EXPERIMENT NUMBER 7

Name of Student: Sangeet Agrawal	PRN: 21070122140	Section: CS-B3

Title: Understand Network monitoring tool

Aim: Study and demonstrate Wireshark for packet capturing and monitoring

Objective: To make students learn and demonstrate Wireshark as network monitoring tool

Theory: [Source - https://www.wireshark.org/docs/wsug_html_chunked/index.html**]**

Wireshark is a network packet analyzer. A network packet analyzer presents captured packet data in as much detail as possible. Wireshark is available for free, is open source, and is one of the best packet analyzers available today.

Here are some reasons to use Wireshark:

- Network administrators use it to troubleshoot network problems
- Network security engineers use it to examine security problems
- QA engineers use it to verify network applications
- Developers use it to debug protocol implementations
- People use it to learn network protocol internals

The following are some of the many features Wireshark provides:

- Available for UNIX and Windows.
- Capture live packet data from a network interface.
- Open files containing packet data captured with tcpdump/WinDump, Wireshark, and many other packet capture programs.
- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.
- Save packet data captured.
- Export some or all packets in a number of capture file formats.
- Filter packets on many criteria.
- Search for packets on many criteria.
- Colorize packet display based on filters.
- Create various statistics.

Here are some things Wireshark does not provide:

- Wireshark isn’t an intrusion detection system. It will not warn you when someone does strange things on your network that he/she isn’t allowed to do. However, if strange things happen, Wireshark might help you figure out what is really going on.

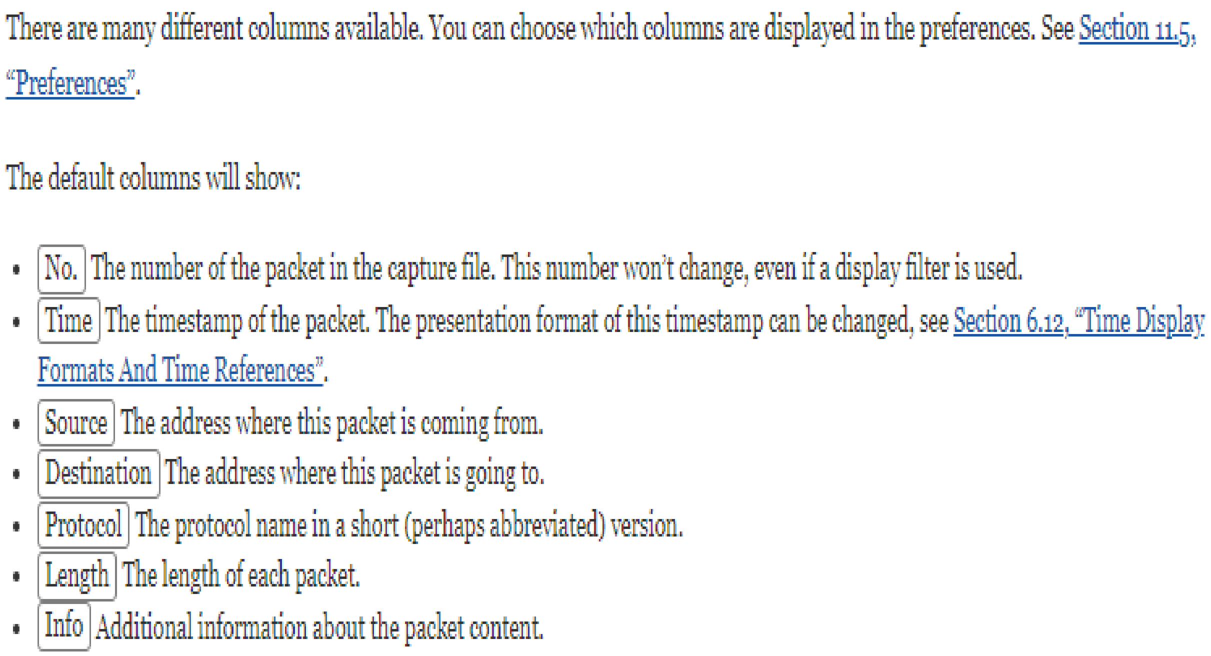
- Wireshark will not manipulate things on the network, it will only “measure” things from it. Wireshark doesn’t send packets on the network or do other active things (except domain name resolution, but that can be disabled).

A Brief History Of Wireshark

In late 1997 Gerald Combs needed a tool for tracking down network problems and wanted to learn more about networking so he started writing Ethereal (the original name of the Wireshark project) as a way to solve both problems. Ethereal was initially released after several pauses in development in July 1998 as version 0.2.0. Within days patches, bug reports, and words of encouragement started arriving and Ethereal was on its way to success.

In October, 1998 Guy Harris was looking for something better than tcpview so he started applying patches and contributing dissectors to Ethereal. In late 1998 Richard Sharpe, who was giving TCP/IP courses, saw its potential on such courses and started looking at it to see if it supported the protocols he needed. While it didn’t at that point new protocols could be easily added. So he started contributing dissectors and contributing patches. In 2006 the project moved house and re-emerged under a new name: Wireshark.

In 2008, after ten years of development, Wireshark finally arrived at version 1.0. This release was the first deemed complete, with the minimum features implemented. Its release coincided with the first Wireshark Developer and User Conference, called Sharkfest. In 2015 Wireshark 2.0 was released, which featured a new user interface. In 2023 Wireshark moved to the Wireshark Foundation, a nonprofit corporation that operates under section 501(c)(3) of the U.S. tax code. The foundation provides the project’s infrastructure, hosts SharkFest, our developer and user conference, and promotes low-level network education.

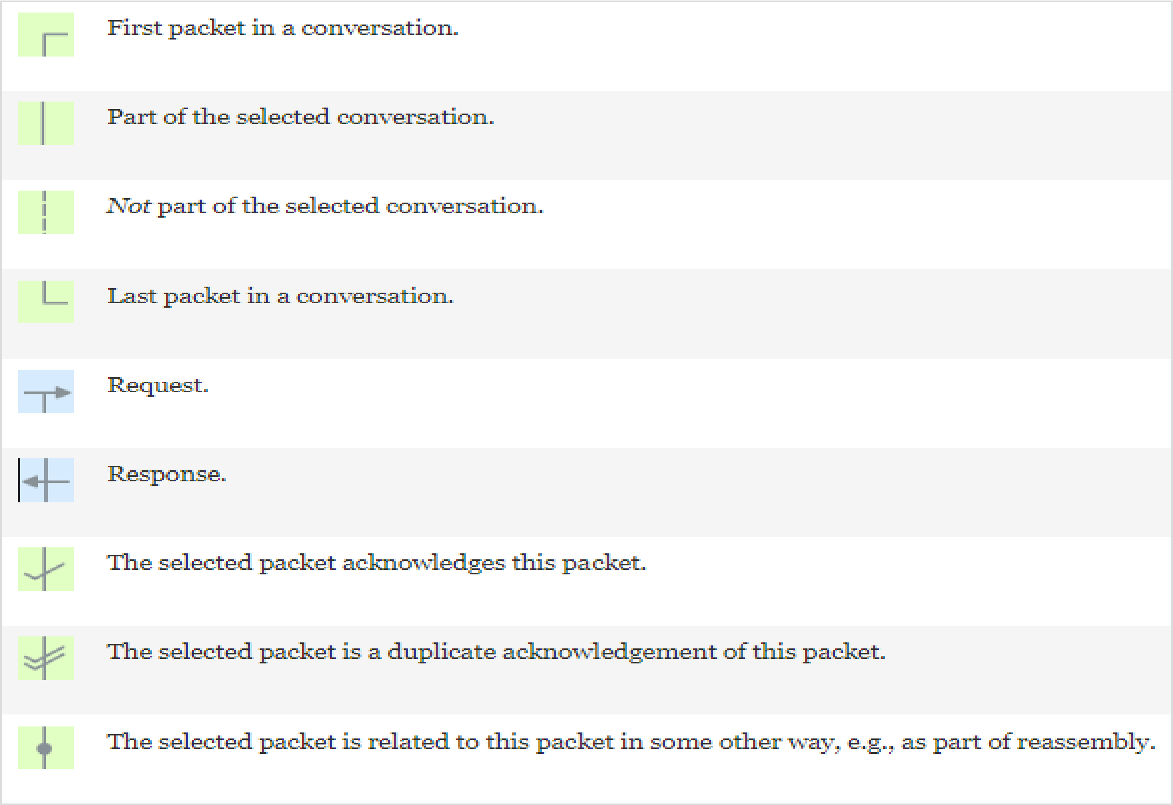


There are many different columns available. You can choose which columns are displayed in the preferences. See Section 11.5, “Preferences”.

The default columns will show:

- 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 the packet. The presentation format of this timestamp can be changed, see Section 6.12, “Time Display Formats And Time References”.
- Source The address where this packet is coming from.
- Destination The address where this packet is going to.
- Protocol The protocol name in a short (perhaps abbreviated) version.
- Length The length of each packet.
- Info Additional information about the packet content.

Related packet symbols



┌	First packet in a conversation.
│	Part of the selected conversation.
┆	*Not* part of the selected conversation.
└	Last packet in a conversation.
→	Request.
←	Response.
✓	The selected packet acknowledges this packet.
✓✓	The selected packet is a duplicate acknowledgement of this packet.
●	The selected packet is related to this packet in some other way, e.g., as part of reassembly.

Tools for Study:

1) WinPcap
2) Wireshark

Reference web links:

1) https://www.winpcap.org/
2) https://www.wireshark.org/
3) https://www.imperva.com/learn/ddos/high-orbit-ion-cannon/
4) https://www.kali.org/tools/hping3/
5) https://www.wireshark.org/docs/wsug_html_chunked/index.html

Theory question:

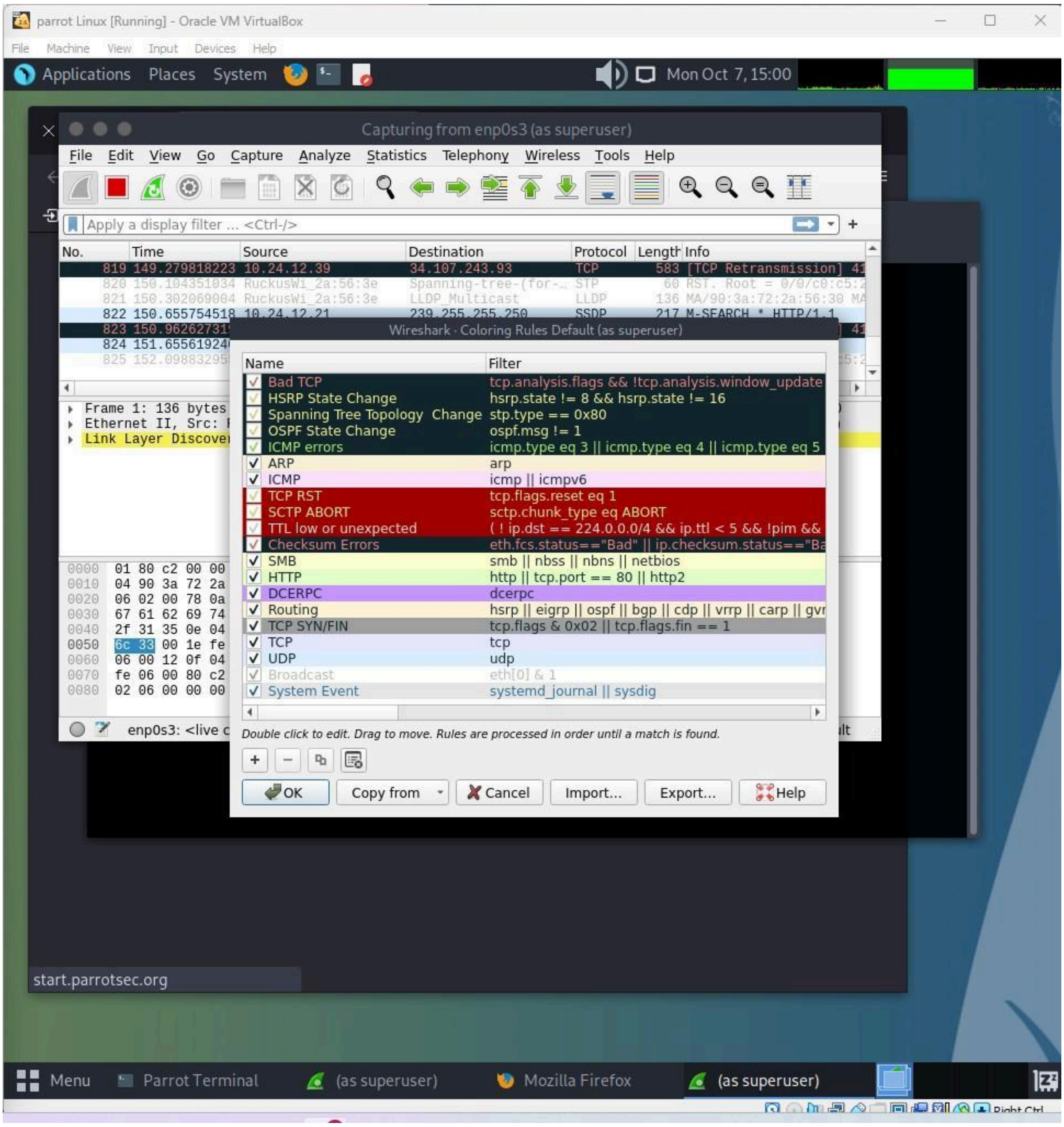
1) Differentiate between HOIC and LOIC

Aspect	HOIC (High Orbit Ion Cannon)	LOIC (Low Orbit Ion Cannon)
Attack Type	HTTP-based DoS/DDoS	TCP/UDP/ICMP DoS/DDoS
Targeting	Multi-threaded, can target multiple URLs	Single-target at a time
Usability	More advanced, requires configuration	Simple and user-friendly
Power	More powerful, harder to mitigate	Less powerful but still dangerous
Usage	Used in larger-scale DDoS attacks	Commonly used in basic DDoS attacks
Anonymity	Requires proxy for anonymity	No built-in anonymity features

2) Differentiate between Winpcap and Wireshark

Aspect	WinPcap	Wireshark
Functionality	Packet capture library (API)	Full packet capture and analysis tool
Role	Backend for packet capturing	Frontend for packet analysis
Operating Environment	CLI/Programmatic use	GUI-based interface
Platform Dependency	Windows-only	Cross-platform (Windows, macOS, Linux)
Usage	Used by apps like Wireshark for capturing	Complete tool for capturing and analyzing packets
Replacement	Deprecated (replaced by Npcap)	Actively maintained and updated

3) Perform DoS/DDoS attack using any open source tool of Kali OS or Windows OS and capture the packets during attack using wireshark.


parrot Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Mon Oct 7, 15:00
Capturing from enp0s3 (as superuser)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
enp0s3: <live c
start.parrotsec.org
Menu
Parrot Terminal
(as superuser)
Mozilla Firefox
(as superuser)
Right Ctrl

No.	Time	Source	Destination	Protocol	Length	Info
819	149.279818223	10.24.12.39	34.107.243.93	TCP	583	[TCP Retransmission] 41
820	150.104351034	RuckusWi_2a:56:3e	Spanning-tree-(for-…	STP	60	RST. Root = 0/0/c0:c5:2
821	150.302069004	RuckusWi_2a:56:3e	LLDP_Multicast	LLDP	136	MA/90:3a:72:2a:56:30 MA
822	150.655754518	10.24.12.21	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
823	150.96262731					41
824	151.65561924					
825	152.09883295					5:2

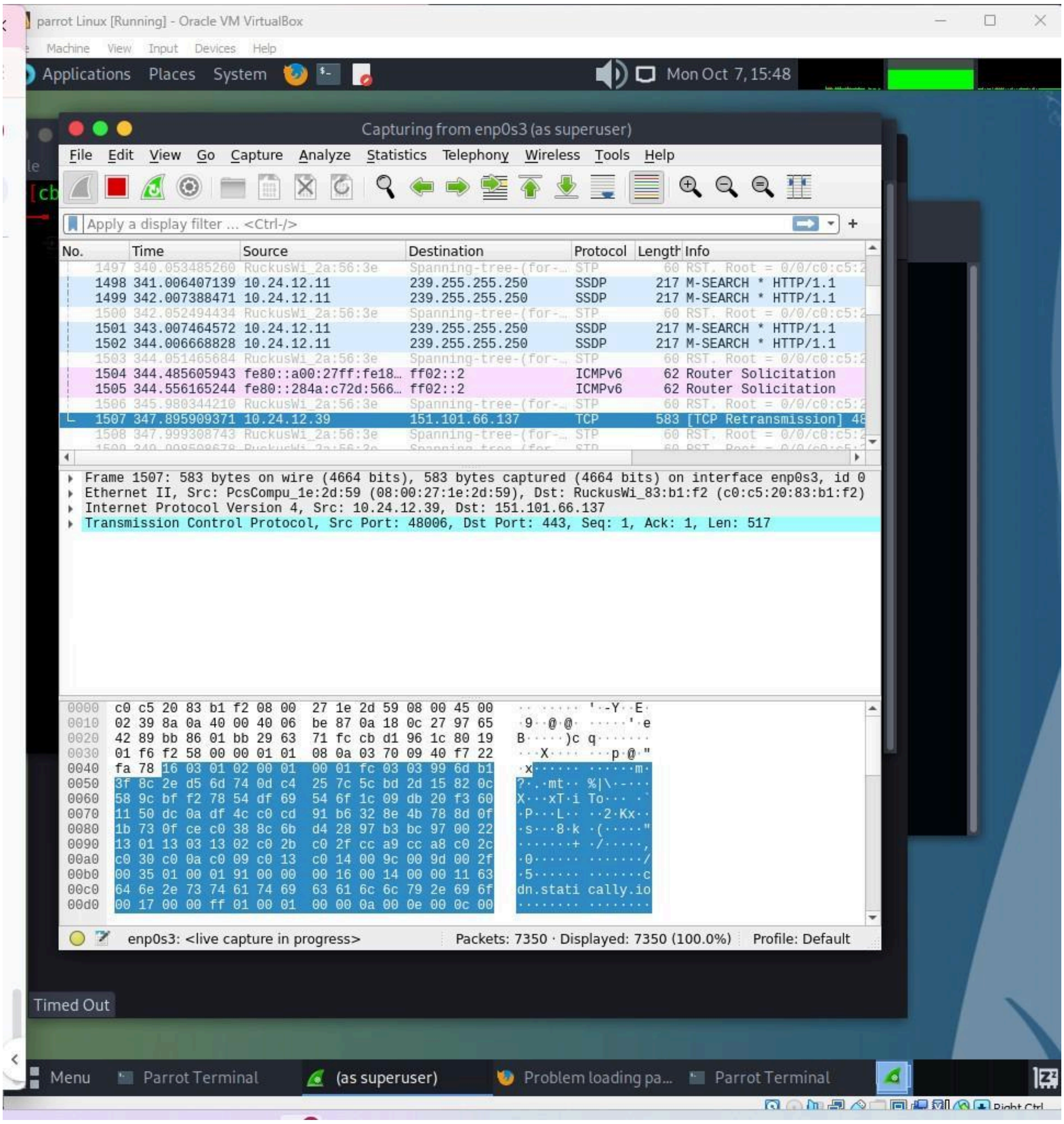
```
Frame 1: 136 bytes
Ethernet II, Src: F
Link Layer Discover
```

```
0000  01 80 c2 00 00
0010  04 90 3a 72 2a
0020  06 02 00 78 0a
0030  67 61 62 69 74
0040  2f 31 35 0e 04
0050  6c 33 00 1e fe
0060  06 00 12 0f 04
0070  fe 06 00 80 c2
0080  02 06 00 00 00
```

Wireshark · Coloring Rules Default (as superuser)

Name	Filter
✓ Bad TCP	tcp.analysis.flags && !tcp.analysis.window_update
✓ 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 5
✓ 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=="Ba
✓ SMB	smb || nbss || nbns || netbios
✓ HTTP	http || tcp.port == 80 || http2
✓ DCERPC	dcerpc
✓ Routing	hsrp || eigrp || ospf || bgp || cdp || vrrp || carp || gvr
✓ TCP SYN/FIN	tcp.flags & 0x02 || tcp.flags.fin == 1
✓ TCP	tcp
✓ UDP	udp
✓ Broadcast	eth[0] & 1
✓ System Event	systemd_journal || sysdig

Double click to edit. Drag to move. Rules are processed in order until a match is found.

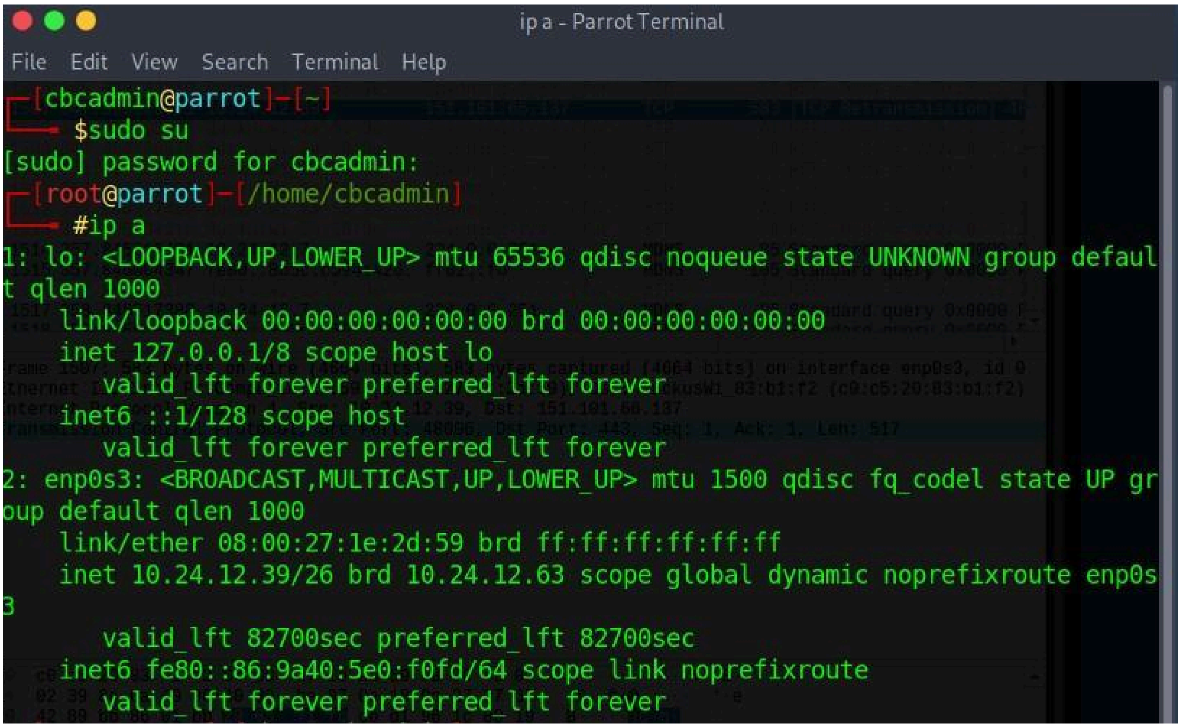

parrot Linux [Running] - Oracle VM VirtualBox
Machine View Input Devices Help
Applications Places System
Mon Oct 7, 15:48
Capturing from enp0s3 (as superuser)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Timed Out
Menu
Parrot Terminal
(as superuser)
Problem loading pa…
Parrot Terminal

```
No.   Time           Source                 Destination            Protocol Length Info
 1497 340.053485260 RuckusWi_2a:56:3e      Spanning-tree-(for-…   STP        60 RST. Root = 0/0/c0:c5:2
 1498 341.006407139 10.24.12.11            239.255.255.250        SSDP      217 M-SEARCH * HTTP/1.1
 1499 342.007388471 10.24.12.11            239.255.255.250        SSDP      217 M-SEARCH * HTTP/1.1
 1500 342.052494434 RuckusWi_2a:56:3e      Spanning-tree-(for-…   STP        60 RST. Root = 0/0/c0:c5:2
 1501 343.007464572 10.24.12.11            239.255.255.250        SSDP      217 M-SEARCH * HTTP/1.1
 1502 344.006668828 10.24.12.11            239.255.255.250        SSDP      217 M-SEARCH * HTTP/1.1
 1503 344.051465684 RuckusWi_2a:56:3e      Spanning-tree-(for-…   STP        60 RST. Root = 0/0/c0:c5:2
 1504 344.485605943 fe80::a00:27ff:fe18…   ff02::2                ICMPv6     62 Router Solicitation
 1505 344.556165244 fe80::284a:c72d:566…   ff02::2                ICMPv6     62 Router Solicitation
 1506 345.980344210 RuckusWi_2a:56:3e      Spanning-tree-(for-…   STP        60 RST. Root = 0/0/c0:c5:2
 1507 347.895909371 10.24.12.39            151.101.66.137         TCP       583 [TCP Retransmission] 48
 1508 347.999308743 RuckusWi_2a:56:3e      Spanning-tree-(for-…   STP        60 RST. Root = 0/0/c0:c5:2
```

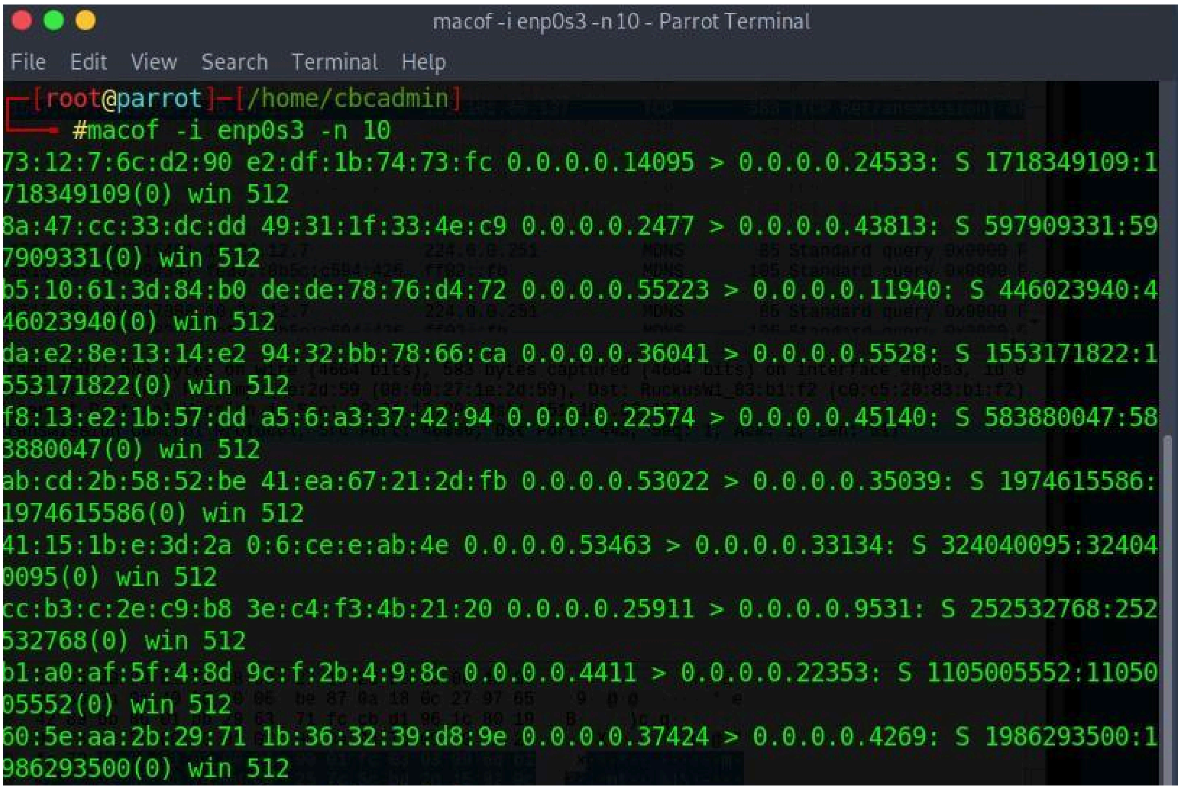
```
▸ Frame 1507: 583 bytes on wire (4664 bits), 583 bytes captured (4664 bits) on interface enp0s3, id 0
▸ Ethernet II, Src: PcsCompu_1e:2d:59 (08:00:27:1e:2d:59), Dst: RuckusWi_83:b1:f2 (c0:c5:20:83:b1:f2)
▸ Internet Protocol Version 4, Src: 10.24.12.39, Dst: 151.101.66.137
▸ Transmission Control Protocol, Src Port: 48006, Dst Port: 443, Seq: 1, Ack: 1, Len: 517
```

```
0000  c0 c5 20 83 b1 f2 08 00  27 1e 2d 59 08 00 45 00   ········ '·-Y··E·
0010  02 39 8a 0a 40 00 40 06  be 87 0a 18 0c 27 97 65   ·9··@·@· ·····'·e
0020  42 89 bb 86 01 bb 29 63  71 fc cb d1 96 1c 80 19   B·····)c q·······
0030  01 f6 f2 58 00 00 01 01  08 0a 03 70 09 40 f7 22   ···X···· ···p·@·"
0040  fa 78 16 03 01 02 00 01  00 01 fc 03 03 99 6d b1   ·x······ ······m·
0050  3f 8c 2e d5 6d 74 0d c4  25 7c 5c bd 2d 15 82 0c   ?·.·mt·· %|\·-···
0060  58 9c bf f2 78 54 df 69  54 6f 1c 09 db 20 f3 60   X···xT·i To··· ·`
0070  11 50 dc 0a df 4c c0 cd  91 b6 32 8e 4b 78 8d 0f   ·P···L·· ··2·Kx··
0080  1b 73 0f ce c0 38 8c 6b  d4 28 97 b3 bc 97 00 22   ·s···8·k ·(·····"
0090  13 01 13 03 13 02 c0 2b  c0 2f cc a9 cc a8 c0 2c   ·······+ ·/·····,
00a0  c0 30 c0 0a c0 09 c0 13  c0 14 00 9c 00 9d 00 2f   ·0······ ·······/
00b0  00 35 01 00 01 91 00 00  00 16 00 14 00 00 11 63   ·5······ ·······c
00c0  64 6e 2e 73 74 61 74 69  63 61 6c 6c 79 2e 69 6f   dn.stati cally.io
00d0  00 17 00 00 ff 01 00 01  00 00 0a 00 0e 00 0c 00   ········ ········
```

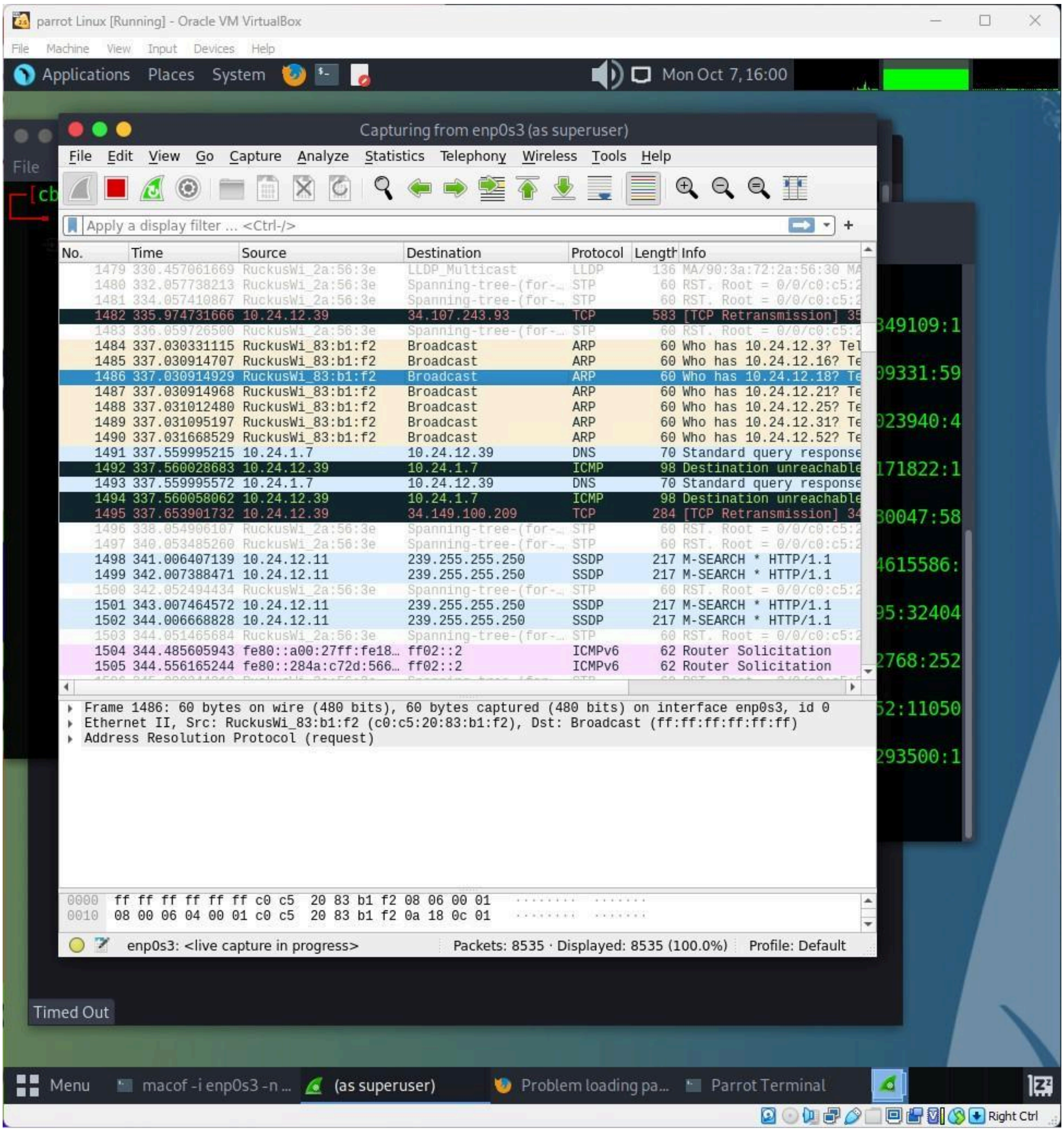
enp0s3: <live capture in progress> Packets: 7350 · Displayed: 7350 (100.0%) Profile: Default



```
┌─[cbcadmin@parrot]─[~]
└──╼ $sudo su
[sudo] password for cbcadmin:
┌─[root@parrot]─[/home/cbcadmin]
└──╼ #ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:1e:2d:59 brd ff:ff:ff:ff:ff:ff
    inet 10.24.12.39/26 brd 10.24.12.63 scope global dynamic noprefixroute enp0s
3
       valid_lft 82700sec preferred_lft 82700sec
    inet6 fe80::86:9a40:5e0:f0fd/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```


macof -i enp0s3 -n 10 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/cbcadmin]
└──╼ #macof -i enp0s3 -n 10
73:12:7:6c:d2:90 e2:df:1b:74:73:fc 0.0.0.0.14095 > 0.0.0.0.24533: S 1718349109:1
718349109(0) win 512
8a:47:cc:33:dc:dd 49:31:1f:33:4e:c9 0.0.0.0.2477 > 0.0.0.0.43813: S 597909331:59
7909331(0) win 512
b5:10:61:3d:84:b0 de:de:78:76:d4:72 0.0.0.0.55223 > 0.0.0.0.11940: S 446023940:4
46023940(0) win 512
da:e2:8e:13:14:e2 94:32:bb:78:66:ca 0.0.0.0.36041 > 0.0.0.0.5528: S 1553171822:1
553171822(0) win 512
f8:13:e2:1b:57:dd a5:6:a3:37:42:94 0.0.0.0.22574 > 0.0.0.0.45140: S 583880047:58
3880047(0) win 512
ab:cd:2b:58:52:be 41:ea:67:21:2d:fb 0.0.0.0.53022 > 0.0.0.0.35039: S 1974615586:
1974615586(0) win 512
41:15:1b:e:3d:2a 0:6:ce:e:ab:4e 0.0.0.0.53463 > 0.0.0.0.33134: S 324040095:32404
0095(0) win 512
cc:b3:c:2e:c9:b8 3e:c4:f3:4b:21:20 0.0.0.0.25911 > 0.0.0.0.9531: S 252532768:252
532768(0) win 512
b1:a0:af:5f:4:8d 9c:f:2b:4:9:8c 0.0.0.0.4411 > 0.0.0.0.22353: S 1105005552:11050
05552(0) win 512
60:5e:aa:2b:29:71 1b:36:32:39:d8:9e 0.0.0.0.37424 > 0.0.0.0.4269: S 1986293500:1
986293500(0) win 512
```


parrot Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Mon Oct 7, 16:00
Capturing from enp0s3 (as superuser)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Timed Out
Menu
macof -i enp0s3 -n ...
(as superuser)
Problem loading pa...
Parrot Terminal
Right Ctrl

```
No.   Time         Source                 Destination            Protocol Length Info
 1479 330.457061669 RuckusWi_2a:56:3e     LLDP_Multicast         LLDP     136 MA/90:3a:72:2a:56:30 MA
 1480 332.057738213 RuckusWi_2a:56:3e     Spanning-tree-(for-…   STP       60 RST. Root = 0/0/c0:c5:2
 1481 334.057410867 RuckusWi_2a:56:3e     Spanning-tree-(for-…   STP       60 RST. Root = 0/0/c0:c5:2
 1482 335.974731666 10.24.12.39           34.107.243.93          TCP      583 [TCP Retransmission] 35
 1483 336.059726500 RuckusWi_2a:56:3e     Spanning-tree-(for-…   STP       60 RST. Root = 0/0/c0:c5:2
 1484 337.030331115 RuckusWi_83:b1:f2     Broadcast              ARP       60 Who has 10.24.12.3? Tel
 1485 337.030914707 RuckusWi_83:b1:f2     Broadcast              ARP       60 Who has 10.24.12.16? Te
 1486 337.030914929 RuckusWi_83:b1:f2     Broadcast              ARP       60 Who has 10.24.12.18? Te
 1487 337.030914968 RuckusWi_83:b1:f2     Broadcast              ARP       60 Who has 10.24.12.21? Te
 1488 337.031012480 RuckusWi_83:b1:f2     Broadcast              ARP       60 Who has 10.24.12.25? Te
 1489 337.031095197 RuckusWi_83:b1:f2     Broadcast              ARP       60 Who has 10.24.12.31? Te
 1490 337.031668529 RuckusWi_83:b1:f2     Broadcast              ARP       60 Who has 10.24.12.52? Te
 1491 337.559995215 10.24.1.7             10.24.12.39            DNS       70 Standard query response
 1492 337.560028683 10.24.12.39           10.24.1.7              ICMP      98 Destination unreachable
 1493 337.559995572 10.24.1.7             10.24.12.39            DNS       70 Standard query response
 1494 337.560058062 10.24.12.39           10.24.1.7              ICMP      98 Destination unreachable
 1495 337.653901732 10.24.12.39           34.149.100.209         TCP      284 [TCP Retransmission] 34
 1496 338.054906107 RuckusWi_2a:56:3e     Spanning-tree-(for-…   STP       60 RST. Root = 0/0/c0:c5:2
 1497 340.053485260 RuckusWi_2a:56:3e     Spanning-tree-(for-…   STP       60 RST. Root = 0/0/c0:c5:2
 1498 341.006407139 10.24.12.11           239.255.255.250        SSDP     217 M-SEARCH * HTTP/1.1
 1499 342.007388471 10.24.12.11           239.255.255.250        SSDP     217 M-SEARCH * HTTP/1.1
 1500 342.052494434 RuckusWi_2a:56:3e     Spanning-tree-(for-…   STP       60 RST. Root = 0/0/c0:c5:2
 1501 343.007464572 10.24.12.11           239.255.255.250        SSDP     217 M-SEARCH * HTTP/1.1
 1502 344.006668828 10.24.12.11           239.255.255.250        SSDP     217 M-SEARCH * HTTP/1.1
 1503 344.051465684 RuckusWi_2a:56:3e     Spanning-tree-(for-…   STP       60 RST. Root = 0/0/c0:c5:2
 1504 344.485605943 fe80::a00:27ff:fe18…  ff02::2                ICMPv6    62 Router Solicitation
 1505 344.556165244 fe80::284a:c72d:566…  ff02::2                ICMPv6    62 Router Solicitation

▸ Frame 1486: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface enp0s3, id 0
▸ Ethernet II, Src: RuckusWi_83:b1:f2 (c0:c5:20:83:b1:f2), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▸ Address Resolution Protocol (request)

0000  ff ff ff ff ff ff c0 c5  20 83 b1 f2 08 06 00 01   ........ ........
0010  08 00 06 04 00 01 c0 c5  20 83 b1 f2 0a 18 0c 01   ........ ........

enp0s3: <live capture in progress>    Packets: 8535 · Displayed: 8535 (100.0%)    Profile: Default
```

4) Explain important details about the packets captured in the previous question.

I. Source and Destination IPs:

- The source IP represents the machine from which the attack originated or appeared to originate. In a DoS attack, this is typically the IP address of the targeted system.
- The destination IP refers to the IP address of the machine being targeted by the attack.

II. Types of Packets:

- Depending on the nature of the attack, packets may include ICMP (used in ping floods), TCP SYN (used in SYN floods), or UDP packets.
- For HTTP floods, a large number of HTTP GET or POST requests are typically observed.

III. Packet Volume and Frequency:

- A DoS attack is characterized by a high frequency of identical or nearly identical packets, usually transmitted in rapid succession, as seen in attacks like ICMP or SYN floods.

IV. Flags (in TCP packets):

- In TCP-based attacks, such as SYN floods, multiple TCP packets with the SYN flag are transmitted, indicating repeated attempts to establish connections without completing them.
- The targeted server may respond with SYN-ACK packets, but the attacker fails to complete the handshake.

V. Packet Size:

- Packet sizes may vary depending on the tools used. For instance, UDP floods may include large packets to maximize bandwidth consumption, while ICMP floods typically involve uniform packet size.

Note: Students are suggested to use Linux OS-based tools or free Windows OS-based tools.

Conclusion: We have successfully implemented Wireshark for packet capturing and network monitoring. This study provided students with hands-on experience in analyzing network traffic, enhancing their skills in detecting patterns and identifying potential threats effectively.