

# Cyber Security Internship Report

## Task-1

Abhijit Sudheer

August 12, 2025

### 1 Introduction

This report outlines the completion of Task 1: Introduction to Network Security Basics as part of my cybersecurity internship. The task involved researching fundamental network security concepts, implementing basic security measures on my local network using NixOS, and analyzing network traffic with Wireshark to observe these concepts in practice.

### 2 Network Security Concepts

This section provides a summary of the foundational network threats and security measures as researched for this task.

#### 2.1 Network Threats

- **Viruses:** Malicious code that attaches itself to a legitimate program or file and requires a user to execute it to spread.
- **Worms:** Self-replicating malware that spreads autonomously across a network, exploiting vulnerabilities without requiring user interaction.
- **Trojans:** Malware disguised as legitimate software. Trojans trick users into installing them, after which they can perform a variety of malicious actions.
- **Phishing:** A social engineering attack where malicious actors impersonate a trusted entity to trick users into revealing sensitive information, such as passwords or credit card numbers.

#### 2.2 Security Measures

- **Firewalls:** A security system that monitors and controls incoming and outgoing network traffic based on predefined security rules. It acts as a

```
# Enable the nftables firewall.
networking.firewall.enable = true;

# Define specific ports to allow inbound TCP traffic.
networking.firewall.allowedTCPPorts = [ 80 443 ];

# Define specific ports to allow inbound UDP traffic.
networking.firewall.allowedUDPPorts = [ 53 67 68 ];
```

Figure 1: The configuration.nix file showing the added firewall rules. These lines define the specific ports that are allowed through the firewall.

barrier between a trusted internal network and untrusted external networks.

- **Encryption:** The process of encoding information so that only authorized parties can access it. For Wi-Fi networks, protocols like **WPA2** and **WPA3** use encryption to protect data transmitted wirelessly.
- **Secure Network Configurations:** Best practices for securing a network, including changing default passwords on routers and other devices, disabling unnecessary services, and using strong authentication methods.

### 3 Implementation and Analysis

This section details the practical security measures implemented and the subsequent analysis of network traffic to verify their effectiveness.

#### 3.1 Firewall Configuration on NixOS

I configured a firewall on my NixOS system using `nftables`, which is the default firewall framework. The configuration was done through the declarative configuration.nix file. The primary goal was to enforce a "deny-by-default" security posture, allowing only essential network services to communicate with the system from the outside.

The following lines were added to my `/etc/nixos/configuration.nix` file to enable the firewall and specify the allowed ports:

```
1 networking.firewall.enable = true;
2 networking.firewall.allowedTCPPorts = [ 80 443 ];
3 networking.firewall.allowedUDPPorts = [ 53 67 68 ];
```

```

# Warning: table ip mangle is managed by iptables-nft, do not touch!
table ip mangle {
    chain PREROUTING {
        type filter hook prerouting priority mangle; policy accept;
        counter packets 778 bytes 385032 jump nixos-fw-rpfilter
    }
}

chain nixos-fw-rpfilter {
    fib saddr . mark . iif oif != 0 counter packets 778 bytes 385032 return
    udp sport 67 udp dport 68 counter packets 0 bytes 0 return
    ip saddr 0.0.0.0 ip daddr 255.255.255.255 udp sport 68 udp dport 67 counter packets 0 bytes 0 return
    counter packets 0 bytes 0 drop
}

# Warning: table ip6 mangle is managed by iptables-nft, do not touch!
table ip6 mangle {
    chain PREROUTING {
        type filter hook prerouting priority mangle; policy accept;
        counter packets 38 bytes 2520 jump nixos-fw-rpfilter
    }

    chain nixos-fw-rpfilter {
        fib saddr . mark . iif oif != 0 counter packets 36 bytes 2328 return
        counter packets 2 bytes 192 drop
    }
}

```

Figure 2: Image 2: The terminal output from `sudo nft list ruleset`, showing the active firewall rules for IPv4 and IPv6. This confirms that the rules from the configuration file were successfully applied and are actively filtering traffic.

After running `sudo nixos-rebuild switch`, I verified that the rules were active using the `nft` command. The terminal output confirmed that the specified ports were open for incoming traffic, while all other connections were logged and dropped. This provides evidence that the implemented security measures are functioning correctly.

### 3.1.1 The Importance of Port Filtering

The specific ports filtered in the configuration play a critical role in network security and functionality.

- **TCP Port 80 (HTTP) 443 (HTTPS):** These ports are essential for web browsing and web services. HTTP traffic (port 80) is unencrypted and vulnerable to snooping, which is why most modern websites use HTTPS (port 443), where data is encrypted, providing confidentiality and integrity.
- **UDP Port 53 (DNS):** This port is used for the Domain Name System, which translates human-readable domain names (e.g., `google.com`) into machine-readable IP addresses. Filtering this port would prevent a system from resolving domain names and accessing the internet.
- **UDP Ports 67 68 (DHCP):** These ports are used by the Dynamic Host Configuration Protocol, which automatically assigns IP addresses to devices on a network. They are crucial for a device to join and communicate on a local network.

Leaving unnecessary ports open can pose a significant security risk, as each open port is a potential entry point for attackers. By using a firewall to close unused

1.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	ARP	60 Gratuitous ARP for 192.168.0.1 (Request)
2.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	ARP	60 Who has 192.168.0.120? Tell 192.168.0.1
3.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	ARP	42 192.168.0.120 is at 00:10:00:00:00:00
4.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	ARP	42 Who has 192.168.0.120? Tell 192.168.0.1
5.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	ARP	60 Gratuitous ARP for 192.168.0.1 (Request)
6.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	DNS	80 Standard query 0x00000000 A google.com OPT
7.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	DNS	80 Standard query response 0x00000000 A google.com A 142.251.220.78
8.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	DNS	133 Standard query response 0x00000000 A google.com AAAA 2804:6800:4800:80c::200e AAAA 2804:6800:4800:80c::200e
9.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	1294 Initial, DCID=55e3476346580388, SCID=38b90f, PKN: 0, CRYPTO, CRYPTO
10.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	1294 0-RTT, SCID=55e3476346580388, SCID=38b90f, PKN: 0, CRYPTO, CRYPTO
11.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	1294 Initial, DCID=55e3476346580388, SCID=38b90f, PKN: 0, CRYPTO, CRYPTO
12.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	1294 0-RTT, SCID=55e3476346580388, SCID=38b90f, PKN: 0, CRYPTO, CRYPTO
13.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	1294 Initial, DCID=55e3476346580388, SCID=38b90f, PKN: 0, CRYPTO, CRYPTO
14.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	1294 0-RTT, SCID=55e3476346580388, SCID=38b90f, PKN: 0, CRYPTO, CRYPTO
15.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	85 Initial, DCID=38b90f, SCID=F5e3476346580388, PKN: 1, ACK
16.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	85 Initial, DCID=38b90f, SCID=F5e3476346580388, PKN: 2, ACK
17.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	1294 Initial, DCID=38b90f, SCID=F5e3476346580388, PKN: 3, ACK, PADDING
18.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	1294 Initial, DCID=38b90f, SCID=F5e3476346580388, PKN: 4, ACK, PADDING
19.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	1294 Initial, DCID=38b90f, SCID=F5e3476346580388, PKN: 5, ACK, PADDING
20.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	1294 Initial, DCID=38b90f, SCID=F5e3476346580388, PKN: 6, ACK, CRYPTO, PADDING
21.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	609 Protected Payload (KCP), DCID=38b90f
22.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	609 Protected Payload (KCP), DCID=38b90f
23.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	1294 Protected Payload (KCP), DCID=F5e3476346580388
24.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	1294 Protected Payload (KCP), DCID=F5e3476346580388
25.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	592 Protected Payload (KCP), DCID=F5e3476346580388
26.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	592 Protected Payload (KCP), DCID=F5e3476346580388
27.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	60 Who has 192.168.0.120? Tell 192.168.0.1
28.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	42 192.168.0.120 is at 00:10:00:00:00:00
29.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	165 Protected Payload (KCP), DCID=38b90f
30.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	60 Protected Payload (KCP), DCID=38b90f
31.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	74 Protected Payload (KCP), DCID=F5e3476346580388
32.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	73 Protected Payload (KCP), DCID=38b90f
33.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	74 Protected Payload (KCP), DCID=F5e3476346580388
34.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	74 Protected Payload (KCP), DCID=F5e3476346580388
35.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	1298 Protected Payload (KCP), DCID=38b90f
36.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	159 Protected Payload (KCP), DCID=38b90f
37.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	96 Protected Payload (KCP), DCID=38b90f
38.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	266 Protected Payload (KCP), DCID=38b90f
39.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	78 Protected Payload (KCP), DCID=F5e3476346580388
40.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	1298 Protected Payload (KCP), DCID=38b90f
41.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	75 Protected Payload (KCP), DCID=F5e3476346580388
42.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	QUIC	255 Application Data (KCP), DCID=38b90f
43.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	TLSv1.2	230 Application Data
44.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	TLSv1.2	71 57616 - 443 Len=29
45.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	TCP	40 443 - 5898 [ACK] Seq=190 Ack=73 Win=326 Len=0 TSval=1433084854 Tsecr=1427898434
46.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	UDP	71 443 - 5898 Len=29
47.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	UDP	71 443 - 5898 Len=29
48.0.0.0.0.0.0.0.0.0	192.168.0.120	192.168.0.1	TLSv1.2	145 Application Data

Figure 3: Image 3: A screenshot of the Wireshark capture window, with traffic filtered to highlight different protocols.

ports, the attack surface of the system is drastically reduced, mitigating risks such as unauthorized access, port scanning, and denial-of-service attacks.

### 3.2 Network Traffic Analysis with Wireshark

To understand how these security concepts apply in a real-world scenario, I used Wireshark to capture network traffic on my wlan0 interface. I used filters to isolate and analyze specific types of traffic, which provided a clear view of the various protocols and data packets flowing through my network.

Key observations from the analysis included:

- **DNS Traffic (UDP/53):** By using the filter `dns`, I was able to see my computer sending requests to a DNS server to resolve domain names to IP addresses, which is a fundamental part of web browsing.
- **HTTP Traffic (TCP/80):** I used the filter `http` while visiting an unencrypted website. This allowed me to capture and inspect packets containing the plain-text HTTP protocol, highlighting the security risk of unencrypted communication.
- **HTTPS Traffic (TCP/443):** By contrast, using the filter `tls` on traffic to a secure website showed encrypted data, demonstrating how HTTPS protects the integrity and confidentiality of communication.

## 4 Conclusion

This task provided a comprehensive understanding of network ports, their functions, and the role of a firewall in network security. I learned that a firewall

serves as a critical barrier for a system, and that port filtering is a fundamental technique for reducing a system's attack surface by selectively allowing or denying network traffic. The Wireshark analysis was instrumental in visually demonstrating how network protocols operate and confirming that the implemented firewall rules were actively managing network traffic. This project also highlighted the value of a layered security approach, incorporating additional practices such as regular software updates and multi-factor authentication, to ensure a system remains robustly protected and safe from potential threats.