Naima Mumin
https://www.linkedin.com/in/naima-mumin-a48750270/

Uncovering Network Vulnerabilities Using Snort and Nmap

Objective:

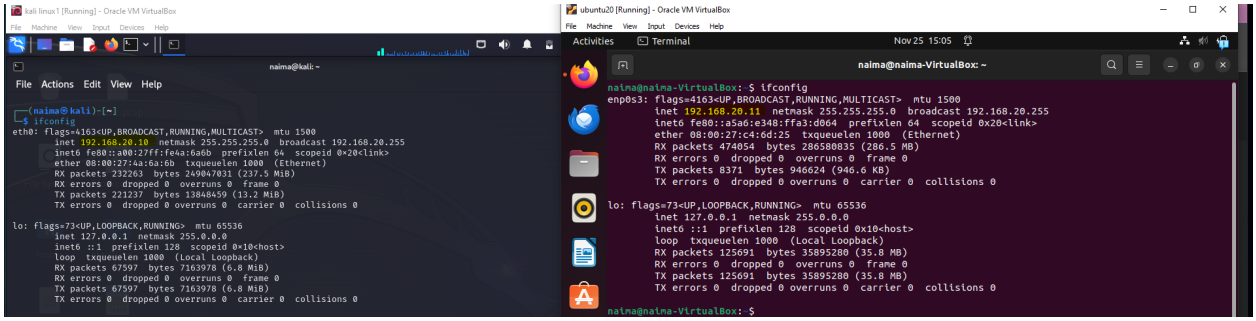
- In this project, my goal is to delve deeply into understanding the detection of unauthorized access within networks, employing tools like Snort and Nmap. By trying things out and looking closely at what happens, I hope to find strong ways to make networks safer from unauthorized access.

Requirement:

- Operating System:
  - Linux and Ubuntu
- Tools:
  - Snort (Intrusion Detection System)
  - Nmap (Vulnerability Scanner)

1. Determine which Operating system Ip address will be using target and host

……..



2. Open the snort file where I will be writing the rules

Commands:

- ls
- cd etc/snort/rules
- sudo vim local.rules

3. **TCP SYN Rule**:

- Rule:

- alert tcp any any -> 192.168.20.14 any (flags: s; content: “Nmap”; msg: “Nmap TCP SYN Traffic Detected1!!”; sid: 100001;)

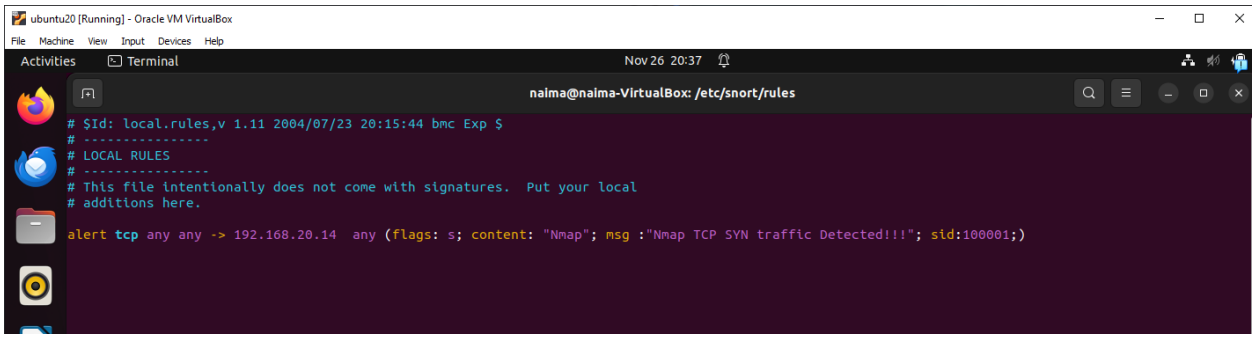
Explanation of this rule:

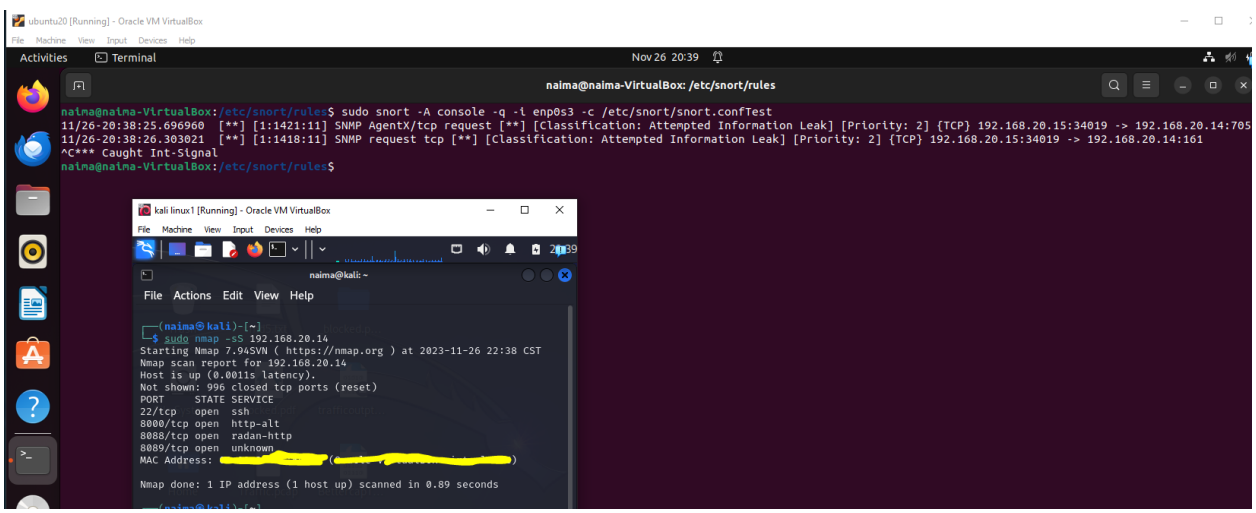
- This rule will alert me if it detects internet messages trying to connect my ubuntu Ip address(192168.20.14) using the SYN flag and containing the term “Nmap”
- This SYN scan will help me see if someone is trying to check into my computer to see which ports are open for communication without fully establishing a complete connection. By using this rule, I can identify the person trying this technique, helping me stay aware and take action if needed.

- Nmap scan:
  - sudo nmap -sS 192.168.20.14

Explanation:

- Running this nmap scan and tries to do SYN scan, but the rule I wrote it in my snort will detect it





4. TCP Rule

Detecting TCP Port 443

- Rule:
  - alert tcp any any -> 192.168.20.11 443 (msg: “TCP port 443 Detected!”; sid: 100002;)

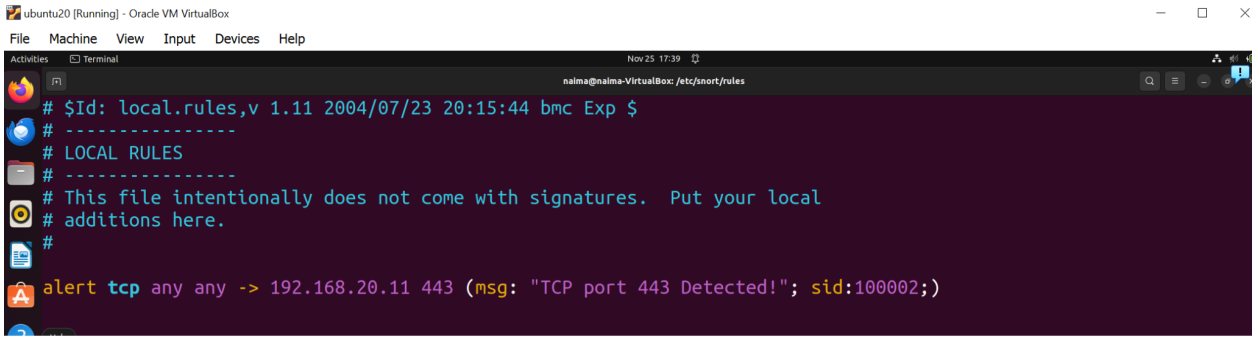
Explanation:

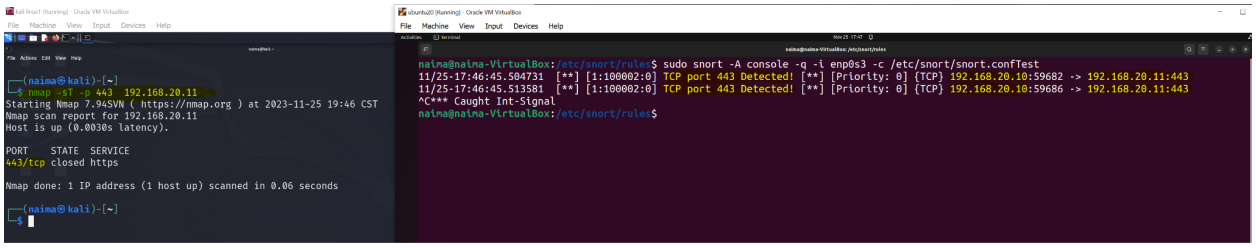
- This alerts me if there's an attempt from anywhere on the internet to connect to port 443 of the computer at 192.168.20.11 using the TCP protocol. If this happens, it will alert me saying "TCP port 443 Detected!" so I can take an action from that on

- Nmap scan:
  - Nmap -sT -p 443 192.168.20.11

Explanation:

- I'm using Nmap to check if port 443 is open on the device at IP address 192.168.20.11





5. SSH Rule

- Rule:
  - alert tcp 192.168.20.10 any -> HOME_NET 22 (msg:"SSH Detected!!"; sid: 100003;)

Explanation:

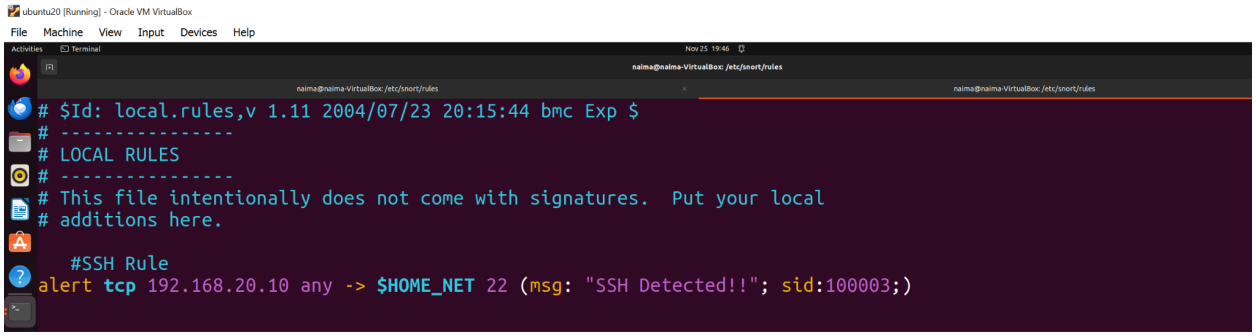
- I made this rule to alert me if my Linux device at 192.168.20.10 tries to access SSH on port 22 within my network identified as HOME_NET, which is my Ubuntu machine. It's a way to watch specifically for SSH activity from my Linux device inside my network.
- Check if my SSH is allowed in my linux machine

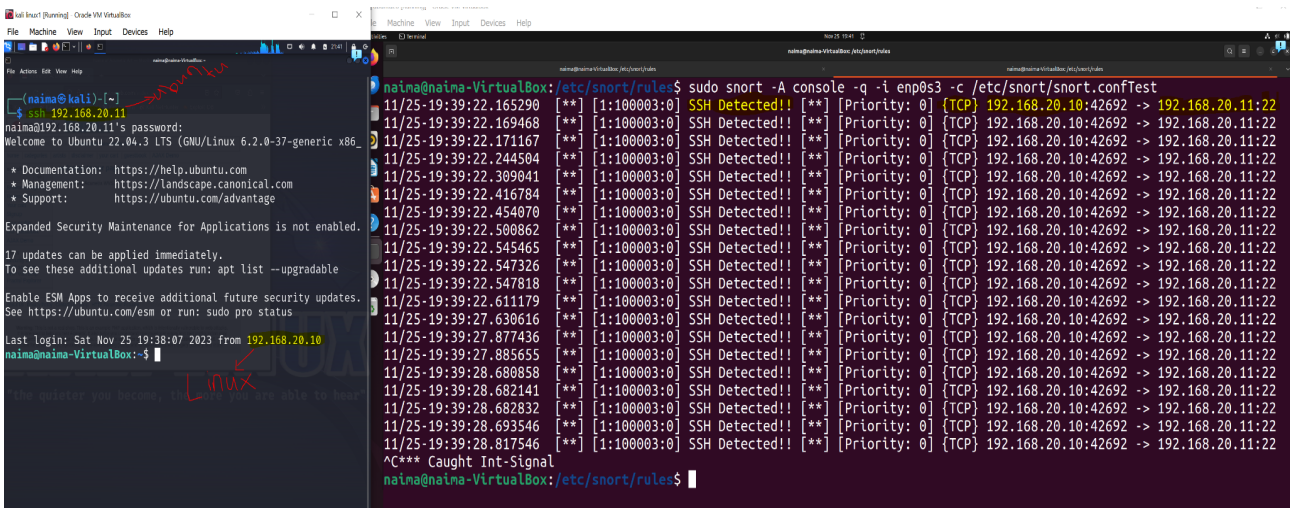
Command:

- sudo ufw status
- Sudo ufw systemctl status ssh
- Sudo systemctl enable ssh
- Sudo systemctl start ssh

Command I need to run:

- ssh 192.168.20.11





6. FIN RULE

- Rule:

- Alert Tcp 192.168.20.15 any -> 192.168.20.14 any (msg: “Nmap FIN Scan Detected!”; sid: 1000004;)

Explanation of the Rule:

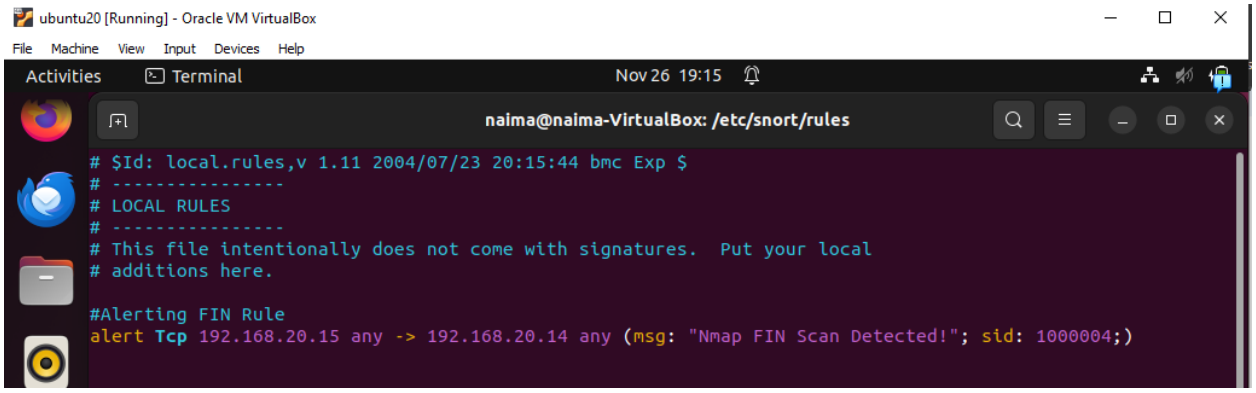
- This rule helps me know if a device at IP address 192.168.20.15 tries to quietly scan my network’s ports using a tricky signal. If it detects this, it logs it as “Nmap FIN Scan Detected!” with the ID 1000004 for me to check

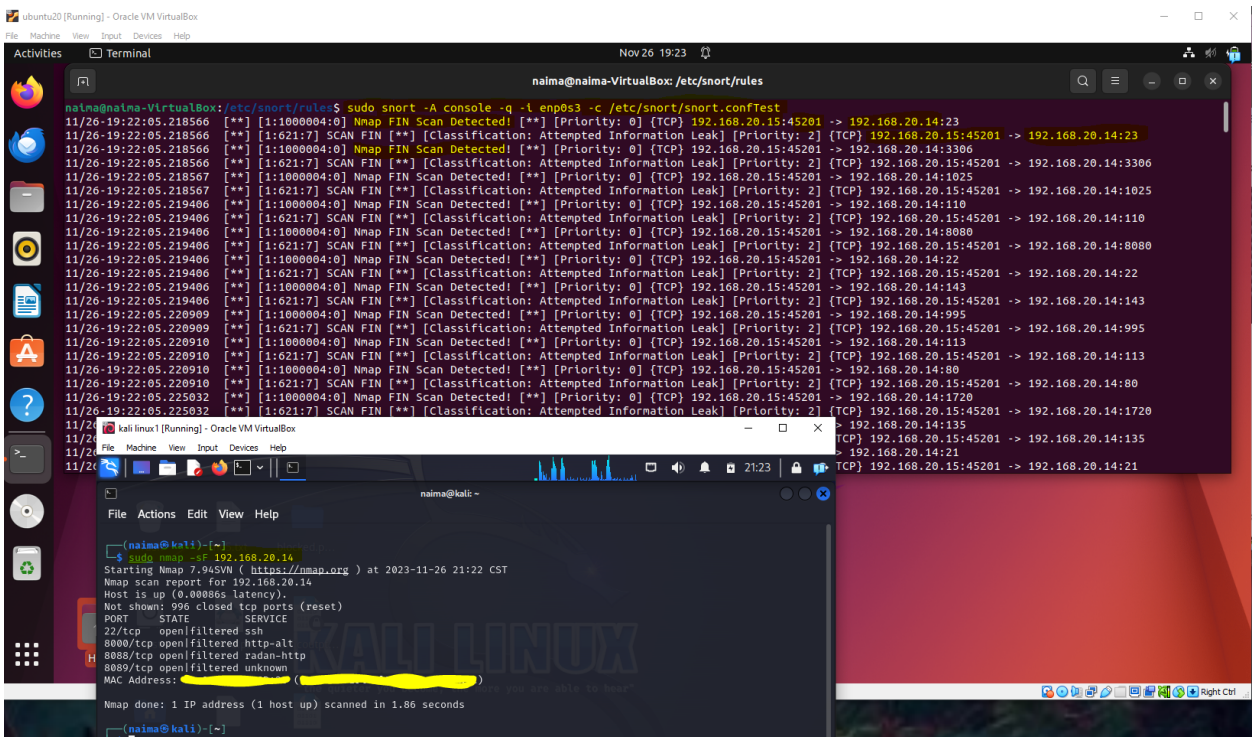
- Nmap Scan:

- nmap -sF 192.168.20.14

Explanation:

- This is the stealthily scan that attempts to discover open ports on the target device by sending TCP packets with the FIN flag set.





5. **Alerting and Rejecting ICMP Rules**

a. Alerting ICMP Rule

- Rule:
  - alert ICMP 192.168.20.10 any -> 192.168.20.11 any (msg: “ICMP Ping scan Detected!”; sid:1000005;)

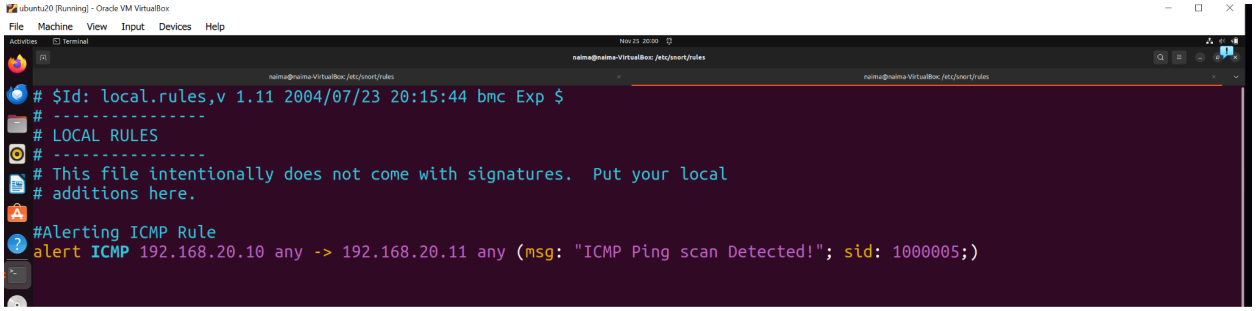
Explanation:

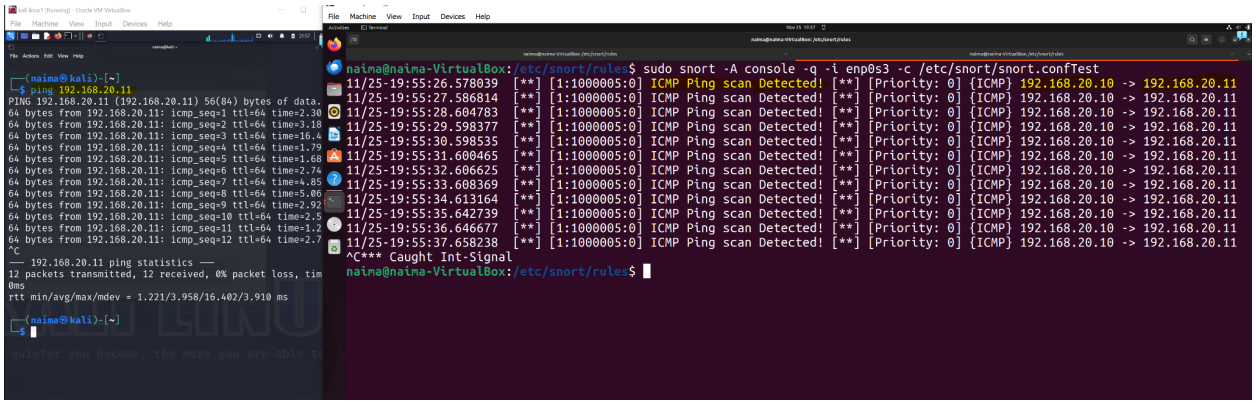
- This alerts me if the IP 192.168.20.10 sends an ICMP(Internet COntrol Message protocol) ping to the IP address 192.168.20.11.This helps detect such ping activities, logging them as “ICMP PIng scan Detected!” with the unique ID of 1000005.

- Ping Scan:
  - ping 192.168.20.11

Explanation:

- This is basically saying check if the ip address of 192.168.20.11 is live or reachable





b. **Rejecting ICMP RULE:**

Rule:

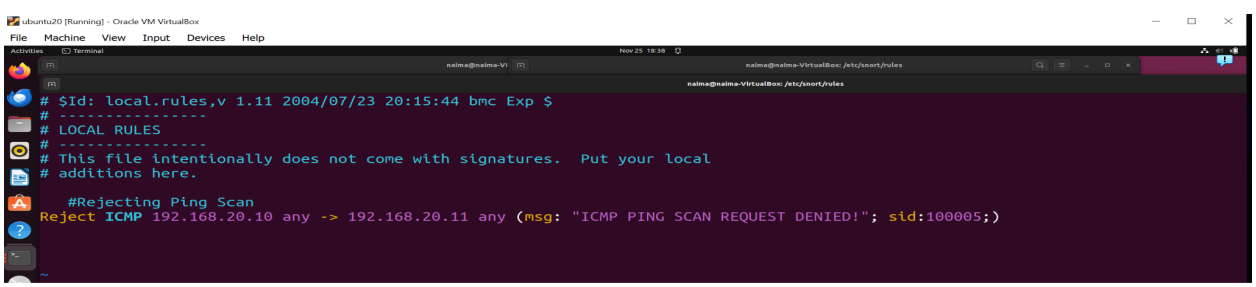
- Reject ICMP 192.168.20.10 any -> 192.168.20.11 any (msg: “ICMP PING SCAN REQUEST DENIED!”; sid:1000005;)

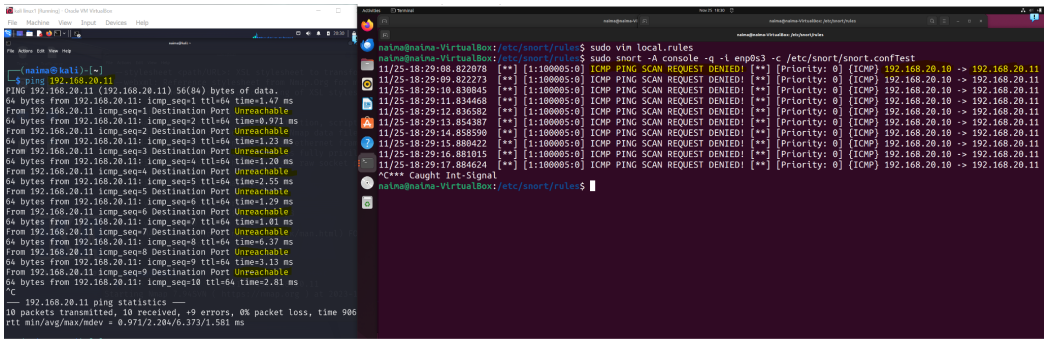
Explanation of the rule:

- This rule tells the network to reject or block any attempt by the device at IP 192.168.20.10 to ping or check the device at 192.168.20.11. I set up this rule to deny or stop these kinds of ping requests, and I should get a message saying “ICMP PING SCAN REQUEST DENIED!” with the ID of 1000005

Ping Scan:

- ping 192.168.20.11





6. Alert UDP

- Rule:

- alert udp any any -> $HOME_NET 53 (content: “Nmap”; msg: “Nmap UDP port 53 Detected!”; sid:100006

Explanation:

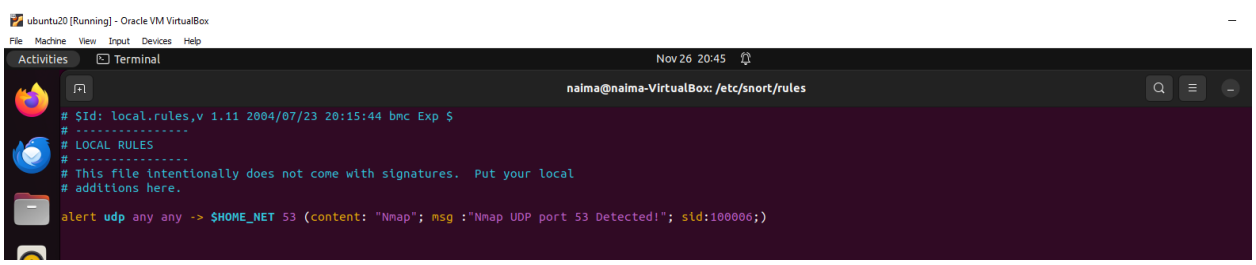
- This Snort rule is set to alert whenever it sees UDP traffic containing the specific text “Nmap” traveling to a designated destination on port 53

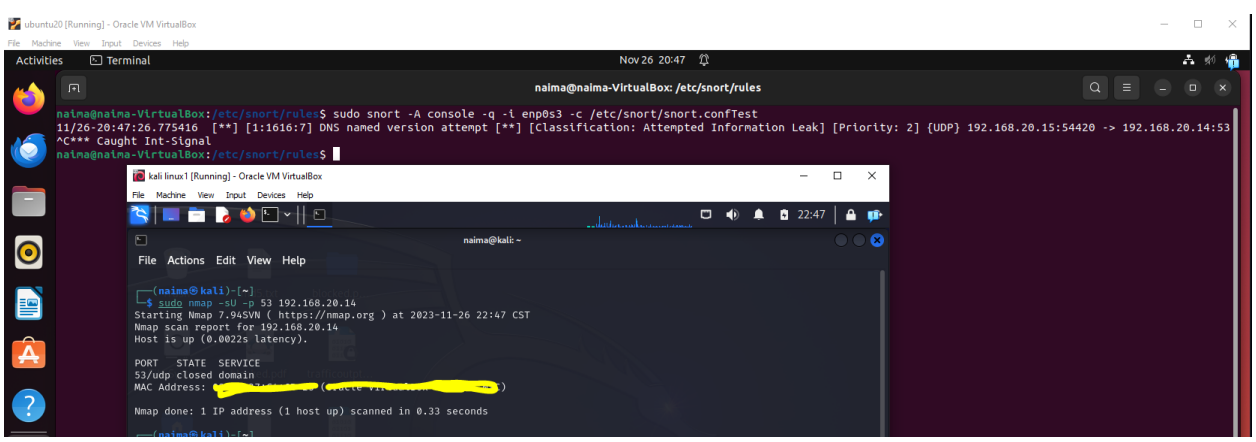
Nmap Scan:

- sudo nmap -sU 192.168.20.14

Explanation:

- It checks for open UDP ports on the Ubuntu machine at IP address 192.168.20.14 using a UDP scan





Summary:

- This project has provided me with valuable insights into identifying unauthorized network access using Snort and Nmap. By making rules and running checks, I'm learning to detect sneaky attempts to break in. It's hands-on learning, and I'm excited about expanding my knowledge in Cybersecurity.