NSE Coursework: Network attack and defence

Project Report

Authors: Ahmed Bedair, Yukesh Shrestha, Varishdan Kumariah

April 23, 2024

Contents

Chapter 1

Level 1: Build a network and test its connectivity

1.1 Network Topology

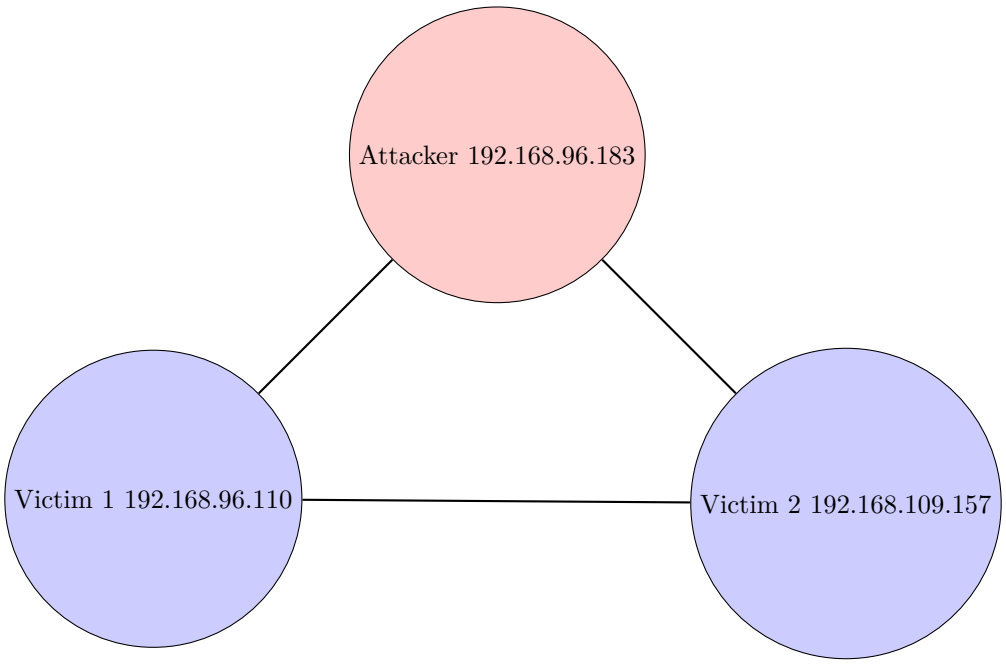

Figure 1.1: Network Topology

1.2 Connectivity Tests

The following screenshots demonstrate the successful ICMP ping tests between each machine, confirming that the machines are fully connected, as shown in Figure 1.1. The machines are also fully connected to the internet. We used the `iperf` tool instead of just browsing, as it reliably generates similar traffic on demand, further enhancing our ability to test the network's performance.

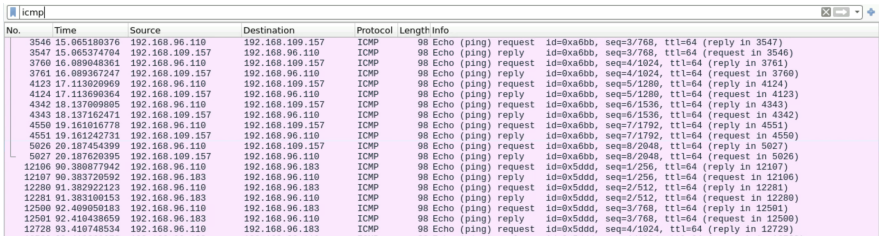

Figure 1.2: Pings from Attacker to Other Machines

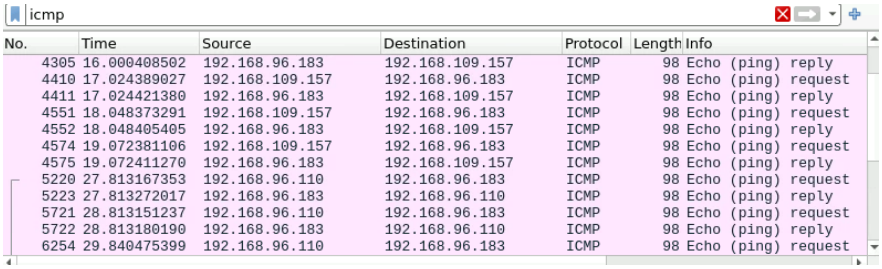

Figure 1.3: Pings from Victim 1 to Other Machines

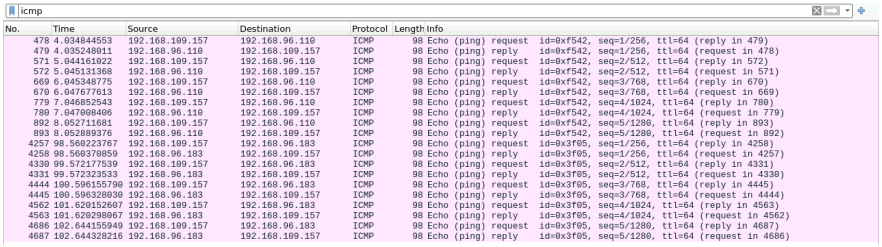

Figure 1.4: Pings from Victim 2 to Other Machines

Chapter 2

Level 2: Generate and analyse traffic on your network

2.1 Protocol Analysis

To generate network traffic, we utilised the `iperf` tool to create UDP traffic between the machines in our network. We chose UDP traffic because the `iperf` tool automatically displays performance metrics such as jitter and packet loss when sending UDP traffic. We then analysed the captured traffic using Wireshark, a powerful network protocol analyser, to gain insights into the network's behaviour and performance.

Figure 2.1 provides a comprehensive view of the protocol hierarchy of the captured traffic. We used Wireshark's built-in filters to focus specifically on the UDP traffic generated by the `iperf` tool. The protocol analysis revealed that the TCP protocol dominated the network communication when including TCP traffic, likely due to the background connections the VM was making to the rest of the KCL network. However, the UDP traffic generated by the `iperf` tool was visible as a separate protocol in the Wireshark analysis, allowing us to isolate and analyse the UDP traffic independently.

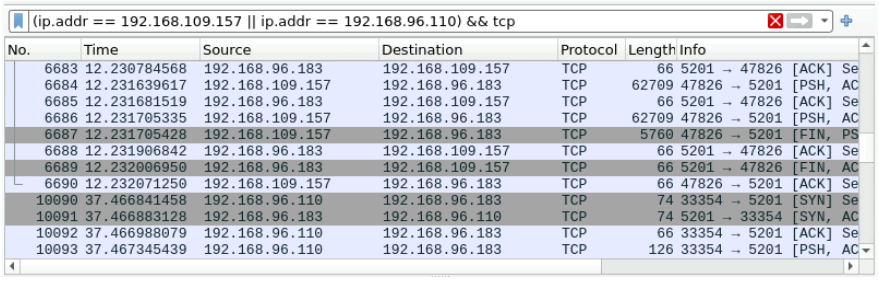

Figure 2.1: Wireshark Protocol Analysis

2.2 Packet Analysis

We can examine individual packets at the packet level to understand their structure and contents. Figure 2.2 displays a UDP packet captured during the traffic generation. The packet details show the source and destination IP addresses, ports, packet length, and checksum. This information is crucial for troubleshooting and identifying any anomalies in the network communication. While these packets are not malicious in this case, we can use this information to identify malicious packets at future levels.

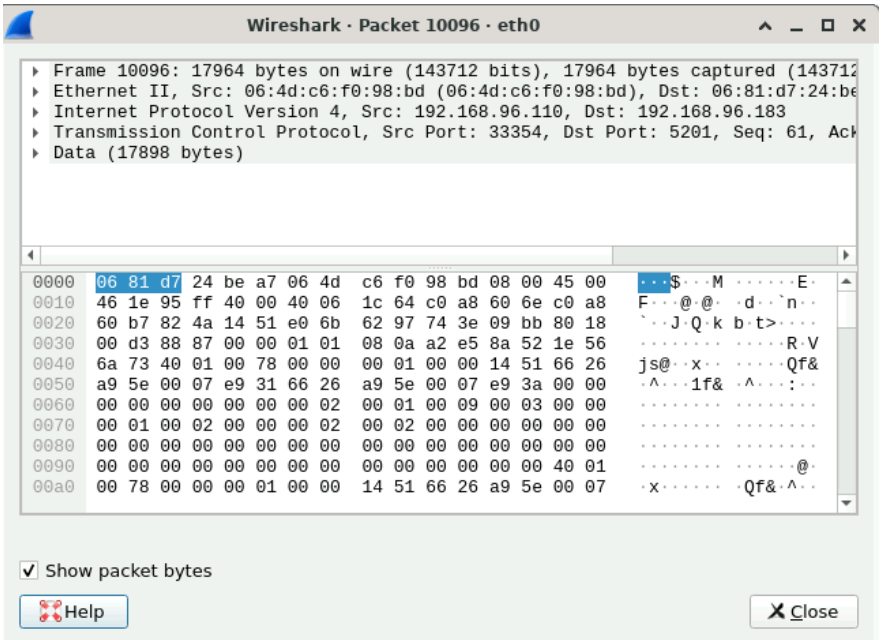

Figure 2.2: Wireshark Packet Analysis

2.3 Flow Analysis

Our analysis extended to the flow level, where we used Wireshark's Conversations feature to group packets into logical connections between endpoints. Figure 2.3 presents the UDP conversations in the captured traffic. Each row represents a unique UDP connection, showing the source and destination addresses, the number of packets exchanged, and the total amount of data transferred. This flow-level analysis demonstrates how we used `iperf` to generate UDP traffic between the machines in our network, setting up the attacker as an `iperf` server and the victims as `iperf` clients, with a separate connection between the attacker and each victim.

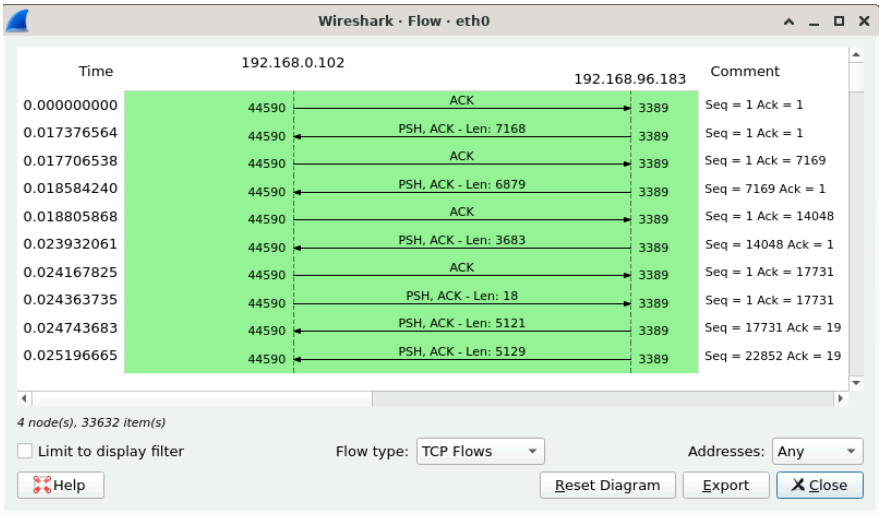

Figure 2.3: Wireshark Flow Analysis

2.4 Performance Analysis

Finally, we assessed the network performance by measuring key metrics such as throughput, delay, and packet loss. Figure 2.4 summarises the performance analysis results obtained from `iperf`. The UDP traffic achieved an average bitrate of 3.81 Gbits/sec, which, as we will observe in future levels, is exceptionally high and will drop when we introduce malicious traffic. This baseline measurement provides a reference point for evaluating the impact of network attacks on the overall performance.

Figure 2.4: Performance Analysis

Chapter 3

Level 3: Network attacks

We conducted two network attacks at this level: a TCP SYN Flood attack with IP spoofing and an ICMP Flood attack. The objective was to observe the impact of these attacks on the network and analyse the network traffic using Wireshark.

3.1 TCP SYN Flood Attack with IP Spoofing

We initiated a TCP SYN Flood attack with IP spoofing from the attacker machine, targeting one of the victim machines. Figure 3.1 illustrates the attacker's perspective during the attack, utilising the `hping3` tool to generate a flood of TCP SYN packets with spoofed source IP addresses.

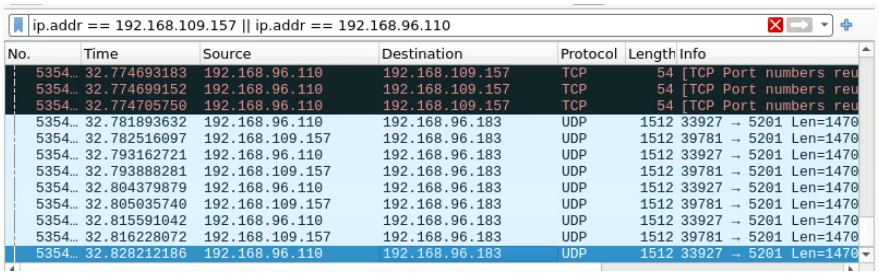

Figure 3.1: TCP SYN Flood with IP Spoofing (Attacker)

On victim 2's side, as depicted in Figure 3.2, we observed a high volume of incoming TCP SYN packets, indicating the impact of the flood attack. The victim machine attempted to respond with SYN-ACK packets. However, since the source IP addresses were spoofed to be those of

Victim 1, the responses were sent to Victim one instead of the actual attacker, allowing us to perform a denial-of-service attack on both victim machines with a single attack, demonstrating a successful amplification attack.

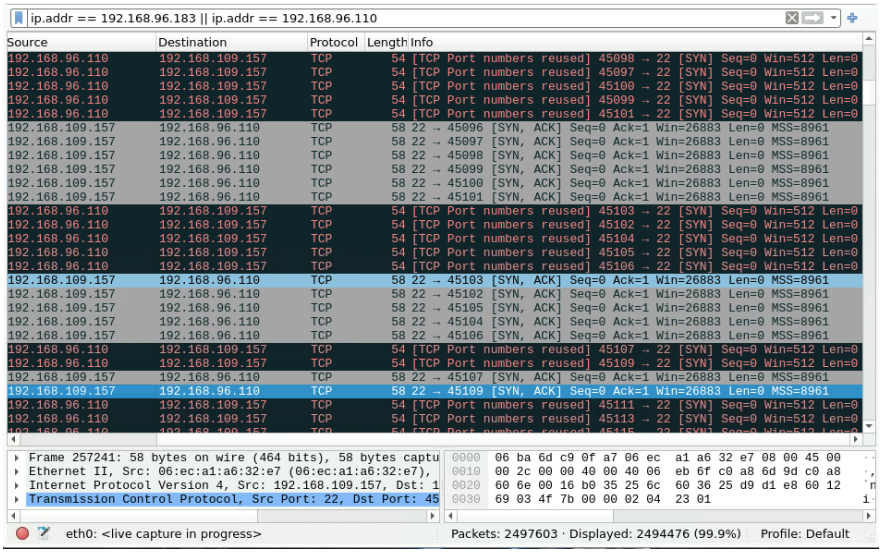

Figure 3.2: TCP SYN Flood with IP Spoofing (Victim)

3.2 ICMP Flood Attack

We also executed an ICMP Flood attack from the attacker machine, simultaneously targeting both victim machines. Figure 3.3 showcases the attacker's perspective, employing the `hping3` tool to generate a high volume of ICMP echo request packets.

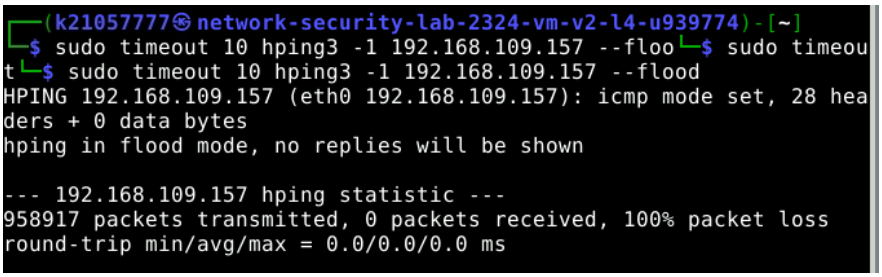

Figure 3.3: ICMP Flood (Attacker)

The victim machines were inundated with incoming ICMP echo requests, which consumed their resources and bandwidth as they attempted to process and respond to the flood of packets. This attack effectively overwhelmed the victim machines, impacting their ability to handle legitimate network traffic.

3.3 Packet Analysis

To assess the impact of the attacks on the network traffic, we captured packets using Wireshark during the attack scenarios. Figure 3.4 presents a sample of the captured packets.

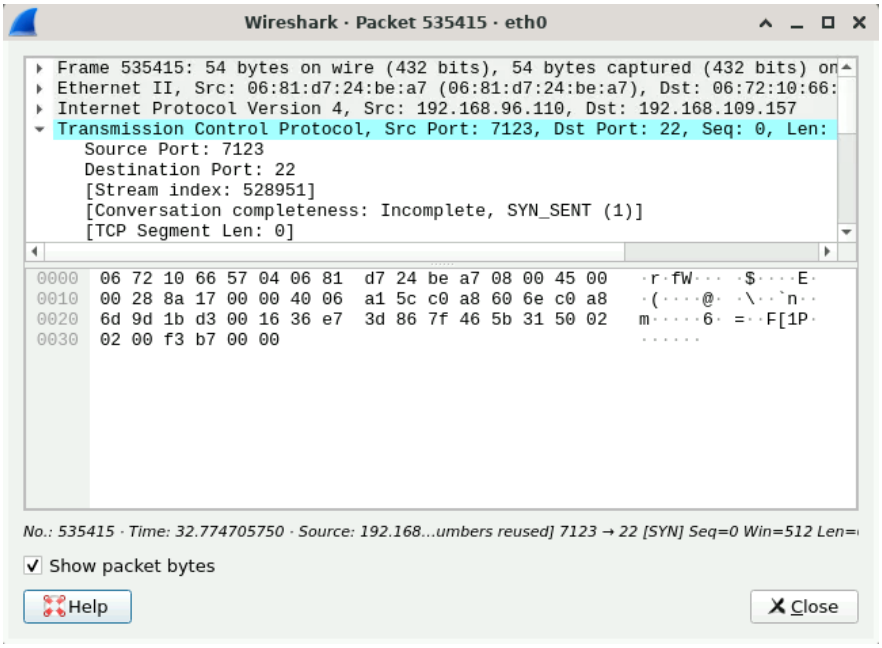

Figure 3.4: Packet Analysis

Upon closer examination of the TCP SYN packets, we discovered that the source IP addresses were spoofed, making it challenging for the victim machines to establish a complete TCP connection. Conversely, the ICMP echo request packets overwhelmed the victim machines with a barrage of requests, depleting their resources.

3.4 Flow Analysis

To gain deeper insights into the attack traffic patterns, we conducted flow analysis using Wireshark's Conversations feature. Figure 3.5 presents the flow analysis for the TCP SYN Flood attack, highlighting the substantial number of TCP flows initiated by the attacker towards the victim machine.

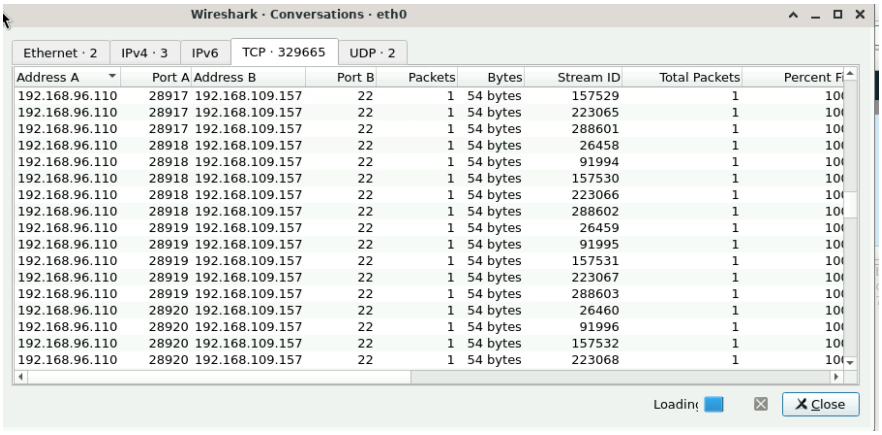

Figure 3.5: Flow Analysis 1

Similarly, Figure 3.6 illustrates the flow analysis for the ICMP Flood attack, revealing the high volume of ICMP flows originating from the attacker and directed towards both victim machines.

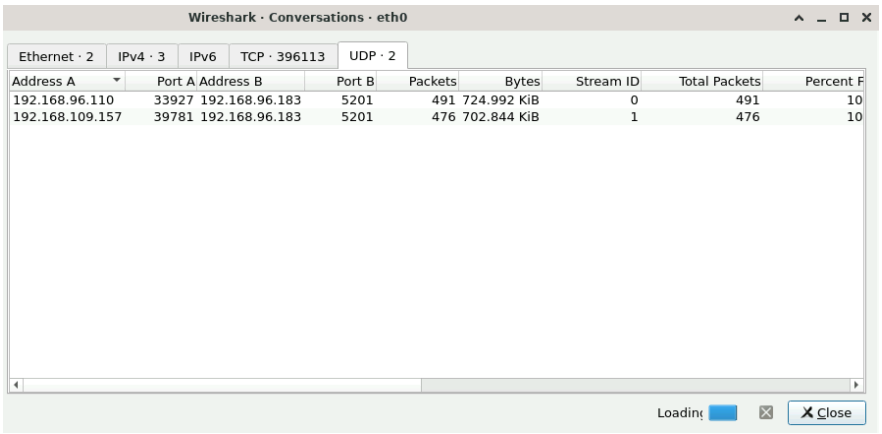

Figure 3.6: Flow Analysis 2

The flow analysis provides valuable insights into the intensity and distribution of the attack traffic, emphasising the significant impact on the targeted victim machines. By visualising the flow patterns, we can better understand the scale and effectiveness of the attacks.

Chapter 4

Level 4: Network defence

Chapter 5

Level 5: Critical evaluation and reflection