Ex.No.3 Perform Network Analysis

Aim: To monitor live network capturing packet and analyze over live network using Wireshark and Kali Linux.

Wireshark:

Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible. The most common task in Network Forensics is packet analysis which can be commonly done with a packet analyzer. Packet analyzers will capture the traffic, decodes raw data, and analyzes each packet based on protocols within it. Common packet analyzers are Wireshark and tcpdump in kali linux. Get the latest copy of the program from the Wireshark website at https://www.wireshark.org/download.html

Default columns in Wireshark:

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 TCP segment
Length	Length of the frame in bytes

Wireshark capturing modes:

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)

Wireshark filter types:

Capture filter	Filter packets during capture
Display Filter	Hide Packets from a capture display

Wireshark protocol values:

ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp and udp

As a default, in kali linux, the interface is given as eth0 instead of ether.

Wireshark Syntax usage for filters:

Filter type	Fields					
Capture filter	Protocol	Direction	Host	Value	Logical operator	Expression
Example	tcp	src	192.168.1.1	80	and	tcp dest 202.164.30.1

Display filter	Protocol	String1	String2	Comparison operator	Value	Logical operator	Expression
Example	http	dest	ip	==	192.168.1.1	and	tcp port

Main tool bar items in Wireshark:

Toolbar Icon	Toolbar Item	Menu Item	Description
	Start	Capture → Start	Uses the same packet capturing options as the previous session, or uses defaults if no options were set
	Stop	Capture → Stop	Stops currently active capture
	Restart	Capture → Restart	Restarts active capture session
	Options...	Capture → Options...	Opens "Capture Options" dialog box
	Open...	File → Open...	Opens "File open" dialog box to load a capture for viewing
	Save As...	File → Save As...	Save current capture file
	Close	File → Close	Close current capture file
	Reload	View → Reload	Reloads current capture file
	Find Packet...	Edit → Find Packet...	Find packet based on different criteria
	Go Back	Go → Go Back	Jump back in the packet history

Toolbar Icon	Toolbar Item	Menu Item	Description
	Go Forward	Go → Go Forward	Jump forward in the packet history
	Go to Packet...	Go → Go to Packet...	Go to specific packet
	Go To First Packet	Go → First Packet	Jump to first packet of the capture file
	Go To Last Packet	Go → Last Packet	Jump to last packet of the capture file
	Auto Scroll in Live Capture	View → Auto Scroll in Live Capture	Auto scroll packet list during live capture
	Colorize	View → Colorize	Colorize the packet list (or not)
	Zoom In	View → Zoom In	Zoom into the packet data (increase the font size)
	Zoom Out	View → Zoom Out	Zoom out of the packet data (decrease the font size)
	Normal Size	View → Normal Size	Set zoom level back to 100%
	Resize Columns	View → Resize Columns	Resize columns, so the content fits to the width

Common filtering commands in Wireshark:

Usage	Filter syntax
Wireshark Filter by IP	ip.addr == 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 address	!(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 address and port	ip.addr == 10.10.50.1 and Tcp.port == 25

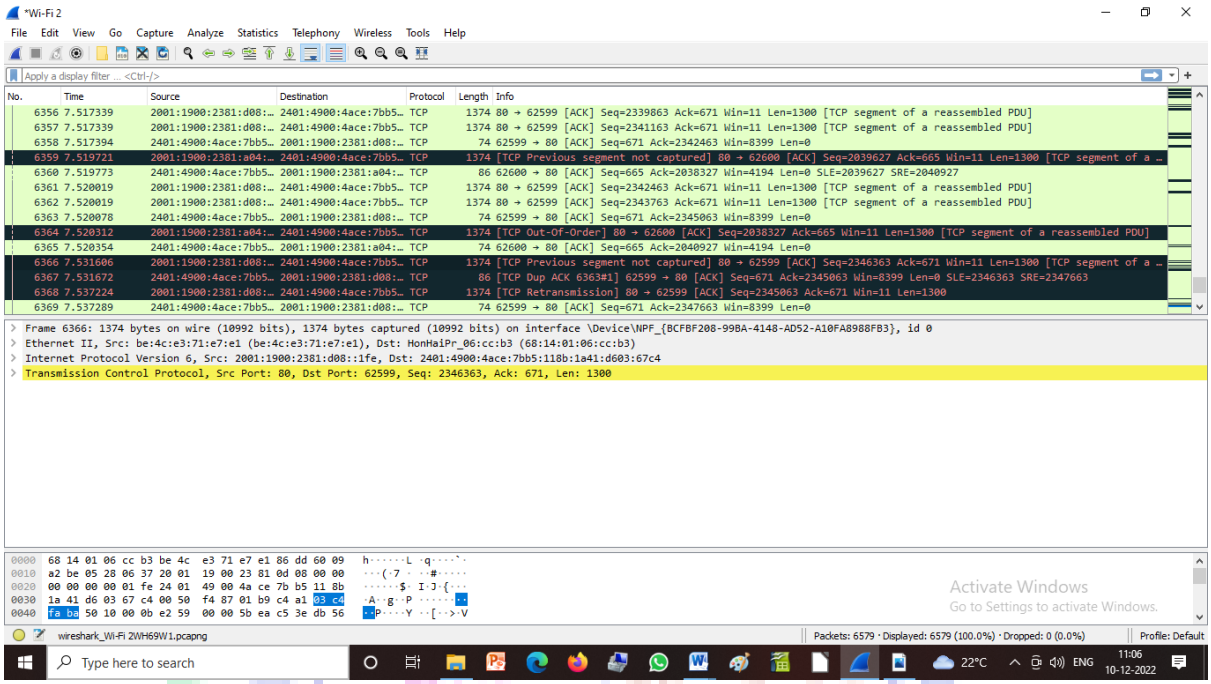
Usage	Filter syntax
Filter by URL	http.host == "host name"
Filter by time stamp	frame.time >= "June 02, 2019 18:04:00"
Filter SYN flag	tcp.flags.syn == 1
	tcp.flags.syn == 1 and tcp.flags.ack == 0
Wireshark Beacon Filter	wlan.fc.type_subtype = 0x08
Wireshark broadcast filter	eth.dst == ff:ff:ff:ff:ff:ff
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.flags.reset == 1

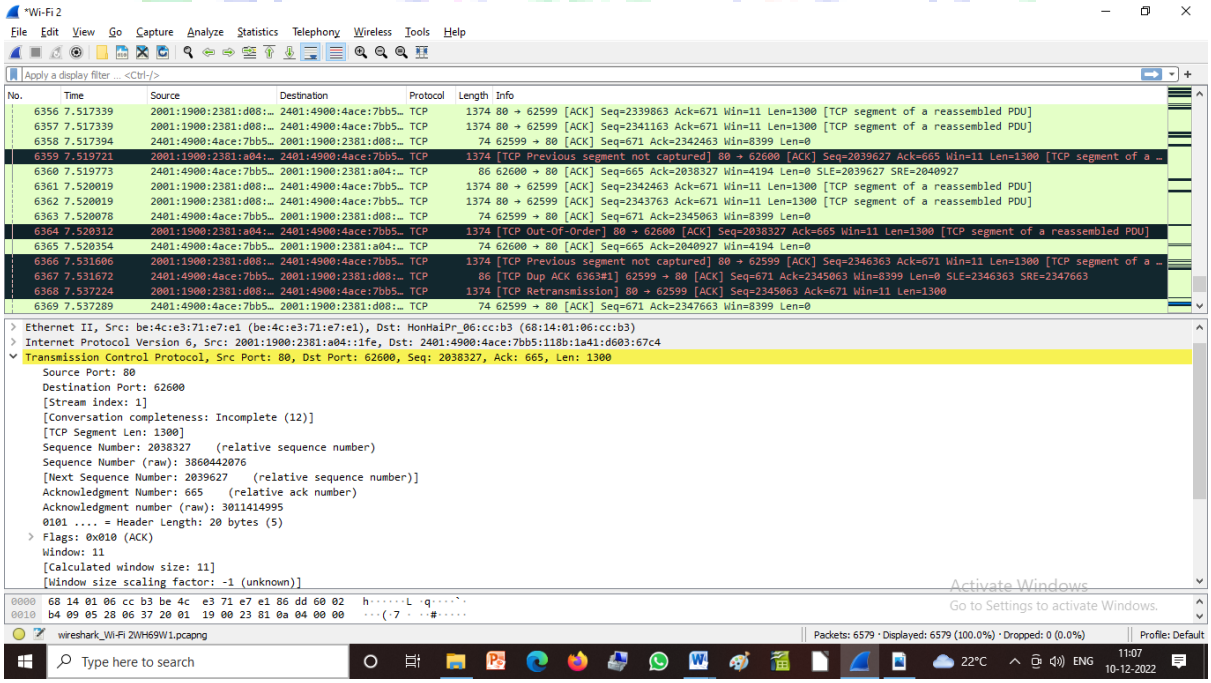
Procedure:

Capturing live network:

1. Select Interfaces from Capture menu.
2. Select Options choice from the drop down menu.
3. Select the required network.
4. Click the start button to capture the network packets
5. Click the red squared stop button to stop the network packet capturing.
6. Select File menu and select Save option. Give the name for the file as test1.
7. To analyse the network, open the saved file using Wreshark.
8. Packets of the given network are displayed in the packet list pane.
9. Select the packet which we want to analyse.
10. Expand the given packet and analyse n tree view or byte view.

Output from Wireshark:



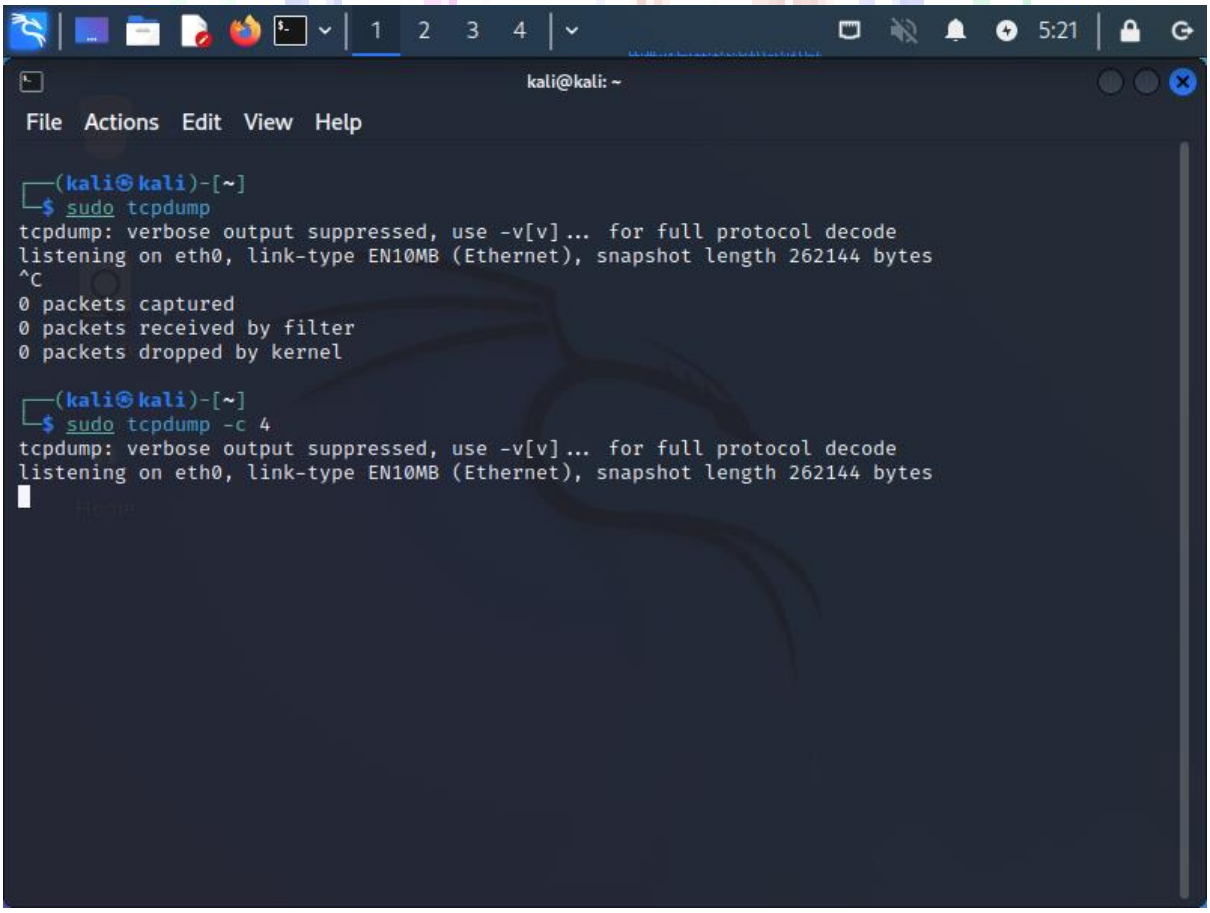


Network analysis using Kali Linux

Procedure to analyzing the given network:

1. Open kali linux
2. Open terminal
3. To capture the packets of current network interface, type as sudo tcpdump
4. To capture specific number of packets, type as sudo tcpdump –c 4. It will capture 4 packets from the network
5. To print captured packets in ASCII format, type as sudo tcpdump –c 4 –A
6. To print all available interfaces to our system, type as sudo tcpdump –D
7. To write the captured packets into a file, type as sudo tcpdump –w one.pcap
8. To read the packet information from the stored file, type as sudo tcpdump –r one.pcap
9. To capture packets with IP address use sudo tcpdump –n
10. To capture only tcp packets, type as sudo tcpdump tcp

Output using Kali Linux:



```
0 packets captured
0 packets received by filter
0 packets dropped by kernel

┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -c 4
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
05:22:15.425148 IP6 fe80::8b22:c18a:231d:9da9 > ip6-allrouters: ICMP6, router solicitation, len
gth 8
05:22:15.471237 IP 10.0.2.15.36635 > mahendraadds.mahendra.local.domain: 3814+ PTR? 9.a.d.9.d.1
.3.2.a.8.1.c.2.2.b.8.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa. (90)
05:22:15.472605 IP mahendraadds.mahendra.local.domain > 10.0.2.15.36635: 3814 NXDomain 0/1/0 (1
54)
05:22:15.594208 IP 10.0.2.15.35869 > mahendraadds.mahendra.local.domain: 34918+ PTR? 2.0.0.10.i
n-addr.arpa. (39)
4 packets captured
7 packets received by filter
0 packets dropped by kernel

┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -A
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel

┌──(kali㉿kali)-[~]
└─$
```

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -c 4 -A
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
05:29:51.626086 IP6 fe80::8b22:c18a:231d:9da9 > ip6-allrouters: ICMP6, router solicitation, len
gth 8
`.....:..........".."#......................o.....
05:29:51.703950 IP 10.0.2.15.57125 > mahendraadds.mahendra.local.domain: 41280+ PTR? 9.a.d.9.d.
1.3.2.a.8.1.c.2.2.b.8.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa. (90)
E..v|.@.@..c
...
....%.5.b...@...........9.a.d.9.d.1.3.2.a.8.1.c.2.2.b.8.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arp
a.....
05:29:51.705192 IP mahendraadds.mahendra.local.domain > 10.0.2.15.57125: 41280 NXDomain 0/1/0 (
154)
E.......@.d
...
....5.%...E.@...........9.a.d.9.d.1.3.2.a.8.1.c.2.2.b.8.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arp
a......L.........4.b.ip6-servers.P.nstld.iana.org.x..6......... :.....
05:29:51.806372 IP 10.0.2.15.35943 > mahendraadds.mahendra.local.domain: 45707+ PTR? 2.0.0.10.i
n-addr.arpa. (39)
E..C..@.@.o.
...
....g.5./.Q.............2.0.0.10.in-addr.arpa.....
4 packets captured
7 packets received by filter
0 packets dropped by kernel

┌──(kali㉿kali)-[~]
└─$
```

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -D
1.eth0 [Up, Running, Connected]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
7.dbus-system (D-Bus system bus) [none]
8.dbus-session (D-Bus session bus) [none]

┌──(kali㉿kali)-[~]
└─$
```

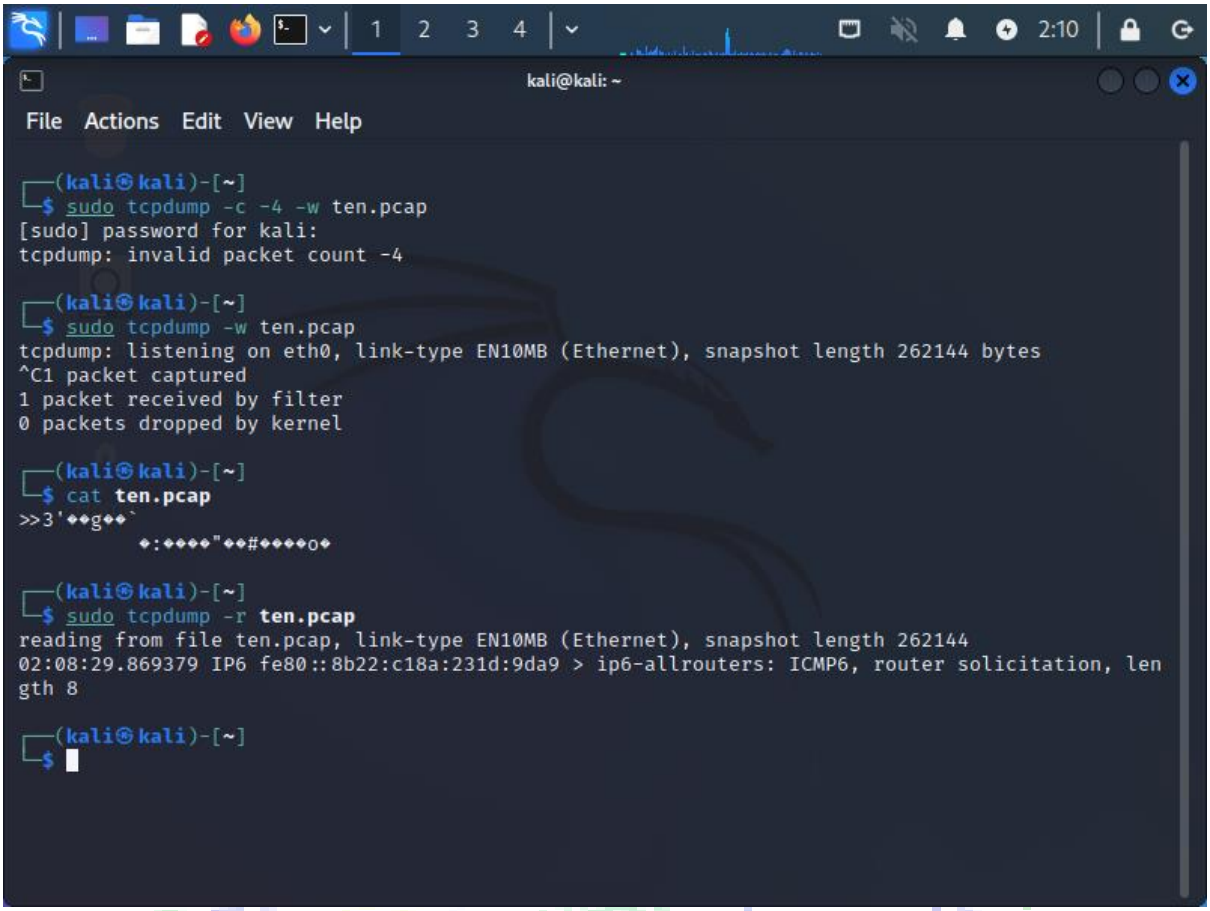
```
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel

┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -w one.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes


^C0 packets captured
0 packets received by filter
0 packets dropped by kernel

┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -c 4
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
01:07:15.701690 IP6 fe80::8b22:c18a:231d:9da9 > ip6-allrouters: ICMP6, router solicitation, len
gth 8
01:07:15.817702 ARP, Request who-has 10.0.2.2 tell 10.0.2.15, length 28
01:07:15.817982 ARP, Reply 10.0.2.2 is-at 52:54:00:12:35:02 (oui Unknown), length 46
01:07:15.817994 IP 10.0.2.15.54290 > mahendraadds.mahendra.local.domain: 63191+ PTR? 9.a.d.9.d.
1.3.2.a.8.1.c.2.2.b.8.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa. (90)
4 packets captured
11 packets received by filter
0 packets dropped by kernel

┌──(kali㉿kali)-[~]
└─$
```



RESULT:

Thus network analysis is performed using Wireshark and Kali Linux.