Understanding Guide to Nmap Firewall Scan (Part 2)

December 2, 2017   By Raj Chandel

In our **previous** article we had demonstrated "Nmap firewall scan (part 1)" by making use of Iptable rules and then try to bypass firewall filter to perform NMAP Advance scanning, today we are going to discuss the second part of it.

Requirement

Attacker: Kali Linux

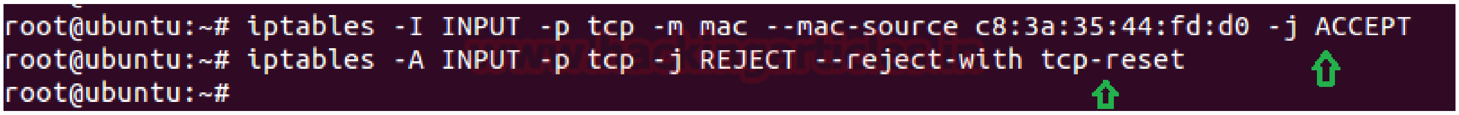
Target: Ubuntu

Spoof MAC Address Scan

Allow TCP Packet from Specific Mac Address

If network admin wants to establish TCP connect from specific MAC address and do not want to connect with another system then he could use following Iptable rules to apply firewall filter in his network.

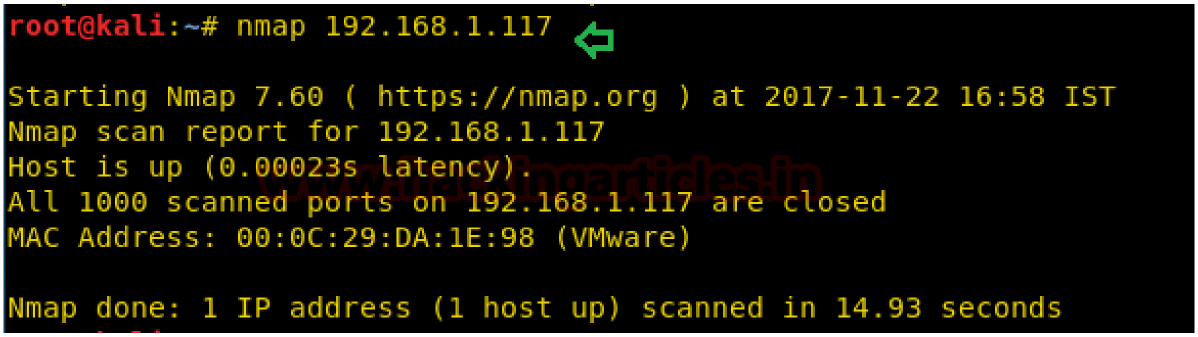
```
iptables -I INPUT -p tcp -m mac --mac-source "AA:AA:AA:AA:AA:AA" -j ACCEPT
iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset
```



```
root@ubuntu:~# iptables -I INPUT -p tcp -m mac --mac-source c8:3a:35:44:fd:d0 -j ACCEPT
root@ubuntu:~# iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset
root@ubuntu:~#
```

Now when the attacker will perform basic network scanning on the target's network, he could not able to enumerate ports and running service of a victim's system.

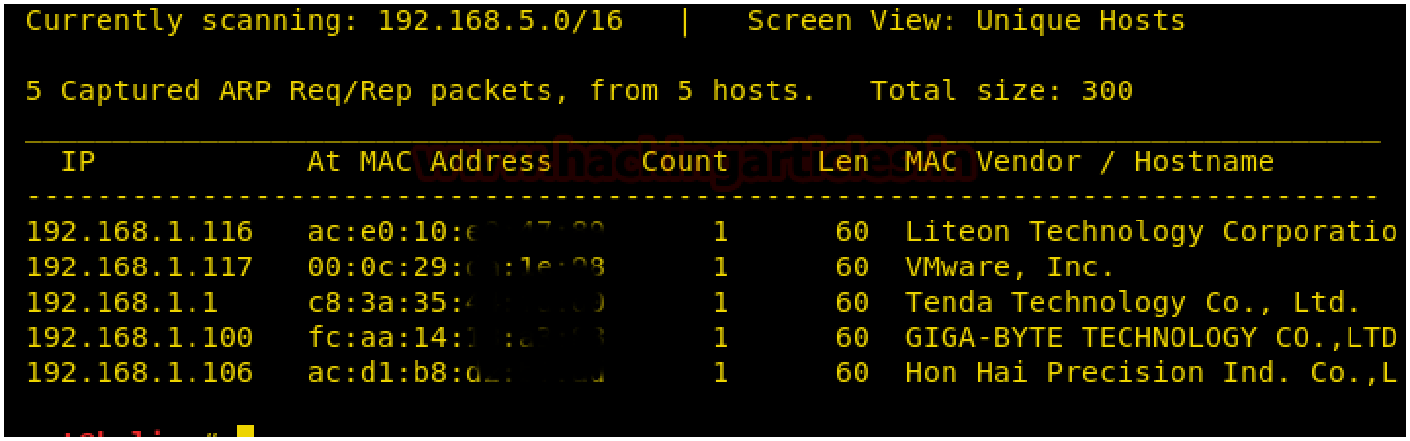
```
nmap 192.168.1.117
```



```
root@kali:~# nmap 192.168.1.117

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-22 16:58 IST
Nmap scan report for 192.168.1.117
Host is up (0.00023s latency).
All 1000 scanned ports on 192.168.1.117 are closed
MAC Address: 00:0C:29:DA:1E:98 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 14.93 seconds
```

In order to bypass above applied filter attacker may run **netdiscover command** or **nmap Host Scan** in Kali Linux terminal to identify the active host in the network. As a result he will get a table which contains MAC address and

IP address of the active host in the local network.



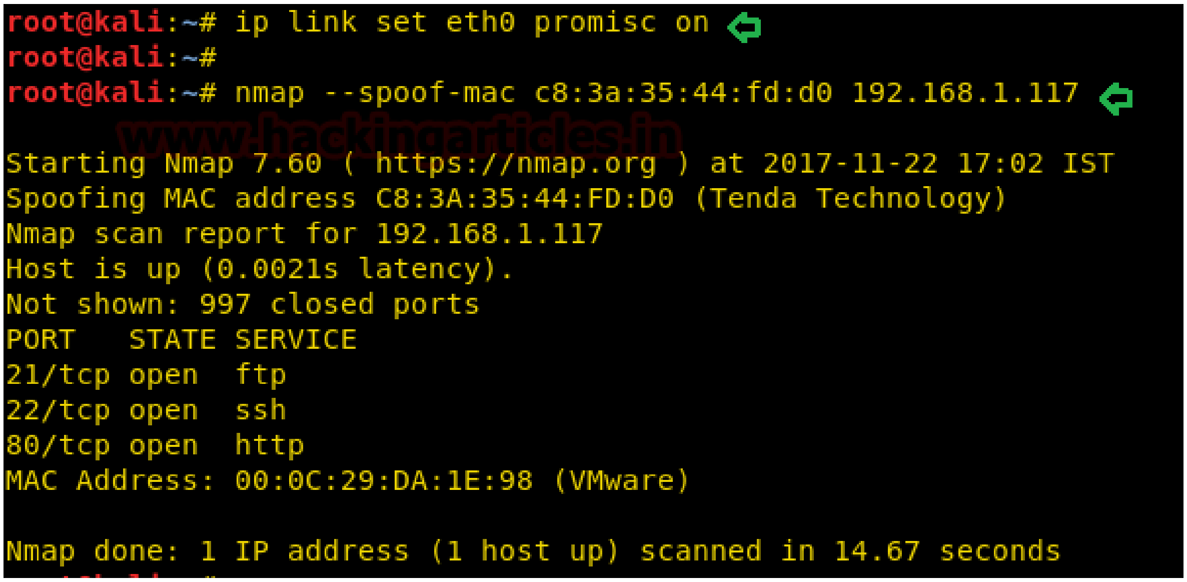
```
Currently scanning: 192.168.5.0/16   |   Screen View: Unique Hosts

5 Captured ARP Req/Rep packets, from 5 hosts.   Total size: 300
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.1.116   ac:e0:10:[redacted]      1       60  Liteon Technology Corporatio
192.168.1.117   00:0c:29:[redacted]      1       60  VMware, Inc.
192.168.1.1     c8:3a:35:[redacted]      1       60  Tenda Technology Co., Ltd.
192.168.1.100   fc:aa:14:[redacted]      1       60  GIGA-BYTE TECHNOLOGY CO.,LTD
192.168.1.106   ac:d1:b8:[redacted]      1       60  Hon Hai Precision Ind. Co.,L
```

Now either use one by one all MAC address in nmap command or save all MAC address in a text file and give its path in nmap command but to perform this attacker first need to enable “Promiscuous mode” of his network. Well, to do so type given below commands first for Promiscuous mode and second for nmap scanning.

```
ip link set eth0 promisc on
nmap --spoof-mac AA:AA:AA:AA:AA:AA 192.168.1.117
```

Hence if you are lucky to spoof correct Mac address then you can easily bypass the firewall filter and able to establish TCP connect with victim’s network for port enumeration.

Nice!!! If you will notice in given below image you will observe open ports of the target’s network.



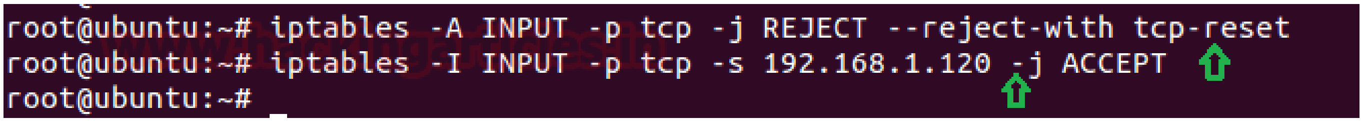
```
root@kali:~# ip link set eth0 promisc on
root@kali:~#
root@kali:~# nmap --spoof-mac c8:3a:35:44:fd:d0 192.168.1.117

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-22 17:02 IST
Spoofing MAC address C8:3A:35:44:FD:D0 (Tenda Technology)
Nmap scan report for 192.168.1.117
Host is up (0.0021s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:0C:29:DA:1E:98 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 14.67 seconds
```

Spoof IP Address

Allow TCP Packet from Specific IP

If network admin wants to establish TCP connect from specific IP and do not want to connect with another system then he could use following Iptable rules to apply firewall filter in his network.

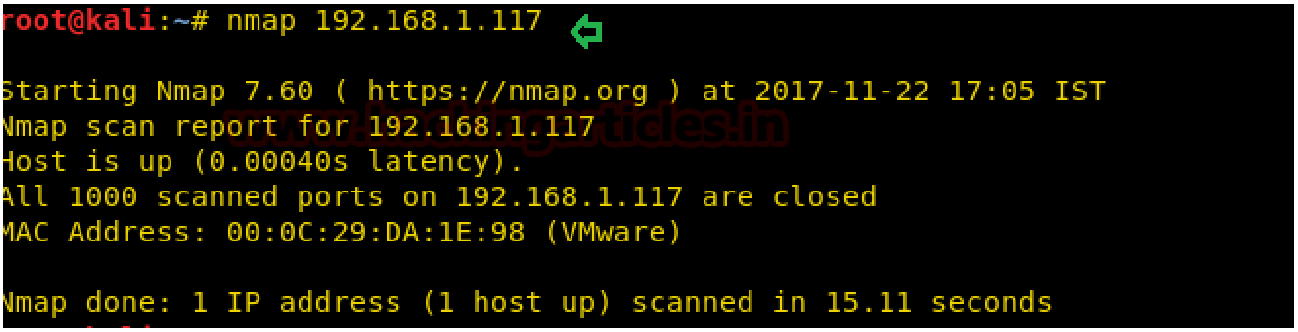
```
iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset
iptables -I INPUT -p tcp -s 192.168.1.120 -j ACCEPT
```



```
root@ubuntu:~# iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset
root@ubuntu:~# iptables -I INPUT -p tcp -s 192.168.1.120 -j ACCEPT
root@ubuntu:~# _
```

Now when again attacker will perform basic network scanning on the target's network, he could not able to enumerate ports and running service of victim's system.

```
nmap 192.168.1.117
```



```
root@kali:~# nmap 192.168.1.117

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-22 17:05 IST
Nmap scan report for 192.168.1.117
Host is up (0.00040s latency).
All 1000 scanned ports on 192.168.1.117 are closed
MAC Address: 00:0C:29:DA:1E:98 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 15.11 seconds
```

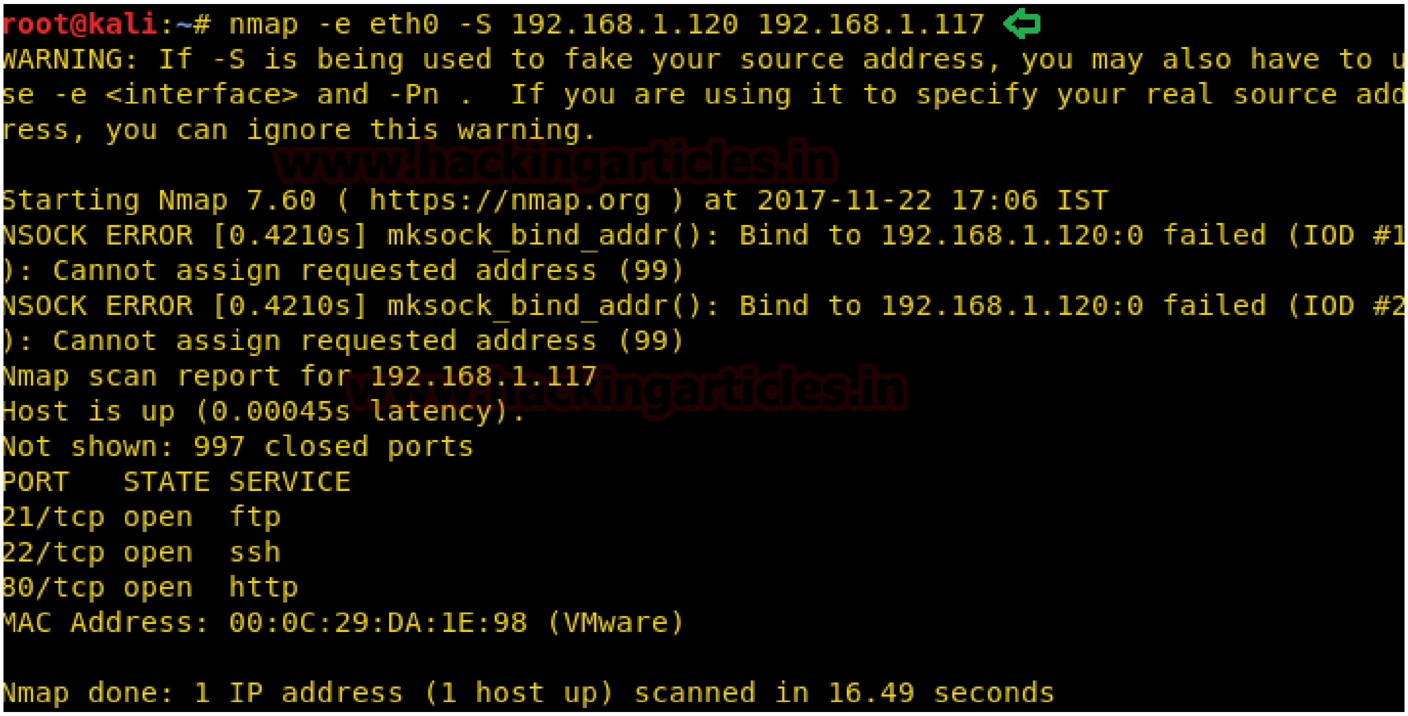
In order to bypass above applied filter attacker may again run **netdiscover command** or **nmap Host Scan** in Kali Linux terminal to identify the active host in the network. As a result he will get a table which contains MAC address and IP address of the active host in the local network.

Now either use one by one all IP address in nmap command or save all IP address in a text file and give its path in nmap command and then execute the following command:

```
nmap -e eth0 -S 192.168.1.120 192.168.1.117
```

Hence if you are lucky to spoof correct IP address then you can easily bypass the firewall filter and able to establish TCP connect with victim's network for port enumeration.

Great!! If you will notice in given below image you will observe open ports of target's network.



```
root@kali:~# nmap -e eth0 -S 192.168.1.120 192.168.1.117
WARNING: If -S is being used to fake your source address, you may also have to u
se -e <interface> and -Pn .  If you are using it to specify your real source add
ress, you can ignore this warning.

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-22 17:06 IST
NSOCK ERROR [0.4210s] mksock_bind_addr(): Bind to 192.168.1.120:0 failed (IOD #1
): Cannot assign requested address (99)
NSOCK ERROR [0.4210s] mksock_bind_addr(): Bind to 192.168.1.120:0 failed (IOD #2
): Cannot assign requested address (99)
Nmap scan report for 192.168.1.117
Host is up (0.00045s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:0C:29:DA:1E:98 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 16.49 seconds
```

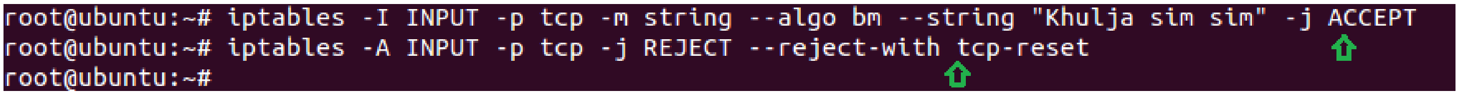
Data-String Scan

Allow TCP Packet from Specific String

If network admin wants to establish TCP connect from a system which contains a specific string and do not want to connect with other system does not contain that special string packets then he could use following Iptable rules to apply firewall filter in his network.

```
iptables -I INPUT -p tcp -m string --algo bm --string "Khulja sim sim" -j ACCEPT
iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset
```

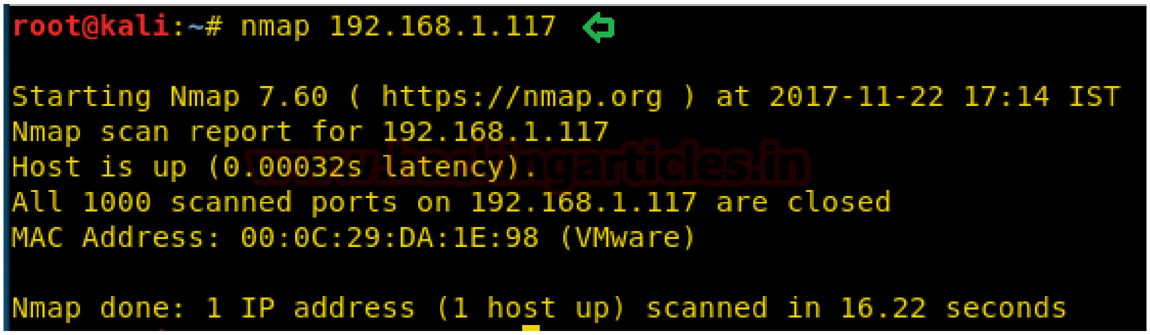
In above rule, you can see we had used **"Khulja sim sim"** as a special string to establish TCP connection. Hence only those TCP connection could be established which contain "Khulja sim sim" in packets.



```
root@ubuntu:~# iptables -I INPUT -p tcp -m string --algo bm --string "Khulja sim sim" -j ACCEPT
root@ubuntu:~# iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset
root@ubuntu:~#
```

Now when again attacker will perform basic network scanning on target's network, he could not able to enumerate ports and running service of victim's system because traffic generates from his network does not contain special string in packets thus firewall of target system will discard all TCP packet of attacker's network.

```
nmap 192.168.1.117
```



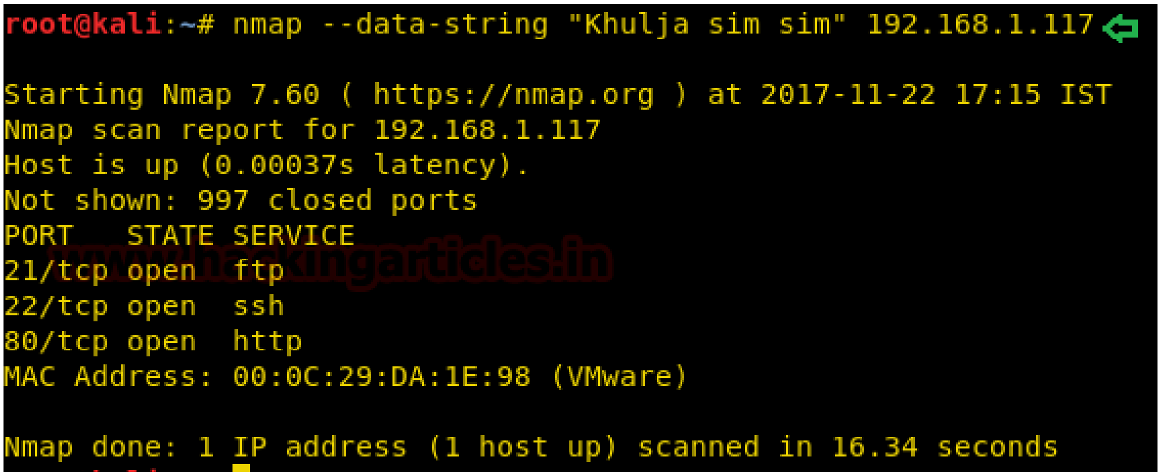
```
root@kali:~# nmap 192.168.1.117

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-22 17:14 IST
Nmap scan report for 192.168.1.117
Host is up (0.00032s latency).
All 1000 scanned ports on 192.168.1.117 are closed
MAC Address: 00:0C:29:DA:1E:98 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 16.22 seconds
```

If the attacker somehow sniffs special string "khulja sim sim" to connect with target's network then he could use –data-string argument in nmap command to bypass the firewall.

```
nmap --data-string "Khulja sim sim" 192.168.1.117
```

Hence if you are lucky to sniff correct data string then you can easily bypass the firewall filter and able to establish TCP connect with victim's network for port enumeration.

Wonderful!! If you will notice given below image you will observe open ports of target's network.



```
root@kali:~# nmap --data-string "Khulja sim sim" 192.168.1.117

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-22 17:15 IST
Nmap scan report for 192.168.1.117
Host is up (0.00037s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:0C:29:DA:1E:98 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 16.34 seconds
```

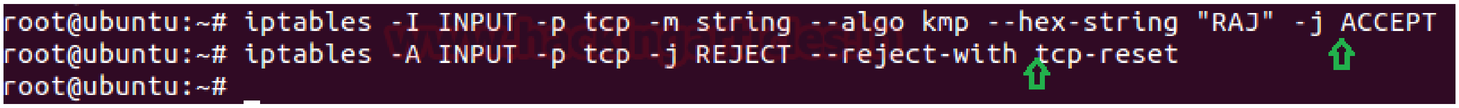
Hex String Scan

Allow TCP Packet from Specific Hex String

If network admin wants to establish TCP connect from a system which contains the hexadecimal value of the particular string and do not want to connect with other system does not contain the hexadecimal value of that special string in packets then he could use following Iptable rules to apply firewall filter in his network.

```
iptables -I INPUT -p tcp -m string --algo kmp --hex-string "RAJ" -j ACCEPT
iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset
```

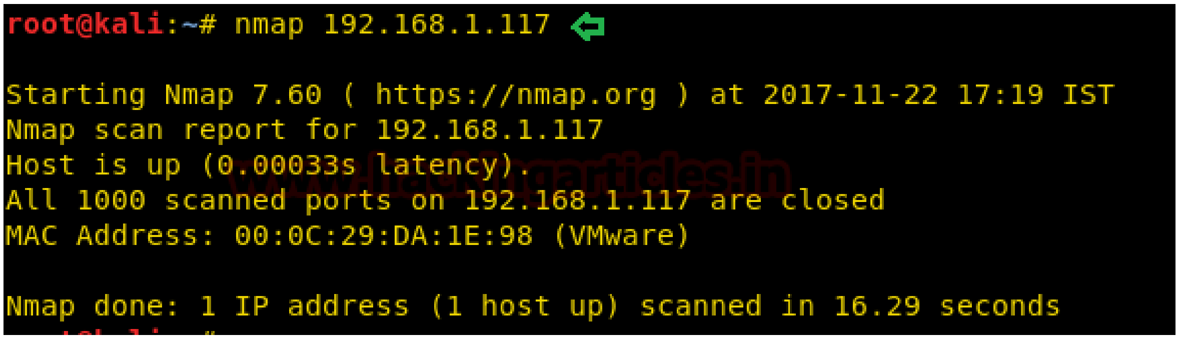
In the above rule, you can see we had used hex value for **"RAJ"** as a special string to establish a TCP connection. Hence only those TCP connection could be established which contain hex value of "RAJ" in the packet.



```
root@ubuntu:~# iptables -I INPUT -p tcp -m string --algo kmp --hex-string "RAJ" -j ACCEPT
root@ubuntu:~# iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset
root@ubuntu:~# _
```

Now when again attacker will perform basic network scanning on target's network, he could not able to enumerate ports and running service of victim's system because traffic generates from his network does not contain hex value of the special string in packets thus firewall of target system will discard all TCP packet of attacker's network.

```
nmap 192.168.1.117
```



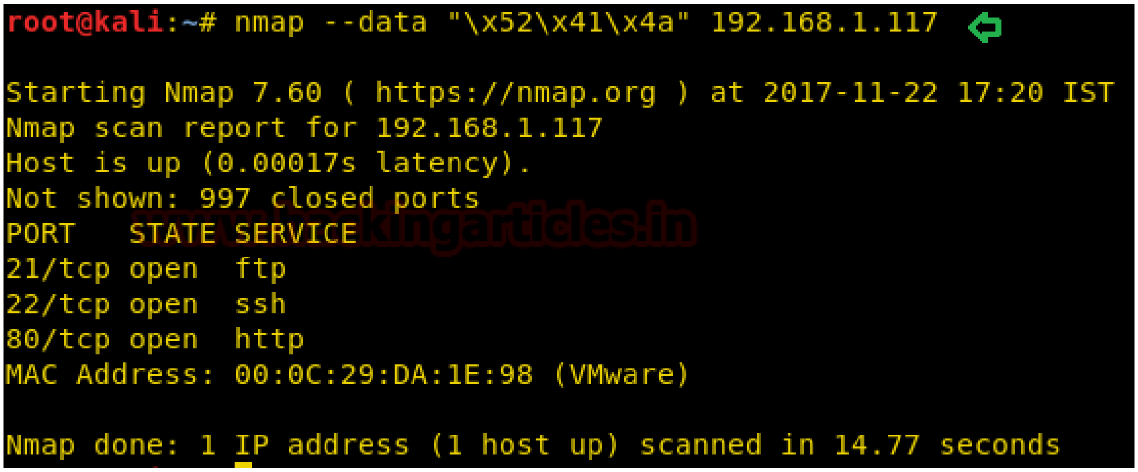
```
root@kali:~# nmap 192.168.1.117

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-22 17:19 IST
Nmap scan report for 192.168.1.117
Host is up (0.00033s latency).
All 1000 scanned ports on 192.168.1.117 are closed
MAC Address: 00:0C:29:DA:1E:98 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 16.29 seconds
```

If an attacker somehow sniffs special string "RAJ" to connect with target's network then he could use its hex values with –data argument in nmap command to bypass the firewall.

```
nmap --data "\x52\x41\x4a" 192.168.1.117
```

Hence if you are lucky to sniff correct hex value of particular data string then you can easily bypass the firewall filter and able to establish TCP connect with victim's network for port enumeration.

Hence, if you will notice given below image you will observe open ports of the target's network.



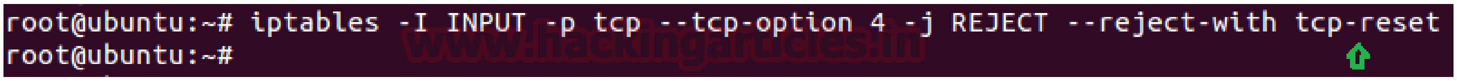
```
root@kali:~# nmap --data "\x52\x41\x4a" 192.168.1.117

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-22 17:20 IST
Nmap scan report for 192.168.1.117
Host is up (0.00017s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:0C:29:DA:1E:98 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 14.77 seconds
```

IP-Options Scan

Reject TCP Packets contains tcp-option

By default, nmap sends 24 bytes of TCP data in which 4 bytes of data is reserve for TCP Options if network admin rejects 4 bytes tcp –option packet to discord tcp connection to prevent his network from scanning. Type following iptable rule to reject 4-bit tcp-option in his network:

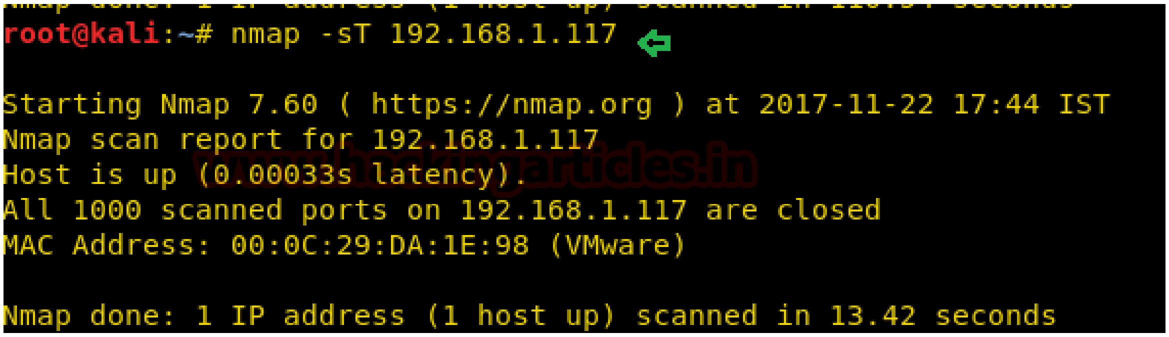
```
iptables -I INPUT -p tcp --tcp-option 4  -j REJECT --reject-with tcp-reset
```



```
root@ubuntu:~# iptables -I INPUT -p tcp --tcp-option 4 -j REJECT --reject-with tcp-reset
root@ubuntu:~#
```

Now when an attacker will perform TCP scanning [sT] on the target's network, he could not able to enumerate ports and running service of the victim's system. Since tcp-option is 4 bytes hence firewall discard tcp packet of attacker's network.

```
nmap -sT 192.168.1.117
```



```
root@kali:~# nmap -sT 192.168.1.117

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-22 17:44 IST
Nmap scan report for 192.168.1.117
Host is up (0.00033s latency).
All 1000 scanned ports on 192.168.1.117 are closed
MAC Address: 00:0C:29:DA:1E:98 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 13.42 seconds
```

The IP protocol gives numerous options that could be placed in packet headers. Contrasting the omnipresent TCP options, IP options are seldom observed because of security reasons. The most powerful way to specify IP options is to simply pass in hexadecimal data as the argument to –ip-options.

Precede every hex byte value with \x. You may repeat certain characters by following them with an asterisk and then the number of times you wish them to repeat. For example, \x01\x07\x04\x00*4 is the same as\x01\x07\x04\x00\x00\x00\x00 this is also called NuLL bytes

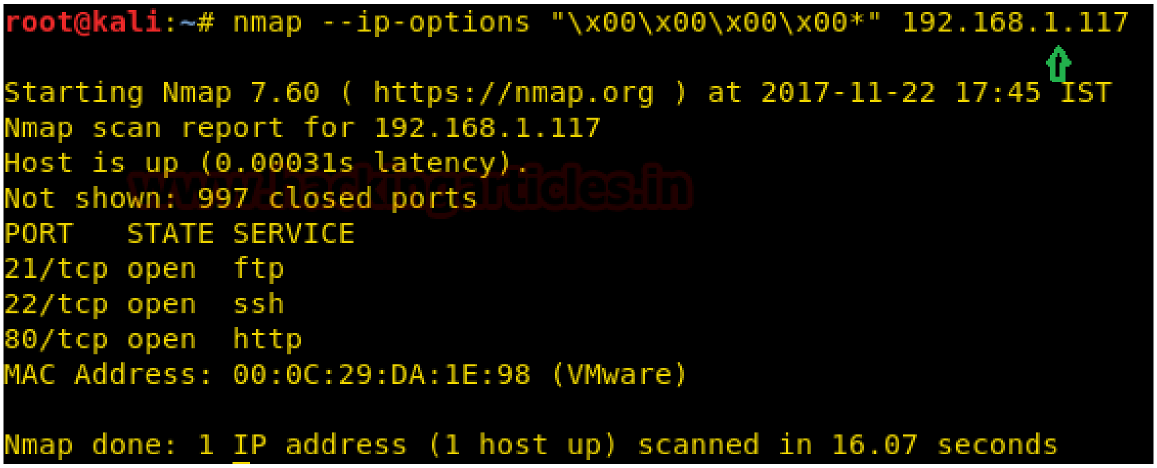
Now type the following command with the ip-option argument as shown below:

```
nmap --ip-options "\x00\x00\x00\x00\x00*" 192.168.1.117
```

Note that if you denote the number of bytes that is not a multiple of four; an incorrect IP header length will be set in the IP packet. The reason for this is that the IP header length field can only express multiples of four. In those cases, the length is computed by dividing the header length by 4 and rounding down.

GOOD! If you will notice given below image you will observe open ports of the target's network.

https://nmap.org/book/nping-man-ip-options.html



```
root@kali:~# nmap --ip-options "\x00\x00\x00\x00*" 192.168.1.117

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-22 17:45 IST
Nmap scan report for 192.168.1.117
Host is up (0.00031s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:0C:29:DA:1E:98 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 16.07 seconds
```