**VNCERT**

*Đà Nẵng, ngày 26/03/2023*

**Mục Lục**

[**1.** **WIRESHRK** 4](#_Toc130754435)

[1.1 Wireshark là gì? 4](#_Toc130754436)

[1.2 Wireshark hoạt động như thế nào? 5](#_Toc130754437)

[1.3 Chức năng của Wireshark là gì? 6](#_Toc130754438)

[1.4 Hướng dẫn sử dụng Wireshark 7](#_Toc130754439)

[1.4.1 Sử dụng Wireshark để bắt gói tin 7](#_Toc130754440)

[1.4.2 Giao diện WireShark 8](#_Toc130754441)

[1.5 Phân tích gói tin với Wireshark 10](#_Toc130754442)

[1.5.1 Tìm kiếm gói tin (Find Packet) 10](#_Toc130754443)

[1.5.2 Wireshark Filter 11](#_Toc130754444)

[1.5.3 Display Filter 15](#_Toc130754445)

[1.5.4 Cách đọc gói tin trong Wireshark 20](#_Toc130754446)

**Danh Mục Hình Ảnh**

[Hình 1. 1 WireShark là gì? 4](#_Toc130754576)

[Hình 1. 2 Được dùng nhiều ở các cơ quan chính phủ, tập đoàn, tổ chức phi lợi nhuận và tổ chức giáo dục 5](#_Toc130754577)

[Hình 1. 3 Bắt các lưu lượng mạng kết nối Ethernet, Wireless, Bluetooth,.. 5](#_Toc130754578)

[Hình 1. 4 Chức năng và các tính năng nổi bật 6](#_Toc130754579)

[Hình 1. 5 Bắt gói tin mạng wifi 7](#_Toc130754580)

[Hình 1. 6 Dừng bắt gói tin 8](#_Toc130754581)

[Hình 1. 7 Giao diện gồm: Packet List, Packet Detail, Packet Byte 9](#_Toc130754582)

[Hình 1. 8 Mở gói tin 9](#_Toc130754583)

[Hình 1. 9 Tìm kiếm gói tin 10](#_Toc130754584)

[Hình 1. 10 Cú pháp tổng quan để bắt gói tin 13](#_Toc130754585)

[Hình 1. 11 Nhập expression vào Filter textbox 15](#_Toc130754586)

[Hình 1. 12 Ví dụ Display Filter 18](#_Toc130754587)

[Hình 1. 13 Chọn Filter mặc định trong WireShark 19](#_Toc130754588)

[Hình 1. 14 Xem đầy đủ đoạn hội thoại trong Display Filter 19](#_Toc130754589)

[Hình 1. 15 Hiển thị các packet trong đoạn hội thoại 20](#_Toc130754590)

[Hình 1. 16 Phân tích chi tiết gói tin 20](#_Toc130754591)

[Hình 1. 17 Phân tích gói tin bằng Wireshark 22](#_Toc130754592)

[Hình 1. 18 Tạo filter 22](#_Toc130754593)

**Danh Mục Bảng Biểu**

[Bảng 1. 1 Options tìm kiếm packet 11](#_Toc130754706)

[Bảng 1. 2 Một số qualifiers 12](#_Toc130754707)

[Bảng 1. 3 Một vài Wireshark Expression cho phần Capture Filter 15](#_Toc130754708)

[Bảng 1. 4 Comparison Operators 16](#_Toc130754709)

[Bảng 1. 5 Logical Operators 16](#_Toc130754710)

[Bảng 1. 6 Wireshark Expression 18](#_Toc130754711)

# **WIRESHRK**

## Wireshark là gì?

**Wireshark**là một ứng dụng dùng để bắt (capture), phân tích và xác định các vấn đề liên quan đến network như: rớt gói tin, kết nối chậm, hoặc các truy cập bất thường. Phần mềm này cho phép quản trị viên hiểu sâu hơn các Network Packets đang chạy trên hệ thống, qua đó dễ dàng xác định các nguyên nhân chính xác gây ra lỗi.



Hình 1. WireShark là gì?

Sử dụng **WireShark**có thể capture các packet trong thời gian thực (realtime), lưu trữ chúng lại và phân tích chúng offline. Ngoài ra, nó cũng bao gồm các filter, color coding và nhiều tính năng khác, cho phép người dùng tìm hiểu sâu hơn về lưu lượng mạng cũng như inspect (kiểm tra) các packets.

Ứng dụng được viết bằng ngôn ngữ C và hệ điều hành Cross-platform, ngoài ra hiện này gồm có các bản phân phối Linux, Windows, OS X, FreeBSD, NetBSD và OpenBSD. Đây là một phần mềm mã nguồn mở, được cấp phép GPL, và do đó miễn phí sử dụng, tự do chia sẻ và sửa đổi.

Đây là phần mềm hoàn toàn an toàn để sử dụng. Nhiều cơ quan chính phủ, tập đoàn, tổ chức phi lợi nhuận và tổ chức giáo dục đều **sử dụng Wireshark** để khắc phục các sự cố, cũng như ứng dụng vào việc giảng dạy. Có thể nói, cách tốt nhất để tìm hiểu về mạng chính là sử dụng để theo dõi các lưu lượng truy cập.

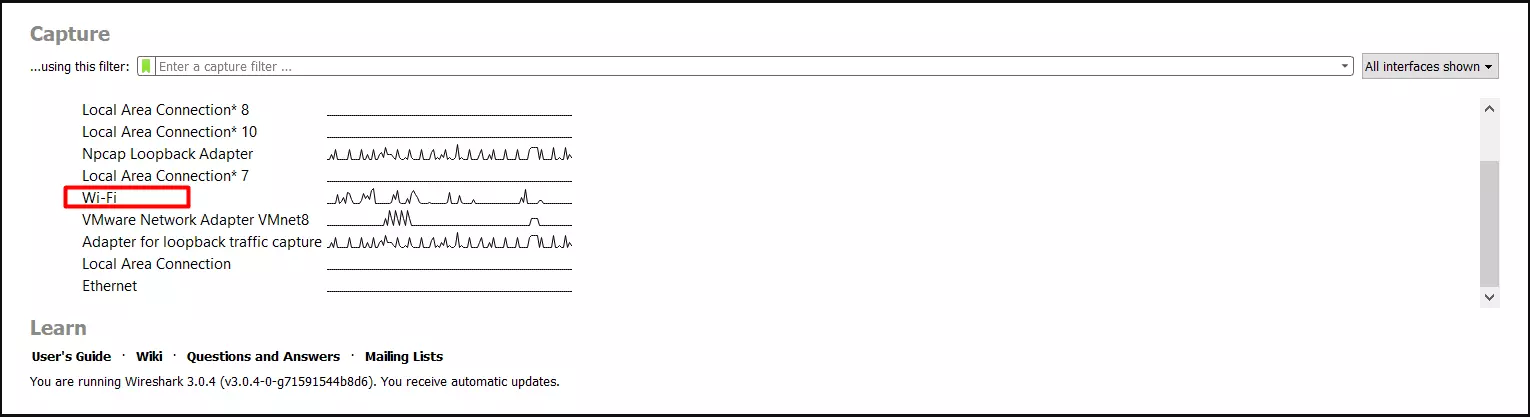


Hình 1. Được dùng nhiều ở các cơ quan chính phủ, tập đoàn, tổ chức phi lợi nhuận và tổ chức giáo dục

Dĩ nhiên, đây là một công cụ có khả năng dò tìm dữ liệu trong các packet rất mạnh mẽ. Do đó cũng có nhiều câu hỏi được đặt ra xoay quanh tính hợp pháp của nó. Nhiều người dùng cũng như chuyên gia đã nhấn mạnh rằng, chỉ nên sử dụng trên các mạng mà mình có thẩm quyền. Việc dùng Wireshark để xem các packet không được cho phép hoàn toàn không được khuyến khích.

## ****Wireshark hoạt động như thế nào?****

Như đã đề cập ở trên,đây là một công cụ dùng để capture và phân tích các packet. Nó capture các lưu lượng mạng trên mạng cục bộ, sau đó sẽ lưu trữ nó để phân tích offline. Có thể capture các lưu lượng mạng từ các kết nối Ethernet, Bluetooth, Wireless (IEEE.802.11), Token Ring, Frame Relay…

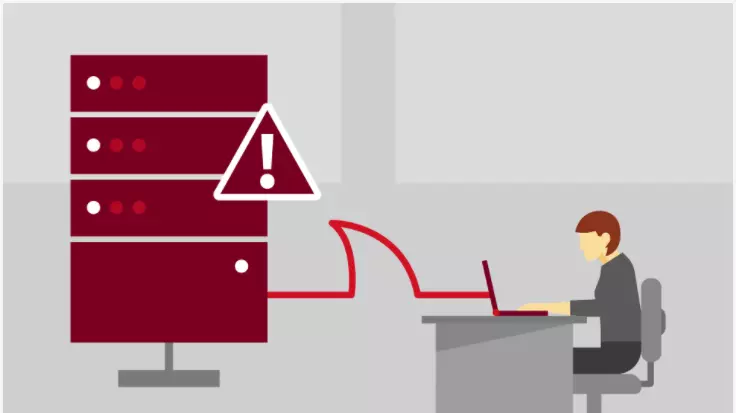


Hình 1. Bắt các lưu lượng mạng kết nối Ethernet, Wireless, Bluetooth,..

Wireshark cho phép thiết lập filter (bộ lọc) trước khi bắt đầu capture hoặc thậm chí là trong quá trình phân tích. Do đó, ta có thể thu hẹp phạm vi tìm kiếm trong quá trình theo dõi mạng.

## Chức năng của Wireshark là gì?

Các dữ liệu capture ở dạng binary được chuyển thành định dạng giúp bạn dễ dàng đọc được. Với hơn 2000 giao thức mạng được hỗ trợ, điều này giúp xác định lưu lượng truy cập đang đi qua hệ thống mạng của bạn.



Hình 1. Chức năng và các tính năng nổi bật

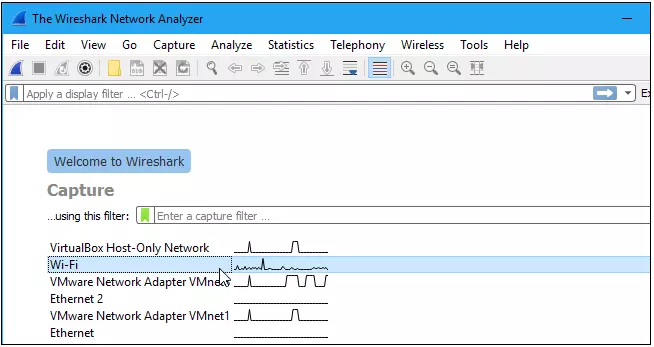
Với khối lượng lưu lượng khổng lồ truyền hệ thống mạng, các công cụ của Wireshark sẽ giúp lọc các lưu lượng đó ra để phân tích chúng. Bộ capture filter sẽ chỉ thu thập các loại lưu lượng truy cập mà bạn quan tâm. Sau đó, hiển thị chi tiết lưu lượng mà bạn muốn kiểm tra. Module phân tích giao thức mạng cũng cung cấp các công cụ tìm kiếm, các biểu thức và tô màu, giúp dễ phân tích hơn.

* **Các tính năng nổi bật của phần mềm bắt gói tin Wireshark:**
* Hỗ trợ phân tích sâu hàng trăm giao thức và liên tục được cập nhật.
* Live capture và phân tích offline.
* Hoạt động đa nền tảng: Windows, Linux, MacOS, Solaris, FreeBSD, OpenBSD…
* Các gói tin đã capture có thể xem bằng giao diện hoặc sử dụng command line (tshark).
* Display filter mạnh mẽ.
* Hỗ trợ phân tích VoIP chuyên sâu.
* Hỗ trợ read/write nhiều định dạng: tcpdump (libpcap), Pcap NG, Catapult DCT2000, Cisco Secure IDS iplog, Microsoft Network Monitor, Network General Sniffer® (compressed and uncompressed), Sniffer® Pro, and NetXray®, Network Instruments Observer, NetScreen snoop, Novell LANalyzer, RADCOM WAN/LAN Analyzer, Shomiti/Finisar Surveyor, Tektronix K12xx, Visual Networks Visual UpTime, WildPackets EtherPeek/TokenPeek/AiroPeek …
* File capture được nén bằng gzip có thể được giải nén “on the fly”.
* Capture dữ liệu từ [Ethernet](https://vietnix.vn/ethernet-la-gi/), IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI …
* Hỗ trợ decryption của nhiều giao thức như: IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, and WPA/WPA2.
* Coloring rules cho phép thiết lập màu sắc cho các packet giúp phân tích nhanh và hiệu quả hơn.
* Output có thể export sang XML, PostScript®, CSV, hoặc plain text.

## ****Hướng dẫn sử dụng Wireshark****

### Sử dụng Wireshark để bắt gói tin

Sau khi download và cài đặt, ta có thể khởi động nó bằng cách double-click vào tên của [Network interface](https://vietnix.vn/card-mang-la-gi/) trong danh sách phía dưới “Capture” để bắt đầu bắt gói tin trên card mạng đó. Đường biểu diễn phía sau tên Interface thể hiện lưu lượng mạng đang sử dụng.



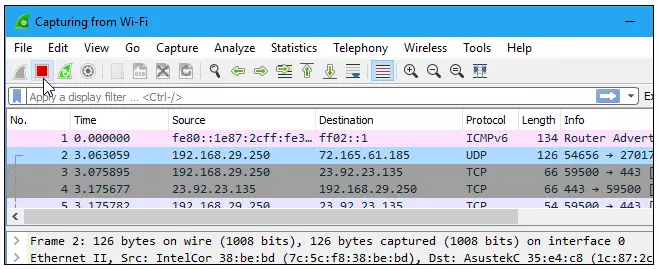
Hình 1. Bắt gói tin mạng wifi

Sau đó, các packet sẽ bắt đầu hiển thị theo thời gian thực. Wireshark sẽ capture từng packet được gửi đến hoặc đi từ hệ thống của ta.

Nếu chế độ Promiscuous được enable (theo mặc định), ta cũng có thể xem tất cả các packet khác trên mạng thay vì chỉ các packet được gửi đến network adapter của mình.

Để kiểm tra chế độ Promiscuous, click vào Capture > Options và kiểm tra xem hộp “Enable promiscuous mode on all interfaces” có được kích hoạt chưa (nằm ở dưới cùng của cửa sổ).

Click vào nút **“Stop”** màu đỏ (ở góc trên bên trái của cửa sổ – hoặc chọn **“Capture > Stop”**) nếu muốn dừng việc capture lại.

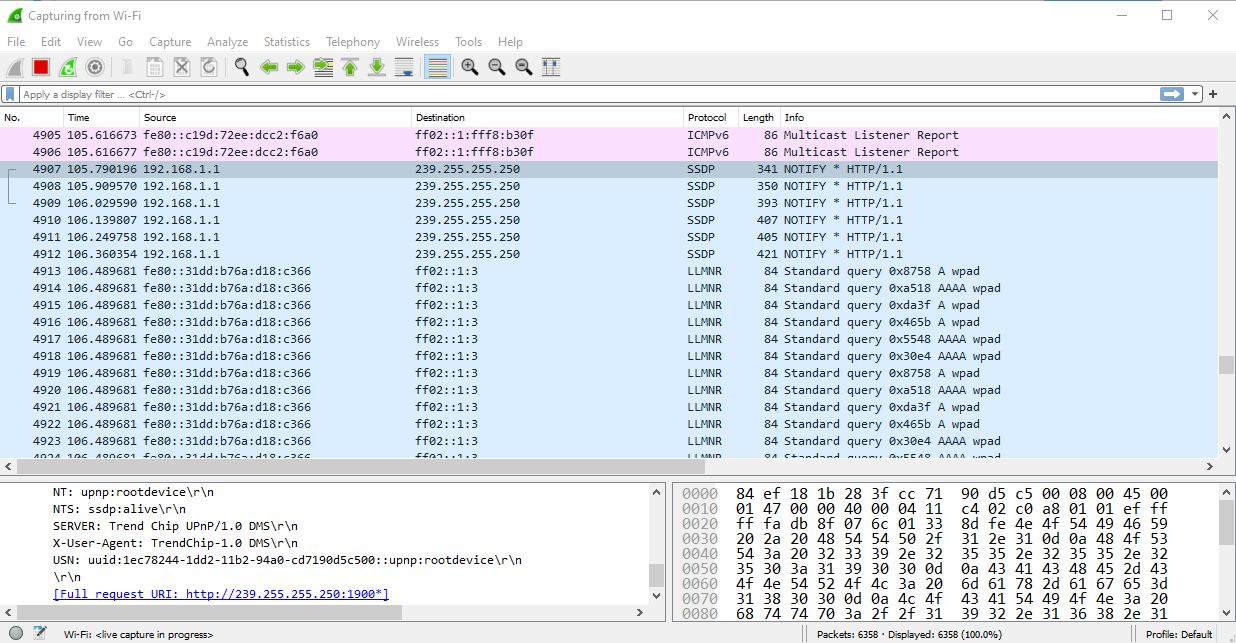


Hình 1. Dừng bắt gói tin

Ngoài cách bắt gói tin và sử dụng giao diện như trên, bạn cũng có thể dùng cách bắt gói tin bằng cách sử dụng command line được đề cập ở phần nâng cao phía dưới bài viết.

### **Giao diện WireShark**

Bạn sẽ dành rất nhiều thời gian để thao tác trên giao diện chính của phần mềm này. Đây là nơi liệt kê danh sách các packet đã được capture, parse và thể hiện dưới định dạng mà chúng ta có thể dễ dàng đọc thông tin và phân tích chúng.



Hình 1. Giao diện gồm: Packet List, Packet Detail, Packet Byte

* **Giao diện chính của Wireshark được chia thành 3 phần:**

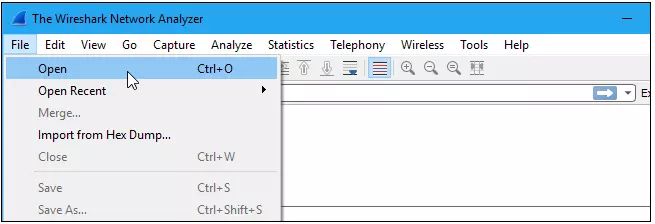
**Packet List**: Chứa danh sách toàn bộ packet của file capture hiện tại. Nó thể hiện số thứ tự của gói tin, thời gian mà mà gói tin được bắt, source và destination IP, protocol của packet, chiều dài gói tin và các thông tin tổng quan khác.

**Packet Details**: Khi bạn chọn một gói tin ở phần Packet List, thông tin chi tiết của gói tin sẽ được thể hiện ở phần Packet Detail. Các thông tin chi tiết có thể được collapsed hoặc expanded bằng cách click vào mũi tên hình tam giác ở đầu dòng.

**Packet Bytes**: Thể hiện packet ở định dạng raw dưới dạng hex hoặc binary. Thể hiện cách mà packet được truyền trên đường truyền.

**Mở gói tin và lưu gói tin**

Để **mở gói tin bằng Wireshark**, chọn “**File** > **Open**” và tìm đến đường dẫn của file cần mở.



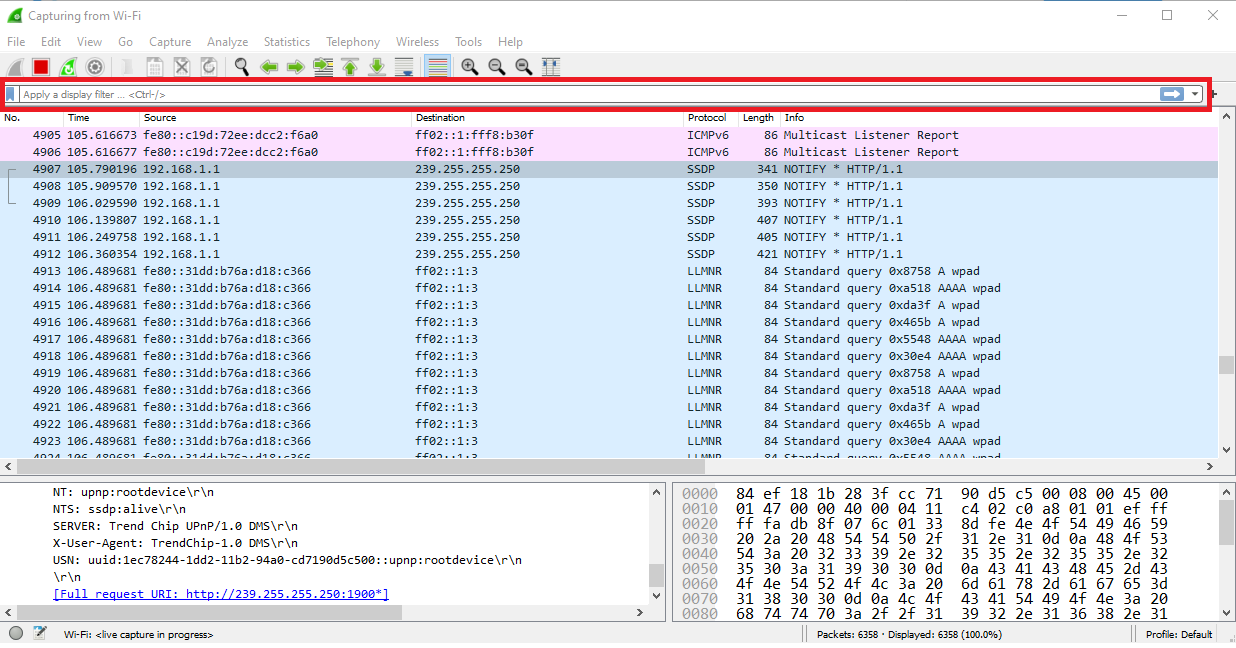
Hình 1. Mở gói tin

Để lưu gói tin đã capture, click vào **“File > Save”**, sau đó chọn dường dẫn để lưu trữ, đặt tên cho file capture và định dạng sẽ lưu.

## Phân tích gói tin với Wireshark

### Tìm kiếm gói tin (Find Packet)

Để tìm kiếm gói tin, chúng ta có thể sử dụng thanh công cụ **“Find Packet”**bằng cách bấm phím **Ctrl + F,** một hộp thoại mới sẽ xuất hiện nằm giữa thanh Filter và Packet List:



Hình 1. Tìm kiếm gói tin

**Chúng ta có thể tìm kiếm packet dựa vào:**

**Display Filter:** nhập vào một biểu thức filter (expression-based filter), Wireshark sẽ tìm kiếm các gói tin khớp với biểu thức này.

**Hex value:** Tìm kiếm dựa trên giá trị Hex.

**String:** tìm kiếm dựa trên chuỗi dữ liệu.

**Regular Expression:** Tìm kiếm dựa trên biểu thức Regex.

|  |  |
| --- | --- |
| **Options** | **Ví dụ** |
| Display filter | tcp.src port==80 hoặc ip.src==192.168.1.1 |
| Hex | 010108ffff |
| String | Quantrilinux.vn hoặc GET / |
| Regular Expression | GET .\* HTTP |

Bảng 1. Options tìm kiếm packet

### Wireshark Filter

Filter cho phép bạn lọc ra những packet nào sẽ dùng để phân tích. Sử dụng **Wireshark filter** bằng cách khai báo một biểu thức để quy định việc thêm vào (inclusion) hoặc loại bỏ (exclusion) các gói tin. Nếu có những gói tin bạn không cần phân tích, có thể viết filter để loại bỏ chúng. Ngược lại, có những gói tin quan trọng bạn muốn phân tích kỹ, có thể viết filter để lọc riêng chúng ra. Có hai loại filter chính:

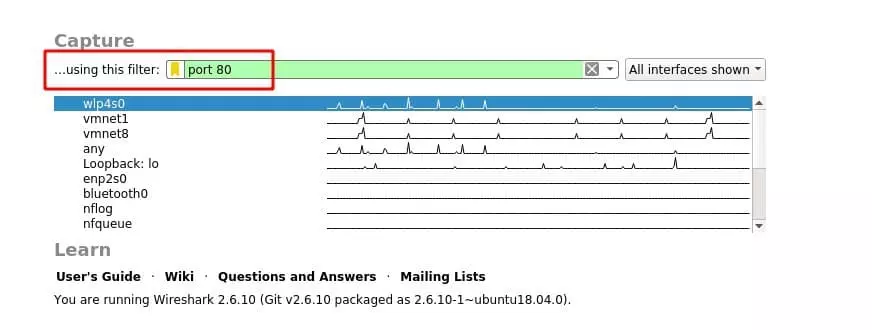
**Capture Filters:** Chỉ định các packet sẽ được capture và quá trình bắt gói tin chỉ capture những packet thỏa điều kiện này.

**Display filters:** Áp dụng filter lên các gói tin đã được capture, mục tiêu là để ẩn đi những packet không cần thiết và chỉ thể hiện những packet thỏa điều kiện chỉ định.**Capture Filter**và **Display Filter** sử dụng cấu trúc ngữ pháp khác nhau nên chúng ta sẽ xem xét chi tiết từng loại.

**Capture Filter**

Được áp dụng trong quá trình bắt gói tin để giới hạn số lượng gói tin sẽ được bắt. Lý do chính để sử dụng filter này nhằm cải thiện performance và giới hạn số lượng dữ liệu capture được chỉ chứa các thông tin chúng ta quan tâm, giúp việc phân tích trở nên hiệu quả hơn. Điều này cực kỳ hữu ích khi áp dụng bắt gói tin bằng Wireshark trên các hệ thống có lưu lượng mạng cao, dữ liệu trao đổi lớn.

Chúng ta có thể khai báo biểu thức cho Capture Filter ở  “**Capture** > **Capture Filters**” hoặc khai báo ở phần “**…using this filter**” khi lựa chọn card mạng:



Hình 1.9 Khai báo Capture Filter

**Wireshark Capture Filter sử dụng cú pháp của Berkeley Packet Filter (BPF):**

Mỗi filter gọi là một expression.

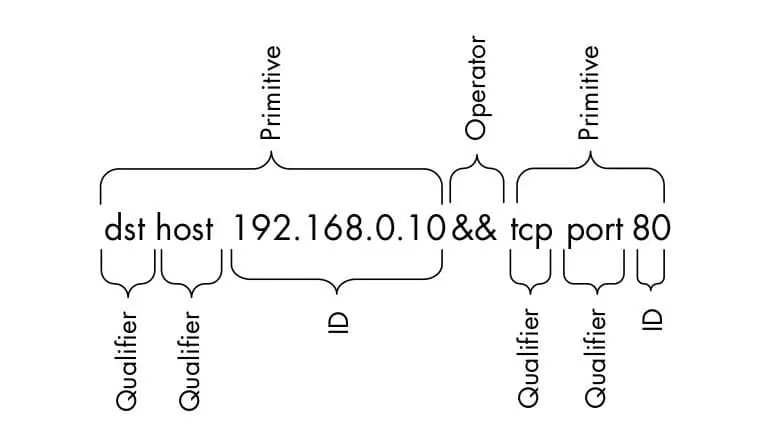
Mỗi expression chứa một hoặc nhiều primitives. Các primitives được kết hợp với nhau bằng các “Logical Operator” như AND (&&), OR (||) và NOT (!) .

Mỗi **primitives**chứa một hoặc nhiều **qualifiers**, theo sau là một ID name hoặc number. Các BPF Qualifiers bao gồm:

|  |  |  |
| --- | --- | --- |
| **Qualifiers** | **Mô tả** | **Ví dụ** |
| **Type** | Chỉ định ID name hoặc number ta sẽ tham chiếu | host, net, port |
| **Dir** | Chỉ định hướng của dữ liệu (transfer direction) | src, dst |
| **Proto** | Protocol | ether, ip, tcp, udp, http, ftp |

Bảng 1. Một số qualifiers

**Cú pháp tổng quan:** Bắt các gói tin gửi đến host 192.168.0.10 và sử dụng giao thức TCP, port 80:



Hình 1. Cú pháp tổng quan để bắt gói tin

**Một vài Wireshark Expression tham khảo cho phần Capture Filter:**

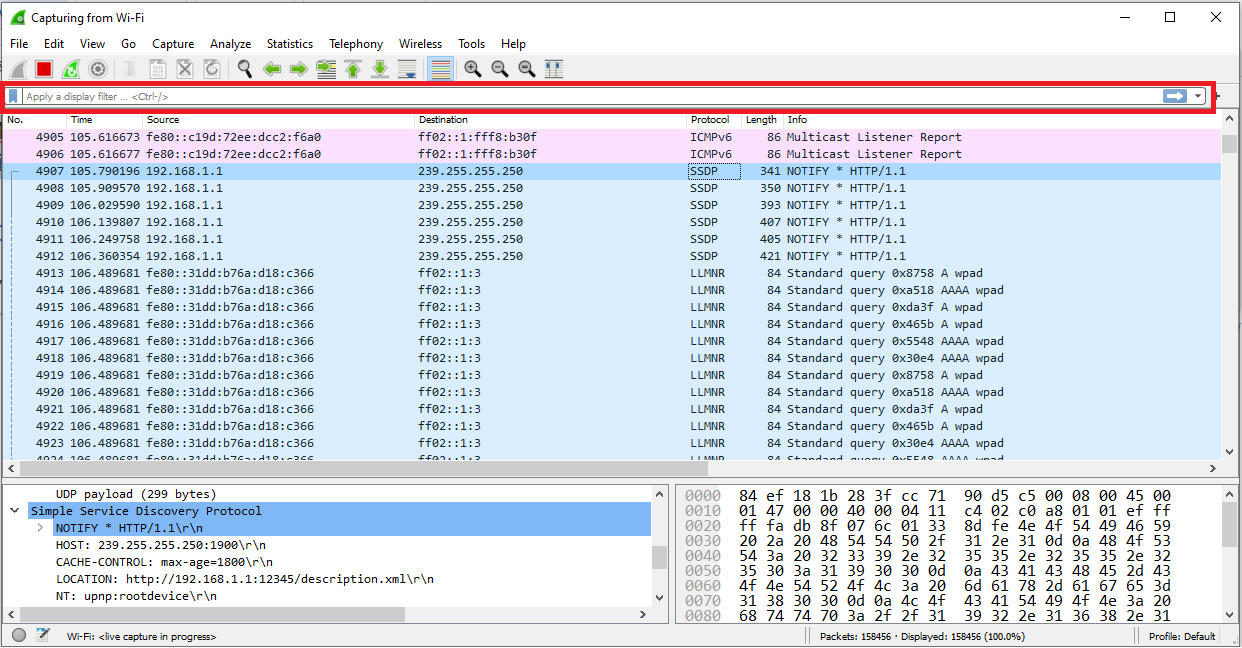
|  |  |
| --- | --- |
| **Expression** | **Ý nghĩa** |
| host 172.18.5.4 | Wireshark filter by IP: Bắt gói tin liên quan đến IP 172.18.5.4 |
| src 192.168.0.10 | Wireshark filter source IP: Bắt các gói tin có source IP là 192.168.0.10 |
| dst 192.168.0.1 | Wireshark filter destination IP: Bắt các gói tin có destination IP là 192.168.0.1 |
| net 192.168.0.0/24 hoặc: net 192.168.0.0 mask 255.255.255.0 | Bắt gói tin liên quan đến subnet 192.168.0.0/24 |
| src net 192.168.0.0/24 hoặc: src net 192.168.0.0 mask 255.255.255.0 | Bắt các gói tin có source IP thuộc subnet 192.168.0.0/24 |
| dst net 192.168.0.0/24 hoặc: dst net 192.168.0.0 mask 255.255.255.0 | Bắt các gói tin có destination IP thuộc subnet 192.168.0.0/24 |
| port 53 | Wireshark port filter: Bắt gói tin DNS |
| port 67 or port 68 | Bắt gói tin DHCP |
| host 192.168.1.1 and not (port 80 or 443) hoặc: host 192.168.1.1 and not port 80 and not port 443 | Capture tất cả traffic liên quan đến IP 192.168.1.1 nhưng không phải traffic HTTP/HTTPS |
| (tcp[0:2] > 1500 and tcp[0:2] < 1550) or (tcp[2:2] > 1500 and tcp[2:2] < 1550) hoặc: tcp portrange 1501-1549 | Capture các packet nằm trong range port từ 1501-1549 |
| ip | Wireshark IPv4 filter |
| ip6 | Wireshark IPv6 filter |
| tcp | Bắt gói tin TCP |
| udp | Bắt gói tin UDP |
| icmp | Bắt gói tin ICMP |
| http | Wireshark HTTP filter |
| https | Wireshark HTTPS filter |
| tcp[13] & 32 == 32 | TCP packets với cờ URG được bật |
| tcp[13] & 16 == 16 | TCP packets với cờ ACK được bật |
| tcp[13] & 8 == 8 | TCP packets với cờ PSH được bật |
| tcp[13] & 4 == 4 | TCP packets với cờ RST được bật |
| tcp[13] & 2 == 2 | TCP packets với cờ SYN được bật |
| tcp[13] & 1 == 1 | TCP packets với cờ FIN được bật |
| icmp[0:2] == 0x0301 | ICMP destination unreachable, host unreachable |

Bảng 1. Một vài Wireshark Expression cho phần Capture Filter

### Display Filter

Display Filter giúp lọc ra những packet thỏa điều kiện trong file capture để thể hiện lên cho người dùng. Display filter chỉ lọc và thể hiện packet thỏa điều kiện chứ không xóa bỏ những packet không thỏa điều kiện, dữ liệu trong file capture hoàn toàn không bị ảnh hưởng.

Sử dụng Display Filter bằng cách nhập biểu thức (expression) vào Filter textbox phía trên phần Packet List. Bạn cũng có thể nhấp vào phần “Expression” để lựa chọn các pre-defined filters có sẵn ứng với từng giao thức.



Hình 1. Nhập expression vào Filter textbox

**Cú pháp của Wireshark Display Filter phần lớn tuân theo cú pháp:**

*protocol.feature.subfeature COMPARISION\_OPERATOR value LOGICAL\_OPERATOR protocol.feature.subfeature COMPARISION\_OPERATOR value*

Trong đó, **Comparison Operators** bao gồm:

|  |  |
| --- | --- |
| **Operator** | **Ý nghĩa** |
| == | Bằng (equal to) |
| != | Không bằng (not equal to) |
| > | Lớn hơn (greater than) |
| < | Nhỏ hơn |
| >= | Lớn hơn hoặc bằng (greater than or equal) |
| <= | Nhỏ hơn hoặc bằng (less than or equal) |

Bảng 1. Comparison Operators

**Logical Operators**bao gồm:

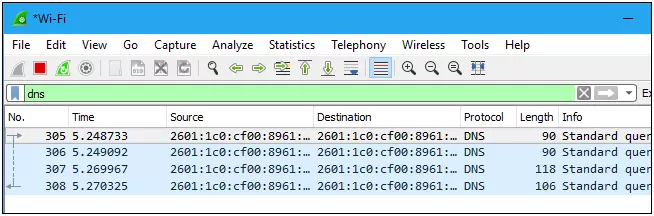
|  |  |
| --- | --- |
| **Operator** | **Ý nghĩa** |
| and | tất cả các điều kiện phải được thỏa mãn |
| or | một trong các điều kiện được thoả mãn |
| xor | một và chỉ một điều kiện được thỏa mãn |
| not | không điều kiện nào được phép thảo mãn |

Bảng 1. Logical Operators

**Một vài Wireshark Expression tham khảo cho phần Display Filter:**

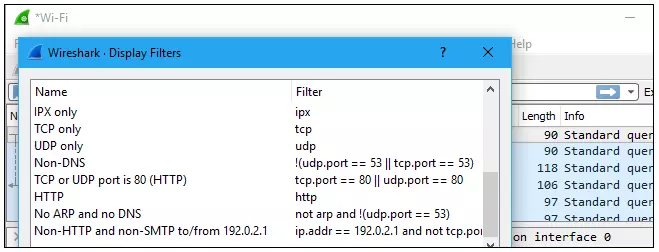
|  |  |
| --- | --- |
| **Expression** | **Ý nghĩa** |
| tcp.port eq 25 or icmp | Lọc gói tin TCP liên quan port 25 hoặc sử dụng giao thức ICMP |
| ip.src==192.168.0.0/16 and ip.dst==192.168.0.0/16 | Lọc traffic trao đổi trong mạng LAN của subnet 192.168.0.0/16 |
| tcp.window\_size == 0 && tcp.flags.reset != 1 | TCP buffer full và source kết nối báo hiệu cho Destination ngừng gửi dữ liệu |
| udp contains 81:60:03 | UDP packet chứa 3 bytes 81:60:03 ở vị trí bất kỳ trong header hoặc payload |
| http.request.uri matches “gl=se$” | HTTP request có URL tận cùng bằng chuỗi “gl=se” |
| ip.addr == 192.168.0.1 hoặc: ip.src == 192.168.0.1 or ip.dst == 192.168.0.1 | Wireshark filter by ip: Lọc tất cả traffic liên quan đến IP 192.168.0.1 |
| ! ( ip.addr == 192.168.0.1 ) hoặc: ! (ip.src == 192.168.0.1 or ip.dst == 192.168.0.1) | Lọc tất cả traffic KHÔNG liên quan đến IP 192.168.0.1 |
| tcp.flags.syn == 1 | Các gói tin TCP có cờ SYN được bật |
| tcp.flags.syn == 1 && tcp.flags.ack == 1 | Các gói tin TCP có cờ SYN/ACK được bật |
| http.host == “quantrilinux.vn” | HTTP request có Host header là “quantrilinux.vn” |
| http.response.code == 404 | Các HTTP request có response status code là 404 |
| smtp || imap || pop | Traffic liên quan đến email (SMTP, IMAP, POP) |
| ! tcp.port == 22 | Loại bỏ traffic SSH |
| ! arp | Loại bỏ traffic ARP |
| ip.version == 4 | Wireshark IPv4 filter: Lọc tất cả các gói tin IP version 4 |
| tcp.srcport == 80 | Wireshark port filter: Lọc tất cả gói tin TCP có source port là 80 |
| tcp.port == 80 | Lọc tất cả các gói tin có liên quan đến port 80 |
| udp.port == 67 or udp.port == 68 | Traffic DHCP |
| dns | Filter traffic liên quan DNS |
| http | Wireshark http filter |
| https | Wireshark https filter |
| ip.src == 192.168.0.1 | Wireshark filter source ip |
| ip.dst == 192.168.0.1 | Wireshark filter destination ip |

Bảng 1. Wireshark Expression

****

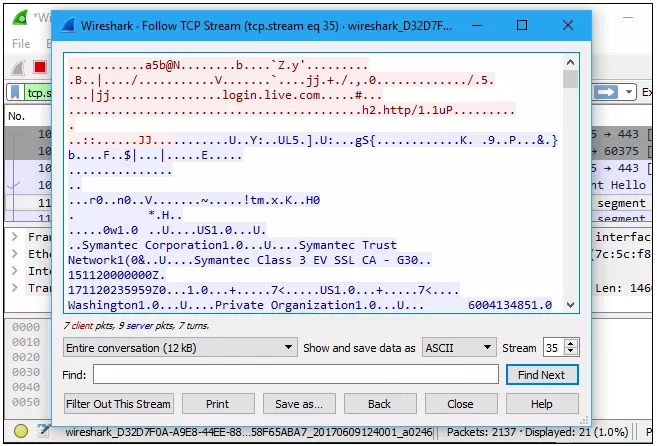
Hình 1. Ví dụ Display Filter

Ngoài ra, người dùng có thể Click vào**Analyze** > **Display Filters**để chọn một filter trong các filter mặc định. Từ đây, ta có thể thêm hoặc custom các filter và lưu chúng để có thể dễ dàng truy cập sau này.

****

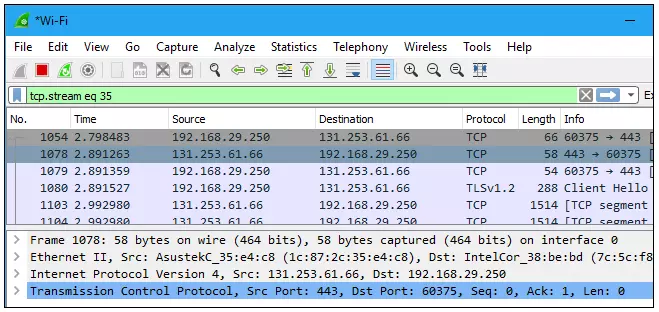
Hình 1. Chọn Filter mặc định trong WireShark

Một tính năng hữu ích khác là **“Follow TCP stream”**, chọn một packet rồi click chuột phải vào packet chọn**“Follow > TCP Stream”**. Sau đó, một hộp thoại sẽ hiện ra cho thấy dữ liệu trao đổi giữa Client và Server trong luồng tương ứng và các packet liên quan. Ta có thể click vào các giao thức khác trong menu Follow để xem đầy đủ các đoạn hội thoại, nếu có thể.



Hình 1. Xem đầy đủ đoạn hội thoại trong Display Filter

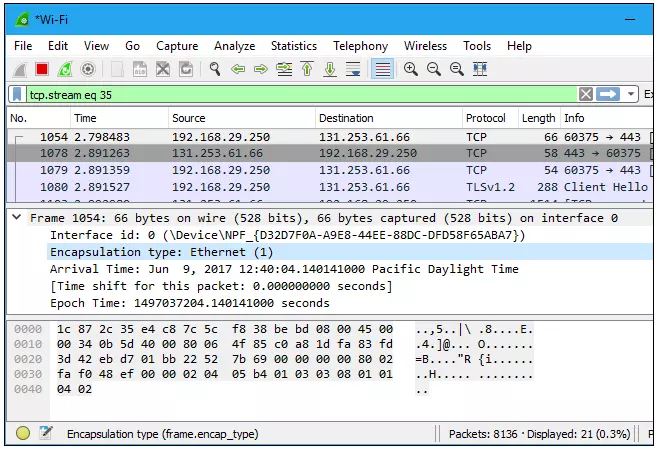
Cuối cùng, đóng cửa sổ lại và một filter sẽ được áp dụng tự động. Bây giờ, **Wireshark** đang hiển thị các packet tạo nên đoạn hội thoại đó.



Hình 1. Hiển thị các packet trong đoạn hội thoại

### **Cách đọc gói tin trong Wireshark**

Click vào một packet để chọn nó, sau đó xem thêm chi tiết về nó.



Hình 1. Phân tích chi tiết gói tin

Ở khung cửa sổ Paket List sẽ cung cấp cho chúng ta các thông tin như:

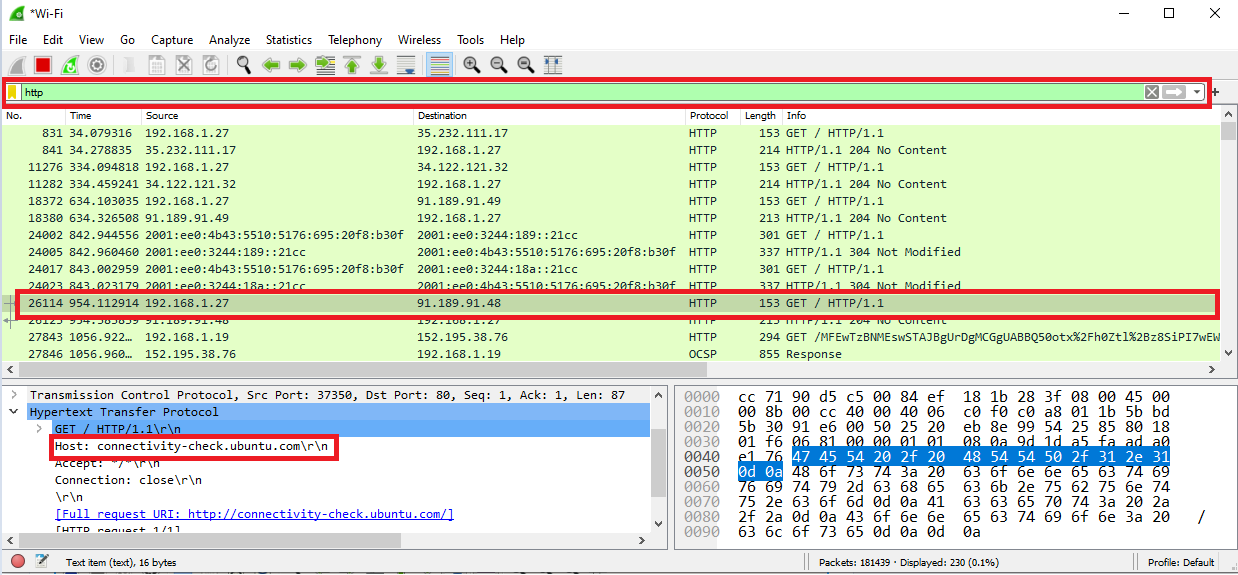
* No: Số thứ tự của gói tin trong file capture hiện tại.
* Time: Thời gian tương đối mà gói tin này được bắt, tính từ lúc bắt đầu quá trình bắt gói tin.
* Source: địa chỉ source IP của kết nối.
* Destination: địa chỉ destination IP của kết nối.
* Length: chiều dài của gói tin.
* Protocol: giao thức của gói tin
* Info: các thông tin tổng quan liên quan đến gói tin.

Ở khung cửa sổ của Packet Details sẽ cho ta thông tin chi tiết từng Layer của packet như:

* Frame: Interface
* Ethernet: Destination, Source, Mac Address
* Internet: Source IP, Destination IP, TTL, Protocol, Flags, Checksum, Identification, Total Length…
* TCP/UDP/ICMP: Source Port, Destination Port, Sequence Number, ACK Number, Flags, TCP Options …
* Application Layer: HTTP, DNS, SMTP…

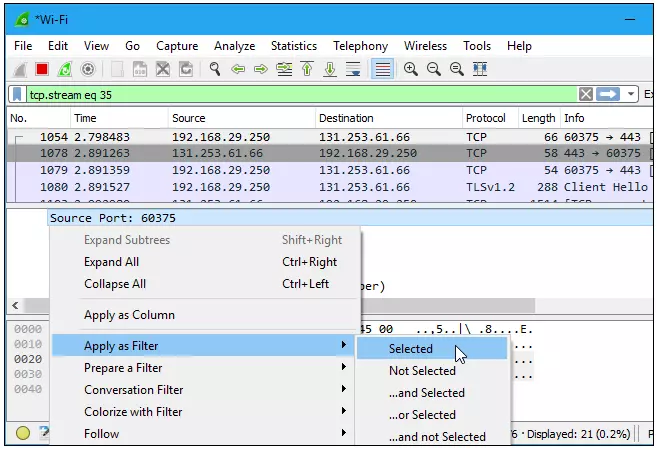
Chúng ta có thể click vào hình mũi tên ở đầu mỗi dòng để thể hiện thêm thông tin chi tiết.

Ở khung cửa sổ của Packet Bytes thể hiện gói tin ở dạng Hex. Khi chúng ta click chọn 1 trường nào đó ở phần Packet Details, những bytes liên quan đến phần đó sẽ được tô đậm ở phần Packet Bytes tương ứng.



Hình 1. Phân tích gói tin bằng Wireshark

Ta cũng có thể tạo thêm các filter ở đây. Chỉ cần click chuột phải vào một trong số các chi tiết, rồi chọn **Apply as Filter** để tạo một filter dựa theo đó.



Hình 1. Tạo filter