết rằng kết nối của mình có thể bị nghe lén, hắn có thể sử dụng mã hóa để bảo mật kết nối của mình.
- Ngày nay, gần như không thể phá vỡ mã hóa nhưng thực tế là kết nối của nghi phạm với máy chủ khác được mã hóa mọi lúc có thể chỉ ra rằng máy chủ kia là đồng phạm của nghi phạm.

5.2 TCP/IP

- Ở lớp mạng, Internet Protocol (IP) chịu trách nhiệm định hướng các gói tin do TCP tạo ra qua mạng (ví dụ: Internet) bằng cách thêm thông tin nguồn và đích có thể được các bộ định tuyến trên toàn mạng diễn giải.
- Các mạng gói kỹ thuật số di động như GPRS, sử dụng các giao thức tương tự như IP, do đó các phương pháp được mô tả cho IP cũng hoạt động với chúng.
- Để định tuyến đúng, mọi bộ định tuyến trung gian phải có một bảng định tuyến để biết nơi gửi gói tin tiếp theo.

- Các bảng định tuyến này là một trong những nguồn thông tin tốt nhất nếu điều tra tội phạm kỹ thuật số và cố gắng truy tìm kẻ tấn công. Để làm được điều này, cần phải theo dõi các gói tin của kẻ tấn công, đảo ngược tuyến đường gửi và tìm máy tính mà gói tin đến từ đó (tức là kẻ tấn công).

5.3 Phân thích lưu lượng được mã hóa

- Với sự gia tăng của mã hóa TLS trên internet, tính đến tháng 4 năm 2021, ước tính một nửa số phần mềm độc hại sử dụng TLS để tránh bị phát hiện. Phân tích lưu lượng được mã hóa kiểm tra lưu lượng để xác định lưu lượng được mã hóa đến từ phần mềm độc hại và các mối đe dọa khác bằng cách phát hiện các kết hợp đáng ngờ của các đặc điểm TLS, thường là đến các mạng không phổ biến hoặc máy chủ. Một cách tiếp cận khác để phân tích lưu lượng được mã hóa sử dụng cơ sở dữ liệu dấu vân tay được tạo ra, mặc dù các kỹ thuật này đã bị chỉ trích là dễ bị tin tặc bỏ qua và không chính xác.

5.4 Internet

- Internet có thể là nguồn bằng chứng kỹ thuật số phong phú bao gồm trình duyệt web, email, nhóm tin tức , cuộc trò chuyện đ trò chuyện đồng bộ và lưu lượng ngang hàng . Ví dụ, nhật ký máy chủ web có thể được sử dụng để hiển thị khi nào (hoặc nếu) nghi phạm truy cập thông tin liên quan đến hoạt động tội phạm. Tài khoản email thường có thể chứa bằng chứng hữu ích; nhưng tiêu đề email dễ bị làm giả và do đó, Network forensics có thể được sử dụng để chứng minh nguồn gốc chính xác của tài liệu có tính buộc tội. Network forensicscũng có thể được sử dụng để tìm ra ai đang sử dụng một máy tính cụ thể bằng cách trích xuất thông tin tài khoản người dùng từ lưu lượng mạng.

6 Wireless Forensics

- Wireless Forensics là một phân ngành của Network Forensics. Mục tiêu chính của Wireless Forensics là cung cấp phương pháp luận và công cụ cần thiết để thu thập và phân tích (không dây) lưu lượng mạng, có thể được trình bày dưới dạng bằng chứng kỹ thuật số hợp lệ tại tòa án. Bằng chứng thu thập được có thể tương ứng với dữ liệu thuần túy hoặc, với việc sử dụng rộng rãi các công nghệ Voice_over_IP (VoIP), đặc biệt là qua mạng không dây, có thể bao gồm các cuộc trò chuyện bằng giọng nói.
- Phân tích lưu lượng mạng không dây cũng tương tự như trên mạng có dây, tuy nhiên có thể cần cân nhắc thêm các biện pháp bảo mật không dây .

7 Các giao thức mạng phổ biến:

- Giao thức IP (Internet Protocol – Giao thức Liên mạng): là một giao thức hướng dữ liệu được sử dụng bởi các máy chủ nguồn và đích để truyền dữ liệu trong một liên mạng chuyển mạch gói. Dữ liệu trong một liên mạng IP được gửi theo các khối được gọi là các gói (packet hoặc datagram). Cụ thể, IP không cần thiết lập các đường truyền trước khi một máy chủ gửi các gói tin cho một máy khác mà trước đó nó chưa từng liên lạc với. Giao thức IP cung cấp một dịch vụ gửi dữ liệu không đảm bảo (còn gọi là cố gắng cao nhất), nghĩa là nó hầu như không đảm bảo gì về gói dữ liệu. Gói dữ liệu có thể đến nơi mà không còn nguyên vẹn, nó có thể đến không theo thứ tự (so với các gói khác được gửi giữa hai máy nguồn và đích đó), nó có thể bị trùng lặp hoặc bị mất hoàn toàn. Nếu một phần mềm ứng dụng cần được bảo đảm, nó có thể được cung cấp từ nơi khác, thường từ các giao thức giao vận nằm phía trên IP.

- Giao thức TCP (Transmission Control Protocol – Giao thức điều khiển truyền vận): là một trong các giao thức cốt lõi của bộ giao thức TCP/IP. Sử dụng TCP, các ứng dụng trên các máy chủ được nối mạng có thể tạo các "kết nối" với nhau, mà qua đó chúng có thể trao đổi dữ liệu hoặc các gói tin. Giao thức này đảm bảo chuyển giao dữ liệu tới nơi nhận một cách đáng tin cậy và đúng thứ tự. TCP còn phân biệt giữa dữ liệu của nhiều ứng dụng (chẳng hạn, dịch vụ Web và dịch vụ thư điện tử) đồng thời chạy trên cùng một máy chủ. TCP hỗ trợ nhiều giao thức ứng dụng phổ biến nhất trên Internet và các ứng dụng kết quả, trong đó có WWW, thư điện tử và Secure Shell. Trong bộ giao thức TCP/IP, TCP là tầng trung gian giữa giao thức IP bên dưới và một ứng dụng bên trên. Các ứng dụng thường cần các kết nối đáng tin cậy kiểu đường ống để liên lạc với nhau, trong khi đó, giao thức IP không cung cấp những dòng kiểu đó, mà chỉ cung cấp dịch vụ chuyển gói tin không đáng tin cậy. TCP làm nhiệm vụ của tầng giao vận trong mô hình OSI đơn giản của các mạng máy tính.

- Giao thức UDP (User Datagram Protocol) là một trong những giao thức cốt lõi của giao thức TCP/IP. Dùng UDP, chương trình trên mạng máy tính có thể gửi những dữ liệu ngắn được gọi là datagram tới máy khác. UDP không cung cấp sự tin cậy và thứ tự truyền nhận mà TCP làm; các gói dữ liệu có thể đến không đúng thứ tự hoặc bị mất mà không có thông báo. Tuy nhiên UDP nhanh và hiệu quả hơn đối với các mục tiêu như kích thước nhỏ và yêu cầu khắt khe về thời gian. Do bản chất không trạng thái của nó nên nó hữu dụng đối với việc trả lời các truy vấn nhỏ với số lượng lớn người yêu cầu. Những ứng dụng phổ biến sử

dụng UDP như DNS (Domain Name System), ứng dụng streaming media, Voice over IP, Trivial File Transfer Protocol (TFTP), và game trực tuyến.

- Giao thức FTP (File Transfer Protocol - Giao thức truyền tập tin) thường được dùng để trao đổi tập tin qua mạng lưới truyền thông dùng giao thức TCP/IP (chẳng hạn như Internet - mạng ngoại bộ - hoặc internet - mạng nội bộ). Hoạt động của FTP cần có hai máy tính, một máy chủ và một máy khách). Máy chủ FTP, dùng chạy phần mềm cung cấp dịch vụ FTP, gọi là trình chủ, lắng nghe yêu cầu về dịch vụ của các máy tính khác trên mạng lưới. Máy khách chạy phần mềm FTP dành cho người sử dụng dịch vụ, gọi là trình khách, thì khởi đầu một liên kết với máy chủ. Một khi hai máy đã liên kết với nhau, máy khách có thể xử lý một số thao tác về tập tin, như tải tập tin lên máy chủ, tải tập tin từ máy chủ xuống máy của mình, đổi tên của tập tin, hoặc xóa tập tin ở máy chủ v.v. Vì giao thức FTP là một giao thức chuẩn công khai, cho nên bất cứ một công ty phần mềm nào, hay một lập trình viên nào cũng có thể viết trình chủ FTP hoặc trình khách FTP. Hầu như bất cứ một nền tảng hệ điều hành máy tính nào cũng hỗ trợ giao thức FTP. Điều này cho phép tất cả các máy tính kết nối với một mạng lưới có nền TCP/IP, xử lý tập tin trên một máy tính khác trên cùng một mạng lưới với mình, bất kể máy tính ấy dùng hệ điều hành nào (nếu các máy tính ấy đều cho phép sự truy cập của các máy tính khác, dùng giao thức FTP). Hiện nay trên thị trường có rất nhiều các trình khách và trình chủ FTP, và phần đông các trình ứng dụng này cho phép người dùng được lấy tự do, không mất tiền.

- Giao thức SMTP (Simple Mail Transfer Protocol - giao thức truyền tải thư tín đơn giản) là một chuẩn truyền tải thư điện tử qua mạng Internet. SMTP dùng cổng 25 của giao thức TCP. Để xác định trình chủ SMTP của một tên miền nào đấy (domain name), người ta dùng một mẫu tin MX (Mail eXchange - Trao đổi thư) của DNS (Domain Name System - Hệ thống tên miền). SMTP định nghĩa tất cả những gì đã làm với email. Nó xác định cấu trúc của các địa chỉ, yêu cầu tên miền và bất cứ điều gì liên quan đến email. SMTP cũng xác định các yêu cầu cho Post Office Protocol (POP) và truy cập Internet Message Protocol (IMAP) máy chủ, do đó email được gửi đúng cách.

- Giao thức HTTP (HyperText Transfer Protocol - Giao thức truyền tải siêu văn bản) là một trong năm giao thức chuẩn về mạng Internet, được dùng để liên hệ thông tin giữa Máy cung cấp dịch vụ (Web server) và Máy sử dụng dịch vụ

(Web client) là giao thức Client/Server dùng cho World Wide WebWWW, HTTP là một giao thức ứng dụng của bộ giao thức TCP/IP (các giao thức nền tảng cho Internet).

- Giao thức HTTPS (Hypertext Transfer Protocol Secure) là một sự kết - 16 - hợp giữa giao thức HTTP và giao thức bảo mật SSL hay TLS cho phép trao đổi thông tin một cách bảo mật trên Internet. Giao thức HTTPS thường được dùng trong các giao dịch nhạy cảm cần tính bảo mật cao.

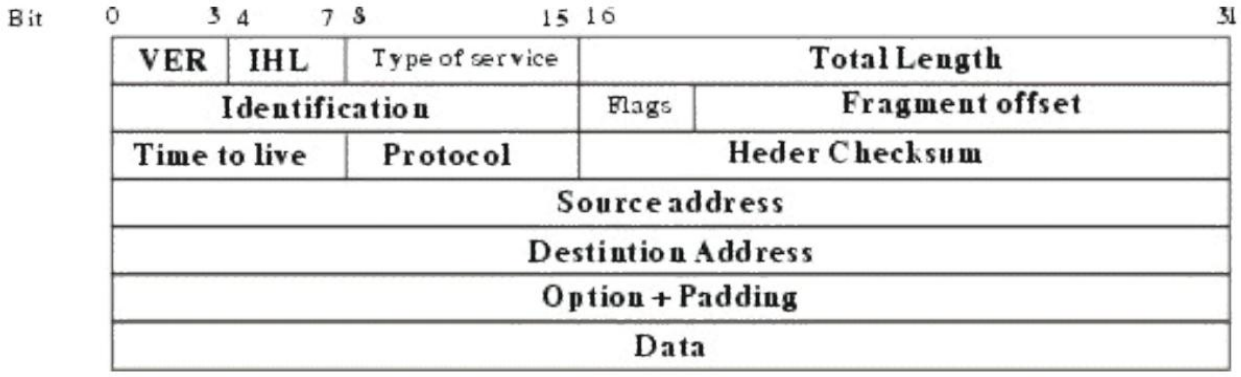
- Giao thức TELNET (TErminaL NETwork) là một giao thức mạng được dùng trên các kết nối với Internet hoặc các kết nối tại mạng máy tính cục bộ LAN. TELNET thường được dùng để cung cấp những phiên giao dịch đăng nhập, giữa các máy trên mạng Internet, dùng dòng lệnh có tính định hướng người dùng. Tên của nó có nguồn gốc từ hai chữ tiếng Anh "telephone network" (mạng điện thoại), vì chương trình phần mềm được thiết kế, tạo cảm giác như một thiết bị đầu cuối được gắn vào một máy tính khác.

- Giao thức SSH (Secure Shell) là một giao thức mạng dùng để thiết lập kết nối mạng một cách bảo mật. SSH hoạt động ở lớp trên trong mô hình phân lớp TCP/IP. Các công cụ SSH (như là OpenSSH, ...) cung cấp cho người dùng cách thức để thiết lập kết nối mạng được mã hoá để tạo một kênh kết nối riêng tư. Hơn nữa tính năng tunneling của các công cụ này cho phép chuyển tải các giao vận theo các giao thức khác.

- Giao thức ICMP (Internetwork Control Message Protocol) cho phép việc thử nghiệm và khắc phục các sự cố của giao thức TCP/IP. ICMP định nghĩa các các thông điệp được dùng để xác định khi nào một hệ thống mạng có thể phân phối các gói tin. Thật ra, ICMP là một thành phần bắt buộc của mọi hiện thực IP. Trong một vài trường hợp, một gateway hoặc một máy đích sẽ cần giao tiếp với máy nguồn để báo cáo lại các lỗi xảy ra trong quá trình xử lý gói tin. Trong trường hợp đó, ICMP sẽ được dùng. ICMP sử dụng IP như thể nó nằm ở một mức cao hơn

8 Các dạng gói tin:

- Gói tin IP:
  - Các gói IP bao gồm dữ liệu từ lớp bên trên đưa xuống và thêm vào một IP Header.

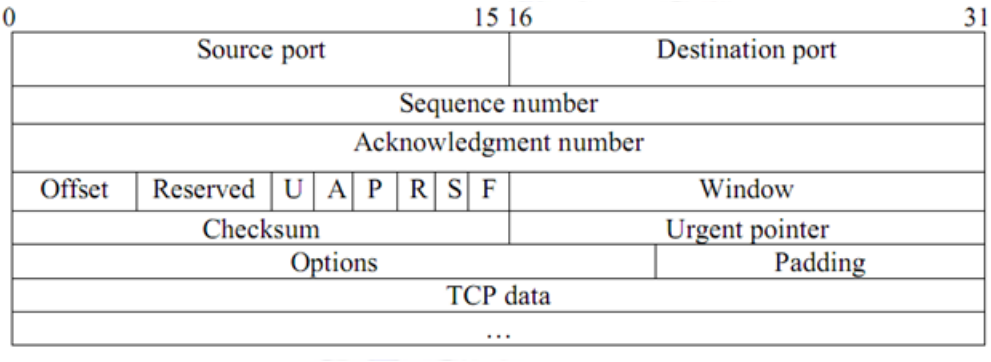


Hình 2. Khuôn dạng của một gói tin IP

- Version chỉ ra phiên bản hiện hành của IP đang được dùng, có 4 bit. Nếu trường này khác với phiên bản IP của thiết bị nhận, thiết bị nhận sẽ từ chối và loại bỏ các gói tin này.
- IP Header Length (HLEN) – Chỉ ra chiều dài của header theo các từ 32 bit. Đây là chiều dài của tất cả các thông tin Header.
- Type Of Services (TOS): Chỉ ra tầm quan trọng được gán bởi một giao thức lớp trên đặc biệt nào đó, có 8 bit.
- Total Length – Chỉ ra chiều dài của toàn bộ gói tính theo byte, bao gồm dữ liệu và header, có 16 bit. Để biết chiều dài của dữ liệu chỉ cần lấy tổng chiều dài này trừ đi HLEN.
- Identification – Chứa một số nguyên định danh hiện hành, có 16 bit. Đây là chỉ số tuần tự.
- Flag – Một field có 3 bit, trong đó có 2 bit có thứ tự thấp điều khiển sự phân mảnh. Một bit cho biết gói có bị phân mảnh hay không và gói kia cho biết gói có phải là mảnh cuối cùng của chuỗi gói bị phân mảnh hay không.
- Fragment Offset – Được dùng để ghép các mảnh Datagram lại với nhau, có 13 bit.
- Time To Live (TTL) – Chỉ ra số bước nhảy (hop) mà một gói có thể đi qua.Con số này sẽ giảm đi một khi một gói tin đi qua một router. Khi bộ đếm đạt tới 0 gói này sẽ bị loại. Đây là giải pháp nhằm ngăn chặn tình trạng lặp vòng vô hạn của gói nào đó.
- Protocol – Chỉ ra giao thức lớp trên, chẳng hạn như TCP hay UDP, tiếp nhận các gói tin khi công đoạn xử lí IP hoàn tất, có 8 bit.
- Header CheckSum – Giúp bảo dảm sự toàn vẹn của IP Header, có 16 bit.
- Source Address – Chỉ ra địa chỉ của node truyền diagram, có 32 bit.

- Destination Address – Chỉ ra địa chỉ IP của Node nhận, có 32 bit.
- Padding – Các số 0 được bổ sung vào trường này để đảm bảo IP Header luôn là bội số của 32 bit.
- Data – Chứa thông tin lớp trên, chiều dài thay đổi đến 64Kb.

❖ Gói tin TCP:



Hình 3. Khuôn dạng một gói tin TCP

- Source port: Số hiệu của cổng tại máy tính gửi.
- Destination port: Số hiệu của cổng tại máy tính nhận.
- Sequence number: Trường này có 2 nhiệm vụ. Nếu cờ SYN bật thì nó là số thứ tự gói ban đầu và byte đầu tiên được gửi có số thứ tự này cộng thêm 1. Nếu không có cờ SYN thì đây là số thứ tự của byte đầu tiên.
- Acknowledgement number: Nếu cờ ACK bật thì giá trị của trường chính là số thứ tự gói tin tiếp theo mà bên nhận cần.
- Data offset: Trường có độ dài 4 bít qui định độ dài của phần header (tính theo đơn vị từ 32 bít). Phần header có độ dài tối thiểu là 5 từ (160 bit) và tối đa là 15 từ (480 bít).
- Reserved: Dành cho tương lai và có giá trị là 0.
  - Flags (hay Control bits): Bao gồm 6 cờ:
  - URG: Cờ cho trường Urgent pointer
  - ACK: Cờ cho trường Acknowledgement
  - PSH: Hàm Push
  - RST: Thiết lập lại đường truyền
  - SYN: Đồng bộ lại số thứ tự
  - FIN: Không gửi thêm số liệu
- Window: Số byte có thể nhận bắt đầu từ giá trị của trường báo nhận (ACK).
- Checksum: 16 bít kiểm tra cho cả phần header và dữ liệu.

9 Kỹ thuật lọc gói:

9.1 WireShark:

- Các cú pháp lọc gói trên WireShark

WIRESHARK DISPLAY FILTERS · PART 1 packetlife.net

Ethernet

eth.addr	eth.len	eth.src
eth.dst	eth.lg	eth.trailer
eth.ig	eth.multicast	eth.type

IEEE 802.1Q

vlan.cfi	vlan.id	vlan.priority
vlan.etype	vlan.len	vlan.trailer

IPv4

ip.addr	ip.fragment.overlap.conflict
ip.checksum	ip.fragment.toolongfragment
ip.checksum_bad	ip.fragments
ip.checksum_good	ip.hdr_len
ip.dsfield	ip.host
ip.dsfield.ce	ip.id
ip.dsfield.dscp	ip.len
ip.dsfield.ect	ip.proto
ip.dst	ip.reassembled_in
ip.dst_host	ip.src
ip.flags	ip.src_host
ip.flags.df	ip.tos
ip.flags.mf	ip.tos.cost
ip.flags.rb	ip.tos.delay
ip.frag_offset	ip.tos.precedence
ip.fragment	ip.tos.reliability
ip.fragment.error	ip.tos.throughput
ip.fragment.multipletails	ip.ttl
ip.fragment.overlap	ip.version

IPv6

ipv6.addr	ipv6.hop_opt
ipv6.class	ipv6.host
ipv6.dst	ipv6.nipv6_hone_address
ipv6.dst_host	ipv6.nipv6_length
ipv6.dst_opt	ipv6.nipv6_type
ipv6.flow	ipv6.nxt
ipv6.fragment	ipv6.opt.padl
ipv6.fragment.error	ipv6.opt.padn
ipv6.fragment.more	ipv6.plen
ipv6.fragment.multipletails	ipv6.reassembled_in
ipv6.fragment.offset	ipv6.routing_hdr
ipv6.fragment.overlap	ipv6.routing_hdr.addr
ipv6.fragment.overlap.conflict	ipv6.routing_hdr.left
ipv6.fragment.toolongfragment	ipv6.routing_hdr.type
ipv6.fragments	ipv6.src
ipv6.fragment.id	ipv6.src_host
ipv6.hlin	ipv6.version

ARP

arp.dst.hw_nac	arp.proto.size
arp.dst.proto_ipv4	arp.proto.type
arp.hw.size	arp.src.hw_nac
arp.hw.type	arp.src.proto_ipv4
arp.opcode	

TCP

tcp.ack	tcp.options.qs
tcp.checksum	tcp.options.sack
tcp.checksum_bad	tcp.options.sack_le
tcp.checksum_good	tcp.options.sack_perm
tcp.continuation_to	tcp.options.sack_re
tcp.dstport	tcp.options.time_stamp
tcp.flags	tcp.options.wscale
tcp.flags.ack	tcp.options.wscale_val
tcp.flags.cwr	tcp.pdu.last_frame
tcp.flags.ecn	tcp.pdu.size
tcp.flags.fin	tcp.pdu.time
tcp.flags.push	tcp.port
tcp.flags.reset	tcp.reassembled_in
tcp.flags.syn	tcp.segment
tcp.flags.urg	tcp.segment.error
tcp.hdr_len	tcp.segment.multipletails
tcp.len	tcp.segment.overlap
tcp.nxtseq	tcp.segment.overlap.conflict
tcp.options	tcp.segment.toolongfragment
tcp.options.cc	tcp.segments
tcp.options.ccecho	tcp.seq
tcp.options.ccnew	tcp.srcport
tcp.options.echo	tcp.time_delta
tcp.options.echo_reply	tcp.time_relative
tcp.options.md5	tcp.urgent_pointer
tcp.options.mss	tcp.window_size
tcp.options.mss_val	

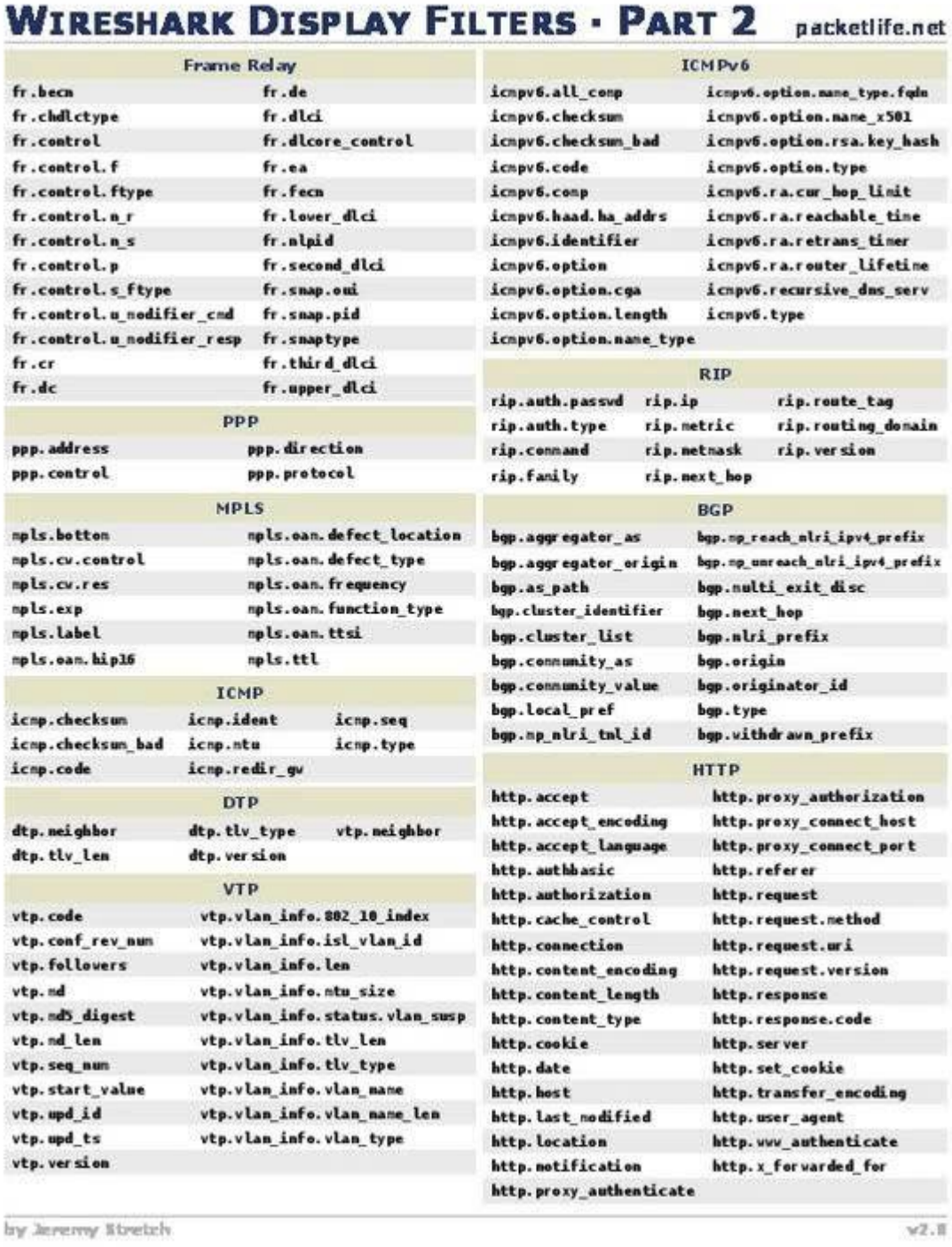
UDP

udp.checksum	udp.dstport	udp.srcport
udp.checksum_bad	udp.length	
udp.checksum_good	udp.port	

Operators	Logic	
eq or ==	and or &&	Logical AND
ne or !=	or or ||	Logical OR
gt or >	xor or ^^	Logical XOR
lt or <	not or !	Logical NOT
ge or >=	[n] [-]	Substring operator
le or <=		

by Jeremy Stretch v2.0

Hình 4. Cú pháp lọc gói WireShark Part1



WIRESHARK DISPLAY FILTERS · PART 2

packetlife.net

Frame Relay

fr.becn	fr.de
fr.chdlctype	fr.dlci
fr.control	fr.dlcore_control
fr.control.f	fr.ea
fr.control.ftype	fr.fecn
fr.control.n_r	fr.lower_dlci
fr.control.n_s	fr.nlpid
fr.control.p	fr.second_dlci
fr.control.s_ftype	fr.snap.oui
fr.control.u_modifier_cmd	fr.snap.pid
fr.control.u_modifier_resp	fr.snaptype
fr.cr	fr.third_dlci
fr.dc	fr.upper_dlci

PPP

ppp.address	ppp.direction
ppp.control	ppp.protocol

MPLS

mpls.bottom	mpls.oam.defect_location
mpls.cw.control	mpls.oam.defect_type
mpls.cw.res	mpls.oam.frequency
mpls.exp	mpls.oam.function_type
mpls.label	mpls.oam.ttsi
mpls.oam.hip16	mpls.ttl

ICMP

icmp.checksum	icmp.ident	icmp.seq
icmp.checksum_bad	icmp.mtu	icmp.type
icmp.code	icmp.redir_gw	

DTP

dtp.neighbor	dtp.tlv_type	vtp.neighbor
dtp.tlv_len	dtp.version	

VTP

vtp.code	vtp.vlan_info.802_10_index
vtp.conf_rev_num	vtp.vlan_info.isl_vlan_id
vtp.followers	vtp.vlan_info.len
vtp.md	vtp.vlan_info.mtu_size
vtp.md5_digest	vtp.vlan_info.status.vlan_susp
vtp.md_len	vtp.vlan_info.tlv_len
vtp.seq_num	vtp.vlan_info.tlv_type
vtp.start_value	vtp.vlan_info.vlan_name
vtp.upd_id	vtp.vlan_info.vlan_name_len
vtp.upd_ts	vtp.vlan_info.vlan_type
vtp.version	

ICMPv6

icmpv6.all_comp	icmpv6.option.name_type.fqdn
icmpv6.checksum	icmpv6.option.name_x501
icmpv6.checksum_bad	icmpv6.option.rsa.key_hash
icmpv6.code	icmpv6.option.type
icmpv6.comp	icmpv6.ra.cur_hop_limit
icmpv6.haad.ha_addrs	icmpv6.ra.reachable_time
icmpv6.identifier	icmpv6.ra.retrans_timer
icmpv6.option	icmpv6.ra.router_lifetime
icmpv6.option.cga	icmpv6.recursive_dns_serv
icmpv6.option.length	icmpv6.type
icmpv6.option.name_type	

RIP

rip.auth.passwd	rip.ip	rip.route_tag
rip.auth.type	rip.metric	rip.routing_domain
rip.command	rip.netmask	rip.version
rip.family	rip.next_hop	

BGP

bgp.aggregator_as	bgp.mp_reach_nlri_ipv4_prefix
bgp.aggregator_origin	bgp.mp_unreach_nlri_ipv4_prefix
bgp.as_path	bgp.multi_exit_disc
bgp.cluster_identifier	bgp.next_hop
bgp.cluster_list	bgp.nlri_prefix
bgp.community_as	bgp.origin
bgp.community_value	bgp.originator_id
bgp.local_pref	bgp.type
bgp.mp_nlri_tnl_id	bgp.withdrawn_prefix

HTTP

http.accept	http.proxy_authorization
http.accept_encoding	http.proxy_connect_host
http.accept_language	http.proxy_connect_port
http.authbasic	http.referer
http.authorization	http.request
http.cache_control	http.request.method
http.connection	http.request.uri
http.content_encoding	http.request.version
http.content_length	http.response
http.content_type	http.response.code
http.cookie	http.server
http.date	http.set_cookie
http.host	http.transfer_encoding
http.last_modified	http.user_agent
http.location	http.www_authenticate
http.notification	http.x_forwarded_for
http.proxy_authenticate	

by Jeremy Stretch v2.0

Hình 5. Cú pháp lọc trên WireShark P2

9.2 Tshark:

- Lệnh tshark là một công cụ dòng lệnh mạnh mẽ dựa trên Wireshark để bắt gói tin và phân tích lưu lượng mạng. Nó thường được sử dụng trong các môi trường không có giao diện đồ họa hoặc khi cần tự động hóa quá trình bắt và phân tích dữ liệu mạng.
- Cú pháp:

tshark [**-i** <capture interface>|-] [**-f** <capture filter>] [**-2**] [**-r** <infile>] [**-w** <outfile>|-] [**options**] [<filter>]

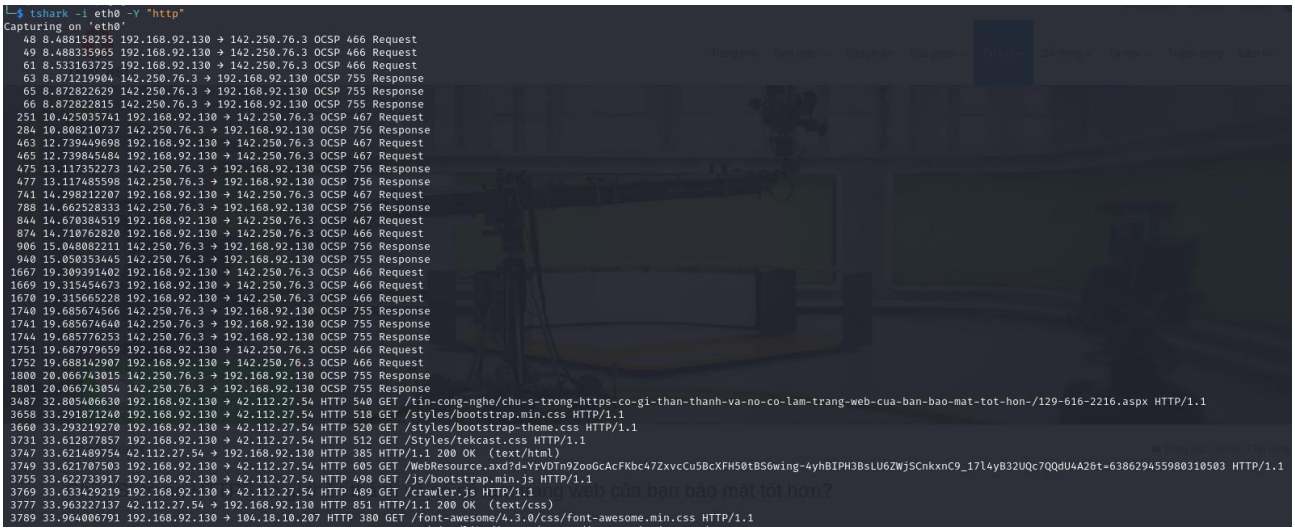
tshark -G [<report type>] [--elastic-mapping-filter <protocols>] [**-C** <profile>]

tshark -h|--help

tshark -v|--version

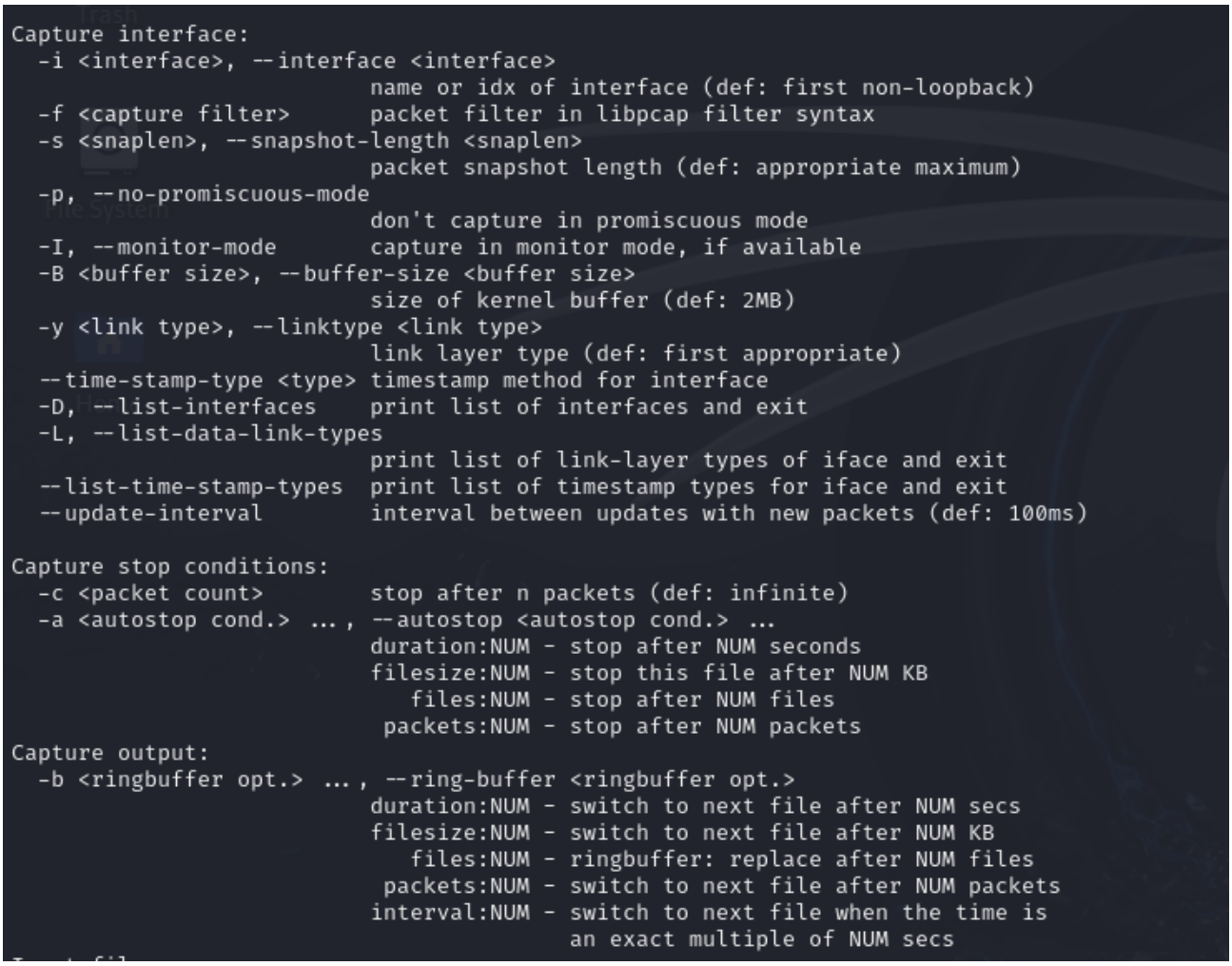
Hình 6. Cú pháp Tshark

- -i <capture interface>: Chỉ định giao diện mạng để bắt gói tin
- -f <capture filter>: Sử dụng bộ lọc để lọc gói tin trong khi bắt.
- -2: Bật chế độ quét hai lần để áp dụng bộ lọc hiển thị chính xác hơn. Tùy chọn này hữu ích khi bạn muốn đảm bảo các bộ lọc phức tạp có thể truy cập vào tất cả các gói tin.
- -r <infile>: Đọc các gói tin từ một tệp đã lưu trước đó (file .pcap hoặc .pcapng).
- -w <outfile>: Ghi các gói tin bắt được vào tệp tin (file .pcap).
- <filter>: Bộ lọc hiển thị (display filter) được áp dụng sau khi gói tin đã được bắt.



Hình 7. Ví dụ về việc bắt gói tin http với tshark

- -h: thể hiện các cú pháp có sẵn được phép sử dụng trong tshark.



Hình 8. Một vài option trong tshark

- -v: thông tin phiên bản

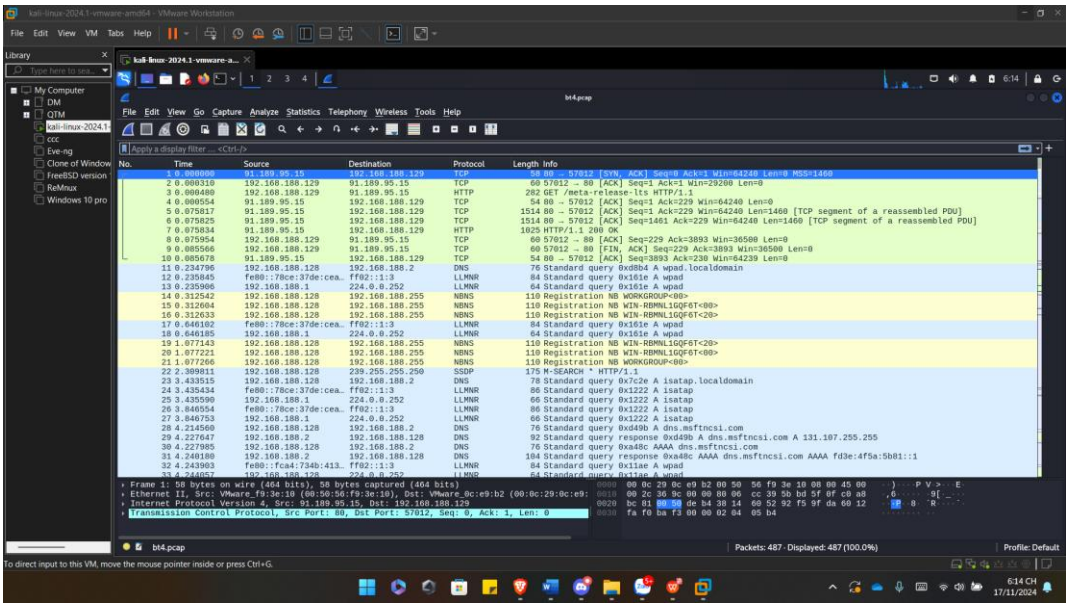
```
└─$ tshark -v
TShark (Wireshark) 4.2.5 (Git v4.2.5 packaged as 4.2.5-1).

Copyright 1998-2024 Gerald Combs <gerald@wireshark.org> and contributors.
Licensed under the terms of the GNU General Public License (version 2 or later).
This is free software; see the file named COPYING in the distribution. There is
NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Compiled (64-bit) using GCC 13.2.0, with GLib 2.80.2, with libpcap, with POSIX
capabilities (Linux), with libnl 3, with zlib 1.3.1, with PCRE2, with Lua 5.2.4,
with GnuTLS 3.8.5 and PKCS #11 support, with Gcrypt 1.10.3, with Kerberos (MIT),
with MaxMind, with nghttp2 1.61.0, with nghttp3 0.8.0, with brotli, with LZ4,
with Zstandard, with Snappy, with libxml2 2.9.14, with libsmi 0.4.8, with binary
plugins.

Running on Linux 6.8.11-amd64, with 11th Gen Intel(R) Core(TM) i5-11400H @
2.70GHz (with SSE4.2), with 1965 MB of physical memory, with GLib 2.80.4, with
libpcap 1.10.4 (with TPACKET_V3), with zlib 1.3.1, with PCRE2 10.42 2022-12-11,
with c-ares 1.31.0, with GnuTLS 3.8.6, with Gcrypt 1.11.0, with nghttp2 1.62.1,
with nghttp3 0.8.0, with brotli 1.1.0, with LZ4 1.9.4, with Zstandard 1.5.6,
with libsmi 0.4.8, with LC_TYPE=en_US.UTF-8, binary plugins supported.
```

Hình 9. Thông tin phiên bản đang sử dụng của tshark

II. Triển khai:

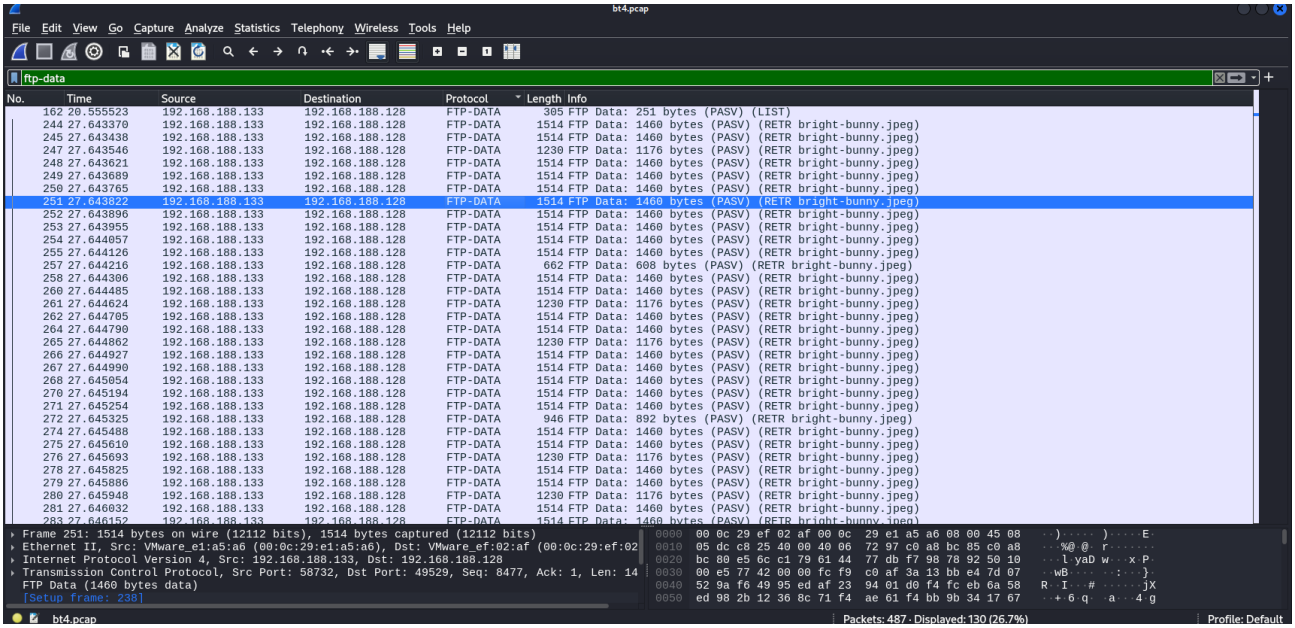
1 Tiến hành chiết xuất file nghi ngờ:

- Ta nhận được 1 file pcap từ người thách thức và yêu cầu tìm cờ.



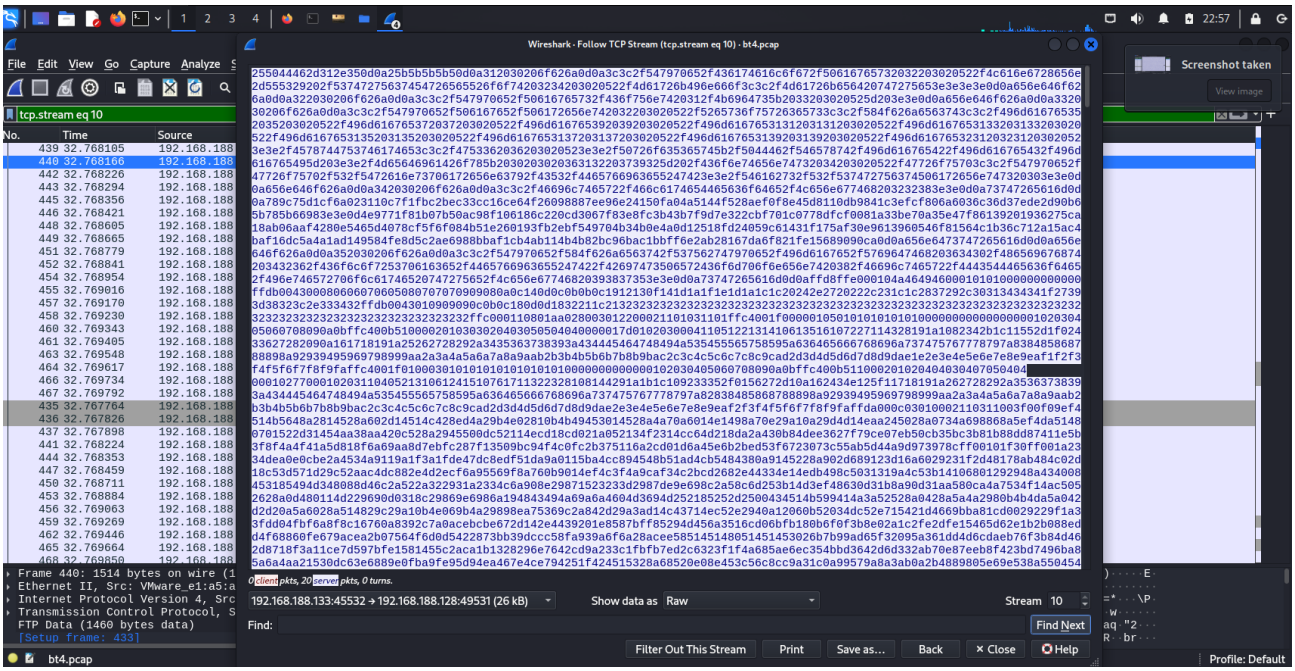
Hình 10. Nhận tìm cờ

- Ta nghi ngờ các đối tượng đã thực hiện truyền file qua giao thức FTP, tiến hành kiểm tra, ngoài FTP ta nhận thấy có FTP-Data.



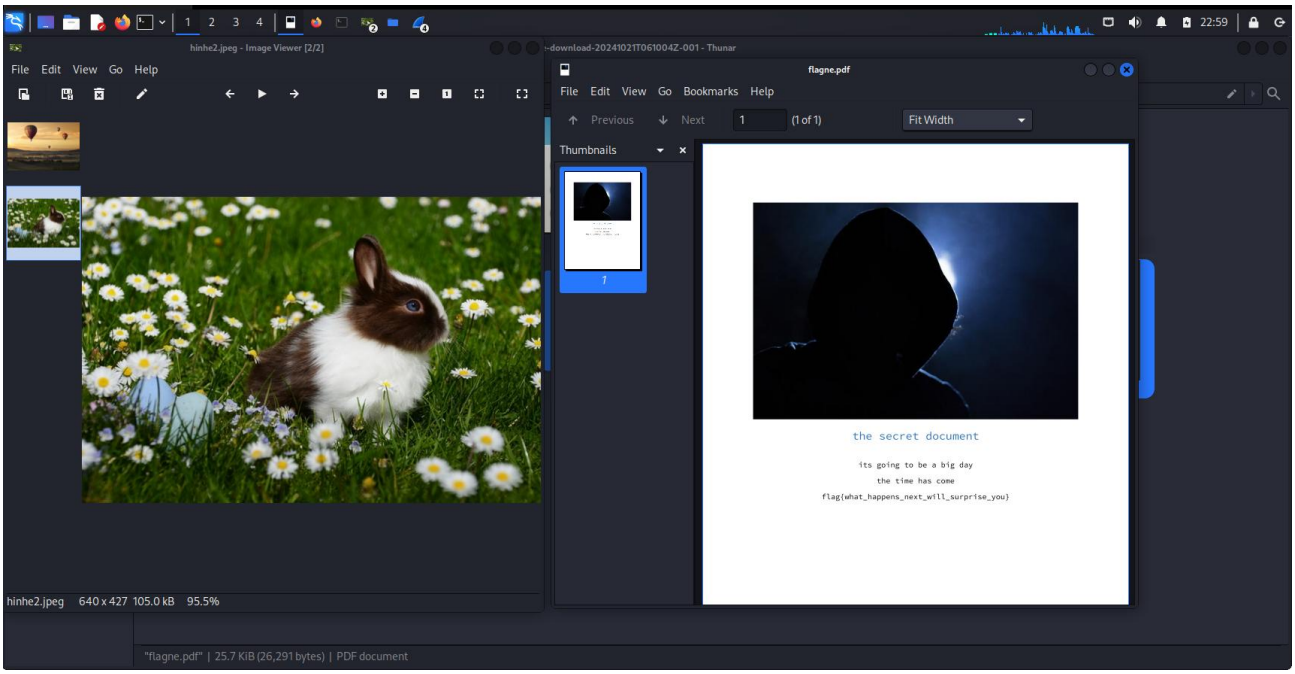
Hình 11. Tiến hành kiểm tra bằng wireshark

- Nhận thấy các file đáng ngờ có dung lượng lớn ta tiến hành triển xuất lần lược các file có đuôi JPEG, PDF.



Hình 12. Tiến hành triết xuất file

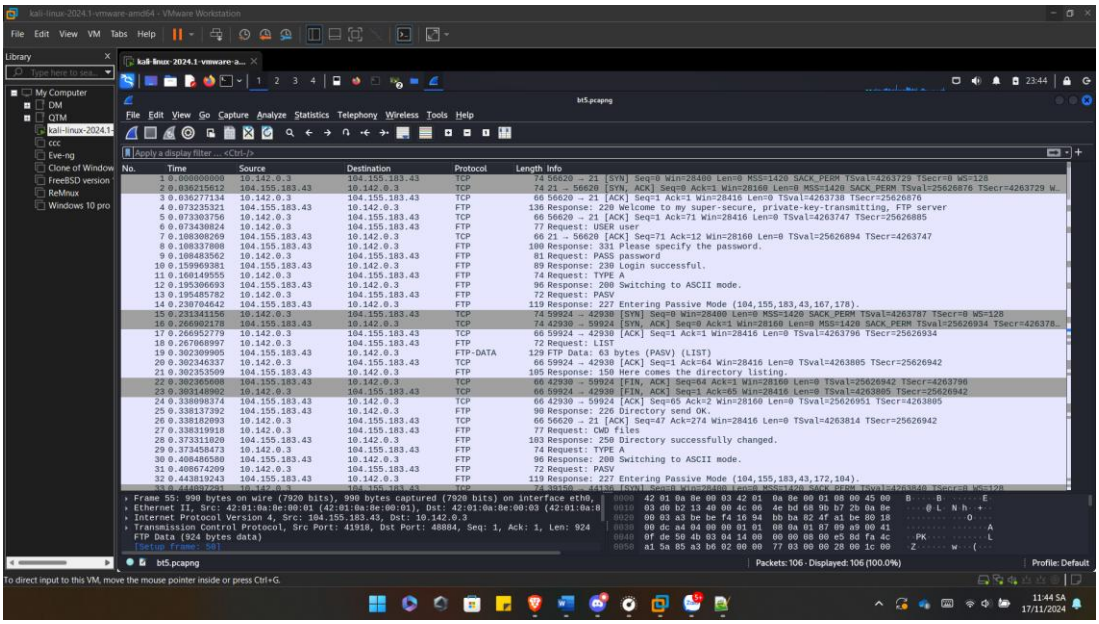
❖ Kết quả ta đã tìm thấy cờ của đối tượng.



Hình 13. Kết quả ta triết xuất thành công file có flag

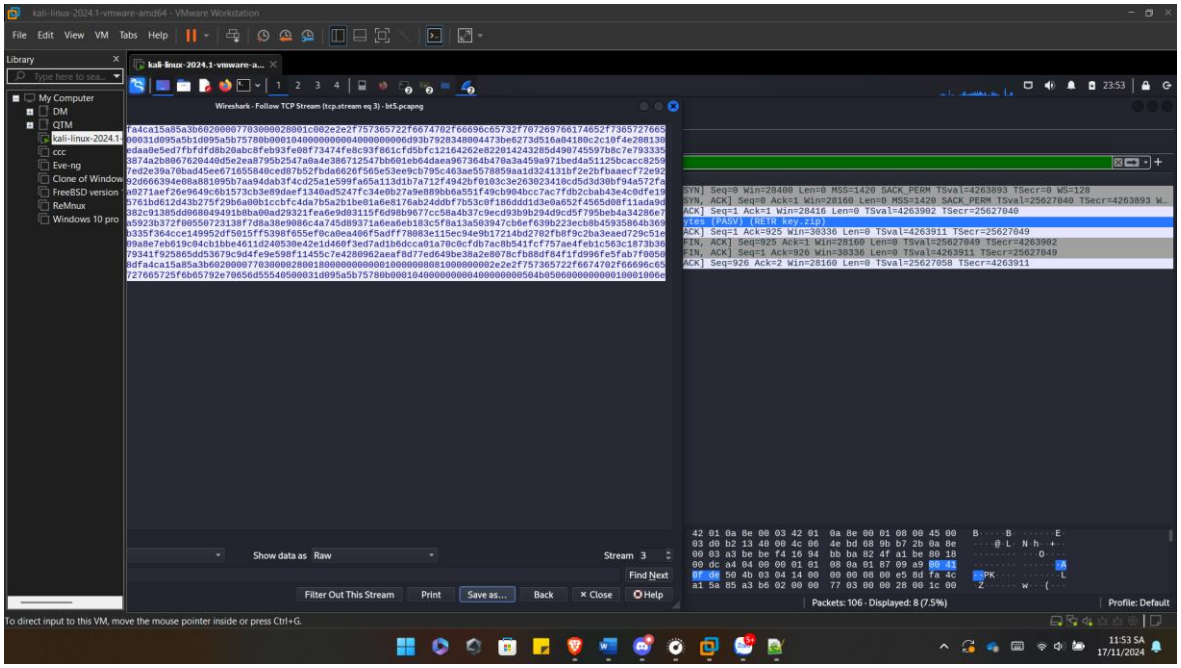
2 Tiến hành mở khóa các file:

❖ Ta nhận được được 1 file pcap và được thách thức tìm ra cờ.



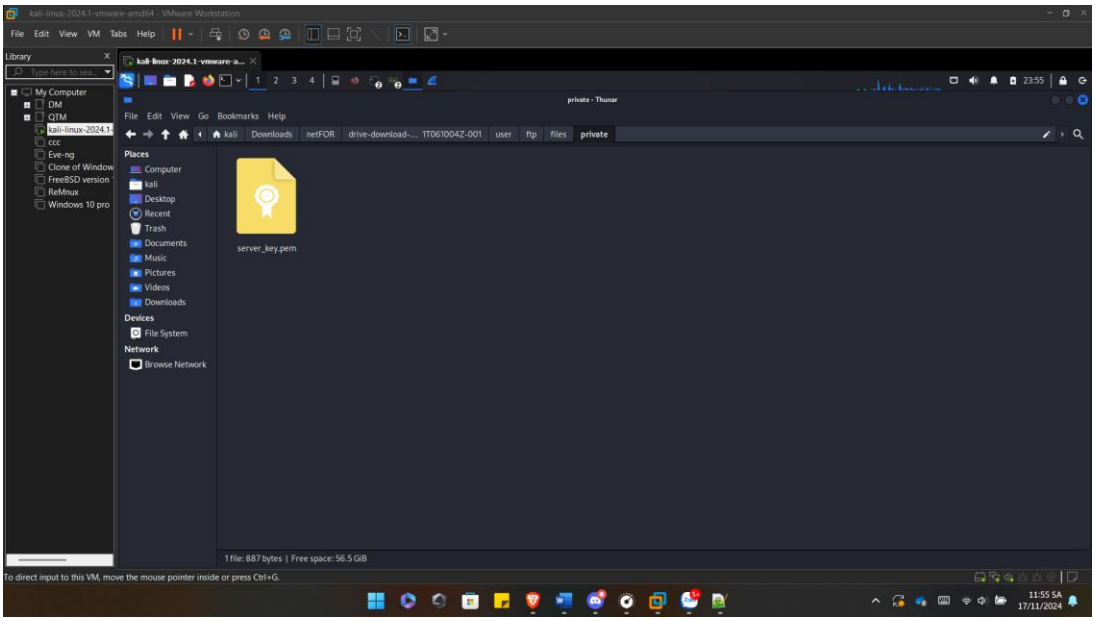
Hình 14. Nhận được file xác định cờ

- Ta tiến hành lọc gói tin FTP-DATA thì nhận thấy được file key.zip đáng nghi. Nhận thấy có thể giúp ích ta tiến hành chiết xuất file.



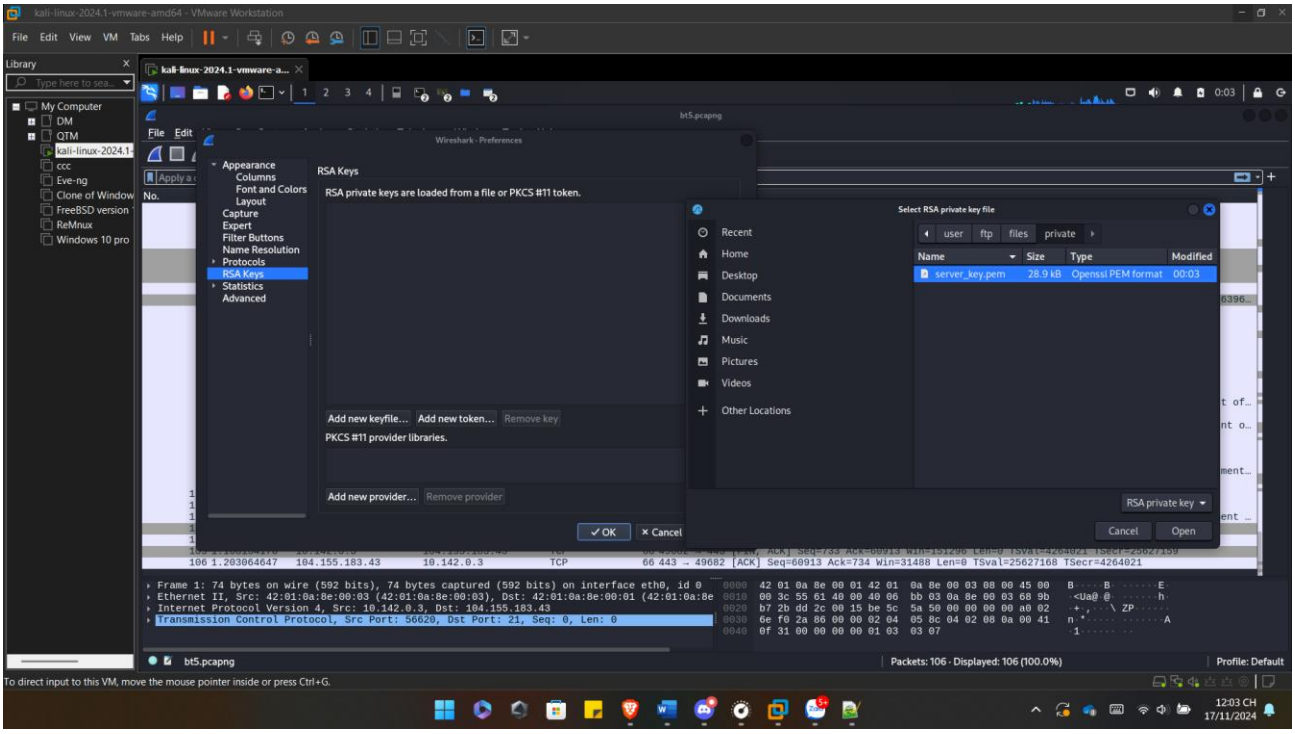
Hình 15. Chiết xuất file key.zip

- Khi mở ra ta thấy file Server_key.pem, tiến hành tìm hiểu thì .pem là một định dạng tệp được sử dụng rộng rãi để lưu trữ và truyền tải các chứng chỉ số, khóa mật mã, và các dữ liệu bảo mật khác trong các hệ thống mã hóa.



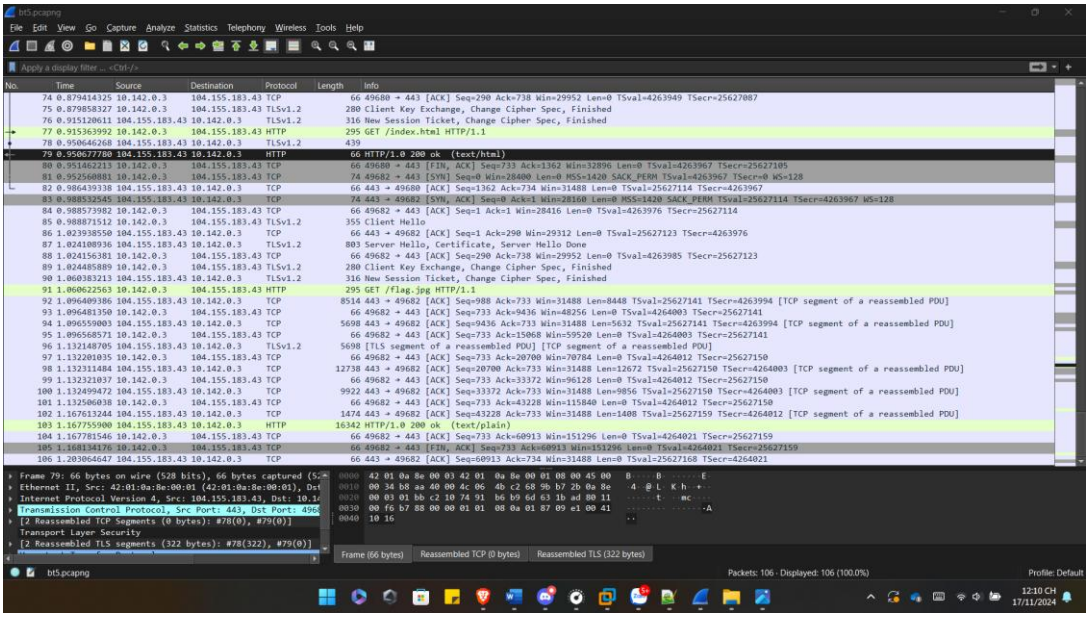
Hình 16. Tìm được server_key.pem

❖ Tiến hành add Server_key.pem vào để giải mã các gói tin được mã hóa theo TLS.



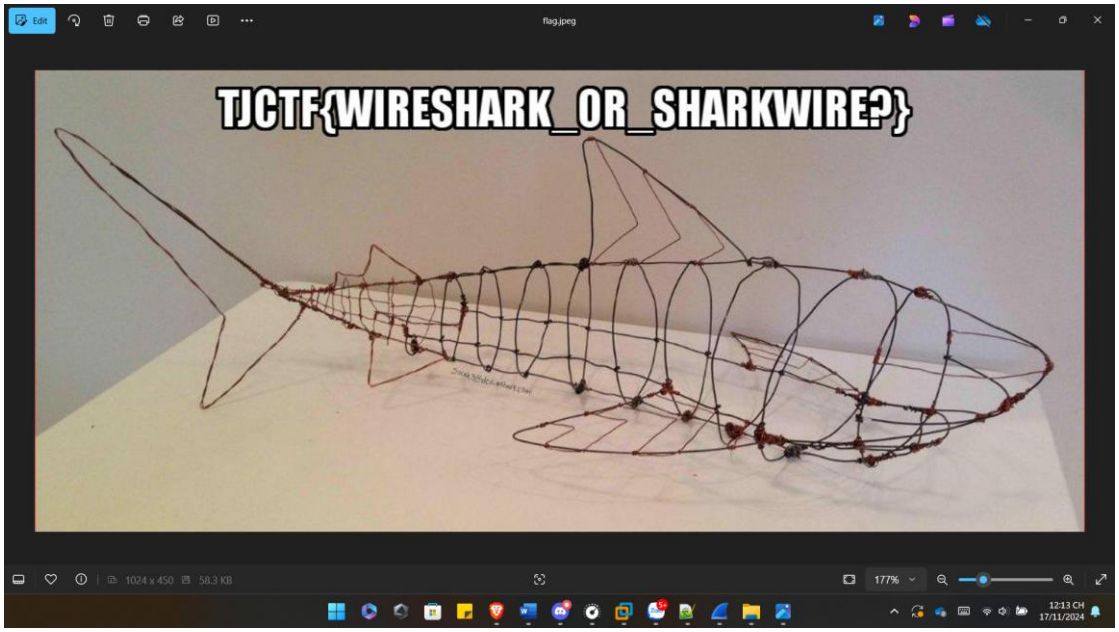
Hình 17. Thêm server_key.pem để giải mã

❖ Khi add key vào ta thấy một số gói TLSv1.2 đã chuyển đổi thành HTTP. Ta nhận thấy file flag.jpeg. Tiến hành chiết xuất file.



Hình 18. Một vài gói tin đã được giải mã

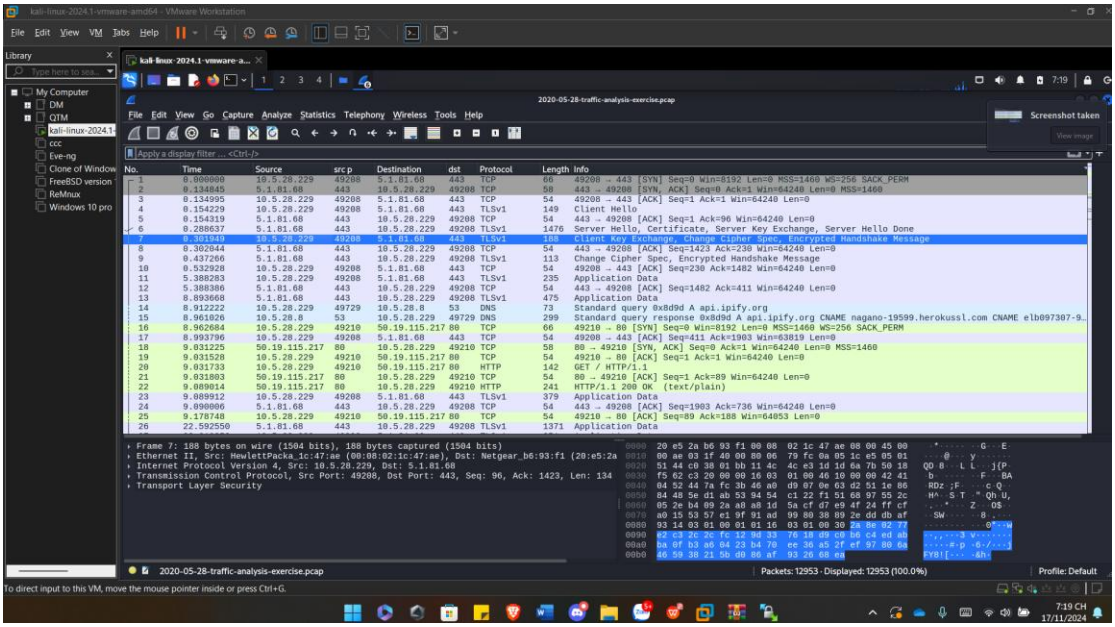
❖ Kết quả



Hình 19. Flag

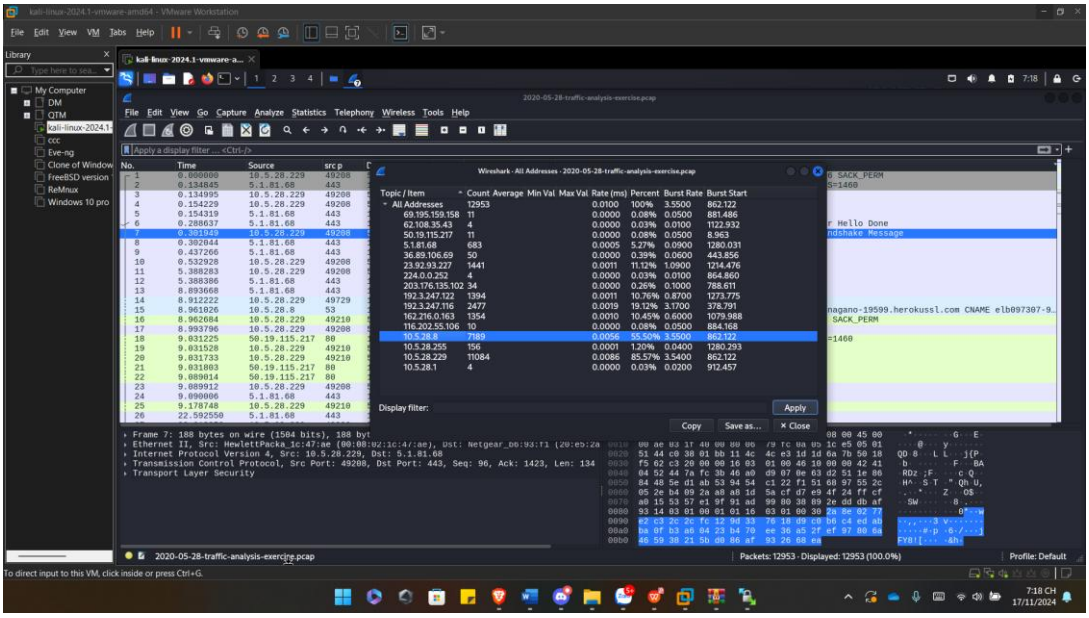
3 Tiến hành nhận dạng tấn công:

❖ Ta nhận được traffic.pcap, yêu cầu điều tra vì nghi ngờ đã bị tấn công chưa rõ phương thức.



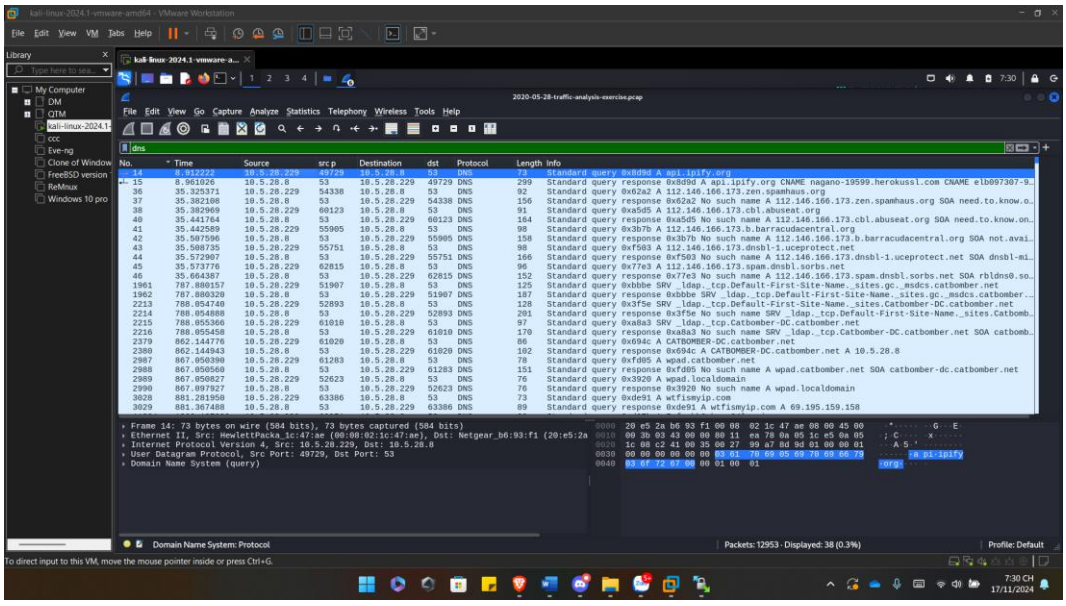
Hình 20. Nhận file yêu cầu điều tra cuộc tấn công

❖ Khi ta xuất các list các IPv4 thì nhận thấy có 2 ip có lượng gói tin nhiều nhất là 10.5.28.229 và 10.5.28.8



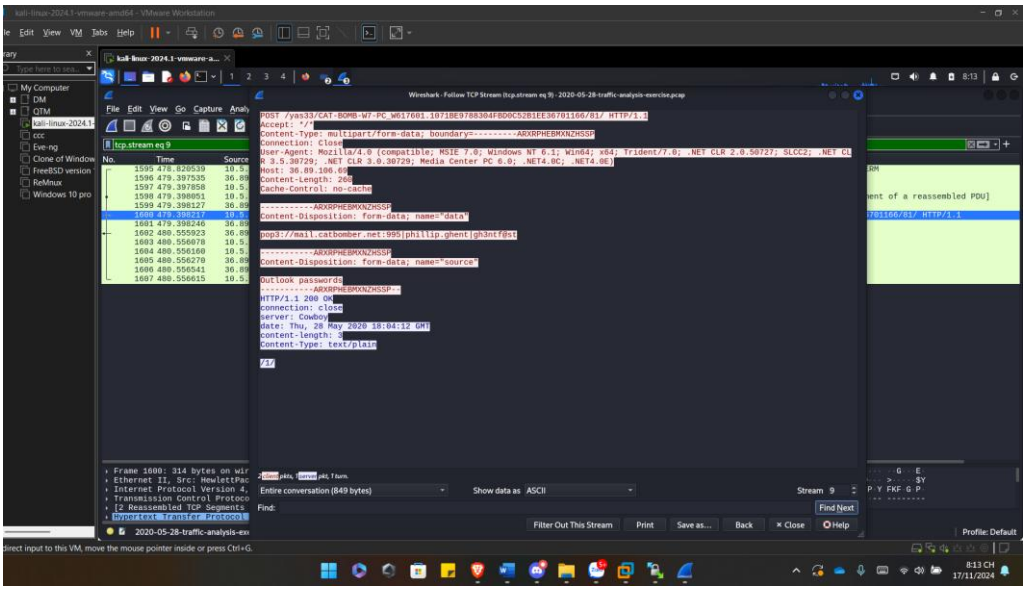
Hình 21. List các ipv4

❖ Tiến hành ip.addr == 10.5.28.8 thì phát hiện đây là máy chủ dns và ip 10.5.28.299 là máy nạn nhân, nhưng điều khả nghi ở đây là nạn nhân bị tra cứa mỗi một lần phân giải tên miền.



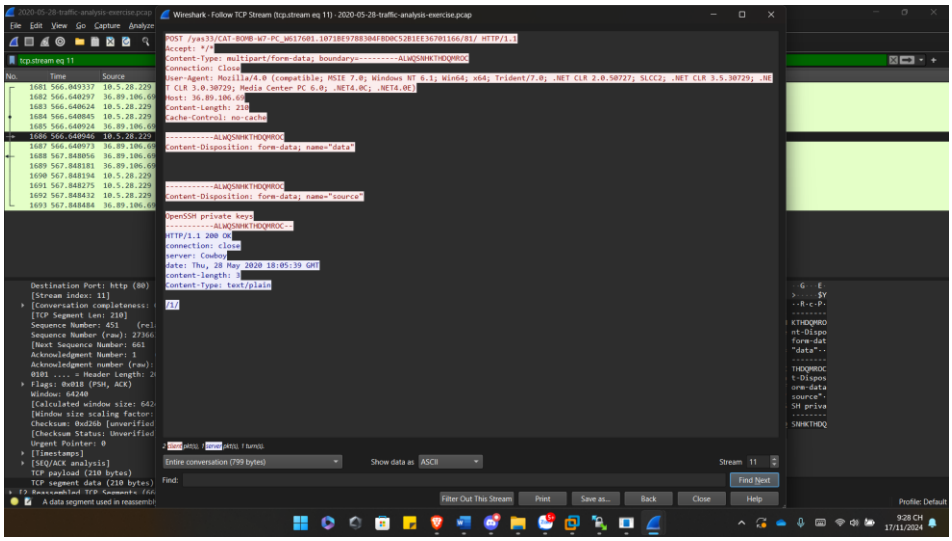
Hình 22. Nhận thấy máy nạn nhân bị tra ip

- Tiến hành kiểm tra xem nạn nhân đã duyệt web nào thì thấy nạn nhân thực hiện POST lên 1 địa chỉ lạ. Kiểm tra chi tiết thì nhận thấy nạn nhân truy cập một host khá lạ 36.89.106.69 và đồng thời xác nhận được server là Cowboy có vẻ khả nghi. Mac address là 20:e5:2a:b6:93:f1



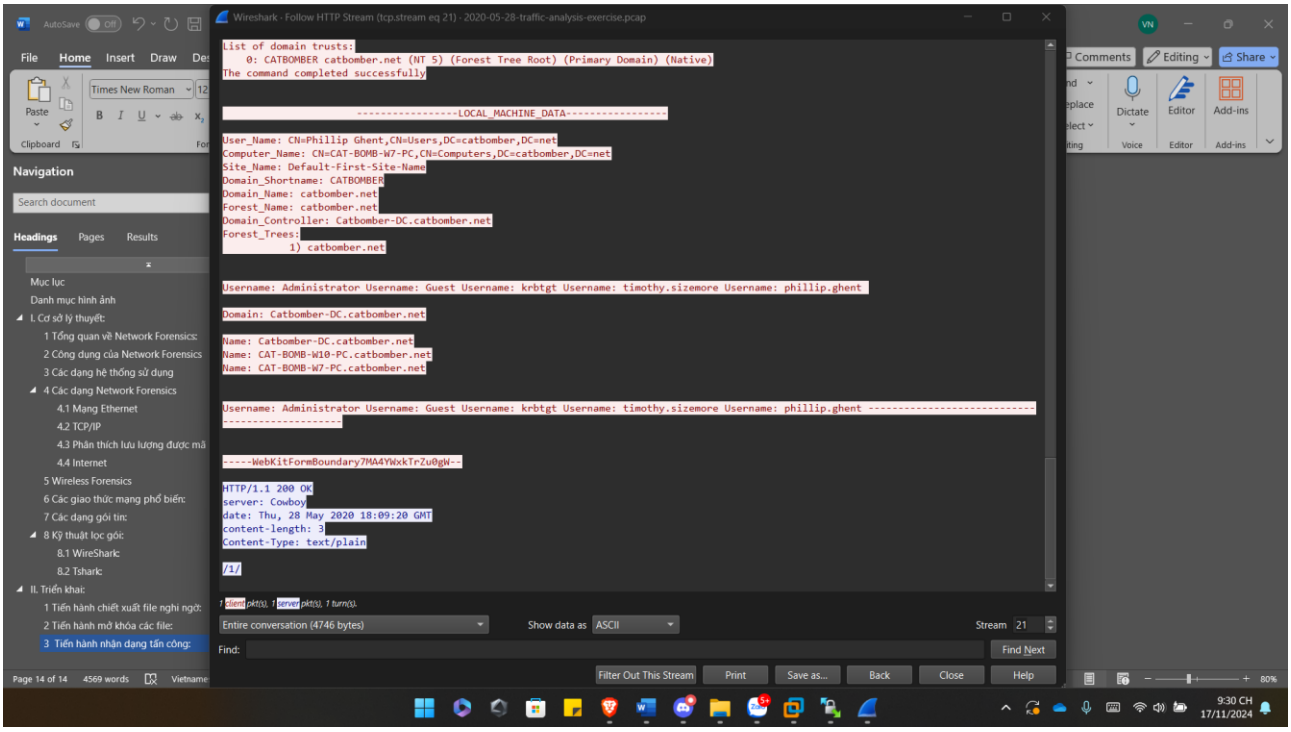
Hình 23. Xác định server khả nghi

- Tiếp tục kiểm tra ta nhận thấy máy nạn nhân bị rò ri tài khoản mail, Outlook password, OpenVPN and configs, OpenSSH private keys.



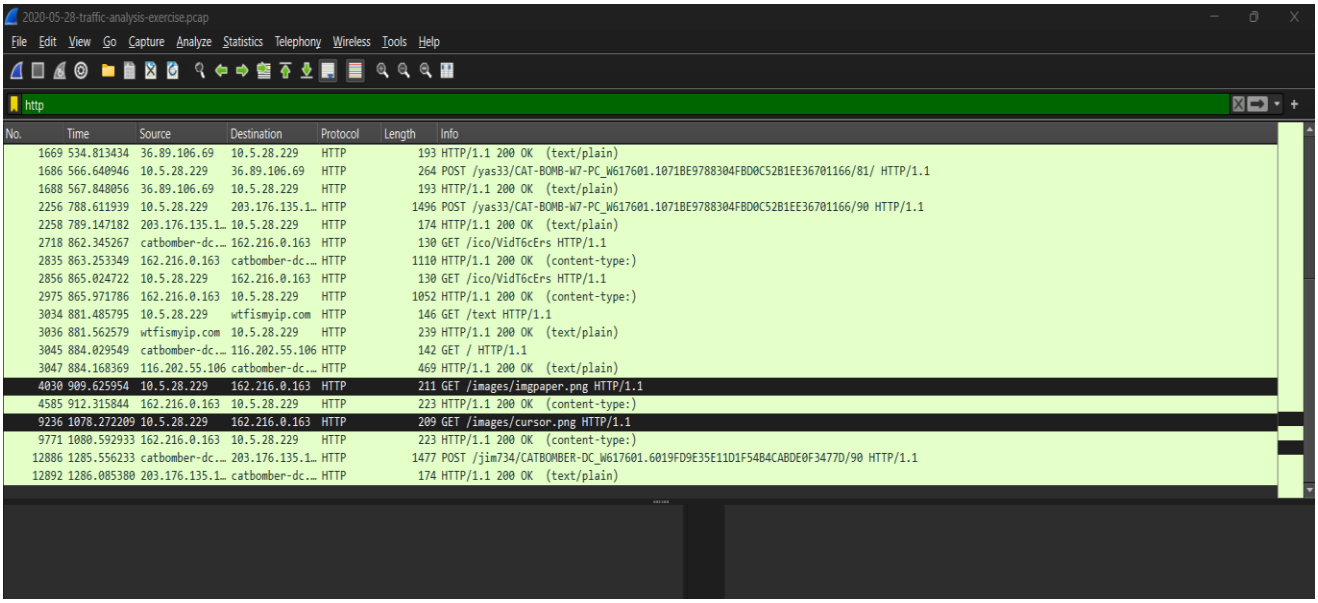
Hình 24. Rò rỉ username và password

- Tiếp tục kiểm tra ta nhận thấy các file thực thi được khởi chạy. Các file thực thi khởi chạy hiển thị thông tin hệ thống của nạn nhân.



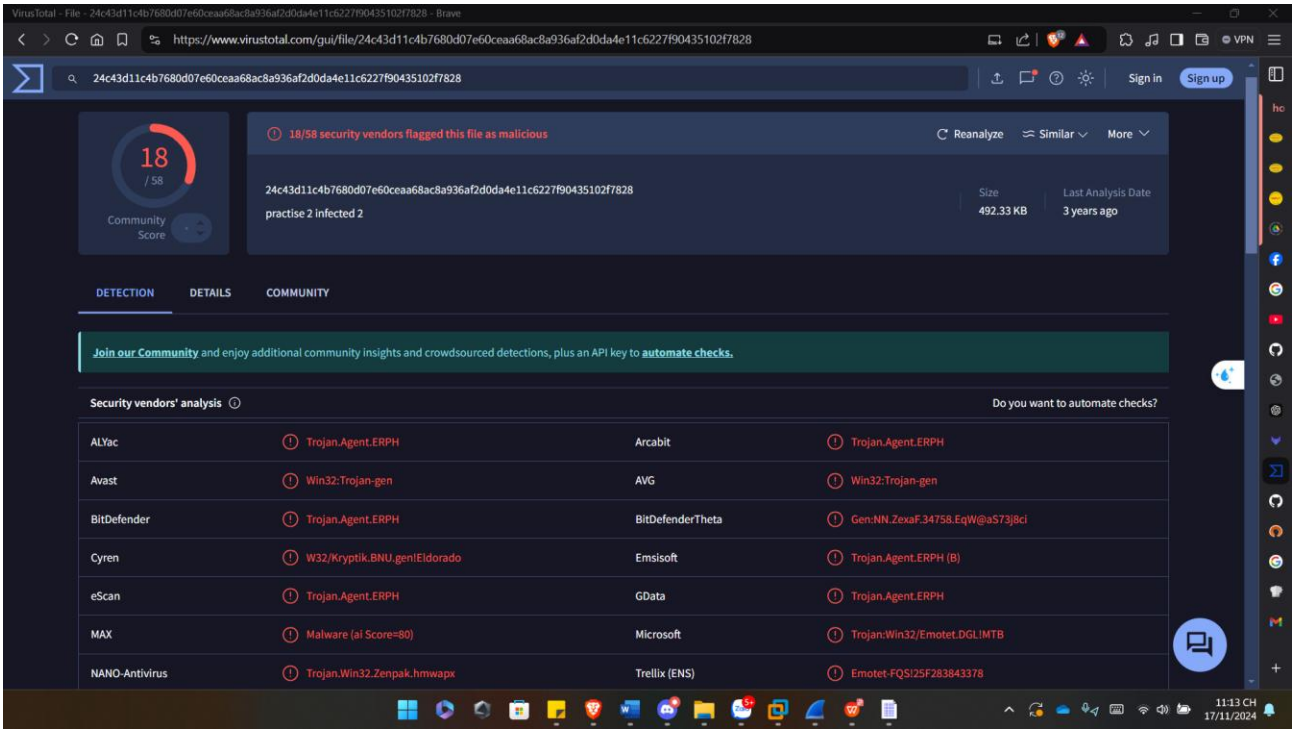
Hình 25. Rò rỉ thông tin hệ thống

- Ta nhận thấy vài file đáng nghi của nạn nhân như imagepaper.png, cursor.png vì hex có MZ, đây có thể là file thực thi của windows. Tiến hành chiết xuất file để kiểm tra.



Hình 26. Phát hiện các file đáng nghi

- Tiến hành đổi định dạng file thành exe và check bằng tool online VirusTotal. Xác nhận được có malware lấy thông tin người dùng.



Hình 27. Xác nhận có malware