Network Traffic Capture and Analysis Using Wireshark

Objective:

To capture and analyze live network packets using Wireshark, identifying different protocols and understanding their behavior in normal network communication.

Tool Used:

Wireshark (preinstalled on Kali Linux)

Environment:

Kali Linux running in a Virtual Machine (VM) using the eth0 interface.

Procedure:

1. Launched Wireshark on Kali Linux.
2. Selected the active network interface: eth0.
3. Started live packet capture.
4. Generated network traffic by visiting multiple websites (google.com, amazon.com, youtube.com) and running "ping 8.8.8.8" in the terminal.
5. Stopped the capture after around one minute.
6. Applied filters to observe different protocols (HTTP, DNS, TCP, ICMP).
7. Viewed protocol hierarchy and packet details.
8. Saved the capture file as networkCapture.pcap for analysis.

Protocols Identified:

Protocol	Description	Example Observation
DNS	Resolves domain names to IP addresses	Queries to resolve *google.com*, *amazon.com*
HTTP	Handles web traffic between browser and servers	GET requests and responses from visited websites
TCP	Transport layer protocol ensuring reliable data transmission	TCP handshakes and data segments
ICMP	Used for network diagnostics (ping)	Echo requests and replies for *8.8.8.8*
ARP	Resolves local network IPs to MAC addresses	ARP requests/replies within the VM network

Findings:

- DNS queries appeared before visiting each site, showing name resolution.
- HTTP and TCP packets dominated web traffic.
- ICMP packets confirmed external network connectivity.
- ARP packets handled local network communication between the VM and the host.

Conclusion:

Successfully captured and analyzed live network traffic using Wireshark. Identified multiple protocols at various OSI layers and observed their interaction during regular internet activity. Gained practical experience in protocol-level network analysis and packet inspection.

Deliverables:

networkCapture.pcap – captured traffic file