Port Scanning using Metasploit with IPTables

March 5, 2018 By Raj Chandel

Scanning port is a technique used by penetration tester for identifying the state of computer network services associated with the particular port number. For example, port 80 is available for HTTP service and port 22 is available for SSH service. We suggest using Nmap for enumerating port state, for best practice click **here** and learn Nmap working in detail.

Moreover, Metasploit also serves port scanning for enumerating computer network services and make it easier as compare to Nmap.

Let's start!!

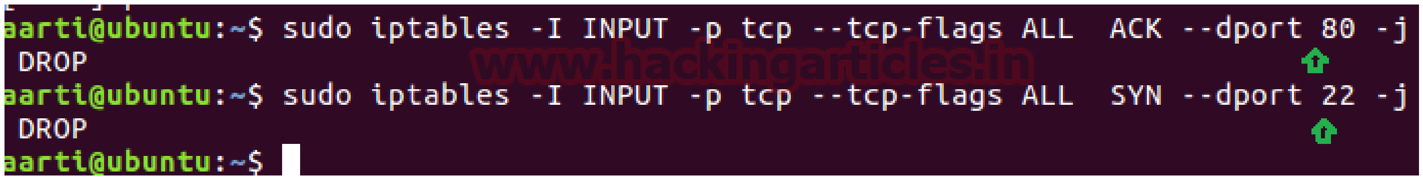
Requirement

Attacker: Kali Linux (192.168.1.103)

Target: Ubuntu (192.168.1.105)

Open the terminal and add given below iptables rules for incoming packet traffic in target's network which will drop the tcp ACK packet on port 80 and SYN packet on port 22 respectively.

```
sudo iptables -I INPUT -p tcp --tcp-flags ALL ACK --dport 80 -j DROP
sudo iptables -I INPUT -p tcp --tcp-flags ALL SYN --dport 22 -j DROP
```



```
aarti@ubuntu:~$ sudo iptables -I INPUT -p tcp --tcp-flags ALL  ACK --dport 80 -j
 DROP
aarti@ubuntu:~$ sudo iptables -I INPUT -p tcp --tcp-flags ALL  SYN --dport 22 -j
 DROP
aarti@ubuntu:~$
```

ACK Scan

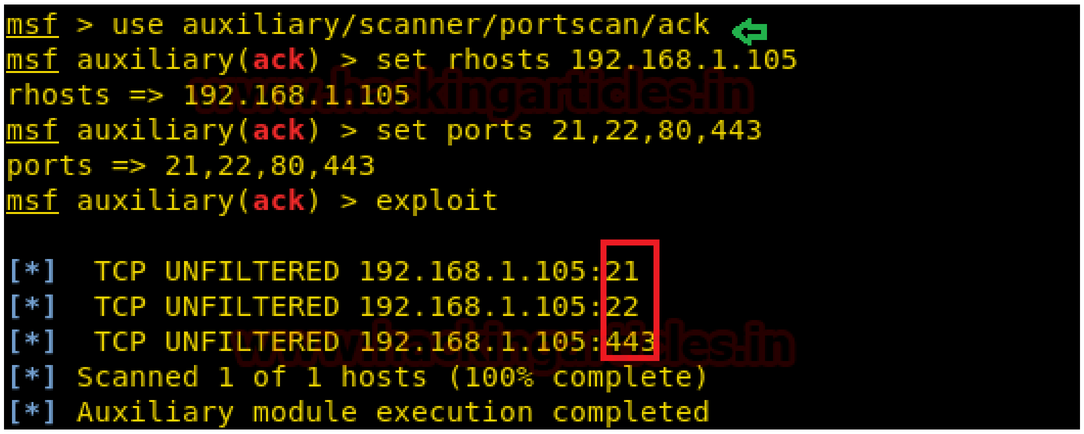
Now open the terminal in your Kali Linux and **type msfconsole** to load Metasploit framework and execute given below auxiliary command to run the specific module.

This module will Map out firewall rulesets with a raw ACK scan. Any unfiltered ports found means a stateful firewall is not in place for them.

Now as specified in given below command this module will send ack packet on port 21, 22, 80,443 to enumerate state of the firewall for these ports. If it receives reset packet as a reply from destination port then it will display **unfiltered state** for that particular port and if does not receive reset packet from destination port then it will not show any comment for that particular port which means the port is protected by the firewall.

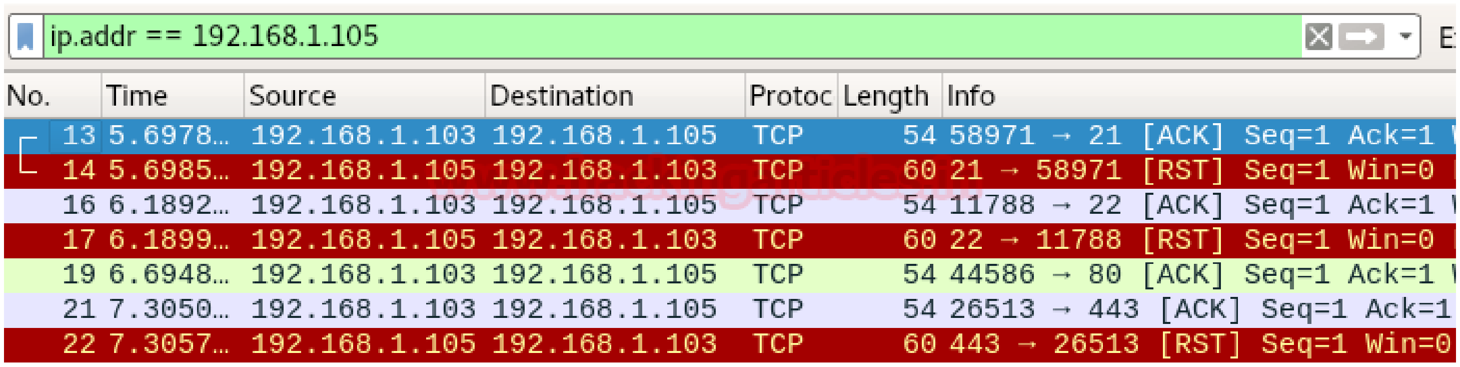
```
use auxiliary/scanner/portscan/ack
msf auxiliary(ack) > set rhosts 192.168.1.105
msf auxiliary(ack) > set ports 21,22,80,443
msf auxiliary(ack) >exploit
```

From given below image you can observe that it is showing **TCP unfiltered** for **port 21,22,443** and did not comment for port 80 hence port 80 is filtered. This scan can be only used for identifying the state of the firewall in terms of port filter or unfiltered.



```
msf > use auxiliary/scanner/portscan/ack
msf auxiliary(ack) > set rhosts 192.168.1.105
rhosts => 192.168.1.105
msf auxiliary(ack) > set ports 21,22,80,443
ports => 21,22,80,443
msf auxiliary(ack) > exploit

[*]  TCP UNFILTERED 192.168.1.105:21
[*]  TCP UNFILTERED 192.168.1.105:22
[*]  TCP UNFILTERED 192.168.1.105:443
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

We had used Wireshark for demonstrating ack scan and here you can observe that port 80 doesn't reply with RST packet which means ack packet for port 80 has been blocked by the network administrator.



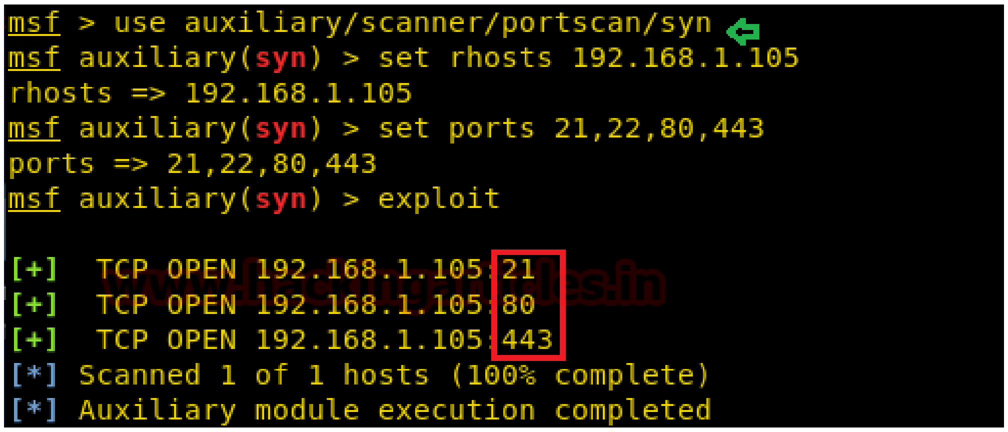
```
ip.addr == 192.168.1.105

No.  Time      Source         Destination    Protoc Length Info
  13 5.6978… 192.168.1.103 192.168.1.105 TCP        54 58971 → 21 [ACK] Seq=1 Ack=1
  14 5.6985… 192.168.1.105 192.168.1.103 TCP        60 21 → 58971 [RST] Seq=1 Win=0
  16 6.1892… 192.168.1.103 192.168.1.105 TCP        54 11788 → 22 [ACK] Seq=1 Ack=1
  17 6.1899… 192.168.1.105 192.168.1.103 TCP        60 22 → 11788 [RST] Seq=1 Win=0
  19 6.6948… 192.168.1.103 192.168.1.105 TCP        54 44586 → 80 [ACK] Seq=1 Ack=1
  21 7.3050… 192.168.1.103 192.168.1.105 TCP        54 26513 → 443 [ACK] Seq=1 Ack=1
  22 7.3057… 192.168.1.105 192.168.1.103 TCP        60 443 → 26513 [RST] Seq=1 Win=0
```

SYN Scan

This module enumerates open TCP services using a raw SYN scan, the here syn packet will be sent on port 21, 22, 80,443 to enumerate state open/closed for these ports. If it receives syn, ack packet as a reply from destination port then it will display **OPEN state** for that particular port and if does not receives syn, ack packet from destination port then it will not show any comment for that particular port which indicates **filtered** or **Closed state** for that particular port.

```
use auxiliary/scanner/portscan/syn
msf auxiliary(syn) > set rhosts 192.168.1.105
msf auxiliary(syn) > set ports 21,22,80,443
msf auxiliary(syn) >exploit
```

From given below image you can observe that it is showing **TCP OPEN** for **port 21,80,443** and did not comment for port 22 hence port 22 is filtered or closed.

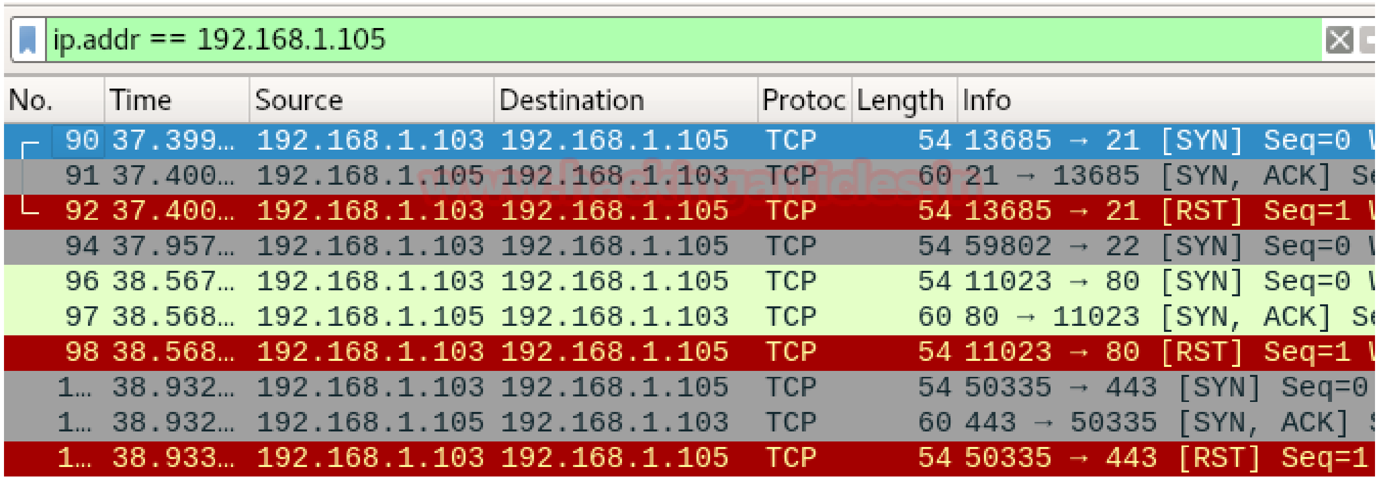


```
msf > use auxiliary/scanner/portscan/syn
msf auxiliary(syn) > set rhosts 192.168.1.105
rhosts => 192.168.1.105
msf auxiliary(syn) > set ports 21,22,80,443
ports => 21,22,80,443
msf auxiliary(syn) > exploit

[+]  TCP OPEN 192.168.1.105:21
[+]  TCP OPEN 192.168.1.105:80
[+]  TCP OPEN 192.168.1.105:443
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Again we had used Wireshark for demonstrating syn scan and here you can observe that port 22 doesn't reply with SYN, ACK packets which mean SYN packet for port 22 has been blocked by the network administrator.

Moreover, you can observe the following packet communication between the source and destination port.

- Source port sends SYN packet to the destination port
- Source port receives SYN, ACK packet from the destination port
- Source port sends RST packet to the destination port



ip.addr == 192.168.1.105

No.	Time	Source	Destination	Protoc	Length	Info
90	37.399…	192.168.1.103	192.168.1.105	TCP	54	13685 → 21 [SYN] Seq=0 W
91	37.400…	192.168.1.105	192.168.1.103	TCP	60	21 → 13685 [SYN, ACK] Se
92	37.400…	192.168.1.103	192.168.1.105	TCP	54	13685 → 21 [RST] Seq=1 W
94	37.957…	192.168.1.103	192.168.1.105	TCP	54	59802 → 22 [SYN] Seq=0 W
96	38.567…	192.168.1.103	192.168.1.105	TCP	54	11023 → 80 [SYN] Seq=0 W
97	38.568…	192.168.1.105	192.168.1.103	TCP	60	80 → 11023 [SYN, ACK] Se
98	38.568…	192.168.1.103	192.168.1.105	TCP	54	11023 → 80 [RST] Seq=1 W
1…	38.932…	192.168.1.103	192.168.1.105	TCP	54	50335 → 443 [SYN] Seq=0
1…	38.932…	192.168.1.105	192.168.1.103	TCP	60	443 → 50335 [SYN, ACK] S
1…	38.933…	192.168.1.103	192.168.1.105	TCP	54	50335 → 443 [RST] Seq=1

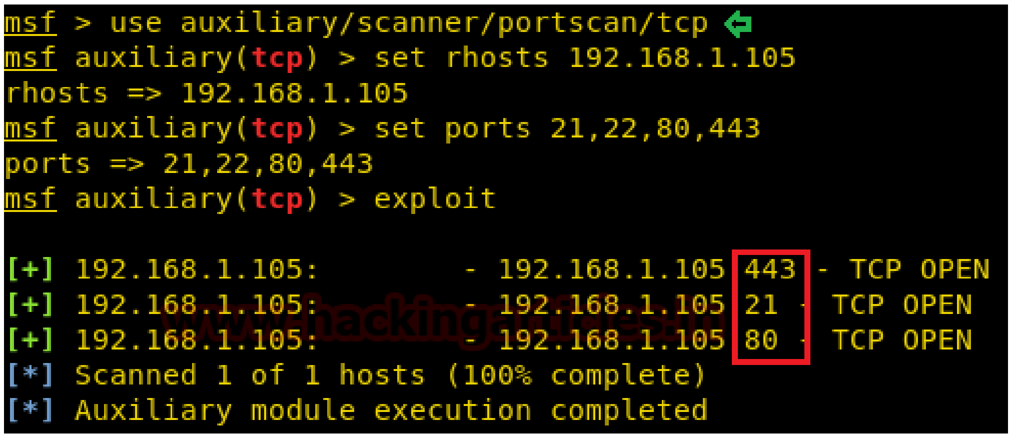
TCP Scan

Enumerate open TCP services by performing a full TCP connect on each port. This does not need administrative privileges on the source machine, which may be useful if pivoting.

```
use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > set rhosts 192.168.1.105
msf auxiliary(tcp) > set ports 21,22,80,443
msf auxiliary(tcp) >exploit
```

This scan is similar as SYN scan only the difference is that it follows TCP full communication i.e. 4-way handshake and SYN scan is followed half TCP communication.

From given below image you can observe that it is showing **TCP OPEN** for **port 21,80,443** and did not comment for port 22 hence **port 22 is filtered or closed**.

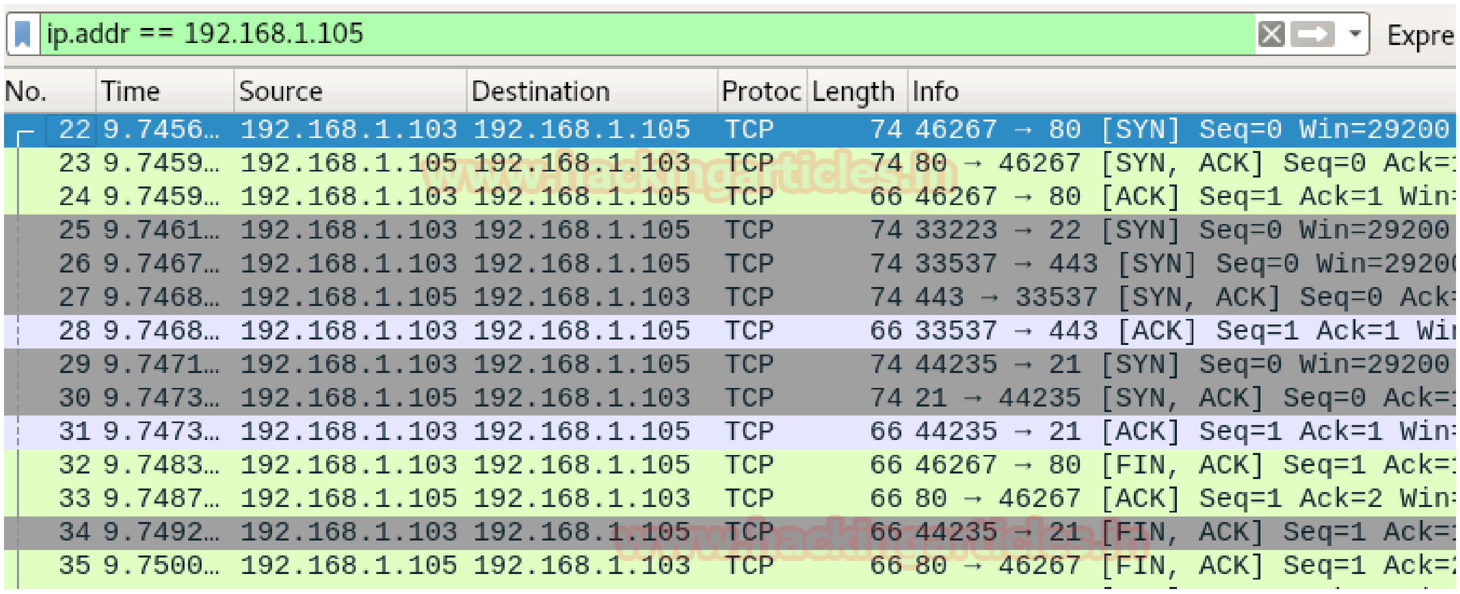


```
msf > use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > set rhosts 192.168.1.105
rhosts => 192.168.1.105
msf auxiliary(tcp) > set ports 21,22,80,443
ports => 21,22,80,443
msf auxiliary(tcp) > exploit

[+] 192.168.1.105:        - 192.168.1.105:443 - TCP OPEN
[+] 192.168.1.105:        - 192.168.1.105:21 - TCP OPEN
[+] 192.168.1.105:        - 192.168.1.105:80 - TCP OPEN
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Here you can observe that port 22 doesn't reply with SYN, ACK packets which mean SYN packet for port 22 has been blocked by the network administrator.

Moreover, you can observe the following packet communication between the source and destination port.

- Source port sends SYN packet to the destination port
- Source port receives SYN, ACK packet from the destination port
- Source port sends ACK packet to the destination port
- Source port sends FIN, ACK packet to the destination port


```
ip.addr == 192.168.1.105                                              Expre
No. Time      Source        Destination   Protoc Length Info
 22 9.7456… 192.168.1.103 192.168.1.105  TCP     74 46267 → 80 [SYN] Seq=0 Win=29200
 23 9.7459… 192.168.1.105 192.168.1.103  TCP     74 80 → 46267 [SYN, ACK] Seq=0 Ack=
 24 9.7459… 192.168.1.103 192.168.1.105  TCP     66 46267 → 80 [ACK] Seq=1 Ack=1 Win
 25 9.7461… 192.168.1.103 192.168.1.105  TCP     74 33223 → 22 [SYN] Seq=0 Win=29200
 26 9.7467… 192.168.1.103 192.168.1.105  TCP     74 33537 → 443 [SYN] Seq=0 Win=2920
 27 9.7468… 192.168.1.105 192.168.1.103  TCP     74 443 → 33537 [SYN, ACK] Seq=0 Ack
 28 9.7468… 192.168.1.103 192.168.1.105  TCP     66 33537 → 443 [ACK] Seq=1 Ack=1 Wi
 29 9.7471… 192.168.1.103 192.168.1.105  TCP     74 44235 → 21 [SYN] Seq=0 Win=29200
 30 9.7473… 192.168.1.105 192.168.1.103  TCP     74 21 → 44235 [SYN, ACK] Seq=0 Ack=
 31 9.7473… 192.168.1.103 192.168.1.105  TCP     66 44235 → 21 [ACK] Seq=1 Ack=1 Win
 32 9.7483… 192.168.1.103 192.168.1.105  TCP     66 46267 → 80 [FIN, ACK] Seq=1 Ack=
 33 9.7487… 192.168.1.105 192.168.1.103  TCP     66 80 → 46267 [ACK] Seq=1 Ack=2 Win
 34 9.7492… 192.168.1.103 192.168.1.105  TCP     66 44235 → 21 [FIN, ACK] Seq=1 Ack=
 35 9.7500… 192.168.1.105 192.168.1.103  TCP     66 80 → 46267 [FIN, ACK] Seq=1 Ack=
```

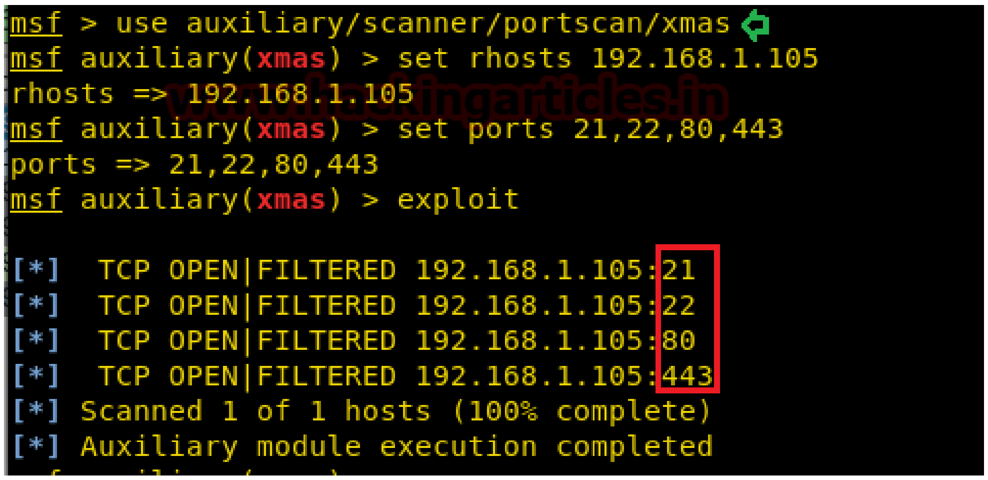
XMAS Scan

Enumerate open|filtered TCP services using a raw “Xmas” scan; this sends probes containing the FIN, PSH, and URG flags.

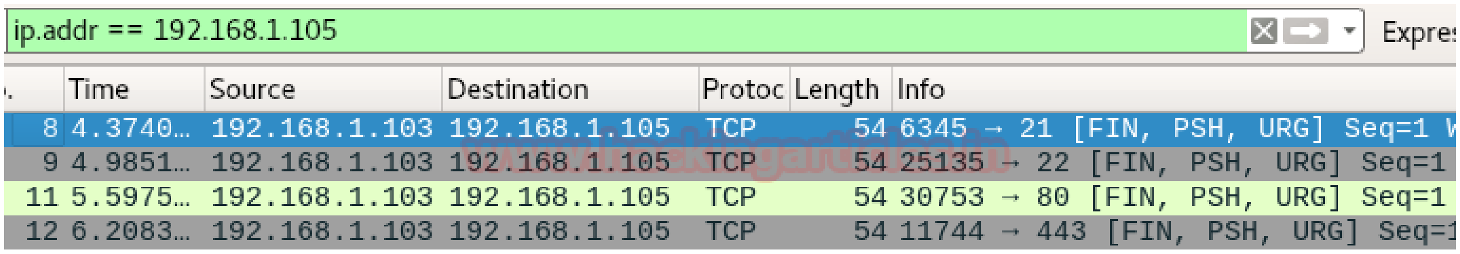
Instead of using TCP 3-way handshake communication this scan uses other tcp flags for TCP communication for enumerating state of ports.

```
use auxiliary/scanner/portscan/xmas
msf auxiliary(xmas) > set rhosts 192.168.1.105
msf auxiliary(xmas) > set ports 21,22,80,443
msf auxiliary(xmas) >exploit
```

From given below image you can observe that, this time it has shown **TCP OPEN| FILTERED** for all ports i.e. **21,22,80,443**


```
msf > use auxiliary/scanner/portscan/xmas
msf auxiliary(xmas) > set rhosts 192.168.1.105
rhosts => 192.168.1.105
msf auxiliary(xmas) > set ports 21,22,80,443
ports => 21,22,80,443
msf auxiliary(xmas) > exploit

[*]  TCP OPEN|FILTERED 192.168.1.105:21
[*]  TCP OPEN|FILTERED 192.168.1.105:22
[*]  TCP OPEN|FILTERED 192.168.1.105:80
[*]  TCP OPEN|FILTERED 192.168.1.105:443
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

If you notice given below image here source port sends FIN, PUSH and URG packets to the destination and destination port didn’t send any reply to source port which indicates above specified port are open and if any destination port sends RST, ACK packet to source port then it indicated that particular port is closed.



```
ip.addr == 192.168.1.105                                          Expres
.   Time       Source         Destination    Protoc Length Info
  8 4.3740…   192.168.1.103  192.168.1.105  TCP        54 6345 → 21 [FIN, PSH, URG] Seq=1 W
  9 4.9851…   192.168.1.103  192.168.1.105  TCP        54 25135 → 22 [FIN, PSH, URG] Seq=1
 11 5.5975…   192.168.1.103  192.168.1.105  TCP        54 30753 → 80 [FIN, PSH, URG] Seq=1
 12 6.2083…   192.168.1.103  192.168.1.105  TCP        54 11744 → 443 [FIN, PSH, URG] Seq=1
```