

Wireshark For Pentester: A Beginner's Guide

April 13, 2021 By Raj Chandel

Wireshark is an open-source application and it is the world's foremost and widely-used network protocol analyzer that lets you see what's happening on your network at a microscopic level. Just Because it can drill down and read the contents of each packet, it's used to troubleshoot network problems and test software.

Table of contents

- What is Wireshark
- Features
- Installation of Wireshark
- Introduction to Wireshark UI Basic
- Packet Capturing
- Display filter fields
- Building Display Filter Expressions
- Some Useful Filters
- Hands-On Practice

What is Wireshark

Wireshark is an open-source widely used network packet or protocol analyzer. It is an essential tool for security professionals or system administrators. It is used to analyze the structure of different network protocols and has the ability to demonstrate application. Wireshark can be operated in different platforms such as Windows, Unix, Linux and employs the GTK+ widget toolkit or PCAP for packet capturing. IT also has terminal-based free software versions like Tshark. Wireshark shares many characteristics with tcpdump only the difference is that it supports a graphical user interface (GUI) and has information filtering features.

Features

The following are the features that Wireshark provides:

- Can be operated on *UNIX and Windows*.
- *Capture live* packet data from a network interface.
- *Open files* containing packet data captured (PCAP Files) with tcpdump/WinDump, Wireshark, and many other packet capture programs.
- *Import* packets from text files containing hex dumps of packet data.
- Display filters are used to filter and organize the data display.
- Display packets with *very detailed protocol information*.
- New protocols can be scrutinized by creating plug-ins.

- *Captured Traffic can also be trace Voice Over Internet (VOIP) calls over the network.*
- *Export some or all packets in several capture file formats.*
- *Filter packets on many criteria.*
- *Search for packets on many criteria.*
- *Colorize packet display based on filters.*
- *Create various statistics.*
- *...and a lot more!*

Installation of Wireshark

For Windows

Wireshark can be downloaded at no cost from the official website of Wireshark for both Windows and macOS.
Here you can select and download the latest stable version of Wireshark

Wireshark · Download


https://www.wireshark.org/download.html?__cf_chl_captcha_tk__=9e91273e9b30734bb9d

WIRESHARK NEWS Get Acquainted ▾ Get Hel

Download Wireshark

The current stable release of Wireshark is 3.4.4. It supersedes all previous releases.

Stable Release (3.4.4) ^

 **Windows Installer (64-bit)**
Windows Installer (32-bit)
Windows PortableApps® (32-bit)
macOS Intel 64-bit .dmg
Source Code

Old Stable Release (3.2.12) ^

Documentation ^

Not What You're Looking For?

Older Releases

All present and past releases can be found in [our download area](#).

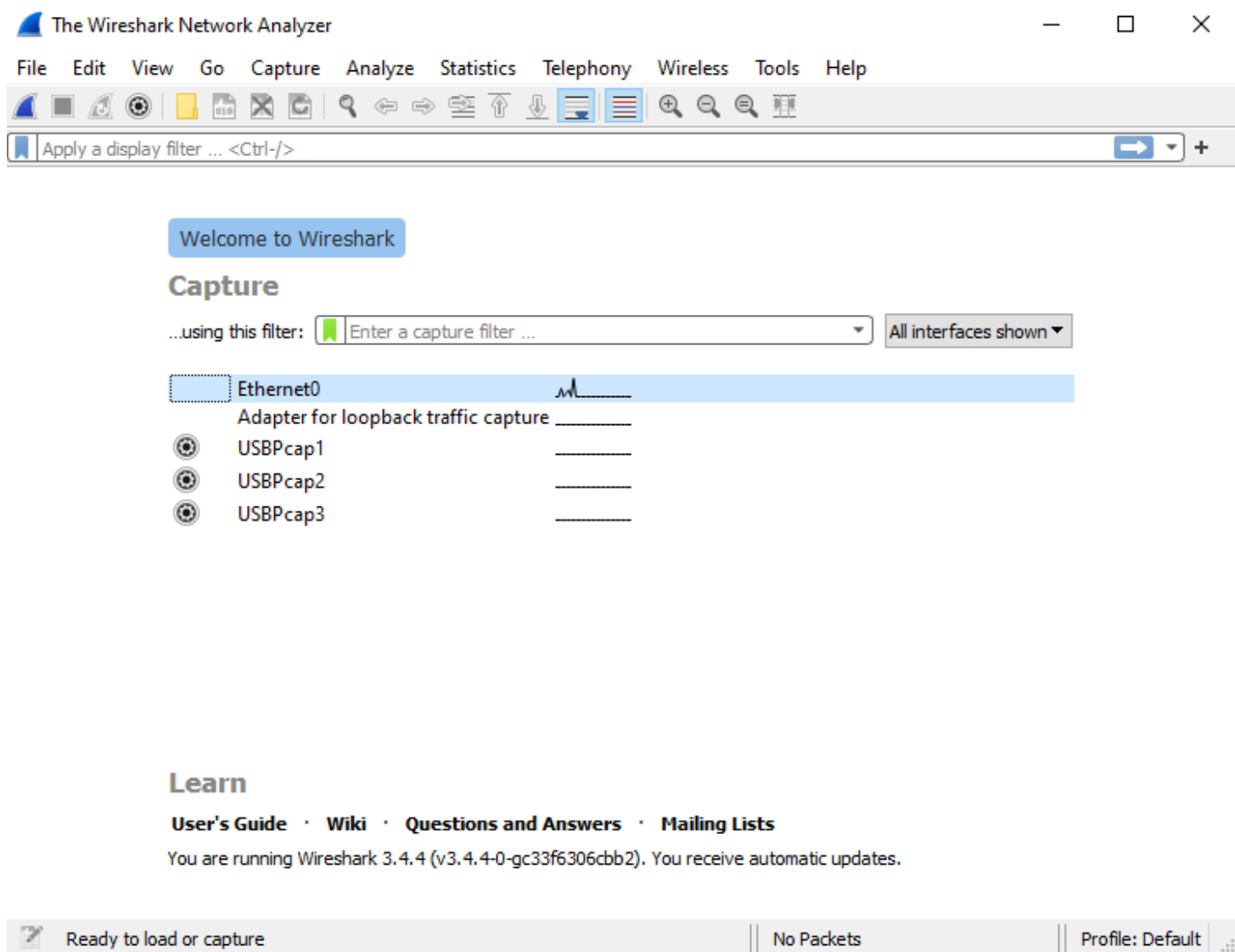
Installation Notes

For a complete list of system requirements and supported platforms, please consult [the User's Guide](#).

Information about each release can be found in [the release notes](#).

After downloading the Wireshark navigate to the downloads directory and run the Wireshark setup. During the installation process of Wireshark, choose to install Npcap if prompted as these include libraries required for live data capture.

After the installation of Wireshark, you must be logged in to the device as an administrator to use Wireshark. In Windows 10 simply search Wireshark and **Run as administrator**. In macOS right-click the Wireshark app icon and select **Get Info**. In the **Sharing & Permissions** settings, give the admin **Read & Write** privileges.



For Linux

Wireshark is also available for Linux and other UNIX like platforms including Red Hat, and FreeBSD.

To download Wireshark, open a terminal and type the following command to install Wireshark:

```
apt install wireshark
```

```
root@ubuntu: ~  
root@ubuntu:~# apt install wireshark  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  libfprint-2-tod1 libllvm10 linux-headers-5.4.0-42  
  linux-headers-5.4.0-42-generic linux-image-5.4.0-42-generic  
  linux-modules-5.4.0-42-generic linux-modules-extra-5.4.0-42-generic  
Use 'sudo apt autoremove' to remove them.  
The following additional packages will be installed:  
  libc-ares2 libdouble-conversion3 liblua5.2-0 libpcre2-16-0 libqt5core5a  
  libqt5dbus5 libqt5gui5 libqt5multimedia5 libqt5multimedia5-plugins  
  libqt5multimediagsttools5 libqt5multimediawidgets5 libqt5network5  
  libqt5opengl5 libqt5printsupport5 libqt5svg5 libqt5widgets5 libsmi2ldbl  
  libsnappy1v5 libspandsp2 libssh-gcrypt-4 libwireshark-data libwireshark13  
  libwiretap10 libwsutil11 libxcb-xinerama0 libxcb-xinput0  
  qt5-gtk-platformtheme qttranslations5-l10n wireshark-common wireshark-qt  
Suggested packages:  
  qt5-image-formats-plugins qtwayland5 snmp-mibs-downloader geoipupdate  
  geoip-database geoip-database-extra libjs-leaflet  
  libjs-leaflet.markercluster wireshark-doc
```

Press ‘Y’ when prompted to occupy additional space. During installation, Wireshark configuration will ask “should non-super users be able to capture the packets?”. For security purpose, it is not advisable to allow non-super users to access Wireshark. As of now, continue by pressing ‘yes’. Wireshark installation will continue and successfully install into the system.

Package configuration

Configuring wireshark-common

Dumpcap can be installed in a way that allows members of the "wireshark" system group to capture packets. This is recommended over the alternative of running Wireshark/Tshark directly as root, because less of the code will run with elevated privileges.

For more detailed information please see `/usr/share/doc/wireshark-common/README.Debian.gz` once the package is installed.

Enabling this feature may be a security risk, so it is disabled by default. If in doubt, it is suggested to leave it disabled.

Should non-superusers be able to capture packets?

<Yes>

<No>

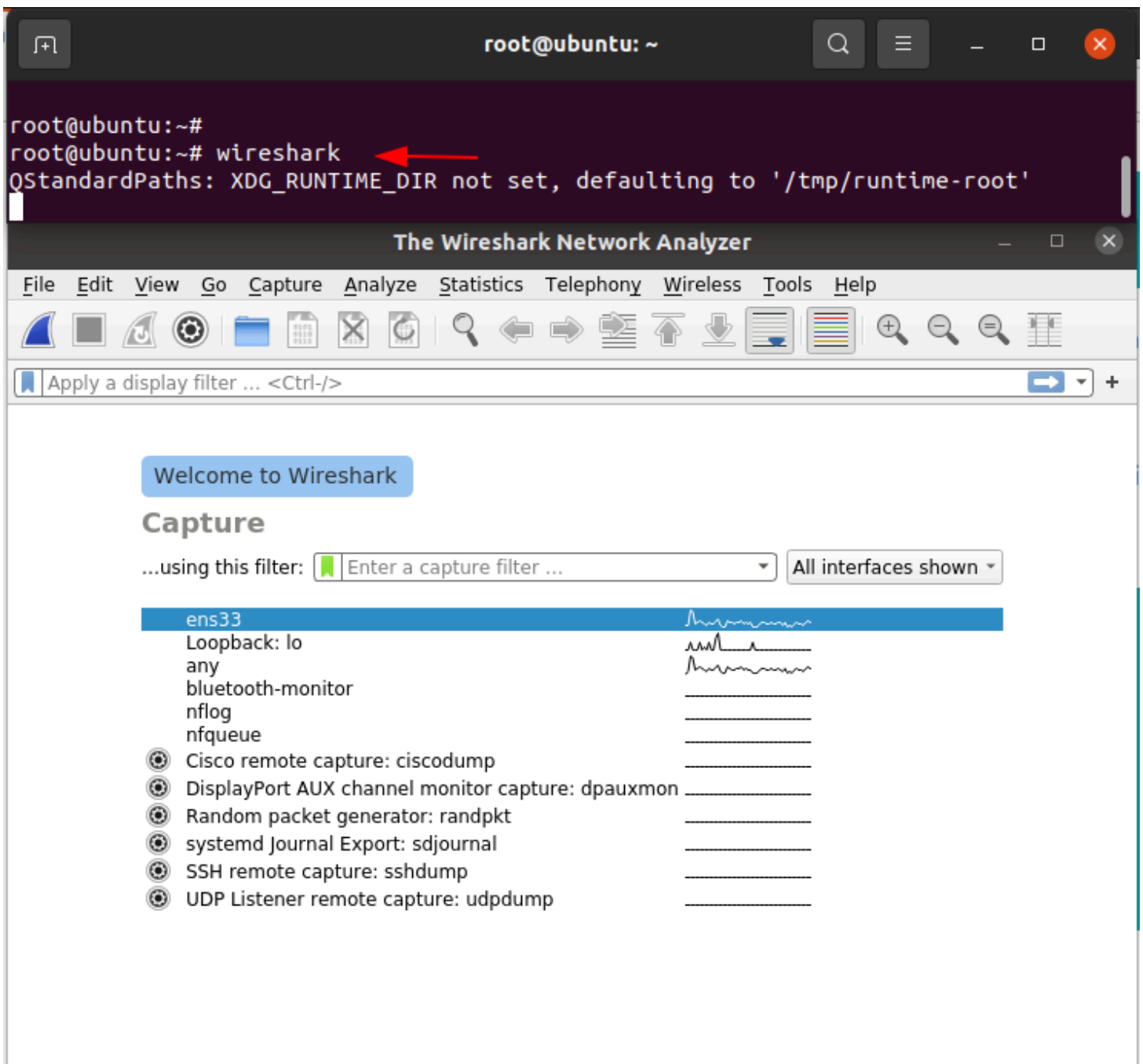
Type the following code to verify the installation package of Wireshark:

```
apt show Wireshark
```

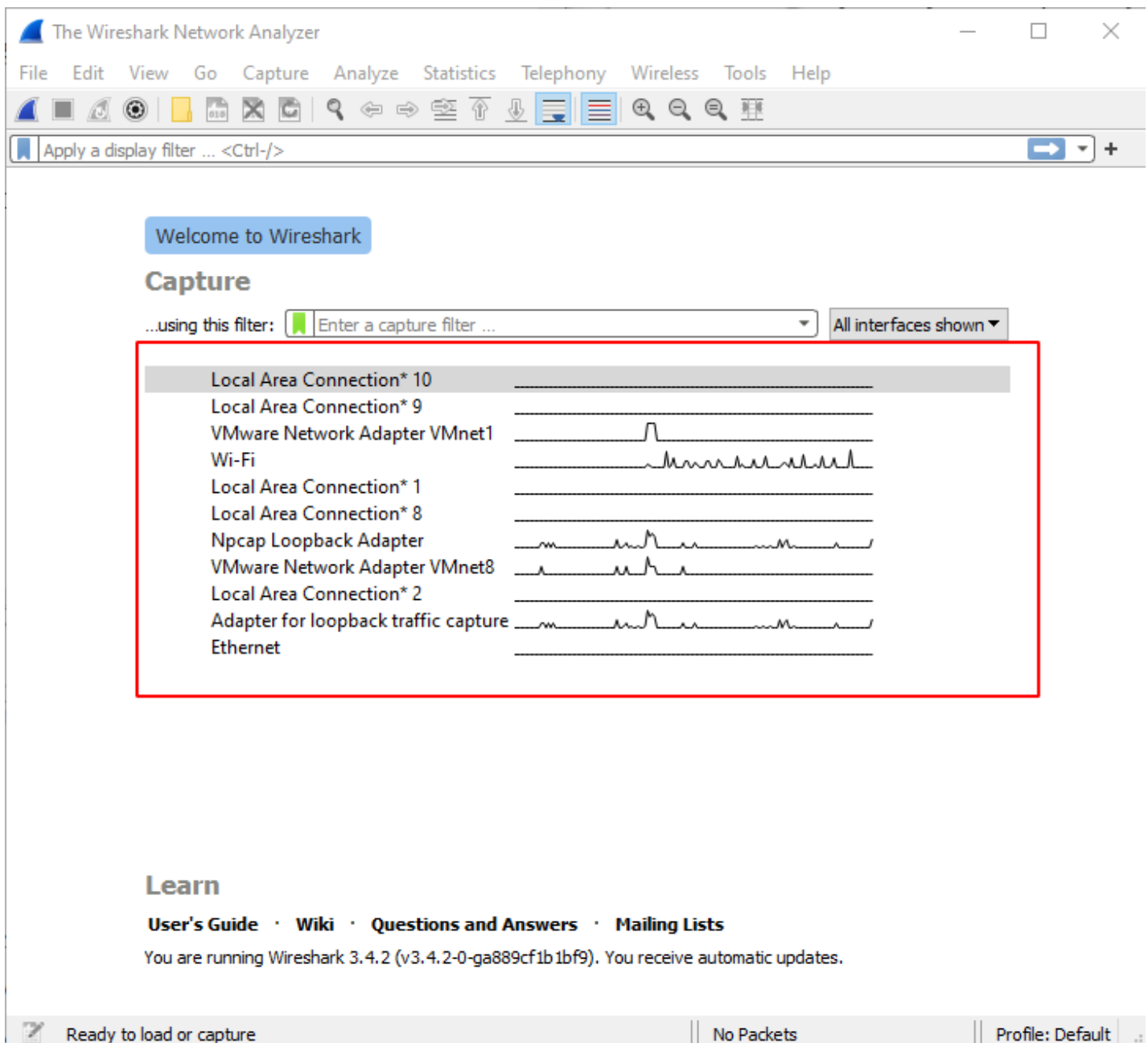
```
root@ubuntu: ~  
root@ubuntu:~# apt show wireshark  
Package: wireshark  
Version: 3.2.3-1  
Priority: optional  
Section: universe/net  
Origin: Ubuntu  
Maintainer: Balint Reczey <rbalint@ubuntu.com>  
Bugs: https://bugs.launchpad.net/ubuntu/+filebug  
Installed-Size: 59.4 kB  
Depends: wireshark-qt (= 3.2.3-1)  
Conflicts: ethereal (<< 1.0.0-3)  
Replaces: ethereal (<< 1.0.0-3)  
Homepage: https://www.wireshark.org/  
Download-Size: 5,088 B  
APT-Manual-Installed: yes  
APT-Sources: http://us.archive.ubuntu.com/ubuntu focal/universe amd64 Packages  
Description: network traffic analyzer - meta-package  
Wireshark is a network "sniffer" - a tool that captures and analyzes  
packets off the wire. Wireshark can decode too many protocols to list  
here.  
.  
This is a meta-package for Wireshark.  
root@ubuntu:~#
```

And to open the Wireshark run the following command and the Wireshark application will be visible as below:

Wireshark



Whenever you open Wireshark you will be prompted with the following screen.



Here you can see different network interfaces on your device. In the above image we can see there is a lot of traffic being communicated through the Wi-Fi interface. In most cases, you will only be able to see traffic going in and out of your own device, however, some wireless network cards can be set into monitor mode so that you will be able to see traffic from other wireless devices on the network.

Introduction to Wireshark UI Basics

As of this now you have installed Wireshark into your systems and likely excited to get started capturing your first packets. Without wasting of much time let's get started!!!

Now we're going to explore

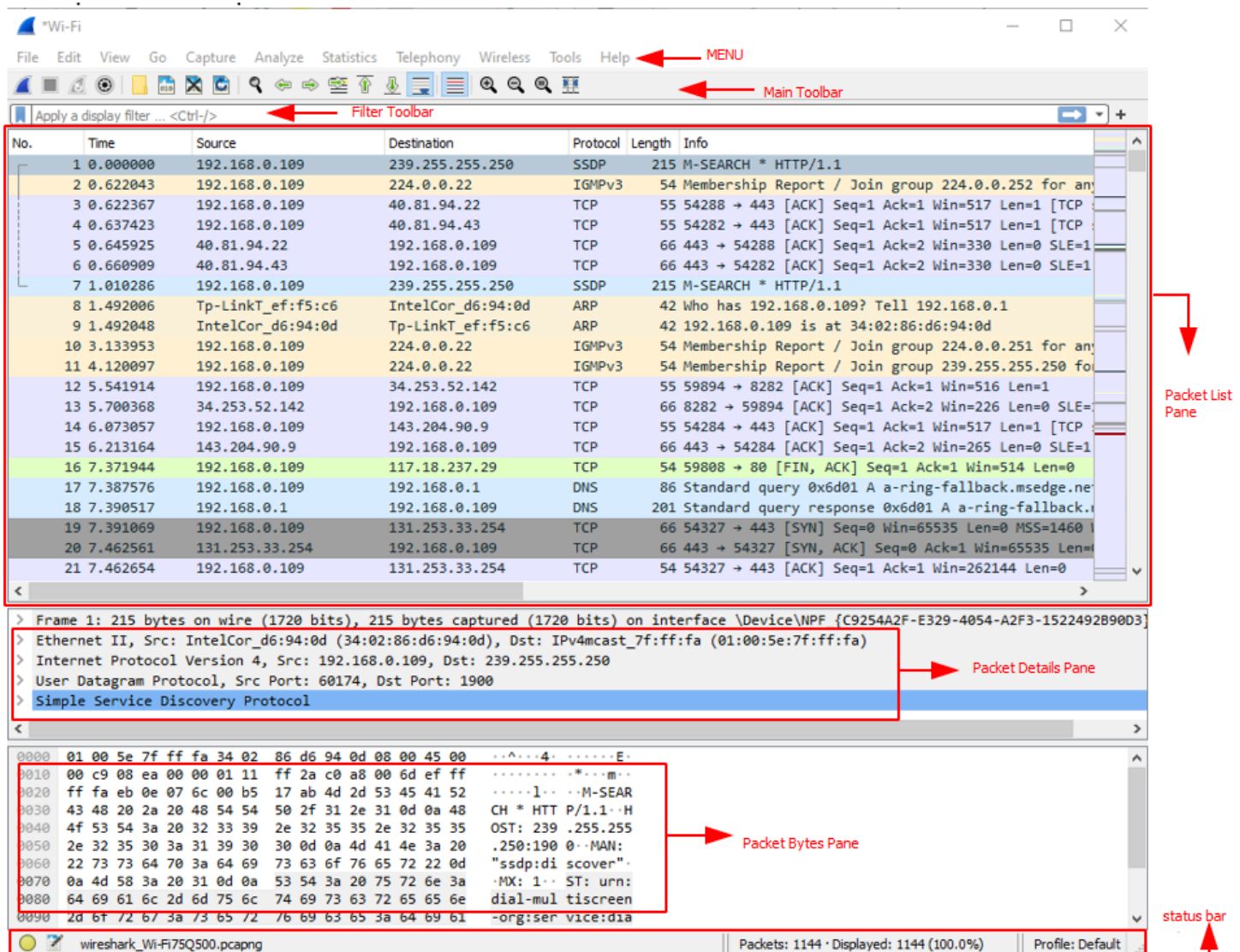
- How Wireshark interface works
- How to view packets in Wireshark

- How to capture packets in Wireshark
- How to perform Trace Analysis in Wireshark
- How to filter packets in Wireshark
-and much more things!!

Wireshark can be started through windows program manager by searching Wireshark or also can be started through the command line by typing “Wireshark” in the directory of Wireshark.

The Main Window

Let’s quickly take a look at the Wireshark user interface. Usually, you would see this similar scenario after some packets are captured or loaded.



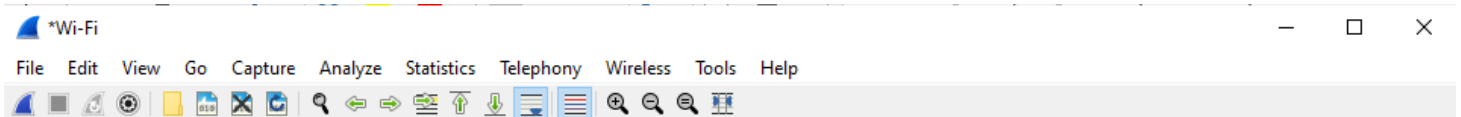
Wireshark main window consists of these parts that are commonly called GUI programs.

1. The menu is used to start actions
2. The main toolbar quick access to frequently used items from the menu
3. Filter Toolbar allows user to set display filters to filter which packet should be displayed

4. The Packet list pane displays a summary of each packet captured.
5. The Packet details pane displays the packet selected in the packet list pane
6. The packet bytes pane displays the data from the packet selected in the packet list pane and highlights the field selected in the packet details pane
7. The status bar shows some detailed information about the current program state and the captured data.

The Menu

Wireshark main menu is located at the top of the main window (window, Linux).



The main menu contains the following Items:

File

This menu contains items to open and merge capture files, save, print, or export capture files in different Formats

Edit

This menu contains items to find a packet, time reference or mark one or more packets, handle configuration profiles, and set your preferences; (cut, copy, and paste are not presently implemented). The Wireshark Edit menu contains the fields as shown in the below image

View

This menu controls the display of the captured data, including colourization of packets, zooming the font, showing a packet in a separate window, expanding and collapsing trees in packet details.

Go

This menu contains items to go to a specific packet.

Capture

This menu allows you to start and stop captures and edit capture filters. Some of the important filters that make our capture more efficient are described below.

Analyze

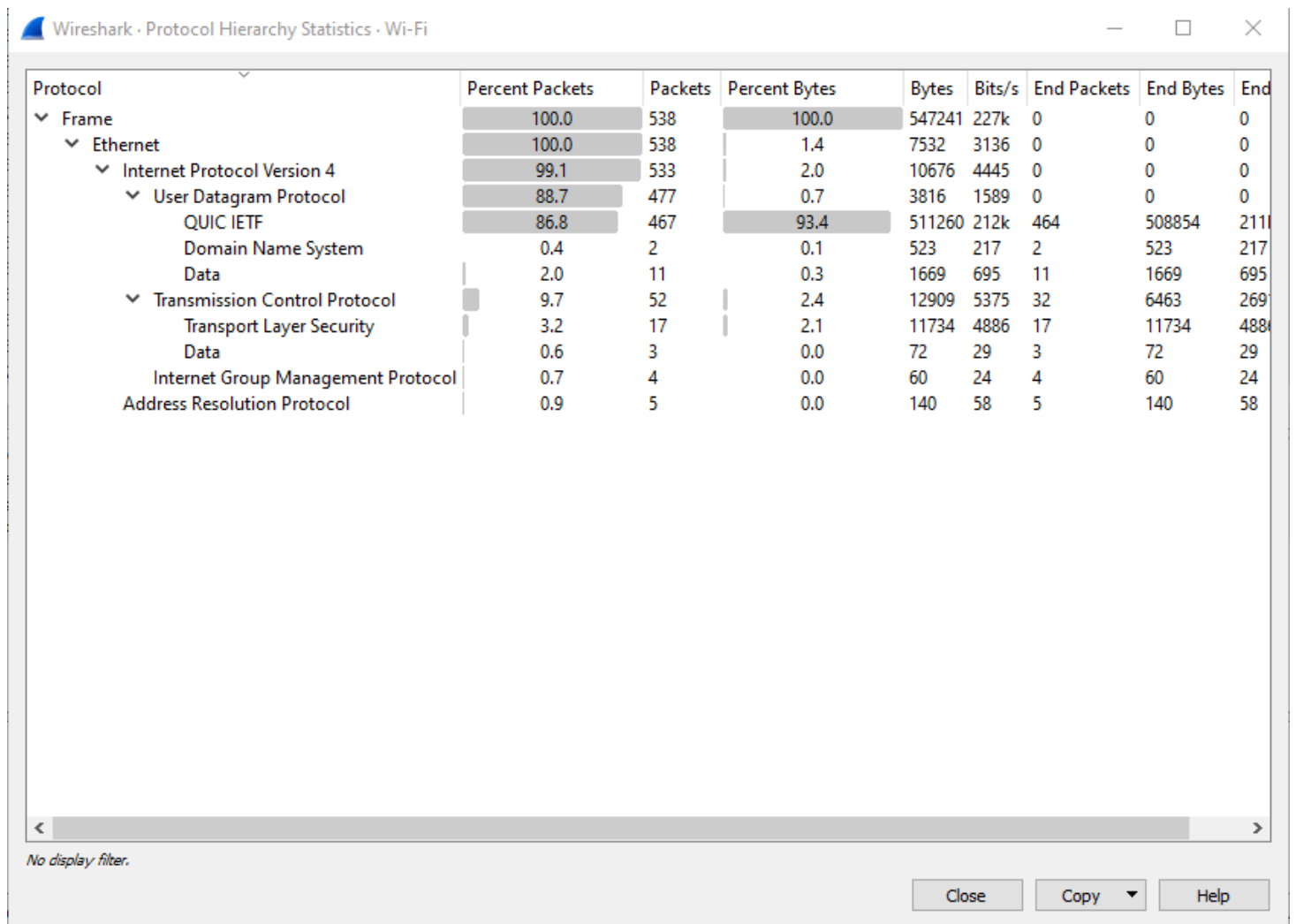
This menu contains items to manipulate display filters, enable or disable the dissection of protocols, configure user-specified decodes and follow a TCP stream.

Statistics

This menu contains items to display various statistic windows, including a summary of the packets that have been captured, a display protocol hierarchy statistics and much more. Some of the important filters that make our Trace analysis more efficient are described below.

Statistics -> Protocol Hierarchy

- Presents descriptive statistics per protocol.
- Useful for determining the types, amounts, and relative proportions of protocols within a trace



Statistics -> Conversations

- Generates descriptive statistics about each conversation for each protocol in the trace.

Wireshark · Conversations · Wi-Fi

Ethernet · 6		IPv4 · 49		IPv6	TCP · 18		UDP · 49							
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A	
5.189.160.21	59176	192.168.0.109	49798	2	500	1	139	1	361	35.707985	0.0006	—	—	
105.112.101.150	137	192.168.0.109	137	3	276	0	0	3	276	65.620324	3.0218	0	730	
176.17.110.118	137	192.168.0.109	137	3	276	0	0	3	276	83.700050	3.0228	0	730	
188.143.94.195	59904	192.168.0.109	49798	2	476	1	145	1	331	59.696701	0.0005	—	—	
192.168.0.101	5353	224.0.0.251	5353	5	680	5	680	0	0	64.941035	22.0139	247	0	
192.168.0.109	49798	54.70.28.180	6881	2	254	1	145	1	109	3.982163	1.0198	1137	855	
192.168.0.109	49798	118.238.105.185	6881	2	523	1	377	1	146	4.502544	0.0003	—	—	
192.168.0.109	59213	192.168.0.1	53	2	607	1	89	1	518	6.911299	0.0025	—	—	
192.168.0.109	49798	159.196.228.199	3251	2	500	1	361	1	139	10.067895	0.0007	—	—	
192.168.0.109	49798	45.14.225.8	6339	1	145	1	145	0	0	10.965520	0.0000	—	—	
192.168.0.109	49798	185.157.221.247	25401	2	233	1	112	1	121	14.520769	0.0004	—	—	
192.168.0.109	59215	163.53.87.205	443	1,747	2128k	221	20k	1,526	2107k	16.560878	30.5271	5471	552k	
192.168.0.109	49798	131.147.215.124	22840	2	476	1	145	1	331	17.976309	0.1503	7718	17k	
192.168.0.109	59218	239.255.255.250	1900	4	860	4	860	0	0	21.159028	3.0222	2276	0	
192.168.0.109	49798	85.175.24.75	36090	2	505	1	359	1	146	24.322242	0.0005	—	—	
192.168.0.109	49798	41.190.31.14	26305	1	145	1	145	0	0	24.983319	0.0000	—	—	
192.168.0.109	59220	142.250.192.238	443	11	6933	4	2934	7	3999	26.473376	0.1109	211k	288k	
192.168.0.109	49798	54.214.105.212	6881	2	254	1	145	1	109	31.972508	1.0482	1106	831	
192.168.0.109	49798	93.104.54.130	49001	4	1392	2	1001	2	391	38.122057	19.0303	420	164	
192.168.0.109	49798	223.225.170.221	32023	1	145	1	145	0	0	38.961315	0.0000	—	—	
192.168.0.109	49798	188.242.253.229	41337	2	476	1	145	1	331	45.981992	0.1964	5906	13k	
192.168.0.109	49798	197.252.202.95	21046	2	505	1	359	1	146	45.996464	0.0006	—	—	
192.168.0.109	51987	192.168.0.1	53	2	235	1	87	1	148	47.987934	0.0713	9766	16k	
192.168.0.109	137	197.252.202.95	137	3	276	3	276	0	0	48.059949	3.0185	731	0	
192.168.0.109	49798	88.109.57.199	6895	2	505	1	359	1	146	48.716617	0.0006	—	—	
192.168.0.109	49798	194.87.220.20	9850	2	476	1	145	1	331	52.979140	0.2039	5688	12k	
192.168.0.109	49798	88.127.230.103	23712	2	505	1	359	1	146	55.449494	0.0006	—	—	
192.168.0.109	49798	77.133.61.87	44900	2	476	1	331	1	145	56.046705	0.0005	—	—	
192.168.0.109	63221	192.168.0.1	53	3	373	2	170	1	203	57.475076	0.4599	2957	3531	
192.168.0.109	62884	192.168.0.1	53	2	518	1	77	1	441	57.599999	0.0047	—	—	
192.168.0.109	49798	91.144.176.221	40331	2	476	1	145	1	331	59.983077	0.2035	5700	13k	
192.168.0.109	52095	192.168.0.1	53	3	443	2	174	1	269	61.521663	0.5958	2336	3611	
192.168.0.109	49798	105.112.101.150	23853	4	1010	2	718	2	292	63.530956	25.9380	221	90	
192.168.0.109	58471	192.168.0.1	53	2	237	1	88	1	149	65.545202	0.0745	9450	16k	

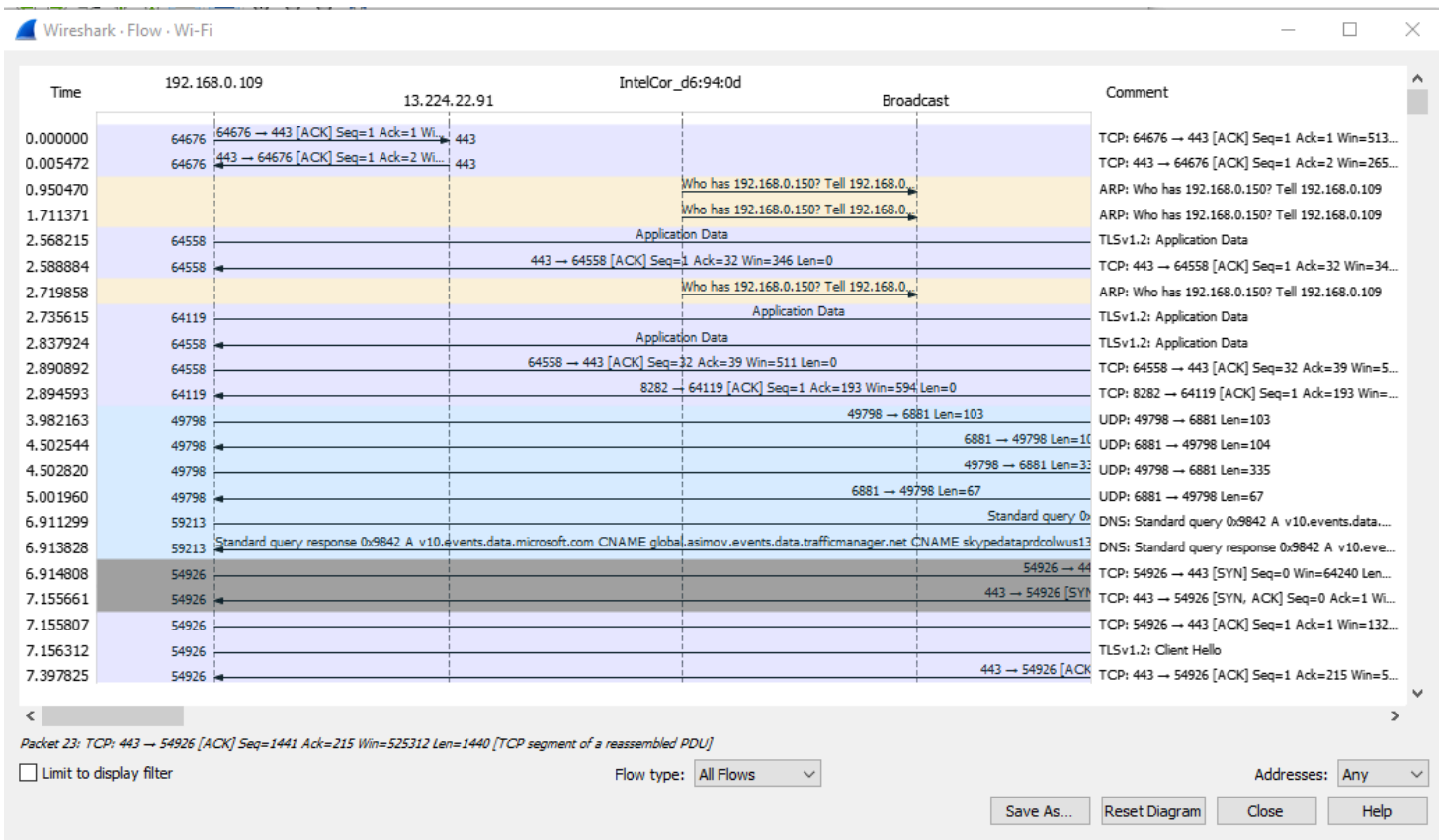
☐ Name resolution☐ Limit to display filter☐ Absolute start time

Conversation Types ▾

Copy ▾Follow Stream...Graph...CloseHelp

Statistics -> Flow Graph

- Generates a sequence graph for the selected traffic.
- Useful for understanding seq. and ack. calculations.



Telephony

This menu contains items to display various telephony related statistic windows, including a media analysis, flow diagrams, display protocol hierarchy statistics and much more.

Wireless

This menu contains items to display Bluetooth and IEEE 802.11 wireless statistics.

Tools

This menu contains various tools available in Wireshark, such as creating Firewall ACL Rules.

Help











This menu contains items to help the user, e.g. access to some basic help, manual pages of the various command-line tools, online access to some of the webpages, and the usual dialogue.











The Main Toolbar

The main toolbar provides quick access to frequently used items from the menu. This toolbar can customize by the user.



Actions of this filter toolbar are described below

Toolbar Icon	Toolbar Item	Description
	Start	Starts capturing packets with the same options as the last capture or the default options if none were set
	Stop	Stops the currently running capture
	Restart	Restarts the current capture session
	Options	Opens the “Capture Options” dialog box
	Open	Opens the file open dialog box, which allows you to load a capture file for viewing
	Save As	Save the current capture file to whatever file you would like
	Close	Closes the current capture. If you have not saved the capture, you will be asked to save it first
	Reload	Reloads the current capture file
	Find packet	Find a packet based on different criteria
	Go back	Jump back in the packet history. Hold down the Alt key (Option key on macOS) to go forward in the selection history

Toolbar Icon	Toolbar Item	Description
	Go Forward	Jump forward in the packet history. Hold down the Alt key (Option on macOS) to go forward in the selection history.
	Go To Packet	Go to a specific packet
	Go To First Packet	Jump to the last packet of the capture file
	Go To Last Packet	Jump to the last packet of the capture file
	Auto Scroll in Live capture	Auto scroll packet list while doing a live capture (or not)
	Colorize	Colorize the packet list (or not)
	Zoom In	Zoom out of the packet data (increase the font size)
	Zoom Out	Zoom out of the packet data (decrease the font size)
	Normal Size	Set zoom level back to 100%
	Resize Columns	Resize columns, so the content fits into them.







Reference: – https://www.wireshark.org/docs/wsug_html_chunked/ChUseMainToolbarSection.html

The Filter Toolbar

The filter toolbar lets you quickly edit and apply display filters.



Actions of this filter toolbar are described below

Toolbar Icon	Name	Description
	Bookmarks	Manage or select save filters.
	Filter input	The area is provided to enter or edit a display string. A syntax check of your filter string while you are typing such as the background will turn red if you enter an incomplete or invalid string and will become green when you enter a valid string.
	Clear	Reset the current display filter and clear the edit area.
	Apply	Apply the current value in the edit area as the new display filter.
	Recent	Select from a list of recently applied filters.
	Add Button	Add a new filter button.

The packet list pane

The packet list pane displays all the packets in the order they were recorded.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.109	239.255.255.250	SSDP	215	M-SEARCH * HTTP/1.1
2	0.622043	192.168.0.109	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.252 for any
3	0.622367	192.168.0.109	40.81.94.22	TCP	55	54288 → 443 [ACK] Seq=1 Ack=1 Win=517 Len=1 [TCP
4	0.637423	192.168.0.109	40.81.94.43	TCP	55	54282 → 443 [ACK] Seq=1 Ack=1 Win=517 Len=1 [TCP
5	0.645925	40.81.94.22	192.168.0.109	TCP	66	443 → 54288 [ACK] Seq=1 Ack=2 Win=330 Len=0 SLE=1
6	0.660909	40.81.94.43	192.168.0.109	TCP	66	443 → 54282 [ACK] Seq=1 Ack=2 Win=330 Len=0 SLE=1
7	1.010286	192.168.0.109	239.255.255.250	SSDP	215	M-SEARCH * HTTP/1.1
8	1.492006	Tp-LinkT_ef:f5:c6	IntelCor_d6:94:0d	ARP	42	Who has 192.168.0.109? Tell 192.168.0.1
9	1.492048	IntelCor_d6:94:0d	Tp-LinkT_ef:f5:c6	ARP	42	192.168.0.109 is at 34:02:86:d6:94:0d
10	3.133953	192.168.0.109	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.251 for any
11	4.120097	192.168.0.109	224.0.0.22	IGMPv3	54	Membership Report / Join group 239.255.255.250 fo
12	5.541914	192.168.0.109	34.253.52.142	TCP	55	59894 → 8282 [ACK] Seq=1 Ack=1 Win=516 Len=1
13	5.700368	34.253.52.142	192.168.0.109	TCP	66	8282 → 59894 [ACK] Seq=1 Ack=2 Win=226 Len=0 SLE=
14	6.073057	192.168.0.109	143.204.90.9	TCP	55	54284 → 443 [ACK] Seq=1 Ack=1 Win=517 Len=1 [TCP
15	6.213164	143.204.90.9	192.168.0.109	TCP	66	443 → 54284 [ACK] Seq=1 Ack=2 Win=265 Len=0 SLE=1
16	7.371944	192.168.0.109	117.18.237.29	TCP	54	59808 → 80 [FIN, ACK] Seq=1 Ack=1 Win=514 Len=0
17	7.387576	192.168.0.109	192.168.0.1	DNS	86	Standard query 0x6d01 A a-ring-fallback.msedge.ne
18	7.390517	192.168.0.1	192.168.0.109	DNS	201	Standard query response 0x6d01 A a-ring-fallback.i
19	7.391069	192.168.0.109	131.253.33.254	TCP	66	54327 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460
20	7.462561	131.253.33.254	192.168.0.109	TCP	66	443 → 54327 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=
21	7.462654	192.168.0.109	131.253.33.254	TCP	54	54327 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0

Each line in the packet list corresponds to one packet in the capture file select the lines to get more details. More details will be displayed In the Packet details pane and packet byte panes.

There are lots of column available such as

- No: -The number of the packet in the capture file. This number won't change, even if a display filter is used.
- Time: -The timestamp of when the packet was captured is displayed in this column. The presentation format of this timestamp can be changed.
- Source: -The address where this packet is coming from.
- Destination: -The address where this packet is going.

- Protocol: -The highest-level protocol that Wireshark can detect.
- Length: -The length in bytes of each packet.
- Info: -Additional information about the packet content.

The packet details pane

The packet details pane shows the selected or current packet in a detailed form.

```
> Frame 5: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{C9254A2F-E329-4054-A2F3-1522492B90D3}, id 0
> Ethernet II, Src: Tp-LinkT_ef:f5:c6 (1c:3b:f3:ef:f5:c6), Dst: IntelCor_d6:94:0d (34:02:86:d6:94:0d)
> Internet Protocol Version 4, Src: 142.250.193.227, Dst: 192.168.0.109
▼ Transmission Control Protocol, Src Port: 443, Dst Port: 57048, Seq: 1, Ack: 2, Len: 0
    Source Port: 443
    Destination Port: 57048
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence Number: 1 (relative sequence number)
    Sequence Number (raw): 3028339892
    [Next Sequence Number: 1 (relative sequence number)]
    Acknowledgment Number: 2 (relative ack number)
    Acknowledgment number (raw): 4155320799
    1000 .... = Header Length: 32 bytes (8)
> Flags: 0x010 (ACK)
    Window: 261
    [Calculated window size: 261]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0xa055 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
> Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), SACK
> [SEQ/ACK analysis]
> [Timestamps]
```

The above pane shows the protocols and protocol fields of the packet selected in the “packet list” pane. The protocols are shown in a tree which can be expanded and collapsed.”

The Packet Bytes pane

The packet bytes pane shows the data of the selected or current packet in hex dump style.

```
0000 34 02 86 d6 94 0d 1c 3b f3 ef f5 c6 08 00 45 00 4.....; .....E.
0010 00 34 b7 cf 00 00 3c 06 b5 01 8e fa c1 e3 c0 a8 .4.....<.....
0020 00 6d 01 bb de d8 b4 80 cc b4 f7 ad 29 df 80 10 .m.....)....
0030 01 05 a0 55 00 00 01 01 05 0a f7 ad 29 de f7 ad ...U.....)....
0040 29 df )..
```

The packet bytes pane shows a canonical hex dump of the packet data. Each line contains the data offset, sixteen hexadecimal bytes and sixteen ASCII bytes. Non-printable bytes are replaced with period “.”

The status bar

The status bar displays informational messages such as



The colourized bullet

The left side shows the highest expert information in the currently loaded capture file. Hovering the mouse on the colourized bullet will show you a description of the expert information level.

The edit icon

This allows you to add a comment to the capture file using the capture file properties dialogue.

The middle

It shows the current number of packets in the capture file. The following values are displayed:

Packets

The number of packets is being captured.

Displayed

The number of packets is being displayed.

Marked

The number of marked packets. Only displayed if you mark any packets in the capture.

Dropped

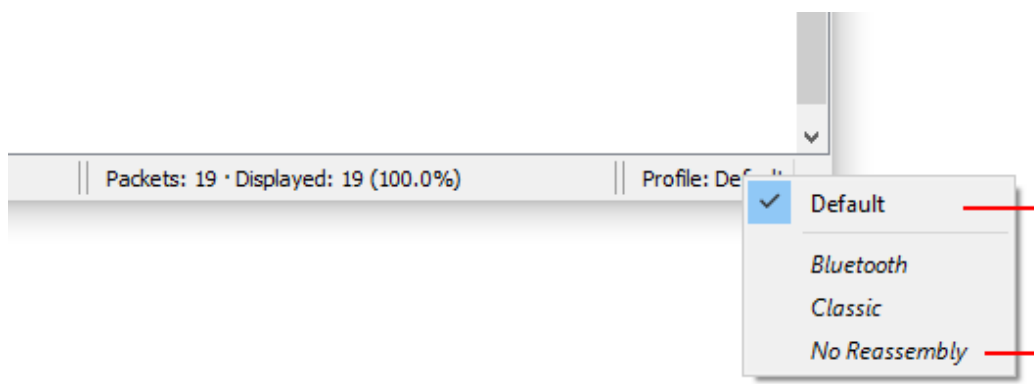
It shows the number of dropped packets. only displayed If Wireshark was unable to capture all packets.

Ignored

It shows the number of ignored packets and it will only be displayed if you ignore any of the packets.

The right side

it shows the selected configuration profile. Clicking on this part of the status bar will bring up a menu with all available configuration profiles, and selecting from this list will change the configuration profile.



Packet Capturing

The following methods can be used to start capturing packets

You can double-click on the interface in the welcome screen of Wireshark

If you already know the name of the capture interface then you can start Wireshark from the command line by running the following command:

```
wireshark -I eth0 -k
```

This will start Wireshark capturing on interface eth0

Once you have captured some packets you can view the packets that are displayed in the packet list pane by simply clicking on a packet on a packet list pane, which will bring up the selected packet in the tree view and byte view panes. As soon you capture some traffic then you need to apply some filter to make it easily understandable.

Wireshark has two filtering languages

- Capture filters
- Display filters

Capture filters are used for filtering when capturing packets and display filters are used for filtering which packets are displayed. Wireshark provides a display filter language that enables you to precise control which packets are displayed

Display filter fields

Wireshark's display filters a bar located right above the column display section. To only display packets containing a particular protocol, type the protocol into Wireshark's display filter toolbar Wireshark offers a list of suggestion based on the text that you typed.

For example, to only display TCP packets, type tcp into Wireshark's display filter toolbar.

mergedtest.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.analysis

No.	Time	Source	Destination	Protocol	Length	Info
1	0	10.135.168.135	10.135.168.135	FTP	62	[TCP ZeroWindow] Request: TYPE I
2	0	10.135.168.135	10.135.168.135	TCP	60	1040 → 149 [SYN] Seq=0 Win=0 Len=0
3	0	10.135.168.135	10.135.168.135	TCP	60	1040 → 21 [ACK] Seq=1 Ack=1 Win=4380
4	0	10.135.168.135	10.135.168.135	TCP	72	[TCP Port numbers reused] 1040 → 21 [ACK] Seq=1 Ack=1 Win=4380
5	6	10.135.168.135	10.135.168.135	TCP	60	4995 → 21 [SYN] Seq=0 Win=0 Len=0
6	6	10.135.168.135	10.135.168.135	TCP	60	4995 → 21 [ACK] Seq=1 Ack=1 Win=4380
7	6	10.135.168.135	10.135.168.135	TCP	72	[TCP ZeroWindow] [TCP Retransmission]
8	6	10.135.168.135	10.135.168.135	FTP	60	[TCP ZeroWindow] [TCP Previous segment]
9	1	10.135.168.135	10.135.168.135	IPv4	60	Control Message (38)
10	1	0.135.168.135	10.135.168.135	TCP	60	4974 → 21 [ACK] Seq=1 Ack=1 Win=4380
11	1	10.135.168.135	10.135.168.135	FTP	72	[TCP ZeroWindow] Request: USPR DWSData
12	1	10.135.168.135	10.135.168.135	TCP	60	[TCP ZeroWindow] 4974 → 16405 [PSH, ACK]
13	1	10.135.168.135	10.135.168.135	TCP	60	4954 → 21 [SYN] Seq=0 Win=0 Len=0
14	1	10.135.168.135	10.135.168.135	TCP	60	4954 → 21 [ACK] Seq=1 Ack=1 Win=4380
15	1	10.135.168.135	10.135.168.135	FTP	72	[TCP ZeroWindow] Request: USER DWSData
16	1	10.135.168.135	10.135.168.135	FTP	60	[TCP ZeroWindow] Request: QUIT
17	2	10.135.168.135	10.135.168.135	TCP	60	4933 → 21 [SYN] Seq=0 Win=0 Len=0

Frame 27018: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface unknown, id 0

- Interface id: 0 (unknown)
- Encapsulation type: Ethernet (1)
- Arrival Time: Feb 27, 2013 22:03:13.788019000 India Standard Time
- [Time shift for this packet: 0.000000000 seconds]
- Epoch Time: 1361982793.788019000 seconds
- [Time delta from previous captured frame: 0.000120000 seconds]
- [Time delta from previous displayed frame: 0.000120000 seconds]
- [Time since reference or first frame: 120422382.795179000 seconds]
- Frame Number: 27018
- Frame Length: 60 bytes (480 bits)

Similarly, to only display packets containing a particular field, type the field into Wireshark's display filter toolbar. For example, to only display HTTP requests, type **http.request** into Wireshark's display filter toolbar and it will accept the expression and works as intended

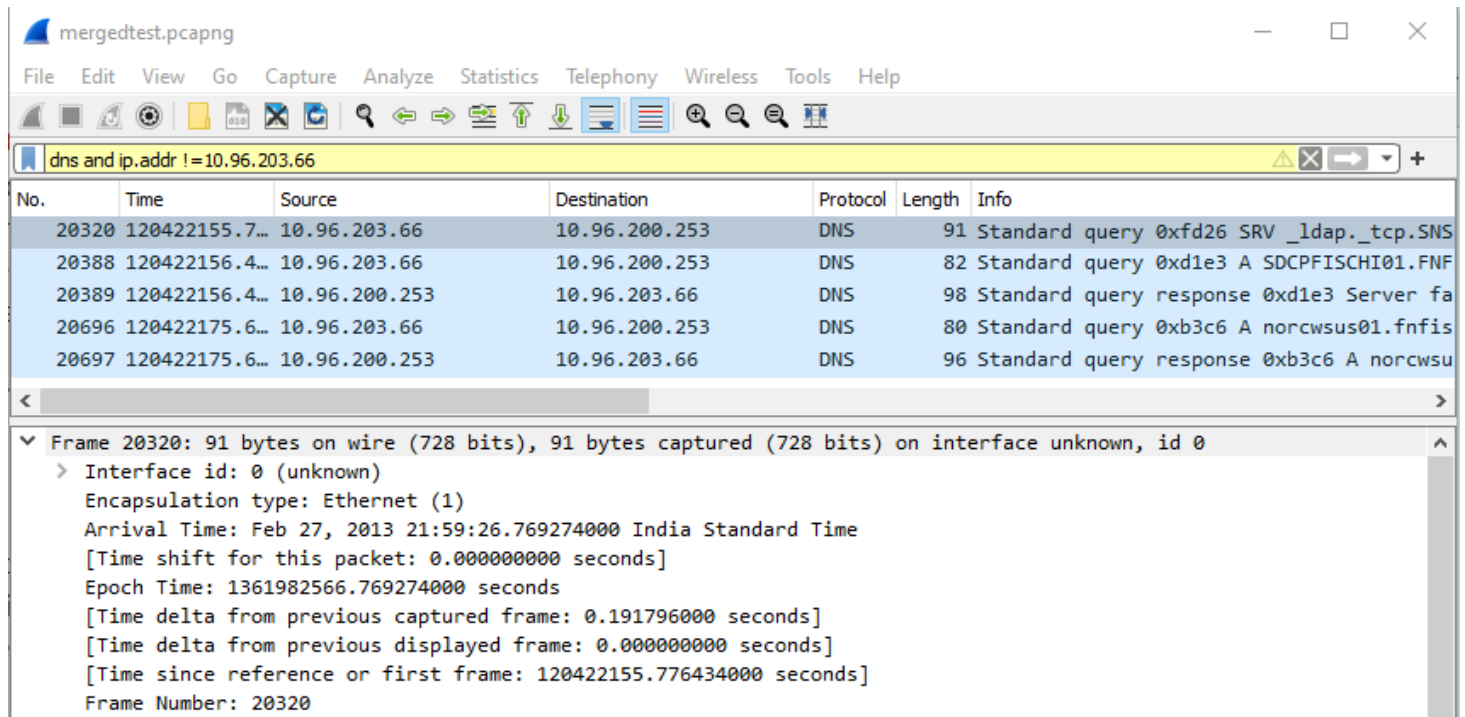
mergedtest.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.request

No.	Time	Source	Destination	Protocol	Length	Info
20704	120422175.6...	10.96.203.66	10.121.1.161	HTTP	2465	CCM_POST /ccm_system/request HTTP/1.1
24537	120422319.7...	10.96.203.66	10.121.1.161	HTTP	345	CCM_POST /ccm_system/request HTTP/1.1
30698	234614544.6...	128.93.101.51	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
30703	234614545.1...	fe80::a0b3:b308:64d8:...	ff02::c	SSDP	153	M-SEARCH * HTTP/1.1
30704	234614545.1...	128.93.101.51	239.255.255.250	SSDP	139	M-SEARCH * HTTP/1.1
30705	234614545.1...	fe80::a0b3:8408:64d8:...	ff02::c	SSDP	181	M-SEARCH * HTTP/1.1
30706	234614545.1...	128.93.101.51	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
30707	234614545.1...	fe80::a0b3:8408:64d8:...	ff02::c	SSDP	179	M-SEARCH * HTTP/1.1
30710	234614545.2...	128.93.101.51	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
30720	234614546.2...	128.93.101.51	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
30738	234614549.2...	128.93.101.51	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
30743	234614552.2...	128.93.101.51	239.121.255.250	SSDP	175	M-SEARCH * HTTP/1.1
30750	234614555.2...	128.93.101.51	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1

A similar example of Wireshark display filter accepting an expression but it does not work as intended such as type DNS and **ip.addr !=10.96.203.66**



As you saw above the expression works but not intended.

As we have noticed these packet captures have different colours. So, what are these colours intended for...?

Don't get confused with a different type of colour packets. These colours are intended for

- Gray – TCP packets
- Black with red letters – TCP Packets with errors
- Green – HTTP Packets
- Light Blue – UDP Packets
- Pale Blue – ARP Packets
- Lavender – ICMP Packets
- Black with green letters – ICMP Packets with errors

Note: – Colourings can be changed under View -> Colouring Rules

Building Display Filter expressions.

we can build display filters that compare values using a different type of comparison operator.

For example to only display packets to or from the IP address 10.96.200.253 use **ip.addr==10.96.200.253** .

Wireshark display filter uses Boolean expressions, so we can specify values and chain them together. A complete list of available comparison operators is shown below.

Description	operator	Example
Equal or (eq)	==	ip.src==192.168.0.134
Not equal or (ne)	!=	ip.src!=192.168.0.134
Greater than or (gt)	>	frame.len > 20
Less than or (lt)	<	frame.len < 150
Greater than or equal to or (ge)	>=	frame.len >= 0x50
Less than or equal to or (le)	<=	frame.len <= 0x30
Bitwise_and	&	tcp.flags & 0x03
Logical AND or and	&&	ip.src==192.168.0.134 and tcp.flags.fin
Logical or		ip.src==192.168.0.134 or ip.src==192.168.1.1

Some Useful Filters

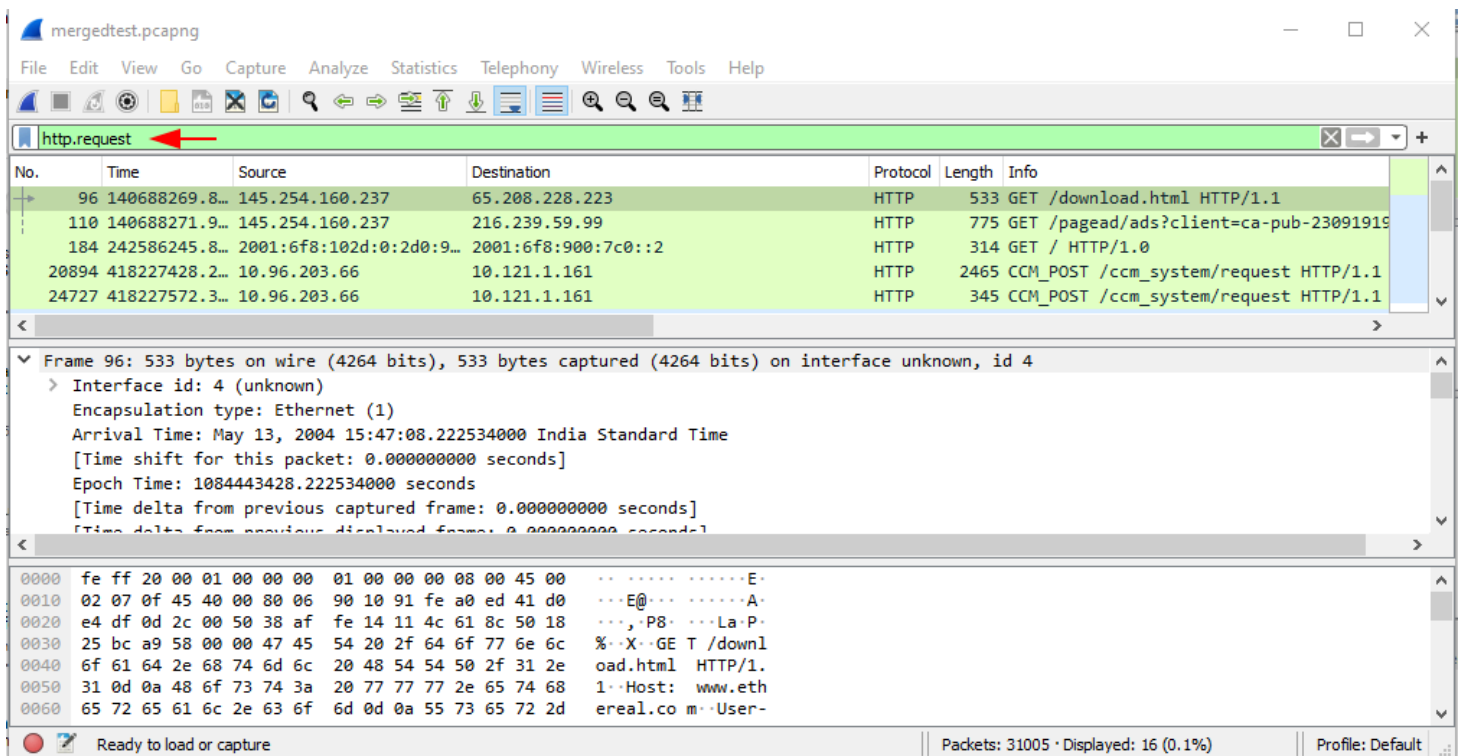
Here are some filter expressions that can be used as a way to quickly review web traffic.

Let's understand this with some sort of methods like how we are going to filter some infectious traffic.

Open the packet capture and apply the following filter: "http.request". This filter will show all HTTP post requests.

Also, you can find the total no. of packets at the bottom of the Wireshark screen that are 16 of these packets.

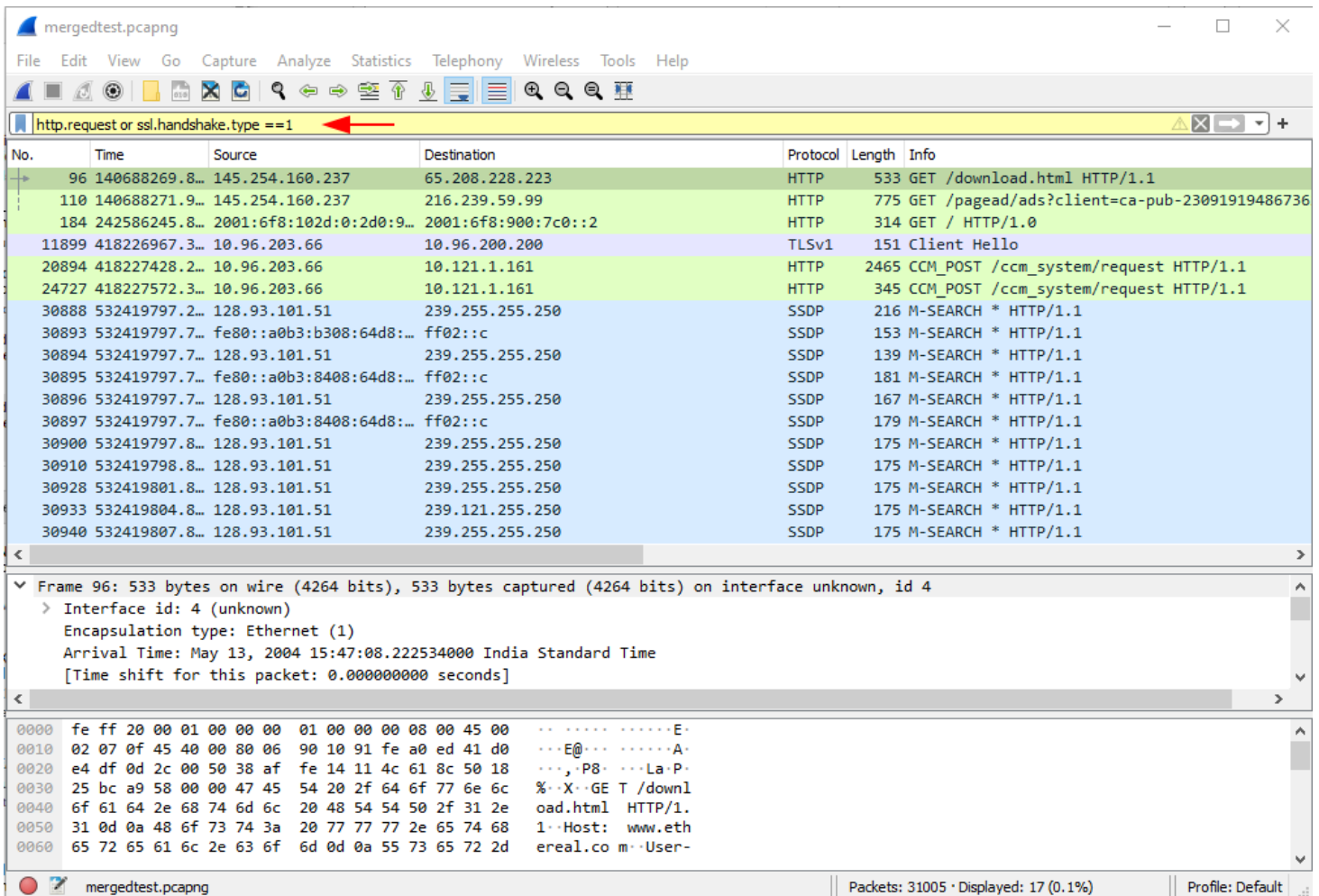
```
http.request
```

After that to reveal all the URLs for HTTP requests, Domain names we can use the following expression as a way to quickly review web traffic

`http.request` or `ssl.handshake.type == 1`

The value ***http.request*** reveals URLs for HTTP requests, and ***ssl.handshake.type == 1*** reveal domain names used in HTTPS or SSL/TLS traffic. Filtering with this display filter can outline the flow of events for the web traffic.



By modifying these types of filters, you can drill down the infectious traffic.

Hands-On Practice

Let's understand Wireshark with some sort of Questions

Q1. Find out the total no. of TCP syn packet for port 80

Answer: – To reveal all the TCP syn packets we can use the following expression as a way to quickly review web traffic for port 80. Also, you can find the total no. of packets at the bottom of the Wireshark screen that are 4 of these packets.

```
tcp.flags.syn == 1 and tcp.flags.ack == 0 && tcp.port == 80
```

mergedtest.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.flags.syn == 1 and tcp.flags.ack == 0 && tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
93	140688268.924...	145.254.160.237	65.208.228.223	TCP	62	3372 → 80 [SYN] Seq=0 Win=8760 Len=0 MSS=1460 SACK_PERM=1
181	242586245.802...	2001:6f8:102d:0:...	2001:6f8:900:7c0::2	TCP	94	59201 → 80 [SYN] Seq=0 Win=5760 Len=0 MSS=1440 SACK_PERM=1
20888	418227428.233...	10.96.203.66	10.199.1.161	TCP	66	54169 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK
24724	418227572.323...	10.96.203.66	10.121.1.161	TCP	66	54252 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK

< >

> Frame 93: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface unknown, id 4
 > Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe:ff:20:00:01:00 (fe:ff:20:00:01:00)
 > Internet Protocol Version 4, Src: 145.254.160.237, Dst: 65.208.228.223
 > Transmission Control Protocol, Src Port: 3372, Dst Port: 80, Seq: 0, Len: 0

```

0000  fe ff 20 00 01 00 00 00 01 00 00 00 08 00 45 00  ..  ....  ....E.
0010  00 30 0f 41 40 00 80 06 91 eb 91 fe a0 ed 41 d0  .0.A@  ....  ....A.
0020  e4 df 0d 2c 00 50 38 af fe 13 00 00 00 00 70 02  ...,P8  ....  ....p.
0030  22 38 c3 0c 00 00 02 04 05 b4 01 01 04 02      "8.  ....  ....
  
```

mergedtest.pcapng || Packets: 31005 · Displayed: 4 (0.0%) || Profile: Default

Q2. Filter out all the packet with the http response code 200.

Answer: – The value *http.response* reveals URLs for HTTP responses, and *HTTP status code 200* means success. The client has requested documents from the server. The server has replied to the client and given the client the documents. Filtering with this display filter can outline the flow of events for the web traffic.

http.response.code == 200

mergedtest.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.response.code == 200

No.	Time	Source	Destination	Protocol	Length	Info
119	140688272.879...	216.239.59.99	145.254.160.237	HTTP	214	HTTP/1.1 200 OK (text/html)
130	140688273.770...	65.208.228.223	145.254.160.237	HTTP/...	478	HTTP/1.1 200 OK
186	242586245.817...	2001:6f8:900:7c0...	2001:6f8:102d:0:2d0...	HTTP	901	HTTP/1.1 200 OK (text/html)
24735	418227572.484...	10.121.1.161	10.96.203.66	HTTP	1434	HTTP/1.1 200 OK Continuation

< >

> Frame 119: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface unknown, id 4
 > Ethernet II, Src: fe:ff:20:00:01:00 (fe:ff:20:00:01:00), Dst: Xerox_00:00:00 (00:00:01:00:00:00)
 > Internet Protocol Version 4, Src: 216.239.59.99, Dst: 145.254.160.237
 > Transmission Control Protocol, Src Port: 80, Dst Port: 3371, Seq: 1431, Ack: 722, Len: 160
 > [2 Reassembled TCP Segments (1590 bytes): #118(1430), #119(160)]

0000 00 00 01 00 00 00 fe ff 20 00 01 00 08 00 45 10E.
 0010 00 c8 85 cf 00 00 37 06 b6 12 d8 ef 3b 63 91 fe7....;c..
 0020 a0 ed 00 50 0d 2b 2e 6b 59 1a 36 c2 20 f9 50 18 ...P+.k Y.6. P.
 0030 7a e4 de 29 00 00 47 9a 63 ee 2f a7 b8 0c ab 3b z...)..G. c./...;
 0040 e5 ad 96 8a 89 c4 d9 1c 75 ee ab 25 e8 5c f7 c1u..%.\..
 0050 0b 7e 8e de fb 82 87 87 2b 92 25 40 a5 d9 79 50 ~.....+.%@..yP
 0060 cf b2 c0 97 6f c7 ab 25 0d 29 90 11 a5 24 c0 50o..%.)...\$.P
 0070 d7 2b 2d 9b ea 84 4a 8a 79 51 5c f0 76 9b 72 81 +----J. yQ\ v.r.
 0080 89 6d be 88 6a 09 f7 68 0a 15 f9 d9 5b ef a4 b0 .m..j..h[...

Frame (214 bytes) Reassembled TCP (1590 bytes) Uncompressed entity body (3608 bytes)

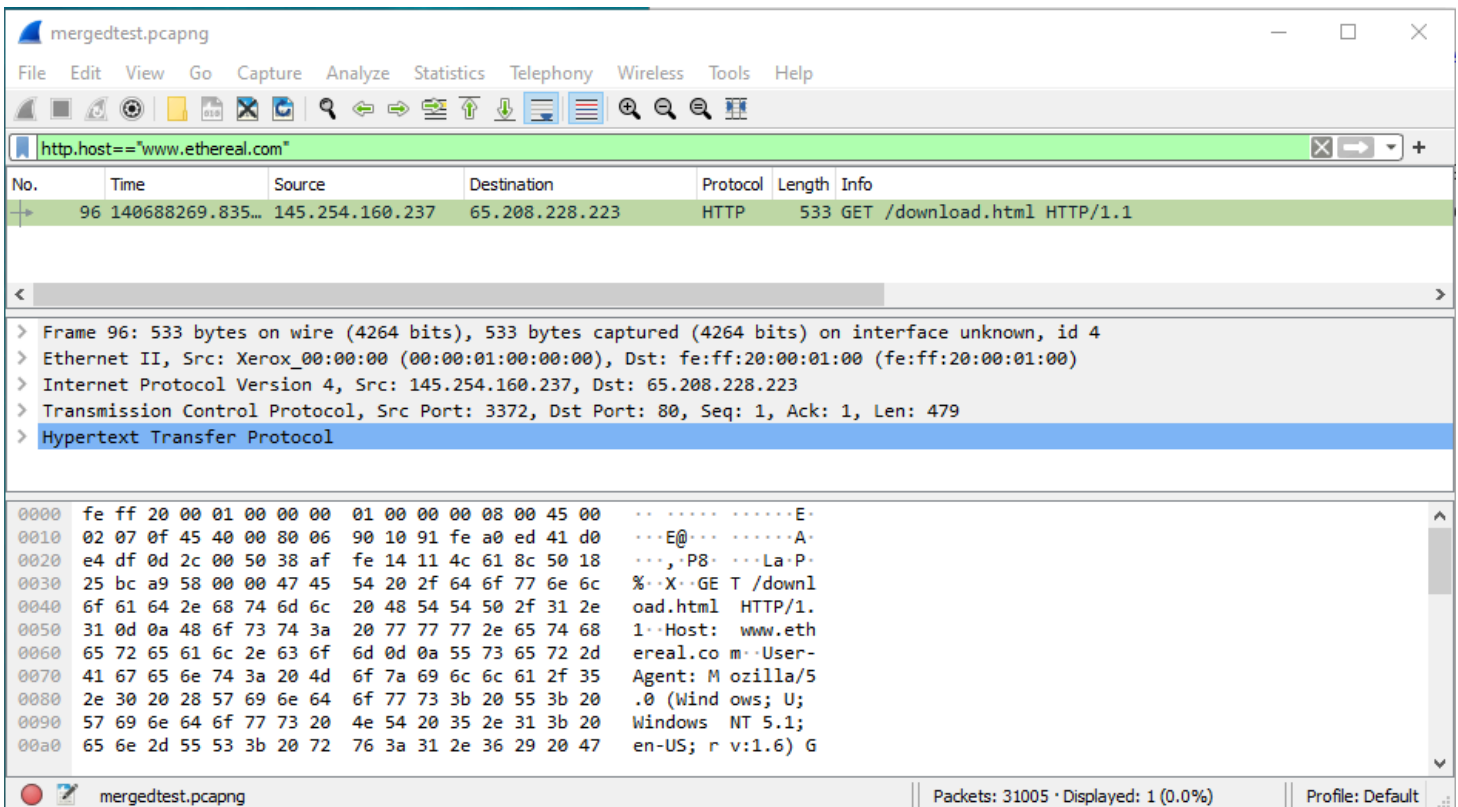
mergedtest.pcapng || Packets: 31005 · Displayed: 4 (0.0%) || Profile: Default

Q.3 Attacker try to download the malicious file from www.ethereal.com. Write down the filter to identify the http host.

Answer: – In this case, we have to find out the host who have visited the malicious website. As we know each website have own URL. So simply we can find out the host by using the following expression who have visited a malicious website.

`http.host=="www.ethereal.com"`

Or `http.host=="URL"`



The image shows a Wireshark capture of an HTTP GET request. The packet list shows a single packet (No. 96) from 145.254.160.237 to 65.208.228.223, protocol HTTP, length 533 bytes. The packet details pane shows the structure of the packet: Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
96	140688269.835...	145.254.160.237	65.208.228.223	HTTP	533	GET /download.html HTTP/1.1

Frame 96: 533 bytes on wire (4264 bits), 533 bytes captured (4264 bits) on interface unknown, id 4

- Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe:ff:20:00:01:00 (fe:ff:20:00:01:00)
- Internet Protocol Version 4, Src: 145.254.160.237, Dst: 65.208.228.223
- Transmission Control Protocol, Src Port: 3372, Dst Port: 80, Seq: 1, Ack: 1, Len: 479
- Hypertext Transfer Protocol

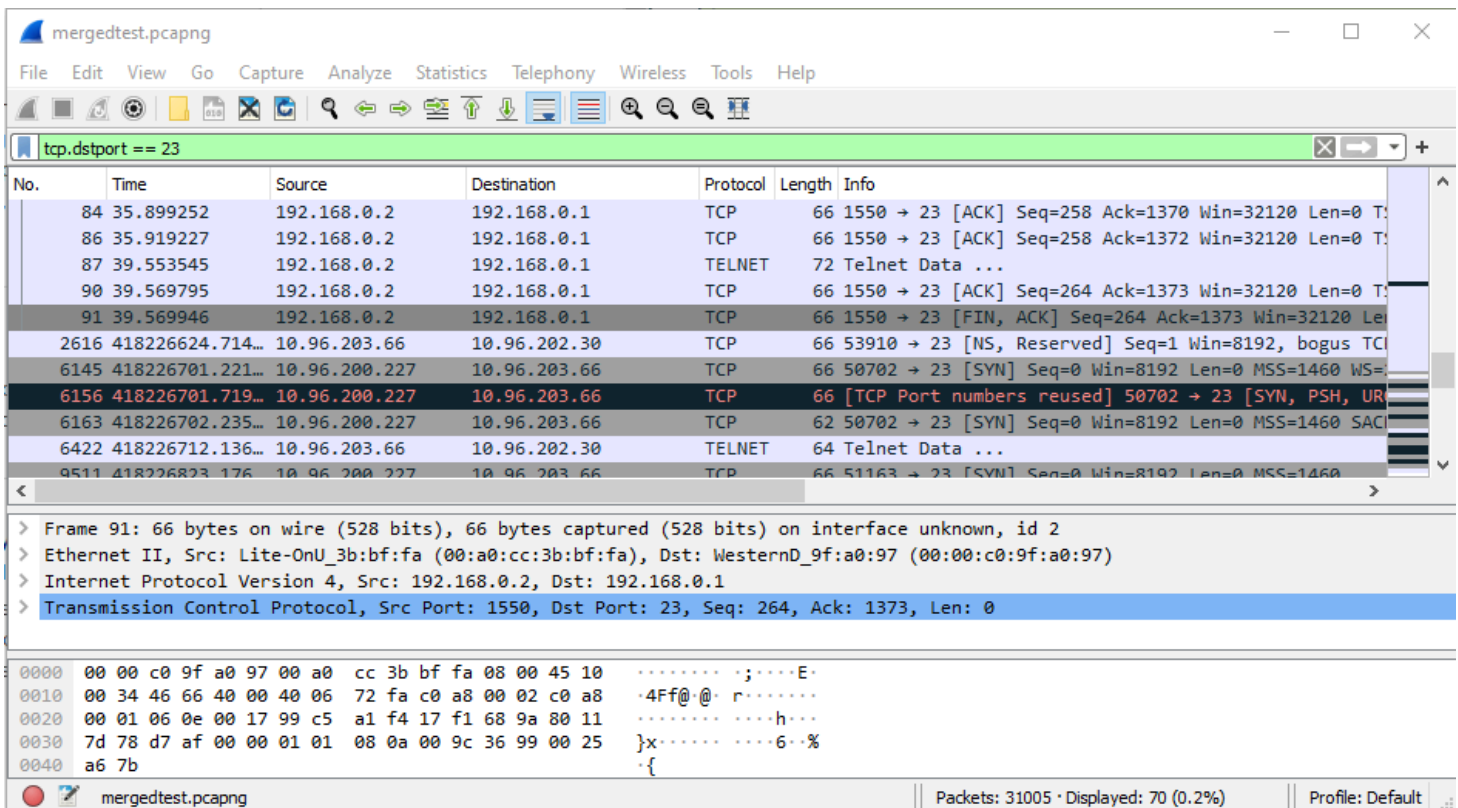
```

0000  fe ff 20 00 01 00 00 00 01 00 00 00 08 00 45 00  .. ...E.
0010  02 07 0f 45 40 00 80 06 90 10 91 fe a0 ed 41 d0  ...E@...A.
0020  e4 df 0d 2c 00 50 38 af fe 14 11 4c 61 8c 50 18  ...P8...LaP.
0030  25 bc a9 58 00 00 47 45 54 20 2f 64 6f 77 6e 6c  %..X..GE T /downl
0040  6f 61 64 2e 68 74 6d 6c 20 48 54 54 50 2f 31 2e  oad.html HTTP/1.
0050  31 0d 0a 48 6f 73 74 3a 20 77 77 77 2e 65 74 68  1..Host: www.eth
0060  65 72 65 61 6c 2e 63 6f 6d 0d 0a 55 73 65 72 2d  ereal.co m..User-
0070  41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35  Agent: M ozilla/5
0080  2e 30 20 28 57 69 6e 64 6f 77 73 3b 20 55 3b 20  .0 (Wind ows; U;
0090  57 69 6e 64 6f 77 73 20 4e 54 20 35 2e 31 3b 20  Windows NT 5.1;
00a0  65 6e 2d 55 53 3b 20 72 76 3a 31 2e 36 29 20 47  en-US; r v:1.6) G
  
```

Q4. Write down the filter to identify the destination port 23.

Answer: – Answer is quite simple... you can use the following expression to filter out the destination port 23

`tcp.dstport == 23`



The image shows a Wireshark capture filtered by the expression `tcp.dstport == 23`. The packet list shows multiple packets (No. 84, 86, 87, 90, 91, 2616, 6145, 6156, 6163, 6422, 9511) from 192.168.0.2 to 192.168.0.1, protocol TCP, length 66 bytes. The packet details pane shows the structure of the packet: Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
84	35.899252	192.168.0.2	192.168.0.1	TCP	66	1550 → 23 [ACK] Seq=258 Ack=1370 Win=32120 Len=0 T
86	35.919227	192.168.0.2	192.168.0.1	TCP	66	1550 → 23 [ACK] Seq=258 Ack=1372 Win=32120 Len=0 T
87	39.553545	192.168.0.2	192.168.0.1	TELNET	72	Telnet Data ...
90	39.569795	192.168.0.2	192.168.0.1	TCP	66	1550 → 23 [ACK] Seq=264 Ack=1373 Win=32120 Len=0 T
91	39.569946	192.168.0.2	192.168.0.1	TCP	66	1550 → 23 [FIN, ACK] Seq=264 Ack=1373 Win=32120 Len=0 T
2616	418226624.714...	10.96.203.66	10.96.202.30	TCP	66	53910 → 23 [NS, Reserved] Seq=1 Win=8192, bogus TC
6145	418226701.221...	10.96.200.227	10.96.203.66	TCP	66	50702 → 23 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=
6156	418226701.719...	10.96.200.227	10.96.203.66	TCP	66	[TCP Port numbers reused] 50702 → 23 [SYN, PSH, UR
6163	418226702.235...	10.96.200.227	10.96.203.66	TCP	62	50702 → 23 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SAC
6422	418226712.136...	10.96.203.66	10.96.202.30	TELNET	64	Telnet Data ...
9511	418226823.176...	10.96.200.227	10.96.203.66	TCP	66	51163 → 23 [SYN] Seq=0 Win=8192 Len=0 MSS=1460

Frame 91: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface unknown, id 2

- Ethernet II, Src: Lite-OnU_3b:bf:fa (00:a0:cc:3b:bf:fa), Dst: WesternD_9f:a0:97 (00:00:c0:9f:a0:97)
- Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.1
- Transmission Control Protocol, Src Port: 1550, Dst Port: 23, Seq: 264, Ack: 1373, Len: 0

```

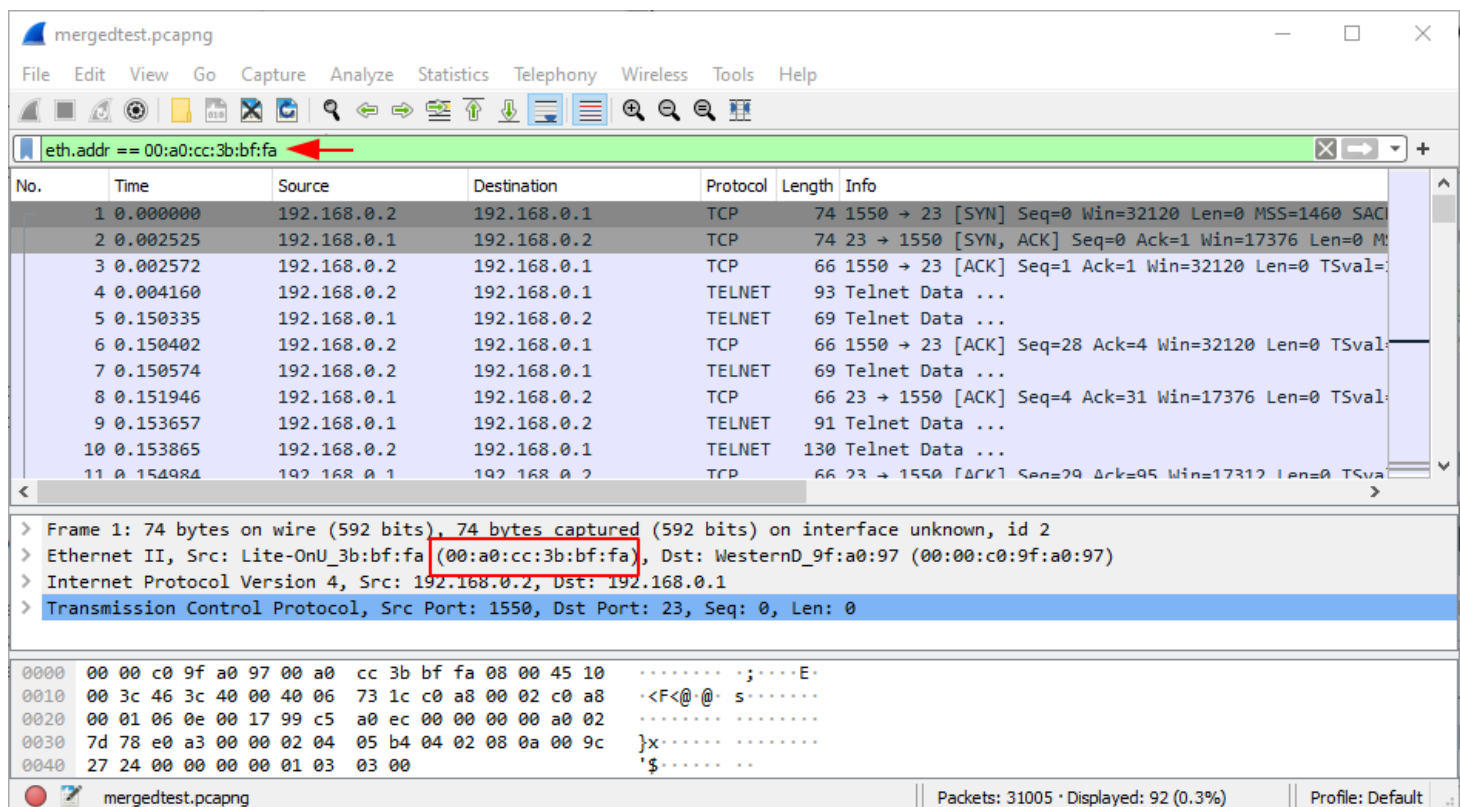
0000  00 00 c0 9f a0 97 00 a0 cc 3b bf fa 08 00 45 10  .....;....E.
0010  00 34 46 66 40 00 40 06 72 fa c0 a8 00 02 c0 a8  .4Ff@.@.r.....
0020  00 01 06 0e 00 17 99 c5 a1 f4 17 f1 68 9a 80 11  .....h.....
0030  7d 78 d7 af 00 00 01 01 08 0a 00 9c 36 99 00 25  }x.....6..%
0040  a6 7b                                           .{
  
```

Q5. Filter out the packets on behalf of the mac address.

Answer: – Apply the following expression to filter out the traffic for the specific mac address.

```
eth.addr == 00:a0:cc:3b:bf:fa
```

Or eth.addr == “mac addr”



Q6. Write down the filter to identify the IP address 10.96.203.66 for port 80 also including the IP address 10.121.1.161. find out the total no. of the packet.

Answer: – In this situation, we can create our custom filter for these types of random scenarios by using logical operators such as

```
ip.addr==10.96.203.66 and tcp.port==80 &&!(ip.addr==10.121.1.161)
```

By applying this filter, we can easily find out the total packets that are 3 of these packets.

mergedtest.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr==10.96.203.66 and tcp.port==80 &&! (ip.addr==10.121.1.161)

No.	Time	Source	Destination	Protocol	Length	Info
20888	418227428.233...	10.96.203.66	10.199.1.161	TCP	66	54169 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK...
20896	418227428.337...	37.0.1.161	10.96.203.66	TCP	258	80 → 54169 [PSH, ACK] Seq=1 Ack=1 Win=254 Len=204 [TCP se...
20900	418227428.533...	10.96.203.66	10.253.1.161	TCP	54	54169 → 80 [ACK] Seq=1 Ack=1 Win=16475 Len=0

< >

> Frame 20888: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface unknown, id 0
 > Ethernet II, Src: Dell_ed:b2:b5 (00:19:b9:ed:b2:b5), Dst: All-HSRP-routers_cb (00:00:0c:07:ac:cb)
 > Internet Protocol Version 4, Src: 10.96.203.66, Dst: 10.199.1.161
 > Transmission Control Protocol, Src Port: 54169, Dst Port: 80, Seq: 0, Len: 0

```

0000  00 00 0c 07 ac cb 00 19 b9 ed b2 b5 08 00 45 00  .....E.
0010  00 34 62 8f 40 00 80 06 00 00 0a 60 cb 42 0a c7  -4b.@...`B.
0020  01 a1 d3 99 00 50 69 c7 07 5d 00 00 00 00 80 02  ....Pi.].
0030  20 00 e1 e2 00 00 02 04 05 b4 01 03 03 02 01 01  .....
0040  04 02  ..
  
```

mergedtest.pcapng || Packets: 31005 · Displayed: 3 (0.0%) || Profile: Default

Q7. Find out the flag hidden in the provided pcap file that contains the user name for

1. All users of the Ftp session
2. Find out the credentials used for the Telnet session
3. Find out which command is being executed during the telnet session.

Answer: – Do it by yourself. By getting the flag to submit the flag in the comment section.

All the best.

You can download the pcap file from [here](#).

Source: <https://www.wireshark.org/>