

Roll No: 2103163
Batch: C32
Name: Om Shete

Experiment No 8

Aim: Study and implementation of packet sniffer tool: wireshark

Description:

What is Wireshark?

Wireshark is a widely used, open source [network analyzer](#) that can capture and display real-time details of network traffic. It is particularly useful for troubleshooting network issues, analyzing network protocols and ensuring network security.

Networks must be monitored to ensure smooth operations and security. Popular with academic institutions, government agencies, corporations and nonprofits, Wireshark is one such tool that can offer an in-depth view into network activities, diagnose [network performance issues](#) or identify [potential security threats](#).

Key features of Wireshark

Wireshark seeks to simplify and enhance the process of network traffic analysis. Each function is designed to offer unique insights and control over network activities. Here are some of its core features:

- **Packet capture (PCAP).** Converts network traffic into a human-readable format, making it easier to understand and diagnose concerns.
- **Real-time analysis.** Provides a live view of network traffic, offering immediate insights into ongoing network activities.
- **Filtering capabilities.** Enables users to focus on specific types of network traffic, making analysis more efficient and targeted.
- **Graphical user interface (GUI).** Designed for ease of use, ensures that both beginners and experts can navigate and analyze data effectively

Common uses for Wireshark

Wireshark can be used to examine the details of traffic at a variety of levels, ranging from connection-level information to the bits constituting a single [packet](#).

Roll No: 2103163

Batch: C32

Name: Om Shete

PCAP can provide a network administrator with information about individual packets, including transmit time, source, destination, [protocol](#) type and [header](#) data. This information can be useful for evaluating security events and troubleshooting network security device issues.

Wireshark's capabilities extend beyond just monitoring to address other network administration tasks:

- **Network troubleshooting.** Pinpoints and resolves network issues with the comprehensive data Wireshark provides.
- **Security analysis.** Detects and analyzes potential security threats in the network.
- **Performance analysis.** Monitors and [optimizes network performance](#) to ensure smooth operations.
- **Protocol analysis.** Gains insights into the behavior of individual protocols within the network.

Supported file formats in Wireshark

Wireshark is known for its versatility and the wide array of file formats it supports. The primary file format used by Wireshark to save PCAPs is PcapNG, which stands for Packet Capture Next Generation. This format is recognized for its flexibility in capturing and storing packet data.

To support [interoperability](#) with third-party protocol analyzers, Wireshark also has the ability to read and save packet data in other file formats, including CAP and PCAP.

Roll No: 2103163

Batch: C32

Name: Om Shete

Results:

The image shows a Wireshark packet capture window titled '*eno1'. The packet list on the left shows a series of SSDP M-SEARCH messages followed by an HTTP GET request (No. 74). The selected packet (No. 74) is expanded, showing the following details:

- Frame 74: 153 bytes on wire (1224 bits), 153 bytes captured (1224 bits) on interface eno1, id 0
- Ethernet II, Src: Micro-St_c2:99:83 (d8:bb:c1:c2:99:83), Dst: 9c:53:22:05:6a:19 (9c:53:22:05:6a:19)
- Internet Protocol Version 4, Src: 192.168.31.52, Dst: 91.189.91.97
- Transmission Control Protocol, Src Port: 46922, Dst Port: 80, Seq: 1, Ack: 1, Len: 87
- Hypertext Transfer Protocol
 - GET / HTTP/1.1\r\n
 - Host: connectivity-check.ubuntu.com\r\n
 - Accept: */*\r\n
 - Connection: close\r\n
 - \r\n
 - [Full request URI: <http://connectivity-check.ubuntu.com/>]
 - [HTTP request 1/1]
 - [Response in frame: 75]

The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

Wireshark_eno12L45K2.pcapng Packets: 2878 · Displayed: 285 (9.9%) Profile: Default

The image shows a Wireshark packet capture window titled '*eno1'. The packet list on the left shows a series of DNS queries followed by a DNS response (No. 69). The selected packet (No. 69) is expanded, showing the following details:

- Frame 69: 356 bytes on wire (2848 bits), 356 bytes captured (2848 bits) on interface eno1, id 0
- Ethernet II, Src: 9c:53:22:05:6a:19 (9c:53:22:05:6a:19), Dst: Micro-St_c2:99:83 (d8:bb:c1:c2:99:83)
- Internet Protocol Version 4, Src: 203.212.24.18, Dst: 192.168.31.52
- User Datagram Protocol, Src Port: 53, Dst Port: 54228
- Domain Name System (response)
 - Transaction ID: 0xcb75
 - Flags: 0x8180 Standard query response, No error
 - Questions: 1
 - Answer RRs: 12
 - Authority RRs: 3
 - Additional RRs: 1
 - Queries
 - Answers
 - Authoritative nameservers
 - Additional records
 - [Request In: 68]
 - [Time: 0.00198421 seconds]

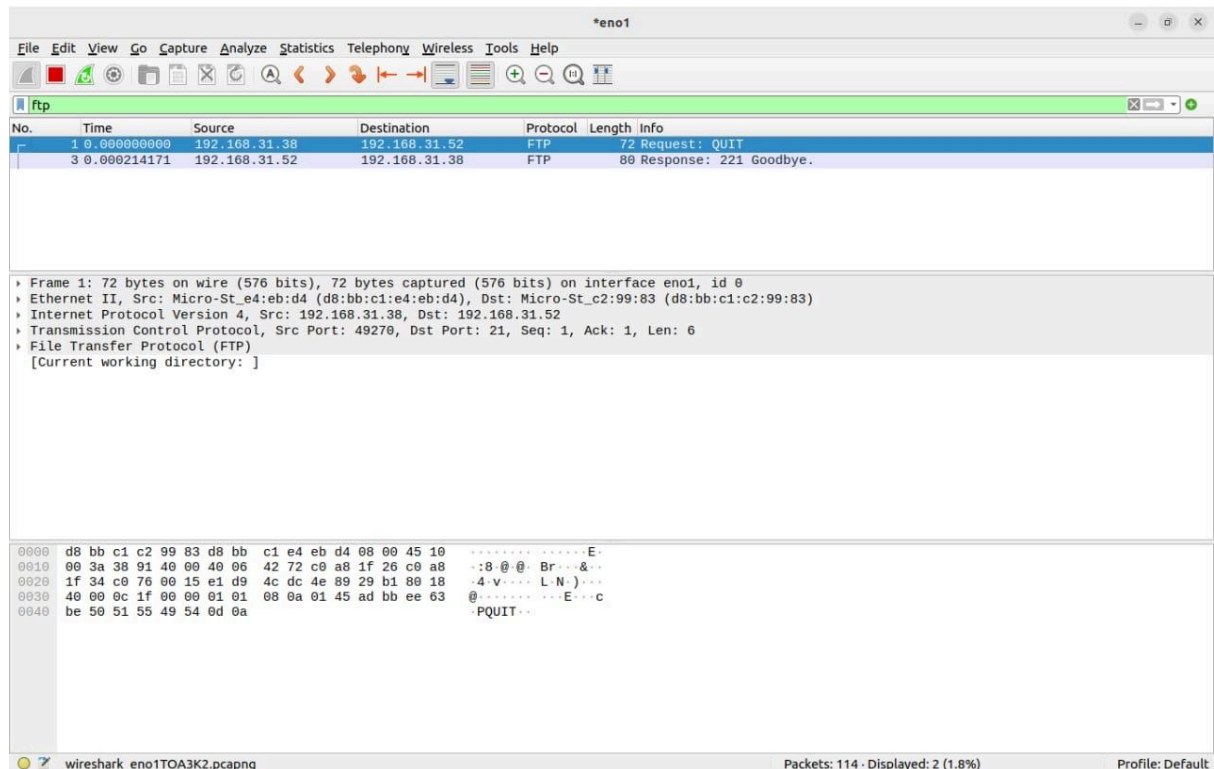
The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

Wireshark_eno12L45K2.pcapng Packets: 2130 · Displayed: 38 (1.8%) Profile: Default

Roll No: 2103163

Batch: C32

Name: Om Shete



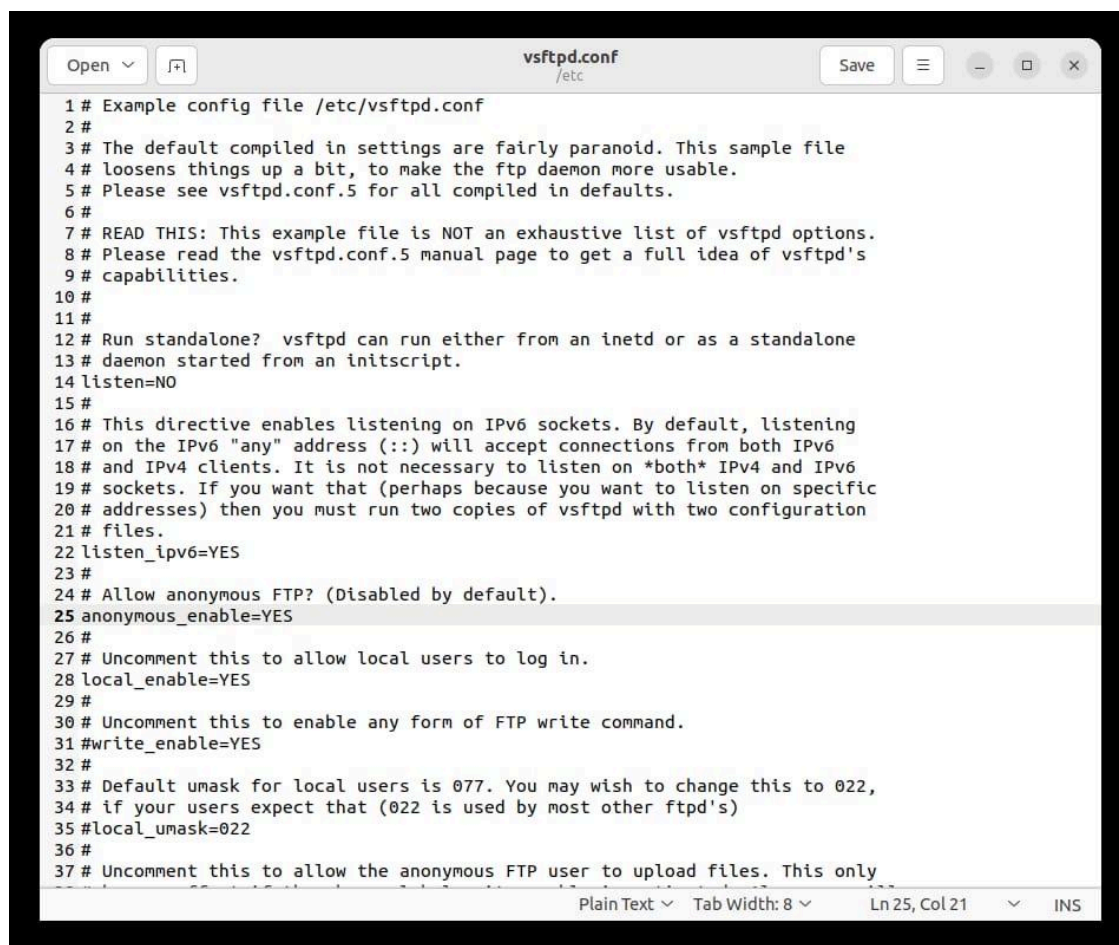
Wireshark packet capture window showing an FTP session on interface eno1. The packet list shows two packets: a QUIT request and a 221 Goodbye response. The packet details show the Ethernet II, IP, and FTP layers. The packet bytes show the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.31.38	192.168.31.52	FTP	72	Request: QUIT
3	0.000214171	192.168.31.52	192.168.31.38	FTP	80	Response: 221 Goodbye.

Frame 1: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface eno1, id 0
Ethernet II, Src: Micro-St_e4:eb:d4 (d8:bb:c1:e4:eb:d4), Dst: Micro-St_c2:99:83 (d8:bb:c1:c2:99:83)
Internet Protocol Version 4, Src: 192.168.31.38, Dst: 192.168.31.52
Transmission Control Protocol, Src Port: 49270, Dst Port: 21, Seq: 1, Ack: 1, Len: 6
File Transfer Protocol (FTP)
[Current working directory:]

0000 d8 bb c1 c2 99 83 d8 bb c1 e4 eb d4 08 00 45 10E.
0010 00 3a 38 91 40 00 40 06 42 72 c0 a8 1f 26 c0 a8 ..:8 @ @ Br...&..
0020 1f 34 c0 76 00 15 e1 d9 4c dc 4e 89 29 b1 80 18 -4-v...L.N)...
0030 40 00 0c 1f 00 00 01 01 08 0a 01 45 ad bb ee 63 @.....E...c
0040 be 50 51 55 49 54 0d 0aPQUIT...

wireshark_eno1TOA3K2.pcapng Packets: 114 · Displayed: 2 (1.8%) Profile: Default



```
1 # Example config file /etc/vsftpd.conf
2 #
3 # The default compiled in settings are fairly paranoid. This sample file
4 # loosens things up a bit, to make the ftp daemon more usable.
5 # Please see vsftpd.conf.5 for all compiled in defaults.
6 #
7 # READ THIS: This example file is NOT an exhaustive list of vsftpd options.
8 # Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
9 # capabilities.
10 #
11 #
12 # Run standalone? vsftpd can run either from an inetd or as a standalone
13 # daemon started from an initscript.
14 listen=NO
15 #
16 # This directive enables listening on IPv6 sockets. By default, listening
17 # on the IPv6 "any" address (:::) will accept connections from both IPv6
18 # and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
19 # sockets. If you want that (perhaps because you want to listen on specific
20 # addresses) then you must run two copies of vsftpd with two configuration
21 # files.
22 listen_ipv6=YES
23 #
24 # Allow anonymous FTP? (Disabled by default).
25 anonymous_enable=YES
26 #
27 # Uncomment this to allow local users to log in.
28 local_enable=YES
29 #
30 # Uncomment this to enable any form of FTP write command.
31 write_enable=YES
32 #
33 # Default umask for local users is 077. You may wish to change this to 022,
34 # if your users expect that (022 is used by most other ftpd's)
35 local_umask=022
36 #
37 # Uncomment this to allow the anonymous FTP user to upload files. This only
```

Plain Text Tab Width: 8 Ln 25, Col 21 INS

Roll No: 2103163

Batch: C32

Name: Om Shete

The image shows a Wireshark packet capture analysis of a SYN packet. The filter is set to 'tcp.port == 80'. The packet list shows a SYN packet (No. 70) from 192.168.31.52 to 91.189.91.97. The packet details pane shows the following information:

- Frame 70: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eno1, id 0
- Ethernet II, Src: Micro-St_c2:99:83 (d8:bb:c1:c2:99:83), Dst: 9c:53:22:05:6a:19 (9c:53:22:05:6a:19)
- Internet Protocol Version 4, Src: 192.168.31.52, Dst: 91.189.91.97
- Transmission Control Protocol, Src Port: 46922, Dst Port: 80, Seq: 0, Len: 0
- Source Port: 46922
- Destination Port: 80
- [Stream index: 4]
- [Conversation completeness: Complete, WITH_DATA (31)]
- [TCP Segment Len: 0]
- Sequence Number: 0 (relative sequence number)
- Sequence Number (raw): 3356294333
- [Next Sequence Number: 1 (relative sequence number)]
- Acknowledgment Number: 0
- Acknowledgment number (raw): 0
- 1010 ... = Header Length: 40 bytes (10)
- Flags: 0x002 (SYN)
- Window: 64240

The packet bytes pane shows the raw data of the SYN packet.

The image shows a Wireshark packet capture analysis of a DNS query. The filter is set to 'ip.addr == 192.168.31.7'. The packet list shows a DNS query (No. 342) from 192.168.31.7 to 192.168.31.255. The packet details pane shows the following information:

- Frame 36: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eno1, id 0
- Ethernet II, Src: Dell_1a:23:05 (d0:07:e5:1a:23:05), Dst: IPv4mcast_16 (01:00:5e:00:00:16)
- Internet Protocol Version 4, Src: 192.168.31.7, Dst: 224.0.0.22
- Internet Group Management Protocol

The packet bytes pane shows the raw data of the DNS query.

Roll No: 2103163

Batch: C32

Name: Om Shete

Wireshark capture window titled *eno1. The filter bar shows 'ip.addr == 192.168.31.7 && ip.addr == 192.168.31.255'. The packet list shows 364 packets, all of which are Name query NB DINESH-PC<20> from 192.168.31.7 to 192.168.31.255. The packet details pane shows the structure of a Name query packet, including Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and NetBIOS Name Service. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
99	30.929973293	192.168.31.7	192.168.31.255	NBNS	92	Name query NB DINESH-PC<20>
100	31.679180973	192.168.31.7	192.168.31.255	NBNS	92	Name query NB DINESH-PC<20>
102	32.429009530	192.168.31.7	192.168.31.255	NBNS	92	Name query NB DINESH-PC<20>
108	33.481514585	192.168.31.7	192.168.31.255	NBNS	92	Name query NB DINESH-PC<20>
109	34.231418789	192.168.31.7	192.168.31.255	NBNS	92	Name query NB DINESH-PC<20>
110	34.981145452	192.168.31.7	192.168.31.255	NBNS	92	Name query NB DINESH-PC<20>
342	90.720651925	192.168.31.7	192.168.31.255	NBNS	92	Name query NB DINESH-PC<20>
348	91.470385757	192.168.31.7	192.168.31.255	NBNS	92	Name query NB DINESH-PC<20>
351	92.220422693	192.168.31.7	192.168.31.255	NBNS	92	Name query NB DINESH-PC<20>
358	93.272742528	192.168.31.7	192.168.31.255	NBNS	92	Name query NB DINESH-PC<20>
361	94.022470457	192.168.31.7	192.168.31.255	NBNS	92	Name query NB DINESH-PC<20>
364	94.772544278	192.168.31.7	192.168.31.255	NBNS	92	Name query NB DINESH-PC<20>

Frame 99: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface eno1, id 0
Ethernet II, Src: Dell_1a:23:05 (d8:67:e5:1a:23:05), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 192.168.31.7, Dst: 192.168.31.255
User Datagram Protocol, Src Port: 137, Dst Port: 137
NetBIOS Name Service

0000 ff ff ff ff ff ff d0 67 e5 1a 23 05 00 00 45 00g..#...E.
0010 00 4e 04 6f 00 00 00 11 75 d9 c0 a8 1f 07 c0 a8 ..N.o...u.....
0020 1f ff 00 89 00 89 00 3a df c3 ed d4 01 10 00 01:.....
0030 00 00 00 00 00 20 45 45 45 4a 45 4f 45 46 46E.EJEOEFF
0040 44 45 49 43 4e 46 41 45 44 43 41 43 41 43 41 43 DEICNFAE DCACACAC
0050 41 43 41 43 41 43 41 00 00 20 00 01 ACACACA... ..

wireshark_eno12L45K2.pcapng Packets: 1894 · Displayed: 25 (1.3%) Profile: Default

Wireshark capture window titled *eno1. The filter bar shows '!(arp or icmp or dns)'. The packet list shows 74 packets, including TCP, TLS, and HTTP traffic. The packet details pane shows the structure of a Hypertext Transfer Protocol (HTTP) GET request. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
65	22.108295039	142.250.182.197	192.168.31.52	TCP	66	443 → 47490 [ACK] Seq=1 Ack=40 Win=11831 Len=0 TSval=1208291884 TSecr...
66	22.108295417	142.250.182.197	192.168.31.52	TLSv1.2	105	Application Data
67	22.152446707	192.168.31.52	142.250.182.197	TCP	66	47490 → 443 [ACK] Seq=40 Ack=40 Win=2945 Len=0 TSval=1742840308 TSecr...
70	23.230796687	192.168.31.52	91.189.91.97	TCP	74	46922 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=326...
71	23.397874506	192.168.31.3	224.0.0.251	MDNS	103	Standard query 0x0005 PTR _233637DE._sub._googlecast._tcp.local, "QM"...
72	23.433395286	91.189.91.97	192.168.31.52	TCP	74	80 → 46922 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1440 SACK_PERM=...
73	23.433478870	192.168.31.52	91.189.91.97	TCP	66	46922 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3269790561 TSecr=1...
74	23.433671980	192.168.31.52	91.189.91.97	HTTP	153	GET / HTTP/1.1

Frame 74: 153 bytes on wire (1224 bits), 153 bytes captured (1224 bits) on interface eno1, id 0
Ethernet II, Src: Micro-St_c2:99:83 (d8:bb:c1:c2:99:83), Dst: 9c:53:22:05:6a:19 (9c:53:22:05:6a:19)
Internet Protocol Version 4, Src: 192.168.31.52, Dst: 91.189.91.97
Transmission Control Protocol, Src Port: 46922, Dst Port: 80, Seq: 1, Ack: 1, Len: 87
Hypertext Transfer Protocol

0000 9c 53 22 05 6a 19 d8 bb c1 c2 99 83 08 00 45 00 ..S".j.....E.
0010 00 8b 0d b9 40 00 40 06 95 b9 c0 a8 1f 34 5b bd ...@.#.....4[.
0020 5b 61 b7 4a 00 50 c8 0c fc be c1 20 d0 15 80 18 [a.J.P.....
0030 01 f6 97 78 00 00 01 01 08 0a c2 e5 0b 61 5c d7 ..x.....aV.
0040 20 52 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 RGET / HTTP/1.1
0050 0d 0a 48 6f 73 74 3a 20 63 6f 6e 6e 65 63 74 69 ..Host: connecti
0060 76 69 74 79 2d 63 68 65 63 6b 2e 75 62 75 6e 74 vity-che ck.ubunt
0070 75 2e 63 6f 6d 0d 0a 41 63 63 65 70 74 3a 20 2a u.com .A ccept: *
0080 2f 2a 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 /*- Conn action:
0090 63 6c 6f 73 65 0d 0a 0d 0a ..close...

wireshark_eno12L45K2.pcapng Packets: 3061 · Displayed: 2519 (82.3%) Profile: Default

Name: Om Shete

The image shows a Wireshark packet capture of an HTTP GET request and response. The packet list at the top shows a single packet (No. 74) of type GET, source 192.168.31.52, and destination 192.168.31.52. The packet details pane shows the request line 'GET / HTTP/1.1' and the response line '200 OK (text/html)'. The packet bytes pane shows the raw data of the request and response, including the status bar '200 OK (text/html)'.