



NSE Coursework: Network attack and defence

Project Report

Authors: Ahmed Bedair, Yukesh Shrestha, Varishdan Kumariah

April 23, 2024

Contents

| | | |
|----------|--|-----------|
| 1 | Level 1: Build a network and test its connectivity | 1 |
| 1.1 | Network Topology | 1 |
| 1.2 | Connectivity Tests | 2 |
| 2 | Level 2: Generate and analyse traffic on your network | 3 |
| 2.1 | Protocol Analysis | 3 |
| 2.2 | Packet Analysis | 4 |
| 2.3 | Flow Analysis | 5 |
| 2.4 | Performance Analysis | 5 |
| 3 | Level 3: Network attacks | 7 |
| 3.1 | TCP SYN Flood Attack with IP Spoofing | 7 |
| 3.2 | ICMP Flood Attack | 8 |
| 3.3 | Packet Analysis | 9 |
| 3.4 | Flow Analysis | 9 |
| 4 | Level 4: Network defence | 11 |
| 5 | Level 5: Critical evaluation and reflection | 12 |

Chapter 1

Level 1: Build a network and test its connectivity

1.1 Network Topology

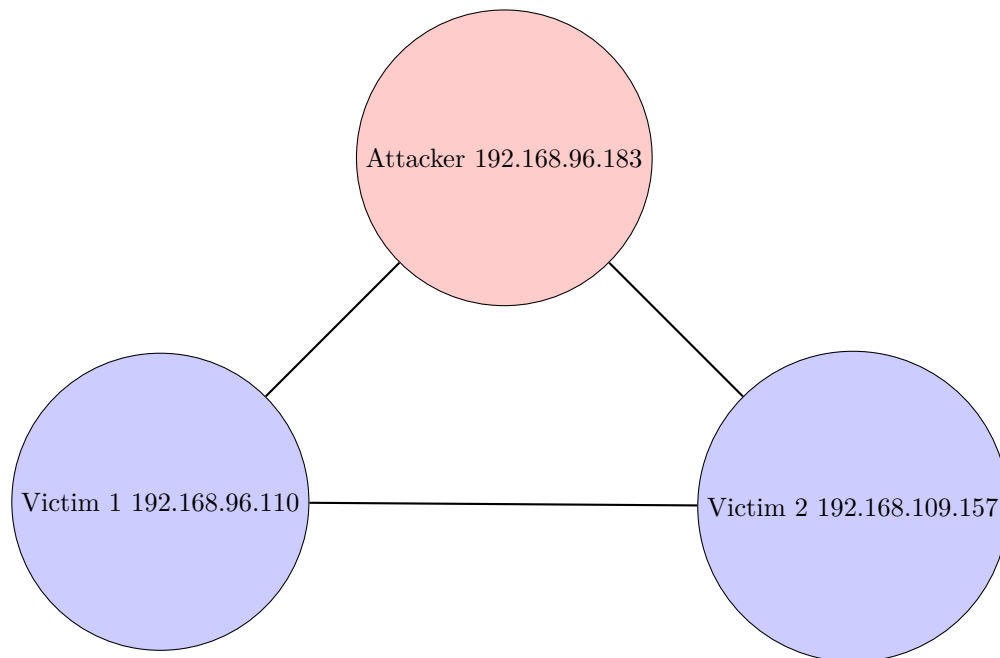
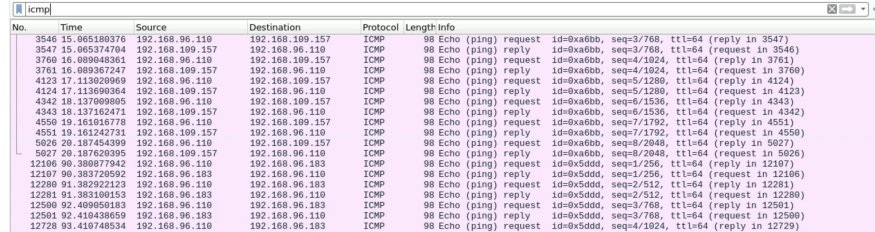


Figure 1.1: Network Topology

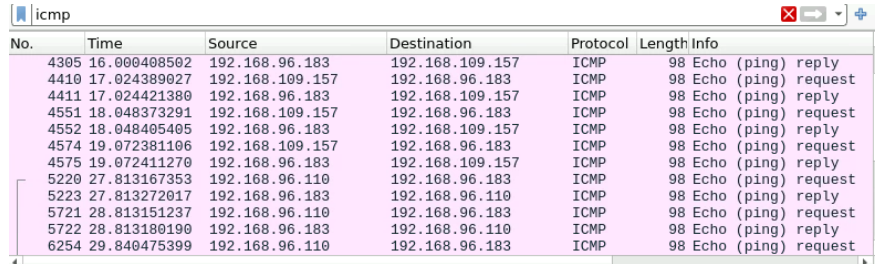
1.2 Connectivity Tests

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.



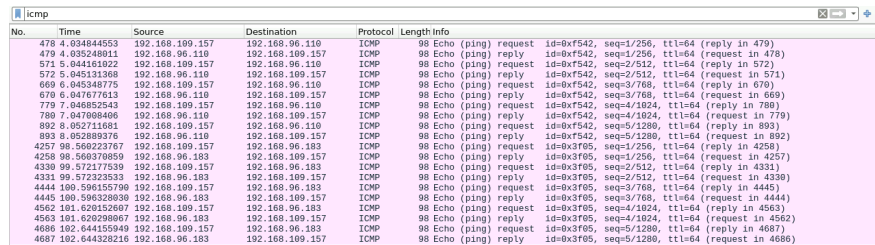
| No. | Time | Source | Destination | Protocol | Length | Info |
|-------|--------------|-----------------|-----------------|----------|--------|--|
| 3546 | 15.065180376 | 192.168.96.110 | 192.168.109.157 | ICMP | 98 | Echo (ping) request id=0xa6bb, seq=3/768, ttl=64 (reply in 3547) |
| 3547 | 15.065374784 | 192.168.109.157 | 192.168.96.110 | ICMP | 98 | Echo (ping) reply id=0xa6bb, seq=3/768, ttl=64 (request in 3546) |
| 3760 | 16.089048361 | 192.168.96.110 | 192.168.109.157 | ICMP | 98 | Echo (ping) request id=0xa6bb, seq=4/1024, ttl=64 (reply in 3761) |
| 3761 | 16.089387247 | 192.168.109.157 | 192.168.96.110 | ICMP | 98 | Echo (ping) reply id=0xa6bb, seq=4/1024, ttl=64 (request in 3760) |
| 4123 | 17.113020969 | 192.168.96.110 | 192.168.109.157 | ICMP | 98 | Echo (ping) request id=0xa6bb, seq=5/1280, ttl=64 (reply in 4124) |
| 4124 | 17.113090364 | 192.168.109.157 | 192.168.96.110 | ICMP | 98 | Echo (ping) reply id=0xa6bb, seq=5/1280, ttl=64 (request in 4123) |
| 4342 | 18.137090985 | 192.168.96.110 | 192.168.109.157 | ICMP | 98 | Echo (ping) request id=0xa6bb, seq=6/1536, ttl=64 (reply in 4343) |
| 4343 | 18.137162471 | 192.168.109.157 | 192.168.96.110 | ICMP | 98 | Echo (ping) reply id=0xa6bb, seq=6/1536, ttl=64 (request in 4342) |
| 4550 | 19.161016778 | 192.168.96.110 | 192.168.109.157 | ICMP | 98 | Echo (ping) request id=0xa6bb, seq=7/1792, ttl=64 (reply in 4551) |
| 4551 | 19.161242731 | 192.168.109.157 | 192.168.96.110 | ICMP | 98 | Echo (ping) reply id=0xa6bb, seq=7/1792, ttl=64 (request in 4550) |
| 5026 | 20.187454399 | 192.168.96.110 | 192.168.109.157 | ICMP | 98 | Echo (ping) request id=0xa6bb, seq=8/2048, ttl=64 (reply in 5027) |
| 5027 | 20.187620395 | 192.168.109.157 | 192.168.96.110 | ICMP | 98 | Echo (ping) reply id=0xa6bb, seq=8/2048, ttl=64 (request in 5026) |
| 12186 | 96.388877842 | 192.168.96.110 | 192.168.96.183 | ICMP | 98 | Echo (ping) request id=0x5ddd, seq=1/256, ttl=64 (reply in 12187) |
| 12187 | 96.393720592 | 192.168.96.183 | 192.168.96.110 | ICMP | 98 | Echo (ping) reply id=0x5ddd, seq=1/256, ttl=64 (request in 12186) |
| 12280 | 91.382922123 | 192.168.96.110 | 192.168.96.183 | ICMP | 98 | Echo (ping) request id=0x5ddd, seq=2/512, ttl=64 (reply in 12281) |
| 12281 | 91.383100153 | 192.168.96.183 | 192.168.96.110 | ICMP | 98 | Echo (ping) reply id=0x5ddd, seq=2/512, ttl=64 (request in 12280) |
| 12500 | 92.409050183 | 192.168.96.110 | 192.168.96.183 | ICMP | 98 | Echo (ping) request id=0x5ddd, seq=3/768, ttl=64 (reply in 12501) |
| 12501 | 92.410438659 | 192.168.96.183 | 192.168.96.110 | ICMP | 98 | Echo (ping) reply id=0x5ddd, seq=3/768, ttl=64 (request in 12500) |
| 12728 | 93.418748534 | 192.168.96.110 | 192.168.96.183 | ICMP | 98 | Echo (ping) request id=0x5ddd, seq=4/1024, ttl=64 (reply in 12729) |

Figure 1.2: Pings from Attacker to Other Machines



| No. | Time | Source | Destination | Protocol | Length | Info |
|------|--------------|-----------------|-----------------|----------|--------|---------------------|
| 4305 | 16.000408502 | 192.168.96.183 | 192.168.109.157 | ICMP | 98 | Echo (ping) reply |
| 4410 | 17.024389027 | 192.168.109.157 | 192.168.96.183 | ICMP | 98 | Echo (ping) request |
| 4411 | 17.024421380 | 192.168.96.183 | 192.168.109.157 | ICMP | 98 | Echo (ping) reply |
| 4551 | 18.048373291 | 192.168.109.157 | 192.168.96.183 | ICMP | 98 | Echo (ping) request |
| 4552 | 18.048405405 | 192.168.96.183 | 192.168.109.157 | ICMP | 98 | Echo (ping) reply |
| 4574 | 19.072381106 | 192.168.109.157 | 192.168.96.183 | ICMP | 98 | Echo (ping) request |
| 4575 | 19.072411270 | 192.168.96.183 | 192.168.109.157 | ICMP | 98 | Echo (ping) reply |
| 5220 | 27.813167353 | 192.168.96.110 | 192.168.96.183 | ICMP | 98 | Echo (ping) request |
| 5223 | 27.813272047 | 192.168.96.183 | 192.168.96.110 | ICMP | 98 | Echo (ping) reply |
| 5721 | 28.813151237 | 192.168.96.110 | 192.168.96.183 | ICMP | 98 | Echo (ping) request |
| 5722 | 28.813180190 | 192.168.96.183 | 192.168.96.110 | ICMP | 98 | Echo (ping) reply |
| 6254 | 29.840475399 | 192.168.96.110 | 192.168.96.183 | ICMP | 98 | Echo (ping) request |

Figure 1.3: Pings from Victim 1 to Other Machines



| No. | Time | Source | Destination | Protocol | Length | Info |
|------|---------------|-----------------|-----------------|----------|--------|---|
| 478 | 4.038444553 | 192.168.109.157 | 192.168.96.110 | ICMP | 98 | Echo (ping) request id=0xf542, seq=1/256, ttl=64 (reply in 479) |
| 479 | 4.038248011 | 192.168.96.110 | 192.168.109.157 | ICMP | 98 | Echo (ping) reply id=0xf542, seq=1/256, ttl=64 (request in 478) |
| 571 | 5.044161022 | 192.168.109.157 | 192.168.96.110 | ICMP | 98 | Echo (ping) request id=0xf542, seq=2/512, ttl=64 (reply in 572) |
| 572 | 5.045131368 | 192.168.96.110 | 192.168.109.157 | ICMP | 98 | Echo (ping) reply id=0xf542, seq=2/512, ttl=64 (request in 571) |
| 609 | 6.045348775 | 192.168.109.157 | 192.168.96.110 | ICMP | 98 | Echo (ping) request id=0xf542, seq=3/768, ttl=64 (reply in 610) |
| 610 | 6.047677613 | 192.168.96.110 | 192.168.109.157 | ICMP | 98 | Echo (ping) reply id=0xf542, seq=3/768, ttl=64 (request in 609) |
| 779 | 7.040825243 | 192.168.109.157 | 192.168.96.110 | ICMP | 98 | Echo (ping) request id=0xf542, seq=4/1024, ttl=64 (reply in 780) |
| 780 | 7.047080406 | 192.168.96.110 | 192.168.109.157 | ICMP | 98 | Echo (ping) reply id=0xf542, seq=4/1024, ttl=64 (request in 779) |
| 892 | 8.052711681 | 192.168.109.157 | 192.168.96.110 | ICMP | 98 | Echo (ping) request id=0xf542, seq=5/1280, ttl=64 (reply in 893) |
| 893 | 8.052809376 | 192.168.96.110 | 192.168.109.157 | ICMP | 98 | Echo (ping) reply id=0xf542, seq=5/1280, ttl=64 (request in 892) |
| 4257 | 98.560223767 | 192.168.109.157 | 192.168.96.183 | ICMP | 98 | Echo (ping) request id=0x3f05, seq=1/256, ttl=64 (reply in 4258) |
| 4258 | 98.560370859 | 192.168.96.183 | 192.168.109.157 | ICMP | 98 | Echo (ping) reply id=0x3f05, seq=1/256, ttl=64 (request in 4257) |
| 4330 | 99.572177539 | 192.168.109.157 | 192.168.96.183 | ICMP | 98 | Echo (ping) request id=0x3f05, seq=2/512, ttl=64 (reply in 4331) |
| 4331 | 99.572323533 | 192.168.96.183 | 192.168.109.157 | ICMP | 98 | Echo (ping) reply id=0x3f05, seq=2/512, ttl=64 (request in 4330) |
| 4444 | 100.590155790 | 192.168.109.157 | 192.168.96.183 | ICMP | 98 | Echo (ping) request id=0x3f05, seq=3/768, ttl=64 (reply in 4445) |
| 4445 | 100.590320830 | 192.168.96.183 | 192.168.109.157 | ICMP | 98 | Echo (ping) reply id=0x3f05, seq=3/768, ttl=64 (request in 4444) |
| 4562 | 101.620152607 | 192.168.109.157 | 192.168.96.183 | ICMP | 98 | Echo (ping) request id=0x3f05, seq=4/1024, ttl=64 (reply in 4563) |
| 4563 | 101.620296067 | 192.168.96.183 | 192.168.109.157 | ICMP | 98 | Echo (ping) reply id=0x3f05, seq=4/1024, ttl=64 (request in 4562) |
| 4686 | 102.644155940 | 192.168.109.157 | 192.168.96.183 | ICMP | 98 | Echo (ping) request id=0x3f05, seq=5/1280, ttl=64 (reply in 4687) |
| 4687 | 102.644328216 | 192.168.96.183 | 192.168.109.157 | ICMP | 98 | Echo (ping) reply id=0x3f05, seq=5/1280, ttl=64 (request in 4686) |

Figure 1.4: Pings from Victim 2 to Other Machines

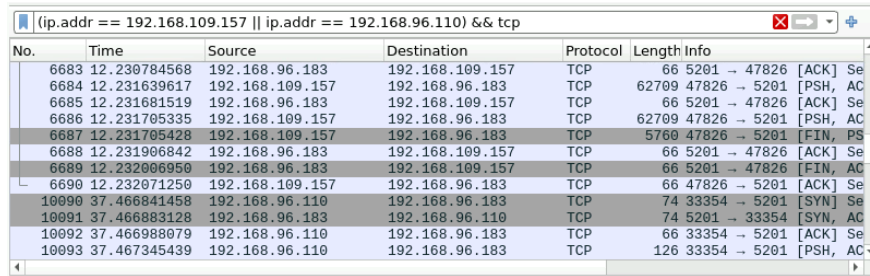
Chapter 2

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.

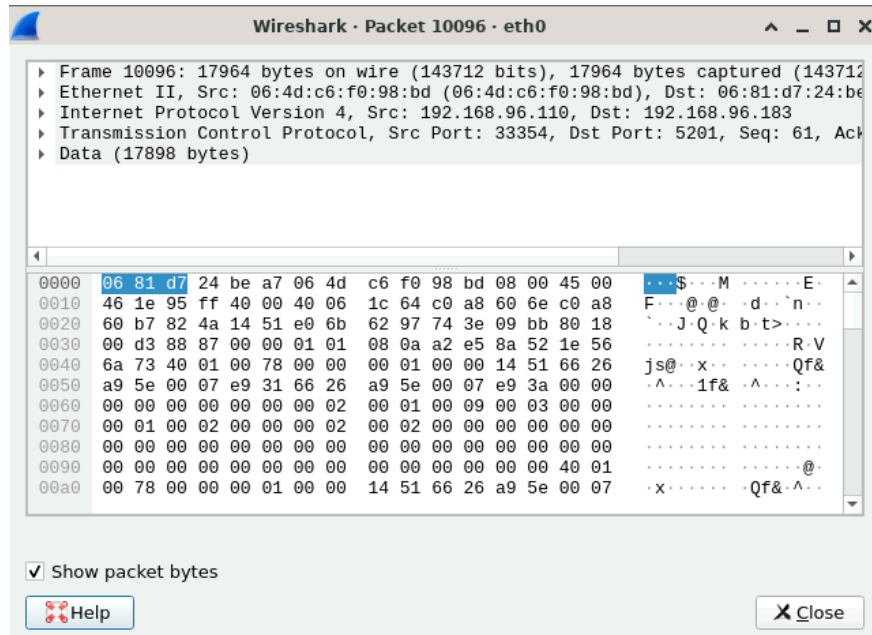


| No. | Time | Source | Destination | Protocol | Length Info |
|-------|--------------|-----------------|-----------------|----------|-----------------------------|
| 6683 | 12.230784568 | 192.168.96.183 | 192.168.109.157 | TCP | 66 5201 → 47826 [ACK] Se |
| 6684 | 12.231639617 | 192.168.109.157 | 192.168.96.183 | TCP | 62709 47826 → 5201 [PSH, AC |
| 6685 | 12.231681519 | 192.168.96.183 | 192.168.109.157 | TCP | 66 5201 → 47826 [ACK] Se |
| 6686 | 12.231705335 | 192.168.109.157 | 192.168.96.183 | TCP | 62709 47826 → 5201 [PSH, AC |
| 6687 | 12.231705428 | 192.168.109.157 | 192.168.96.183 | TCP | 5760 47826 → 5201 [FIN, PS |
| 6688 | 12.231906842 | 192.168.96.183 | 192.168.109.157 | TCP | 66 5201 → 47826 [ACK] Se |
| 6689 | 12.232006950 | 192.168.96.183 | 192.168.109.157 | TCP | 66 5201 → 47826 [FIN, AC |
| 6690 | 12.232071250 | 192.168.109.157 | 192.168.96.183 | TCP | 66 47826 → 5201 [ACK] Se |
| 10090 | 37.466841458 | 192.168.96.110 | 192.168.96.183 | TCP | 74 33354 → 5201 [SYN] Se |
| 10091 | 37.466883128 | 192.168.96.183 | 192.168.96.110 | TCP | 74 5201 → 33354 [SYN, AC |
| 10092 | 37.466988079 | 192.168.96.110 | 192.168.96.183 | TCP | 66 33354 → 5201 [ACK] Se |
| 10093 | 37.467345439 | 192.168.96.110 | 192.168.96.183 | TCP | 126 33354 → 5201 [PSH, AC |

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.



Wireshark · Packet 10096 · eth0

- Frame 10096: 17964 bytes on wire (143712 bits), 17964 bytes captured (143712 bits) on eth0
- Ethernet II, Src: 06:4d:c6:f0:98:bd (06:4d:c6:f0:98:bd), Dst: 06:81:d7:24:be:a7
- Internet Protocol Version 4, Src: 192.168.96.110, Dst: 192.168.96.183
- Transmission Control Protocol, Src Port: 33354, Dst Port: 5201, Seq: 61, Ack: 47826, Len: 17898
- Data (17898 bytes)

0000 06 81 d7 24 be a7 06 4d c6 f0 98 bd 08 00 45 00 ..\$.M...E..

0010 46 1e 95 ff 40 00 40 06 1c 64 c0 a8 60 6e c0 a8 F...@.d..n..

0020 60 b7 82 4a 14 51 e0 6b 62 97 74 3e 09 bb 80 18 `..J.Q.k b.t>...

0030 00 d3 88 87 00 00 01 01 08 0a a2 e5 8a 52 1e 56R.V

0040 6a 73 40 01 00 78 00 00 00 01 00 00 14 51 66 26 js@..x...Qf&

0050 a9 5e 00 07 e9 31 66 26 a9 5e 00 07 e9 3a 00 00 .^...1f&.^...:

0060 00 00 00 00 00 00 00 02 00 01 00 09 00 03 00 00:

0070 00 01 00 02 00 00 00 02 00 02 00 00 00 00 00 00:

0080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00:

0090 00 00 00 00 00 00 00 00 00 00 00 00 00 40 01@.

00a0 00 78 00 00 00 01 00 00 14 51 66 26 a9 5e 00 07 .x...Qf&.^...

☒ Show packet bytes

Help Close

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.

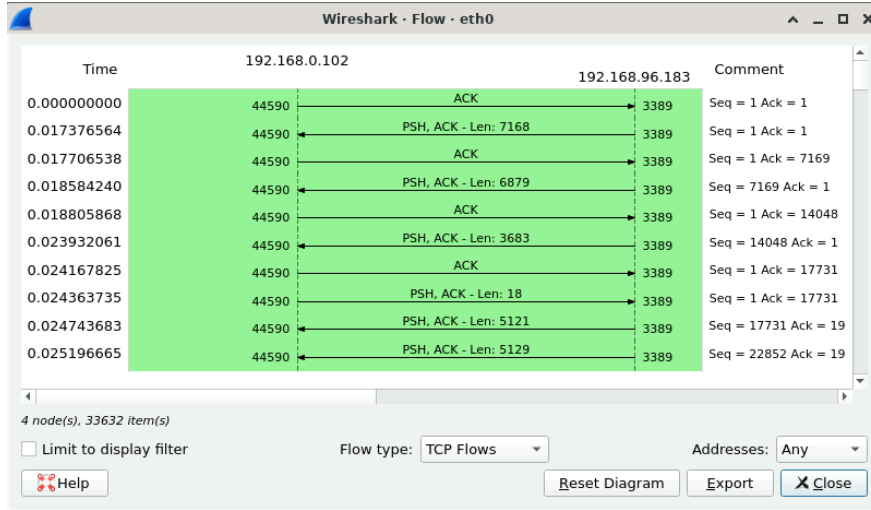


Figure 2.3: Wireshark Flow Analysis

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.

```

[ 6] local 192.168.96.183 port 5201 connected with 192.168.96.110 port 53250 (icwnd
/mss/irrt=87/8949/170)
[ ID] Interval      Transfer      Bandwidth
[ 6] 0.0000-1.0059 sec  434 MBytes  3.62 Gbits/sec
[ 7] local 192.168.96.183 port 5201 connected with 192.168.109.157 port 33474 (icwn
d/mss/irrt=87/8949/146)
[ ID] Interval      Transfer      Bandwidth
[ 7] 0.0000-1.0046 sec  456 MBytes  3.81 Gbits/sec

```

Figure 2.4: Performance Analysis

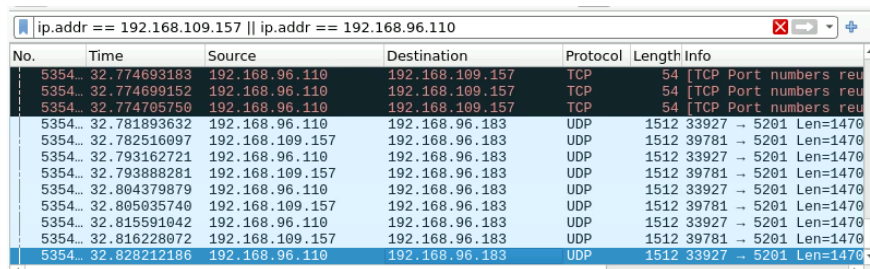
Chapter 3

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.



| No. | Time | Source | Destination | Protocol | Length | Info |
|---------|--------------|-----------------|-----------------|----------|--------|-----------------------|
| 5354... | 32.774693183 | 192.168.96.110 | 192.168.109.157 | TCP | 54 | [TCP Port numbers reu |
| 5354... | 32.774699152 | 192.168.96.110 | 192.168.109.157 | TCP | 54 | [TCP Port numbers reu |
| 5354... | 32.774705750 | 192.168.96.110 | 192.168.109.157 | TCP | 54 | [TCP Port numbers reu |
| 5354... | 32.781893632 | 192.168.96.110 | 192.168.96.183 | UDP | 1512 | 33927 → 5201 Len=1470 |
| 5354... | 32.782516097 | 192.168.109.157 | 192.168.96.183 | UDP | 1512 | 39781 → 5201 Len=1470 |
| 5354... | 32.793162721 | 192.168.96.110 | 192.168.96.183 | UDP | 1512 | 33927 → 5201 Len=1470 |
| 5354... | 32.793888281 | 192.168.109.157 | 192.168.96.183 | UDP | 1512 | 39781 → 5201 Len=1470 |
| 5354... | 32.804379879 | 192.168.96.110 | 192.168.96.183 | UDP | 1512 | 33927 → 5201 Len=1470 |
| 5354... | 32.805035740 | 192.168.109.157 | 192.168.96.183 | UDP | 1512 | 39781 → 5201 Len=1470 |
| 5354... | 32.815591042 | 192.168.96.110 | 192.168.96.183 | UDP | 1512 | 33927 → 5201 Len=1470 |
| 5354... | 32.816228072 | 192.168.109.157 | 192.168.96.183 | UDP | 1512 | 39781 → 5201 Len=1470 |
| 5354... | 32.828212186 | 192.168.96.110 | 192.168.96.183 | UDP | 1512 | 33927 → 5201 Len=1470 |

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.

| Source | Destination | Protocol | Length | Info |
|-----------------|-----------------|----------|--------|--|
| 192.168.96.110 | 192.168.109.157 | TCP | 54 | [TCP Port numbers reused] 45098 → 22 [SYN] Seq=0 Win=512 Len=0 |
| 192.168.96.110 | 192.168.109.157 | TCP | 54 | [TCP Port numbers reused] 45097 → 22 [SYN] Seq=0 Win=512 Len=0 |
| 192.168.96.110 | 192.168.109.157 | TCP | 54 | [TCP Port numbers reused] 45100 → 22 [SYN] Seq=0 Win=512 Len=0 |
| 192.168.96.110 | 192.168.109.157 | TCP | 54 | [TCP Port numbers reused] 45099 → 22 [SYN] Seq=0 Win=512 Len=0 |
| 192.168.96.110 | 192.168.109.157 | TCP | 54 | [TCP Port numbers reused] 45101 → 22 [SYN] Seq=0 Win=512 Len=0 |
| 192.168.109.157 | 192.168.96.110 | TCP | 58 | 22 → 45096 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=8961 |
| 192.168.109.157 | 192.168.96.110 | TCP | 58 | 22 → 45097 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=8961 |
| 192.168.109.157 | 192.168.96.110 | TCP | 58 | 22 → 45098 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=8961 |
| 192.168.109.157 | 192.168.96.110 | TCP | 58 | 22 → 45099 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=8961 |
| 192.168.109.157 | 192.168.96.110 | TCP | 58 | 22 → 45100 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=8961 |
| 192.168.109.157 | 192.168.96.110 | TCP | 58 | 22 → 45101 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=8961 |
| 192.168.96.110 | 192.168.109.157 | TCP | 54 | [TCP Port numbers reused] 45103 → 22 [SYN] Seq=0 Win=512 Len=0 |
| 192.168.96.110 | 192.168.109.157 | TCP | 54 | [TCP Port numbers reused] 45102 → 22 [SYN] Seq=0 Win=512 Len=0 |
| 192.168.96.110 | 192.168.109.157 | TCP | 54 | [TCP Port numbers reused] 45104 → 22 [SYN] Seq=0 Win=512 Len=0 |
| 192.168.96.110 | 192.168.109.157 | TCP | 54 | [TCP Port numbers reused] 45105 → 22 [SYN] Seq=0 Win=512 Len=0 |
| 192.168.96.110 | 192.168.109.157 | TCP | 54 | [TCP Port numbers reused] 45106 → 22 [SYN] Seq=0 Win=512 Len=0 |
| 192.168.109.157 | 192.168.96.110 | TCP | 58 | 22 → 45103 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=8961 |
| 192.168.109.157 | 192.168.96.110 | TCP | 58 | 22 → 45102 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=8961 |
| 192.168.109.157 | 192.168.96.110 | TCP | 58 | 22 → 45105 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=8961 |
| 192.168.109.157 | 192.168.96.110 | TCP | 58 | 22 → 45104 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=8961 |
| 192.168.109.157 | 192.168.96.110 | TCP | 58 | 22 → 45106 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=8961 |
| 192.168.96.110 | 192.168.109.157 | TCP | 54 | [TCP Port numbers reused] 45107 → 22 [SYN] Seq=0 Win=512 Len=0 |
| 192.168.96.110 | 192.168.109.157 | TCP | 54 | [TCP Port numbers reused] 45109 → 22 [SYN] Seq=0 Win=512 Len=0 |
| 192.168.109.157 | 192.168.96.110 | TCP | 58 | 22 → 45107 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=8961 |
| 192.168.109.157 | 192.168.96.110 | TCP | 58 | 22 → 45109 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=8961 |
| 192.168.96.110 | 192.168.109.157 | TCP | 54 | [TCP Port numbers reused] 45111 → 22 [SYN] Seq=0 Win=512 Len=0 |
| 192.168.96.110 | 192.168.109.157 | TCP | 54 | [TCP Port numbers reused] 45113 → 22 [SYN] Seq=0 Win=512 Len=0 |

| | |
|---|--|
| Frame 257241: 58 bytes on wire (464 bits), 58 bytes captured on interface eth0 | 0000 00 ba 6d c9 0f a7 06 ec a1 a6 32 e7 08 00 45 00 |
| Ethernet II, Src: 06:ec:a1:a6:32:e7 (06:ec:a1:a6:32:e7), Dst: 06:ec:a1:a6:32:e7 | 0010 00 2c 00 00 40 00 40 06 eb 6f c0 a8 6d 9d c0 a8 |
| Internet Protocol Version 4, Src: 192.168.109.157, Dst: 192.168.96.110 | 0020 60 6e 00 16 b0 35 25 6c 60 36 25 d9 d1 e8 00 12 |
| Transmission Control Protocol, Src Port: 45103, Dst Port: 22 | 0030 69 03 4f 7b 00 00 02 04 23 01 |

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.

```
(k21057777@network-security-lab-2324-vm-v2-l4-u939774) - [~]
$ sudo timeout 10 hping3 -1 192.168.109.157 --flood
$ sudo timeout 10 hping3 -1 192.168.109.157 --flood
HPING 192.168.109.157 (eth0 192.168.109.157): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown

--- 192.168.109.157 hping statistic ---
958917 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

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.

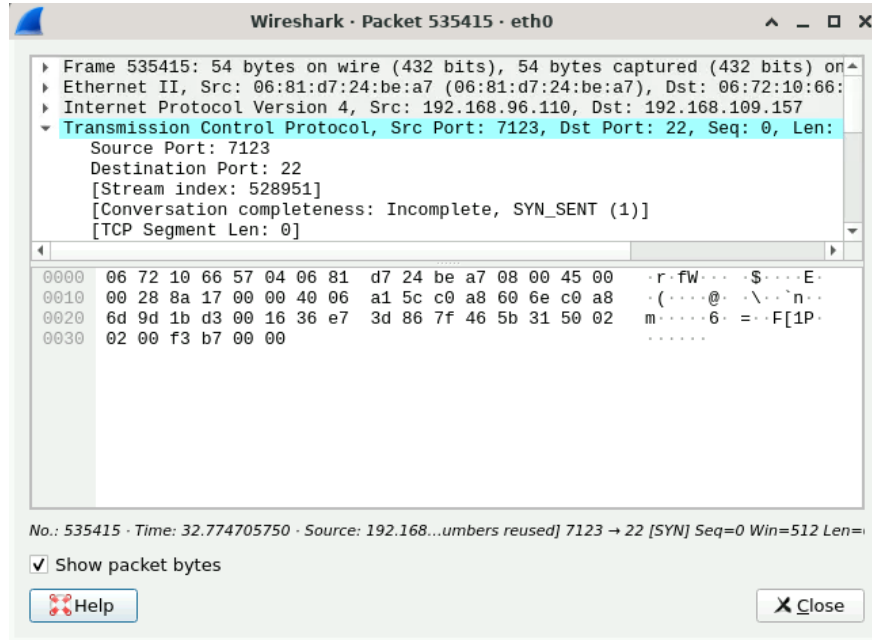


Figure 3.4: Packet Analysis

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.

| Address A | Port A | Address B | Port B | Packets | Bytes | Stream ID | Total Packets | Percent F |
|----------------|--------|-----------------|--------|---------|----------|-----------|---------------|-----------|
| 192.168.96.110 | 28917 | 192.168.109.157 | 22 | 1 | 54 bytes | 157529 | 1 | 100 |
| 192.168.96.110 | 28917 | 192.168.109.157 | 22 | 1 | 54 bytes | 223065 | 1 | 100 |
| 192.168.96.110 | 28917 | 192.168.109.157 | 22 | 1 | 54 bytes | 288601 | 1 | 100 |
| 192.168.96.110 | 28918 | 192.168.109.157 | 22 | 1 | 54 bytes | 26458 | 1 | 100 |
| 192.168.96.110 | 28918 | 192.168.109.157 | 22 | 1 | 54 bytes | 91994 | 1 | 100 |
| 192.168.96.110 | 28918 | 192.168.109.157 | 22 | 1 | 54 bytes | 157530 | 1 | 100 |
| 192.168.96.110 | 28918 | 192.168.109.157 | 22 | 1 | 54 bytes | 223066 | 1 | 100 |
| 192.168.96.110 | 28918 | 192.168.109.157 | 22 | 1 | 54 bytes | 288602 | 1 | 100 |
| 192.168.96.110 | 28919 | 192.168.109.157 | 22 | 1 | 54 bytes | 26459 | 1 | 100 |
| 192.168.96.110 | 28919 | 192.168.109.157 | 22 | 1 | 54 bytes | 91995 | 1 | 100 |
| 192.168.96.110 | 28919 | 192.168.109.157 | 22 | 1 | 54 bytes | 157531 | 1 | 100 |
| 192.168.96.110 | 28919 | 192.168.109.157 | 22 | 1 | 54 bytes | 223067 | 1 | 100 |
| 192.168.96.110 | 28919 | 192.168.109.157 | 22 | 1 | 54 bytes | 288603 | 1 | 100 |
| 192.168.96.110 | 28920 | 192.168.109.157 | 22 | 1 | 54 bytes | 26460 | 1 | 100 |
| 192.168.96.110 | 28920 | 192.168.109.157 | 22 | 1 | 54 bytes | 91996 | 1 | 100 |
| 192.168.96.110 | 28920 | 192.168.109.157 | 22 | 1 | 54 bytes | 157532 | 1 | 100 |
| 192.168.96.110 | 28920 | 192.168.109.157 | 22 | 1 | 54 bytes | 223068 | 1 | 100 |

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.

| Address A | Port A | Address B | Port B | Packets | Bytes | Stream ID | Total Packets | Percent F |
|-----------------|--------|----------------|--------|---------|-------------|-----------|---------------|-----------|
| 192.168.96.110 | 33927 | 192.168.96.183 | 5201 | 491 | 724.992 KiB | 0 | 491 | 100 |
| 192.168.109.157 | 39781 | 192.168.96.183 | 5201 | 476 | 702.844 KiB | 1 | 476 | 100 |

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