

**Title:**

## Network Security Implementation Report

**Table of Contents:**

1. Introduction
  2. Offensive and Defensive Techniques
    - 2.1 Firewall Rules and Nmap Scan Blocking with UFW
    - 2.2 Testing with Altered IP Address
    - 2.3 Snort Rule Configuration and Detection of Nmap Scans
  3. Conclusion
  4. References
- 

## 1. Introduction

This report describes the setup and configuration of two virtual machines (VMs) used for testing network security techniques involving protocol analysis, firewall rules, and intrusion detection systems. The objective is to simulate and observe network vulnerabilities and how security mechanisms can prevent or alert on network scans, specifically focusing on using the Uncomplicated Firewall (UFW) and Snort intrusion detection system.

For this assignment, two VMs were set up:

- **Attacker Machine:** Linux OS (Kali GNU/Linux Rolling Release:2024.4), with IP address [192.168.140.128].
- **Target Machine:** Linux OS (Ubuntu 24.04.1 LTS Release: 24.04), with IP address [192.168.140.132].

The target machine is configured to observe packets and assess the effectiveness of security defenses against scans initiated from the attacker machine.

---

## 2. Offensive and Defensive Techniques

### 2.1 Firewall Rules and Nmap Scan Blocking with UFW

To prevent Nmap scans from the attacking machine, UFW firewall rules were configured on the target machine. UFW serves as an interface for managing the Linux kernel's `netfilter` system, enabling the specification of rules to control incoming and outgoing traffic.

#### 1. Configuring the Firewall Rule

- The UFW rule was set to block all incoming Nmap scans by blocking the attacker machine's IP address. The following command was used:

```
sudo ufw deny from 192.168.140.128
```

```

Nov 26 09:45
mishal@mishal-VMware-Virtual-Platform: ~/Desktop
mishal@mishal-VMware-Virtual-Platform:~/Desktop$ sudo ufw deny from 192.168.140.128
Rules updated
mishal@mishal-VMware-Virtual-Platform:~/Desktop$

```

- **Screenshot 1:** Insert a screenshot of the UFW command execution here.
2. **Verifying the Firewall Rule**
- To confirm that the rule was active, the UFW status was checked to verify the rule addition.

```

Nov 26 09:48
mishal@mishal-VMware-Virtual-Platform: ~/Desktop
mishal@mishal-VMware-Virtual-Platform:~/Desktop$ sudo ufw status
Status: active

To Action From
--
Anywhere DENY 192.168.140.128

mishal@mishal-VMware-Virtual-Platform:~/Desktop$

```

- **Screenshot 2:** Insert a screenshot of the firewall rule confirmation here.
3. **Running Nmap Scan and Observing with Wireshark**
- An Nmap scan was initiated from the attacking machine to test if the firewall blocked the packets.
  - On the target machine, Wireshark was used to capture packets and observe any blocked or allowed connections.

Nov 26 09:53

No.	Time	Source	Destination	Protocol	Length	Info
1	28.698	192.168.140.128	192.168.140.2	TCP	60	40888 → 14441 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1	28.734	192.168.140.128	192.168.140.2	TCP	60	40888 → 7778 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1	28.738	192.168.140.128	192.168.140.2	TCP	60	40888 → 6502 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1	28.738	192.168.140.128	192.168.140.2	TCP	60	40888 → 7402 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1	28.738	192.168.140.128	192.168.140.2	TCP	60	40888 → 45100 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1	28.742	192.168.140.128	192.168.140.2	TCP	60	40890 → 5810 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1	28.742	192.168.140.128	192.168.140.2	TCP	60	40890 → 8333 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1	28.746	192.168.140.128	192.168.140.2	TCP	60	40890 → 16018 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1	28.746	192.168.140.128	192.168.140.2	TCP	60	40890 → 5209 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1	28.746	192.168.140.128	192.168.140.2	TCP	60	40890 → 7097 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1	28.791	192.168.140.128	192.168.140.2	TCP	60	40890 → 14441 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1	28.836	192.168.140.128	192.168.140.2	TCP	60	40890 → 7778 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1	28.839	192.168.140.128	192.168.140.2	TCP	60	40890 → 45100 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1	28.839	192.168.140.128	192.168.140.2	TCP	60	40890 → 7402 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
1	28.839	192.168.140.128	192.168.140.2	TCP	60	40890 → 6502 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Frame 3: 88 bytes on wire (704 bits), 88 bytes captured (704 bits) on interface  
 Ethernet II, Src: VMware\_e0:c7:ba (00:0c:29:e0:c7:ba), Dst: VMware\_f1:b1:41 (0  
 Internet Protocol Version 4, Src: 192.168.140.128, Dst: 192.168.140.2  
 User Datagram Protocol, Src Port: 53888, Dst Port: 53  
 Domain Name System (query)

- **Screenshot 3:** Insert screenshots of the Wireshark capture showing blocked packets here.
4. **Results of the Nmap Scan**
- Following the configuration of the UFW rule, the Nmap scan's outcome was analyzed. Wireshark captures indicated that the firewall effectively blocked

packets from the attacker's IP address, as seen by the lack of response in the Wireshark capture.

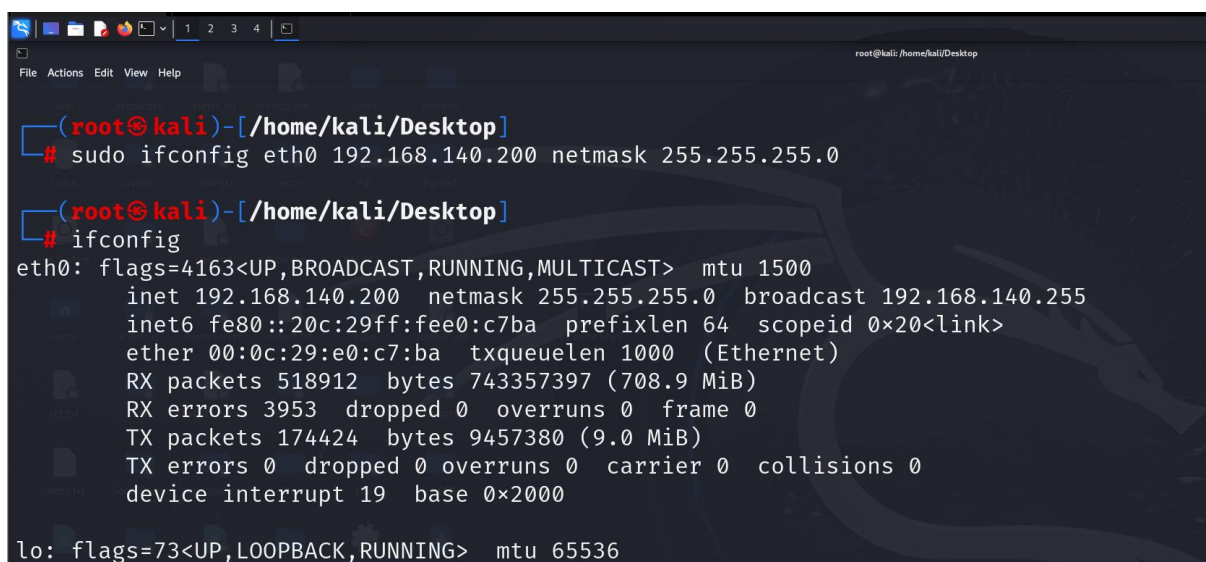
- This result confirms that the UFW rule effectively prevented Nmap from gaining information about open ports on the target machine.

## 2.2 Testing with Altered IP Address

To further assess the UFW's filtering effectiveness, the IP address of the attacking machine was changed, followed by a repeat Nmap TCP scan to verify whether the firewall rule still applied.

### 1. Changing the Attacker IP Address

- The attacker machine's IP was changed using network configuration tools.



```
(root@kali)~[/home/kali/Desktop]
# sudo ifconfig eth0 192.168.140.200 netmask 255.255.255.0

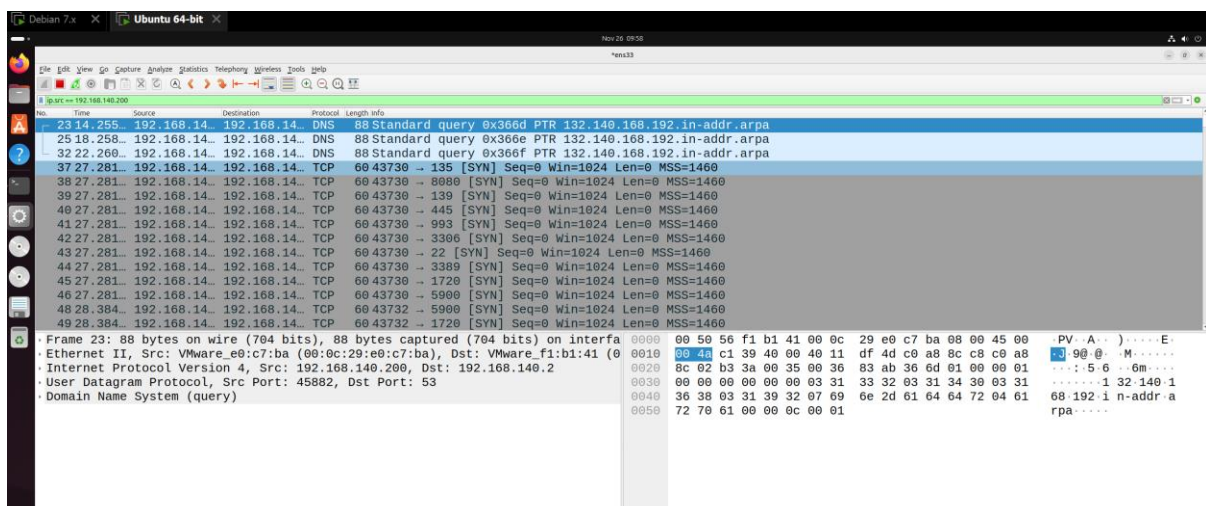
(root@kali)~[/home/kali/Desktop]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.140.200 netmask 255.255.255.0 broadcast 192.168.140.255
    inet6 fe80::20c:29ff:fee0:c7ba prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:e0:c7:ba txqueuelen 1000 (Ethernet)
    RX packets 518912 bytes 743357397 (708.9 MiB)
    RX errors 3953 dropped 0 overruns 0 frame 0
    TX packets 174424 bytes 9457380 (9.0 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 19 base 0x2000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
```

- **Screenshot 4:** Insert a screenshot showing the new IP address of the attacker machine.

### 2. Nmap Scan with Changed IP Address

- After changing the IP, an Nmap TCP scan was run again. With no specific rule for this new IP in the firewall, the packets were not blocked by UFW.



```
Debian 7.x  x  Ubuntu 64-bit  x
Nov 26 09:55
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
192.168.140.200
No. Time Source Destination Protocol Length Info
23 14.255 192.168.14. 192.168.14. DNS 88 Standard query 0x366d PTR 132.140.160.192.in-addr.arpa
25 18.258 192.168.14. 192.168.14. DNS 88 Standard query 0x366e PTR 132.140.160.192.in-addr.arpa
32 22.260 192.168.14. 192.168.14. DNS 88 Standard query 0x366f PTR 132.140.160.192.in-addr.arpa
37 27.281 192.168.14. 192.168.14. TCP 60 43730 -> 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
38 27.281 192.168.14. 192.168.14. TCP 60 43730 -> 8080 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
39 27.281 192.168.14. 192.168.14. TCP 60 43730 -> 139 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
40 27.281 192.168.14. 192.168.14. TCP 60 43730 -> 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
41 27.281 192.168.14. 192.168.14. TCP 60 43730 -> 993 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
42 27.281 192.168.14. 192.168.14. TCP 60 43730 -> 3390 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
43 27.281 192.168.14. 192.168.14. TCP 60 43730 -> 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
44 27.281 192.168.14. 192.168.14. TCP 60 43730 -> 3389 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
45 27.281 192.168.14. 192.168.14. TCP 60 43730 -> 1720 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
46 27.281 192.168.14. 192.168.14. TCP 60 43730 -> 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
48 28.384 192.168.14. 192.168.14. TCP 60 43732 -> 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
49 28.384 192.168.14. 192.168.14. TCP 60 43732 -> 1720 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
Frame 23: 88 bytes on wire (704 bits), 88 bytes captured (704 bits) on interface eth0
Ethernet II, Src: VMware_e0:c7:ba (00:0c:29:e0:c7:ba), Dst: VMware_f1:b1:41 (00:0c:29:f1:b1:41)
Internet Protocol Version 4, Src: 192.168.140.200, Dst: 192.168.140.2
User Datagram Protocol, Src Port: 45882, Dst Port: 53
Domain Name System (query)
0000 00 50 56 f1 b1 41 00 0c 29 e0 c7 ba 00 00 45 00
0010 80 0a c1 39 40 00 40 11 df 4d c0 a8 8c c8 c0 a8
0020 8c 02 b3 3a 00 35 00 36 83 ab 36 6d 01 00 00 01
0030 00 00 00 00 00 00 03 31 33 32 03 31 34 30 03 31
0040 36 38 03 31 39 32 07 69 6e 2d 61 64 64 72 04 61
0050 72 70 61 00 00 0c 00 01
```

- **Screenshot 5:** Insert Wireshark capture showing packets from the new attacker IP.

### 3. Implications of IP Address Change

- This test illustrates the limitation of static IP-based firewall rules, as the attacker machine could bypass the firewall by changing its IP address. The results highlight the need for dynamic or behavioral-based detection systems, which are less dependent on IP address restrictions alone.

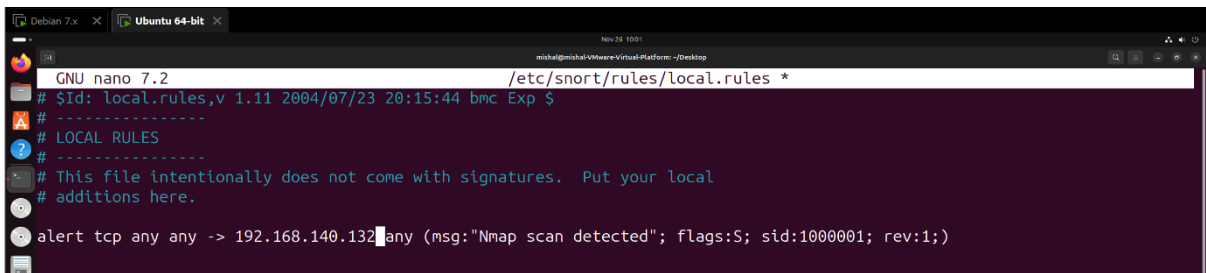
## 2.3 Snort Rule Configuration and Detection of Nmap Scans

To address the limitation of UFW's static IP filtering, Snort was used to write a rule that could detect the signature of an Nmap TCP scan, regardless of the attacker's IP address.

### 1. Setting Up Snort

- Snort was installed and configured on the target machine as an intrusion detection system (IDS).
- The Snort rule created was aimed at identifying the unique characteristics of an Nmap TCP scan, which typically includes a series of SYN packets probing various ports.
- The rule to detect the Nmap scan was written as follows:

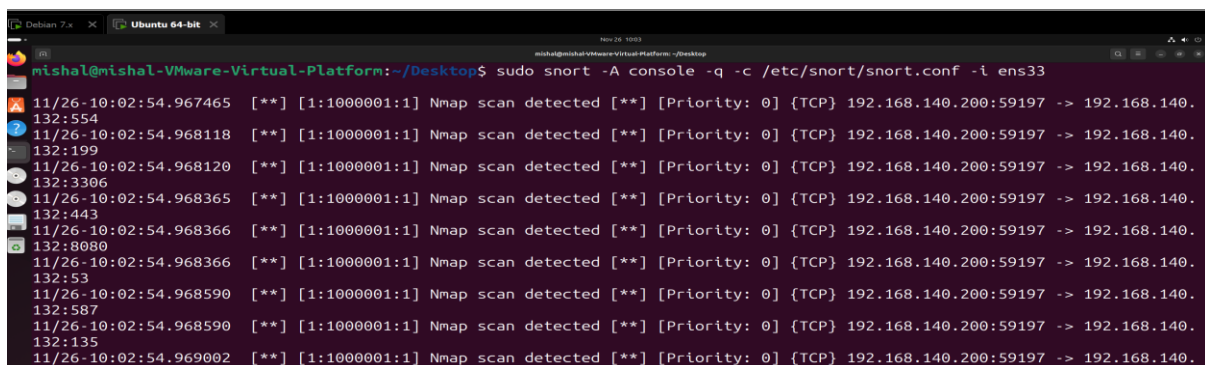
```
alert tcp any any -> [Enter Target IP here] any (msg:"Nmap scan detected"; flags:S; sid:1000001; rev:1;)
```



- **Screenshot 6:** Insert a screenshot of the Snort rule configuration file with the rule details.

### 2. Testing the Snort Rule

- With the Snort rule in place, an Nmap TCP scan was initiated from the attacker machine.
- When Snort detected the scan pattern, it generated an alert, indicating successful detection.



- **Screenshot 7:** Insert screenshots showing Snort alerts generated by the Nmap scan.
3. **Analysis of Snort's Detection Capabilities**
- Snort's alert generation provided evidence that the Nmap TCP scan was successfully detected, showcasing Snort's strength in identifying attack patterns without relying on IP-based filtering.
  - This approach highlights the advantage of IDS systems like Snort in enhancing security by identifying behaviors rather than relying solely on static firewall rules.
- 

### 3. Conclusion

The exercises detailed in this report demonstrate key principles in network security, including the detection and prevention of reconnaissance scans using UFW and Snort. UFW was effective in blocking traffic based on IP but had limitations when the attacker's IP address changed. Snort, however, provided a more flexible detection mechanism by monitoring packet behavior, which is more effective in scenarios with changing IP addresses or when attackers disguise their identities.

These findings underscore the importance of layered network defenses, where both firewalls and IDS systems work together to address various vulnerabilities. Through this hands-on exploration, insights were gained into protocol behavior, firewall rules, and intrusion detection techniques, furthering understanding of real-world network security practices.

---

### 4. References

Provide references to all tools, manuals, and sources used, such as:

- UFW documentation: <https://launchpad.net/u fw>
  - Snort documentation: <https://www.snort.org/>
  - Nmap reference guide: <https://nmap.org/>
  - Relevant textbooks or articles on network security, if applicable.
- 

**Note:** Throughout the report, remember to number each figure and add captions, such as “Figure 1: UFW command to block Nmap scan.” This will keep the report organized and ensure it meets the formatting criteria.

This structure and content outline should provide a strong base for completing your report. Add the screenshots in the designated spots for a comprehensive and well-documented submission

- **UFW:** Used on the target machine to set up firewall rules intended to block traffic from the attacker.
- **Nmap:** Used on the attacker machine to conduct various scans on the target machine.
- **Wireshark:** Running on the target machine to capture and display network packets, providing visual confirmation of the UFW and Snort protections.
- **Snort:** Also on the target machine, Snort detects patterns like the Nmap scan itself and raises alerts if it recognizes potentially malicious scanning behavior.