EXP-4: Analyze Network Traffic using Wireshark Tool

Aim:

Analyze Network Traffic using Wireshark tool/ TCP dump tool

Components and Tools:

System: Desktop Computer/Laptop

Operating system: Windows/Linux

Tool: Wireshark

Description: (next slide)

Procedure: https://www.howtogeek.com/104278/how-to-use-wireshark-to-capture-filter-and-inspect-packets/

- **Getting Wireshark-Wireshark Installation**
- **Capturing Packets**
- **Color Coding**
- **Sample Captures**
- **Filtering Packets**
- **Inspecting Packets**

Lab Report: (last slide)

Description:

- Wireshark is a free and open source packet analyzer used for network troubleshooting and analysis.
- Wireshark can be useful for many different tasks, whether you are a **network engineer**, **security professional** or **system administrator**.
  - **Troubleshooting Network Connectivity**
    - Visually understand packet loss
    - Review TCP retransmission
    - Graph high latency packet responses
  - **Examination of Application Layer Sessions (even when encrypted by SSL/TLS see below)**
    - View full HTTP session, seeing all headers and data for both requests and responses
    - View Telnet sessions, see passwords, commands entered and responses
    - View SMTP or POP3 traffic, reading emails off the wire
  - **Troubleshoot DHCP issues with packet level data**
    - Examine DHCP client broadcast
    - DHCP offer with address and options
    - Client requests for offered address
    - Ack of server acknowledging the request
  - **Extract files from HTTP sessions**
    - Export objects from **HTTP** such as javascript, images, or even executables.
  - **Extract file from SMB sessions**
    - Similar to the **HTTP export option** but able to extract files transferred over **SMB(**Server Message Block protocol**)**, the ever present **Microsoft File Sharing protocol**.
  - **Detect and Examination of Malware**
    - Detect anomalous behaviour that could indicate malware
    - Search for unusual domains or IP address endpoints
    - Use IO graphs to discover regular connections (beacons) to command and control servers
    - Filter out the "normal" and find the unusual
    - Extract large DNS responses and other oddness which may indicate malware
  - **Examination of Port Scans and Other Vulnerability Scan types**
    - Understand what network traffic the vulnerability scanner is sending
    - Troubleshoot vulnerability checks to understand false positives and false negatives

Procedure:

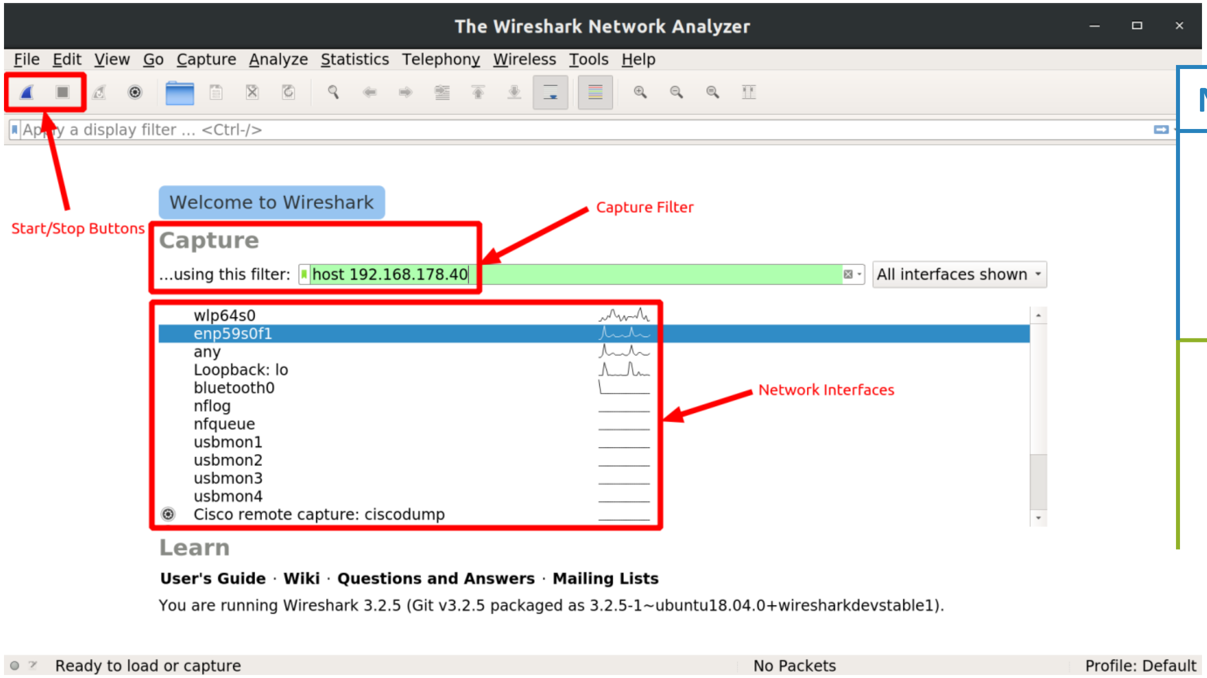
Wireshark Installation

- Preparation
  - To prepare for this activity:
    - Turn on your PC by pressing the power button (Windows will start automatically).
    - Log in if necessary.
- Activity 1 - Determine System Type
  - To determine system type:
    - Use msinfo32 (press Windows key, type "run", then type "Msinfo32") to display the system type. The system type will either be X86-based PC or X64-based PC. X86-based PC is a 32-bit system. X64-based PC is a 64-bit system.
    - Close msinfo32.
- Activity 2 - Download Wireshark
  - To download Wireshark:
    - Open a web browser.
    - Navigate to https://www.wireshark.org/download.html
    - Select Download Wireshark.
    - Select the Wireshark Windows Installer matching your system type, either 32-bit or 64-bit as determined in Activity 1. Save the program in the Downloads folder.
    - Close the web browser.

- Activity 3 - Install Wireshark
  - To install Wireshark:
    - Open Windows Explorer.
    - Select the Downloads folder.
    - Locate the version of Wireshark you downloaded in Activity 2. Double-click on the file to open it.
    - If you see a User Account Control dialog box, select Yes to allow the program to make changes to this computer.
    - Select Next > to start the Setup Wizard.
    - Review the license agreement. If you agree, select I Agree to continue.
    - Select Next > to accept the default components.
    - Select the shortcuts you would like to have created. Leave the file extensions selected. Select Next > to continue.
    - Select Next > to accept the default install location.
    - Select Install to begin installation.
    - Select Next > to install WinPcap.
    - Select Next > to start the Setup Wizard.
    - Review the license agreement. If you agree, select I Agree to continue.
    - Select Install to begin installation.
    - Select Finish to complete the installation of WinPcap.
    - Select Next > to continue with the installation of Wireshark.
    - Select Finish to complete the installation of Wireshark.

Capturing Packets

- After downloading and installing Wireshark, launch it and double-click the name of a network interface under Capture to start capturing packets on that interface.
- For example, if you want to capture traffic on your wireless network, click your wireless interface. You can configure advanced features by clicking Capture > Options, but this isn't necessary for now.


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Start/Stop Buttons
Welcome to Wireshark
Capture Filter
Capture
All interfaces shown
Network Interfaces
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists

…using this filter: host 192.168.178.40

wlp64s0
enp59s0f1
any
Loopback: lo
bluetooth0
nflog
nfqueue
usbmon1
usbmon2
usbmon3
usbmon4
Cisco remote capture: ciscodump

You are running Wireshark 3.2.5 (Git v3.2.5 packaged as 3.2.5-1~ubuntu18.04.0+wiresharkdevstable1).

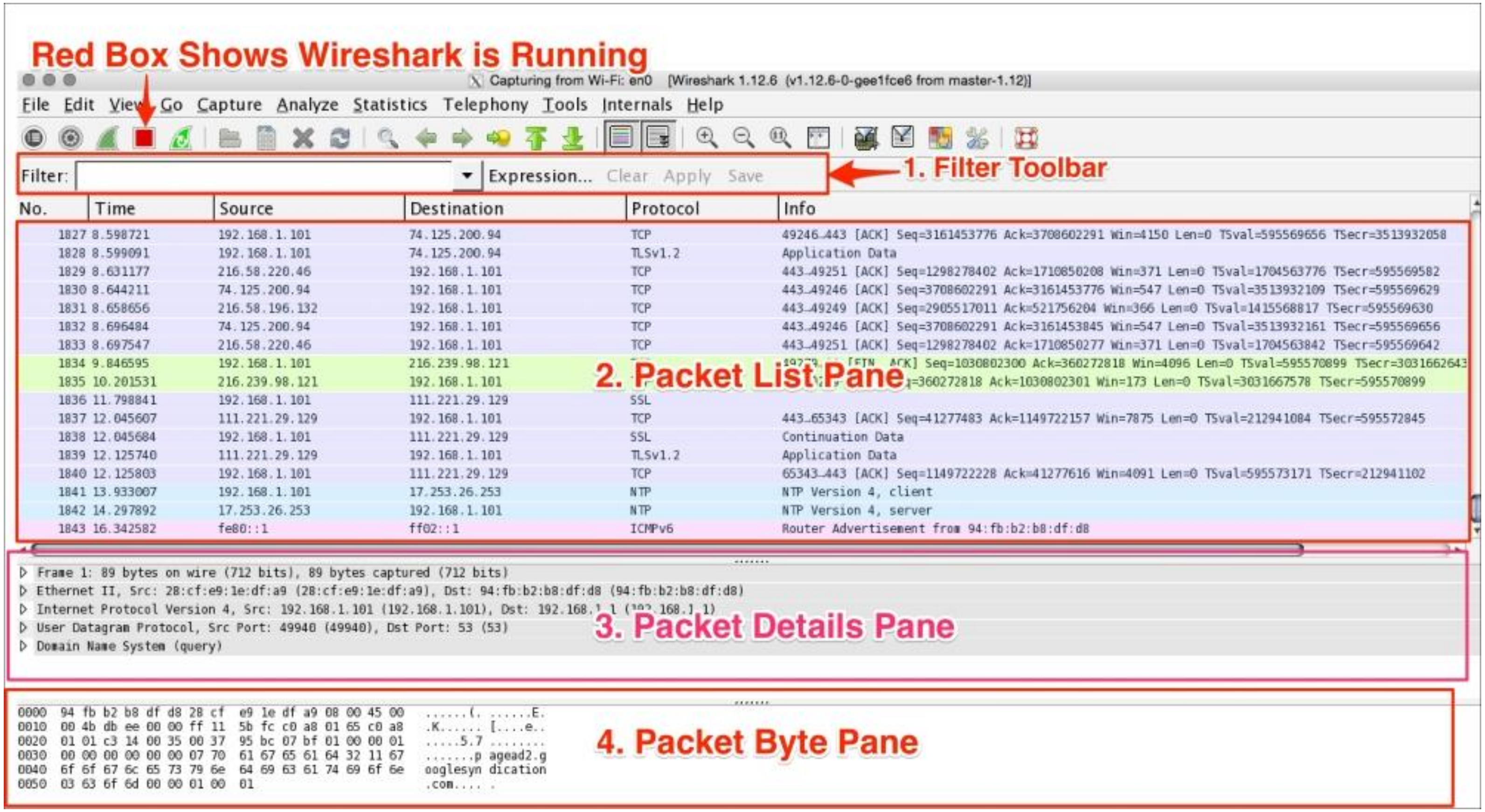
Ready to load or capture | No Packets | Profile: Default

Wireshark Capturing Modes

NAME	DESCRIPTION
Promiscuous mode	Sets interface to capture all packets on a network segment to which it is associated to
Monitor mode	Setup the wireless interface to capture all traffic it can receive (Unix/ Linux only)

As soon as you click the interface's name, you'll see the packets start to appear in real time. Wireshark captures each packet sent to or from your system.

- If you have promiscuous mode enabled---it's enabled by default---you'll also see all the other packets on the network instead of only packets addressed to your network adapter.
- To check if promiscuous mode is enabled, click Capture > Options and verify the "Enable promiscuous mode on all interfaces" checkbox is activated at the bottom of this window.

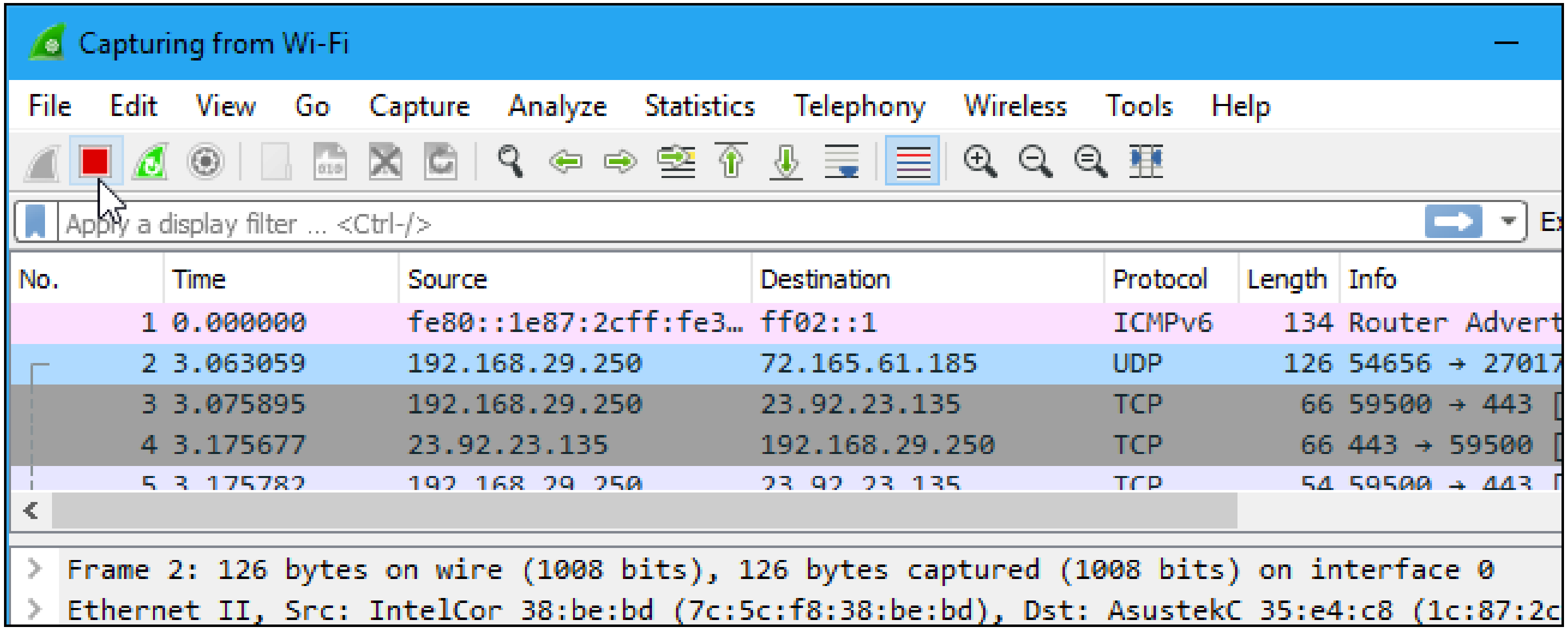

Red Box Shows Wireshark is Running
Capturing from Wi-Fi: en0 [Wireshark 1.12.6 (v1.12.6-0-gee1fce6 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
1. Filter Toolbar
2. Packet List Pane
3. Packet Details Pane
4. Packet Byte Pane

No.	Time	Source	Destination	Protocol	Info
1827	8.598721	192.168.1.101	74.125.200.94	TCP	49246→443 [ACK] Seq=3161453776 Ack=3708602291 Win=4150 Len=0 TSval=595569656 TSecr=3513932058
1828	8.599091	192.168.1.101	74.125.200.94	TLSv1.2	Application Data
1829	8.631177	216.58.220.46	192.168.1.101	TCP	443→49251 [ACK] Seq=1298278402 Ack=1710850208 Win=371 Len=0 TSval=1704563776 TSecr=595569582
1830	8.644211	74.125.200.94	192.168.1.101	TCP	443→49246 [ACK] Seq=3708602291 Ack=3161453776 Win=547 Len=0 TSval=3513932109 TSecr=595569629
1831	8.658656	216.58.196.132	192.168.1.101	TCP	443→49249 [ACK] Seq=2905517011 Ack=521756204 Win=366 Len=0 TSval=1415568817 TSecr=595569630
1832	8.696484	74.125.200.94	192.168.1.101	TCP	443→49246 [ACK] Seq=3708602291 Ack=3161453845 Win=547 Len=0 TSval=3513932161 TSecr=595569656
1833	8.697547	216.58.220.46	192.168.1.101	TCP	443→49251 [ACK] Seq=1298278402 Ack=1710850277 Win=371 Len=0 TSval=1704563842 TSecr=595569642
1834	9.846595	192.168.1.101	216.239.98.121	[illegible]	[illegible] [FIN, ACK] Seq=1030802300 Ack=360272818 Win=4096 Len=0 TSval=595570899 TSecr=3031662643
1835	10.201531	216.239.98.121	192.168.1.101	[illegible]	[illegible]=360272818 Ack=1030802301 Win=173 Len=0 TSval=3031667578 TSecr=595570899
1836	11.798841	192.168.1.101	111.221.29.129	SSL	
1837	12.045607	111.221.29.129	192.168.1.101	TCP	443→65343 [ACK] Seq=41277483 Ack=1149722157 Win=7875 Len=0 TSval=212941084 TSecr=595572845
1838	12.045684	192.168.1.101	111.221.29.129	SSL	Continuation Data
1839	12.125740	111.221.29.129	192.168.1.101	TLSv1.2	Application Data
1840	12.125803	192.168.1.101	111.221.29.129	TCP	65343→443 [ACK] Seq=1149722228 Ack=41277616 Win=4091 Len=0 TSval=595573171 TSecr=212941102
1841	13.933007	192.168.1.101	17.253.26.253	NTP	NTP Version 4, client
1842	14.297892	17.253.26.253	192.168.1.101	NTP	NTP Version 4, server
1843	16.342582	fe80::1	ff02::1	ICMPv6	Router Advertisement from 94:fb:b2:b8:df:d8

```
▷ Frame 1: 89 bytes on wire (712 bits), 89 bytes captured (712 bits)
▷ Ethernet II, Src: 28:cf:e9:1e:df:a9 (28:cf:e9:1e:df:a9), Dst: 94:fb:b2:b8:df:d8 (94:fb:b2:b8:df:d8)
▷ Internet Protocol Version 4, Src: 192.168.1.101 (192.168.1.101), Dst: 192.168.1.1 (192.168.1.1)
▷ User Datagram Protocol, Src Port: 49940 (49940), Dst Port: 53 (53)
▷ Domain Name System (query)
```

```
0000  94 fb b2 b8 df d8 28 cf  e9 1e df a9 08 00 45 00   ......(. ......E.
0010  00 4b db ee 00 00 ff 11  5b fc c0 a8 01 65 c0 a8   .K...... [....e..
0020  01 01 c3 14 00 35 00 37  95 bc 07 bf 01 00 00 01   .....5.7 ........
0030  00 00 00 00 00 00 07 70  61 67 65 61 64 32 11 67   .......p agead2.g
0040  6f 6f 67 6c 65 73 79 6e  64 69 63 61 74 69 6f 6e   ooglesyn dication
0050  03 63 6f 6d 00 00 01 00  01                        .com.... .
```

- Click the red "Stop" button near the top left corner of the window when you want to stop capturing traffic.

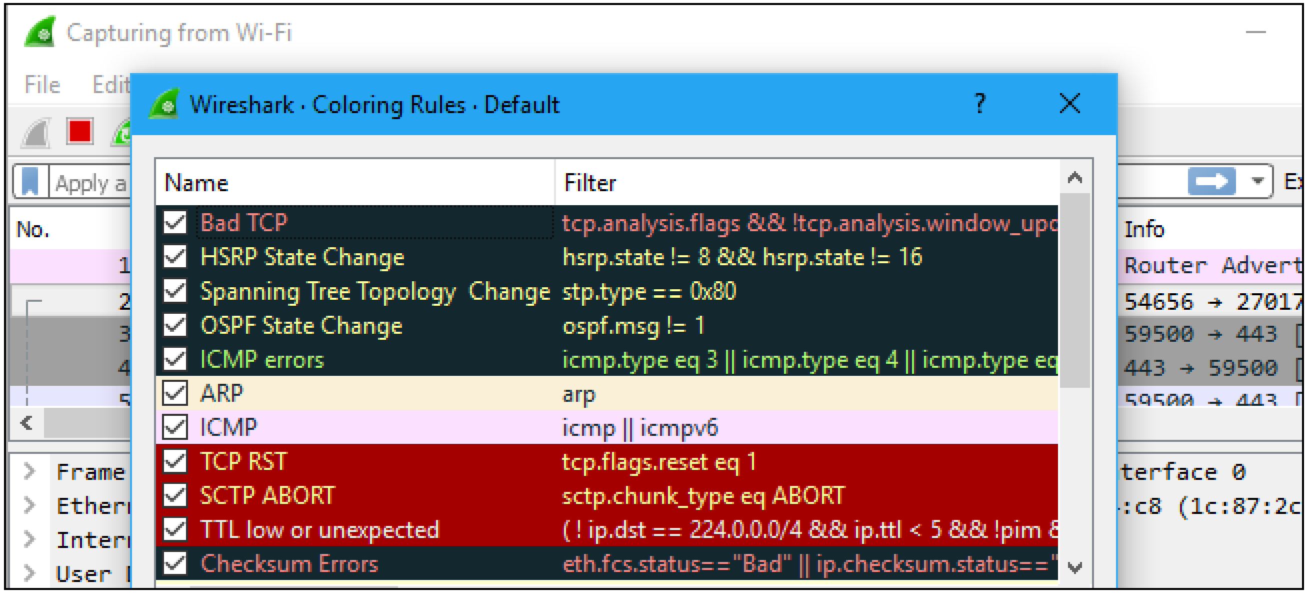


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::1e87:2cff:fe3…	ff02::1	ICMPv6	134	Router Advert
2	3.063059	192.168.29.250	72.165.61.185	UDP	126	54656 → 27017
3	3.075895	192.168.29.250	23.92.23.135	TCP	66	59500 → 443 [
4	3.175677	23.92.23.135	192.168.29.250	TCP	66	443 → 59500 [
5	3.175782	192.168.29.250	23.92.23.135	TCP	54	59500 → 443 [

```
> Frame 2: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits) on interface 0
> Ethernet II, Src: IntelCor 38:be:bd (7c:5c:f8:38:be:bd), Dst: AsustekC 35:e4:c8 (1c:87:2c
```

Color Coding

- You’ll probably see packets highlighted in a variety of different colors. Wireshark uses colors to help you identify the types of traffic at a glance.
- By default, light purple is TCP traffic, light blue is UDP traffic, and black identifies packets with errors---for example, they could have been delivered out of order.
- To view exactly what the color codes mean, click View > Coloring Rules. You can also customize and modify the coloring rules from here, if you like.

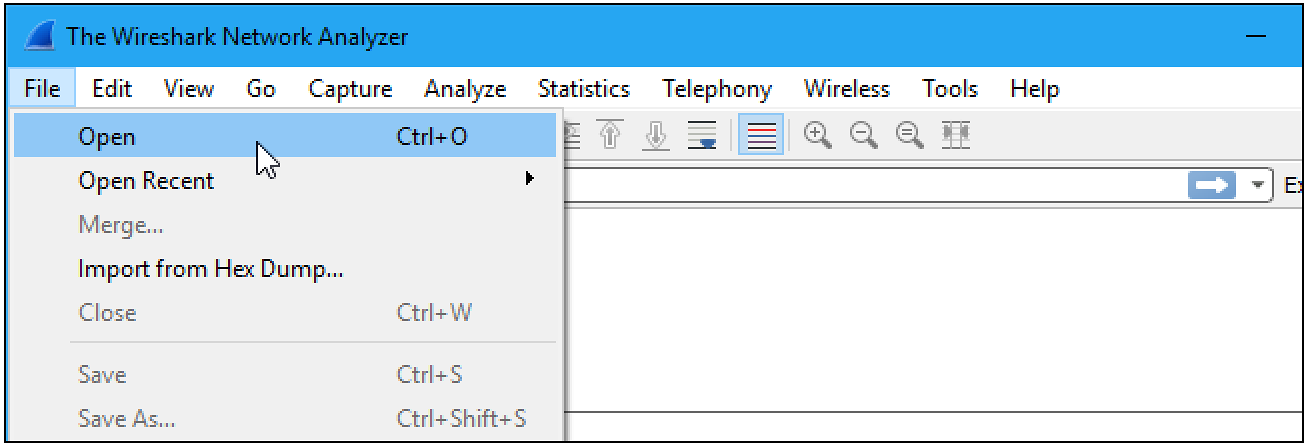


Wireshark · Coloring Rules · Default

Name	Filter
☑ Bad TCP	tcp.analysis.flags && !tcp.analysis.window_upd
☑ HSRP State Change	hsrp.state != 8 && hsrp.state != 16
☑ Spanning Tree Topology Change	stp.type == 0x80
☑ OSPF State Change	ospf.msg != 1
☑ ICMP errors	icmp.type eq 3 || icmp.type eq 4 || icmp.type eq
☑ ARP	arp
☑ ICMP	icmp || icmpv6
☑ TCP RST	tcp.flags.reset eq 1
☑ SCTP ABORT	sctp.chunk_type eq ABORT
☑ TTL low or unexpected	(! ip.dst == 224.0.0.0/4 && ip.ttl < 5 && !pim &
☑ Checksum Errors	eth.fcs.status=="Bad" || ip.checksum.status=="

Sample Captures

- If there's nothing interesting on your own network to inspect, Wireshark's wiki has you covered. The wiki contains a page of sample capture files that you can load and inspect. Click File > Open in Wireshark and browse for your downloaded file to open one.
- You can also save your own captures in Wireshark and open them later. Click File > Save to save your captured packets.



Default Columns In a Packet Capture Output

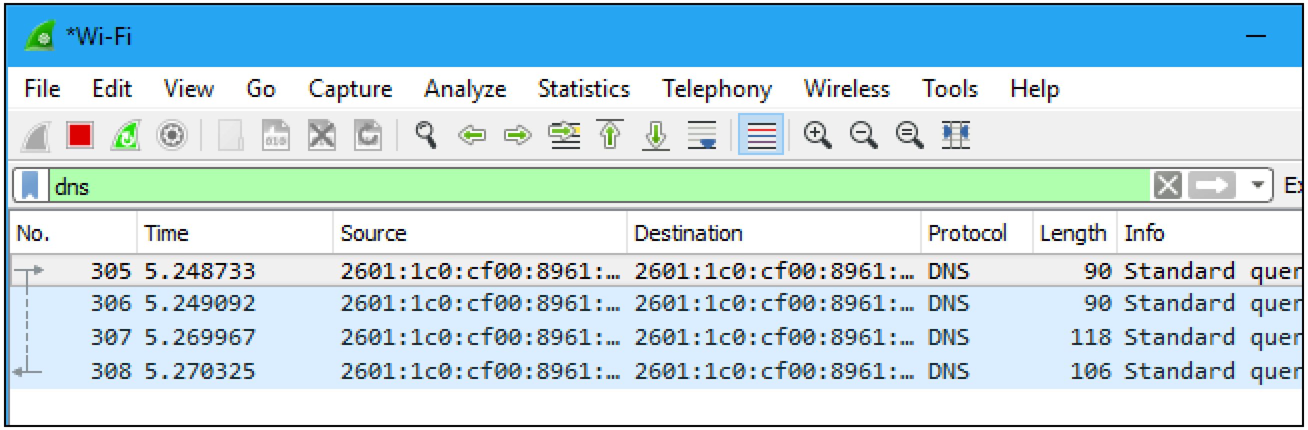
NAME	DESCRIPTION
No.	Frame number from the beginning of the packet capture
Time	Seconds from the first frame
Source (src)	Source address, commonly an IPv4, IPv6 or Ethernet address
Destination (dst)	Destination address
Protocol	Protocol used in the Ethernet frame, IP packet, or TC segment
Length	Length of the frame in bytes

Filtering Packets

Filter Types

NAME	DESCRIPTION
Capture filter	Filter packets during capture
Display filter	Hide packets from a capture display

- The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type “dns” and you’ll see only DNS packets. When you start typing, Wireshark will help you autocomplete your filter.



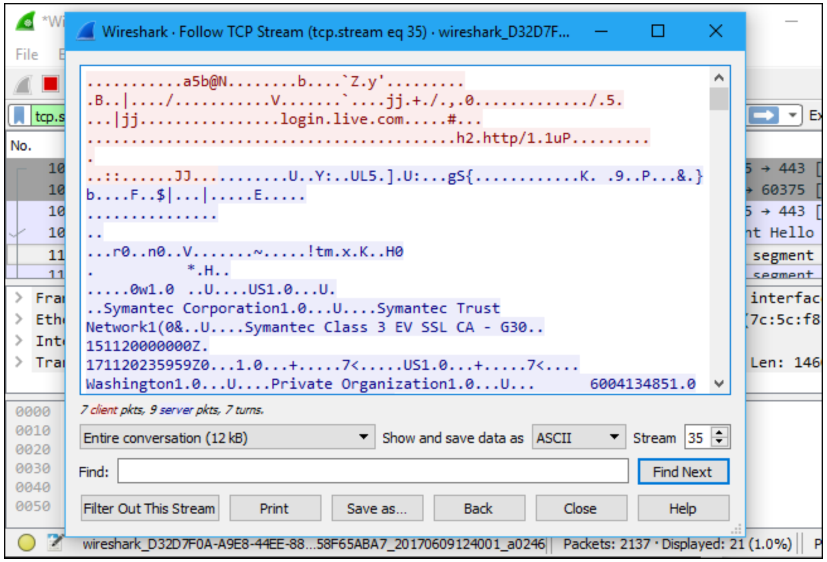
*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Ex

dns

```
No.    Time      Source               Destination          Protocol  Length  Info
  305  5.248733  2601:1c0:cf00:8961:… 2601:1c0:cf00:8961:… DNS           90  Standard quer
  306  5.249092  2601:1c0:cf00:8961:… 2601:1c0:cf00:8961:… DNS           90  Standard quer
  307  5.269967  2601:1c0:cf00:8961:… 2601:1c0:cf00:8961:… DNS          118  Standard quer
  308  5.270325  2601:1c0:cf00:8961:… 2601:1c0:cf00:8961:… DNS          106  Standard quer
```

- You can also click Analyze > Display Filters to choose a filter from among the default filters included in Wireshark. From here, you can add your own custom filters and save them to easily access them in the future.
- For more information on Wireshark's display filtering language, read the Building display filter expressions page in the official Wireshark documentation.

- Another interesting thing you can do is right-click a packet and select Follow > TCP Stream.
- You'll see the full TCP conversation between the client and the server. You can also click other protocols in the Follow menu to see the full conversations for other protocols, if applicable.
- Close the window and you'll find a filter has been applied automatically. Wireshark is showing you the packets that make up the conversation.

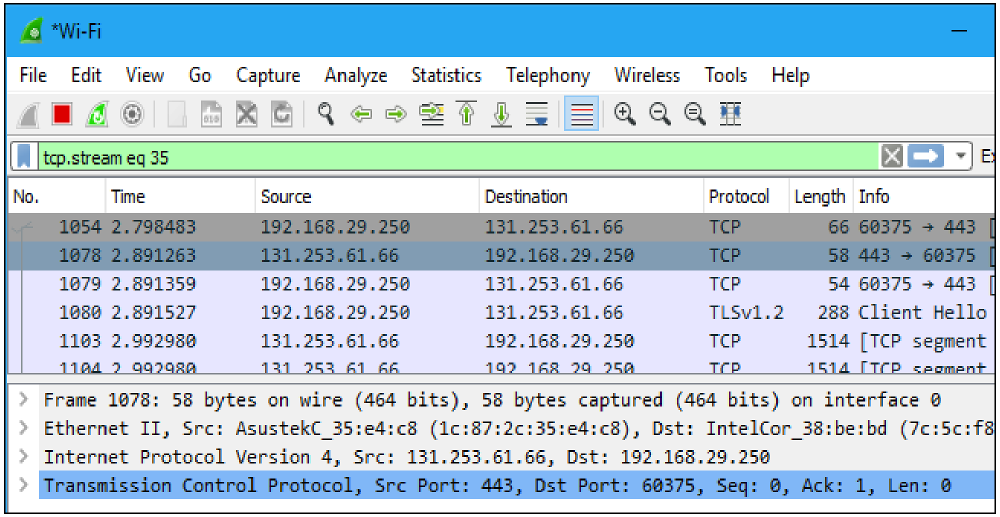

Wireshark · Follow TCP Stream (tcp.stream eq 35) · wireshark_D32D7F...
Find:
Find Next
Filter Out This Stream
Print
Save as...
Back
Close
Help

```
...........a5b@N........b....`Z.y'.........
.B..|..../...........V.......`....jj.+./.,.0............./.5.
...|jj...............login.live.com.....#...
..........................................h2.http/1.1uP.........
.
..:.......JJ...........U..Y:..UL5.].U:...gS{...........K. .9..P...&.}
b....F..$|...|.....E.....
...............
..
...r0..n0..V.......~.....!tm.x.K..H0
.           *.H..
.....0w1.0  ..U....US1.0...U.
..Symantec Corporation1.0...U....Symantec Trust
Network1(0&..U....Symantec Class 3 EV SSL CA - G30..
151120000000Z.
171120235959Z0...1.0...+.....7<.....US1.0...+.....7<....
Washington1.0...U....Private Organization1.0...U...      6004134851.0
```

7 client pkts, 9 server pkts, 7 turns.

Entire conversation (12 kB) — Show and save data as: ASCII — Stream: 35

wireshark_D32D7F0A-A9E8-44EE-88...58F65ABA7_20170609124001_a0246 | Packets: 2137 · Displayed: 21 (1.0%)



tcp.stream eq 35

```
No.    Time      Source           Destination      Protocol  Length Info
  1054 2.798483  192.168.29.250   131.253.61.66    TCP         66 60375 → 443 [
  1078 2.891263  131.253.61.66    192.168.29.250   TCP         58 443 → 60375 [
  1079 2.891359  192.168.29.250   131.253.61.66    TCP         54 60375 → 443 [
  1080 2.891527  192.168.29.250   131.253.61.66    TLSv1.2    288 Client Hello
  1103 2.992980  131.253.61.66    192.168.29.250   TCP       1514 [TCP segment
  1104 2.992980  131.253.61.66    192.168.29.250   TCP       1514 [TCP segment

> Frame 1078: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface 0
> Ethernet II, Src: AsustekC_35:e4:c8 (1c:87:2c:35:e4:c8), Dst: IntelCor_38:be:bd (7c:5c:f8
> Internet Protocol Version 4, Src: 131.253.61.66, Dst: 192.168.29.250
> Transmission Control Protocol, Src Port: 443, Dst Port: 60375, Seq: 0, Ack: 1, Len: 0
```

Filtering Packets (Display Filters)

OPERATOR	DESCRIPTION	EXAMPLE
eq or ==	Equal	ip.dest == 192.168.1.1
ne or !=	Not equal	ip.dest != 192.168.1.1
gt or >	Greater than	frame.len > 10
it or <	less than	frame.len < 10
ge or >=	Greater than or equal	frame.len >= 10
le or <=	Less than or equal	frame.len <= 10

Logical Operators

OPERATOR	DESCRIPTION	EXAMPLE
and or &&	Logical AND	All the conditions should match
or or ||	Logical OR	Either all or one of the conditions should match
xor or ^^	Logical XOR	Exclusive alterations - only one of the two conditions should match not both
not or !	Not (Negation)	Not equal to
[n] […]	Substring operator	Filter a specific word or text

Miscellaneous

NAME	DESCRIPTION
Slice Operator	[…] - Range of values
Membership Operator	{} - In
CTRL+E	Start/Stop Capturing

Capture Filter Syntax

SYNTAX	PROTOCOL	DIRECTION	HOSTS	VALUE	LOGICAL OPERATOR	EXPRESSIONS
Example	tcp	src	192.168.1.1	80	and	tcp dst 202.164.30.1

Display Filter Syntax

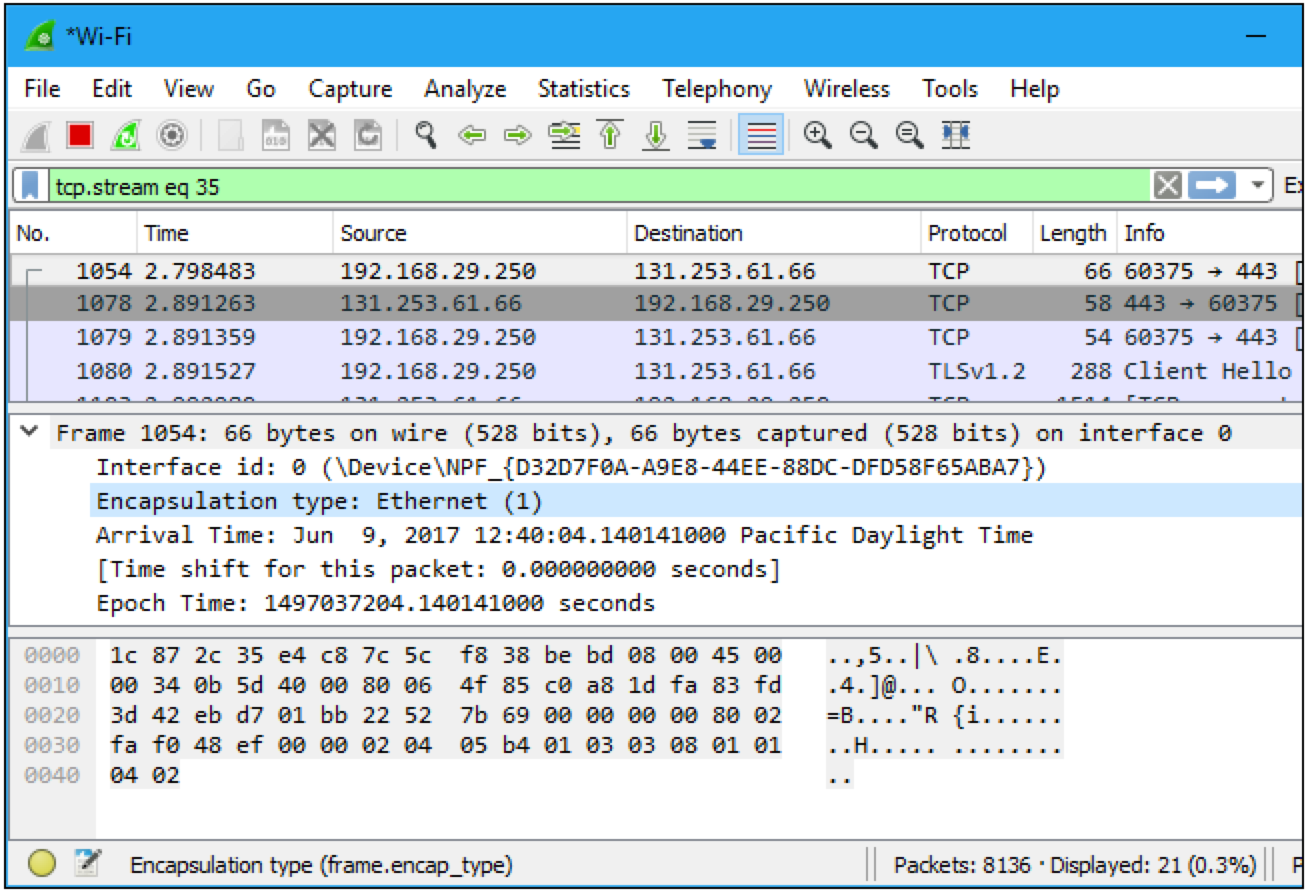
SYNTAX	PROTOCOL	STRING 1	STRING 2	COMPARISON OPERATOR	VALUE	LOGICAL OPERATOR	EXPRESSIONS
Example	http	dest	ip	==	192.168.1.1	and	tcp port

Common Filtering Commands

USAGE	FILTER SYNTAX
Wireshark Filter by IP	ip.add == 10.10.50.1
Filter by Destination IP	ip.dest == 10.10.50.1
Filter by Source IP	ip.src == 10.10.50.1
Filter by IP range	ip.addr >= 10.10.50.1 and ip.addr <=10.10.50.100
Filter by Multiple Ips	ip.addr == 10.10.50.1 and ip.addr == 10.10.50.100
Filter out IP adress	! (ip.addr == 10.10.50.1)
Filter subnet	ip.addr == 10.10.50.1/24
Filter by port	tcp.port == 25
Filter by destination port	tcp.dstport == 23
Filter by ip adress and port	ip.addr == 10.10.50.1 and Tcp.port == 25
Filter by URL	http.host == "host name"
Filter by time stamp	frame.time >= "June 02, 2019 18:04:00"
Filter SYN flag	Tcp.flags.syn == 1 and tcp.flags.ack ==0
Wireshark Beacon Filter	wlan.fc.type_subtype = 0x08
Wireshark broadcast filter	eth.dst == ff:ff:ff:ff:ff:ff
Wireshark multicast filter	(eth.dst[0] & 1)
Host name filter	ip.host = hostname
MAC address filter	eth.addr == 00:70:f4:23:18:c4
RST flag filter	tcp.flag.reset == 1

Inspecting Packets

- Click a packet to select it and you can dig down to view its details.



tcp.stream eq 35

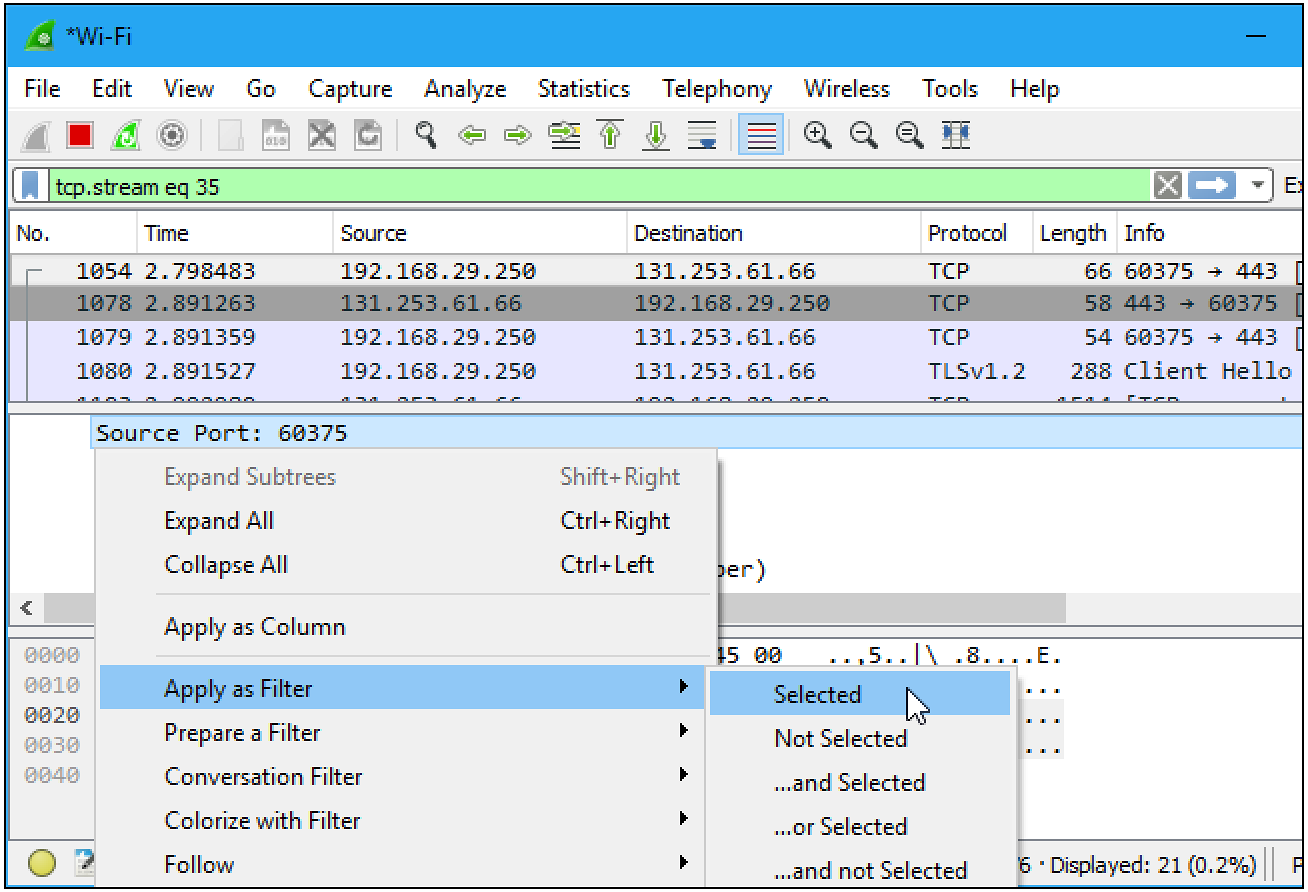
```
No.        Time        Source            Destination       Protocol   Length  Info
     1054  2.798483    192.168.29.250    131.253.61.66     TCP            66  60375 → 443 [
     1078  2.891263    131.253.61.66     192.168.29.250    TCP            58  443 → 60375 [
     1079  2.891359    192.168.29.250    131.253.61.66     TCP            54  60375 → 443 [
     1080  2.891527    192.168.29.250    131.253.61.66     TLSv1.2       288  Client Hello
```

```
Frame 1054: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
    Interface id: 0 (\Device\NPF_{D32D7F0A-A9E8-44EE-88DC-DFD58F65ABA7})
    Encapsulation type: Ethernet (1)
    Arrival Time: Jun  9, 2017 12:40:04.140141000 Pacific Daylight Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1497037204.140141000 seconds
```

```
0000  1c 87 2c 35 e4 c8 7c 5c  f8 38 be bd 08 00 45 00   ..,5..|\ .8....E.
0010  00 34 0b 5d 40 00 80 06  4f 85 c0 a8 1d fa 83 fd   .4.]@... O.......
0020  3d 42 eb d7 01 bb 22 52  7b 69 00 00 00 00 80 02   =B...."R {i......
0030  fa f0 48 ef 00 00 02 04  05 b4 01 03 03 08 01 01   ..H..... ........
0040  04 02                                              ..
```

Encapsulation type (frame.encap_type)   Packets: 8136 · Displayed: 21 (0.3%)

- You can also create filters from here -- just right-click one of the details and use the Apply as Filter submenu to create a filter based on it.



tcp.stream eq 35

No.	Time	Source	Destination	Protocol	Length	Info
1054	2.798483	192.168.29.250	131.253.61.66	TCP	66	60375 → 443 [
1078	2.891263	131.253.61.66	192.168.29.250	TCP	58	443 → 60375 [
1079	2.891359	192.168.29.250	131.253.61.66	TCP	54	60375 → 443 [
1080	2.891527	192.168.29.250	131.253.61.66	TLSv1.2	288	Client Hello

Source Port: 60375

```
0000   45 00   ...,5..|\ .8....E.
0010           ...
0020           ...
0030           ...
0040
```

6 · Displayed: 21 (0.2%)

Report

Part I	
1	What is Wireshark and how do you use it?
2	Can Wireshark see texts?
3	What are the 2 types of filters used by Wireshark?
4	Can Wireshark see incognito?
5	How do you capture packets in Wireshark?
6	What can hackers do with Wireshark?

Part II	
1	Is the frame an outgoing or an incoming frame?
2	Source IP address of the network-layer header in the frame:
3	Destination IP address of the network-layer header in the frame:
4	Total number of bytes in the whole frame:
5	Number of bytes in the Ethernet (data-link layer) header:
6	Number of bytes in the IP header:
7	Number of bytes in the TCP header:
8	Total bytes in the message (at the application layer):