Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Information Technology

Experiment No: 5				
Date of Performance:	07-08-24			
Date of Submission:	07-08-24			
Program Formation/ Execution/ Correction (06)	**Timely Submission (01)**	**Viva (03)**	**Experiment Marks (10)**	**Teacher Signature with date**

EXPERIMENT - 05

Aim : Study of Packet Sniffer Tools WireShark

Theory

Wireshark Overview: Wireshark is a widely used **network protocol analyzer** that allows you to see what's happening on your network at a microscopic level. It captures and analyzes network traffic in real-time, enabling users to examine data packets to troubleshoot issues, perform security analysis, or simply monitor network traffic.

Key Features of Wireshark:

1. **Packet Capture**: Wireshark can capture network packets from any live network interface and record the traffic.
2. **Real-time Analysis**: It provides real-time monitoring of traffic, which is especially useful for identifying network problems as they occur.
3. **Detailed Packet Information**: Each packet captured can be dissected, revealing information such as source, destination, protocol type, and data contents.
4. **Protocol Support**: Wireshark supports a wide range of network protocols, including TCP, UDP, HTTP, DNS, ICMP, etc.
5. **Filters**: Users can apply capture and display filters to narrow down the data for focused analysis.
6. **Export and Report**: The captured data can be saved and exported in various formats for further analysis or reporting.

How Wireshark Works: Wireshark uses the **libpcap** or **WinPcap** libraries to capture packets. When capturing is initiated, Wireshark goes into "promiscuous mode," allowing it to see all the traffic on the network segment to which the capturing device is connected.

Common Use Cases:

1. **Troubleshooting Network Issues**: Identifying dropped packets, latency issues, or misconfigurations.
2. **Security Analysis**: Monitoring suspicious activities, detecting intrusions, or examining malicious traffic.
3. **Protocol Analysis**: Examining the behavior of specific protocols in detail.

Common Filters:

- tcp: Capture only TCP packets.
- udp: Capture only UDP packets.
- ip.addr == 192.168.1.1: Capture packets to or from the specified IP address.

- http: Capture only HTTP traffic.

Implementation

Step-by-Step Guide to Using Wireshark

Step 1: Install Wireshark

```
[12/10/24]seed@VM : ~ / . . . /$ sudo apt-get install wireshark
Reading package lists...
Done Building dependency tree Reading state information...
Done The following
NEW packages will be installed: wireshark 0 upgraded, 1 newly installed.
Need to get 45.7 MB of archives. ... Setting up wireshark (version X.X) ...
```

Step 2: Start Wireshark

Open Wireshark.

- After starting, you'll see a list of network interfaces to choose from.

```
Select interface for live capture:
- Ethernet
- Wi-Fi
- Loopback
- USB network adapter
```

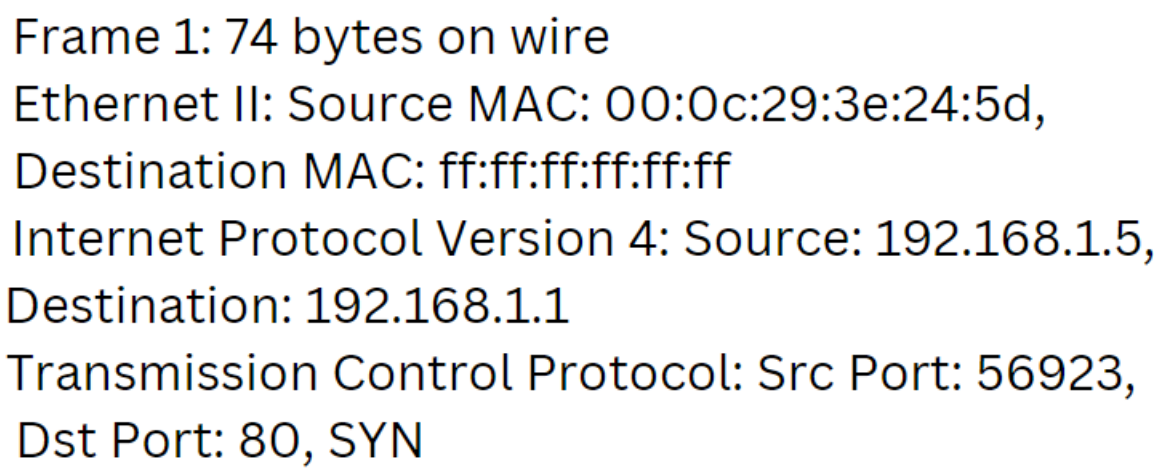
Step 3: Capture Network Traffic

Once Wireshark starts capturing:

- Wireshark will display live traffic in real time, showing each packet's details (protocol, source, destination, etc.).

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00	192.168.1.5	192.168.1.1	TCP	74	File
2	0.01	192.168.1.1	192.168.1.5	TCP	66	Folder
3	0.02	192.168.1.5	192.168.1.1	TCP	56	Folder
4	0.03	192.168.1.5	192.168.1.1	HTTP	104	File
...	...	...	...	...	...	

- **Clicking on any packet will show detailed packet information.**



Step 4: Apply Filters

Filters help narrow down specific traffic types.

- To capture **only HTTP traffic**, enter:

No.	Time	Source	Destination	Protocol	Length	Info
5	1.000	192.168.1.5	192.168.1.1	HTTP	520	GET /login HTTP/1.1
6	1.001	192.168.1.1	192.168.1.5	HTTP	1024	HTTP/1.1 200 OK

...

- To filter by a **specific IP address**:

No.	Time	Source	Destination	Protocol	Length	Info
8	1.500	192.168.1.100	192.168.1.1	TCP	74	[SYN] Seq=0 Win=65535 Len=0
9	1.501	192.168.1.1	192.168.1.100	TCP	66	[SYN, ACK] Seq=0 Ack=1

...

Step 5: Analyze Packets

You can inspect individual packets by clicking on them. Each packet can be expanded to reveal the Ethernet, IP, and TCP/UDP layers.

Example of packet inspection:

Ethernet II:

Source: 00:0c:29:3e:24:5d (Vendor)

Destination: ff:ff:ff:ff:ff:ff (Broadcast)

Internet Protocol Version 4:

Source: 192.168.1.100

Destination: 192.168.1.1

Transmission Control Protocol:

Source Port: 55000

Destination Port: 80

[SYN] Seq=0, Win=65535, Len=0

Hypertext Transfer Protocol:

GET / HTTP/1.1

Step 6: Save the Captured Traffic

- Once you have captured the traffic you're interested in, save the capture for future reference:
    - Go to **File > Save As** and save the capture in the desired format (e.g., .pcap).

Conclusion

Wireshark is an incredibly powerful packet-sniffing tool that allows for in-depth analysis of network traffic. In this experiment, we successfully captured real-time network data, applied filters to narrow down specific types of traffic, and analyzed individual packets.

Wireshark is invaluable for network troubleshooting, protocol analysis, and security auditing. Its ability to provide detailed information about every packet on the network makes it one of the most popular tools for network administrators and security professionals.