Network Traffic Analyzer using Wireshark

Network Traffic Analyzer using Wireshark - Multi-Protocol Security Analysis

1. Title Page

Project Title: Network Traffic Analyzer using Wireshark

Name: B. Kartheek

Date: 07-10-2025

2. Objective

To analyze various network protocols using Wireshark to identify how data is transmitted, observe security weaknesses, and compare encrypted versus unencrypted traffic.

3. Tools Used

- Kali Linux (Wireshark installed)
- Metasploitable2 (target VM)
- Wireshark
- FTP, Telnet, SSH, HTTP clients
- Nmap, Netcat

4. Protocols Analyzed

- FTP
- HTTP
- SSH
- Telnet
- TCP
- ICMP
- ARP

5. Environment Setup

- VirtualBox/VMware with Kali Linux and Metasploitable2 VMs
- Host-only or bridged network adapter to enable communication between VMs
- Confirmed IP connectivity using `ping`

6. Packet Capture Method

- Launched Wireshark on Kali Linux
- Selected the appropriate network interface
- Captured traffic while initiating specific protocol-based communication with Metasploitable2
- Applied relevant filters in Wireshark (e.g., `ftp`, `http`, `tcp` etc.)
- Saved and documented packet captures

7. Per-Protocol Analysis (with Screenshots)

FTP (Port 21)

Network Traffic Analyzer using Wireshark

Used `ftp` command to login to Metasploitable2

Observed username and password in plain text

Security Risk: Unencrypted login credentials

HTTP (Port 80)

Accessed Metasploitable2 web server using browser

Observed `GET` and `POST` requests, and login form submissions in plain text

Security Risk: Sensitive data transmitted unencrypted

SSH (Port 22)

Logged in using `ssh`

Wireshark showed encrypted payload, no visible credentials

Security Strength: Fully encrypted

Telnet (Port 23)

Logged in using `telnet`

Observed credentials transmitted in plain text

Security Risk: Unencrypted remote access

TCP

Used `nmap -sS` for TCP SYN scan

Observed 3-way handshake: SYN, SYN-ACK, ACK

Behavior: Reliable, connection-oriented

ICMP

Used `ping` to test connectivity

Observed Echo Request and Echo Reply packets

Use Case: Network diagnostics

ARP

Observed during ping/scans

Showed IP-to-MAC address resolution

Security Risk: Vulnerable to spoofing

8. Protocol Comparison Table

9. Mitigation Suggestions

- Replace FTP/Telnet with SSH or SFTP
- Use HTTPS instead of HTTP
- Monitor ARP traffic to detect spoofing
- Restrict and log ICMP/UDP to prevent misuse

10. Conclusion

This project demonstrated how Wireshark can be used to capture and analyze network traffic from various protocols. It highlighted which protocols transmit sensitive data in plain text and which use encryption to protect communications. This knowledge is critical for identifying vulnerabilities and improving network security.

11. References

- https://www.wireshark.org/docs/
- https://nmap.org/
- https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers