

How to Detect NMAP Scan Using Snort

December 22, 2017 By Raj Chandel

Today we are going to discuss how to Detect NMAP scan using Snort but before moving ahead kindly read our previous articles related to Snort Installation (**Manually** or using **apt-respiratory**) and its **rule configuration** to enable it as IDS for your network.

Basically, in this article, we are testing Snort against NMAP various scan which will help network security analyst to setup snort rule in such a way so that they become aware of any kind of NMAP scanning.

Requirement

Attacker: Kali Linux (NMAP Scan)

Target: Ubuntu (Snort as IDS)

Optional: Wireshark (we have added it in our tutorial so that we can clearly confirm all incoming and outgoing packets of a network)

Let's Begins!!

Identify NMAP Ping Scan

As we know any attacker will start the attack by identifying host status by sending ICMP packet using ping scan. Therefore be smart and add a rule in snort which will analyst NMAP Ping scan when someone tries to scan your network for identifying a live host of a network.

Execute given below command in ubuntu's terminal to open snort local rule file in text editor.

```
sudo gedit /etc/snort/rules/local.rules
```

Now add given below line which will capture the incoming traffic coming on 192.168.1.105(ubuntu IP) network for ICMP protocol.

```
alert icmp any any -> 192.168.1.105 any (msg: "NMAP ping sweep Scan"; dsize:0;sid:
```

Turn on IDS mode of snort by executing given below command in terminal:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```

Now using attacking machine execute given below command to identify the status of the target machine i.e. host is UP or Down.

```
nmap -sP 192.168.1.105 --disable-arp-ping
```

If you will execute above command without parameter “disable arp-ping” then will work as default ping sweep scan which will send arp packets in spite of sending ICMP on targets network and maybe snort not able to capture NMAP Ping scan in that scenario, therefore we had use parameter “disable arp-ping” in the above command.

```
root@kali:~# nmap -sP 192.168.1.105 --disable-arp-ping
Starting Nmap 7.60 ( https://nmap.org ) at 2017-12-20 11:38 EST
Nmap scan report for 192.168.1.105
Host is up (0.0061s latency).
MAC Address: 00:0C:29:6B:71:A7 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

As I had declaimed above why we are involving Wireshark in this tutorial so that you can clearly see the packet sends form attacker network to targets network. Hence in given below image, you can notice ICMP request packet along with ICMP reply packets. These both are parts of network traffic.

ip.addr == 192.168.1.105							Expression...	+
No.	Time	Source	Destination	Protoc	Length	Info		
→ 516	3.7366...	192.168.1.104	192.168.1.105	ICMP	42	Echo (ping) request id=0xa		
517	3.7367...	192.168.1.104	192.168.1.105	TCP	58	37516 → 443 [SYN] Seq=0 Win		
518	3.7367...	192.168.1.104	192.168.1.105	TCP	54	37516 → 80 [ACK] Seq=1 Ack=		
519	3.7367...	192.168.1.104	192.168.1.105	ICMP	54	Timestamp request id=0xa		
— 520	3.7369...	192.168.1.105	192.168.1.104	ICMP	60	Echo (ping) reply id=0xa		
521	3.7369...	192.168.1.105	192.168.1.104	TCP	60	443 → 37516 [SYN, ACK] Seq=		
522	3.7369...	192.168.1.104	192.168.1.105	TCP	54	37516 → 443 [RST] Seq=1 Win		
523	3.7370...	192.168.1.105	192.168.1.104	TCP	60	80 → 37516 [RST] Seq=1 Win=		
524	3.7370...	192.168.1.105	192.168.1.104	ICMP	60	Timestamp reply id=0xa		

Come back to over your target machine where snort is capturing all in-coming traffic. Here, you will observe that it is generating an alert for NMAP Ping Sweep scan. Hence, you can block the attacker’s IP to protect your network from further scanning.

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```

```
pentest@ubuntu:~$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort
.conf -i eth0
12/22-01:52:18.051911  [**] [1:10000004:1] NMAP ping sweep Scan [**] [Priority:
0] {ICMP} 192.168.1.104 -> 192.168.1.105
```

Identify NMAP TCP Scan

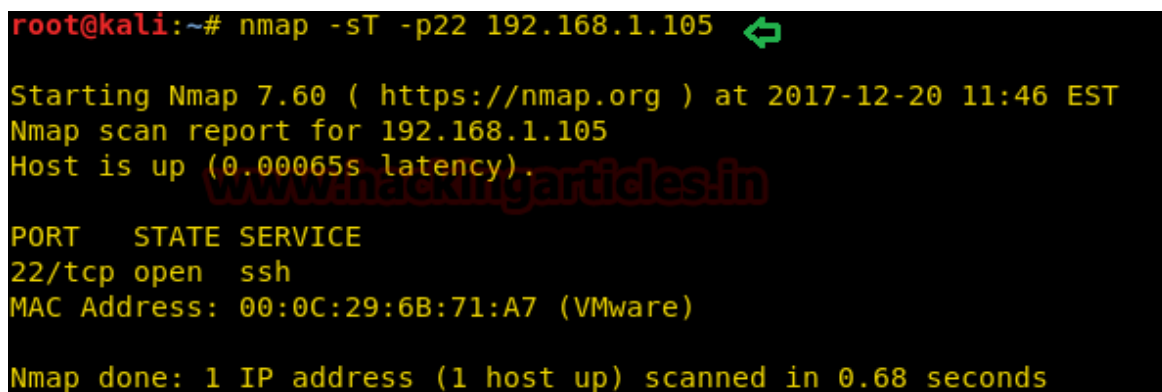
Now in order to connect with the target network, an attacker may go for networking enumeration either using TCP Protocol or UDP protocol. Let's assume attacker may choose TCP scanning for network enumeration then in that situation we can apply the following rule in snort local rule file.

```
alert tcp any any -> 192.168.1.105 22 (msg: "NMAP TCP Scan";sid:10000005; rev:2; )
```

Above rule is only applicable for port 22 so if you want to scan any other port then replace 22 from the port you want to scan or else you can also use "any" to analysis all ports. Enable the NIDS mode of snort as done above.

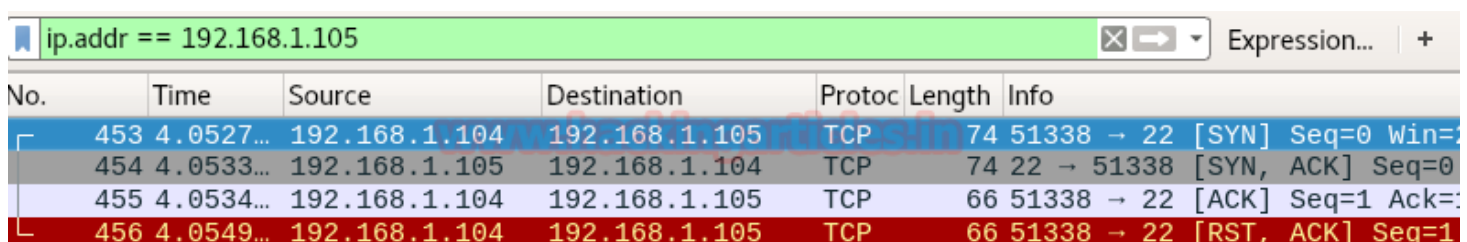
Now again using the attacker machine execute the given below command for TCP scan on port 22.

```
nmap -sT -p22 192.168.1.105
```



```
root@kali:~# nmap -sT -p22 192.168.1.105
Starting Nmap 7.60 ( https://nmap.org ) at 2017-12-20 11:46 EST
Nmap scan report for 192.168.1.105
Host is up (0.00065s latency).
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 00:0C:29:6B:71:A7 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.68 seconds
```

From the image given below, you can observe Wireshark has captured TCP packets from 192.168.1.104 to 192.168.1.105



ip.addr == 192.168.1.105						
No.	Time	Source	Destination	Protoc	Length	Info
453	4.0527...	192.168.1.104	192.168.1.105	TCP	74	51338 → 22 [SYN] Seq=0 Win=0
454	4.0533...	192.168.1.105	192.168.1.104	TCP	74	22 → 51338 [SYN, ACK] Seq=0
455	4.0534...	192.168.1.104	192.168.1.105	TCP	66	51338 → 22 [ACK] Seq=1 Ack=:
456	4.0549...	192.168.1.104	192.168.1.105	TCP	66	51338 → 22 [RST, ACK] Seq=1

Here you can confirm that our snort is absolutely working when the attacker is scanning port 22 using nmap TCP scan and it is showing attacker's IP from where traffic is coming on port 22. Hence you can block this IP to protect your network from further scanning.

```
pentest@ubuntu:~$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
12/20-08:46:53.932278 192.168.1.104:51338 -> 192.168.1.105:22 NMAP TCP Scan [**] [Priority: 0] {TCP}
12/20-08:46:53.932991 192.168.1.104:51338 -> 192.168.1.105:22 NMAP TCP Scan [**] [Priority: 0] {TCP}
12/20-08:46:53.934417 192.168.1.104:51338 -> 192.168.1.105:22 NMAP TCP Scan [**] [Priority: 0] {TCP}
```

Identify NMAP XMAS Scan

As we know that TCP communication follows three-way handshake to established TCP connection with target machine but sometimes instead of using SYN, SYN/ACK, ACK flag attacker choose XMAS scan to connect with the target by sending data packets through Fin, PSH & URG flags.

Let assume attacker may choose XMAS scanning for network enumeration then in that situation we can apply the following rule in snort local rule file.

```
alert tcp any any -> 192.168.1.105 22 (msg:"Nmap XMAS Tree Scan"; flags:FPU; sid:1
```

Again above rule is only applicable for port 22 which will listen for incoming traffic when packets come from Fin, PSH & URG flags. So if you want to scan any other port then replace 22 from the port you want to scan else you can also use “any” to analysis all ports. Enable the NIDS mode of snort as done above.

Now again using the attacker machine execute the given below command for XMAS scan on port 22.

```
nmap -sX -p22 192.168.1.105
```

```
root@kali:~# nmap -sX -p22 192.168.1.105
Starting Nmap 7.60 ( https://nmap.org ) at 2017-12-20 11:54 EST
Nmap scan report for 192.168.1.105
Host is up (0.00047s latency).

PORT      STATE      SERVICE
22/tcp    open|filtered ssh
MAC Address: 00:0C:29:6B:71:A7 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.43 seconds
```

From given below image you can observe that Wireshark is showing 2 packets from attacker machine to target machine has been sent using FIN, PSH, URG flags.

ip.addr == 192.168.1.105							Expression...	+
No.	Time	Source	Destination	Protoc	Length	Info		
590	8.3603...	192.168.1.104	192.168.1.105	TCP	54	38157 → 22 [FIN, PSH, URG]		
592	8.4619...	192.168.1.104	192.168.1.105	TCP	54	38158 → 22 [FIN, PSH, URG]		

Come back to over your target machine where snort is capturing all incoming traffic here you will observe that it is generating an alert for NMAP XMAP scan. Hence you can block the attacker's IP to protect your network from further scanning.

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```

```
pentest@ubuntu:~$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
12/20-08:54:09.407169  [**] [1:1000006:1] Nmap XMAS Tree Scan [**] [Priority: 0]
{TCP} 192.168.1.104:38157 -> 192.168.1.105:22
12/20-08:54:09.508960  [**] [1:1000006:1] Nmap XMAS Tree Scan [**] [Priority: 0]
{TCP} 192.168.1.104:38158 -> 192.168.1.105:22
```

Identify NMAP FIN Scan

Instead of using SYN, SYN/ACK and ACK flag to established TCP connection with the target machine may attacker choose FIN scan to connect with the target by sending data packets through Fin flags only.

Let assume attacker may choose FIN scanning for network enumeration then in that situation we can apply the following rule in snort local rule file.

```
alert tcp any any -> 192.168.1.105 22 (msg:"Nmap FIN Scan"; flags:F; sid:1000008;
```

Again above rule is only applicable for port 22 which will listen for incoming traffic when packets come from Fin Flags. So if you want to scan any other port then replace 22 from the port you want to scan else you can also use “any” to analysis all ports. Enable NIDS mode of snort as done above.

Now again using the attacker machine execute the given below command for FIN scan on port 22.

```
nmap -sF -p22 192.168.1.105
```

```

root@kali:~# nmap -sF -p22 192.168.1.105

Starting Nmap 7.60 ( https://nmap.org ) at 2017-12-20 12:00 EST
Nmap scan report for 192.168.1.105
Host is up (0.00018s latency).
PORT      STATE      SERVICE
22/tcp    open|filtered ssh
MAC Address: 00:0C:29:6B:71:A7 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.48 seconds

```

From given below image you can observe that Wireshark is showing 2 packets from attacker machine to target machine has been sending using FIN flags.

ip.addr == 192.168.1.105							Expression...	+
No.	Time	Source	Destination	Protoc	Length	Info		
723	8.6023...	192.168.1.104	192.168.1.105	TCP	54	39392 → 22 [FIN] Seq=1 Win=		
731	8.7031...	192.168.1.104	192.168.1.105	TCP	54	39393 → 22 [FIN] Seq=1 Win=		
5787	42.777...	192.168.1.105	224.0.0.251	IGMP	60	Membership Report group 224		

Come back to over your target machine where snort is capturing all incoming traffic here you will observe that it is generating an alert for NMAP FIN scan. Hence you can block the attacker's IP to protect your network from further scanning.

```

sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0

```

```

pentest@ubuntu:~$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
12/20-09:00:12.048515  /** [1:1000008:1] Nmap FIN Scan /** [Priority: 0] {TCP}
192.168.1.104:39392 -> 192.168.1.105:22
12/20-09:00:12.149434  /** [1:1000008:1] Nmap FIN Scan /** [Priority: 0] {TCP}
192.168.1.104:39393 -> 192.168.1.105:22

```

Identify NMAP NULL Scan

Instead of using SYN, SYN/ACK and ACK flag to established TCP connection with the target machine may attacker choose NULL scan to connect with the target by sending data packets through NONE flags only.

Let assume attacker may choose NULL scanning for network enumeration then in that situation we can apply the following rule in snort local rule file.

```

alert tcp any any -> 192.168.1.105 22 (msg:"Nmap NULL Scan"; flags:0; sid:1000009;

```


Again above rule is only applicable for port 22 which will listen for incoming traffic when packets come from NONE Flags. So if you want to scan any other port then replace 22 from the port you want to scan else you can also use “any” to analysis all ports. Enable the NIDS mode of snort as done above.

Now again using the attacker machine execute the given below command for the NULL scan on port 22.

```
nmap -sN -p22 192.168.1.105
```

```
root@kali:~# nmap -sN -p22 192.168.1.105 ↩️
Starting Nmap 7.60 ( https://nmap.org ) at 2017-12-20 12:03 EST
Nmap scan report for 192.168.1.105
Host is up (0.00017s latency).
PORT      STATE      SERVICE
22/tcp    open|filtered ssh
MAC Address: 00:0C:29:6B:71:A7 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
```

From given below image you can observe that Wireshark is showing 2 packets from attacker machine to target machine has been sending using NONE flags.

ip.addr == 192.168.1.105							
No.	Time	Source	Destination	Protoc	Length	Info	
889	14.701...	192.168.1.104	192.168.1.105	TCP	54	47435 → 22	[<None>] Seq=1 W
896	14.801...	192.168.1.104	192.168.1.105	TCP	54	47436 → 22	[<None>] Seq=1 W

Come back to over your target machine where snort is capturing all incoming traffic here you will observe that it is generating an alert for NMAP Null scan. Hence you can block the attacker’s IP to protect your network from further scanning.

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```

```
pentest@ubuntu:~$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort
.conf -i eth0 ↩️
12/20-09:03:15.568255  [**] [1:1000009:1] Nmap NULL Scan [**] [Priority: 0] {TCP
} 192.168.1.104:47435 -> 192.168.1.105:22
12/20-09:03:15.668675  [**] [1:1000009:1] Nmap NULL Scan [**] [Priority: 0] {TCP
} 192.168.1.104:47436 -> 192.168.1.105:22
```

Identify NMAP UDP Scan

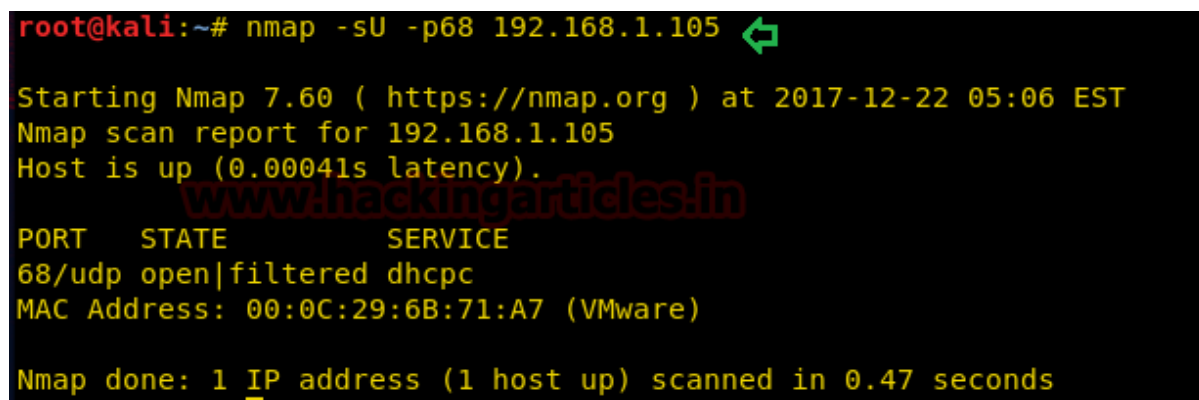
In order to Identify open UDP port and running services attacker may choose NMAP UDP scan to establish a connection with target machine for network enumeration then in that situation, we can apply the following rule in snort local rule file.

```
alert udp any any -> 192.168.1.105 any ( msg:"Nmap UDP Scan"; sid:1000010; rev:1;
```

Again above rule is applicable for every UDP port which will listen for incoming traffic when packets are coming over any UDP port, so if you want to capture traffic for any particular UDP port then replace “any” from that specific port number as done above. Enable the NIDS mode of snort as done above.

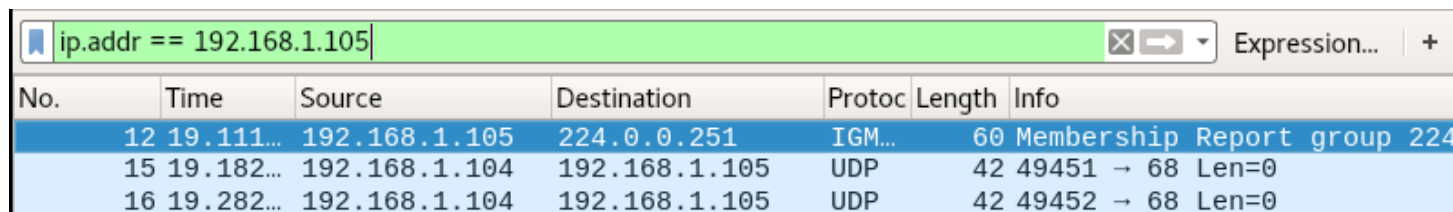
Now again using the attacker machine execute the given below command for a NULL scan on port 22.

```
nmap -sU -p68 192.168.1.105
```



```
root@kali:~# nmap -sU -p68 192.168.1.105
Starting Nmap 7.60 ( https://nmap.org ) at 2017-12-22 05:06 EST
Nmap scan report for 192.168.1.105
Host is up (0.00041s latency).
PORT      STATE      SERVICE
68/udp    open|filtered dhcpc
MAC Address: 00:0C:29:6B:71:A7 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
```

From given below image you can observe that Wireshark is showing 2 packets from attacker machine to target machine has been sending over UDP Port.



No.	Time	Source	Destination	Protoc	Length	Info
12	19.111...	192.168.1.105	224.0.0.251	IGMP...	60	Membership Report group 224
15	19.182...	192.168.1.104	192.168.1.105	UDP	42	49451 → 68 Len=0
16	19.282...	192.168.1.104	192.168.1.105	UDP	42	49452 → 68 Len=0

Come back to over your target machine where snort is capturing all incoming traffic here you will observe that it is generating an alert for NMAP UDP scan. Hence you can block the attacker’s IP to protect your network from further scanning.


```
pentest@ubuntu:~$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort
.conf -i eth0
12/22-02:06:19.669158  [**] [1:1000010:1] NMAP UDP scan [**] [Priority: 0] {UDP}
192.168.1.104:49451 -> 192.168.1.105:68
12/22-02:06:19.769495  [**] [1:1000010:1] NMAP UDP scan [**] [Priority: 0] {UDP}
192.168.1.104:49452 -> 192.168.1.105:68
```