

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 TẦN CÔNG  
ĐỀ TÀI : NETWORK FORENSICS**

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

**Thành Viên :**

**Phạm Hoàng Gia Bảo MSSV:22DH110298**

**Lê Thành Đạt MSSV:22DH110711**

*Tp. Hồ chí minh, Ngày 18 tháng 11 Năm 2024*

# Network Forensics

## Nhận xét của giảng viên

# **Network Forensics**

## **Lời Cảm Ơn**

Để hoàn thành tốt bài báo cáo này, chúng em xin gửi lời cảm ơn chân thành đến giảng viên, Ts Phạm Đình Thắng, người đã hỗ trợ cho chúng em trong quá trình làm bài. Cảm ơn thầy đã đưa ra những góp ý, nhận xét để chúng em có thể hoàn thành tốt bài báo cáo này và nộp đúng hạn thời gian bài báo cáo đề ra.

Trong thời gian học tập và làm báo cáo dưới sự hướng dẫn của thầy, chúng em đã có thêm những kiến thức bổ ích, những kinh nghiệm để có thể hiểu về các kỹ thuật tấn công nhờ đó mà có thể điều tra ra được những thông tin và có được những kỹ năng để có thể làm được bài báo cáo về chủ đề “Network Forensics” trong môn điều tra tấn công.

Mặc dù đã nỗ lực trong việc hoàn thiện báo cáo, nhưng do thời gian có hạn, bước đầu đi vào tìm hiểu những kỹ thuật và thực hành trong chủ đề “Network Forensics”, với lượng kiến thức nông cạn và hạn chế, nhiều bỡ ngỡ khi làm một thứ mới mẻ mình chưa bao giờ được làm, nên bài báo cáo về đề tài: “Network Forensics” của chúng em chắc chắn vẫn còn rất nhiều sai sót nên chúng em rất mong rằng mình có thể nhận được những lời góp ý quý báu của các thầy cô để chúng em có thể hoàn thiện kiến thức của bản thân mình hơn trong việc điều tra tấn công trong các đồ án kì sau.

Một lần nữa chúng em xin chân thành cảm ơn thầy và luôn mong nhận sự đóng góp của quý thầy cô.

Cuối lời, chúng em xin kính chúc quý thầy cô Khoa Công Nghệ Thông Tin luôn dồi dào sức khoẻ và thành công hơn nữa trong sự nghiệp trồng người của mình. Chúng em trân trọng cảm ơn!

## **MỤC LỤC**

### **Mục Lục**

<b>Nhận xét của giảng viên .....</b>	2
<b>Lời Cảm Ơn.....</b>	3
<b>MỤC LỤC .....</b>	4
<b>DANH MỤC HÌNH ẢNH .....</b>	6
<b>1. Network Forensics là gì? .....</b>	8
<b>1.1. Định nghĩa .....</b>	8
<b>1.2. Mục tiêu .....</b>	8
<b>1.3. Ứng dụng .....</b>	8
<b>2. IP.....</b>	8
<b>2.1. Địa chỉ ip .....</b>	8
<b>2.2. Protocol .....</b>	9
<b>2.3. Header ip .....</b>	9
<b>2.4. Fragmentation.....</b>	9
<b>3. TCP.....</b>	9
<b>3.1. Cổng nguồn (Source Port) và cổng đích (Destination Port) .....</b>	9
<b>3.2. Số thứ tự (Sequence Number).....</b>	9
<b>3.3. Số xác nhận (Acknowledgment Number) .....</b>	9
<b>3.4. Flags (Cờ).....</b>	9
<b>3.5. Kết nối ba bước (TCP Three-Way Handshake) .....</b>	10
<b>4. UDP.....</b>	10
<b>4.1. Định nghĩa .....</b>	10
<b>4.2. Cấu trúc gói .....</b>	10
<b>4.3. Đặc điểm chính.....</b>	10
<b>5. Ethernet header .....</b>	10
<b>5.1. Định nghĩa .....</b>	11
<b>5.2. Cấu trúc .....</b>	11
<b>6. Các kỹ thuật lọc gói tin .....</b>	11

# Network Forensics

6.1.	Lọc theo giao thức.....	11
6.2.	Lọc theo địa chỉ ip .....	11
6.3.	Lọc theo địa chỉ MAC.....	12
6.4.	Lọc theo cổng (port).....	12
6.5.	Lọc theo Flag trong TCP .....	12
6.6.	Lọc theo giao thức tầng cao .....	12
6.7.	Lọc theo kích thước gói tin (Packet Length) .....	13
6.8.	Lọc gói tin lỗi.....	13
7.	Các kĩ thuật dùng wireshark.....	13
1/	Profile check .....	13
2/	Wireshark Filter .....	15
3/	Filtering for Web Traffic .....	17
4/	Creating Filter Buttons .....	20
5/	Filtering for Non-Web Traffic .....	22
6/	Filtering for FTP Traffic .....	23
7/	Filtering for Email (Spambot) Traffic .....	25
1/	HTTPS Web Traffic .....	28
2/	Encryption Key Log File .....	30
3/	HTTPS Traffic Without the Key Log File.....	31
4/	Loading the Key Log File.....	31
5/	HTTPS Traffic With the Key Log File .....	34
8.	Demo .....	38
	Bảng Phân Công .....	49
	Tài Liệu Tham Khảo .....	50

## **DANH MỤC HÌNH ẢNH**

Hình 1. Step 1 - Profile check .....	14
Hình 2. Step 2 - profile check .....	15
Hình 3. Step 1 - Wireshark Filter.....	16
Hình 4. Step 2 - Wireshark Filter.....	16
Hình 5. Các toán tử trong Wireshark Filter .....	16
Hình 6. Step 1 - Filtering for Web Traffic .....	18
Hình 7. Step 2 - Filtering for Web Traffic .....	19
Hình 8. Step 3 - Filtering for Web Traffic .....	20
Hình 9. Step 1 - Creating Filter Buttons .....	20
Hình 10. Step 2 - Creating Filter Buttons .....	20
Hình 11. Trick - Creating Filter Buttons.....	21
Hình 12. Step 3 - Creating Filter Buttons .....	22
Hình 13. Step 4 - Creating Filter Buttons .....	22
Hình 14. Step 5 - Creating Filter Buttons .....	22
Hình 15. Filtering for Non-Web Traffic .....	23
Hình 16. Filtering for FTP Traffic .....	24
Hình 17. Trick for FTP .....	24
Hình 18. Kết quả nhập biểu thức .....	25
Hình 19. Lọc các smtp và dns .....	26
Hình 20. Lọc các command request.....	27
Hình 21. Biểu thức để lọc spam bot.....	28
Hình 22. Step 1 - HTTPS Web Traffic .....	29
Hình 23. Step 2 - HTTPS Web Traffic .....	30
Hình 24. Các mã hoá kí tự .....	31
Hình 25. HTTPS Traffic Without the Key Log File.....	31
Hình 26. Step 1 - Loading the Key Log File .....	32

## **Network Forensics**

Hình 27. Step 2 - Loading the Key Log File .....	33
Hình 28. Step 3 - Loading the Key Log File .....	34
Hình 29. kết quả bộ lọc .....	34
Hình 30. Xem theo các luồng .....	35
Hình 31. Trích xuất file.....	37
Hình 32. Các file có thể trích xuất .....	38
Hình 33. Lọc giao thức web.....	38
Hình 34. Tìm thấy mật khẩu từ web .....	39
Hình 35. nghi ngờ giao thức FTP có thể có tk và mk.....	39
Hình 36. Tìm theo luồng sẽ thấy tk và mk.....	40
Hình 37. Kết quả tài khoản mật khẩu của phương thức telnet .....	41
Hình 38. Nghi ngờ các phương thức bị mã hoá.....	42
Hình 39. Tìm thấy key giải mã hoá từ FTP .....	42
Hình 40. Khôi phục lại key .....	42
Hình 41. Thêm key giải mã hoá vào wireshark .....	43
Hình 42. Decode giao thức bị mã hoá.....	43
Hình 43. Khôi phục lại được dữ liệu.....	44
Hình 44. Lọc gói .....	45
Hình 45. Khôi phục file .....	47
Hình 46. Ta có một file doc và một file exe .....	47
Hình 47. ta checksum hai file theo sha-256.....	47
Hình 48. file doc.....	48
Hình 49. file exe.....	48

## 1. Network Forensics là gì?

### 1.1. Định nghĩa

**Network Forensics** là một lĩnh vực trong **An ninh mạng** tập trung vào việc **thu thập, phân tích và điều tra** dữ liệu lưu lượng mạng để xác định, điều tra các cuộc tấn công mạng, hoặc giải quyết các vấn đề liên quan đến bảo mật. Nó được xem như một nhánh của **Pháp y kỹ thuật số (Digital Forensics)** nhưng tập trung chủ yếu vào các hoạt động xảy ra trên mạng máy tính.

### 1.2. Mục tiêu

- Phát hiện sự cố bảo mật: Ví dụ như các cuộc tấn công DDoS, xâm nhập mạng, lỗ hổng bảo mật, hoặc mã độc.
- Thu thập bằng chứng: Giúp xác định nguồn gốc của cuộc tấn công hoặc xác minh hành vi không phù hợp.
- Tái dựng sự cố: Phân tích và mô phỏng lại các hoạt động mạng để hiểu chính xác điều gì đã xảy ra.
- Hỗ trợ điều tra pháp lý: Sử dụng dữ liệu mạng làm bằng chứng trong các vụ kiện hoặc điều tra tội phạm mạng,

### 1.3. Ứng dụng

- Phòng chống và phát hiện tấn công mạng.
- Điều tra hành vi không phù hợp trong mạng nội bộ (Insider Threats).
- Xác định và giảm thiểu rủi ro bảo mật.
- Phân tích mã độc hoặc ransomware qua mạng.
- Hỗ trợ pháp lý trong các vụ kiện hoặc điều tra tội phạm mạng.

## 2. IP

### 2.1. Địa chỉ IP

- Source IP: Địa chỉ IP của nguồn gửi gói tin.
- Destination IP: Địa chỉ IP của đích nhận gói tin.

- IPv4 hoặc IPv6: Wireshark hỗ trợ cả hai phiên bản giao thức IP.

### 2.2. Protocol

- IP là một giao thức tầng mạng (Network Layer Protocol).
- Trong Wireshark, các giao thức sử dụng IP như TCP, UDP, ICMP cũng được hiển thị.

### 2.3. Header ip

- Version: Phiên bản giao thức (IPv4 hoặc IPv6).
- Header Length: Độ dài phần header của gói tin IP.
- TTL (Time to Live): Thời gian tồn tại của gói tin, giảm dần mỗi khi đi qua một thiết bị mạng.
- Protocol: Giao thức tầng trên (ví dụ: TCP, UDP, ICMP).
- Checksum: Giá trị dùng để kiểm tra tính toàn vẹn của header IP.

### 2.4. Fragmentation

- Nếu gói tin IP bị chia nhỏ (fragmentation), Wireshark sẽ hiển thị các thông tin liên quan, chẳng hạn như More Fragments hoặc Fragment Offset.

## 3. TCP

### 3.1. Cổng nguồn (Source Port) và cổng đích (Destination Port)

- Cổng nguồn và cổng đích xác định các ứng dụng đang giao tiếp với nhau trên hai thiết bị.
- Ví dụ: TCP Port 80 thường được sử dụng cho HTTP, Port 443 cho HTTPS.

### 3.2. Số thứ tự (Sequence Number)

- Mỗi gói tin TCP có một số thứ tự để đảm bảo rằng dữ liệu đến đúng thứ tự.

### 3.3. Số xác nhận (Acknowledgment Number)

- Gửi lại cho bên gửi để xác nhận rằng gói tin đã được nhận thành công.

### 3.4. Flags (Cờ)

Các cờ điều khiển trong gói TCP xác định trạng thái của kết nối:

- SYN: Khởi tạo kết nối.
- ACK: Xác nhận dữ liệu đã nhận.
- FIN: Kết thúc kết nối.
- RST: Đặt lại kết nối khi xảy ra lỗi.
- PSH: Yêu cầu chuyển dữ liệu ngay lập tức.
- URG: Xác định dữ liệu khẩn cấp.

### 3.5. Kết nối ba bước (TCP Three-Way Handshake)

- SYN → SYN-ACK → ACK: Quá trình thiết lập kết nối TCP đáng tin cậy.

## 4. UDP

### 4.1. Định nghĩa

UDP (User Datagram Protocol) trong Wireshark là một giao thức tầng giao vận (Transport Layer) được sử dụng để truyền dữ liệu một cách nhanh chóng nhưng không đảm bảo tính đáng tin cậy như TCP. UDP không có cơ chế kiểm tra thứ tự hoặc xác nhận gói tin, điều này khiến nó phù hợp với các ứng dụng yêu cầu tốc độ cao hoặc không cần dữ liệu hoàn chỉnh, như truyền video, âm thanh (VoIP), hoặc DNS.

### 4.2. Cấu trúc gói

- Source Port và Destination Port: Xác định ứng dụng gửi và nhận.
- Length: Độ dài của toàn bộ gói UDP (header + payload).
- Checksum: Kiểm tra tính toàn vẹn của gói dữ liệu (có thể không sử dụng).

### 4.3. Đặc điểm chính

- Không kết nối (connectionless): Không cần thiết lập trước khi truyền dữ liệu.
- Không có xác nhận hoặc cơ chế sửa lỗi.

## 5. Ethernet header

### 5.1. Định nghĩa

Ethernet Header là thành phần quan trọng trong gói tin Ethernet, nằm ở tầng liên kết dữ liệu (Data Link Layer) của mô hình OSI. Trong Wireshark, khi phân tích một gói tin Ethernet, bạn sẽ thấy phần Ethernet header trước khi đi vào phần payload của các giao thức cao hơn (như IP, TCP, UDP).

### 5.2. Cấu trúc

- Destination MAC Address (6 byte): Địa chỉ MAC của thiết bị nhận gói tin. Được sử dụng để định tuyến gói tin trong mạng cục bộ.
- Source MAC Address (6 byte): Địa chỉ MAC của thiết bị gửi gói tin. Địa chỉ này giúp xác định nguồn gốc của gói tin.
- EtherType (2 byte): Trường này xác định giao thức tầng trên mà gói tin đang sử dụng. Một số giá trị EtherType phổ biến:
  - 0x0800: IPv4.
  - 0x86DD: IPv6.
  - 0x0806: ARP (Address Resolution Protocol).
  - 0x8100: VLAN tagging (IEEE 802.1Q).

## 6. Các kỹ thuật lọc gói tin

### 6.1. Lọc theo giao thức

- tcp: Chỉ hiển thị các gói tin TCP.
- udp: Chỉ hiển thị các gói tin UDP.
- icmp: Chỉ hiển thị các gói tin ICMP (dùng trong Ping).
- dns: Lọc gói tin thuộc giao thức DNS.
- http: Lọc gói tin thuộc giao thức HTTP.

### 6.2. Lọc theo địa chỉ ip

- ip.addr == 192.168.1.1: Hiển thị tất cả gói tin có IP nguồn hoặc đích là 192.168.1.1.

## Network Forensics

- ip.src == 192.168.1.1: Chỉ hiển thị gói tin có IP nguồn là 192.168.1.1.
- ip.dst == 192.168.1.2: Chỉ hiển thị gói tin có IP đích là 192.168.1.2.

### 6.3. Lọc theo địa chỉ MAC

- eth.addr == 00:1A:2B:3C:4D:5E: Hiển thị tất cả gói tin có địa chỉ MAC nguồn hoặc đích là 00:1A:2B:3C:4D:5E.
- eth.src == 00:1A:2B:3C:4D:5E: Chỉ hiển thị gói tin có địa chỉ MAC nguồn cụ thể.
- eth.dst == 00:1A:2B:3C:4D:5F: Chỉ hiển thị gói tin có địa chỉ MAC đích cụ thể.

### 6.4. Lọc theo cổng (port)

- tcp.port == 80: Hiển thị các gói tin TCP sử dụng cổng 80 (HTTP).
- udp.port == 53: Hiển thị các gói tin UDP sử dụng cổng 53 (DNS).
- tcp.srcport == 443: Chỉ hiển thị gói tin TCP có cổng nguồn là 443 (HTTPS).
- udp.dstport == 67: Chỉ hiển thị gói tin UDP có cổng đích là 67 (DHCP).

### 6.5. Lọc theo Flag trong TCP

- tcp.flags.syn == 1: Lọc các gói tin có cờ SYN (gói tin bắt đầu kết nối TCP).
- tcp.flags.fin == 1: Lọc các gói tin có cờ FIN (gói tin kết thúc kết nối TCP).
- tcp.flags.rst == 1: Lọc các gói tin có cờ RST (gói tin reset kết nối).
- tcp.flags.ack == 1: Lọc các gói tin có cờ ACK (gói tin xác nhận trong kết nối).

### 6.6. Lọc theo giao thức tầng cao

- http.request: Hiển thị tất cả gói tin chứa yêu cầu HTTP.
- http.response: Hiển thị tất cả gói tin chứa phản hồi HTTP.

## Network Forensics

- dnsqry.name == "example.com": Lọc các gói DNS yêu cầu truy vấn đến tên miền example.com.

### 6.7. Lọc theo kích thước gói tin (Packet Length)

- frame.len > 1000: Hiển thị các gói tin có độ dài lớn hơn 1000 byte.
- frame.len <= 64: Hiển thị các gói tin có độ dài nhỏ hơn hoặc bằng 64 byte.

### 6.8. Lọc gói tin lỗi

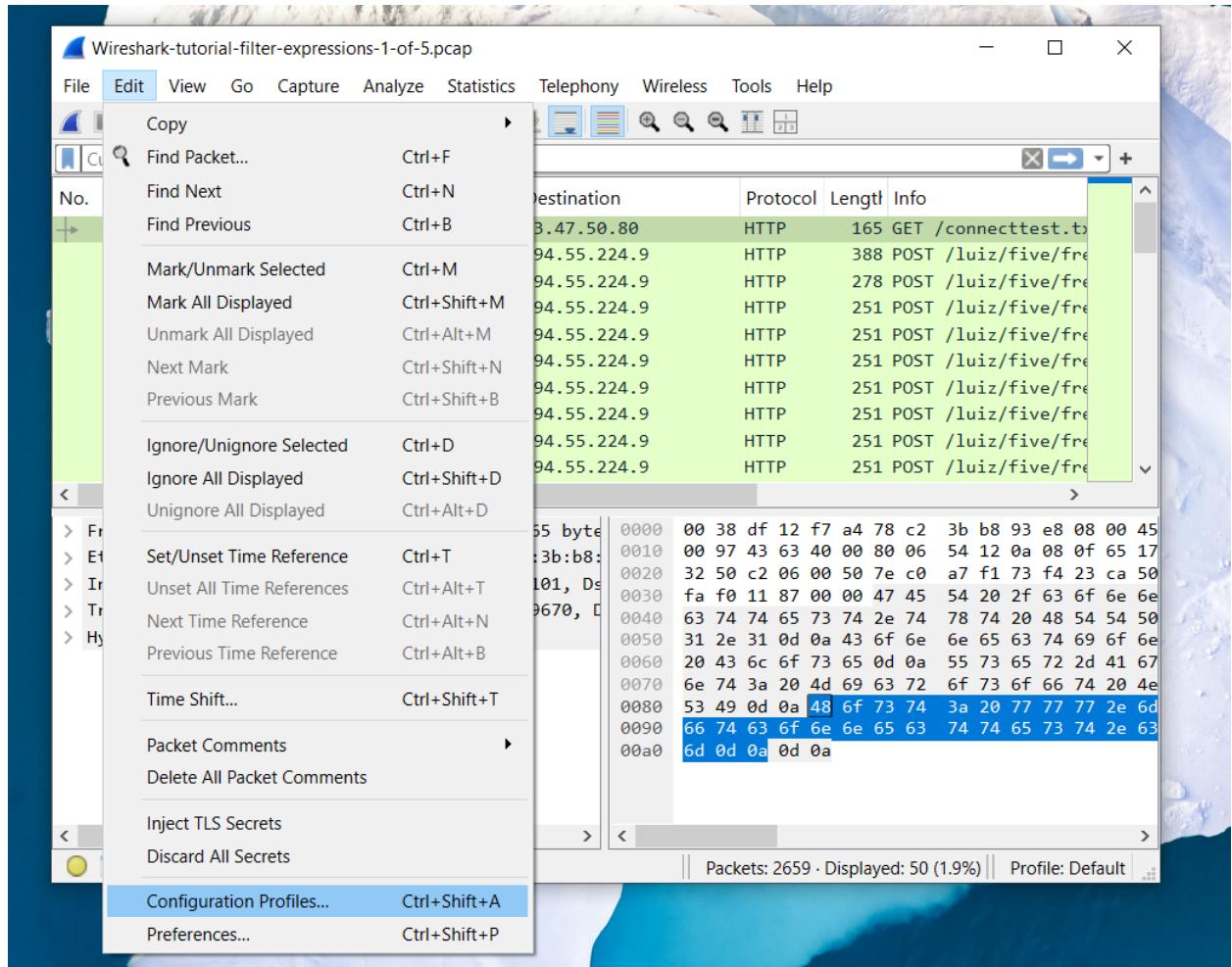
- tcp.analysis.retransmission: Hiển thị các gói tin TCP bị gửi lại (retransmissions).
- tcp.analysis.fast\_retransmission: Lọc các gói tin có lỗi gửi nhanh do phát hiện mất gói tin.
- udp.checksum\_bad == 1: Hiển thị các gói tin UDP có lỗi checksum.

## 7. Các kỹ thuật dùng wireshark

### 1/ Profile check

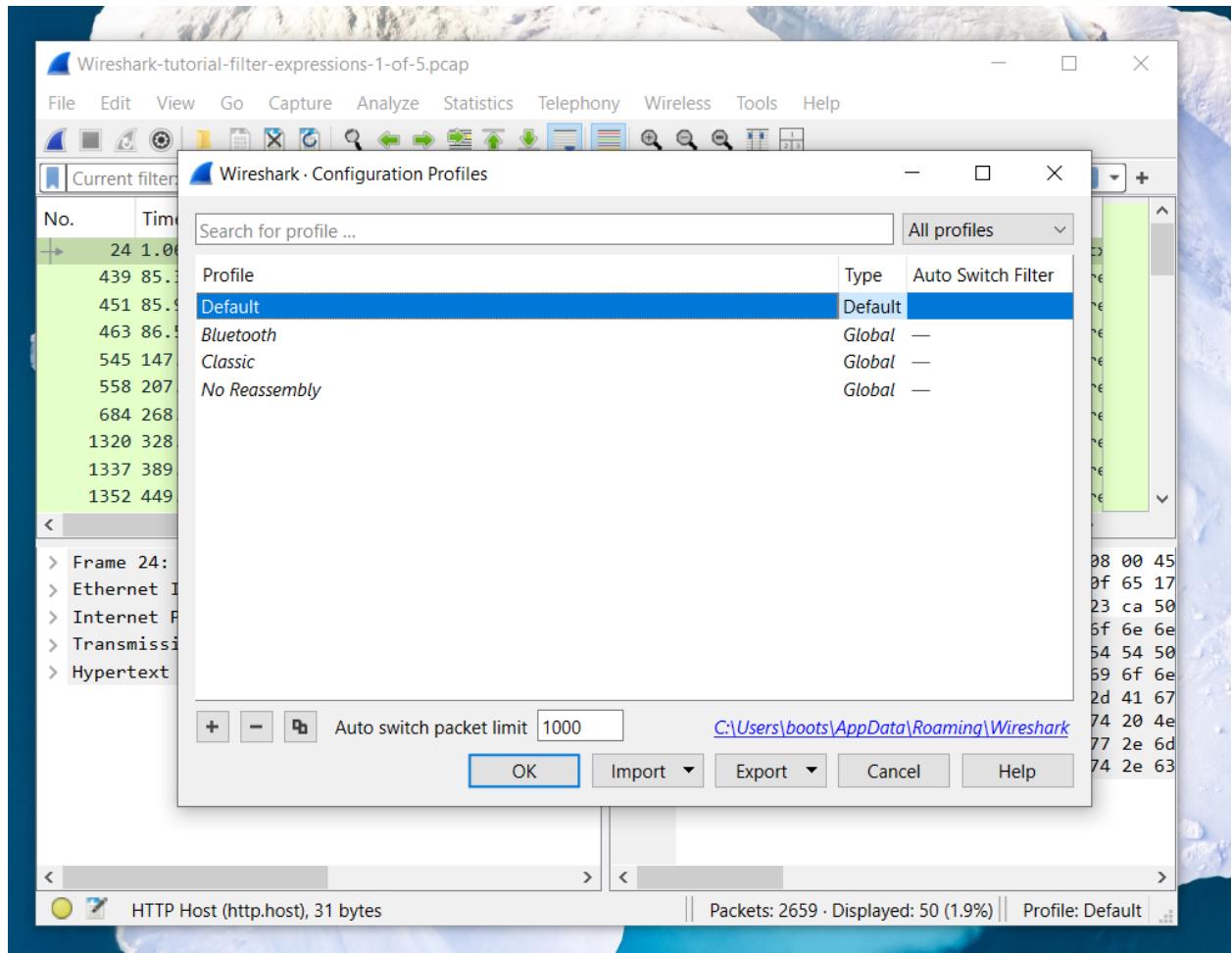
- Kiểm tra profile bằng hình thức sau đây

# Network Forensics



Hình 1. Step 1 - Profile check

## Network Forensics



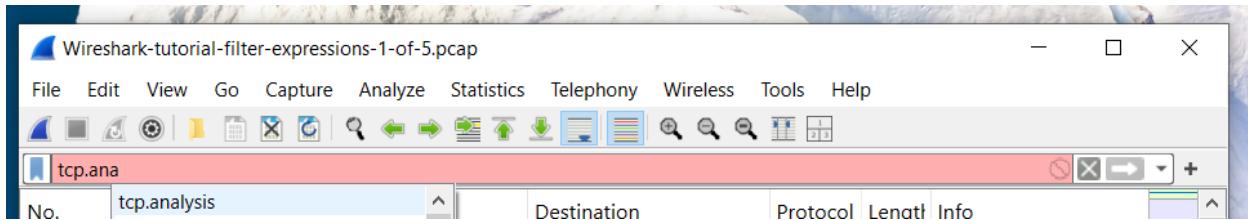
Hình 2. Step 2 - profile check

- Sau khi kiểm tra profile khớp, thì chúng ta hãy tiến tới bộ lọc wireshark

### 2/ Wireshark Filter

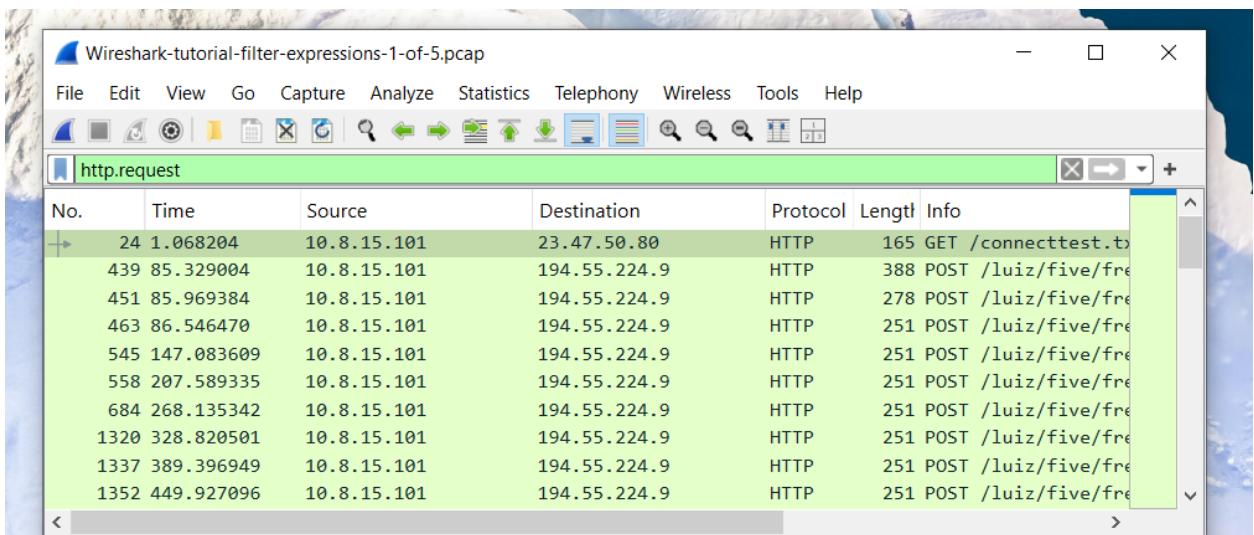
- Trong cấu hình mặc định của Wireshark, bộ lọc hiển thị là một thanh nằm ngay phía trên màn hình cột. Đây là nơi để nhập các biểu thức để lọc chế độ xem của chúng về khung Ethernet, gói IP hoặc phân đoạn TCP từ pcap. Khi nhập vào thanh bộ lọc hiển thị, Wireshark cung cấp một danh sách các đề xuất dựa trên văn bản đã nhập.
- Miễn là thanh bộ lọc hiển thị vẫn còn màu đỏ, biểu thức sẽ không được chấp nhận

## Network Forensics



Hình 3. Step 1 - Wireshark Filter

- Nhập http.request vào bộ lọc hiển thị và nhấn Enter. Nếu thanh lọc có màu xanh lá cây, biểu thức đã được chấp nhận và nó sẽ hoạt động đúng



Hình 4. Step 2 - Wireshark Filter

Boolean Operator	Expression	Alternate Expression
Equals	<code>==</code>	<code>eq</code>
Not	<code>!</code>	<code>not</code>
And	<code>&amp;&amp;</code>	<code>and</code>
Or	<code>  </code>	<code>or</code>

Hình 5. Các toán tử trong Wireshark Filter

Các ví dụ ngẫu nhiên về biểu thức bộ lọc hiển thị Wireshark bao gồm:

- `ip.addr eq 10.8.15[.]1 and dns.qry.name.len > 36`
- `http.request && ip.addr == 10.8.15[.]101`

- http.request || http.response
- dnsqry.name contains microsoft or icmp

### 3/ Filtering for Web Traffic

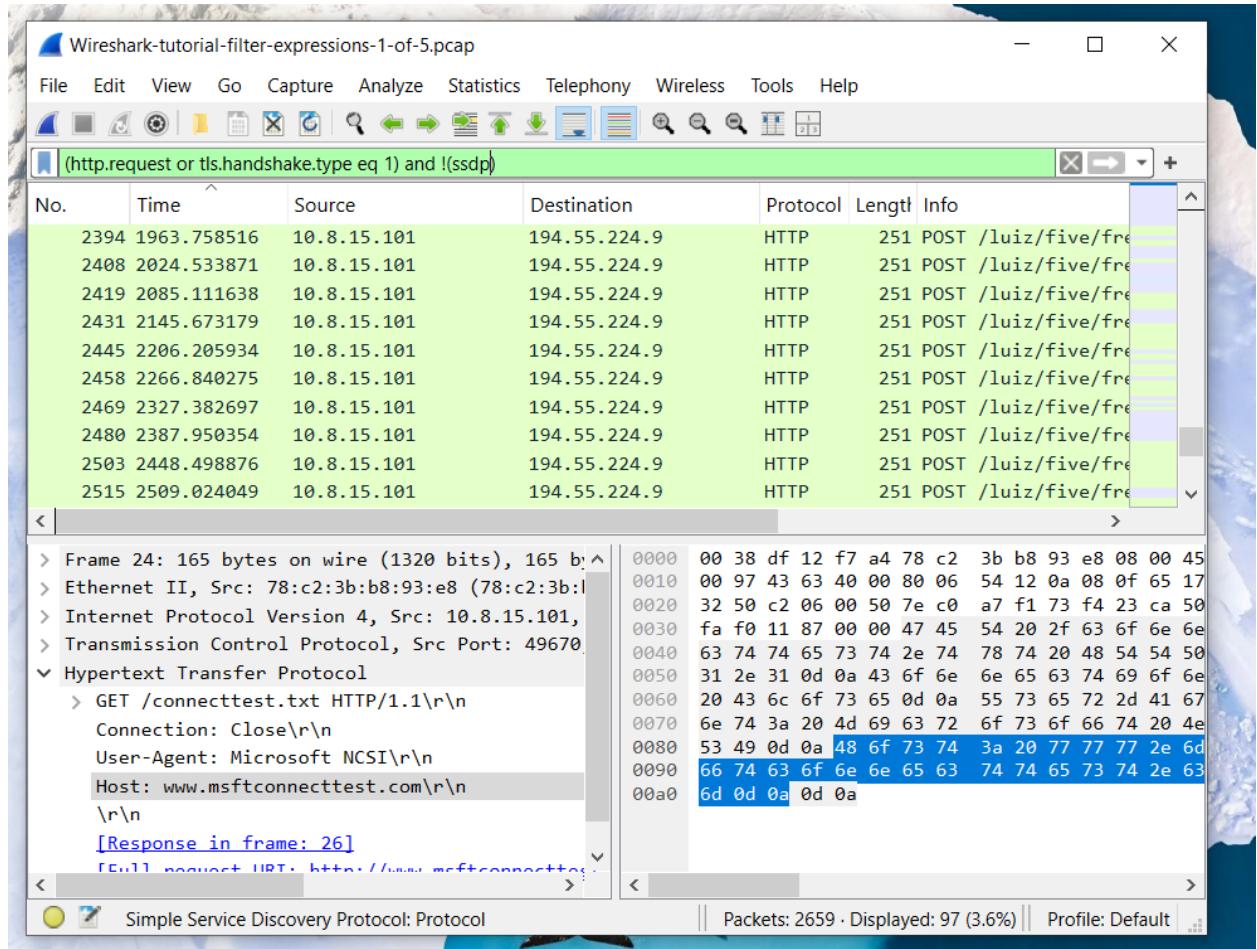
Biểu thức http.request hiển thị URL cho các yêu cầu HTTP và tls.handshake.type eq 1 hiển thị các tên miền được sử dụng trong lưu lượng truy cập HTTPS hoặc SSL / TLS.

Đối với lưu lượng truy cập web được tạo bởi máy chủ Windows, kết quả từ bộ lọc này bao gồm các yêu cầu HTTP qua cổng UDP 1900. SSDP được sử dụng để khám phá các thiết bị plug-and-play và không được liên kết với lưu lượng truy cập web thông thường. Chúng tôi có thể loại trừ lưu lượng ssdp trong kết quả của mình bằng cách sửa đổi biểu thức bộ lọc thành:

**(http.request or tls.handshake.type eq 1) and !(ssdp)**

Mặc dù dấu ngoặc đơn trong biểu thức bộ lọc ở trên không bắt buộc trong Wireshark phiên bản 4, chúng tôi khuyên bạn nên bao gồm chúng để đảm bảo khả năng tương thích biểu thức bộ lọc với các phiên bản Wireshark cũ hơn. Sử dụng bộ lọc này trên pcap đầu tiên của chúng tôi

## Network Forensics

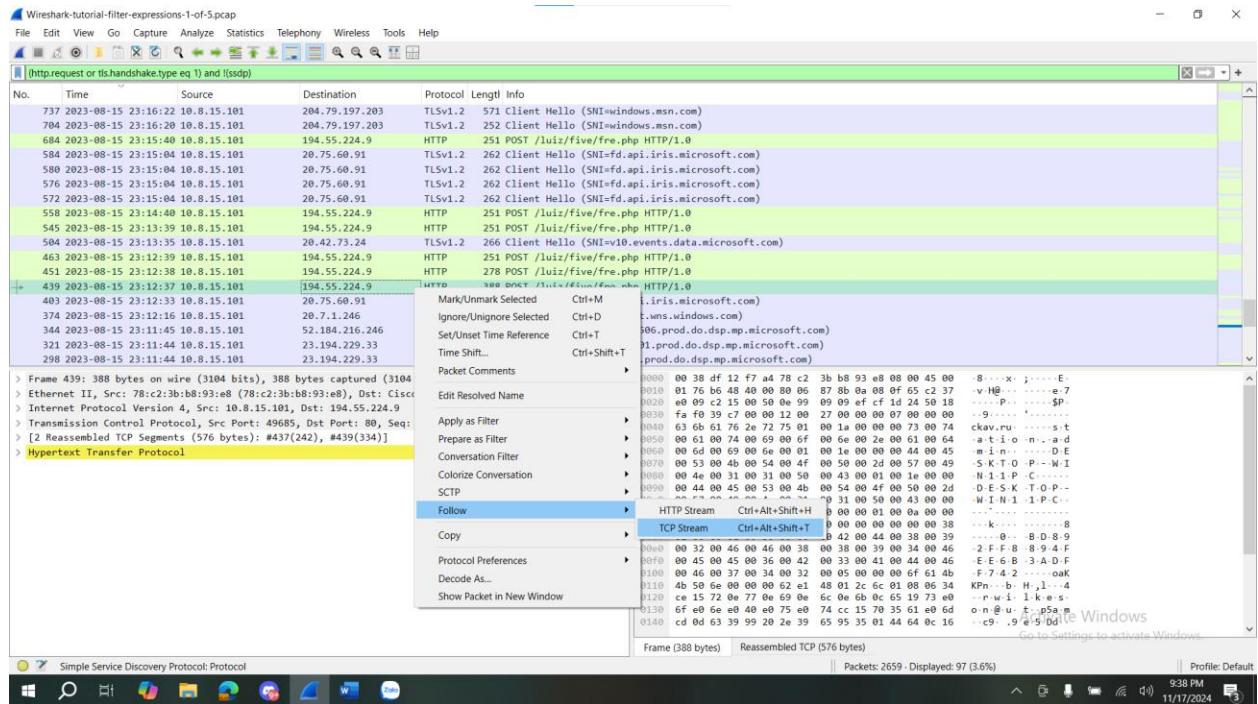


Hình 6. Step 1 - Filtering for Web Traffic

Xem xét lưu lượng truy cập được hiển thị trong Hình 7 cho thấy một số dòng yêu cầu HTTP POST không được mã hóa liên quan đến phần mềm độc hại đến URL hxxp://194.55.224 [.]9/liuz/five/fre.php, được báo cáo cho Threadbox vào tháng 8/2023.

Để kiểm tra lưu lượng truy cập, hãy nhấp vào bất kỳ dòng nào cho lưu lượng truy cập đến 194.55.224 [.]9 để chọn khung, sau đó nhấp chuột phải để hiển thị menu. Từ menu, chọn "Follow" rồi chọn "TCP Stream" hoặc "HTTP Stream", như hình

## Network Forensics



Hình 7. Step 2 - Filtering for Web Traffic

Thao tác này sẽ hiển thị một cửa sổ mới và chúng ta có thể xem lại biểu diễn ASCII về nội dung của lưu lượng HTTP không được mã hóa này

Mở pcap thứ hai trong Wireshark. Đây là lưu lượng truy cập từ nhiễm trùng IcedID (Bokbot) biến thể tiêu chuẩn. Nó chứa lưu lượng HTTP đến vrondafarih[.]lưu lượng com và HTTPS đến cả magiketchinn[.]com và magizanqomo[.]com. Cả ba đều được xác định là tên miền liên quan đến IcedID vào tháng 7/2023.

Sau đây là hình cho thấy các tên miền liên quan đến IcedID này trong pcap thứ hai của chúng tôi bằng cách sử dụng bộ lọc web cơ bản trong Wireshark.

# Network Forensics

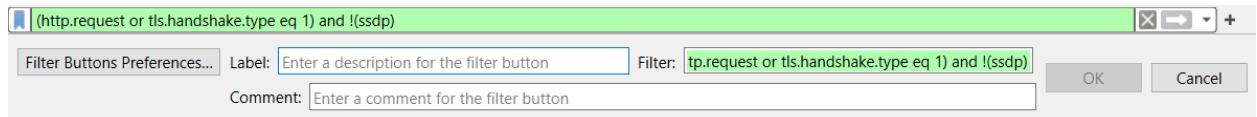
(http.request or tls.handshake.type eq 1) and !(ssdp)					
Time	Dst	Dst port	Host	Info	
2023-07-27 15:16:20	139.59.26.99	80	vrondafarih.com	GET / HTTP/1.1	
2023-07-27 15:17:30	2.56.177.122	443	magizanqomo.com	Client Hello	
2023-07-27 15:17:30	208.111.176...	80	ctldl.windowsupdate.com	GET /msdownload/upda	
2023-07-27 15:17:31	2.56.177.122	443	magizanqomo.com	Client Hello	
2023-07-27 15:17:31	2.56.177.122	443	magizanqomo.com	Client Hello	
2023-07-27 15:17:31	2.56.177.122	443	magizanqomo.com	Client Hello	
2023-07-27 15:17:34	2.56.177.122	443	magiketchinn.com	Client Hello	
2023-07-27 15:22:31	2.56.177.122	443	magiketchinn.com	Client Hello	
2023-07-27 15:27:33	2.56.177.122	443	magiketchinn.com	Client Hello	
2023-07-27 15:32:35	2.56.177.122	443	magiketchinn.com	Client Hello	
2023-07-27 15:37:36	2.56.177.122	443	magiketchinn.com	Client Hello	
2023-07-27 15:42:38	2.56.177.122	443	magiketchinn.com	Client Hello	
2023-07-27 15:47:39	2.56.177.122	443	magiketchinn.com	Client Hello	
2023-07-27 15:52:41	2.56.177.122	443	magiketchinn.com	Client Hello	
2023-07-27 15:52:59	20.49.150.241	443	settings-win.data.micro...	Client Hello	
2023-07-27 15:53:59	72.21.81.240	80	ctldl.windowsupdate.com	GET /msdownload/upda	
2023-07-27 15:53:59	72.21.81.240	80	ctldl.windowsupdate.com	GET /msdownload/upda	
2023-07-27 15:57:43	2.56.177.122	443	magiketchinn.com	Client Hello	
2023-07-27 16:02:45	2.56.177.122	443	magiketchinn.com	Client Hello	

Hình 8. Step 3 - Filtering for Web Traffic

## 4/ Creating Filter Buttons

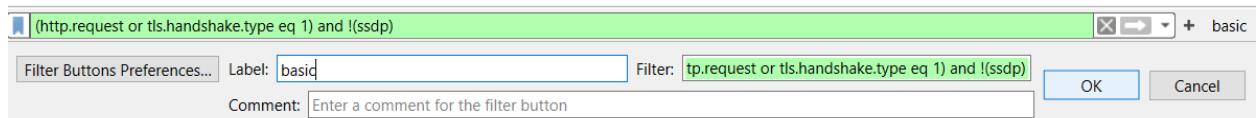
Ở phía bên phải của thanh bộ lọc Wireshark là dấu cộng để thêm nút lọc.

Đảm bảo chúng ta vẫn đang sử dụng bộ lọc web cơ bản được hiển thị. Sau khi đảm bảo bộ lọc này đã được triển khai, hãy nhấp vào dấu cộng như hình



Hình 9. Step 1 - Creating Filter Buttons

Nhấp vào dấu cộng sẽ tạo ra một bảng tạm thời dưới thanh bộ lọc, như đã lưu ý ở trên trong hình. Bộ lọc phải chứa biểu thức đã được triển khai trong thanh bộ lọc. Vì đây là bộ lọc web cơ bản của chúng tôi, hãy nhập basic vào trường Label và nhấp vào nút OK như hình dưới đây



Hình 10. Step 2 - Creating Filter Buttons

## Network Forensics

Button Label	Filter Expression
basic	basic (http.request or tls.handshake.type eq 1) and !(ssdp)
basic+	basic (http.request or tls.handshake.type eq 1 or (tcp.flags.syn eq 1 and tcp.flags.ack eq 0)) and !(ssdp)
basic+dns	basic (http.request or tls.handshake.type eq 1 or (tcp.flags.syn eq 1 and tcp.flags.ack eq 0) or dns) and !(ssdp)

Hình 11. Trick - Creating Filter Buttons

Khi kiểm tra lưu lượng truy cập đáng ngờ trong Wireshark, chúng ta nên sử dụng một phương pháp lũy tiến. Bắt đầu đơn giản với bộ lọc web cơ bản của chúng tôi, sau đó kiểm tra lưu lượng truy cập không phải web khác bằng bộ lọc "basic+".

Trong Bảng , biểu thức bộ lọc "basic+" hiển thị thông tin giống như bộ lọc "basic" của chúng tôi, nhưng nó bao gồm các phân đoạn TCP có cờ SYN chứ không phải cờ ACK bằng cách thêm hoặc (tcp.flags.syn eq 1 và tcp.flags.ack eq 0). Điều này hiển thị các phân đoạn TCP SYN tiết lộ sự bắt đầu của luồng TCP. Với bộ lọc này, chúng tôi có thể tìm thấy lưu lượng truy cập không phải web trong pcap.

Bộ lọc "basic+" cũng hiển thị bất kỳ nỗ lực kết nối TCP nào không thành công. Tùy thuộc vào địa chỉ IP, các nỗ lực kết nối TCP lặp đi lặp lại và không thành công có thể chỉ ra máy chủ C2 ngoại tuyến khi pcap được ghi lại.

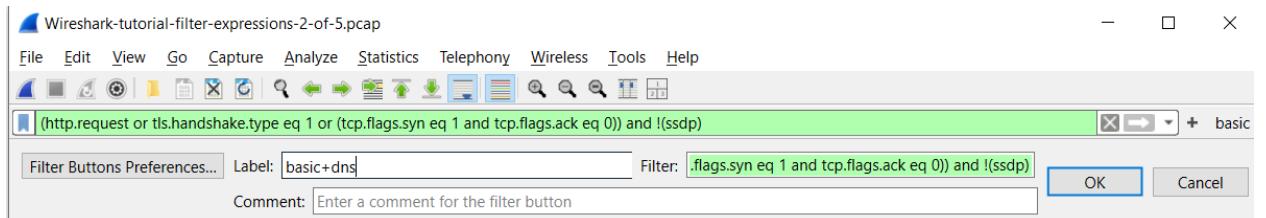
Sau khi kiểm tra bộ lọc "basic +", chúng ta nên xem lại bộ lọc "basic + dns" để kiểm tra xem có bất kỳ hoạt động DNS đáng chú ý nào không.

Trong Bảng 2, biểu thức bộ lọc "basic+dns" hiển thị dữ liệu giống như bộ lọc "basic+" của chúng tôi, nhưng nó bao gồm hoặc dns. Bộ lọc này hiển thị bất kỳ truy vấn DNS nào trong pcap. Nó rất hữu ích để xác định tên miền liên quan đến lưu lượng truy cập không phải web.

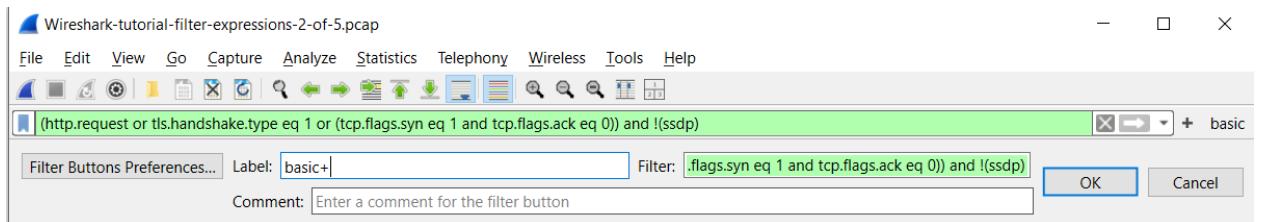
Hơn nữa, nếu máy chủ C2 của mẫu phần mềm độc hại ngoại tuyến khi pcap được ghi lại, bộ lọc này có thể tiết lộ một hoặc nhiều miền C2 liên quan đến

## Network Forensics

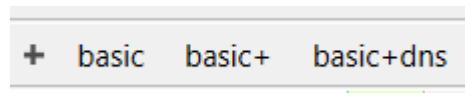
bất kỳ nỗ lực kết nối không thành công nào. Cuối cùng, bộ lọc này có thể tiết lộ các ví dụ về đường hầm DNS



Hình 12. Step 3 - Creating Filter Buttons



Hình 13. Step 4 - Creating Filter Buttons



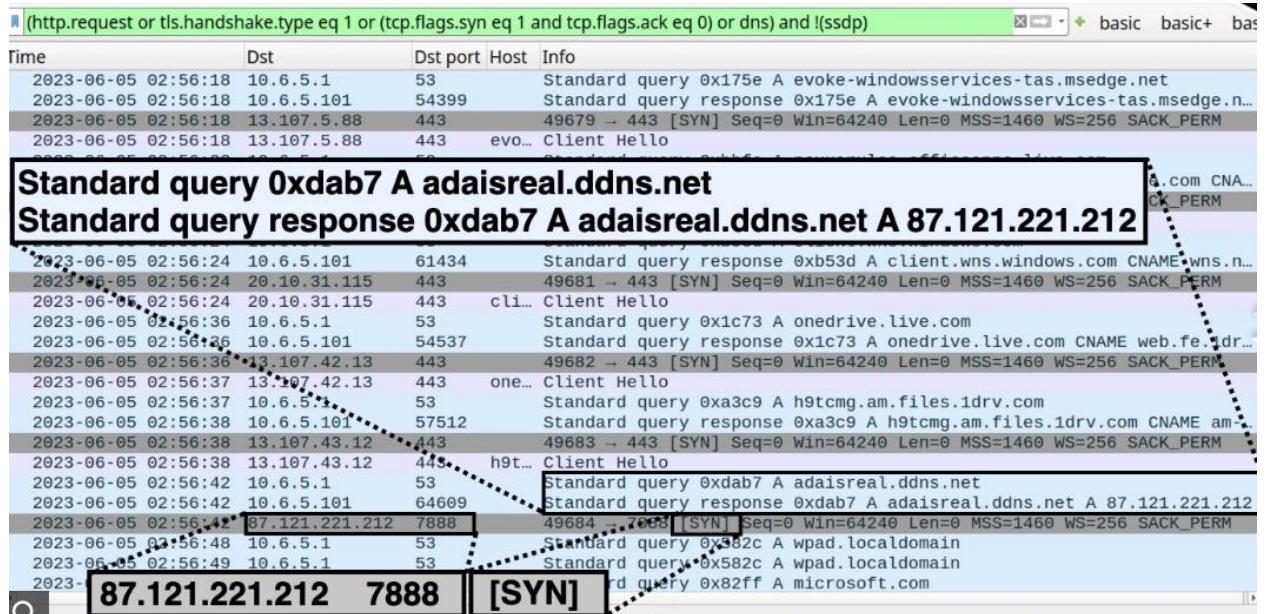
Hình 14. Step 5 - Creating Filter Buttons

### 5/ Filtering for Non-Web Traffic

Mở pcap thứ ba trong Wireshark. Pcap này chứa lưu lượng truy cập sau lây nhiễm được tạo bởi phần mềm độc hại Công cụ truy cập từ xa (RAT). Sử dụng bộ lọc web cơ bản của chúng tôi, không có gì rõ ràng nổi bật trong lưu lượng truy cập. Tuy nhiên, bằng cách sử dụng bộ lọc web "basic + dns" của chúng tôi và cuộn qua các kết quả, chúng tôi có thể thấy mọi thứ rõ ràng

## Network Forensics

hơn. Chúng tôi có thể tìm thấy truy vấn DNS cho adaisreal.ddns[.]ròng giải quyết đến 87.121.221 [.]212, sau đó là một phân đoạn TCP đến địa chỉ IP đó với cờ SYN trên cổng TCP 7888, như thể hiện bên dưới trong Hình



Hình 15. Filtering for Non-Web Traffic

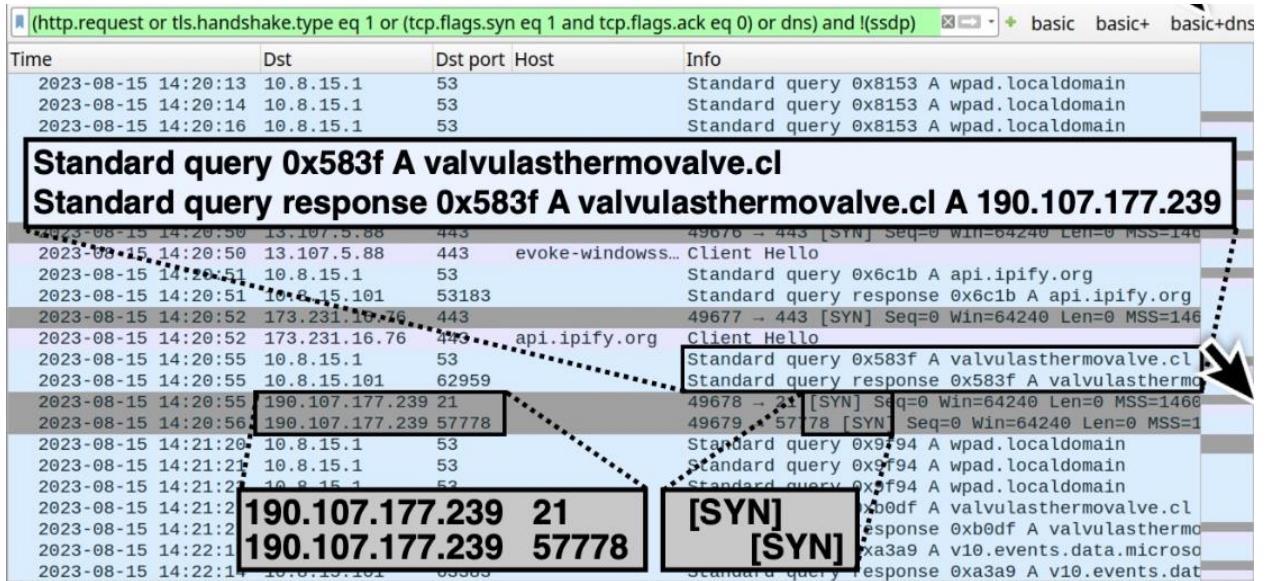
Đây chỉ là một ví dụ, nhưng các RAT khác nhau và các loại phần mềm độc hại khác cũng tạo ra các loại lưu lượng truy cập không phải web tương tự. Bộ lọc "basic+dns" của chúng tôi cung cấp một cách để tìm kiếm hoạt động độc hại không phải web.

### 6/ Filtering for FTP Traffic

Một số lưu lượng lây nhiễm sử dụng các giao thức phổ biến mà Wireshark có thể dễ dàng giải mã. pcap thứ tư của chúng tôi Wireshark-tutorial-filter-expressions-4-of-5.pcap chứa hoạt động sau lây nhiễm gây ra bởi một tệp thực thi phần mềm độc hại tạo ra lưu lượng FTP. Bộ lọc "basic + dns" của chúng tôi hiển thị lưu lượng truy cập qua cổng TCP 21 và một cổng TCP

## Network Forensics

khác sau truy vấn DNS đến valvulasthermovalve [.]cl như thể hiện dưới đây trong Hình



Hình 16. Filtering for FTP Traffic

chúng ta cũng có thể thấy lưu lượng truy cập HTTPS đến api.ipify[.]tổ chức ngay trước khi hoạt động FTP. Mặc dù tên miền này vốn không độc hại, nhưng phần mềm độc hại thường sử dụng dịch vụ để kiểm tra địa chỉ IP của máy chủ bị nhiễm.

Bộ lọc "basic+dns" của chúng tôi có thể giúp tìm lưu lượng FTP không được mã hóa, nhưng các biểu thức bộ lọc khác sẽ phù hợp hơn với tìm kiếm FTP. Hai bộ lọc Wireshark cơ bản cho lưu lượng FTP không được mã hóa được hiển thị bên dưới trong Bảng

Filter Expression	Description
ftp	FTP activity in the control channel (TCP port 21)
ftp-data	FTP activity in the data channel (ephemeral TCP port)

Hình 17. Trick for FTP

## Network Forensics

Biểu thức bộ lọc có mục đích chung để xem xét hoạt động FTP không được mã hóa là:

***ftp.request.command and (ftp-data and tcp.seq eq 1)***

Nhập biểu thức trên vào thanh bộ lọc hiển thị của Wireshark và nhấn enter.

Kết quả sẽ trông tương tự như ảnh chụp màn hình trong Hình

Time	Dst	Dst port	Info
2023-08-15 14:20...	190.107.177...		21 Request: USER cva19491@valvulasthermovalve.cl
2023-08-15 14:20...	190.107.177...		21 Request: PASS LILKOOLL14!!
2023-08-15 14:20...	190.107.177...		21 Request: OPTS utf8 on
2023-08-15 14:20...	190.107.177...		21 Request: PWD
2023-08-15 14:20...	190.107.177...		21 Request: TYPE I
2023-08-15 14:20...	190.107.177...		21 Request: PASV
2023-08-15 14:20...	190.107.177...		21 Request: STOR PW_user1-DESKTOP-USER1PC_2023_08_15_14_20_53.html
2023-08-15 14:20...	190.107.177...		57778 FTP Data: 743 bytes (PASV) (STOR PW_user1-DESKTOP-USER1PC_2023_08_15_14_20_53.htm

Hình 18. Kết quả nhập biểu thức

hiển thị tên người dùng và mật khẩu cho trang FTP bị xâm nhập này, sau đó là lệnh STOR để gửi một tệp HTML đến máy chủ FTP. Điều này thể hiện dữ liệu bị đánh cắp đang được trích xuất từ máy chủ Windows bị nhiễm. Chúng ta có thể làm theo các luồng TCP để xem lại các lệnh FTP và kiểm tra dữ liệu bị đánh cắp. Nếu cần, bạn có thể lưu biểu thức bộ lọc này dưới dạng nút bộ lọc để sử dụng trong tương lai.

### 7/ Filtering for Email (Spambot) Traffic

Ngoài FTP, phần mềm độc hại có thể sử dụng các giao thức phổ biến khác cho lưu lượng truy cập độc hại. Phần mềm độc hại Spambot có thể biến một máy chủ bị nhiễm thành một spambot được thiết kế để liên tục gửi email. Điều này được đặc trưng bởi một lượng lớn các yêu cầu DNS đến các máy chủ thư khác nhau, theo sau là lưu lượng SMTP trên các cổng TCP 25, 465, 587 và các cổng khác ít được liên kết với lưu lượng SMTP

- Câu lệnh ***smtp or dns***

## Network Forensics

Time	Dst	Dst port	Info
2023-08-15 14:20...	10.8.15.1		53 Standard query 0x8153 A wpad.localdomain
2023-08-15 14:20...	10.8.15.1		53 Standard query 0x8153 A wpad.localdomain
2023-08-15 14:20...	10.8.15.1		53 Standard query 0xbe81 A checkappexec.microsoft.com
2023-08-15 14:20...	10.8.15.101		55139 Standard query response 0xbe81 A checkappexec.microsoft.com CNAME wd-prod-s...
2023-08-15 14:20...	10.8.15.1		53 Standard query 0x3fb9 A evoke-windowservices-tas.msedge.net
2023-08-15 14:20...	10.8.15.101		49617 Standard query response 0x3fb9 A evoke-windowservices-tas.msedge.net CNAME
2023-08-15 14:20...	10.8.15.1		53 Standard query 0x6c1b A api.ipify.org
2023-08-15 14:20...	10.8.15.101		53183 Standard query response 0x6c1b A api.ipify.org CNAME api4.ipify.org A 173.2...
2023-08-15 14:20...	10.8.15.1		53 Standard query 0x583f A valvulasthermovalve.cl
2023-08-15 14:20...	10.8.15.101		62959 Standard query response 0x583f A valvulasthermovalve.cl A 190.107.177.239
2023-08-15 14:21...	10.8.15.1		53 Standard query 0x9f94 A wpad.localdomain
2023-08-15 14:21...	10.8.15.1		53 Standard query 0x9f94 A wpad.localdomain
2023-08-15 14:21...	10.8.15.1		53 Standard query 0x9f94 A wpad.localdomain
2023-08-15 14:21...	10.8.15.1		53 Standard query 0xb0df A valvulasthermovalve.cl
2023-08-15 14:21...	10.8.15.101		65342 Standard query response 0xb0df A valvulasthermovalve.cl A 190.107.177.239
2023-08-15 14:22...	10.8.15.1		53 Standard query 0xa3a9 A v10.events.data.microsoft.com
2023-08-15 14:22...	10.8.15.101		63383 Standard query response 0xa3a9 A v10.events.data.microsoft.com CNAME win-g...
2023-08-15 14:22...	10.8.15.1		53 Standard query 0x1f3b A v20.events.data.microsoft.com
2023-08-15 14:22...	10.8.15.101		54794 Standard query response 0x1f3b A v20.events.data.microsoft.com CNAME win-g...

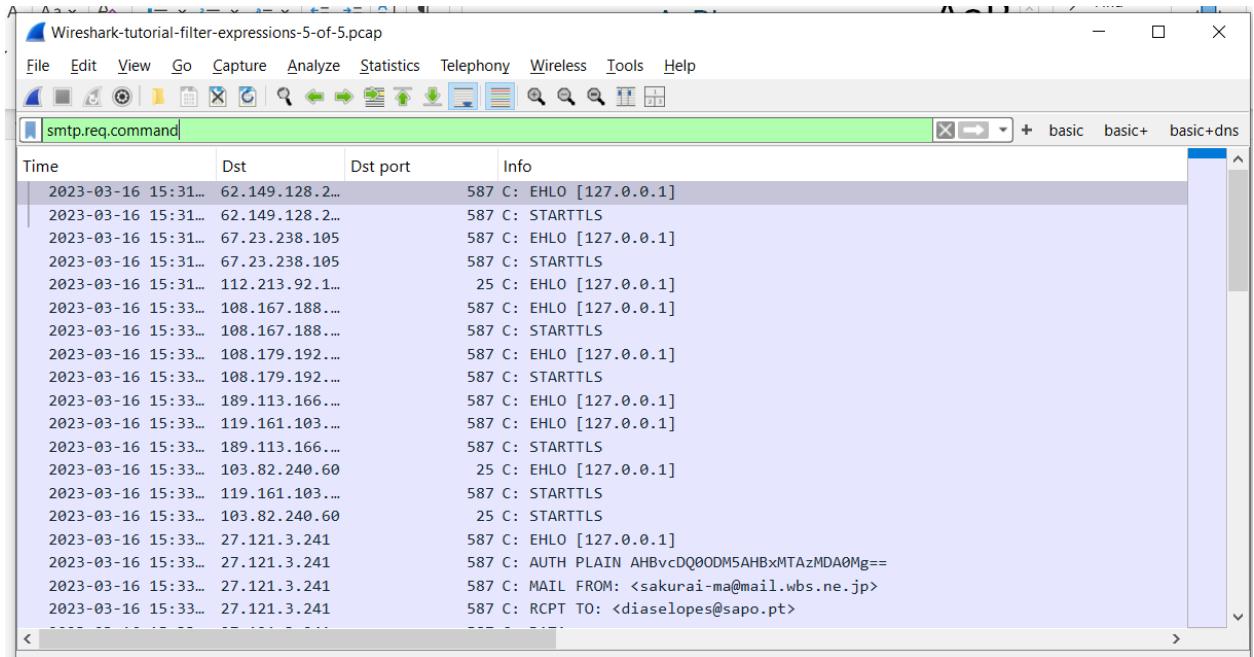
Hình 19. Lọc các smtp và dns

Nếu bạn cuộn qua các kết quả, bạn sẽ tìm thấy một số truy vấn DNS cho các miền máy chủ thư khác nhau và các câu lệnh SMTP khác nhau ở ngoài cùng bên phải trong cột "Info".

Bây giờ hãy nhập bộ lọc sau vào thanh bộ lọc: **smtp.req.command**

Các kết quả hiển thị dưới đây trong hình cho thấy máy chủ bị nhiễm đã liên lạc với một số địa chỉ IP khác nhau cho các máy chủ thư trong một khoảng thời gian tương đối ngắn. Lưu ý cách hầu hết các yêu cầu SMTP trạng thái STARTTLS, thiết lập một đường hầm được mã hóa sau kết nối SMTP ban đầu. Hầu hết lưu lượng email được mã hóa và hầu hết hoạt động của spambot cũng được mã hóa.

## Network Forensics



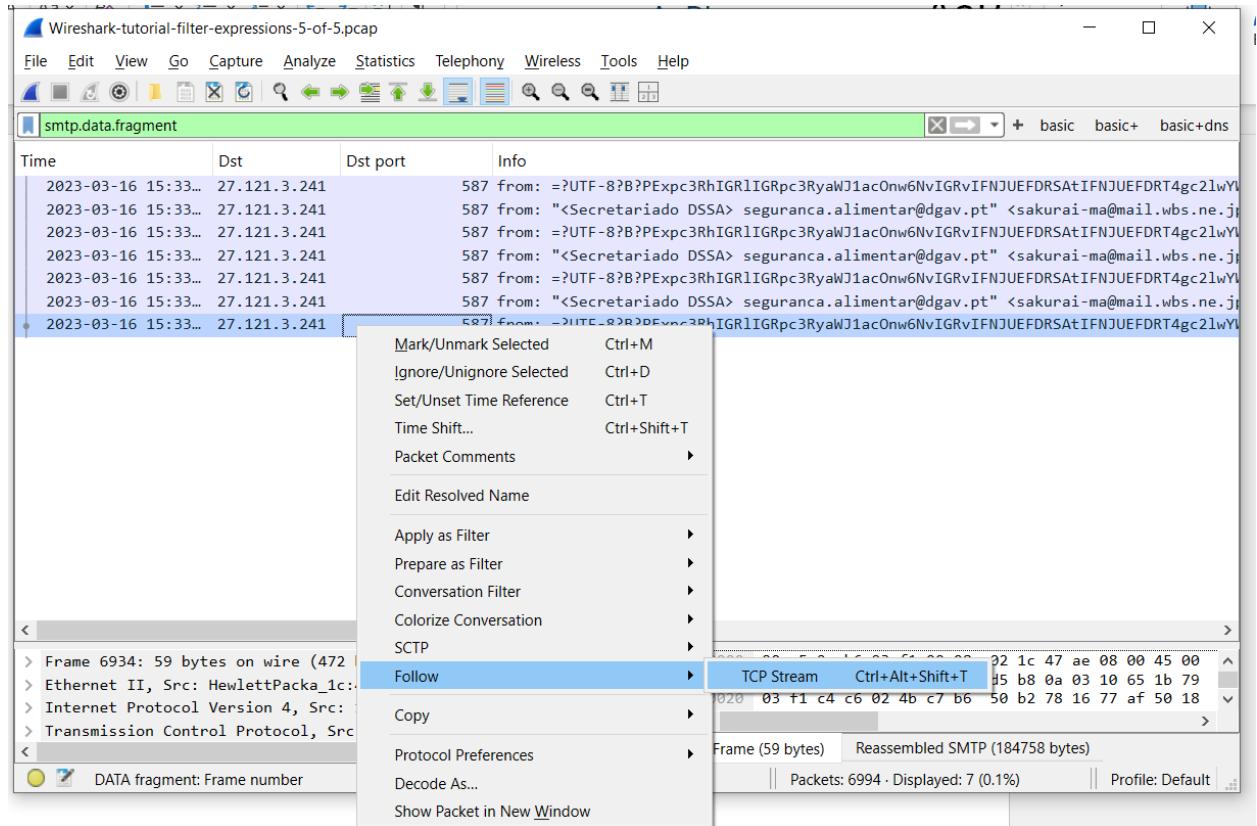
Hình 20. Lọc các command request

Tuy nhiên, lưu lượng truy cập spam bot có thể có các email không được mã hóa mà chúng tôi có thể xem xét. Để tìm các thông báo này, hãy nhập biểu thức sau vào thanh bộ lọc của Wireshark:

**smtp.data.fragment**

Điều này sẽ tiết lộ bảy kết quả trong màn hình cột như thể hiện bên dưới trong hình. Chúng tôi có thể theo dõi luồng TCP cho bất kỳ trong số này để điều tra thêm các thông báo này.

# Network Forensics



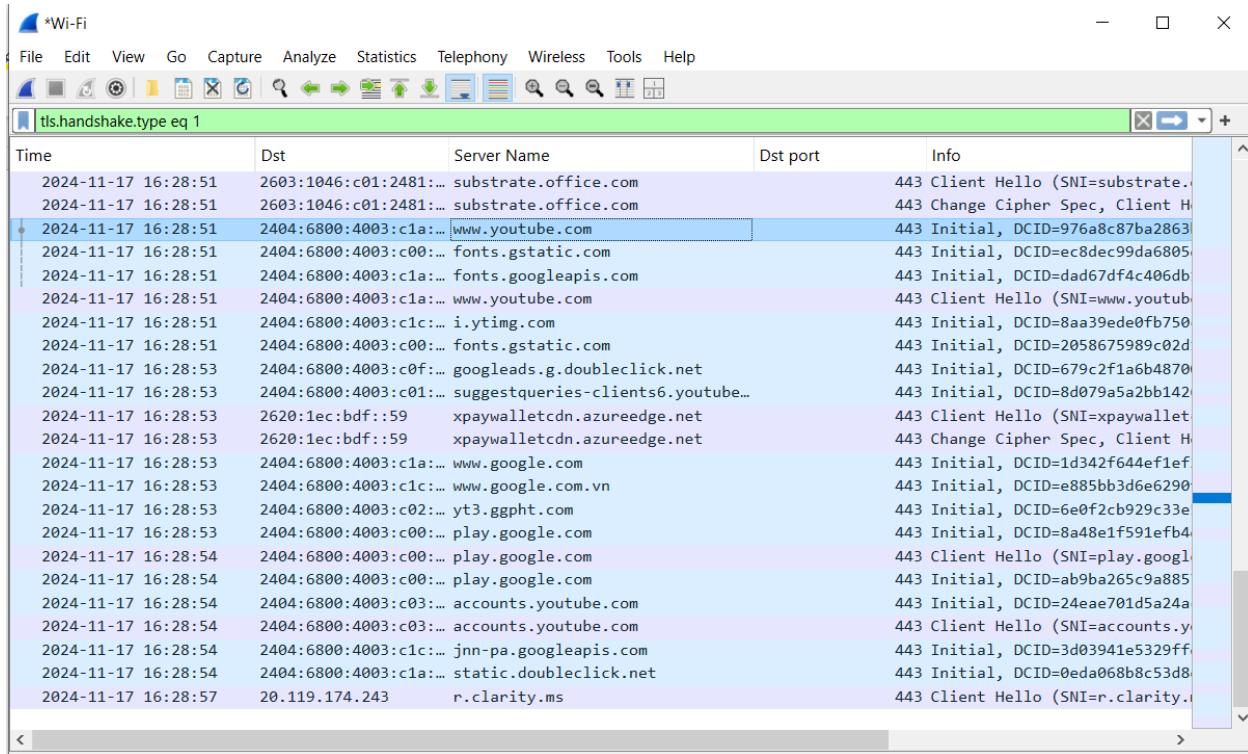
Hình 21. Biểu thức để lọc spam bot

Mặc dù không rõ ràng, đây là những biểu thức lọc phổ biến nhất hữu ích để kiểm tra lưu lượng truy cập spam bot.

## 1/ HTTPS Web Traffic

Truy cập [www.youtube.com](http://www.youtube.com)

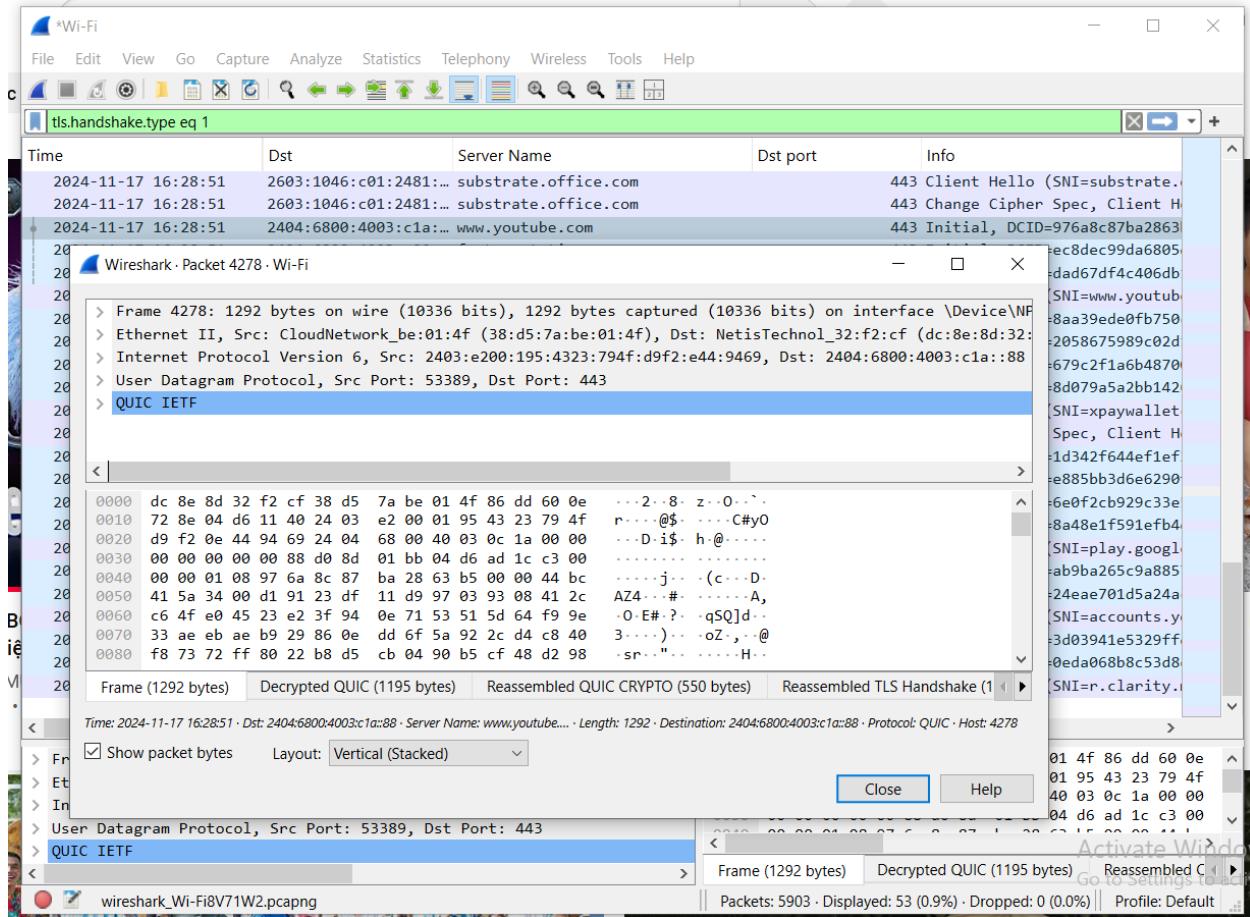
## Network Forensics



Hình 22. Step 1 - HTTPS Web Traffic

Luồng TCP của lưu lượng HTTPS đến và đi từ máy chủ tại www.youtube.com

## Network Forensics



Hình 23. Step 2 - HTTPS Web Traffic

## 2/ Encryption Key Log File

Nhật ký khóa mã hóa là một tệp văn bản. Các nhật ký này được tạo bằng kỹ thuật Man in the Middle (MitM) khi pcap được ghi lại ban đầu. Nếu không có tệp nào như vậy được tạo khi pcap được ghi lại, bạn không thể giải mã lưu lượng HTTPS trong pcap đó.

## Network Forensics



```
Wireshark-tutorial-KeysLogFile.txt - Notepad
File Edit Format View Help
CLIENT_RANDOM 5e85016c2010478311178455b55ee4c6cf4fee8ef941b1a3fa07c3b3f86365c d6b14d253ebe0fb069185666afc8ca76216b1b288bc85b3c6e01a736e757
CLIENT_RANDOM 5e85017170b9e2362c1c72d0dbc4a9133441079bf5ce0bc2999016d120967a 63c841f68786288e84a10dcfaaecabde9081e4b74a2e1362eef50a72e92f
CLIENT_RANDOM 5e850178c6e06ad9d7eb8ac04fea8d4d4e5f58fe3008dbc795cd46688a85c921 62d5fd980a621fd5b9345f02e0758f45fa8a34a6a0cdb55354306aa0f403
CLIENT_RANDOM 5e8501dceb9af936b5e346dc4ac6665f8dc610b37d1a24386fe1a4d67298cbe3 f1b55a36c27c690523b401f4dbe50215ff017d0c0913cd655bdf59d8fa5
CLIENT_RANDOM 5e85026cf7611ba45bcd52a958fe86a531cc3e826cb4bcede4d2cdf79e8 5ffc09c727b394a48d6a4b30ee3d721bda4e3cb4cbd3a4968fa451fc8152
CLIENT_RANDOM 5e850391f7b8619e1c8f65c12cf0731816248133a0c89feb9211cc9e3943747 f0402c77ed02a0e06f99ded18aff4c5a025ba3df2458415c35c69e1f412f5
CLIENT_RANDOM 5e85041398abd602bad132952274a94371868b8dc23e145ceb7ec59562a51b2 937cce1ae96965d9e9287569b321abe94211db27f9b92ab659eeab4298
CLIENT_RANDOM 5e85049b139b551f1ceba4372df8ed41602b0ca006251dd3283c38ddbf7d2782 a1d744fa7bd0a5ebad55d26491a30116b5af4b233313919586173a14686c
```

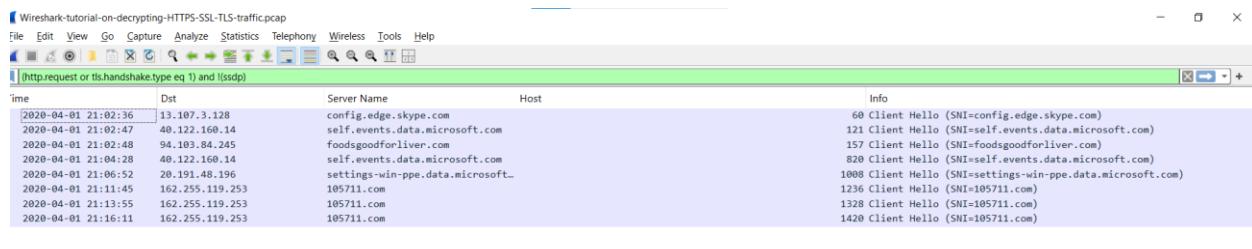
Hình 24. Các mã hoá kí tự

### 3/ HTTPS Traffic Without the Key Log File

Sử dụng bộ lọc web cơ bản như được mô tả trong hướng dẫn trước này về bộ lọc Wireshark. Bộ lọc cơ bản của chúng tôi cho Wireshark là:

**(http.request or tls.handshake.type eq 1) and !(ssdp)**

Pcap này là từ nhiễm phần mềm độc hại Dridex trên máy chủ Windows 10. Tất cả lưu lượng truy cập web, bao gồm cả hoạt động lây nhiễm, là HTTPS. Nếu không có tệp nhật ký khóa, chúng ta không thể thấy bất kỳ chi tiết nào về lưu lượng truy cập, chỉ có địa chỉ IP, cổng TCP và tên miền, như thể hiện trong Hình



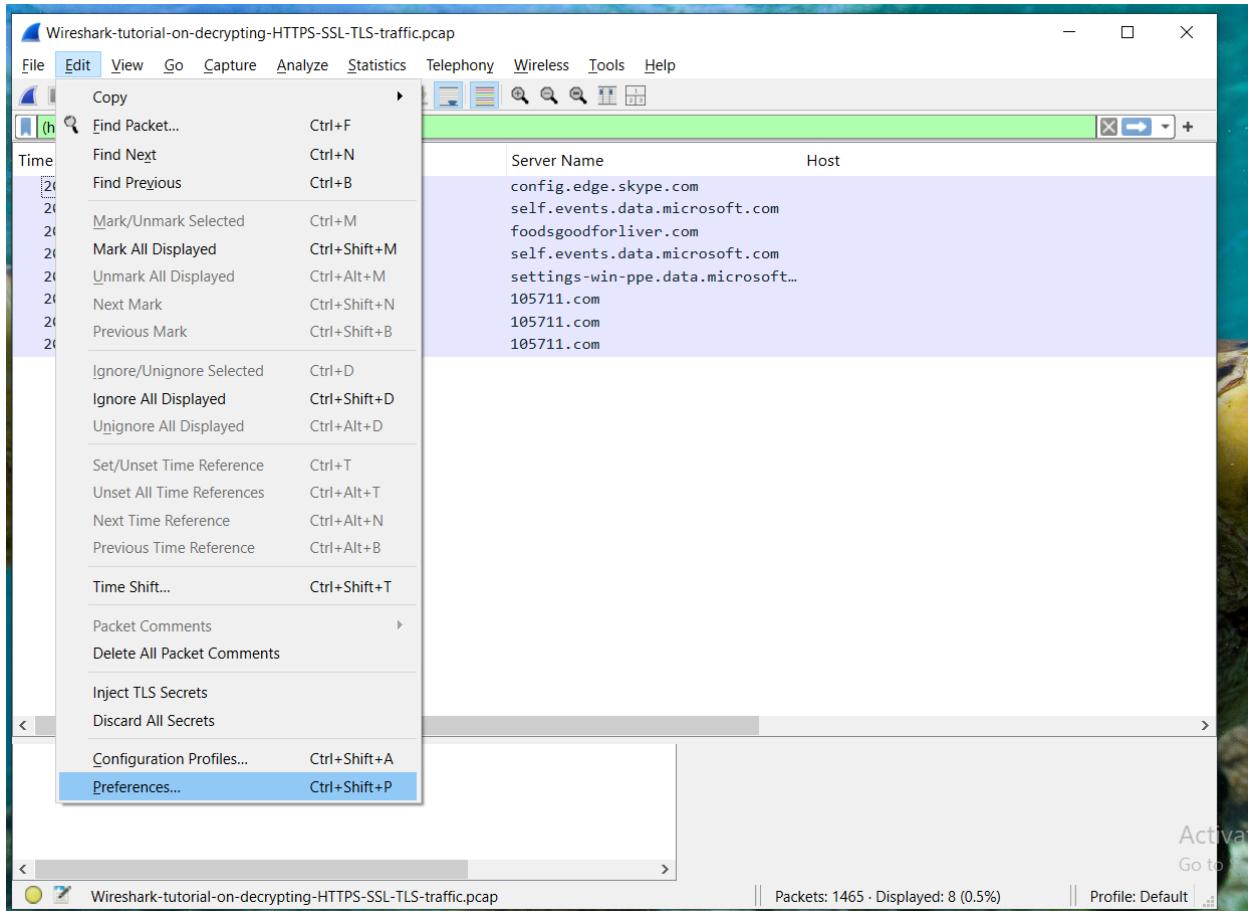
Time	Dst	Server Name	Host	Info
2020-04-01 21:02:36	13.107.3.128	config.edge.skyype.com		60 Client Hello (SNI=config.edge.skyype.com)
2020-04-01 21:02:47	40.122.160.14	self.events.data.microsoft.com		121 Client Hello (SNI=self.events.data.microsoft.com)
2020-04-01 21:02:48	94.193.84.245	foodsgoodforliver.com		157 Client Hello (SNI=foodsgoodforliver.com)
2020-04-01 21:04:28	40.122.160.14	self.events.data.microsoft.com		820 Client Hello (SNI=self.events.data.microsoft.com)
2020-04-01 21:06:52	20.191.48.196	settings-win-ppe.data.microsoft...		1006 Client Hello (SNI=settings-win-ppe.data.microsoft.com)
2020-04-01 21:11:45	162.255.119.253	105711.com		1236 Client Hello (SNI=105711.com)
2020-04-01 21:13:55	162.255.119.253	105711.com		1328 Client Hello (SNI=105711.com)
2020-04-01 21:16:11	162.255.119.253	105711.com		1420 Client Hello (SNI=105711.com)

Hình 25. HTTPS Traffic Without the Key Log File

### 4/ Loading the Key Log File

Sau đó sử dụng đường dẫn menu **Edit --> Preferences** để hiển thị Preferences Menu, như thể hiện trong Hình

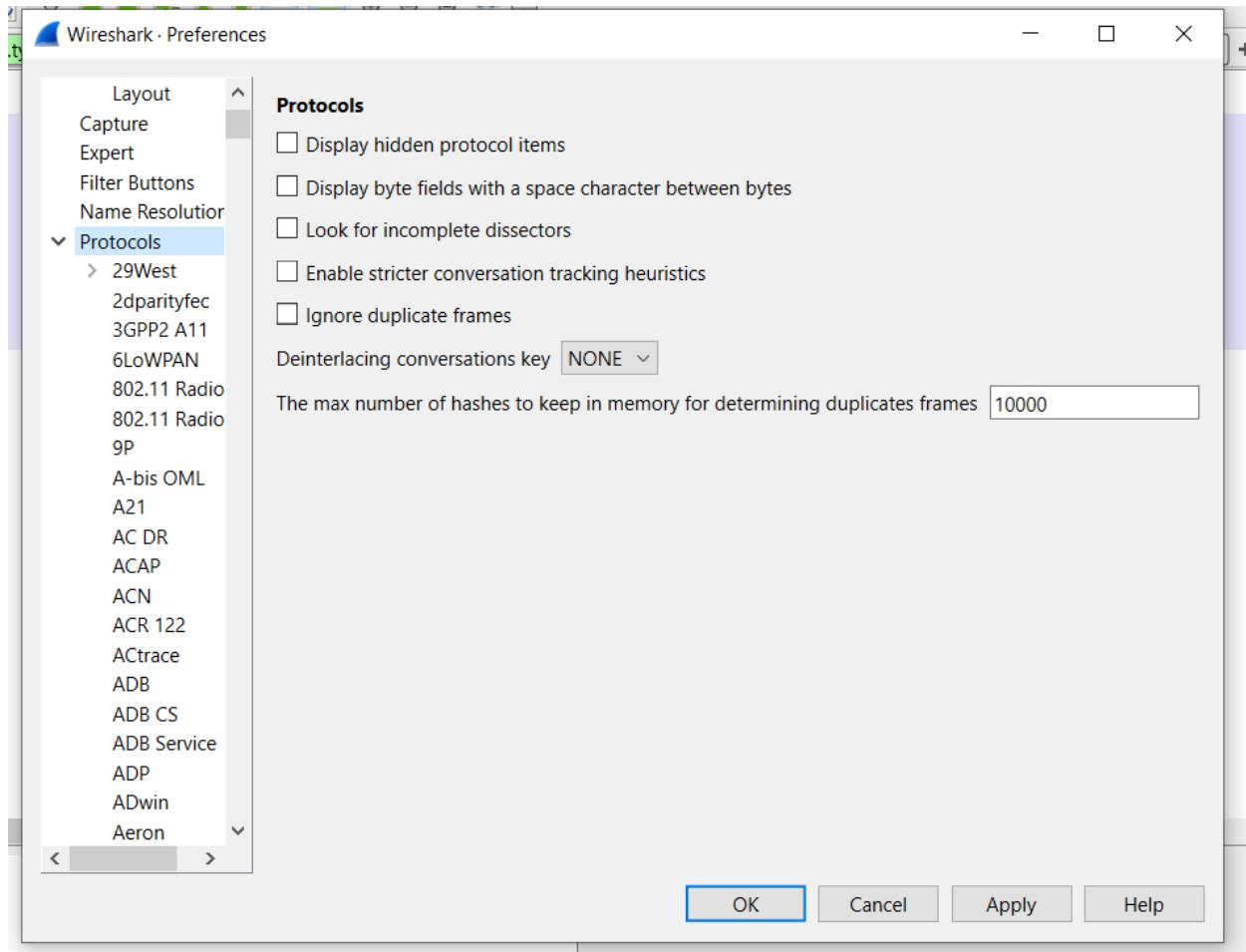
## Network Forensics



Hình 26. Step 1 - Loading the Key Log File

Ở phía bên trái của Preferences Menu, nhấp vào Protocols, như thể hiện trong Hình

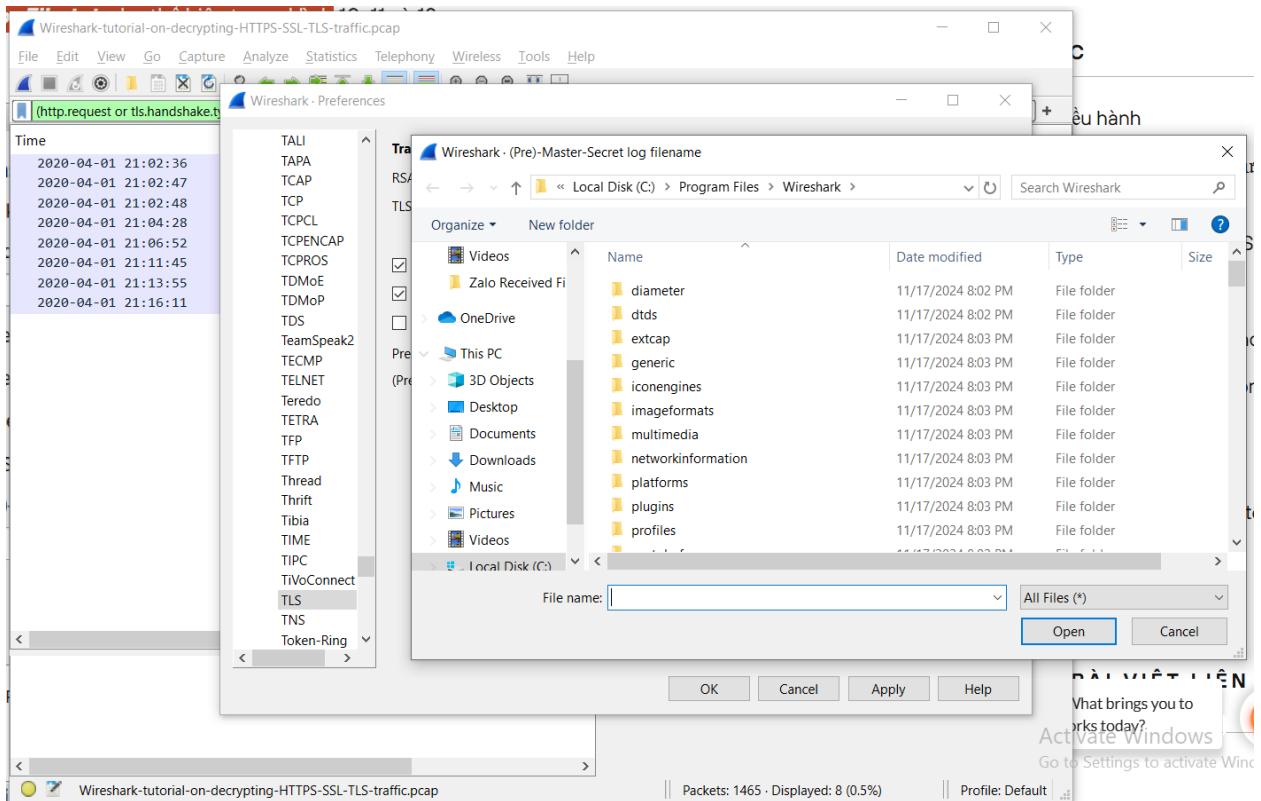
## Network Forensics



Hình 27. Step 2 - Loading the Key Log File

Nếu bạn đang sử dụng Wireshark phiên bản 2.x, hãy cuộn xuống cho đến khi bạn tìm thấy SSL và chọn nó. Nếu bạn đang sử dụng Wireshark phiên bản 3.x, hãy cuộn xuống TLS và chọn nó. Khi bạn đã chọn SSL hoặc TLS, bạn sẽ thấy một dòng cho tên tệp nhât ký (*Pre-Master-Secret*). Nhấp vào nút "Browse" và chọn tệp nhật ký khóa của chúng tôi có tên *Wireshark-tutorial-KeysLogFile.txt*, như thể hiện trong Hình

# Network Forensics



Hình 28. Step 3 - Loading the Key Log File

Sau đó chọn đến file chứa tệp nhật ký nhấn apply và chọn ok

## 5/HTTPS Traffic With the Key Log File

Khi đã nhập vào "OK", khi sử dụng bộ lọc cơ bản, màn hình cột Wireshark của bạn sẽ liệt kê các yêu cầu HTTP được giải mã dưới mỗi dòng HTTPS, như thể hiện trong Hình

Time	Dst	Server Name	Host	Info
2020-04-01 21:02:36	13.107.3.128	config.edge.skype.com		60 Client Hello (SNI=config.edge.skype.com)
2020-04-01 21:02:44	13.107.3.128			78 GET /config/v2/Office/word/16.0.12026.20264/Production/CC7&Clientid=%7bd61AB268-C26A-439D-BB15-2A80EDFCA6A3%7d...
2020-04-01 21:02:47	40.122.160.14	self.events.data.microsoft.com		121 Client Hello (SNI=self.events.data.microsoft.com)
2020-04-01 21:02:47	40.122.160.14			131 POST /OneCollector/1.0/ HTTP/1.1 (application/bond-compact-binary)
2020-04-01 21:02:47	40.122.160.14			142 POST /OneCollector/1.0/ HTTP/1.1 (application/bond-compact-binary)
2020-04-01 21:02:48	94.103.84.245	foodsgoodforliver.com		157 Client Hello (SNI=foodsgoodforliver.com)
2020-04-01 21:02:49	94.103.84.245			165 GET /invest_20.dll HTTP/1.1
2020-04-01 21:04:28	40.122.160.14	self.events.data.microsoft.com		820 Client Hello (SNI=self.events.data.microsoft.com)
2020-04-01 21:04:28	40.122.160.14			830 POST /OneCollector/1.0/ HTTP/1.1 (application/bond-compact-binary)
2020-04-01 21:06:52	20.191.48.196	settings-win-ppe.data.microsoft...		1008 Client Hello (SNI=settings-win-ppe.data.microsoft.com)
2020-04-01 21:06:53	20.191.48.196			1018 GET /settings/v2.0/Storage/StorageHealthEvaluation?os=Windows&deviceClass=Windows.Desktop&appVer=1.0.0.0 HTTP/...
2020-04-01 21:11:45	162.255.119.253	105711.com		1236 Client Hello (SNI=105711.com)
2020-04-01 21:11:45	162.255.119.253			1244 POST /docs.php HTTP/1.1
2020-04-01 21:13:55	162.255.119.253	105711.com		1328 Client Hello (SNI=105711.com)
2020-04-01 21:13:55	162.255.119.253			1336 POST /docs.php HTTP/1.1
2020-04-01 21:16:11	162.255.119.253	105711.com		1420 Client Hello (SNI=105711.com)
2020-04-01 21:16:11	162.255.119.253			1428 POST /docs.php HTTP/1.1

Hình 29. kết quả bộ lọc

## Network Forensics

Trong pcap này, bây giờ chúng ta thấy các yêu cầu HTTP đến các miền microsoft.com và skype.com trước đây bị ẩn trong lưu lượng HTTPS. Chúng tôi cũng tìm thấy lưu lượng truy cập sau đây do nhiễm Dridex:

- foodsgoodforliver[.]com - GET /invest\_20.dll
- 105711[.]com - POST /docs.php
- Yêu cầu GET đối với foodsgoodforliver [.]com trả về tệp DLL cho Dridex. POST yêu cầu 105711[.]com là lưu lượng lệnh và kiểm soát (C2) từ máy chủ Windows bị nhiễm Dridex.
- Chúng tôi có thể xem xét lưu lượng truy cập bằng cách làm theo các luồng HTTP. Nhấp chuột phải vào dòng để chọn nó, sau đó nhấp chuột trái để hiển thị menu để theo dõi luồng HTTP. Hình cho thấy sau luồng HTTP cho yêu cầu HTTP GET tới foodsgoodforliver [.]com.

```
GET /invest_20.dll HTTP/1.1
Connection: Keep-Alive
Accept: */*
User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)
Host: foodsgoodforliver.com

HTTP/1.1 200 OK
Server: nginx
Date: Wed, 01 Apr 2020 21:02:49 GMT
Content-Type: application/octet-stream
Content-Length: 463872
Last-Modified: Wed, 01 Apr 2020 16:29:16 GMT
Connection: keep-alive
ETag: "5e84c15c-71400"
Accept-Ranges: bytes

MZ.....@..... .!..L.!This
program cannot be run in DOS mode.

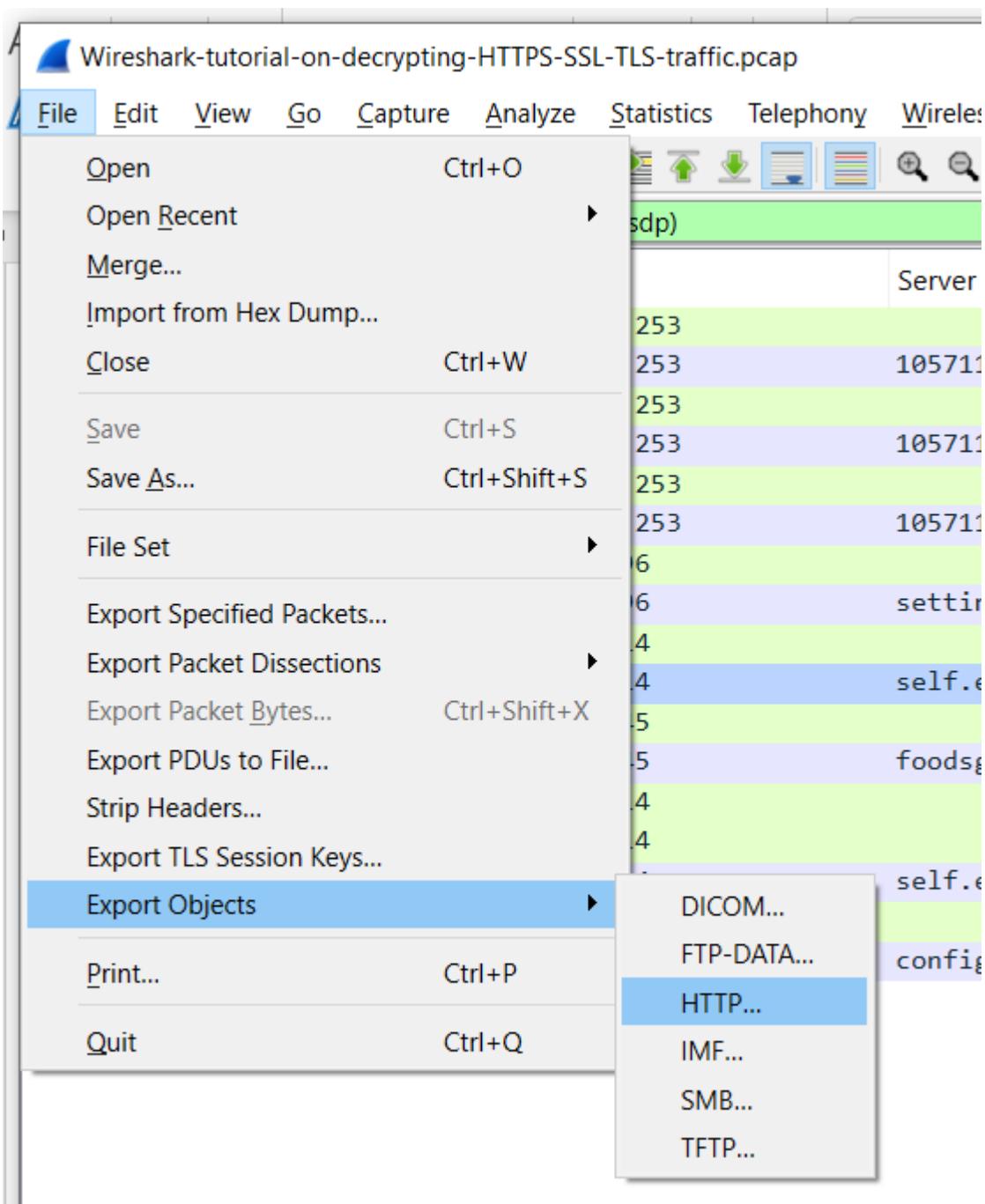
$...../SQ$k2?wk2?wk2?w...wb2?w...w.2?w...ws2?w.i<vy2?w.i:v{2?wbJ.wf2?wk2>w.
2?w.i6vj2?w.i.wj2?w.i=vj2?wRichk2?
W.....PE..L...C..^.....!.....@.....K.....
```

Hình 30. Xem theo các luồng

## **Network Forensics**

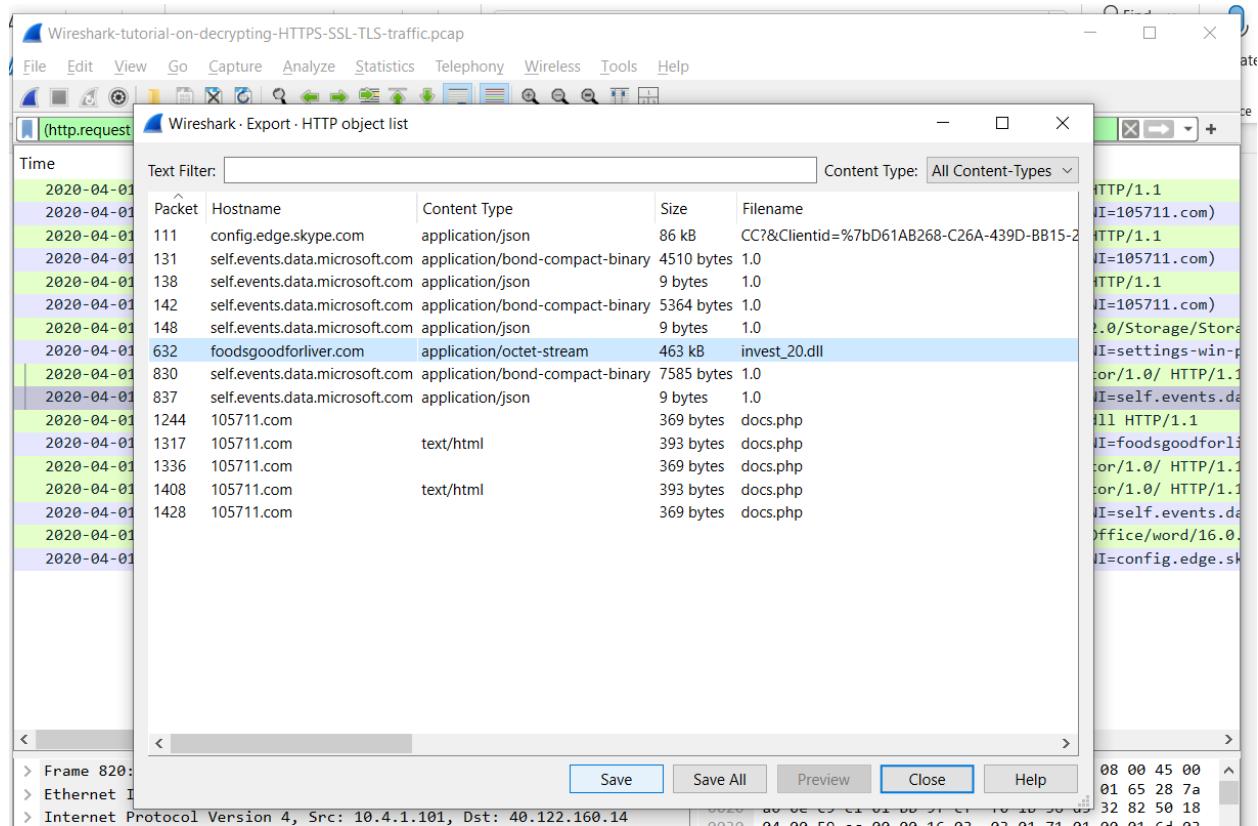
Vì chúng tôi có tệp nhật ký khóa cho lưu lượng truy cập này, giờ đây chúng tôi có thể xuất phần mềm độc hại này từ pcap. Sử dụng đường dẫn menu *File --> Export Objects --> HTTP* để xuất tập tin này từ pcap, như thể hiện trong Hình

## Network Forensics



Hình 31. Trích xuất file

# Network Forensics



Hình 32. Các file có thể trích xuất

## 8. Demo

**Demo 1:** Bạn của bạn, sau khi quên mật khẩu tài khoản của mình, đã gửi cho bạn một file pcap mà anh ấy đã thu thập được trong quá trình đăng nhập vào một trang web. Anh ấy hy vọng bạn có thể giúp anh ấy phục hồi mật khẩu để có thể truy cập lại tài khoản. File pcap này chứa các gói dữ liệu mạng được ghi lại trong khi anh ấy thực hiện việc đăng nhập vào trang web.

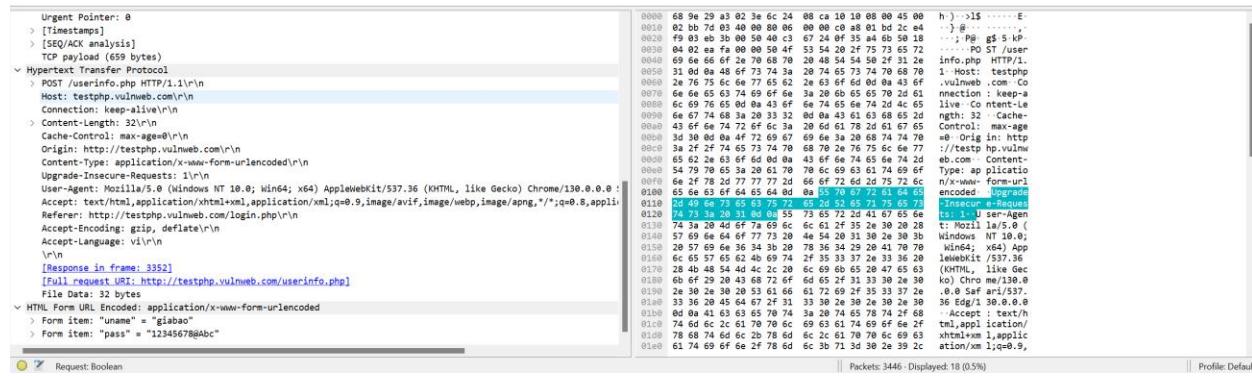
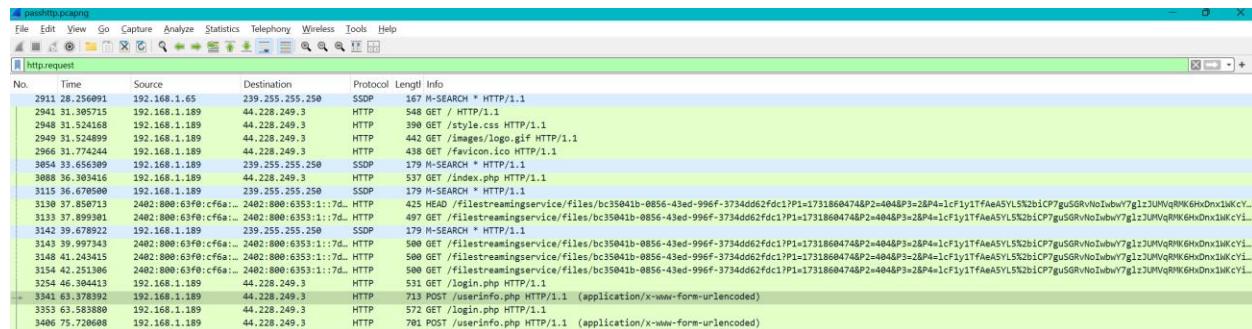
Đầu tiên ta sẽ mở file pcap và lọc các giao thức liên quan tới web như http và https

No.	Time	Source	Destination	Protocol	Length	Info
2911	28.256091	192.168.1.65	239.255.255.258	SSDP	167	M-SEARCH * HTTP/1.1
2941	31.305715	192.168.1.189	44.228.249.3	HTTP	548	GET / HTTP/1.1
2948	31.524165	192.168.1.189	44.228.249.3	HTTP	398	GET /style.css HTTP/1.1
2949	31.524891	192.168.1.189	44.228.249.3	HTTP	442	GET /images/logo.gif HTTP/1.1
2966	31.772444	192.168.1.189	44.228.249.3	HTTP	438	GET /favicon.ico HTTP/1.1
3001	33.658999	192.168.1.189	239.255.255.258	SSDP	179	M-SEARCH * HTTP/1.1
3080	34.678545	192.168.1.189	44.228.249.3	HTTP	507	GET /index.php HTTP/1.1
3115	36.678590	192.168.1.189	239.255.255.258	SSDP	179	N-SEARCH * HTTP/1.1
3129	37.867071	2492.8809:cfa6..	2402.8809:cfa6..	HTTP	435	HEAD /filestreamingservice/files/bc35941b-0856-43ed-996f-3734dd62fd1c1P1+17318604748P2+28P4=1cF1y1TfAeASYLS%2biCPTgu5GRvNoIbwY7g1zJUWqR9MK6HnDnx1kCY...
3133	37.899301	2492.8809:cfa6..	2402.8809:cfa6..	HTTP	497	GET /filestreamingservice/files/bc35941b-0856-43ed-996f-3734dd62fd1c1P1+17318604748P2+28P4=1cF1y1TfAeASYLS%2biCPTgu5GRvNoIbwY7g1zJUWqR9MK6HnDnx1kCY...
3142	39.678922	192.168.1.189	239.255.255.258	SSDP	179	M-SEARCH * HTTP/1.1
3143	39.997343	2492.8809:cfa6..	2402.8809:cfa6..	HTTP	500	GET /filestreamingservice/files/bc35941b-0856-43ed-996f-3734dd62fd1c1P1+17318604748P2+28P4=1cF1y1TfAeASYLS%2biCPTgu5GRvNoIbwY7g1zJUWqR9MK6HnDnx1kCY...
3148	41.243415	2492.8809:cfa6..	2402.8809:cfa6..	HTTP	500	GET /filestreamingservice/files/bc35941b-0856-43ed-996f-3734dd62fd1c1P1+17318604748P2+28P4=1cF1y1TfAeASYLS%2biCPTgu5GRvNoIbwY7g1zJUWqR9MK6HnDnx1kCY...
3154	42.251363	2492.8809:cfa6..	2402.8809:cfa6..	HTTP	500	GET /filestreamingservice/files/bc35941b-0856-43ed-996f-3734dd62fd1c1P1+17318604748P2+28P4=1cF1y1TfAeASYLS%2biCPTgu5GRvNoIbwY7g1zJUWqR9MK6HnDnx1kCY...
3254	46.304413	192.168.1.189	44.228.249.3	HTTP	531	GET /login.php HTTP/1.1
3341	63.378392	192.168.1.189	44.228.249.3	HTTP	713	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
3353	63.583882	192.168.1.189	44.228.249.3	HTTP	572	GET /login.php HTTP/1.1
3406	75.720608	192.168.1.189	44.228.249.3	HTTP	701	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)

Hình 33. Lọc giao thức web

# Network Forensics

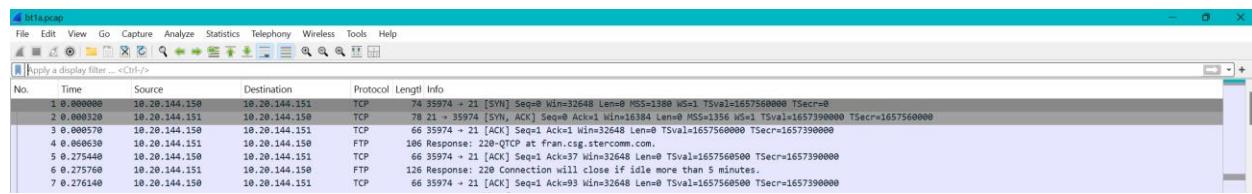
Ta có thể thấy hai giao thức post ta sẽ thử ở đó để xem có mật khẩu không



Hình 34. Tìm thấy mật khẩu từ web

Ta có thể thấy phần tài khoản và pass ở form item mà bạn tôi đã nhập thê là tôi đã kiểm được tài khoản cho anh ấy rồi

**Demo 2:** Trong một lần sử dụng máy tính của mình, bạn phát hiện có một người khác đang sử dụng thiết bị của bạn và thực hiện việc đăng nhập vào một nơi nào đó. Sau đó, bạn quyết định sử dụng công cụ bắt gói tin mạng (như Wireshark) để ghi lại lưu lượng mạng khi người đó đang truy cập trang web. Sau khi thu thập được file pcap chứa các gói tin mạng. Nhờ vào việc phân tích các gói tin, bạn có thể xác định được thông tin đăng nhập của người đó mà không cần phải có mật khẩu ban đầu.



Hình 35. Nghi ngờ giao thức FTP có thể có tk và mk

## Network Forensics

Ta có thể thấy ở đây có nhiều giao thức FTP ta sẽ thử lọc các request xem có thể lấy được mật khẩu và password không

8 4.216600	10.20.144.150	10.20.144.151	FTP	81 Request: USER cdt53500
9 4.217350	10.20.144.151	10.20.144.150	FTP	91 Response: 331 Enter password.
11 7.639420	10.20.144.150	10.20.144.151	FTP	81 Request: PASS cdt53500
13 8.184000	10.20.144.151	10.20.144.150	FTP	95 Response: 230 CDT53500 logged on.

Hình 36. Tìm theo luồng sẽ thấy tk và mk

Ta đã có thể thấy được mật khẩu và pass nhập vô một cách dễ dàng do giao thức FTP sẽ không mã hoá tài khoản, mật khẩu nhập vô

**Demo 3:** Trong một lần sử dụng máy tính của mình, bạn phát hiện một người khác đã truy cập vào máy tính của bạn và sử dụng dịch vụ Telnet để kết nối tới một máy chủ từ xa. Sau khi sự việc xảy ra, bạn quyết định sử dụng công cụ bắt gói tin để ghi lại lưu lượng mạng trong quá trình kết nối Telnet. Sau khi phân tích file pcap mà bạn thu thập được, bạn phát hiện ra các gói tin Telnet chứa thông tin đăng nhập, bao gồm tên tài khoản và mật khẩu của người dùng khi kết nối đến máy chủ từ xa. Với việc phân tích các gói tin này, bạn có thể phục hồi thông tin đăng nhập mà người đó đã sử dụng trong phiên Telnet.

Ta sẽ bấm vào follow để xem luồng TCP của phương thức telnet đó:

The screenshot shows a NetworkMiner capture of a Telnet session. The session starts with a login prompt: "login:". Below it, several carriage returns and line feeds are visible. The user then enters the username "ffaakkee" followed by a carriage return. A password prompt "Password:" follows, and the user enters the password "user". Both the username and password are displayed in red text.

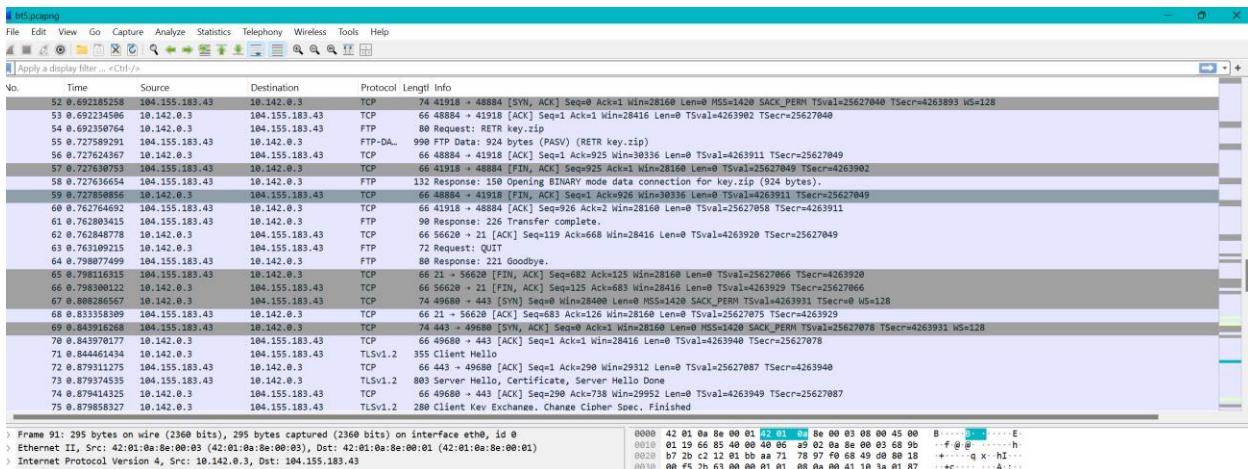
```
login:  
....  
f  
f  
a  
a  
k  
k  
e  
e  
. .  
Password:  
user  
. .
```

Hình 37. Kết quả tài khoản mật khẩu của phương thức telnet

Ta có thể thấy được tài khoản là ffaakkee và mật khẩu là user

**Demo 4:** Một ngày, bạn tôi gửi cho tôi một tệp ` pcap` và nhờ tôi phân tích vì họ nghi ngờ có sự cố bảo mật. Sau khi mở tệp và kiểm tra các gói dữ liệu, tôi phát hiện một số kết nối bất thường và các gói tin lạ, với nhiều yêu cầu không rõ nguồn gốc và các địa chỉ IP ngoại vi đáng ngờ. Các gói tin này có dấu hiệu của việc truy cập trái phép hoặc có thể là một cuộc tấn công. Tôi tiếp tục phân tích các chi tiết như thời gian truyền tải và các mẫu lưu lượng mạng để xác định xem liệu có vấn đề gì nghiêm trọng hay không.

# Network Forensics



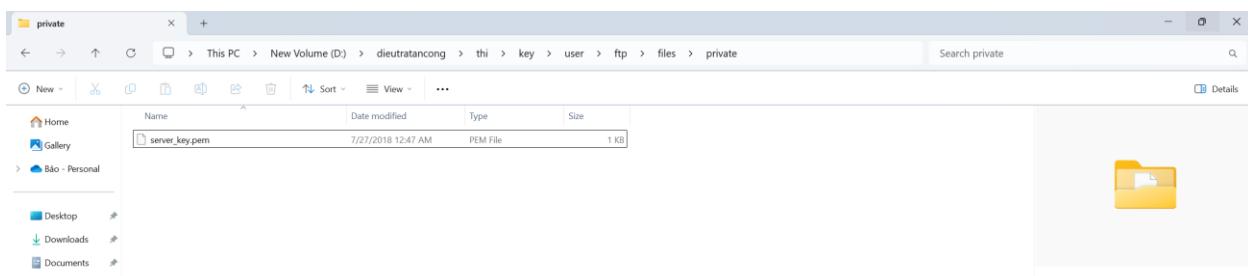
Hình 38. Nghi ngờ các phương thức bị mã hoá

Ta có thể thấy các phương thức tls bị mã hoá đầu tiên ta sẽ thử xem có các nào để decode nó không.



Hình 39. Tìm thấy key giải mã hoá từ FTP

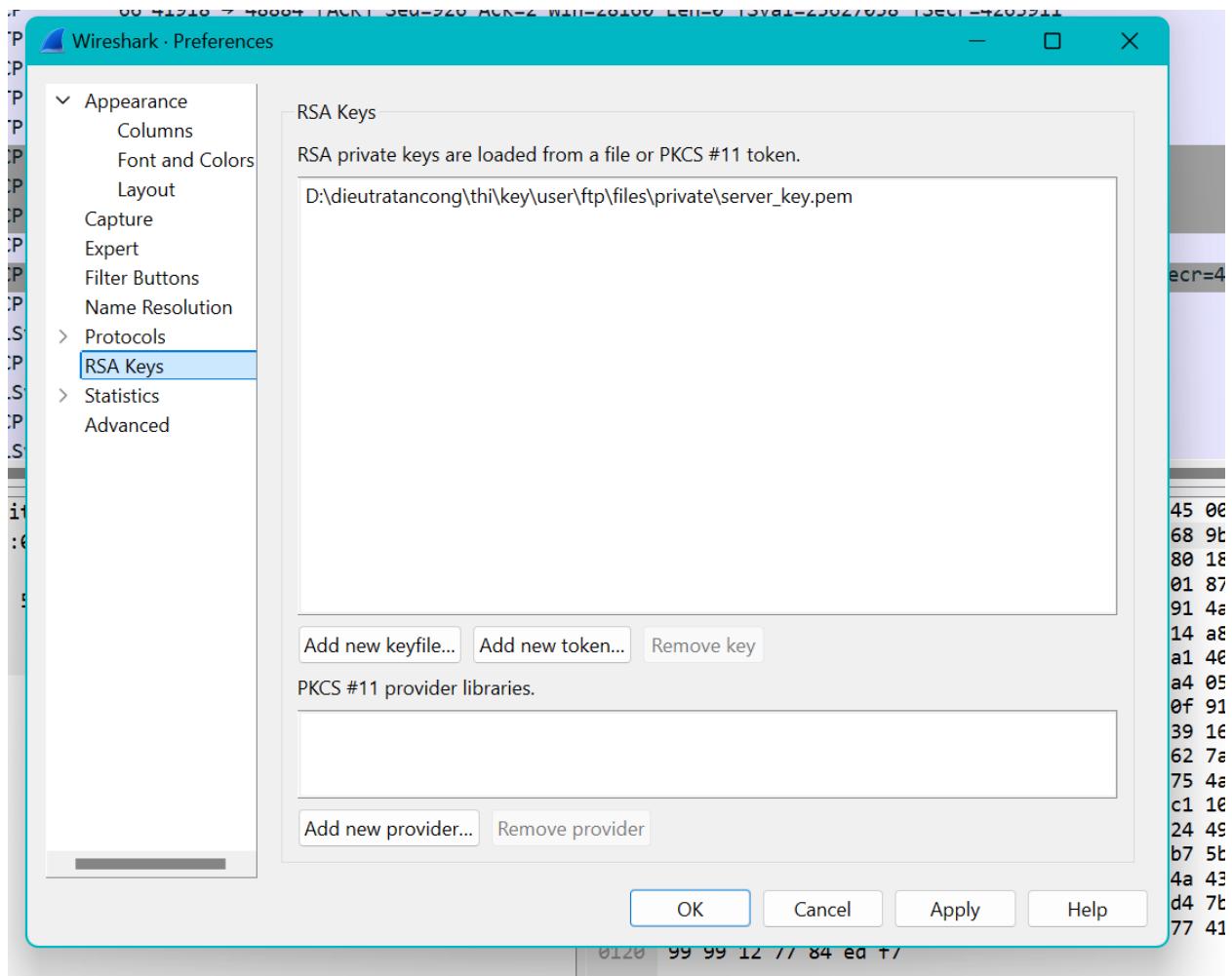
Ta có thể thấy ở đây có một file key.zip ở giao thức ftp ta sẽ thử xem có thể khôi phục được nó không



Hình 40. Khôi phục lại key

Khôi phục xong thì ta có một file như vậy

# Network Forensics



Hình 41. Thêm key giải mã hoá vào wireshark

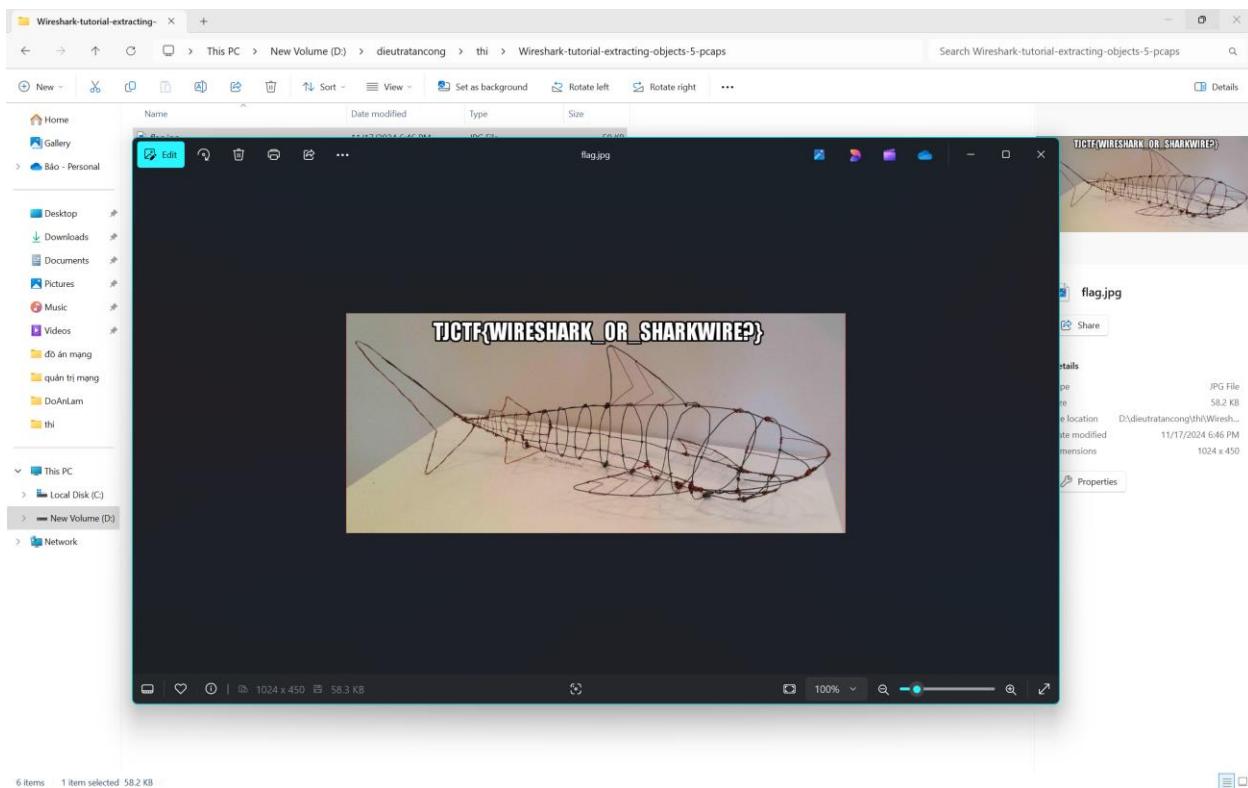
Ta sẽ đến edit -> preferences sau đó import key vào

77 0.915363999 10.142.0.3	104.155.183.43	HTTP 295 GET /index.html HTTP/1.1
78 0.958646268 104.155.183.43	10.142.0.3	TLSv1.2 439
79 0.958677789 104.155.183.43	10.142.0.3	HTTP/1.0 200 ok (text/html)
80 0.959222151 10.142.0.3	104.155.183.43	66 ACK Seq=733 Ack=1362 Wnn=32896 Len=0 TSval=4263987 TSecr=25627185
81 0.959260631 10.142.0.3	104.155.183.43	TCP 74 49682 + 443 [FIN, ACK] Seq=734 Wnn=32896 Len=0 MSSw14285 SACK_PEND TSval=4263987 TSecr=25627185 WS=128
82 0.9884949338 104.155.183.43	10.142.0.3	TCP 66 443 + 49680 [ACK] Seq=1362 Ack=1362 Wnn=31488 Len=0 TSval=25627114 TSecr=4263967
83 0.988532545 104.155.183.43	10.142.0.3	TCP 74 443 + 49682 [SYN, ACK] Seq=0 Ack=1 Wnn=28168 Len=0 MSS=14285 SACK_PEND TSval=25627114 TSecr=4263967 WS=128
84 0.988573982 10.142.0.3	104.155.183.43	TCP 66 49682 + 443 [ACK] Seq=1 Ack=1 Wnn=2816 Len=0 TSval=4263976 TSecr=25627114
85 0.988871512 10.142.0.3	104.155.183.43	TLSv1.2 355 Client Hello
86 1.023938550 104.155.183.43	10.142.0.3	TCP 66 443 + 49682 [ACK] Seq=1 Ack=290 Wnn=29312 Len=0 TSval=25627123 TSecr=4263976
87 1.024108936 104.155.183.43	10.142.0.3	TLSv1.2 803 Server Hello, Certificate, Server Hello Done
88 1.024156381 10.142.0.3	104.155.183.43	TCP 66 49682 + 443 [ACK] Seq=290 Ack=738 Wnn=29952 Len=0 TSval=4263985 TSecr=25627123
89 1.024485889 10.142.0.3	104.155.183.43	TLSv1.2 280 Client Key Exchange, Change Cipher Spec, Finished
90 1.060383213 104.155.183.43	10.142.0.3	TLSv1.2 316 New Session Ticket, Change Cipher Spec, Finished
91 1.060622563 10.142.0.3	104.155.183.43	HTTP 295 GET /flag.jpg HTTP/1.1
92 1.096409386 104.155.183.43	10.142.0.3	TCP 8514 443 + 49682 [ACK] Seq=988 Ack=733 Wnn=31488 Len=8448 TSval=25627141 TSecr=4263994 [TCP PDU reassembled in 96]

Hình 42. Decode giao thức bị mã hoá

Decode một tlsv1.2 thì ta thấy ở đây có một file flag.jpg ta sẽ thử khôi phục nó xem coi nó là gì

## Network Forensics



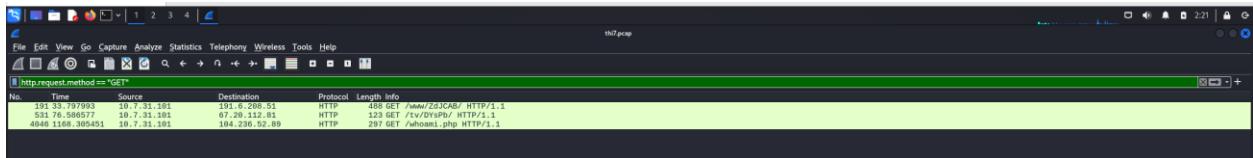
Hình 43. Khôi phục lại được dữ liệu

Ta đã tìm kiếm được điều bí ẩn trong file này.

## Demo 5:

Ta kiểm tra thấy ở đây đã có một giao thức http get lấy một file gì đó về ta sẽ thử lọc gói theo ip và phương thức để xem có file gì được tải về

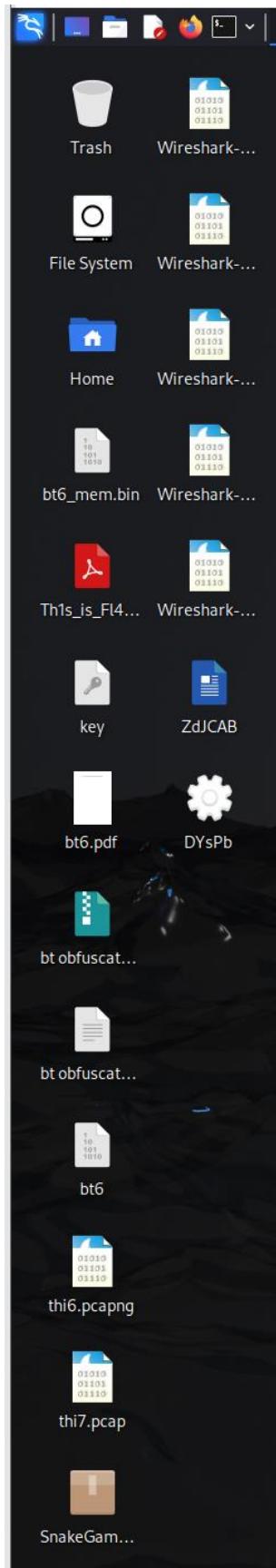
## Network Forensics



Hình 44. Lọc gói

Ta có thể thấy ở đây có 3 file được get trong đó 2 file đầu không có định dạng cụ thể ta sẽ thử khôi phục lại

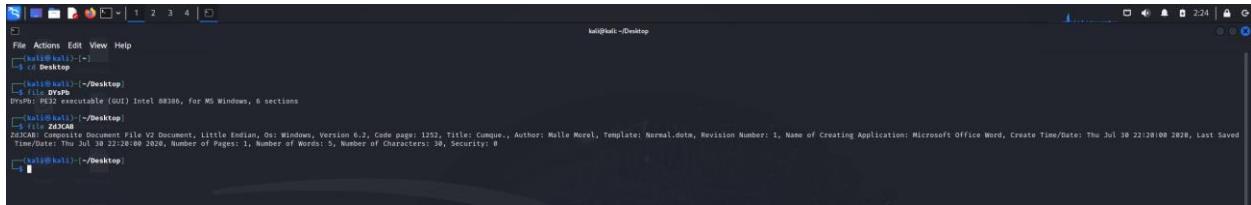
## Network Forensics



# Network Forensics

Hình 45. Khôi phục file

Ta đã khôi phục lại được 2 file và ta sẽ kiểm tra xem nó là file gì

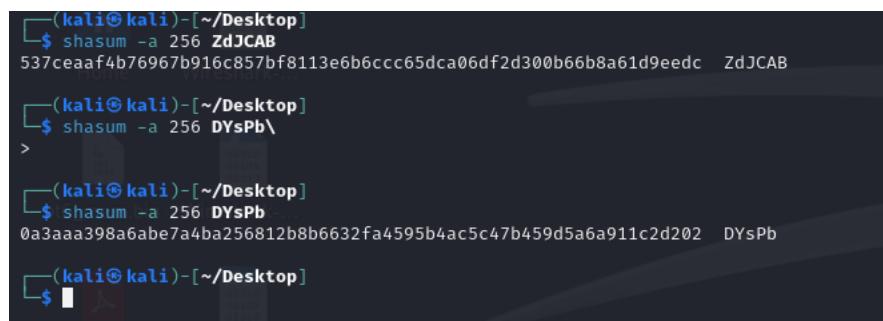


A screenshot of a terminal window titled 'kali@kali: ~/Desktop'. It shows the output of several commands:

- `file D\ysPb` outputs: PE32 executable (GUI) Intel 80386, for MS Windows, 6 sections
- `xdcab -i D\ysPb` outputs: ZDxCAB Composite Document File V2 Document, Little Endian, Os: Windows, Version 6.2, Code pages: 1252, Title: Comque., Author: Halle Morel, Template: Normal.dotm, Revision Number: 1, Name of Creating Application: Microsoft Office Word, Create Time/Date: Thu Jul 30 22:28:00 2020, Last Saved Time/Date: Thu Jul 30 22:28:00 2020, Number of Pages: 5, Number of Words: 30, Number of Characters: 30, Security: 0
- `file D\zJcAB` outputs: Microsoft Word document, DocumentFormat.OpenXml format

Hình 46. Ta có một file doc và một file exe

Ta sẽ kiểm tra nó xem coi hai file đó thế nào



A screenshot of a terminal window titled '(kali㉿kali)-[~/Desktop]'. It shows the output of three 'shasum -a 256' commands:

- For 'D\ysPb': 537ceaaaf4b76967b916c857bf8113e6b6ccc65dca06df2d300b66b8a61d9eedc ZdJCAB
- For 'D\zJcAB': 0a3aaa398a6abe7a4ba256812b8b6632fa4595b4ac5c47b459d5a6a911c2d202 DYsPb
- For the prompt '>': >

Hình 47. ta checksum hai file theo sha-256

Kiểm tra mã trên virustotal

# Network Forensics

49 / 63 security vendors flagged this file as malicious

537ceaa4b76967b916c857bf8113e6b6ccc65dca06df2d300b6b8a61d9eedc  
INVOICE-OR85-923315.doc

Community Score -1

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 11

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Code insights

The provided macros do not exhibit any malicious behavior. Here's a detailed analysis of each macro:

LMNOimyyukrkyj.cls:  
This class module defines a class named "LMNOimyyukrkyj" with no apparent functionality. It simply declares the class and its attributes.

Show more

Crowdsourced AI

Hispec flags this file as malicious

The macros extracted from the document exhibit several behaviors and characteristics commonly associated with malicious intent. Below is a detailed breakdown of these indicators:

Show more

Popular threat label: downloader.w37m/emotet

Threat categories: downloader, trojan

Family labels: w37m, emotet, lemmodir

Security vendors' analysis

Do you want to automate checks?			
Acronis (Static ML)	Suspicious	AhnLab-V3	Downloader/MSOffice.Generic
AliCloud	Trojan[downloader]:MSOffice/Emotet.F...	ALYac	TrojanDownloader.DOC.Gen
Anti-AVL	Trojan/Downloader/MSOffice/Emotet.v...	Arcabit	W32.Agent.Emai!Mail!Gen

Hình 48. file doc

65 / 72 security vendors flagged this file as malicious

0a3aaa398a6abe7a4ba256812b8b6632fa4595b4ac5c47b459d5a6a911c2d202  
TabDrives.EXE

Community Score -1

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 4

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: trojan.emotet/euig

Threat categories: trojan, banker

Family labels: emotet, euig, hotben

Security vendors' analysis

Do you want to automate checks?			
AhnLab-V3	Trojan/Win32.Emotet.R346576	Alibaba	Trojan/Win32/Emotet.9dc2ff5
AliCloud	Trojan[stealer]:Win/Emotet.ARJIMTB	ALYac	Trojan.Agent.Emotet
Anti-AVL	Trojan[Banker]:Win32.Emotet	Arcabit	Trojan.Agent.EUIG
Avast	Win32:Trojan-gen	AVG	Win32:Trojan-gen
Avira (no cloud)	TR/Crypt.Agent.hottbm	BitDefender	Trojan.Agent.EUIG
Bkav Pro	W32.AIDetectMalware	ClamAV	Win.Packed.Emotet-9527874-0
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	CTX	Exe.trojan.emotet
Cylance	Unsafe	Cynet	Malicious (score: 99)
DeepInstinct	MALICIOUS	DrWeb	Trojan.Emotet.994
Elastic	Malicious (high Confidence)	Emsisoft	Trojan.Emotet (A)

Hình 49. file exe

Ta có thể thấy rằng cả 2 file được lấy về máy đều là hai con virus trojan

## **Bảng Phân Công**

*Bảng 1. Bảng phân công*

MSSV	Họ và tên	Công Việc	Độ Hoàn Thành
22DH110298	Phạm Hoàng Gia Bảo	Viết báo cáo, tìm hiểu lý thuyết, Demo	100%
22DH110711	Lê Thành Đạt	Tìm hiểu lý thuyết, các kĩ thuật wireshark	100%

### Tài Liệu Tham Khảo

- [Unit42-Wireshark-tutorials/Wireshark-tutorial-filter-expressions-5-pcaps.zip at main · PaloAltoNetworks/Unit42-Wireshark-tutorials · GitHub](#)
- [Wireshark Tutorial: Exporting Objects From a Pcap](#)
- [Wireshark Tutorial: Display Filter Expressions](#)
- [Wireshark Tutorial: Decrypting HTTPS Traffic \(Includes SSL and TLS\)](#)
- [Đánh hơi thông tin đăng nhập hoặc thu thập mật khẩu trong Wireshark - GeeksforGeeks](#)
- [bai tap net-for - Google Drive](#)
- [ebook - Google Drive](#)