

Naima Mumin

<https://www.linkedin.com/in/naima-mumin-a48750270/>

## Uncovering Network Vulnerabilities Using Snort and Nmap

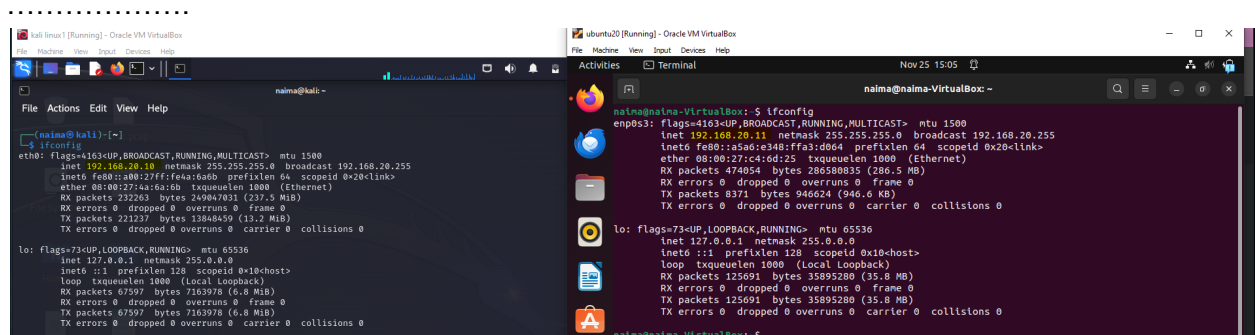
### Objective:

- 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



```
(naima@kali)~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.20.10  netmask 255.255.255.0  broadcast 192.168.20.255
    inet6 fe80::a00:27ff:feaa:bad0  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:aa:da:50  txqueuelen 1000  (Ethernet)
    RX packets 23263  bytes 249047031 (237.5 MiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 22127  bytes 13804459 (13.2 MiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 67597  bytes 7163978 (6.8 MiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 67597  bytes 7163978 (6.8 MiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

naima@naima-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.20.11  netmask 255.255.255.0  broadcast 192.168.20.255
    inet6 fe80::a5a6:e348:ffa3:d064  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:c4:6d:25  txqueuelen 1000  (Ethernet)
    RX packets 474054  bytes 286580835 (286.5 MB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 8371  bytes 946624 (946.0 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

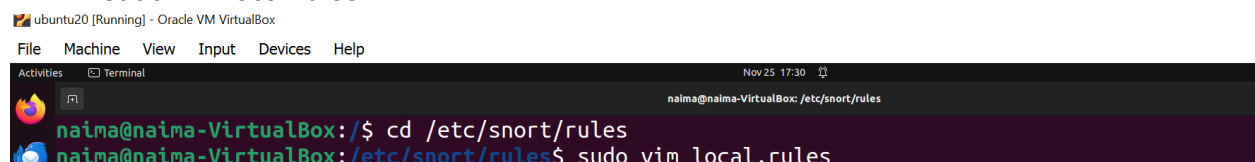
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 125691  bytes 35895288 (35.8 MB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 125691  bytes 35895288 (35.8 MB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

naima@naima-VirtualBox:~$
```

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

### Commands:

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



```
naima@naima-VirtualBox:/$ cd /etc/snort/rules
naima@naima-VirtualBox:/etc/snort/rules$ sudo vim local.rules
```

#### 3. TCP SYN Rule:

- Rule:

- Explanation of this rule:

- Nmap scan:

- Explanation:

- 
- The top screenshot shows a terminal window in a Kali Linux VM. The user is editing the file `/etc/snort/rules`. The content of the file includes a comment about the file not having signatures and a rule named `alert tcp` that triggers an alert when a SYN packet is received from 192.168.20.14. The rule is configured with `flags: s; content: "Nmap"; msg: "Nmap TCP SYN traffic Detected!!!"; sid:100001;`.
- The bottom screenshot shows the same terminal window after running `sudo snort -A console -q -i enp0s3 -c /etc/snort/snort.conf`. The output shows two alerts: one for a SNMP AgentX/tcp request and another for a SNMP request tcp, both classified as "Attempted Information Leak". Below the alerts, the user runs `nc** Caught Int-Signal` and then `nc** Caught Int-Signal` again. A separate terminal window in the background shows the output of a manual `nmap` scan on 192.168.20.14, which identifies the host as up and lists open ports: 22/tcp (ssh), 8080/tcp (http-alt), 8088/tcp (radan-http), and 8089/tcp (unknown).

## Detecting TCP Port 443

- ```
- 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

The first screenshot shows the editing of the Snort rules file in a terminal window. The user has added a new rule to the local.rules file:

```
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
#
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.
#
alert tcp any any -> 192.168.20.11 443 (msg: "TCP port 443 Detected!"; sid:100002;)
```

The second screenshot shows the output of an Nmap scan performed on 192.168.20.11. The scan is successful, showing that port 443/tcp is closed.

```
(naina@kali) ~$ nmap -sT -p 443 192.168.20.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-11-25 19:46 CST
Nmap scan report for 192.168.20.11
Host is up (0.0030s latency).
PORT      STATE SERVICE
443/tcp   closed https
Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
(naina@kali) ~$
```

The third screenshot shows the execution of the Snort rule. The user runs 'sudo snort -A console -q -i enp0s3 -c /etc/snort/snort.conf -t' to test the rule. The output shows that the rule is triggered, displaying a message: 'TCP port 443 Detected!'.

```
naina@naina-VirtualBox: /etc/snort/rules$ sudo snort -A console -q -i enp0s3 -c /etc/snort/snort.conf -t
11/25-17:46:45.504731  [**] [1:100002:0] TCP port 443 Detected! [**] [Priority: 0] (TCP) 192.168.20.10:59682 -> 192.168.20.11:443
11/25-17:46:45.513581  [**] [1:100002:0] TCP port 443 Detected! [**] [Priority: 0] (TCP) 192.168.20.10:59686 -> 192.168.20.11:443
^C*** Caught Int-Signal
naina@naina-VirtualBox: /etc/snort/rules$
```

## 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

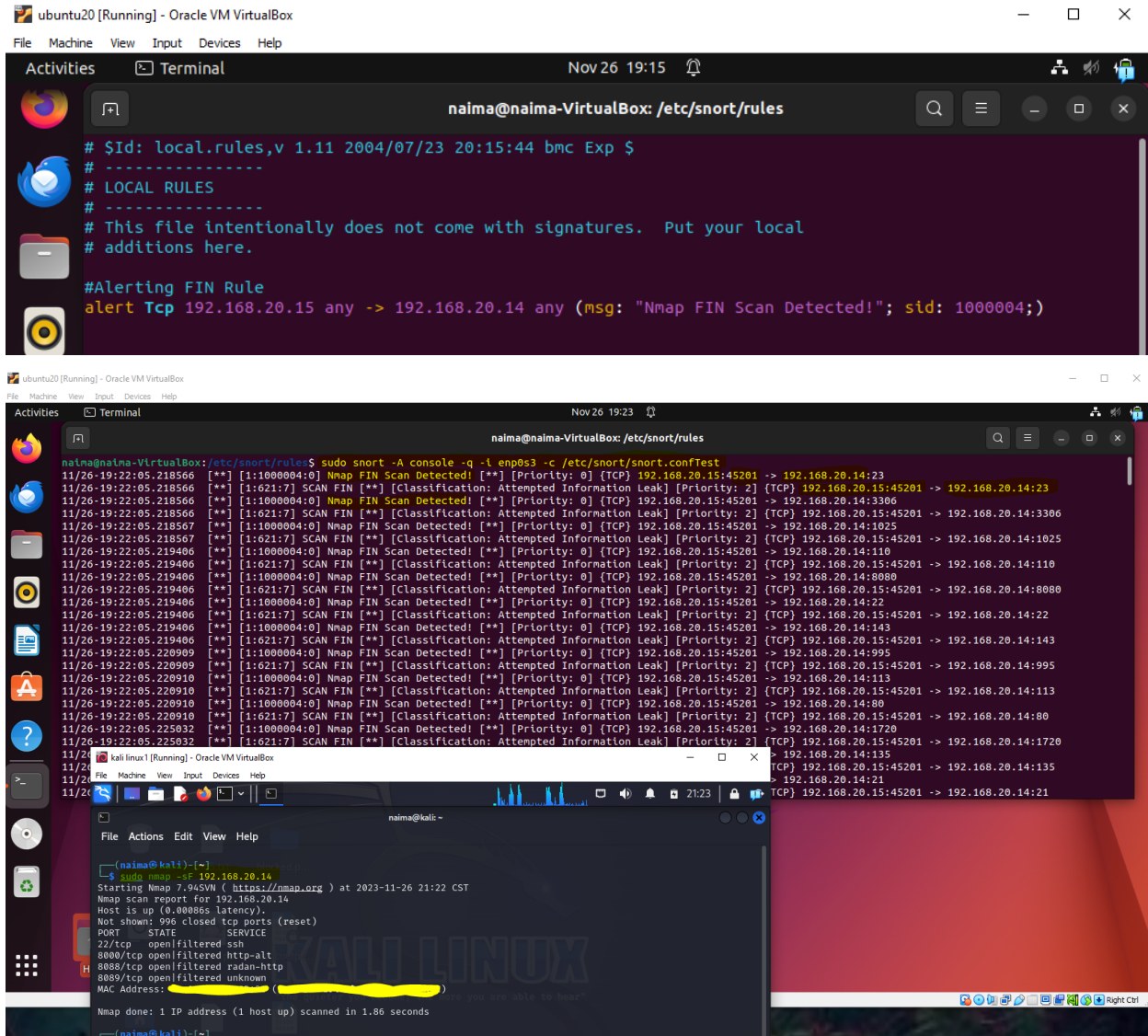
Command:

- `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`





## 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

```
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
#
# LOCAL RULES
#
# This file intentionally does not come with signatures. Put your local
# additions here.
#Alerting ICMP Rule
alert ICMP 192.168.20.10 any -> 192.168.20.11 any (msg: "ICMP Ping scan Detected!"; sid: 1000005;)

naina@naina-VirtualBox:/etc/snort/rules$ sudo snort -A console -q -i enp0s3 -c /etc/snort/snort.confTest
11/25-19:55:26.578039 [**] [1:1000005:0] ICMP Ping scan Detected! [**] [Priority: 0] [ICMP] 192.168.20.10 -> 192.168.20.11
11/25-19:55:27.586814 [**] [1:1000005:0] ICMP Ping scan Detected! [**] [Priority: 0] [ICMP] 192.168.20.10 -> 192.168.20.11
11/25-19:55:28.604783 [**] [1:1000005:0] ICMP Ping scan Detected! [**] [Priority: 0] [ICMP] 192.168.20.10 -> 192.168.20.11
11/25-19:55:29.598377 [**] [1:1000005:0] ICMP Ping scan Detected! [**] [Priority: 0] [ICMP] 192.168.20.10 -> 192.168.20.11
11/25-19:55:30.598535 [**] [1:1000005:0] ICMP Ping scan Detected! [**] [Priority: 0] [ICMP] 192.168.20.10 -> 192.168.20.11
11/25-19:55:31.609465 [**] [1:1000005:0] ICMP Ping scan Detected! [**] [Priority: 0] [ICMP] 192.168.20.10 -> 192.168.20.11
11/25-19:55:32.606625 [**] [1:1000005:0] ICMP Ping scan Detected! [**] [Priority: 0] [ICMP] 192.168.20.10 -> 192.168.20.11
11/25-19:55:33.608369 [**] [1:1000005:0] ICMP Ping scan Detected! [**] [Priority: 0] [ICMP] 192.168.20.10 -> 192.168.20.11
11/25-19:55:34.613164 [**] [1:1000005:0] ICMP Ping scan Detected! [**] [Priority: 0] [ICMP] 192.168.20.10 -> 192.168.20.11
11/25-19:55:35.642739 [**] [1:1000005:0] ICMP Ping scan Detected! [**] [Priority: 0] [ICMP] 192.168.20.10 -> 192.168.20.11
11/25-19:55:36.646677 [**] [1:1000005:0] ICMP Ping scan Detected! [**] [Priority: 0] [ICMP] 192.168.20.10 -> 192.168.20.11
11/25-19:55:37.658238 [**] [1:1000005:0] ICMP Ping scan Detected! [**] [Priority: 0] [ICMP] 192.168.20.10 -> 192.168.20.11
^C*** Caught Int-Signal
naina@naina-VirtualBox:/etc/snort/rules$
```

```
(naina@kali)~$ ping 192.168.20.11
PING 192.168.20.11 (192.168.20.11) 56(84) bytes of data.
64 bytes from 192.168.20.11: icmp_seq=1 ttl=64 time=2.36 ms
64 bytes from 192.168.20.11: icmp_seq=2 ttl=64 time=3.18 ms
64 bytes from 192.168.20.11: icmp_seq=3 ttl=64 time=16.4 ms
64 bytes from 192.168.20.11: icmp_seq=4 ttl=64 time=1.75 ms
64 bytes from 192.168.20.11: icmp_seq=5 ttl=64 time=1.68 ms
64 bytes from 192.168.20.11: icmp_seq=6 ttl=64 time=2.74 ms
64 bytes from 192.168.20.11: icmp_seq=7 ttl=64 time=4.85 ms
64 bytes from 192.168.20.11: icmp_seq=8 ttl=64 time=5.06 ms
64 bytes from 192.168.20.11: icmp_seq=9 ttl=64 time=2.92 ms
64 bytes from 192.168.20.11: icmp_seq=10 ttl=64 time=2.5 ms
64 bytes from 192.168.20.11: icmp_seq=11 ttl=64 time=1.2 ms
64 bytes from 192.168.20.11: icmp_seq=12 ttl=64 time=2.7 ms
^C
--- 192.168.20.11 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time
0ms
rtt min/avg/max/mdev = 1.221/3.958/16.402/3.910 ms
(naina@kali)~$
```

```
#Rejecting Ping Scan
Reject ICMP 192.168.20.10 any -> 192.168.20.11 any (msg: "ICMP PING SCAN REQUEST DENIED!"; sid:1000005;)
```

## 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

```
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
#
# LOCAL RULES
#
# This file intentionally does not come with signatures. Put your local
# additions here.
#Rejecting Ping Scan
Reject ICMP 192.168.20.10 any -> 192.168.20.11 any (msg: "ICMP PING SCAN REQUEST DENIED!"; sid:1000005;)
```



```

naima@kali:~$ ping 192.168.20.11
PING 192.168.20.11 (192.168.20.11) 56(84) bytes of data:
64 bytes from 192.168.20.11: icmp_seq=1 ttl=64 time=1.47 ms
From 192.168.20.11: icmp_seq=1 Destination Port Unreachable
64 bytes from 192.168.20.11: icmp_seq=2 ttl=64 time=0.971 ms
From 192.168.20.11: icmp_seq=2 Destination Port Unreachable
64 bytes from 192.168.20.11: icmp_seq=3 ttl=64 time=1.23 ms
From 192.168.20.11: icmp_seq=3 Destination Port Unreachable
64 bytes from 192.168.20.11: icmp_seq=4 ttl=64 time=1.20 ms
From 192.168.20.11: icmp_seq=4 Destination Port Unreachable
64 bytes from 192.168.20.11: icmp_seq=5 ttl=64 time=2.55 ms
From 192.168.20.11: icmp_seq=5 Destination Port Unreachable
64 bytes from 192.168.20.11: icmp_seq=6 ttl=64 time=1.29 ms
From 192.168.20.11: icmp_seq=6 Destination Port Unreachable
64 bytes from 192.168.20.11: icmp_seq=7 ttl=64 time=1.01 ms
From 192.168.20.11: icmp_seq=7 Destination Port Unreachable
64 bytes from 192.168.20.11: icmp_seq=8 ttl=64 time=0.37 ms
From 192.168.20.11: icmp_seq=8 Destination Port Unreachable
64 bytes from 192.168.20.11: icmp_seq=9 ttl=64 time=3.13 ms
From 192.168.20.11: icmp_seq=9 Destination Port Unreachable
64 bytes from 192.168.20.11: icmp_seq=10 ttl=64 time=2.81 ms
From 192.168.20.11: icmp_seq=10 Destination Port Unreachable
--- 192.168.20.11 ping statistics ---
10 packets transmitted, 10 received, +9 errors, 0% packet loss, time 906
rtt min/avg/max/mdev = 0.971/2.204/6.373/1.581 ms

naima@naima-VirtualBox:/etc/snort/rules$ sudo vim local.rules
naima@naima-VirtualBox:/etc/snort/rules$ sudo snort -A console -q -l enp0s3 -c /etc/snort/snort.confTest
11/25-18:29:08.822078 ** [1:100005:0] ICMP PING SCAN REQUEST DENIED! ** [Priority: 0] (ICMP) 192.168.20.10 -> 192.168.20.11
11/25-18:29:09.822273 ** [1:100005:0] ICMP PING SCAN REQUEST DENIED! ** [Priority: 0] (ICMP) 192.168.20.10 -> 192.168.20.11
11/25-18:29:10.830845 ** [1:100005:0] ICMP PING SCAN REQUEST DENIED! ** [Priority: 0] (ICMP) 192.168.20.10 -> 192.168.20.11
11/25-18:29:11.834468 ** [1:100005:0] ICMP PING SCAN REQUEST DENIED! ** [Priority: 0] (ICMP) 192.168.20.10 -> 192.168.20.11
11/25-18:29:12.836582 ** [1:100005:0] ICMP PING SCAN REQUEST DENIED! ** [Priority: 0] (ICMP) 192.168.20.10 -> 192.168.20.11
11/25-18:29:13.854387 ** [1:100005:0] ICMP PING SCAN REQUEST DENIED! ** [Priority: 0] (ICMP) 192.168.20.10 -> 192.168.20.11
11/25-18:29:14.858590 ** [1:100005:0] ICMP PING SCAN REQUEST DENIED! ** [Priority: 0] (ICMP) 192.168.20.10 -> 192.168.20.11
11/25-18:29:15.868422 ** [1:100005:0] ICMP PING SCAN REQUEST DENIED! ** [Priority: 0] (ICMP) 192.168.20.10 -> 192.168.20.11
11/25-18:29:16.881015 ** [1:100005:0] ICMP PING SCAN REQUEST DENIED! ** [Priority: 0] (ICMP) 192.168.20.10 -> 192.168.20.11
11/25-18:29:17.884624 ** [1:100005:0] ICMP PING SCAN REQUEST DENIED! ** [Priority: 0] (ICMP) 192.168.20.10 -> 192.168.20.11
^C*** Caught Int-Signal
naima@naima-VirtualBox:/etc/snort/rules$
```

## 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

```

naima@naima-VirtualBox:/etc/snort/rules$ sudo vim local.rules
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
#
# LOCAL RULES
#
# This file intentionally does not come with signatures.  Put your local
# additions here.
alert udp any any -> $HOME_NET 53 (content: "Nmap"; msg: "Nmap UDP port 53 Detected!"; sid:100006;)
```

```

naima@naima-VirtualBox:/etc/snort/rules$ sudo snort -A console -q -l enp0s3 -c /etc/snort/snort.confTest
11/26-20:47:26.775416 ** [1:1616:7] DNS named version attempt ** [Classification: Attempted Information Leak] [Priority: 2] (UDP) 192.168.20.15:54420 -> 192.168.20.14:53
^C*** Caught Int-Signal
naima@naima-VirtualBox:/etc/snort/rules$

kali linux1 [Running] - Oracle VM VirtualBox
naima@kali:~$ sudo nmap -sU -p 53 192.168.20.14
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-11-26 22:47 CST
Nmap scan report for 192.168.20.14
Host is up (0.0022s latency).

PORT      STATE SERVICE
53/udp    closed domain
MAC Address: 08:00:27:00:00:00 (VirtualBox__VirtIO__Ethernet__Card)

Nmap done: 1 IP address (1 host up) scanned in 0.33 seconds
naima@kali:~$
```

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.