POC Task 2

Task 2: Networking Toolkit PoC

1. Executive Summary

This PoC created an automated network diagnostic tool (`netinfo.sh`) that collects and logs system networking information. The script successfully generated a comprehensive report with IP configuration, open ports, connectivity tests, and DNS resolution.

2. Objectives

- Create an automated script to collect network information
- Log IP configuration, open ports, and connectivity tests
- Verify proper script execution under user permissions

3. Steps

3.1 Script Creation

```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo -u studentuser tee /home/studentuser/projectX/scripts/netinfo.sh << 'EOF'
#!/bin/bash

# Create network report file
report="/home/studentuser/projectX/network_report.txt"

echo "===== Network Information Report =====" > "$report"
date >> "$report"

# Display IP address, subnet mask, and default gateway
echo -e "\n[1] IP Address, Subnet Mask, and Default Gateway:" >> "$report"
ip -o -4 addr show | awk '{print "IP: "$4}' >> "$report"
ip route | grep '^default' | awk '{print "Gateway: "$3}' >> "$report"

# List open ports
echo -e "\n[2] Open Ports (ss command output):" >> "$report"
ss -tuln >> "$report"

# Ping google.com
echo -e "\n[3] Ping Test to google.com:" >> "$report"
ping -c 4 google.com >> "$report" 2>&1

# DNS resolution for openai.com
echo -e "\n[4] DNS Lookup for openai.com:" >> "$report"
nslookup openai.com >> "$report" 2>&1

echo -e "\nReport generated at: $(date)" >> "$report"
echo "Network report saved to: $report"
EOF
```

3.2 Permission Configuration

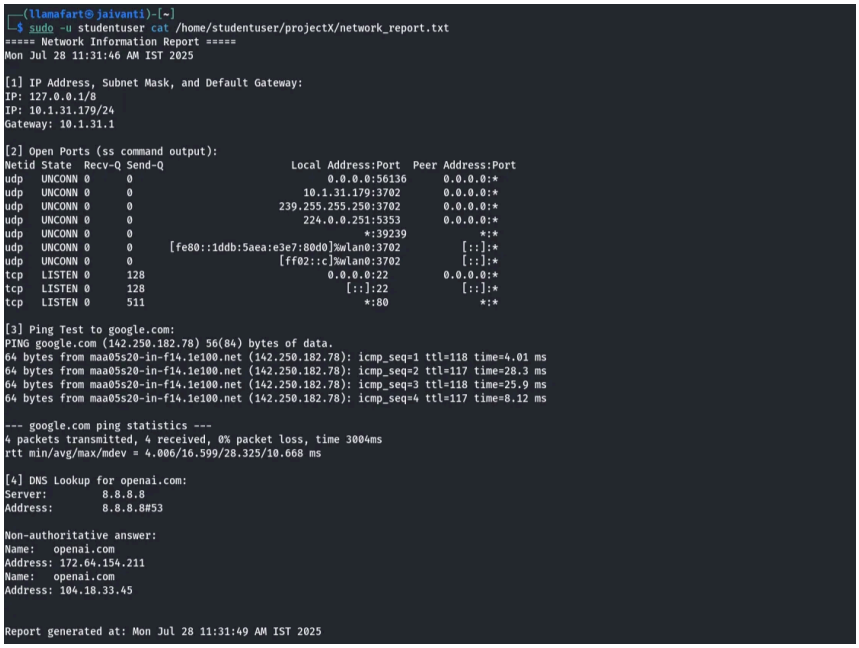
```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo chown studentuser:studentuser /home/studentuser/projectX/scripts/netinfo.sh
```

3.3 Script Execution

```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo chmod +x /home/studentuser/projectX/scripts/netinfo.sh

┌──(llamafart㉿jaivanti)-[~]
└─$ sudo -u studentuser /home/studentuser/projectX/scripts/netinfo.sh
Network report saved to: /home/studentuser/projectX/network_report.txt
```

3.4 Verification



```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo -u studentuser cat /home/studentuser/projectX/network_report.txt
===== Network Information Report =====
Mon Jul 28 11:31:46 AM IST 2025

[1] IP Address, Subnet Mask, and Default Gateway:
IP: 127.0.0.1/8
IP: 10.1.31.179/24
Gateway: 10.1.31.1

[2] Open Ports (ss command output):
Netid State  Recv-Q Send-Q                    Local Address:Port  Peer Address:Port
udp   UNCONN 0      0                             0.0.0.0:56136      0.0.0.0:*
udp   UNCONN 0      0                          10.1.31.179:3702      0.0.0.0:*
udp   UNCONN 0      0                       239.255.255.250:3702      0.0.0.0:*
udp   UNCONN 0      0                         224.0.0.251:5353      0.0.0.0:*
udp   UNCONN 0      0                                 *:39239            *:*
udp   UNCONN 0      0      [fe80::1ddb:5aea:e3e7:80d0]%wlan0:3702         [::]:*
udp   UNCONN 0      0                    [ff02::c]%wlan0:3702         [::]:*
tcp   LISTEN 0      128                           0.0.0.0:22        0.0.0.0:*
tcp   LISTEN 0      128                              [::]:22           [::]:*
tcp   LISTEN 0      511                                 *:80              *:*

[3] Ping Test to google.com:
PING google.com (142.250.182.78) 56(84) bytes of data.
64 bytes from maa05s20-in-f14.1e100.net (142.250.182.78): icmp_seq=1 ttl=118 time=4.01 ms
64 bytes from maa05s20-in-f14.1e100.net (142.250.182.78): icmp_seq=2 ttl=117 time=28.3 ms
64 bytes from maa05s20-in-f14.1e100.net (142.250.182.78): icmp_seq=3 ttl=118 time=25.9 ms
64 bytes from maa05s20-in-f14.1e100.net (142.250.182.78): icmp_seq=4 ttl=117 time=8.12 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 4.006/16.599/28.325/10.668 ms

[4] DNS Lookup for openai.com:
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   openai.com
Address: 172.64.154.211
Name:   openai.com
Address: 104.18.33.45


Report generated at: Mon Jul 28 11:31:49 AM IST 2025
```

5. Conclusion:

This PoC successfully automated network diagnostics through a secure script that executed with restricted permissions. The solution effectively collected and logged critical network configuration data while maintaining system security.