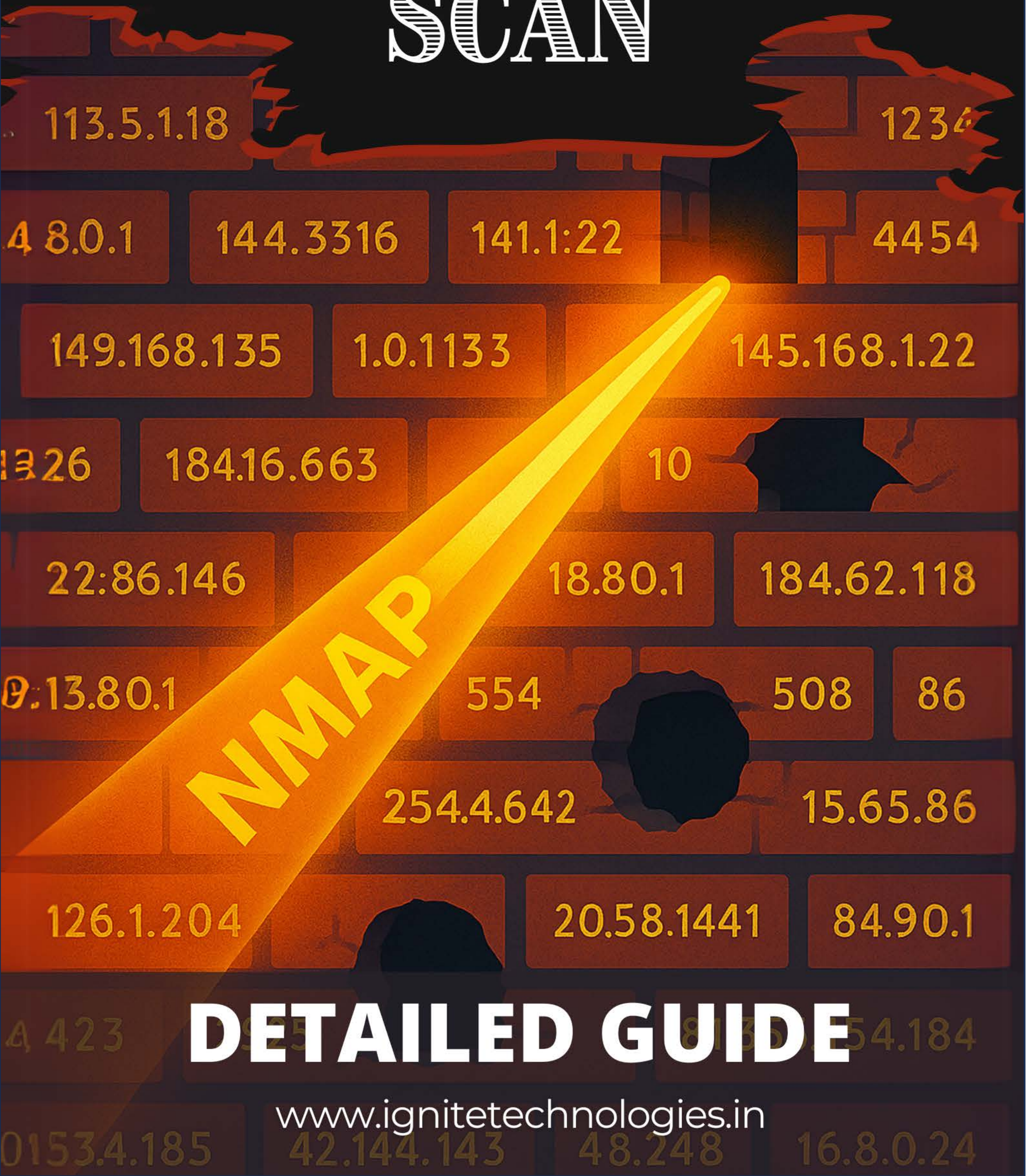
iGNITE
Technologies

NMAP FIREWALL SCAN



DETAILED GUIDE

www.ignitetechnologies.in

Contents

Introduction

Several times you might have used **NMAP** to performing Network scanning for enumerating active Port services of target machine but in some scenario. It is not possible to perform scanning with help of basic scan method especially in case of firewall filter.

Today we are going to demonstrate "Nmap firewall scan" by making use of Iptable rules and try to bypass the firewall filter to perform NMAP Advance scanning.

Let's Begin!!

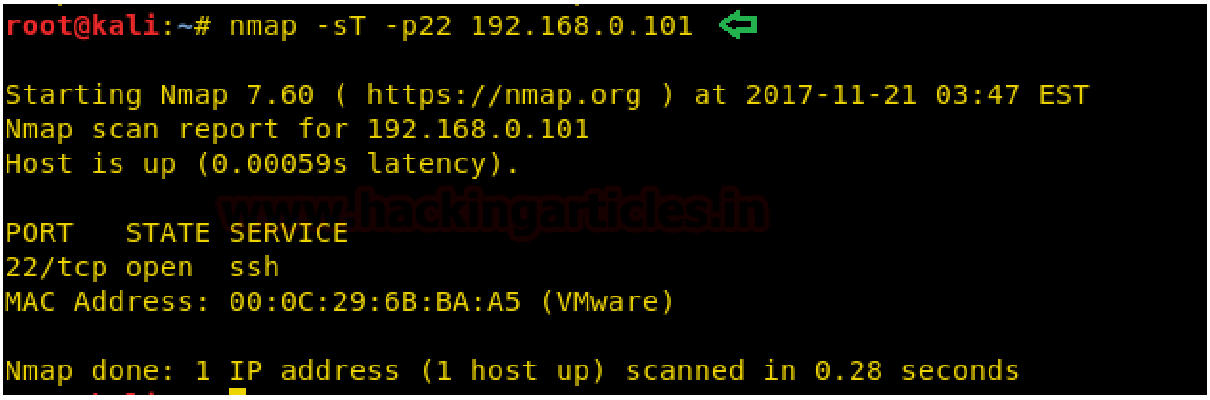
Attacker's IP: 192.168.0.107 [kali linux]
Target's IP: 192.168.0.101 [Ubuntu]

Analysis TCP Scan

Open the terminal in your Kali Linux and execute the following command to perform TCP[sT] scan for open port enumeration.

```
nmap -sT -p22 192.168.1.101
```

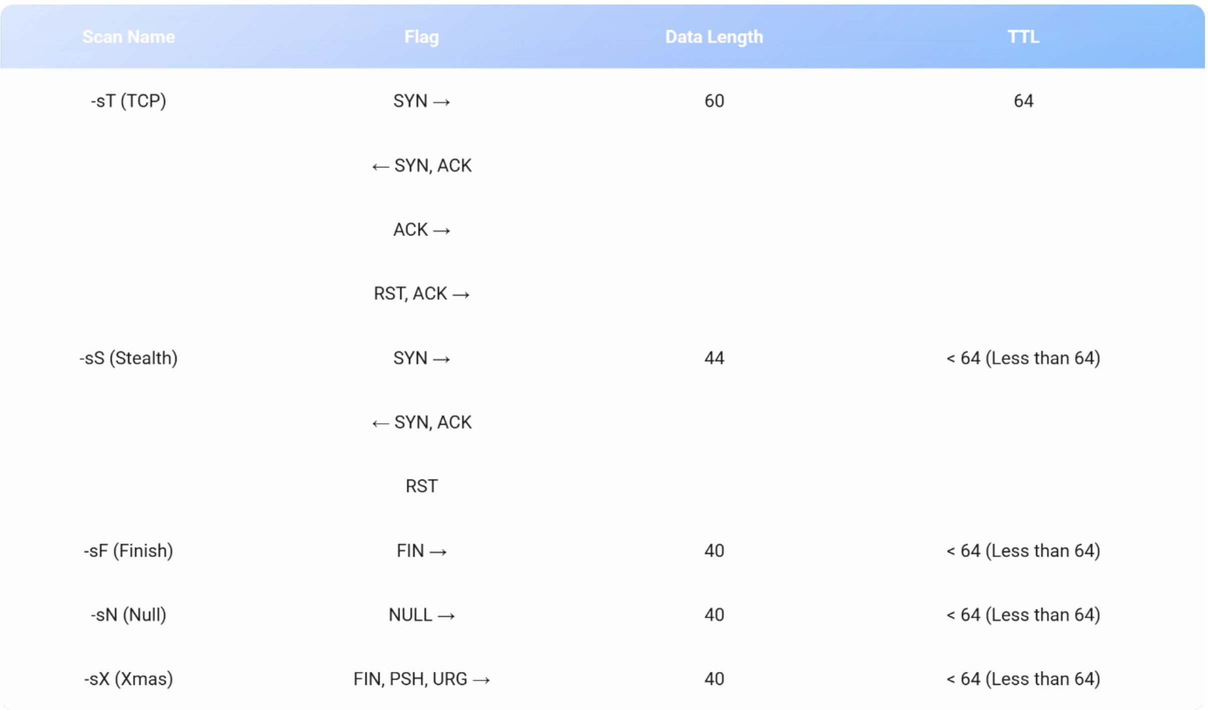
From the image given below you can observe we had scanned port 22 as result it has shown Port 22 is **Open** for SSH service.



```
root@kali:~# nmap -sT -p22 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 03:47 EST
Nmap scan report for 192.168.0.101
Host is up (0.00059s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds
```

When you use Wireshark in order to capture the packet send in the case of TCP while the network is being scanned. Here you need to notice few things such as "flag, Total length and time to live[TTL]" [in layer3].

The following table contains details of Flag, Data length and TTL in different scanning method:

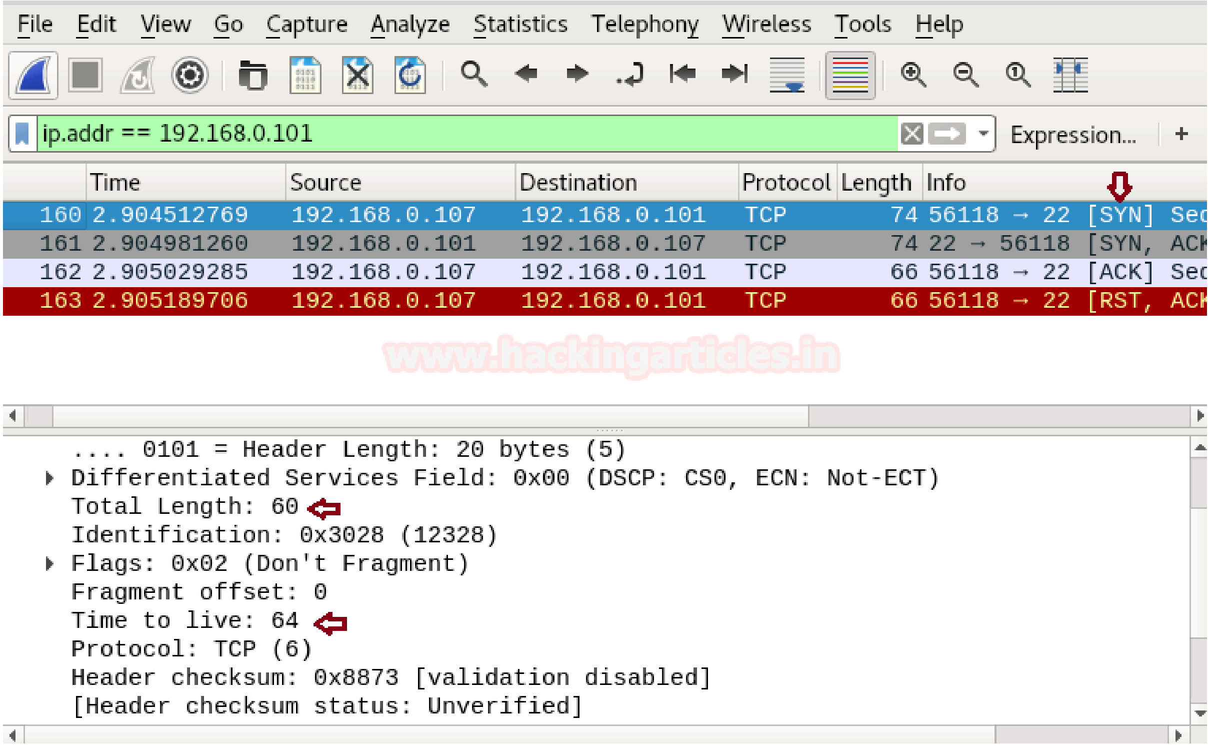


Scan Name	Flag	Data Length	TTL
-sT (TCP)	SYN →	60	64
	← SYN, ACK		
	ACK →		
	RST, ACK →		
-sS (Stealth)	SYN →	44	< 64 (Less than 64)
	← SYN, ACK		
	RST		
-sF (Finish)	FIN →	40	< 64 (Less than 64)
-sN (Null)	NULL →	40	< 64 (Less than 64)
-sX (Xmas)	FIN, PSH, URG →	40	< 64 (Less than 64)

Following image of Wireshark is showing network traffic generated while nmap TCP scan is running. Here, 1st stream indicates **SYN packet** which contains the following information:

Total Length: **60** [data length excluding 14 bytes of Ethernet]

Time to live: **64** [it is maximum TTL of the Linux system in TCP communication]



ip.addr == 192.168.0.101

```
    Time          Source          Destination     Protocol Length Info
160 2.904512769   192.168.0.107   192.168.0.101   TCP         74 56118 → 22 [SYN] Seq
161 2.904981260   192.168.0.101   192.168.0.107   TCP         74 22 → 56118 [SYN, ACK
162 2.905029285   192.168.0.107   192.168.0.101   TCP         66 56118 → 22 [ACK] Seq
163 2.905189706   192.168.0.107   192.168.0.101   TCP         66 56118 → 22 [RST, ACK
```

```
   .... 0101 = Header Length: 20 bytes (5)
 ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
   Total Length: 60
   Identification: 0x3028 (12328)
 ▸ Flags: 0x02 (Don't Fragment)
   Fragment offset: 0
   Time to live: 64
   Protocol: TCP (6)
   Header checksum: 0x8873 [validation disabled]
   [Header checksum status: Unverified]
```

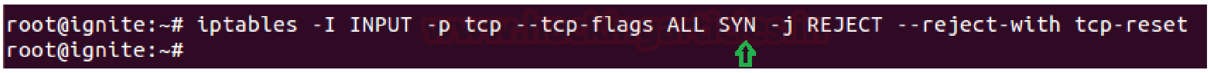
Reject SYN Flag with IPTables

As we know there is the strong fight between security researcher and attacker. To increase network security admin will apply firewall filter which will now prevent 3-way handshake communication in the network. And resists attacker to perform TCP scan by rejecting **SYN packet** in the network.

Execute given below command in Ubuntu to block SYN packet:

```
iptables -I INPUT -p tcp --tcp-flags ALL SYN -j REJECT --reject-with tcp-reset
```

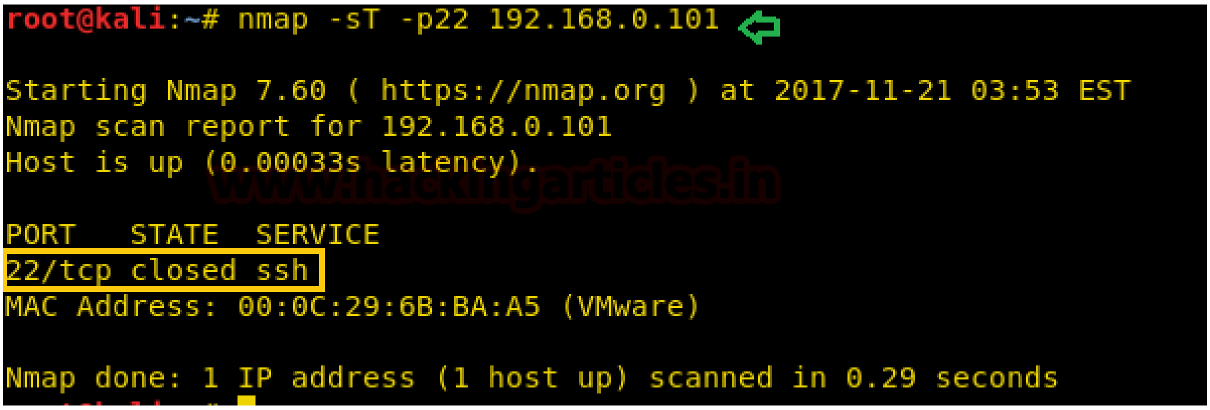
Iptable work as the firewall in the Linux operating system and above iptable rule will reject SYN packet to prevent TCP scan.



```
root@ignite:~# iptables -I INPUT -p tcp --tcp-flags ALL SYN -j REJECT --reject-with tcp-reset
root@ignite:~#
```

Now when the firewall in the target network rejects the SYN packe. The attacker will be unable to enumerate open ports of the target's network even if services are activated.

Now when we [the attacker] execute the TCP scan again, we see that **Port 22 is closed** as shown in the given image.



```
root@kali:~# nmap -sT -p22 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 03:53 EST
Nmap scan report for 192.168.0.101
Host is up (0.00033s latency).

PORT   STATE  SERVICE
22/tcp closed ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
```

Bypass SYN Filter

When the attacker fails to enumerate open port using a TCP scan. Then there are some advanced scanning methods used to bypass such type of firewall filter as given below :

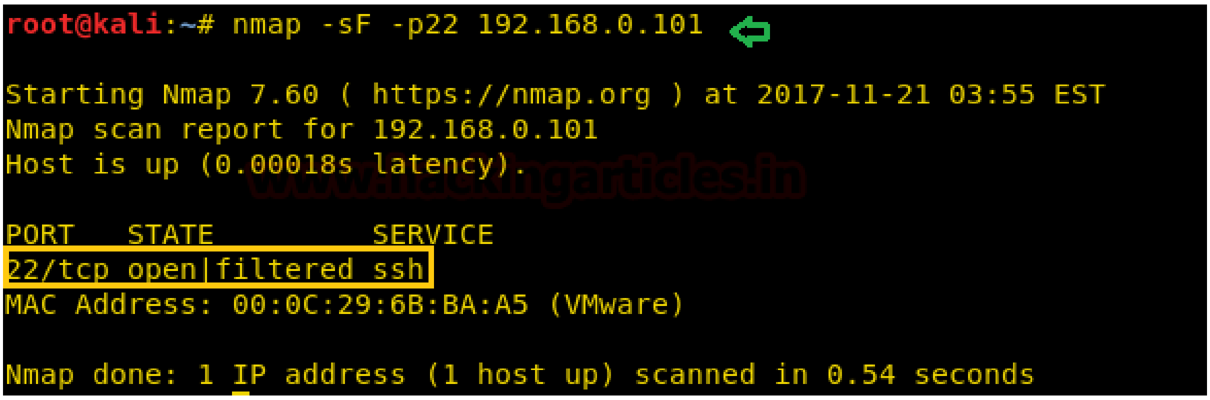
FIN Scan

The TCP connection between the source and destination port typically terminates after the data transfer is complete using the FIN packet. In the place of the SYN packet, Nmastartsrt a FIN scan by sending FIN packet.

Fin Scan only works on Linux machine and does not work on latest version of windows

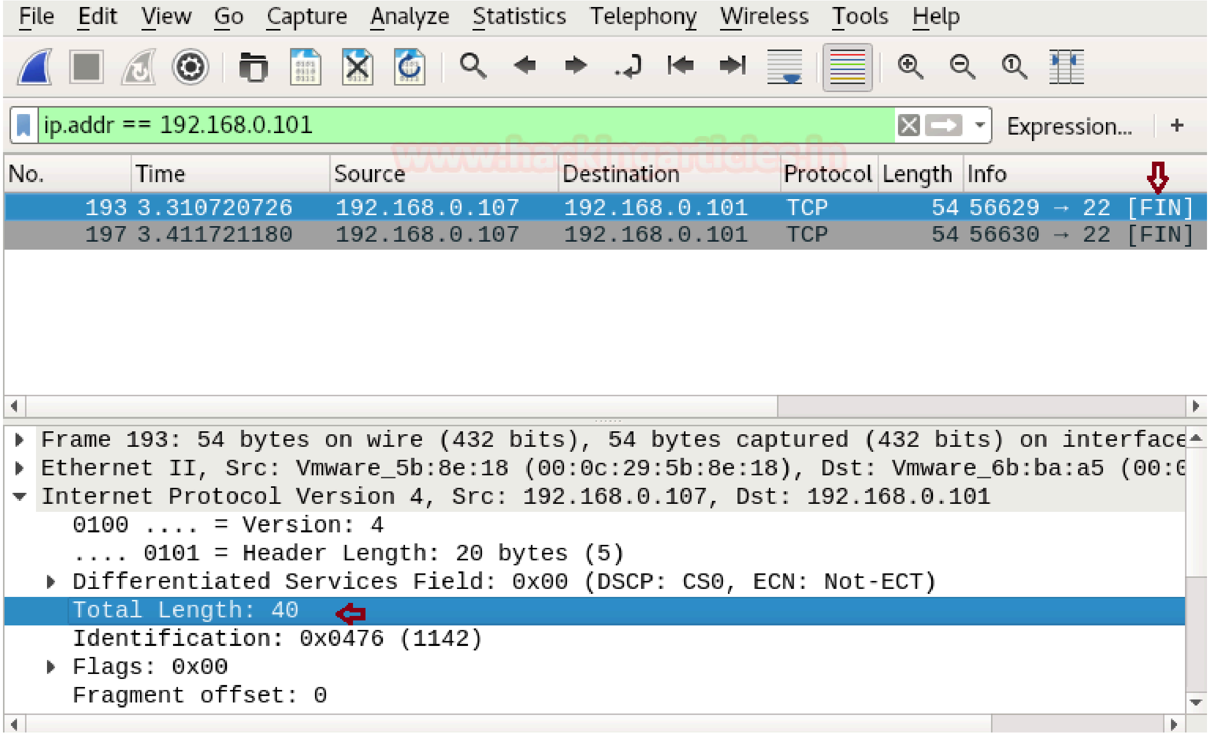
```
nmap -sF -p22 192.168.0.101
```

Frothe m given image you can observe the result that port **22** is **open.**



```
root@kali:~# nmap -sF -p22 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 03:55 EST
Nmap scan report for 192.168.0.101
Host is up (0.00018s latency).

PORT   STATE         SERVICE
22/tcp open|filtered ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.54 seconds
```

When you will capture network traffic for FIN packet. You can bear out “data length” is 40 and “TTL” will be less than 64 every time. Moreover, there is no use of SYN packet to establish TCP communication with target machine.



ip.addr == 192.168.0.101

```
No.      Time          Source          Destination     Protocol Length Info
     193 3.310720726   192.168.0.107   192.168.0.101   TCP          54 56629 → 22 [FIN]
     197 3.411721180   192.168.0.107   192.168.0.101   TCP          54 56630 → 22 [FIN]
```

```
▸ Frame 193: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface
▸ Ethernet II, Src: Vmware_5b:8e:18 (00:0c:29:5b:8e:18), Dst: Vmware_6b:ba:a5 (00:0
▾ Internet Protocol Version 4, Src: 192.168.0.107, Dst: 192.168.0.101
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
    Identification: 0x0476 (1142)
  ▸ Flags: 0x00
    Fragment offset: 0
```

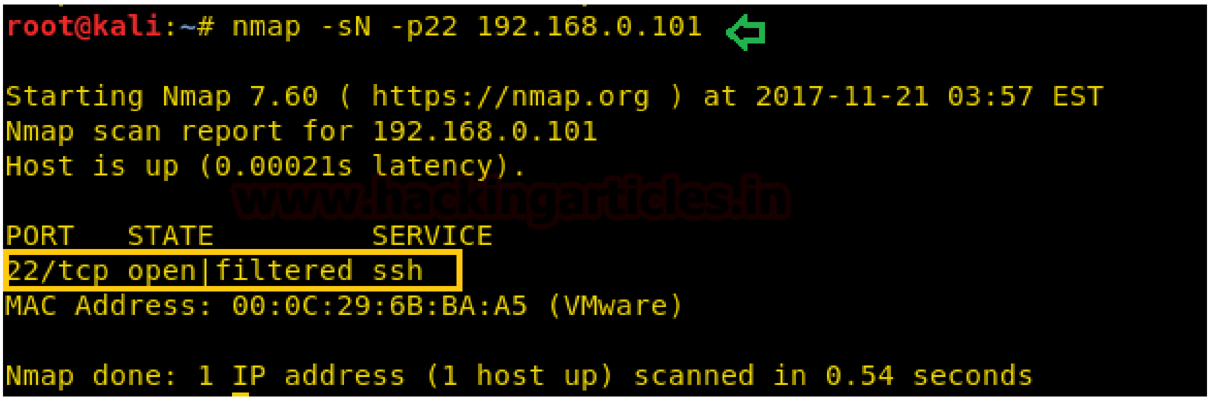
NULL Scan

A series of TCP packets with a sequence number of “zeros” (0000000) make up a Null Scan. Since none of the flags are set, the destination will not know how to reply to the request. It will discard the packet and send no reply, which indicates that the port is open.

Null Scan are only workable in Linux machines and does not work on the latest version of windows

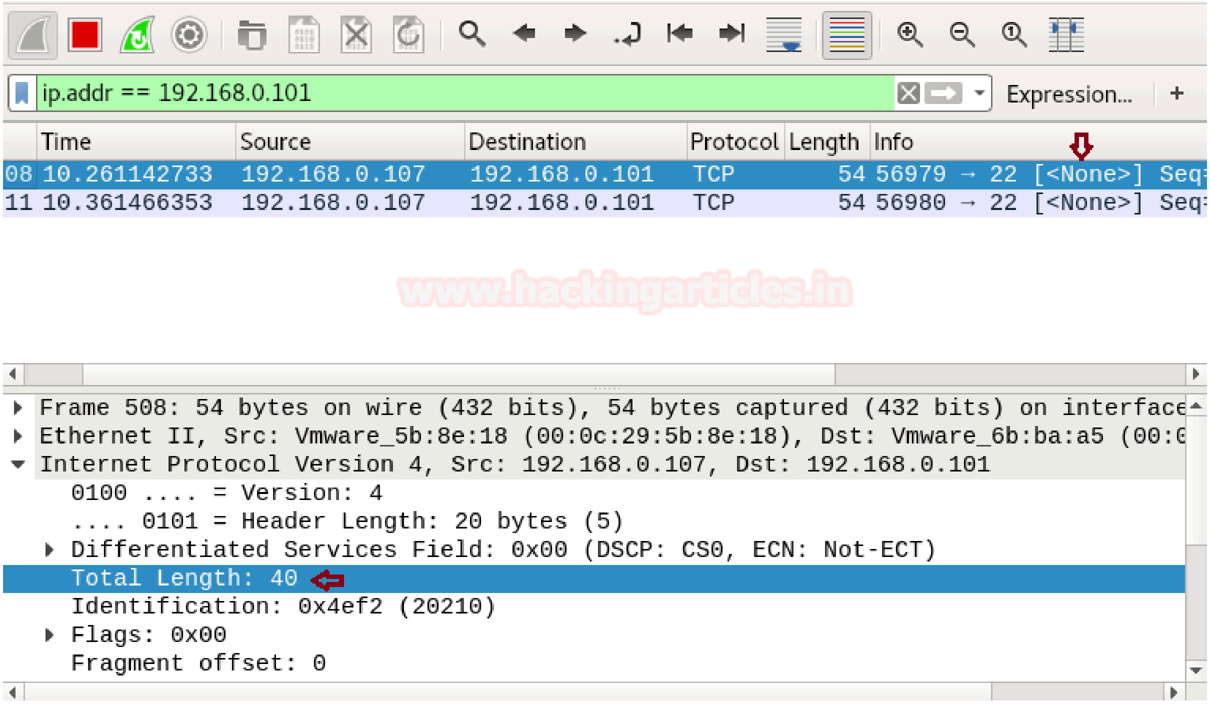
```
nmap -sN -p22 192.168.0.101
```

For the m given image you can observe the result that port **22** is **open.**



```
root@kali:~# nmap -sN -p22 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 03:57 EST
Nmap scan report for 192.168.0.101
Host is up (0.00021s latency).

PORT   STATE         SERVICE
22/tcp open|filtered ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.54 seconds
```

Similarly, When you will capture network traffic for the NULL packet, you can bear out “data length” is 40 and “TTL” will be less than 64 every time. Here also there is no use of SYN packet to establish TCP communication with target machine.



```
    Time          Source         Destination    Protocol Length Info
08 10.261142733  192.168.0.107  192.168.0.101  TCP          54 56979 → 22 [<None>] Seq=
11 10.361466353  192.168.0.107  192.168.0.101  TCP          54 56980 → 22 [<None>] Seq=
```

```
▸ Frame 508: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface
▸ Ethernet II, Src: Vmware_5b:8e:18 (00:0c:29:5b:8e:18), Dst: Vmware_6b:ba:a5 (00:0
▾ Internet Protocol Version 4, Src: 192.168.0.107, Dst: 192.168.0.101
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
    Identification: 0x4ef2 (20210)
  ▸ Flags: 0x00
    Fragment offset: 0
```

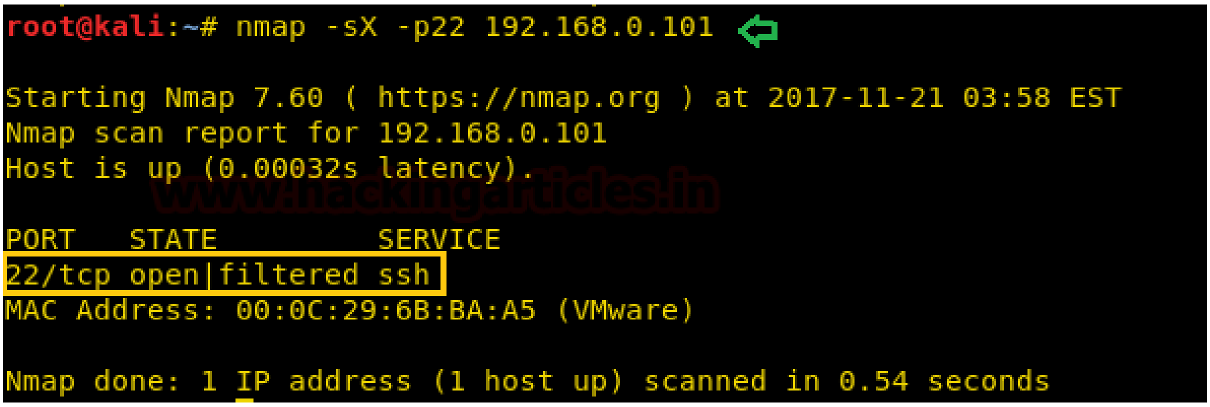
XMAS Scan

These scans manipulate the PSH, URG, and FIN flags of the TCP header, setting the FIN, PSH, and URG flags, lighting the packet up like a Christmas tree. When source sent FIN, PUSH, and URG packet to a specific port and if a port is open. Then destination will discard the packets and will not sent any reply to a source.

Xmas Scan are only workable in Linux machines and does not work on the latest version of windows

```
nmap -sX -p22 192.168.0.101
```

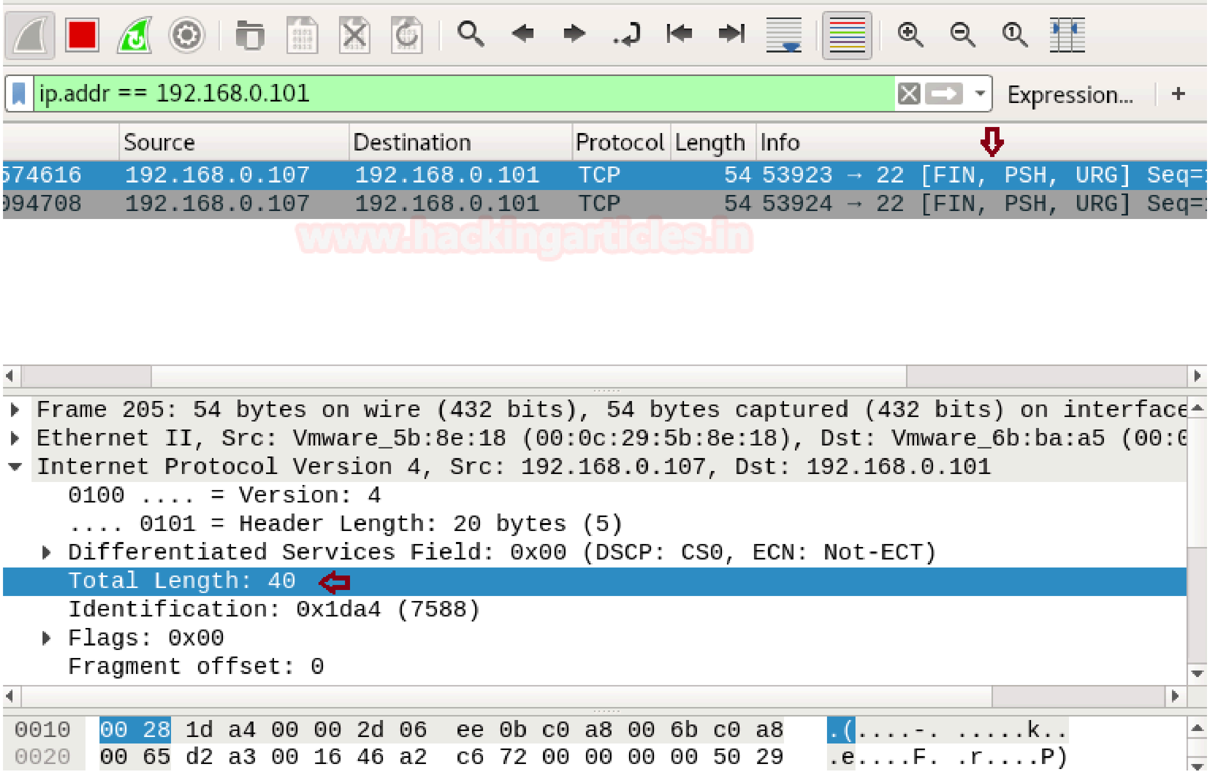
From the given image you can observe the result that port **22** is **open.**



```
root@kali:~# nmap -sX -p22 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 03:58 EST
Nmap scan report for 192.168.0.101
Host is up (0.00032s latency).

PORT   STATE         SERVICE
22/tcp open|filtered ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.54 seconds
```

Similarly, When you will capture network traffic for xmas scan you will get the combination of FIN, PSH and URG flags, here also you can bear out “data length” is 40 and “TTL” will be less than 64 every time.

Conclusion: The TCP connection is established by the 3-way handshake, and if the firewall discards the 3-way handshake to prevent TCP communication. Then FIN, NULL, and XMAS scans are used for the TCP connection.



```
ip.addr == 192.168.0.101

       Source          Destination     Protocol Length Info
574616 192.168.0.107   192.168.0.101   TCP          54 53923 → 22 [FIN, PSH, URG] Seq=
994708 192.168.0.107   192.168.0.101   TCP          54 53924 → 22 [FIN, PSH, URG] Seq=
```

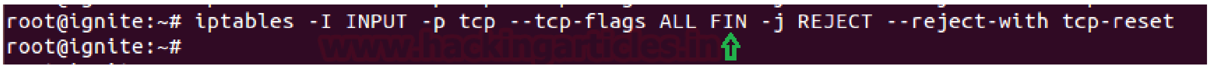
```
▸ Frame 205: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface
▸ Ethernet II, Src: Vmware_5b:8e:18 (00:0c:29:5b:8e:18), Dst: Vmware_6b:ba:a5 (00:0
▾ Internet Protocol Version 4, Src: 192.168.0.107, Dst: 192.168.0.101
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
    Identification: 0x1da4 (7588)
  ▸ Flags: 0x00
    Fragment offset: 0

0010  00 28 1d a4 00 00 2d 06  ee 0b c0 a8 00 6b c0 a8   .(....-. .....k..
0020  00 65 d2 a3 00 16 46 a2  c6 72 00 00 00 00 50 29   .e....F. .r....P)
```

Reject FIN Packet Using IPTABLES Rule

Again admin add a new firewall filter to Prevent Network enumeration from Fin scan which will reject FIN packet in the network.

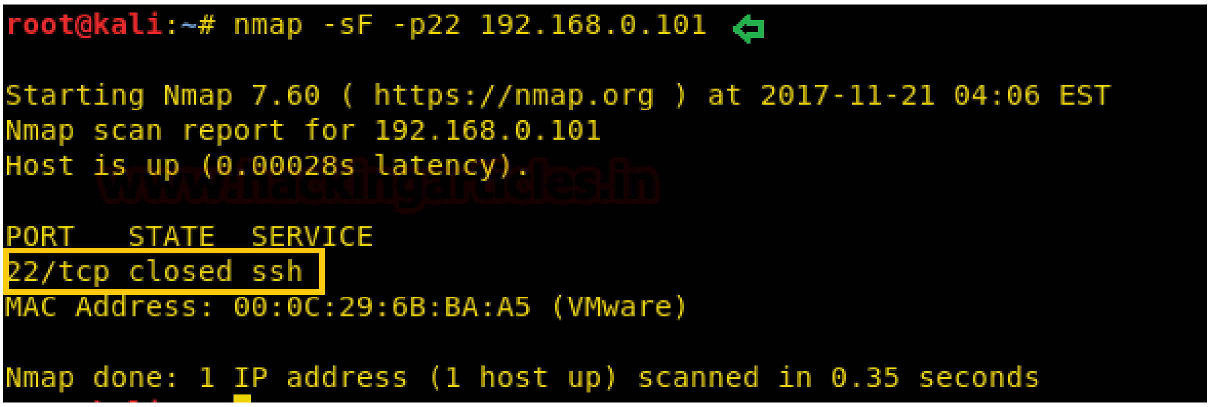
Execute given below command in Ubuntu to block FIN packet:

```
iptables -I INPUT -p tcp --tcp-flags ALL FIN -j REJECT --reject-with tcp-reset
```



```
root@ignite:~# iptables -I INPUT -p tcp --tcp-flags ALL FIN -j REJECT --reject-with tcp-reset
root@ignite:~#
```

Now when the attacker will try to perform advance scan through FIN scan. Then he will not able to enumerate open port information which you can confirm from given below image.

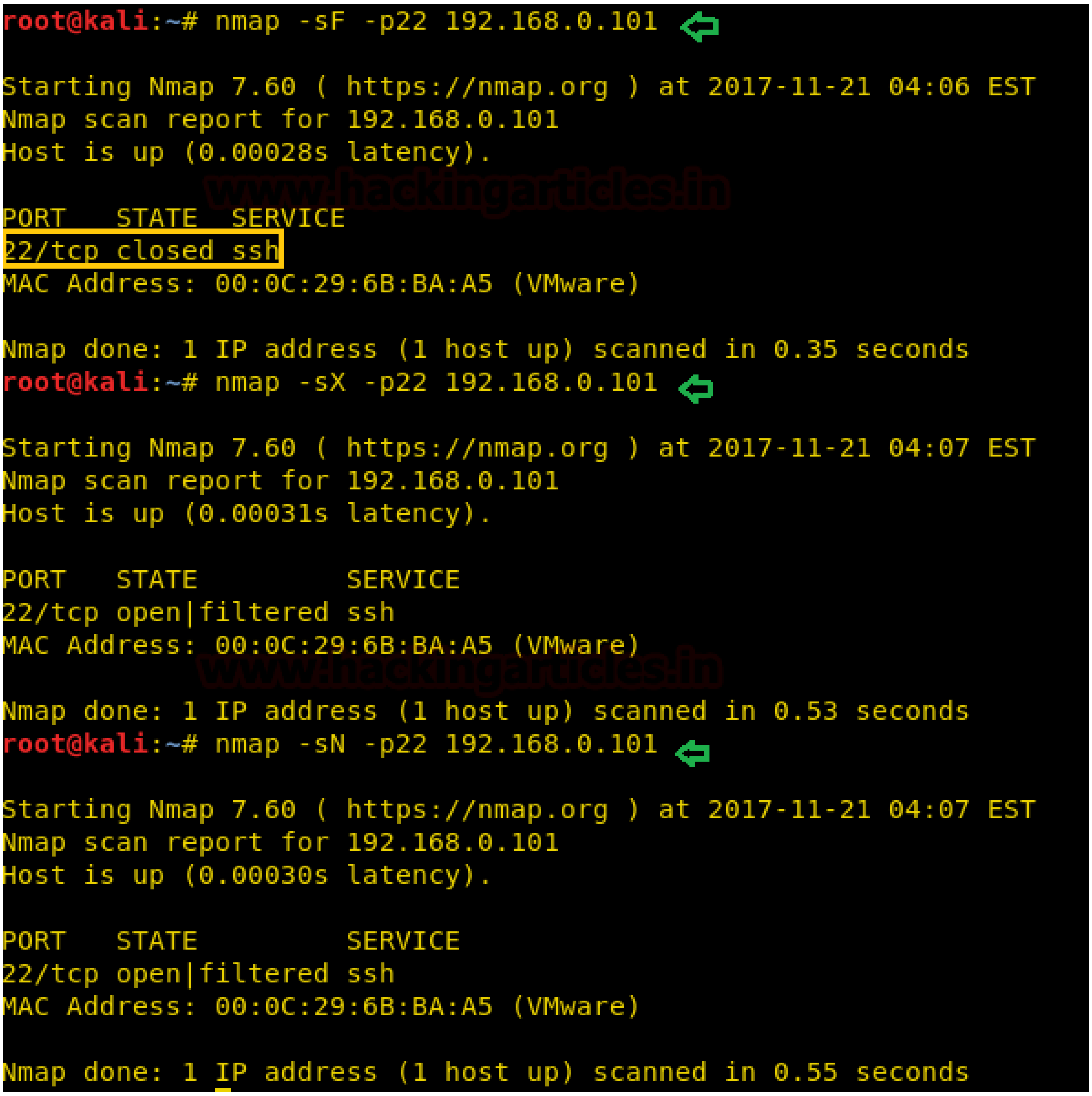


```
root@kali:~# nmap -sF -p22 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:06 EST
Nmap scan report for 192.168.0.101
Host is up (0.00028s latency).

PORT   STATE  SERVICE
22/tcp closed ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
```

At present only Null and Xmas will helpful to perform port enumeration until unless admin has not block traffic coming from these scan. From given below image you can confirm that port **22 is close** when Fin scan is performed while open when Null and Xmas is performed.

To prevent you network from NULL and Xmas scan too, apply given below iptables rule for Null and Xmas respectively:

```
iptables -I INPUT -p tcp --tcp-flags ALL NONE -j REJECT --reject-with tcp-reset
iptables -I INPUT -p tcp --tcp-flags ALL FIN,PSH,URG -j REJECT --reject-with tcp-reset
```



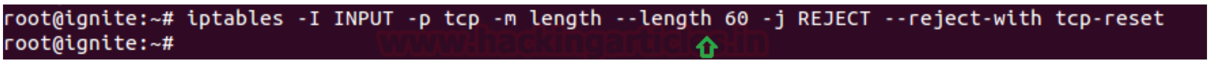
```
root@kali:~# nmap -sF -p22 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:06 EST
Nmap scan report for 192.168.0.101
Host is up (0.00028s latency).

PORT   STATE  SERVICE
22/tcp closed ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
root@kali:~# nmap -sX -p22 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:07 EST
Nmap scan report for 192.168.0.101
Host is up (0.00031s latency).

PORT   STATE         SERVICE
22/tcp open|filtered ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.53 seconds
root@kali:~# nmap -sN -p22 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:07 EST
Nmap scan report for 192.168.0.101
Host is up (0.00030s latency).

PORT   STATE         SERVICE
22/tcp open|filtered ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.55 seconds
```

Reject Data-length with IPTables

As I had discussed above TCP communication based upon 3 factors i.e. “Flag” which I had demonstrated above, “TTL” which I will demonstrate later and “Data length” which I am going to demonstrate.

So now when admin wants secure again his network from TCP scan, instead of applying firewall filter on TCP-flags. He can also apply firewall rule to check “data length” of a specific size. Then stop the incoming network traffic for TCP connection. Execute given below command to apply firewall rule on “data length”. By default 60 is data length use for TCP scan which you can confirm from the table given above.

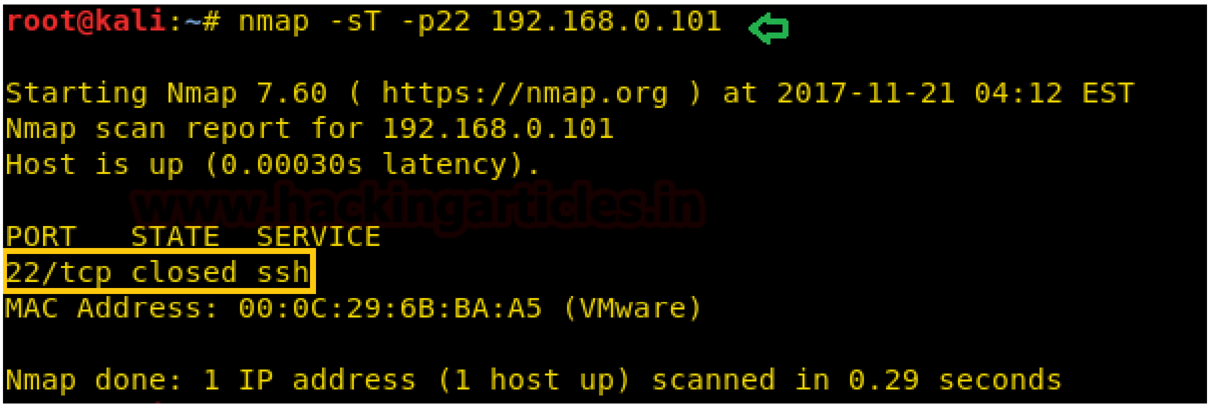
```
iptables -I INPUT -p tcp -m length --length 60 -j REJECT --reject-with tcp-reset
```



```
root@ignite:~# iptables -I INPUT -p tcp -m length --length 60 -j REJECT --reject-with tcp-reset
root@ignite:~#
```

Now when the firewall in the target network blocks the data length of 60 bytes. The attacker will be unable to enumerate the open ports of the target even if the service is activated.

Now when we [the attacker] executed the TCP scan again, we found that **Port 22 is closed** as shown in the given image.



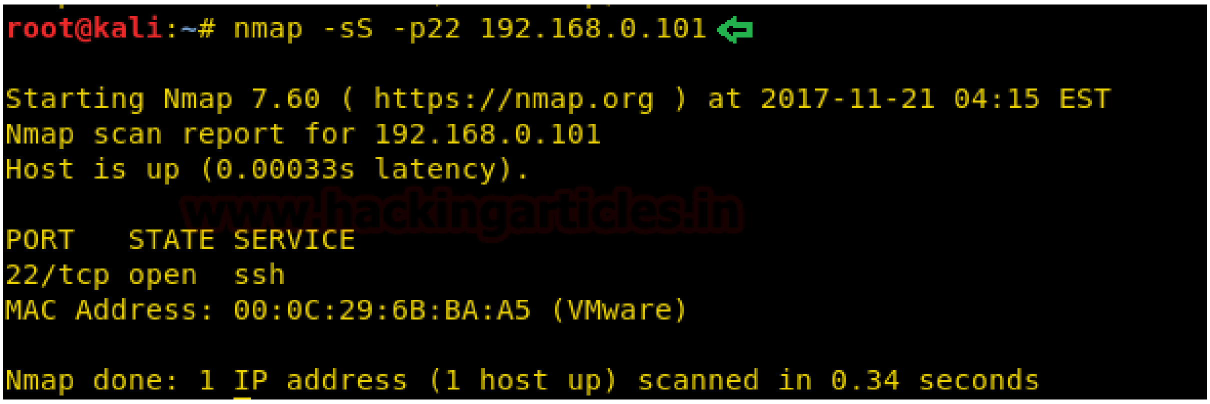
```
root@kali:~# nmap -sT -p22 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:12 EST
Nmap scan report for 192.168.0.101
Host is up (0.00030s latency).

PORT   STATE  SERVICE
22/tcp closed ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
```

Bypass Data-Length Restriction with Stealth Scan

When attacker fail to enumerate open port using TCP [sT] scan then there are some scanning method used to bypass such type of firewall filter as given below:

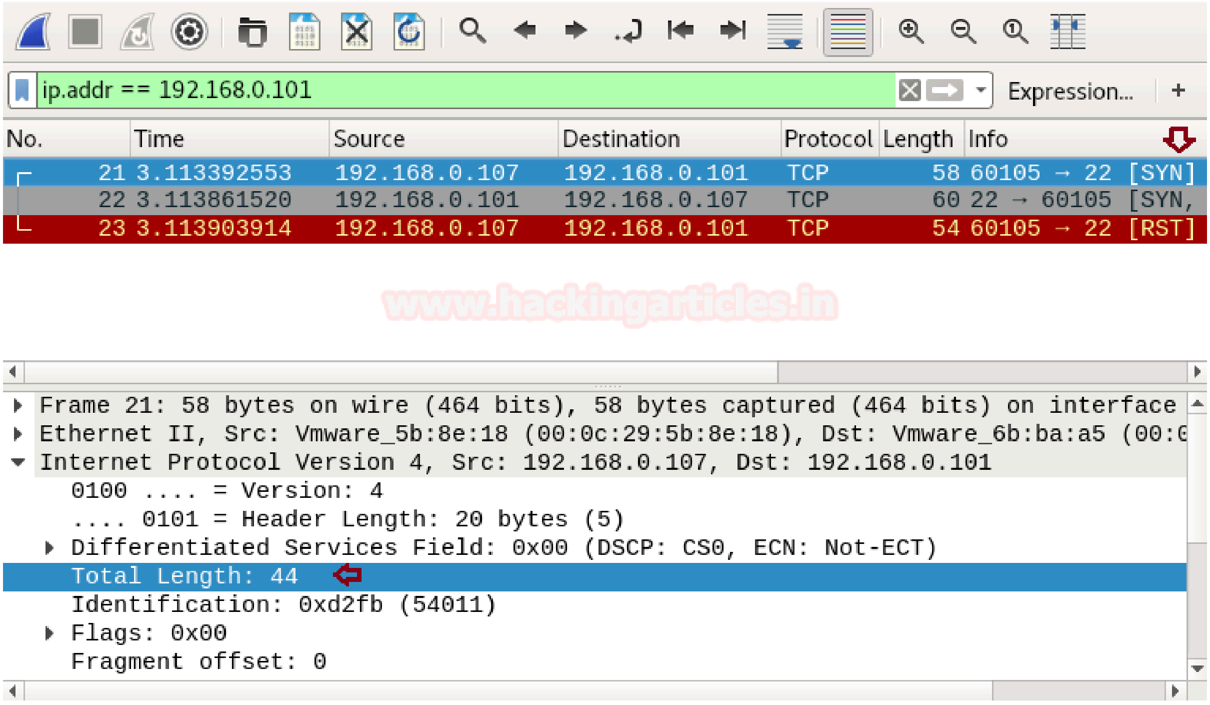
```
nmap -sS -p 22 192.168.0.101
```

From the given below image, you can observe that when you execute the stealth scan [sS], it opens port 22 because the stealth scan sends a data length of 44 by default for the TCP connection.



```
root@kali:~# nmap -sS -p22 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:15 EST
Nmap scan report for 192.168.0.101
Host is up (0.00033s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.34 seconds
```

Stealth scan is much similar to TCP scan and also known as “half open” scanning because it send SYN packet and as response receives SYN/ACK packet from listening port and dump result without sending an ACK packet to listening port. Therefore, if the firewall blocks the “SYN packet,” this scan fails. This scan is only applicable if the firewall blocks data length = 60 or TTL = 64.



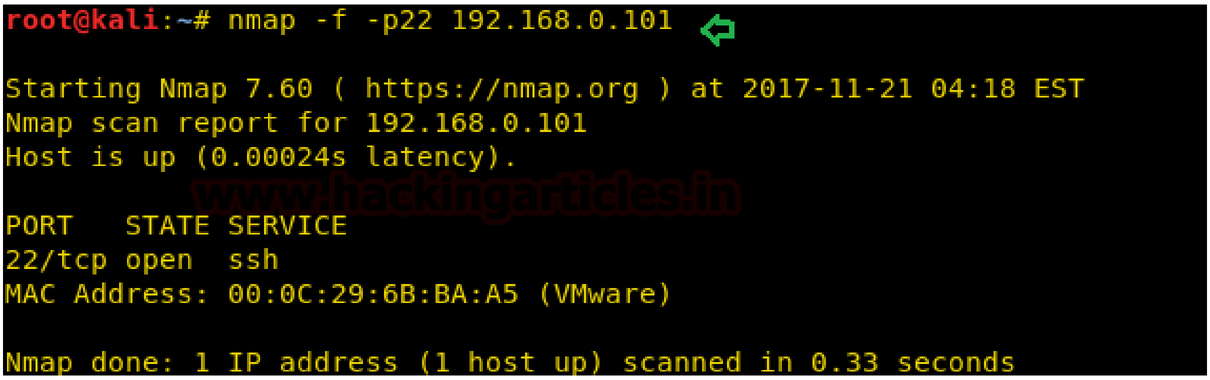
```
No.      Time          Source           Destination      Protocol Length Info
   21 3.113392553   192.168.0.107    192.168.0.101    TCP          58 60105 → 22 [SYN]
   22 3.113861520   192.168.0.101    192.168.0.107    TCP          60 22 → 60105 [SYN,
   23 3.113903914   192.168.0.107    192.168.0.101    TCP          54 60105 → 22 [RST]
```

```
▸ Frame 21: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface
▸ Ethernet II, Src: Vmware_5b:8e:18 (00:0c:29:5b:8e:18), Dst: Vmware_6b:ba:a5 (00:0
▾ Internet Protocol Version 4, Src: 192.168.0.107, Dst: 192.168.0.101
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 44
    Identification: 0xd2fb (54011)
  ▸ Flags: 0x00
    Fragment offset: 0
```

Fragment Scan

The **-f option** causes the requested scan to use tiny fragmented IP packets. The idea is to split up the TCP header over several packets to make it harder for packet filters, intrusion detection systems, and other annoyances to detect what you are doing. So a 20-byte TCP header would be split into three packets, two with eight bytes of the TCP header, and one with the final four.

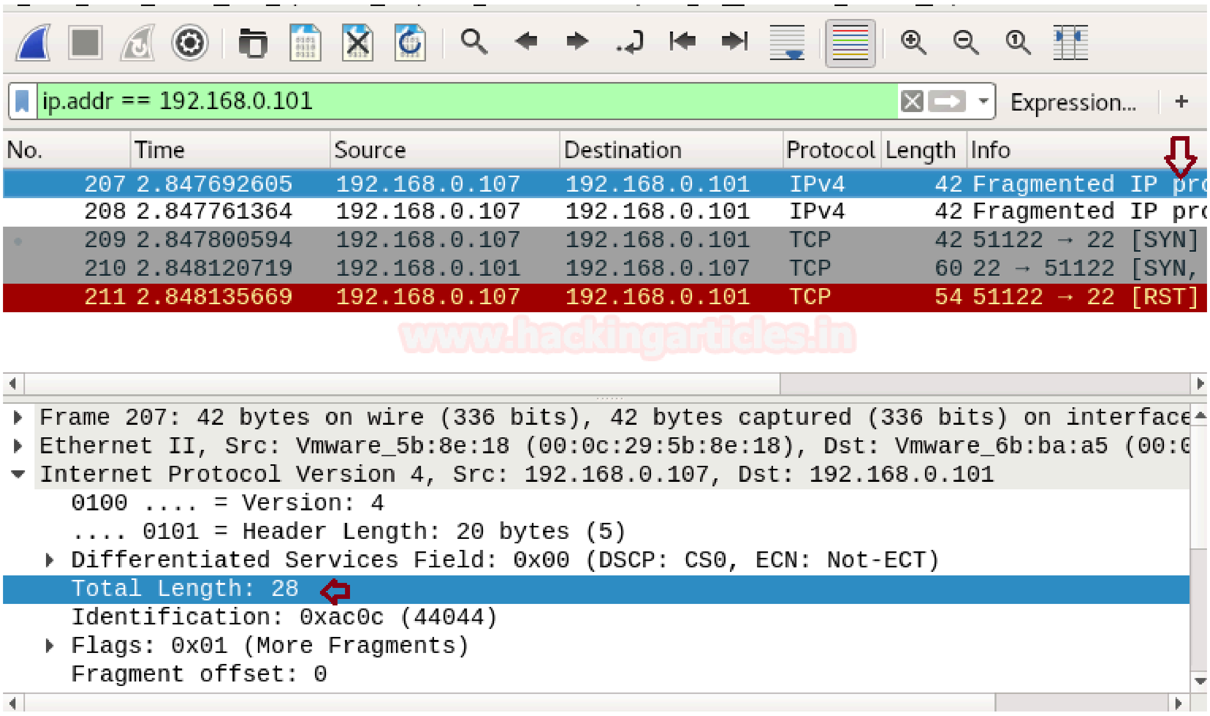
```
nmap -f -p22 192.168.0.101
```



```
root@kali:~# nmap -f -p22 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:18 EST
Nmap scan report for 192.168.0.101
Host is up (0.00024s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.33 seconds
```

When you will capture network traffic, you can bear out "data length" is 28 excluding 14 bytes of Ethernet and "TTL" will be less than 64 every time.

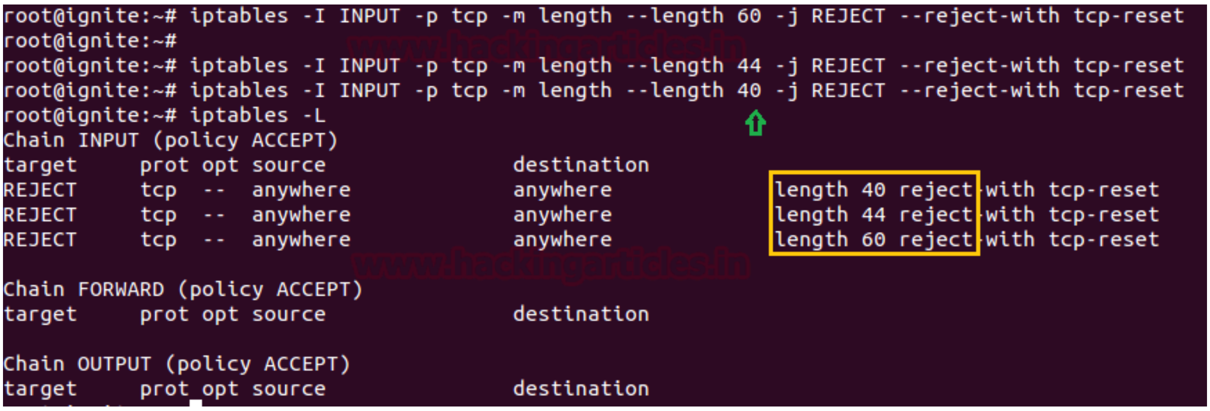
Similarly, you use Fin, Null and Xmas scan whose data length is 40 to enumerate open port of target network.



```
ip.addr == 192.168.0.101
No.  Time          Source         Destination    Protocol Length Info
 207 2.847692605   192.168.0.107  192.168.0.101  IPv4     42 Fragmented IP prc
 208 2.847761364   192.168.0.107  192.168.0.101  IPv4     42 Fragmented IP prc
 209 2.847800594   192.168.0.107  192.168.0.101  TCP      42 51122 → 22 [SYN]
 210 2.848120719   192.168.0.101  192.168.0.107  TCP      60 22 → 51122 [SYN,
 211 2.848135669   192.168.0.107  192.168.0.101  TCP      54 51122 → 22 [RST]

▸ Frame 207: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface
▸ Ethernet II, Src: Vmware_5b:8e:18 (00:0c:29:5b:8e:18), Dst: Vmware_6b:ba:a5 (00:0
▾ Internet Protocol Version 4, Src: 192.168.0.107, Dst: 192.168.0.101
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 28
    Identification: 0xac0c (44044)
  ▸ Flags: 0x01 (More Fragments)
    Fragment offset: 0
```

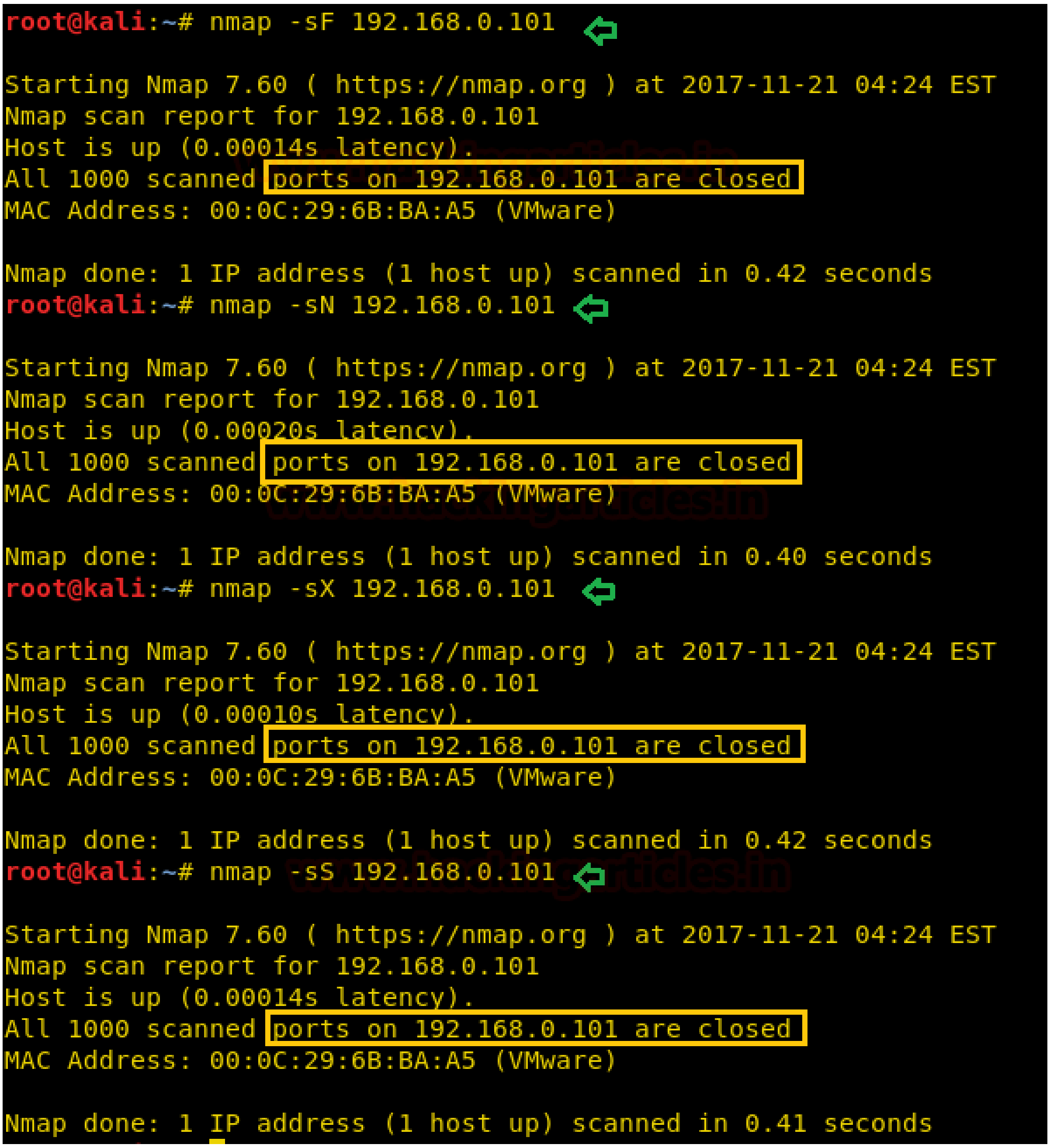
If admin will apply firewall filter to reject data length 40,44 and 60. Then it will not allow the attacker to perform above all scan either basic scan or advance scan by executing following iptables rules.

```
iptables -I INPUT -p tcp -m length --length 60 -j REJECT --reject-with tcp-reset
iptables -I INPUT -p tcp -m length --length 44 -j REJECT --reject-with tcp-reset
iptables -I INPUT -p tcp -m length --length 40 -j REJECT --reject-with tcp-reset
```



```
root@ignite:~# iptables -I INPUT -p tcp -m length --length 60 -j REJECT --reject-with tcp-reset
root@ignite:~#
root@ignite:~# iptables -I INPUT -p tcp -m length --length 44 -j REJECT --reject-with tcp-reset
root@ignite:~# iptables -I INPUT -p tcp -m length --length 40 -j REJECT --reject-with tcp-reset
root@ignite:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     tcp  --  anywhere             anywhere             length 40 reject-with tcp-reset
REJECT     tcp  --  anywhere             anywhere             length 44 reject-with tcp-reset
REJECT     tcp  --  anywhere             anywhere             length 60 reject-with tcp-reset

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

From given below image you can observe now Fin, null, Xmas and stealth scan are some examples which were unable to enumerate open port of target network. All are showing port is close even if service is activated.



```
root@kali:~# nmap -sF 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:24 EST
Nmap scan report for 192.168.0.101
Host is up (0.00014s latency).
All 1000 scanned ports on 192.168.0.101 are closed
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.42 seconds
root@kali:~# nmap -sN 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:24 EST
Nmap scan report for 192.168.0.101
Host is up (0.00020s latency).
All 1000 scanned ports on 192.168.0.101 are closed
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.40 seconds
root@kali:~# nmap -sX 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:24 EST
Nmap scan report for 192.168.0.101
Host is up (0.00010s latency).
All 1000 scanned ports on 192.168.0.101 are closed
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.42 seconds
root@kali:~# nmap -sS 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:24 EST
Nmap scan report for 192.168.0.101
Host is up (0.00014s latency).
All 1000 scanned ports on 192.168.0.101 are closed
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.41 seconds
```

Data Length Scan

Note:- Nmap "--data-length" works with only stealth scan. It won't work any other type of firewall scan. Some UDP and TCP ports get a custom payload by default for example:- FTP and SSH. Nmap will add ssh and Ftp headers while scanning port 21 and 22.

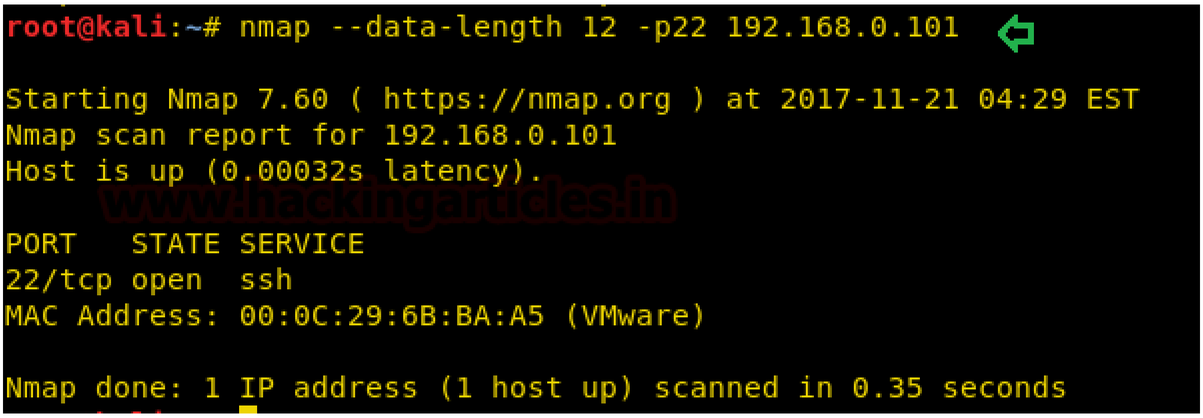
When an attacker is unable to enumerate open port by applying the above scan. Then he should go with nmap “data-length scan” which will bypass above firewall filter too.

By default nmap scan has fix data length as explain above. This scan let you append the random data length of your choice.

Using the following command attacker is trying to enumerate open port by defining data length 12

```
nmap --data-length 12 -p22 192.168.0.101
```

Awesome!! From given below image you can observe port 22 is open.



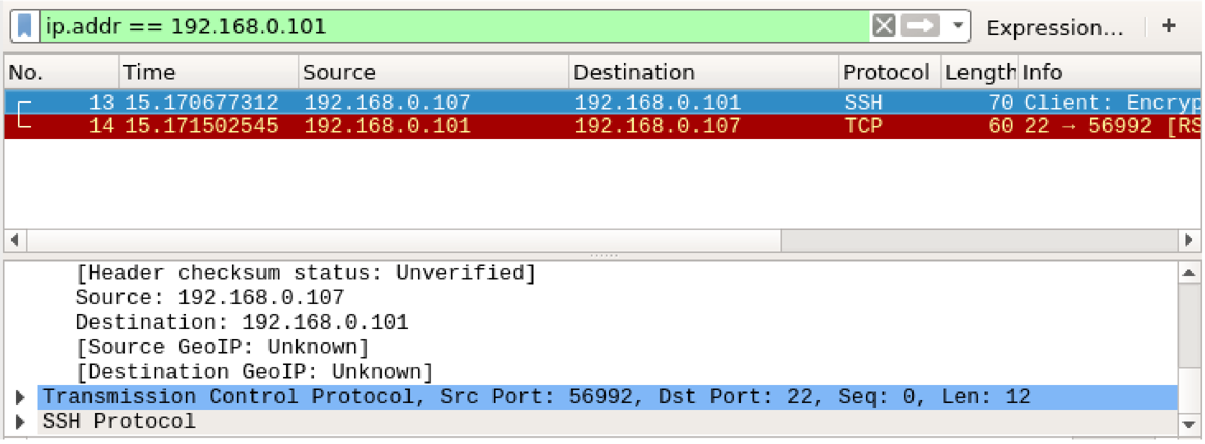
```
root@kali:~# nmap --data-length 12 -p22 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:29 EST
Nmap scan report for 192.168.0.101
Host is up (0.00032s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
```

So when you use Wireshark to capture network traffic generated while executing this scan, you will get “Total length” for TCP as 44.

Size of SSH packet is 70 bytes. Now reduce 14 bytes from its of Ethernet then remains 56 byte. Now reduce 12 bytes of data length which you have define at last total length will **44 bytes** left.

Here, 70 bytes -14 bytes[Ethernet] = 56 bytes

Now, 56 bytes -12 bytes[data-length] = 44 bytes



```
No.   Time          Source         Destination    Protocol Length Info
   13 15.170677312  192.168.0.107  192.168.0.101  SSH          70 Client: Encryp
   14 15.171502545  192.168.0.101  192.168.0.107  TCP          60 22 → 56992 [RS
```

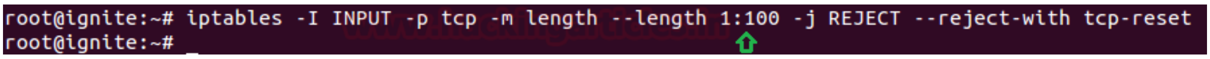
```
    [Header checksum status: Unverified]
    Source: 192.168.0.107
    Destination: 192.168.0.101
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
Transmission Control Protocol, Src Port: 56992, Dst Port: 22, Seq: 0, Len: 12
SSH Protocol
```

Reject Length size 1 to 100

If an admin is aware from nmap data-length scan. Then he should block a complete range of data length to prevent network scanning from the attacker by executing following iptable rule.

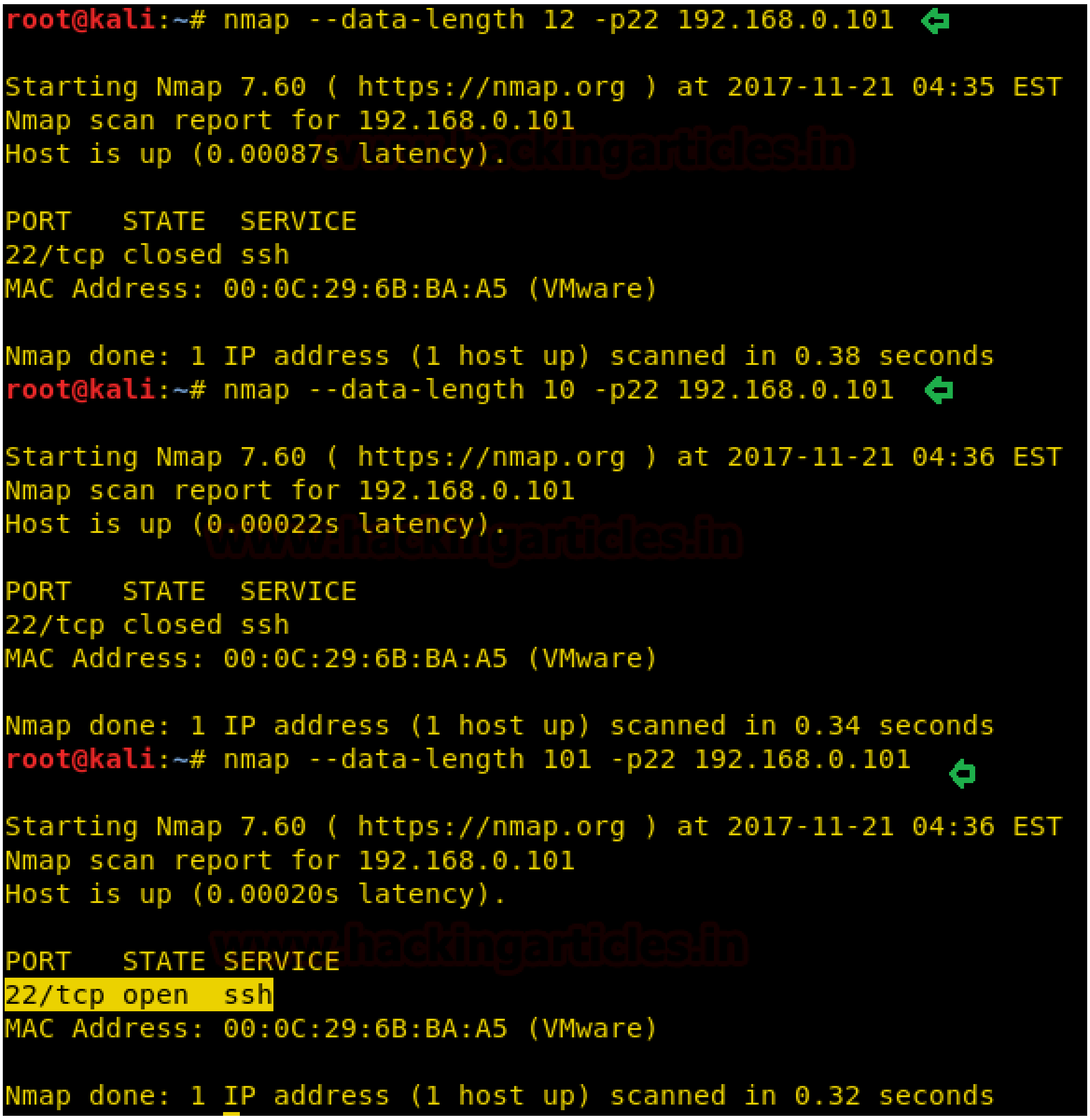
```
iptables -I INPUT -p tcp -m length --length 1:100 -j REJECT --reject-with tcp-reset
```

Now firewall will analysis traffic coming on its network. Then reject the packet which contains data-length from 1 byte to 100 bytes and deny to establish TCP connections with the attacker.



```
root@ignite:~# iptables -I INPUT -p tcp -m length --length 1:100 -j REJECT --reject-with tcp-reset
root@ignite:~# _
```

Now if the attacker sends a data length between 1 byte and 100 bytes. The port scanning fails to enumerate its open state. You can confirm this from the image below, which shows that when the scanner sends data lengths of 12 bytes and 10 bytes, port 22 is closed. As soon as the attacker sends a data length of 101 bytes, which exceeds 100 bytes, port 22 opens.



```
root@kali:~# nmap --data-length 12 -p22 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:35 EST
Nmap scan report for 192.168.0.101
Host is up (0.00087s latency).

PORT   STATE  SERVICE
22/tcp closed ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.38 seconds
root@kali:~# nmap --data-length 10 -p22 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:36 EST
Nmap scan report for 192.168.0.101
Host is up (0.00022s latency).

PORT   STATE  SERVICE
22/tcp closed ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.34 seconds
root@kali:~# nmap --data-length 101 -p22 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:36 EST
Nmap scan report for 192.168.0.101
Host is up (0.00020s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
```

TTL Scan

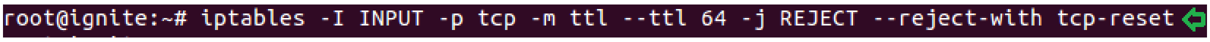
Reject TTL size with IPTables

After applying firewall filter on “TCP flags” and “data length” to secure network from enumeration now add firewall filter for “Time To Live” i.e. **TTL.**

If you had notice the table given in the beginning of the article. You will observe that only TCP Scan [sT] has TTL value equal to 64 else remaining scan has TTL value less than 64 every time. Hence if admin applies firewall filter to reject TTL value 64 then it will prevent network from TCP scanning.

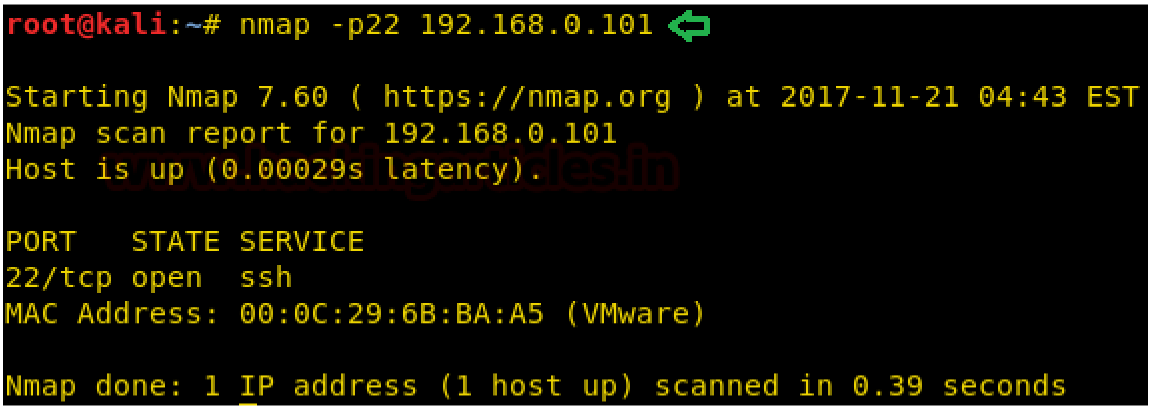
Given below command will add a new firewall rule to check the TTL value of 64 and reject the packet.

```
iptables -I INPUT -p tcp -m ttl --ttl 64 -j REJECT --reject-with tcp-reset
```



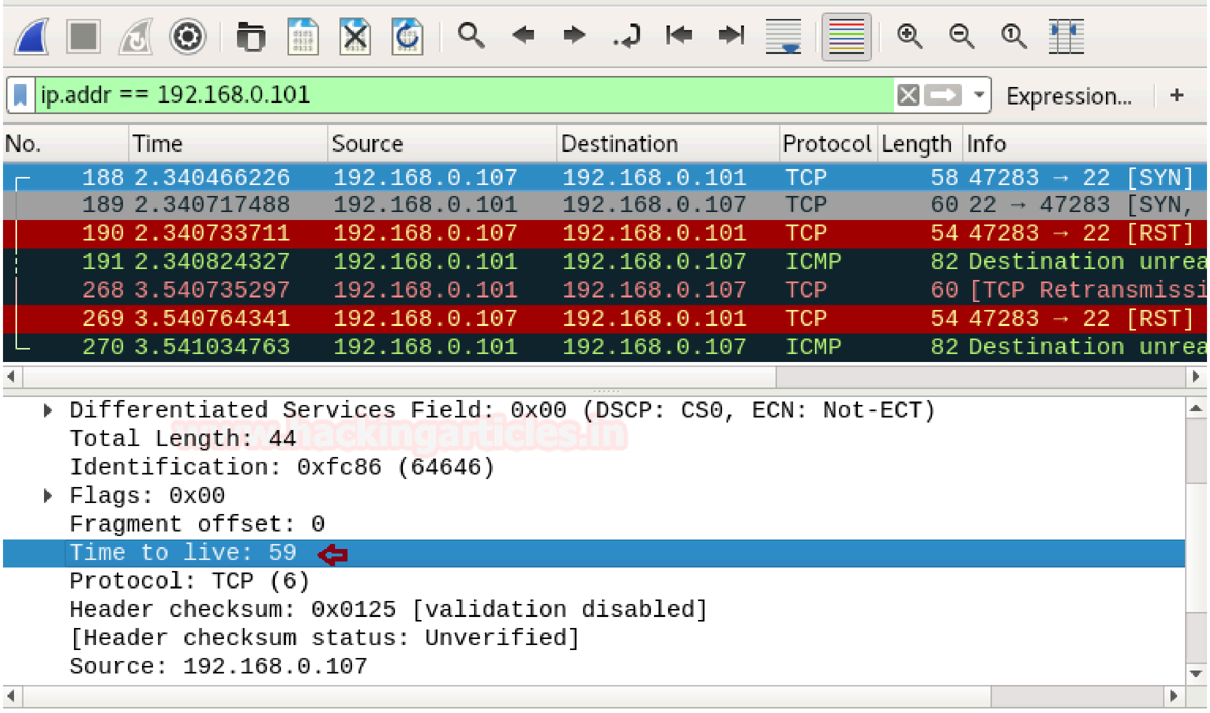
```
root@ignite:~# iptables -I INPUT -p tcp -m ttl --ttl 64 -j REJECT --reject-with tcp-reset
```

Now if an attacker uses “TCP [sT] scan” to enumerate port information, it will always show “port is closed”. If the attacker performs another scan, they will get accurate information related to the port state. From given below image you can observe when “basic scan is execute” to enumerate port details it give “port 22 is open”.



```
root@kali:~# nmap -p22 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:43 EST
Nmap scan report for 192.168.0.101
Host is up (0.00029s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds
```

This happen because the TTL value for “basic scan” is less than 64. The firewall of the target machine will reject only TTL value equal to 64. When we captured the network traffic generated during this scan. We found that the basic scan used a TTL value of 56.



ip.addr == 192.168.0.101

```
No.  Time          Source         Destination    Protocol Length Info
 188 2.340466226   192.168.0.107  192.168.0.101  TCP      58 47283 → 22 [SYN]
 189 2.340717488   192.168.0.101  192.168.0.107  TCP      60 22 → 47283 [SYN,
 190 2.340733711   192.168.0.107  192.168.0.101  TCP      54 47283 → 22 [RST]
 191 2.340824327   192.168.0.101  192.168.0.107  ICMP     82 Destination unrea
 268 3.540735297   192.168.0.101  192.168.0.107  TCP      60 [TCP Retransmissi
 269 3.540764341   192.168.0.107  192.168.0.101  TCP      54 47283 → 22 [RST]
 270 3.541034763   192.168.0.101  192.168.0.107  ICMP     82 Destination unrea
```

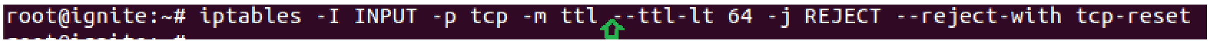
```
▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 44
  Identification: 0xfc86 (64646)
▸ Flags: 0x00
  Fragment offset: 0
  Time to live: 59
  Protocol: TCP (6)
  Header checksum: 0x0125 [validation disabled]
  [Header checksum status: Unverified]
  Source: 192.168.0.107
```

Strengthening Security: Blocking TTL 64 and Below

Now admin has added one more step of security to prevent his network from entire type scanning by rejecting TTL value of 64 and less than 64.

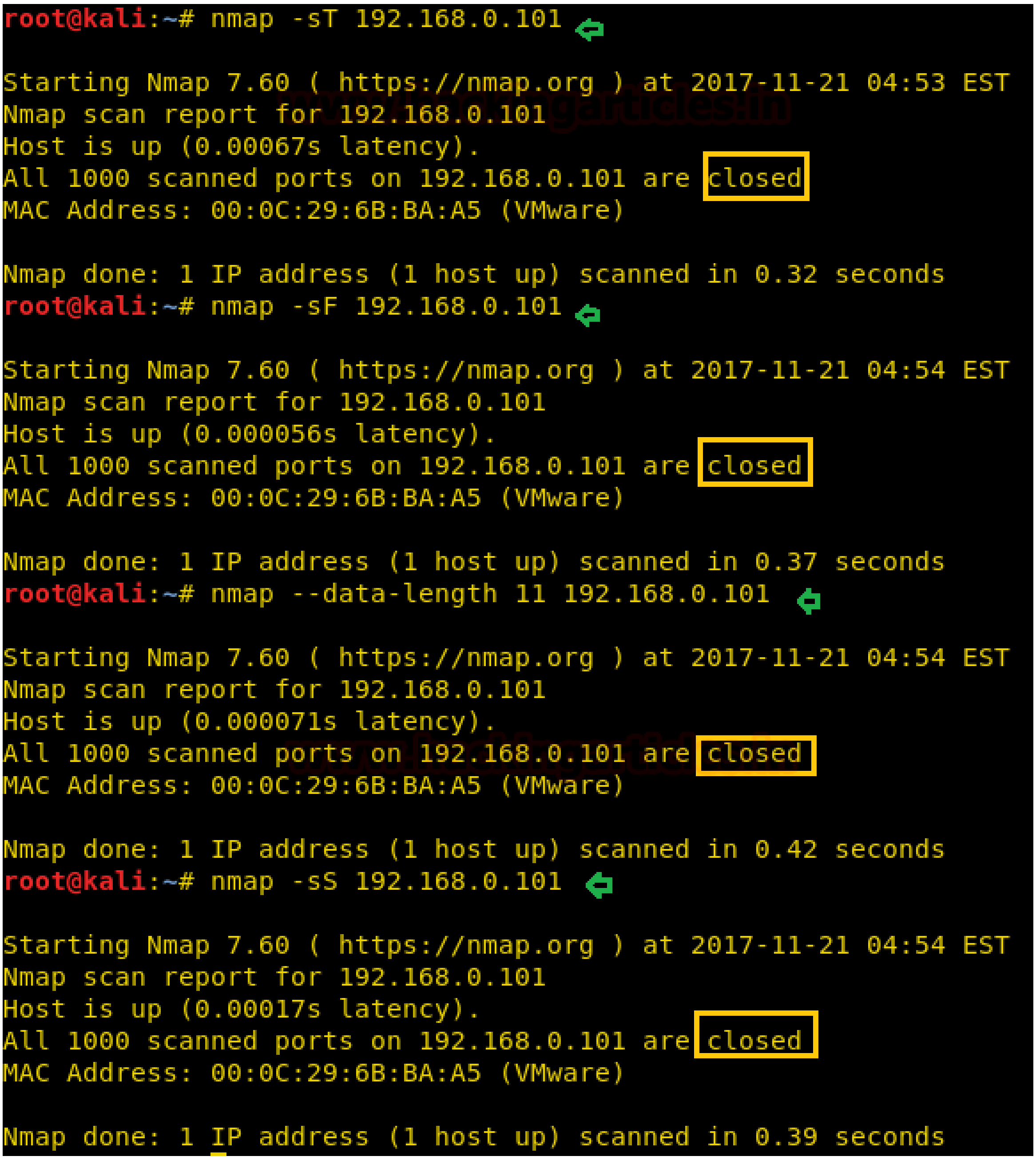
```
iptables -I INPUT -p tcp -m ttl --ttl-lt 64 -j REJECT --reject-with tcp-reset
```

Now firewall will analysis the traffic coming on his network and blocks the packet contains TTL 64 or less than it.



```
root@ignite:~# iptables -I INPUT -p tcp -m ttl --ttl-lt 64 -j REJECT --reject-with tcp-reset
```

Bravo!! Above firewall rule is more powerful than the previous rules because it completely blocks NMAP "basic scan" as well as "advance scan". If you notice the image given below, you will observe that TCP [sT], Fin Scan [sF], Data-length, and Stealth [sS] Scan all fail and show the port as closed.

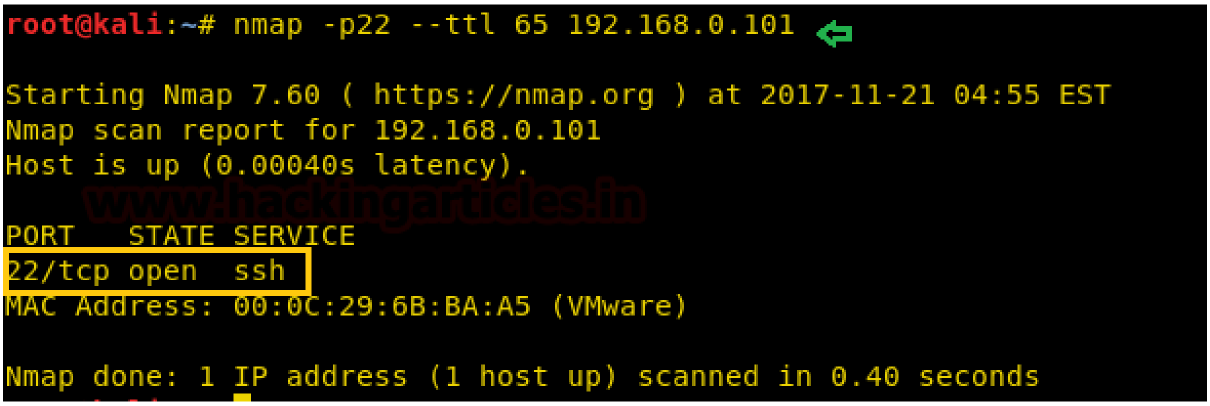


```
root@kali:~# nmap -sT 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:53 EST
Nmap scan report for 192.168.0.101
Host is up (0.00067s latency).
All 1000 scanned ports on 192.168.0.101 are closed
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
root@kali:~# nmap -sF 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:54 EST
Nmap scan report for 192.168.0.101
Host is up (0.000056s latency).
All 1000 scanned ports on 192.168.0.101 are closed
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
root@kali:~# nmap --data-length 11 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:54 EST
Nmap scan report for 192.168.0.101
Host is up (0.000071s latency).
All 1000 scanned ports on 192.168.0.101 are closed
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.42 seconds
root@kali:~# nmap -sS 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:54 EST
Nmap scan report for 192.168.0.101
Host is up (0.00017s latency).
All 1000 scanned ports on 192.168.0.101 are closed
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds
```

Still, there is a second way to enumerate port for an accurate result, by setting TTL value greater than 64. Following command will perform a port scan with defined TTL value i.e. 65 which will bypass firewall filter as 65 is greater than 64.

```
nmap -p22 --ttl 65 192.168.0.101
```

So if the attacker is lucky to guess rejected TTL value or firewall rule and applied correct TTL. Then only port enumeration will get successful as shown in given image port 22 is open.



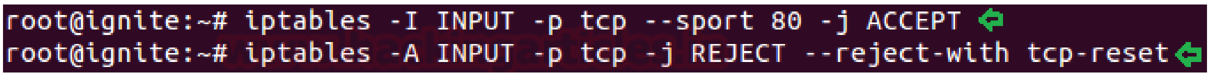
```
root@kali:~# nmap -p22 --ttl 65 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 04:55 EST
Nmap scan report for 192.168.0.101
Host is up (0.00040s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.40 seconds
```

Source Port Scan

Source Port Filter with IPTables

One more step to secure network from scanning is to apply firewall rule to allow traffic from a specific port only and reject traffic from remaining ports.

```
iptables -I INPUT -p tcp --sport 80 -j  ACCEPT
iptables -A INPUT -p tcp -j  REJECT --reject-with tcp-reset
```



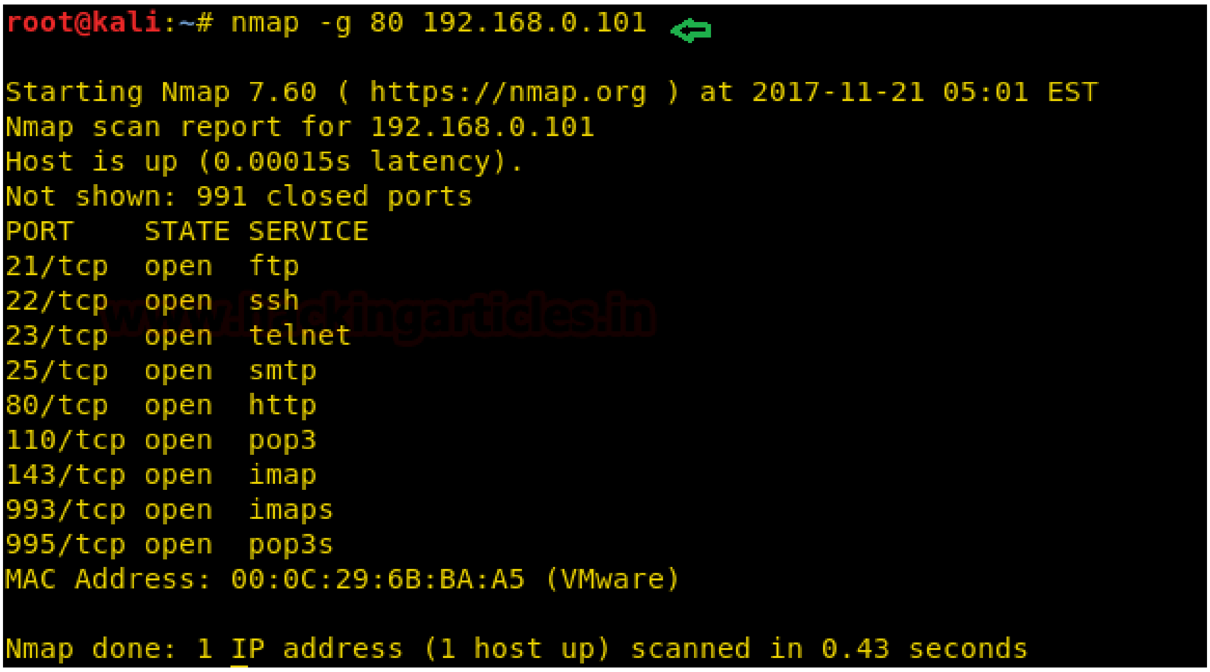
```
root@ignite:~# iptables -I INPUT -p tcp --sport 80 -j ACCEPT
root@ignite:~# iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset
```

Now again NMAP basic and advance will fail to enumerate open port state and if the attacker made a correct guess again firewall filter. Then he can execute NMAP source port scan to enumerate port details.

The option g defines the source port that will carry the network packet to the destination port.

```
nmap -g 80 192.168.0.101
```

Above command will send traffic from port 80 to perform scanning. Hence firewall will allow traffic from source port 80 and as a result show state for open ports.



```
root@kali:~# nmap -g 80 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 05:01 EST
Nmap scan report for 192.168.0.101
Host is up (0.00015s latency).
Not shown: 991 closed ports
PORT    STATE SERVICE
21/tcp  open  ftp
22/tcp  open  ssh
23/tcp  open  telnet
25/tcp  open  smtp
80/tcp  open  http
110/tcp open  pop3
143/tcp open  imap
993/tcp open  imaps
995/tcp open  pop3s
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

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

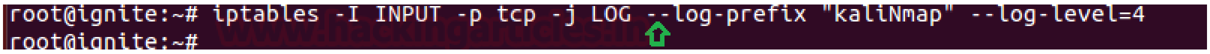
Decoy Scan

Set Firewall Log to capture Attacker IP

Admin can set a firewall rule to create Log for IP from which traffic is coming. It will only create system logs to capture the attacker IP who is performing scanning.

```
iptables -I INPUT -p tcp -j LOG --log-prefix "kaliNmap" --log-level=4
```

Now if the attacker will perform any type of network scanning on the targeted system. Then the firewall will generate its log which will capture his IP.



```
root@ignite:~# iptables -I INPUT -p tcp -j LOG --log-prefix "kaliNmap" --log-level=4
root@ignite:~#
```

Escape from the Firewall log

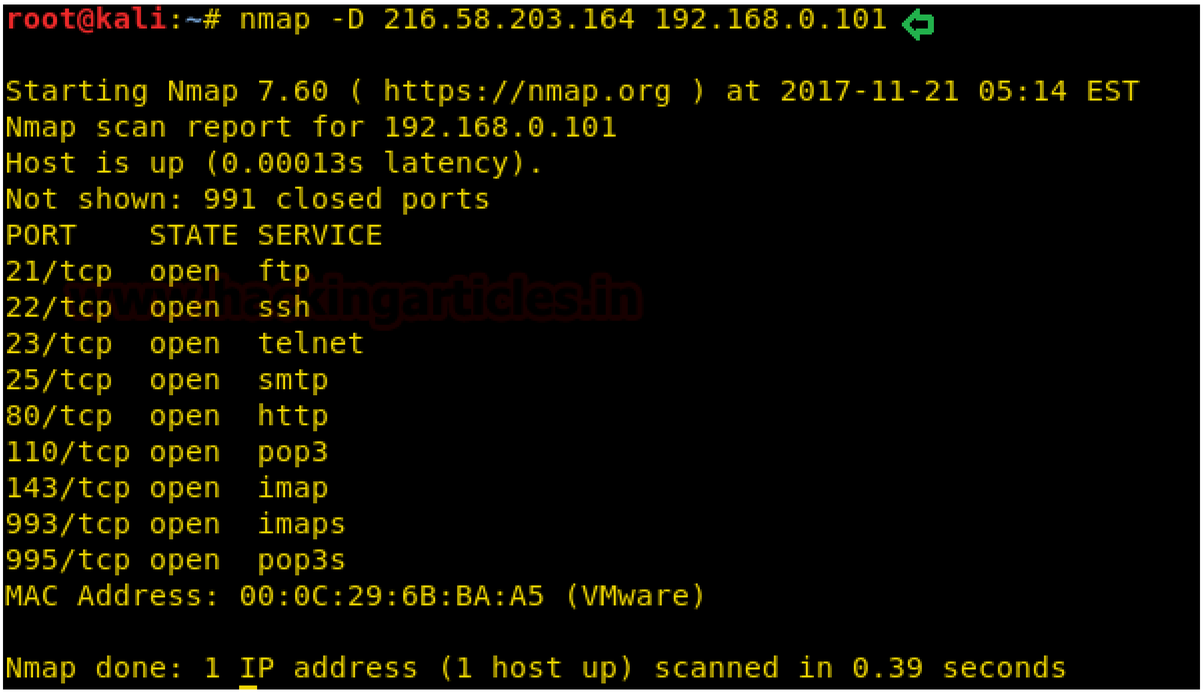
Always use some kind of precaution to escape yourself while performing network scanning. Because in windows “honey pot” and in Linux “iptables” are firewall will make the log of attacker’s IP. In such a situation, it is suggested that you use a Decoy Scan for port enumeration.

Decoy Scan

The -D option makes it look like the trick scanning the target network. It does not hide your own IP. But it makes your IP one of a torrent of others supposedly scanning the victim at the same time. This not only makes the scan look scarier. But reduces the chance of you being trace from your scan (difficult to tell which system is the "real" source).

```
nmap -D 216.58.203.164 192.168.0.101
```

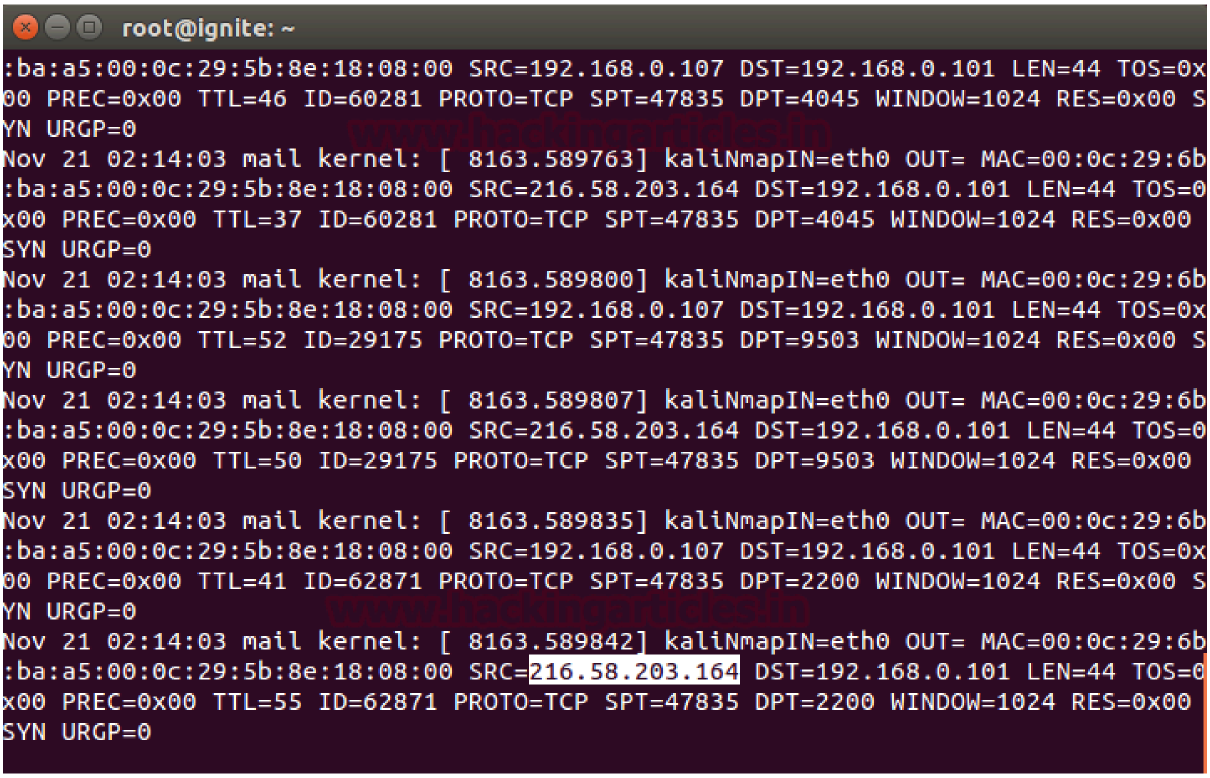
In the above command, we had to use Google IP as a torrent which will reflect as attacker IP in firewall log.



```
root@kali:~# nmap -D 216.58.203.164 192.168.0.101

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 05:14 EST
Nmap scan report for 192.168.0.101
Host is up (0.00013s latency).
Not shown: 991 closed ports
PORT    STATE SERVICE
21/tcp  open  ftp
22/tcp  open  ssh
23/tcp  open  telnet
25/tcp  open  smtp
80/tcp  open  http
110/tcp open  pop3
143/tcp open  imap
993/tcp open  imaps
995/tcp open  pop3s
MAC Address: 00:0C:29:6B:BA:A5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds
```

```
tail -f /var/log/syslog
```

When admin will read the system log. Then he will take higlighted IP as the attacker's IP and may apply the filter on this IP to block incoming traffic from it.



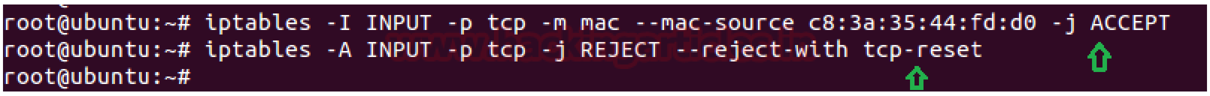
```
:ba:a5:00:0c:29:5b:8e:18:08:00 SRC=192.168.0.107 DST=192.168.0.101 LEN=44 TOS=0x
00 PREC=0x00 TTL=46 ID=60281 PROTO=TCP SPT=47835 DPT=4045 WINDOW=1024 RES=0x00 S
YN URGP=0
Nov 21 02:14:03 mail kernel: [ 8163.589763] kaliNmapIN=eth0 OUT= MAC=00:0c:29:6b
:ba:a5:00:0c:29:5b:8e:18:08:00 SRC=216.58.203.164 DST=192.168.0.101 LEN=44 TOS=0
x00 PREC=0x00 TTL=37 ID=60281 PROTO=TCP SPT=47835 DPT=4045 WINDOW=1024 RES=0x00
SYN URGP=0
Nov 21 02:14:03 mail kernel: [ 8163.589800] kaliNmapIN=eth0 OUT= MAC=00:0c:29:6b
:ba:a5:00:0c:29:5b:8e:18:08:00 SRC=192.168.0.107 DST=192.168.0.101 LEN=44 TOS=0x
00 PREC=0x00 TTL=52 ID=29175 PROTO=TCP SPT=47835 DPT=9503 WINDOW=1024 RES=0x00 S
YN URGP=0
Nov 21 02:14:03 mail kernel: [ 8163.589807] kaliNmapIN=eth0 OUT= MAC=00:0c:29:6b
:ba:a5:00:0c:29:5b:8e:18:08:00 SRC=216.58.203.164 DST=192.168.0.101 LEN=44 TOS=0
x00 PREC=0x00 TTL=50 ID=29175 PROTO=TCP SPT=47835 DPT=9503 WINDOW=1024 RES=0x00
SYN URGP=0
Nov 21 02:14:03 mail kernel: [ 8163.589835] kaliNmapIN=eth0 OUT= MAC=00:0c:29:6b
:ba:a5:00:0c:29:5b:8e:18:08:00 SRC=192.168.0.107 DST=192.168.0.101 LEN=44 TOS=0x
00 PREC=0x00 TTL=41 ID=62871 PROTO=TCP SPT=47835 DPT=2200 WINDOW=1024 RES=0x00 S
YN URGP=0
Nov 21 02:14:03 mail kernel: [ 8163.589842] kaliNmapIN=eth0 OUT= MAC=00:0c:29:6b
:ba:a5:00:0c:29:5b:8e:18:08:00 SRC=216.58.203.164 DST=192.168.0.101 LEN=44 TOS=0
x00 PREC=0x00 TTL=55 ID=62871 PROTO=TCP SPT=47835 DPT=2200 WINDOW=1024 RES=0x00
SYN URGP=0
```

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.

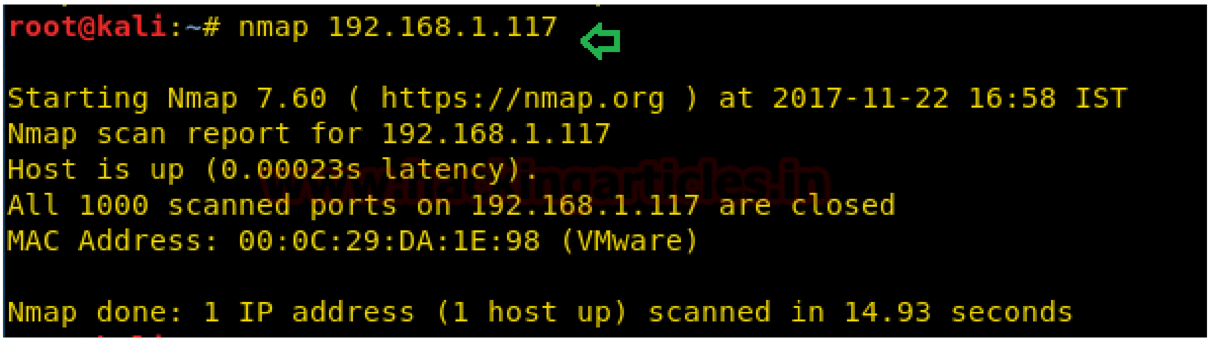
```
iptables -I INPUT -p tcp -m mac --mac-source "AA:AA:AA:AA:AA:AA" -j ACCEPT
iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset
```



```
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:~#
```

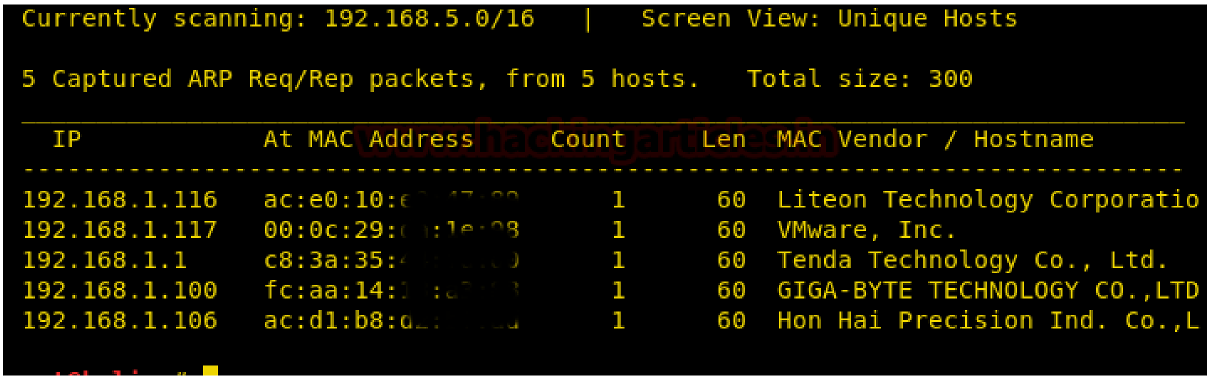
Now when the attacker will perform basic network scanning on the target's network, he could not able to enumerate ports and running service of a victim's system.

```
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.



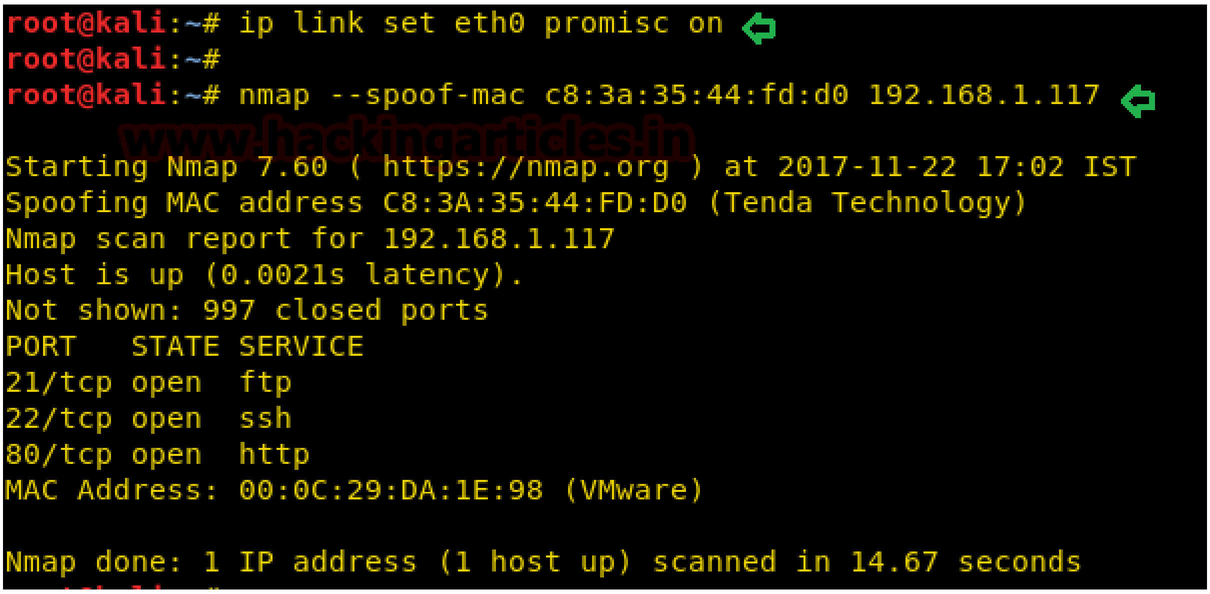
```
 Currently scanning: 192.168.5.0/16   |   Screen View: Unique Hosts

 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]:1e:98       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.



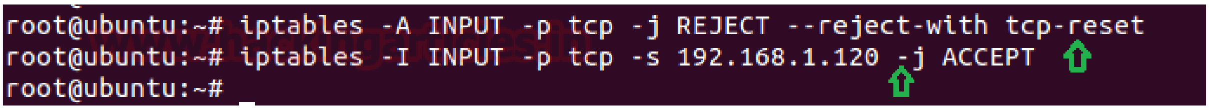
```
root@kali:~# ip link set eth0 promisc on
root@kali:~#
root@kali:~# nmap --spoof-mac c8:3a:35:44:fd:d0 192.168.1.117

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.

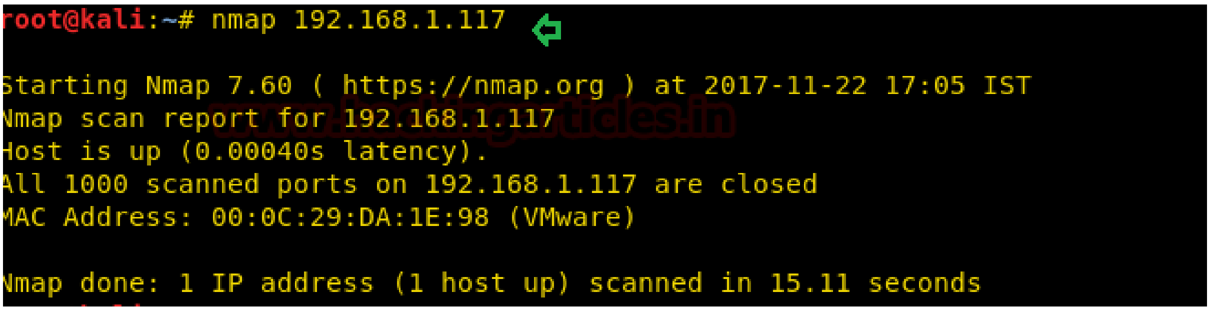
```
iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset
iptables -I INPUT -p tcp -s 192.168.1.120 -j ACCEPT
```



```
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
```

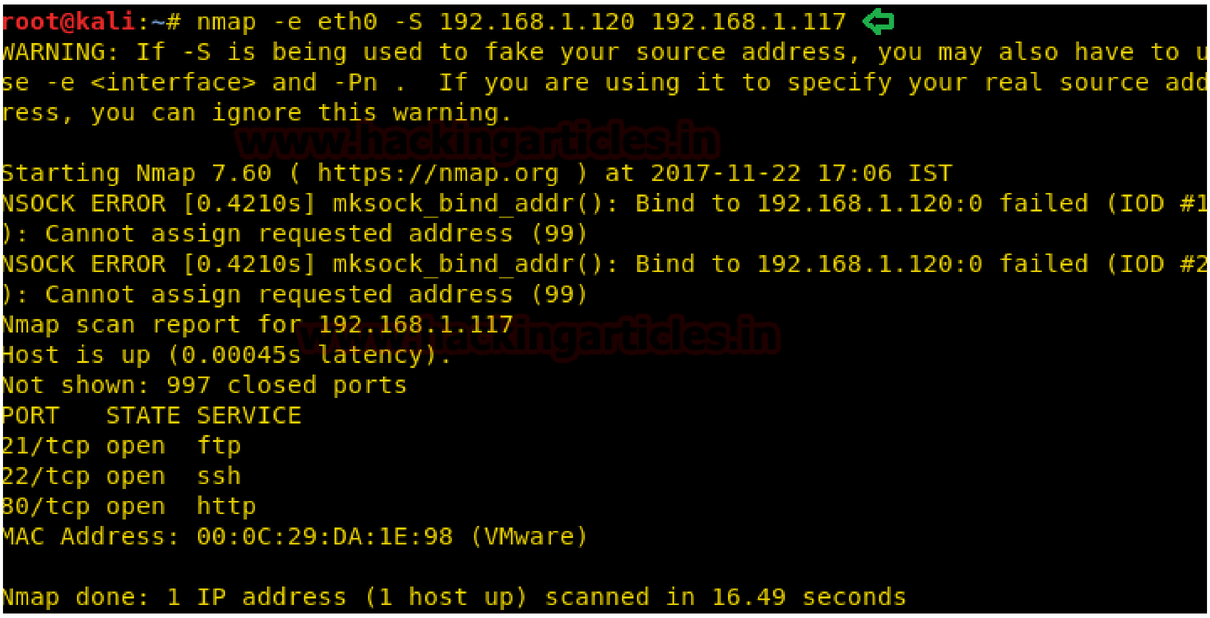
In order to bypass above applied filter attacker may again run **netdiscover command** or **nmap Host Scan** in Kali Linux terminal to identify the active host in the network. As a result he will get a table which contains MAC address and IP address of the active host in the local network.

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
```

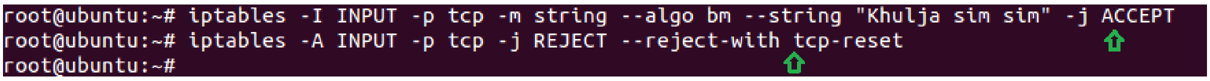
Data-String Scan

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
```

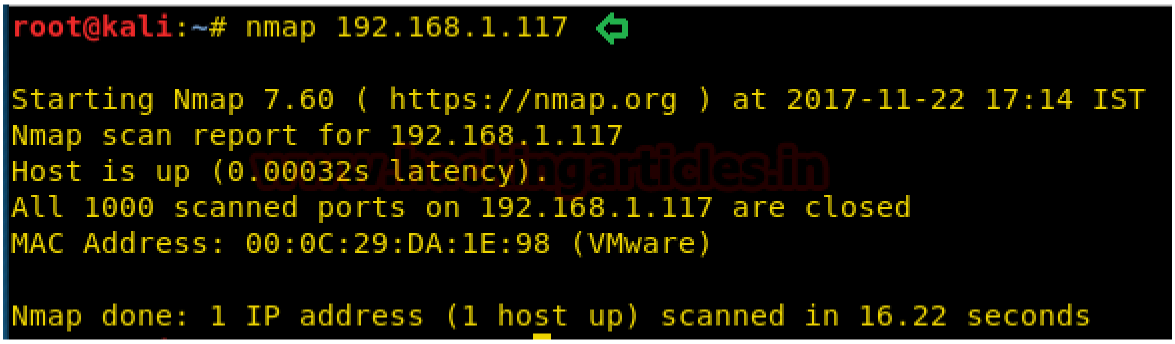
In above rule, you can see we had used **"Khulja sim sim"** as a special string to establish TCP connection. Hence only those TCP connection could be established which contain "Khulja sim sim" in packets.



```
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
```



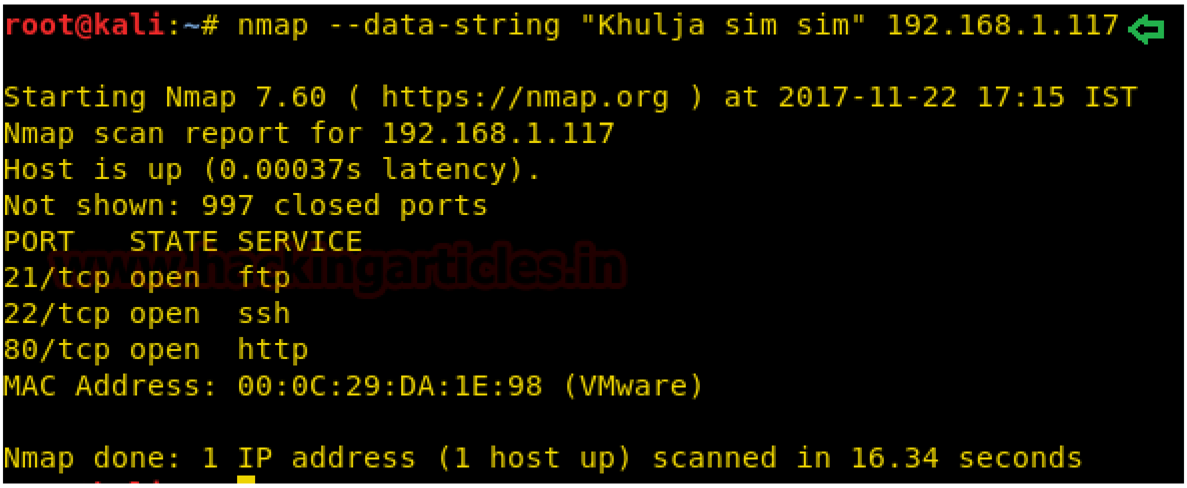
```
root@kali:~# nmap 192.168.1.117

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
```

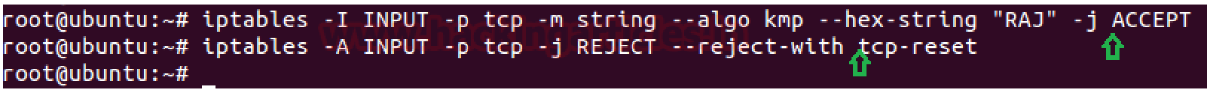
Hex String Scan

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
```

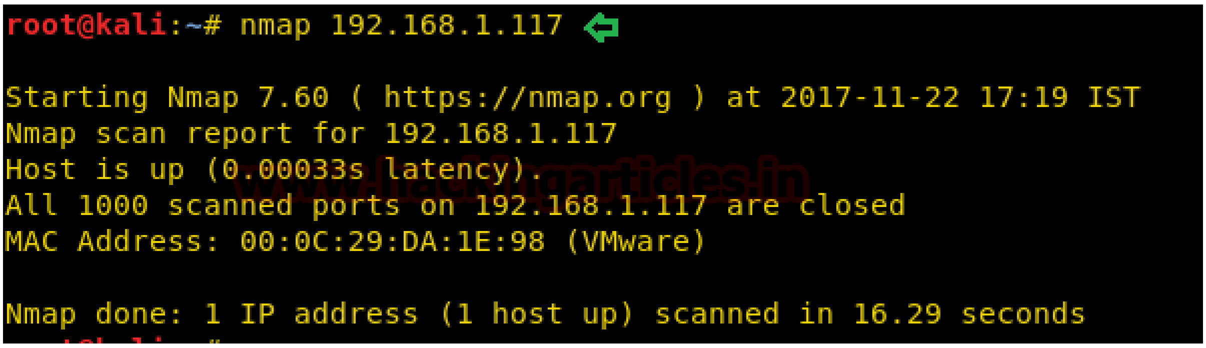
In the above rule, you can see we had used hex value for **"RAJ"** as a special string to establish a TCP connection. Hence only those TCP connection could be established which contain hex value of "RAJ" in the packet.



```
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
```



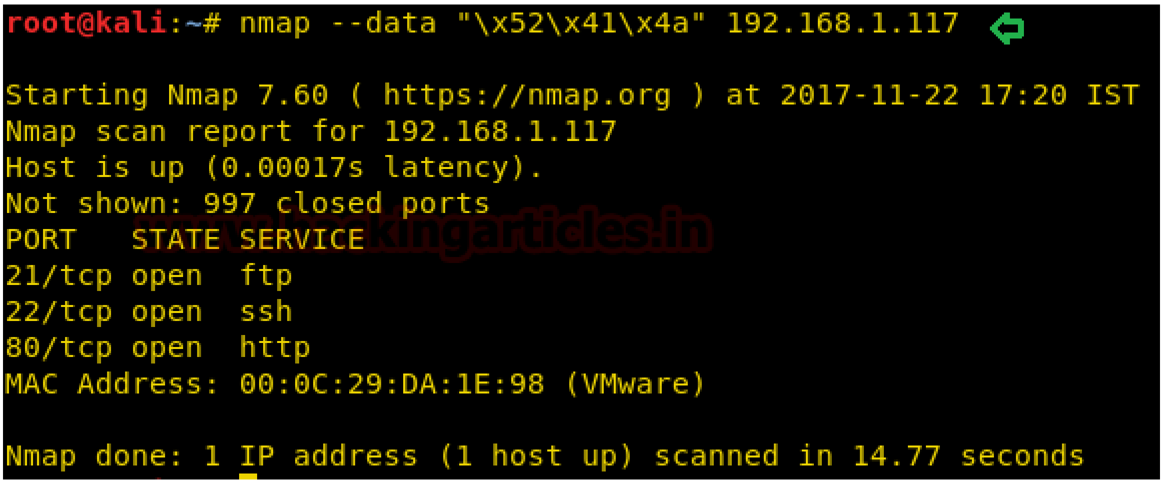
```
root@kali:~# nmap 192.168.1.117

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.



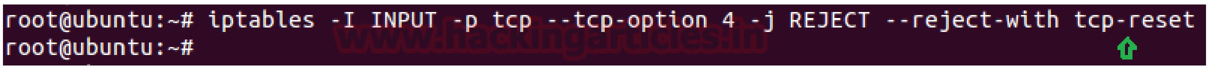
```
root@kali:~# nmap --data "\x52\x41\x4a" 192.168.1.117

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:

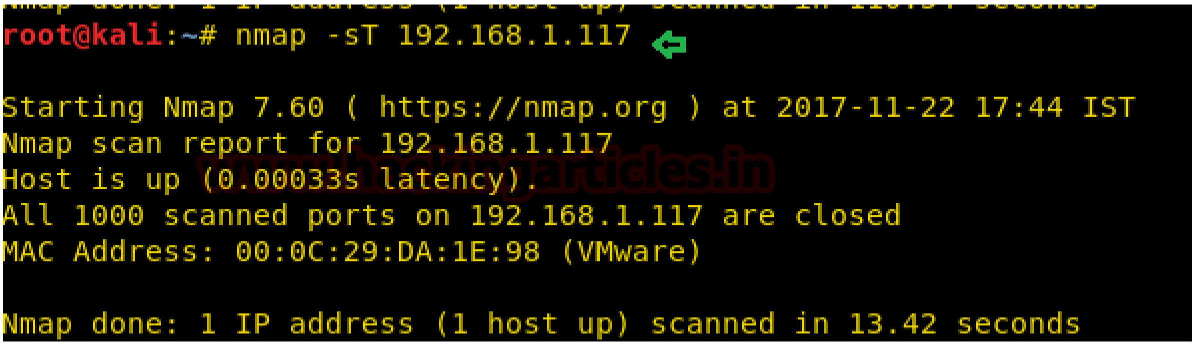
```
iptables -I INPUT -p tcp --tcp-option 4  -j REJECT --reject-with tcp-reset
```



```
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
```

IP Options in Nmap

The IP protocol provides numerous options that can be placed in packet headers. Contrasting the omnipresent TCP options, users seldom observe IP options 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. 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

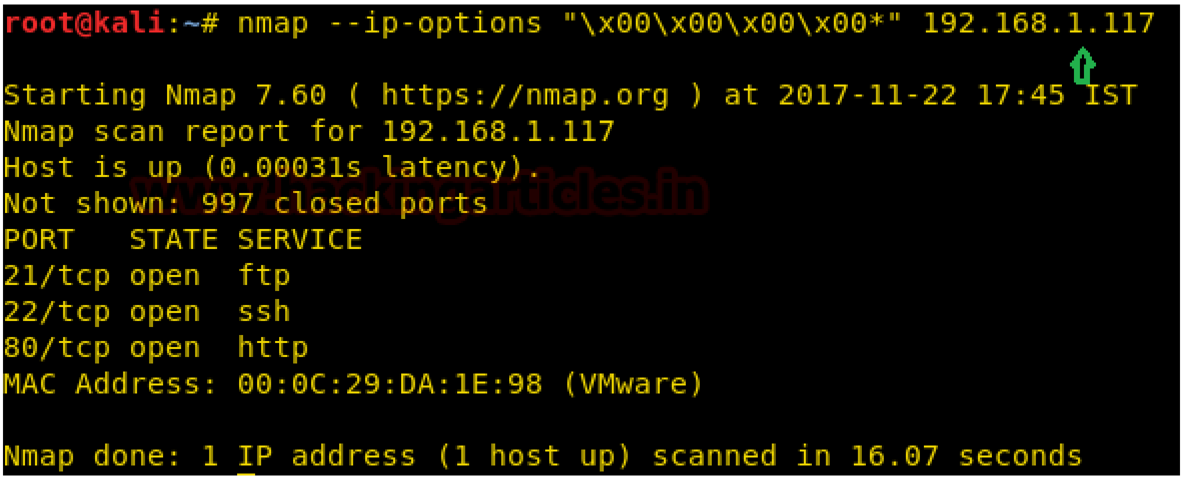
Now type the following command with the ip-option argument as shown below:

```
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

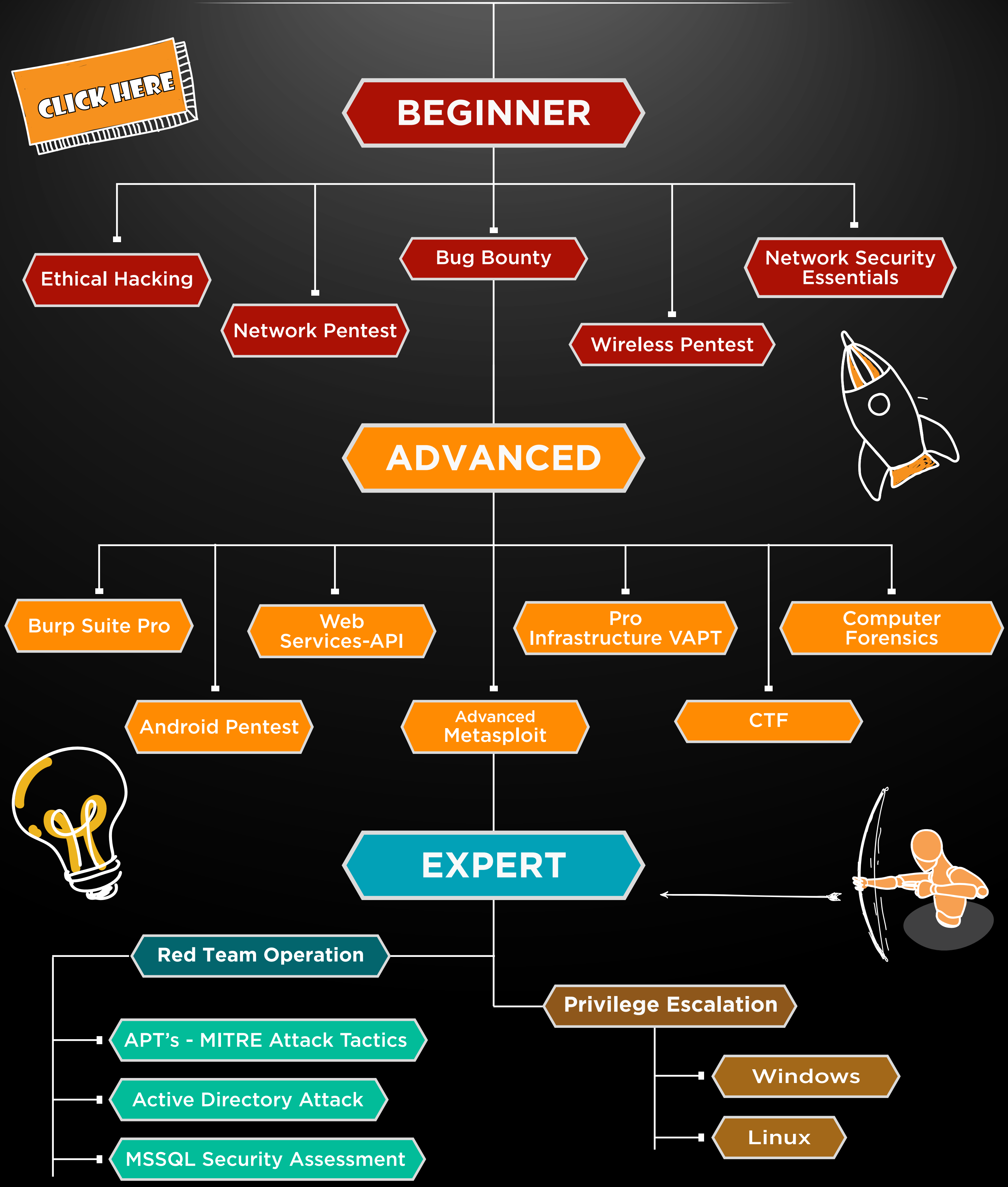


```
root@kali:~# nmap --ip-options "\x00\x00\x00\x00*" 192.168.1.117

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
```

To learn more on Nmap. Follow this **Link.**

iGNITE Technologies

JOIN OUR TRAINING PROGRAMS



CLICK HERE

BEGINNER

- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials

ADVANCED

- Burp Suite Pro
- Android Pentest
- Web Services-API
- Advanced Metasploit
- Pro Infrastructure VAPT
- CTF
- Computer Forensics

EXPERT

- Red Team Operation
  - APT's - MITRE Attack Tactics
  - Active Directory Attack
  - MSSQL Security Assessment
- Privilege Escalation
  - Windows
  - Linux

www.ignitetechnologies.in