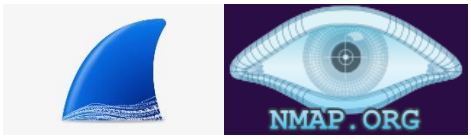
WIRESHARK ANALYSIS OF VARIOUS NMAP SCANS



Author Name: Aditya Srivastava
Twitter: https://twitter.com/evilcghost
LinkedIn: https://www.linkedin.com/in/adityasrivastava0703/

Basic Scan Types (-sT and -sS)

The two basic scan types used most in Nmap are TCP connect() scanning [-sT] and SYN scanning (also known as half-open, or stealth scanning) [-sS]

TCP connect() SCAN (-sT)

TCP connect scan will scan for TCP port like 21,22,23,445 etc. and ensures that ports are available for connection through a 3-way handshake connection between the source and destination.

This scan is very efficient, but the drawback is that it is very easy to detect due to modern firewalls or Intrusion detection system being present on server/victims' side.

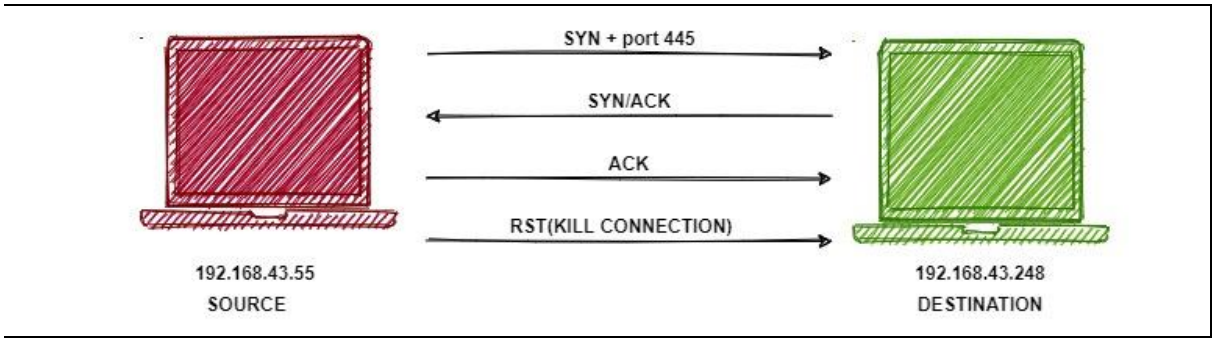
Advantage:

1) Results are highly accurate.
2) Handshake is complete, that ensures secure communication.

Disadvantage:

1) Very noisy, easy to detect.

For open ports



This involves mainly 4 steps - SYN, SYN/ACK, ACK and RST.
First 3 steps complete the handshake while the fourth one resets the connection.
We will try to understand what happens at the data packet level and see how this scan works.
Let's scan a port which is open and at the same time capture the packets in Wireshark for analysis.

Syntax

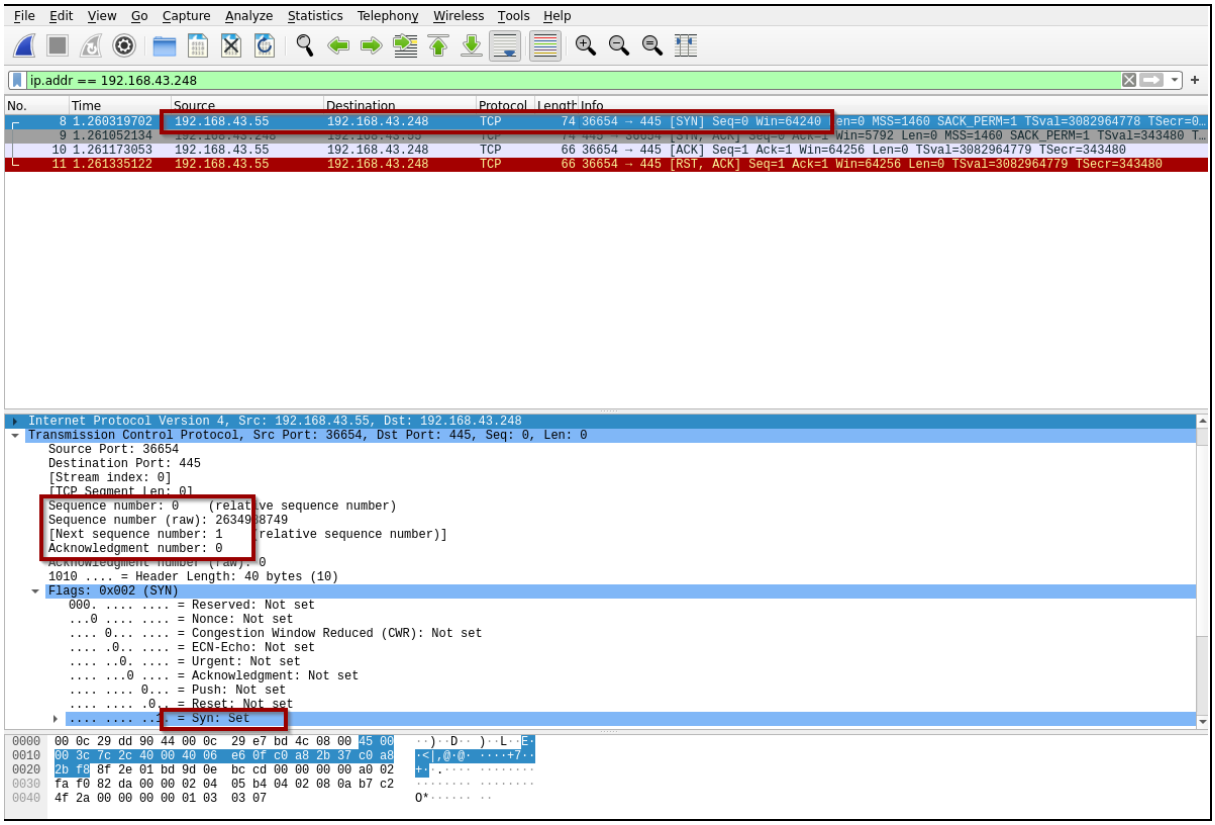
nmap -sT -p <port number> <destination IP>

Scan command

nmap -sT -p 445 192.168.43.248

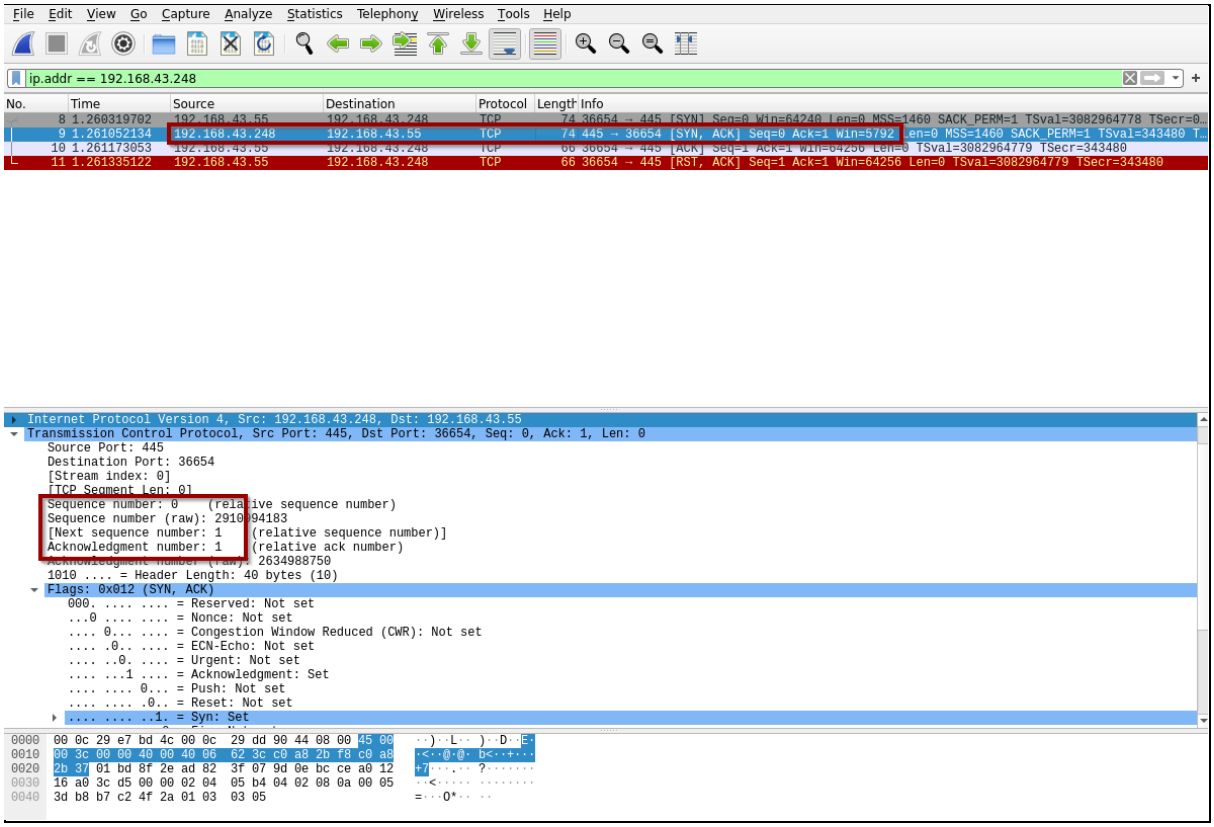
```
kali@kali:~$ sudo nmap -sT -p 445 192.168.43.248
[sudo] password for kali:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-19 19:46 IST
Nmap scan report for 192.168.43.248
Host is up (0.00088s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:DD:90:44 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.41 seconds
kali@kali:~$
```

Nmap tries the 3-way handshake and once the handshake is successful it declares the port is open.
Let us see what happened in the background when we ran TCP connect scan using nmap.

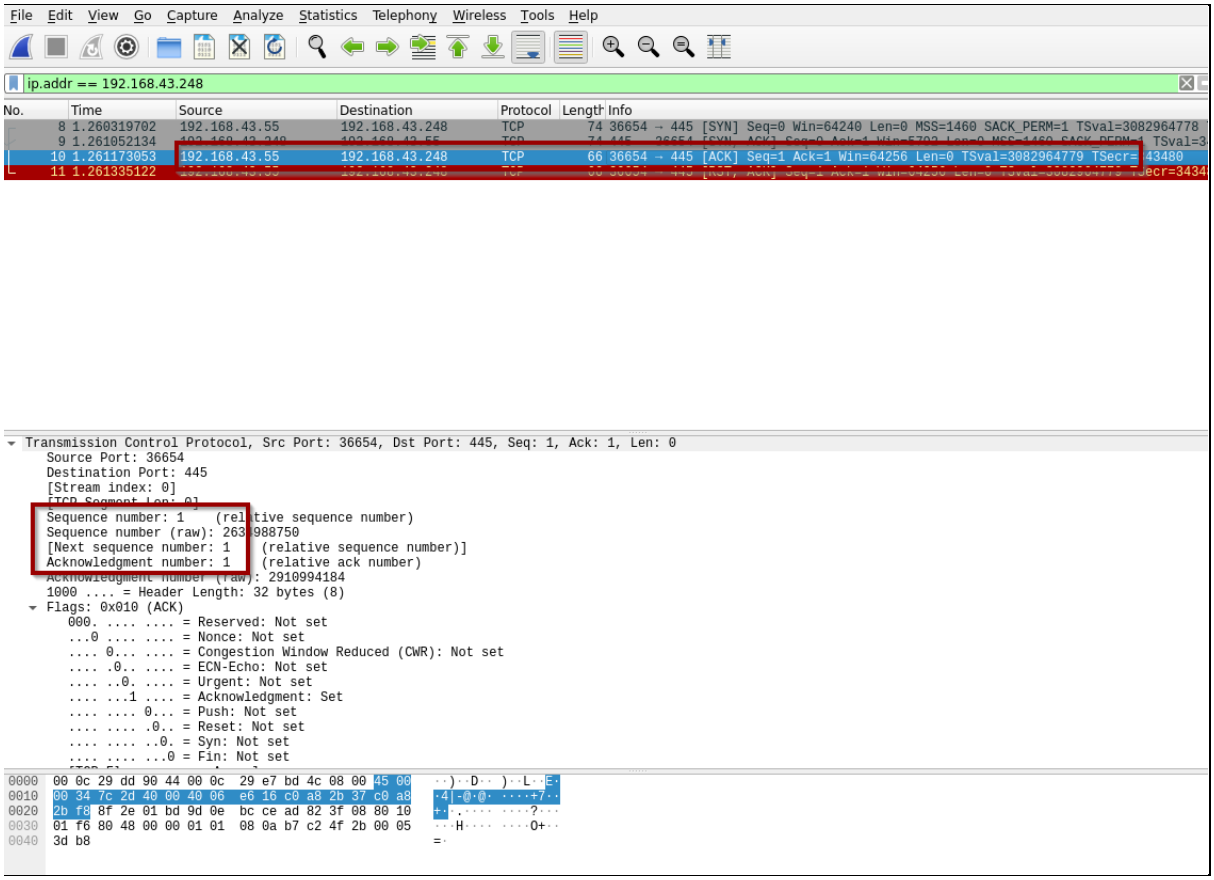


We added a Wireshark filter *ip.addr == 192.168.43.248* to filter packets which deals with 192.168.43.248.

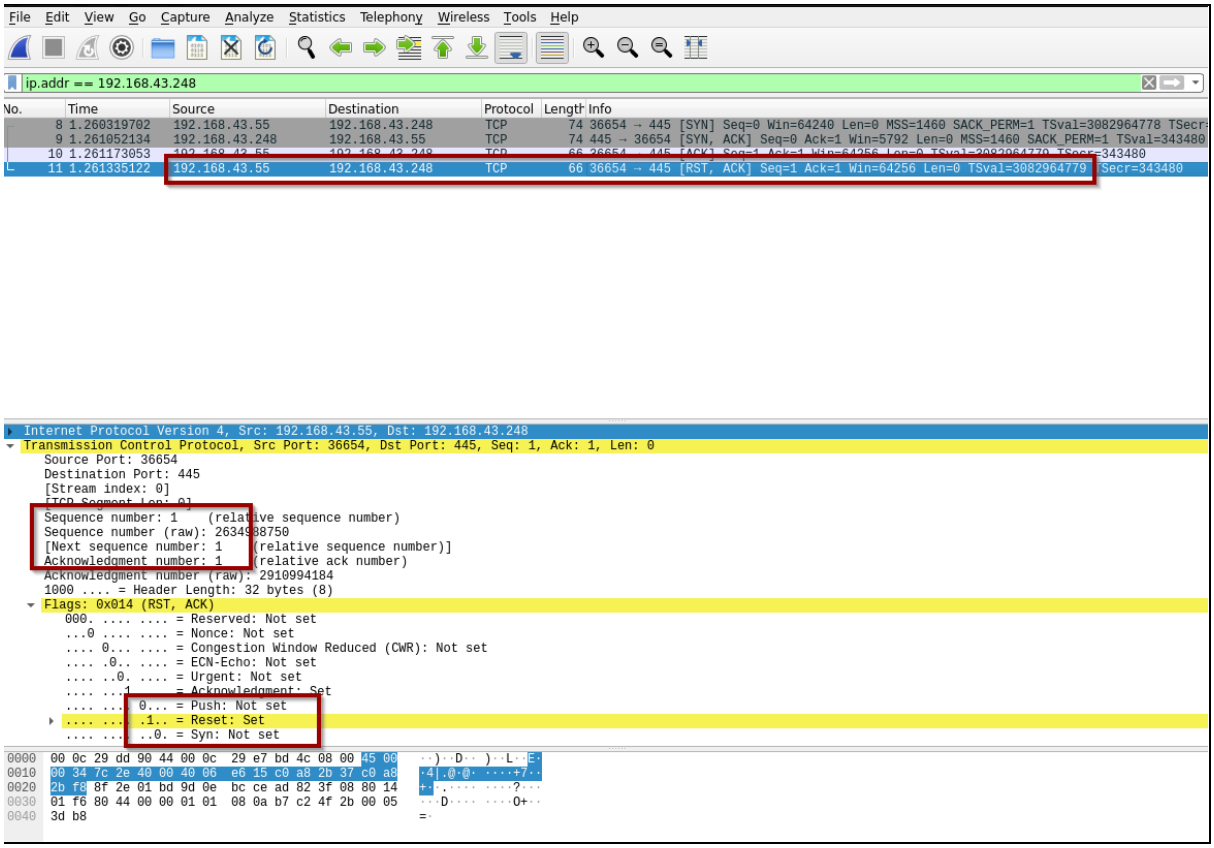
So total of 4 packets are captured, that's what we expected.
Let's analyze the first packet, we see that SYN packet is sent to the destination, in the flag section we see SYN flag is set.
Let's see if the server responds.



In the second packet the server sends an ACK (responds for SYN packet 1) and send SYN packet to the source for his validation. We can see the sequence number is 0 and Acknowledgement number is 1.
Now in packet 3 source should ACK the server.



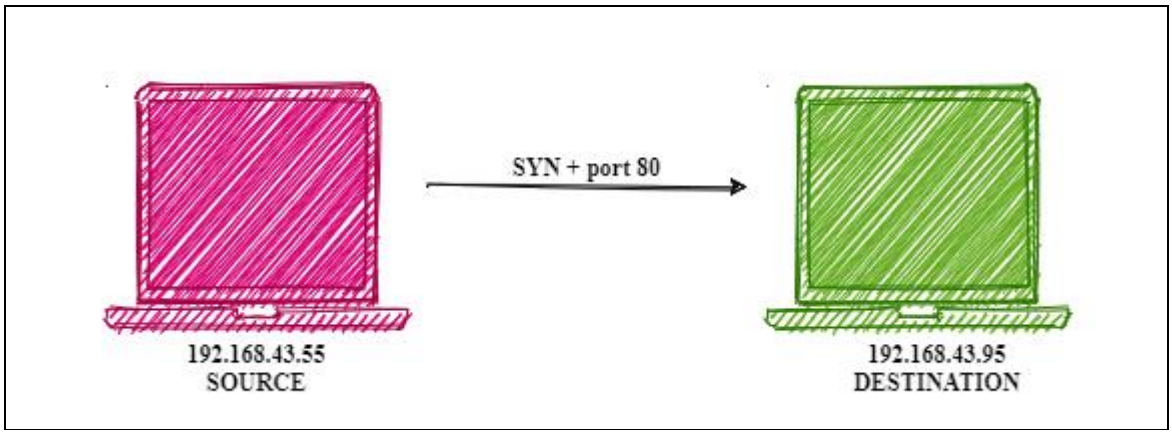
In this step the source sends an ACK packet to the server and thus completes the 3-way handshake.
The last packet completes the connection and then resets it.



Since, the 3-way handshake is completed therefore we know the particular port is open on the destination server.

For Filtered Port

A filtered port indicates the presence of firewall, or any other network issue/security device that is blocking the port, thus nmap cannot identify if it's open or closed. (The machine does not respond at all if firewall is enabled in it).



Syntax

nmap -sT -p <port number> <destination IP>

Nmap command:

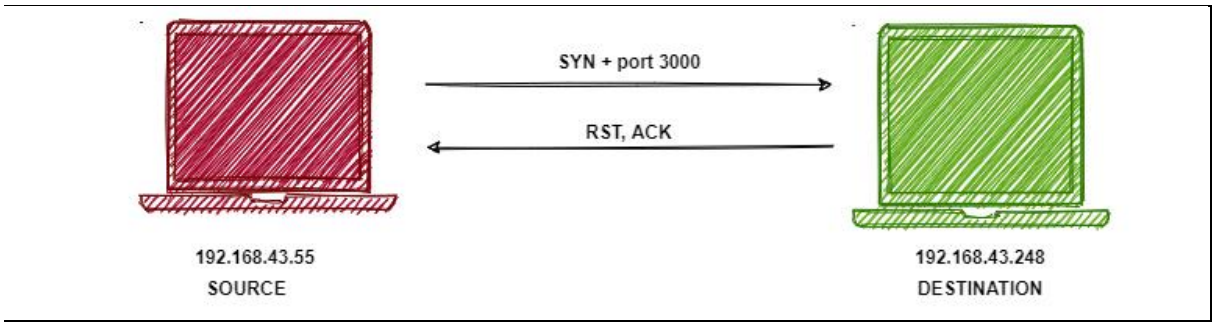
nmap -sT -p 80 192.168.43.95

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sT -p 80 192.168.43.95
Starting Nmap 7.91 ( https://nmap.org ) at 2020-11-12 11:57 IST
Nmap scan report for [redacted]
Host is up (0.00046s latency).

PORT   STATE    SERVICE
80/tcp filtered http
[redacted]
```

The scan shows port as filtered which signifies firewall is on.
Let's analyses what comes up in Wireshark.

Only SYN packets were sent from source (our machine), the destination doesn't respond at all. Firewall simply drops the packets.

For Closed Port



Let's scan a closed port.

Syntax

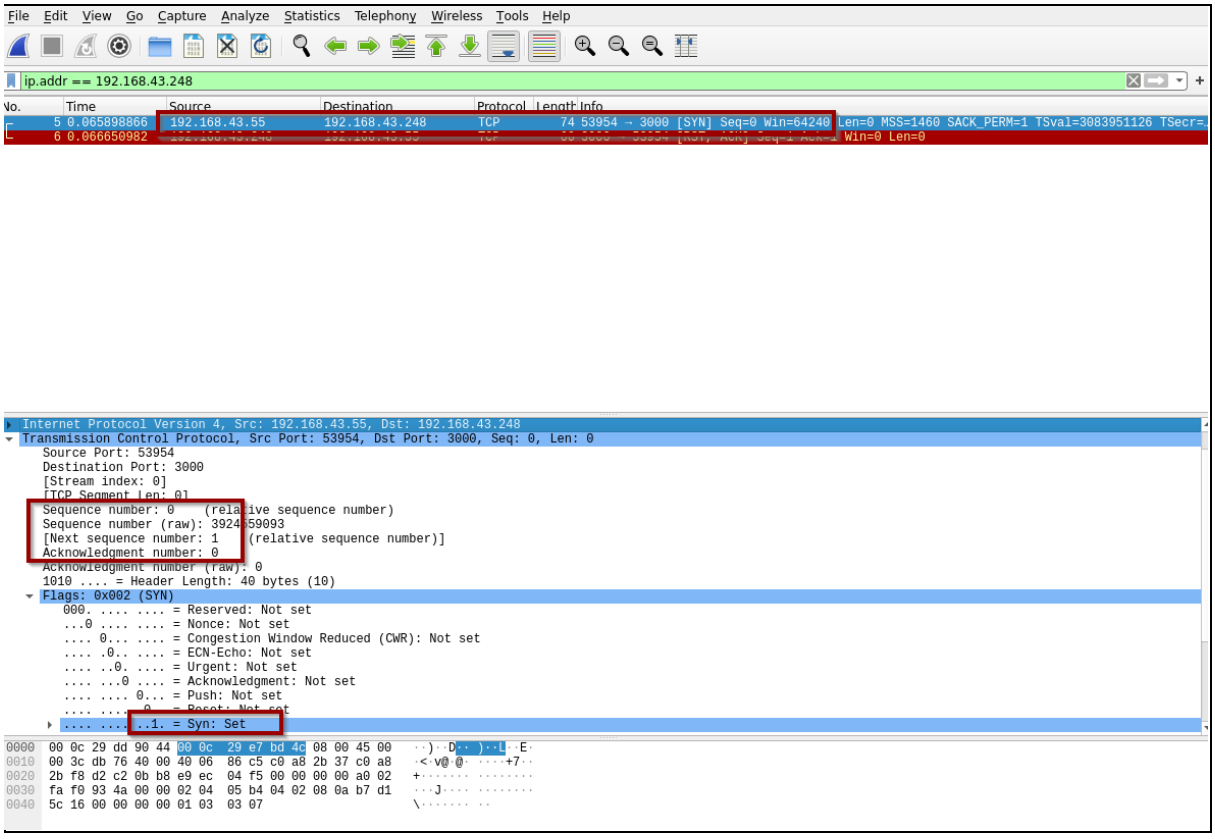
nmap -sT -p <port number> <destination IP>

Nmap scan command

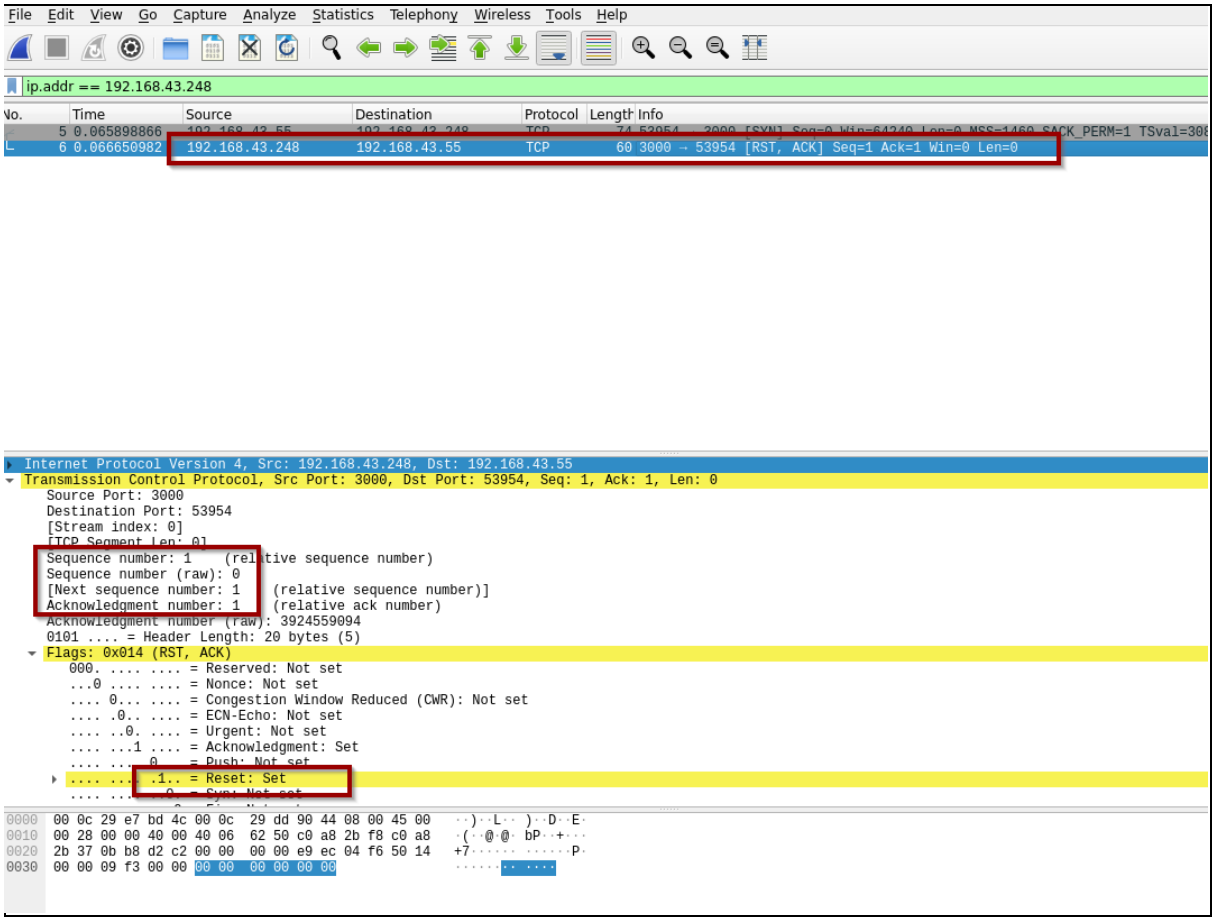
nmap -sT -p 3000 192.168.43.248

```
kali@kali:~$ sudo nmap -sT -p 3000 192.168.43.248
[sudo] password for kali:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-19 20:02 IST
Nmap scan report for 192.168.43.248
Host is up (0.00092s latency).

PORT     STATE  SERVICE
3000/tcp closed ppp
MAC Address: 00:0C:29:DD:90:44 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
kali@kali:~$
```

Since the port is closed 3-way handshake would not be completed. Let's analyze it in Wireshark.



First the source sends a SYN packet to the server.



And the server/destination sends an RST packet and resets the connections and thus 3-way handshake doesn't complete and therefore Nmap declares the port as closed.

TCP SYN (STEALTH) SCAN (-sS)

Stealth scan is one of the most popular scanning techniques. This technique is often referred as half-open scanning as this does not open full TCP connection.

In this we (source) send SYN packet to the destination and the server responds with SYN, ACK packet, then we immediately terminate or kill the connection.

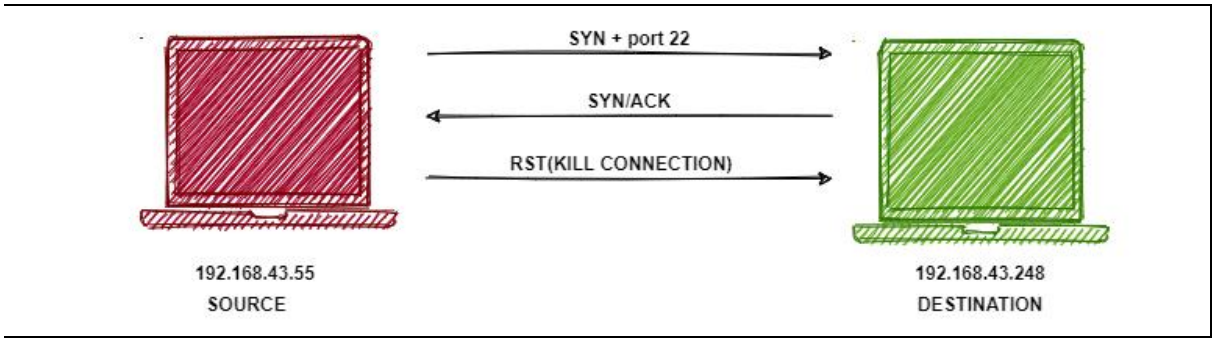
Advantage:

1) As the name suggest the scanning is stealth thus is not detected easily.
2) Faster than -sT because it doesn't complete the three-way handshake.

Disadvantage:

1) Can be used for malicious intent.
2) Handshake is not completed thus not authorized/secured.

For open port



Let’s scan a port which is open and at the same time capture the packets in Wireshark for analysis.

Syntax

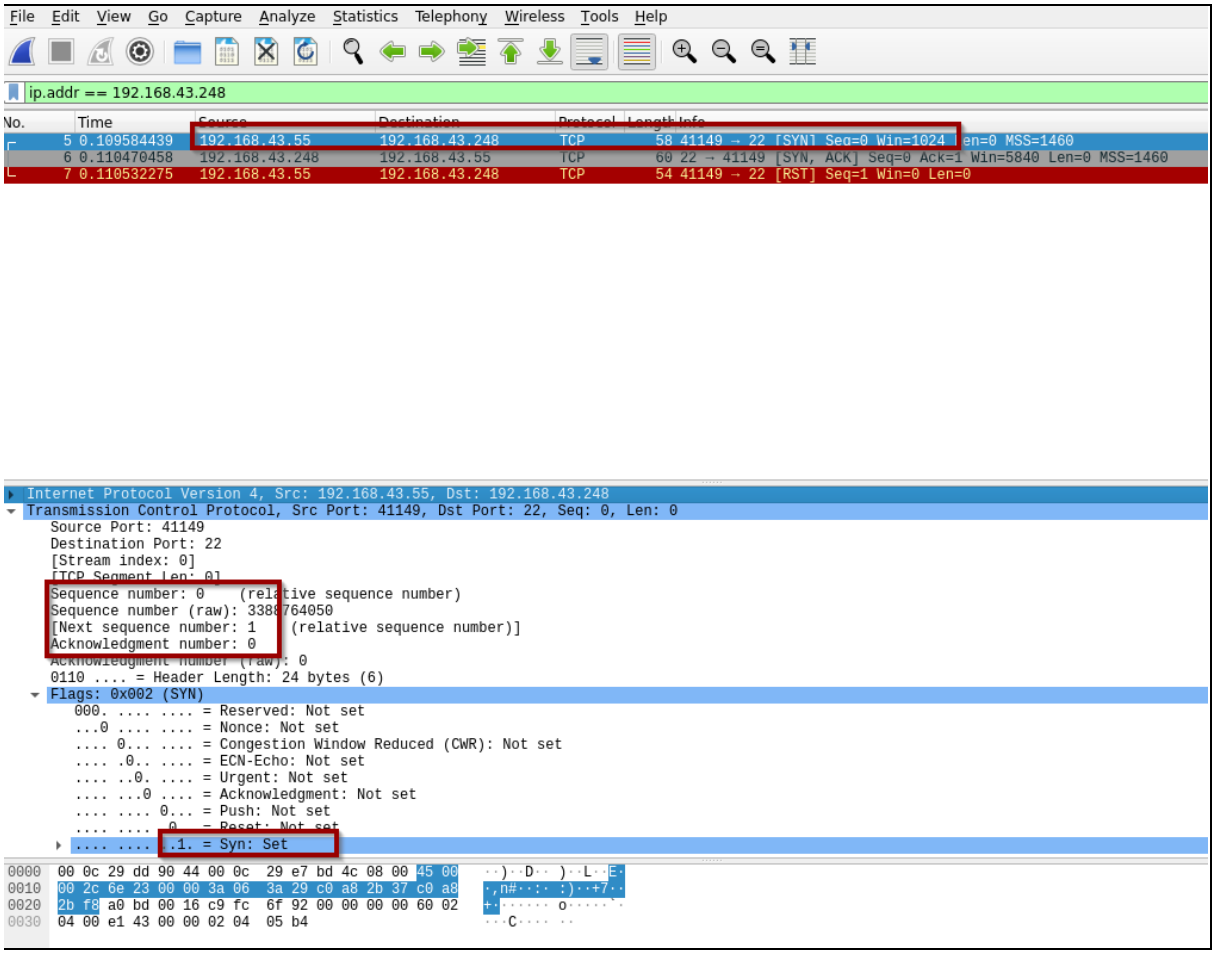
nmap -sS -p <port number> <destination IP>

Nmap command is

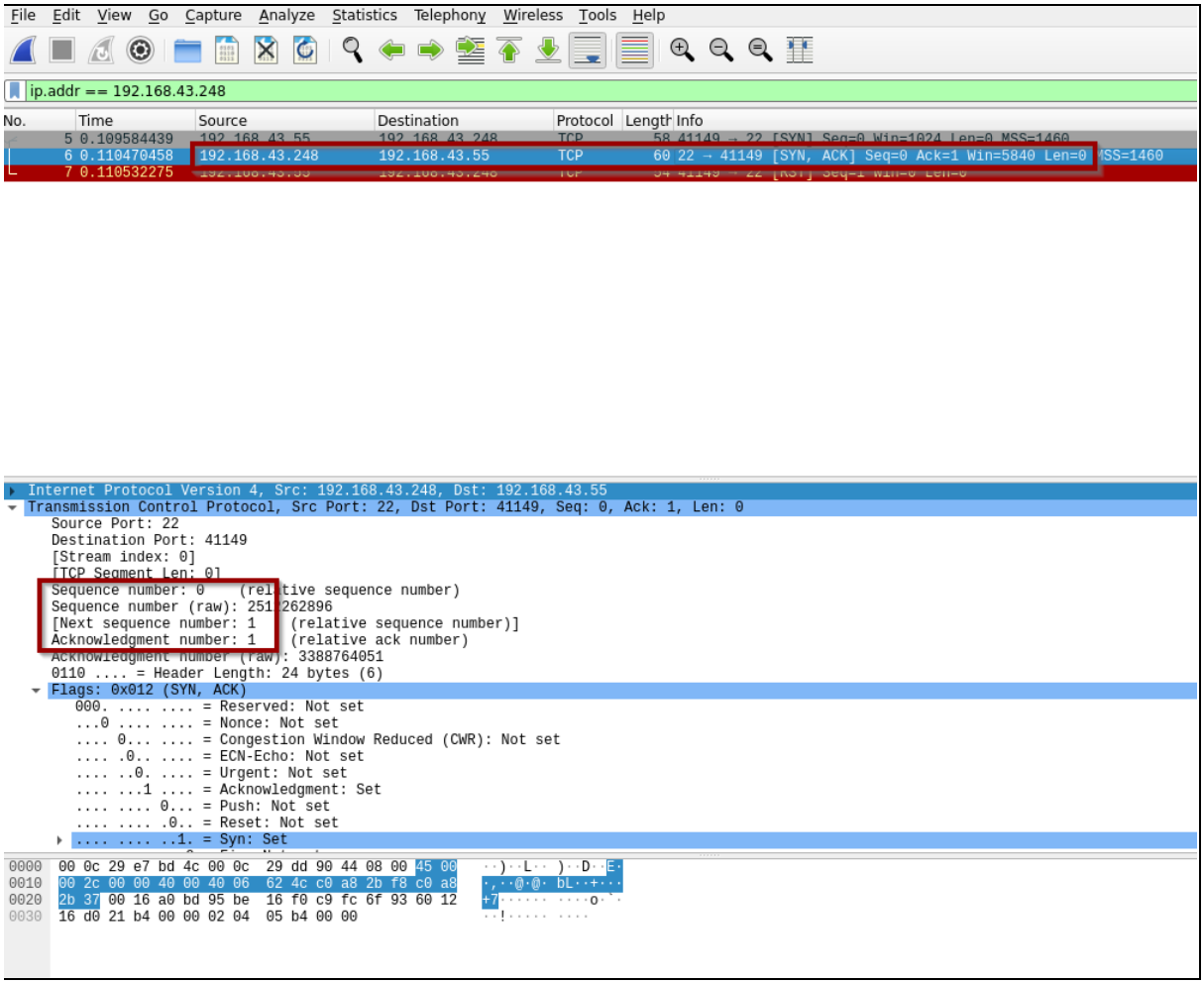
nmap -sS -p 22 192.168.43.248

```
kali@kali:~$ sudo nmap -sS -p 22 192.168.43.248
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 07:51 IST
Nmap scan report for 192.168.43.248
Host is up (0.00091s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:DD:90:44 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.53 seconds
kali@kali:~$
```

Port is open, lets analyze it in Wireshark.

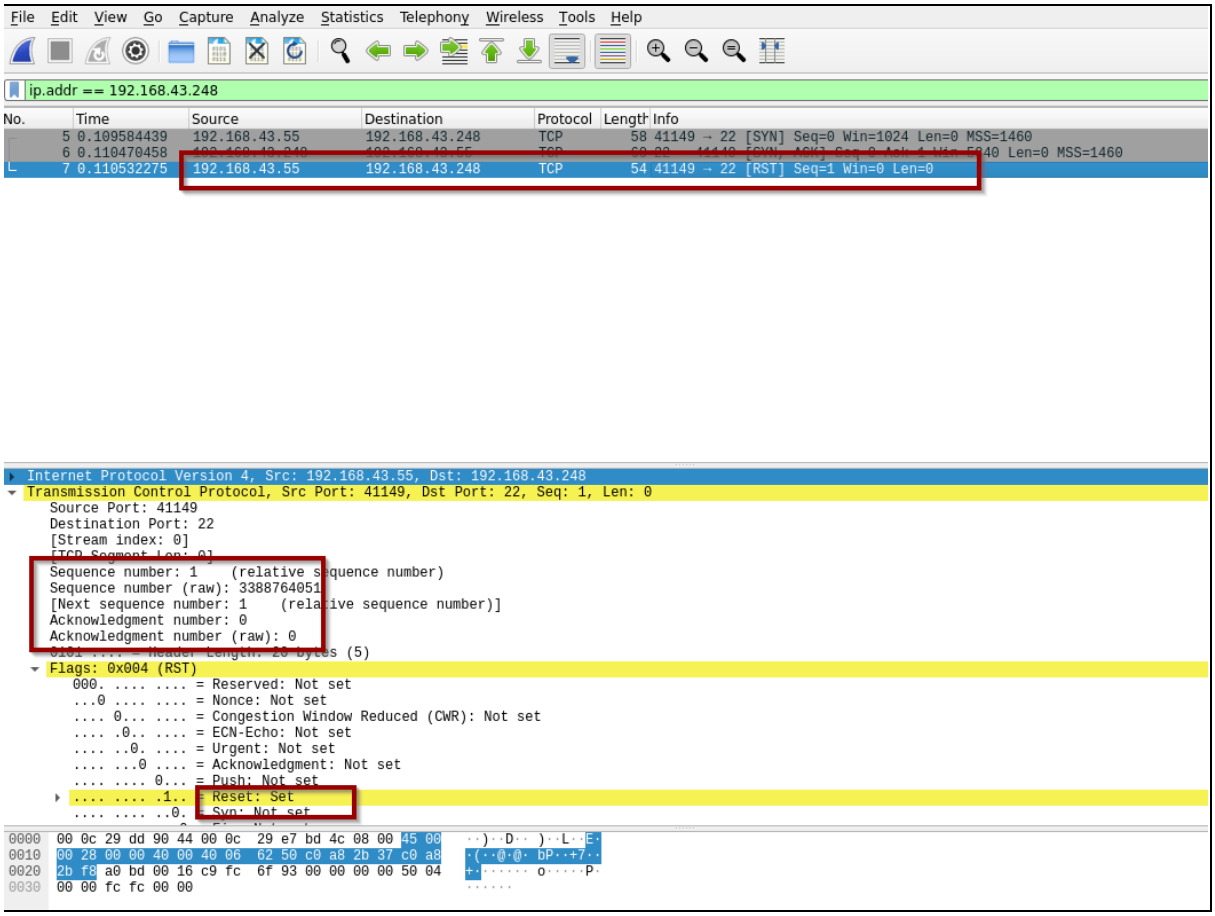


Let’s analyze the first packet, we see that SYN packet is sent to the destination, in the flag section we see SYN is in Set mode.
Let’s see if the server responds.



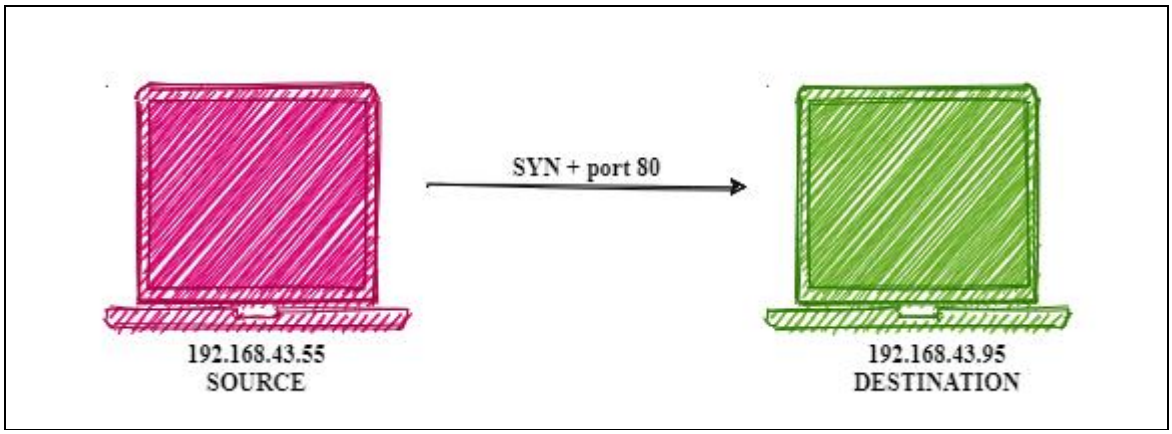
In the second packet the server sends an ACK (responds for SYN packet 1) and send SYN packet to the source for his validation. We can see the sequence number is 0 and Acknowledgement number is 1.

Next step is important, this step makes this scanning Stealthy, instead of sending ACK packets to destination, we send RST (reset) packet and the connection is killed. Since we received SYN and ACK packet from the destination, this means that the port is open.



For Filtered Port

A filtered port indicates the presence of firewall, or any other network issue/security device that is blocking the port, thus nmap cannot identify if it's open or closed. (The machine does not respond at all if firewall is enabled in it).



Syntax

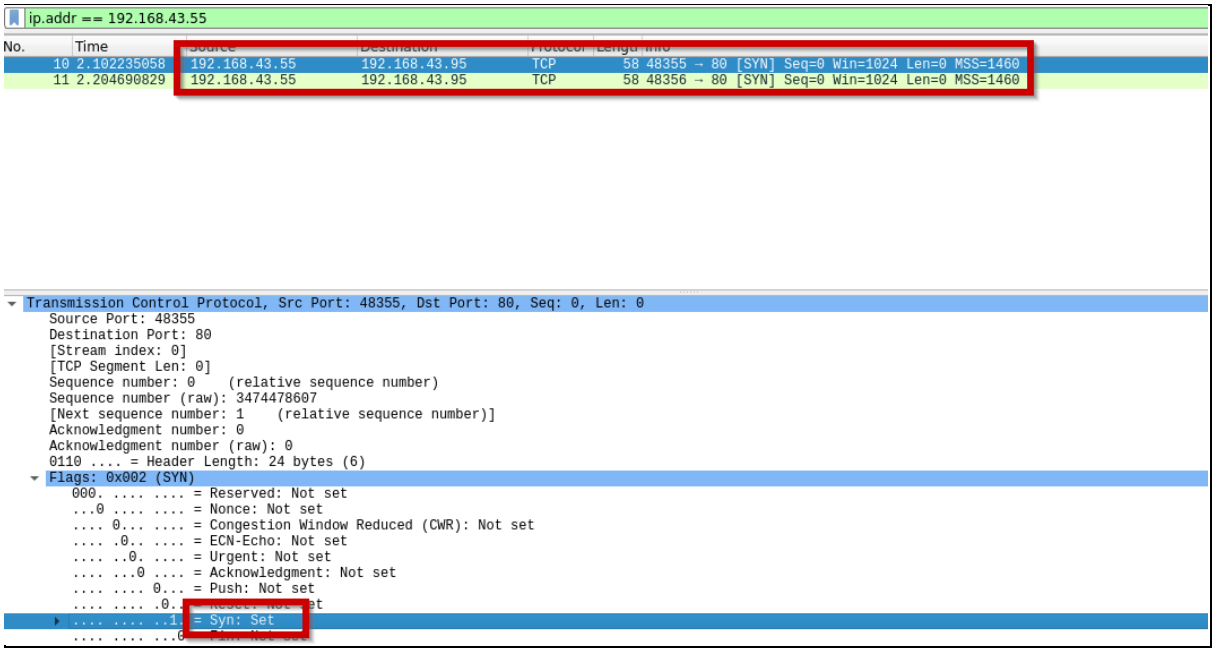
nmap -sS -p <port number> <destination IP>

Nmap command:

nmap -sS -p 80 192.168.43.95

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sS -p 80 192.168.43.95
Starting Nmap 7.91 ( https://nmap.org ) at 2020-11-12 11:53 IST
Nmap scan report for [redacted]
Host is up (0.00043s latency).

PORT   STATE    SERVICE
80/tcp filtered http
MAC Address: [redacted]

Nmap done: 1 IP address (1 host up) scanned in 0.53 seconds
```

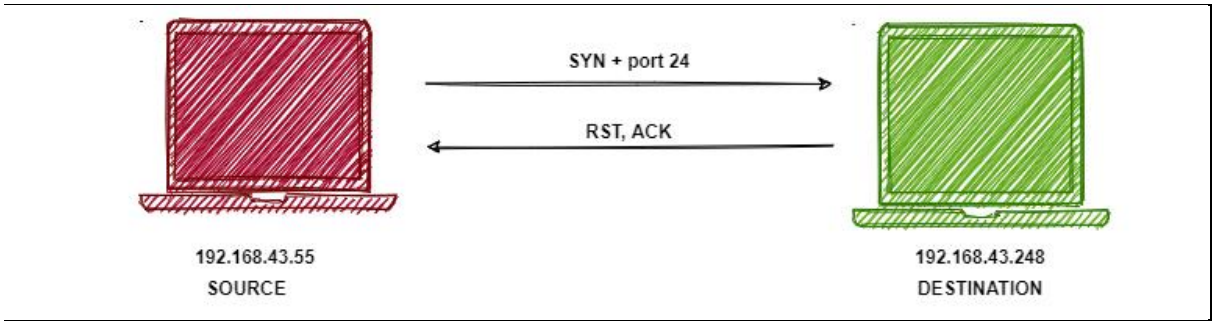
The scan shows port as filtered which signifies firewall is on.
Let's analyses what comes up in Wireshark.



Only SYN packets were sent from source (our machine), the destination doesn't respond at all. Firewall simply drops the packets.

For closed ports

If the port is closed, then it means that we should receive an RST packet from the server in response to our SYN packet.



Let’s scan a closed port.

Syntax

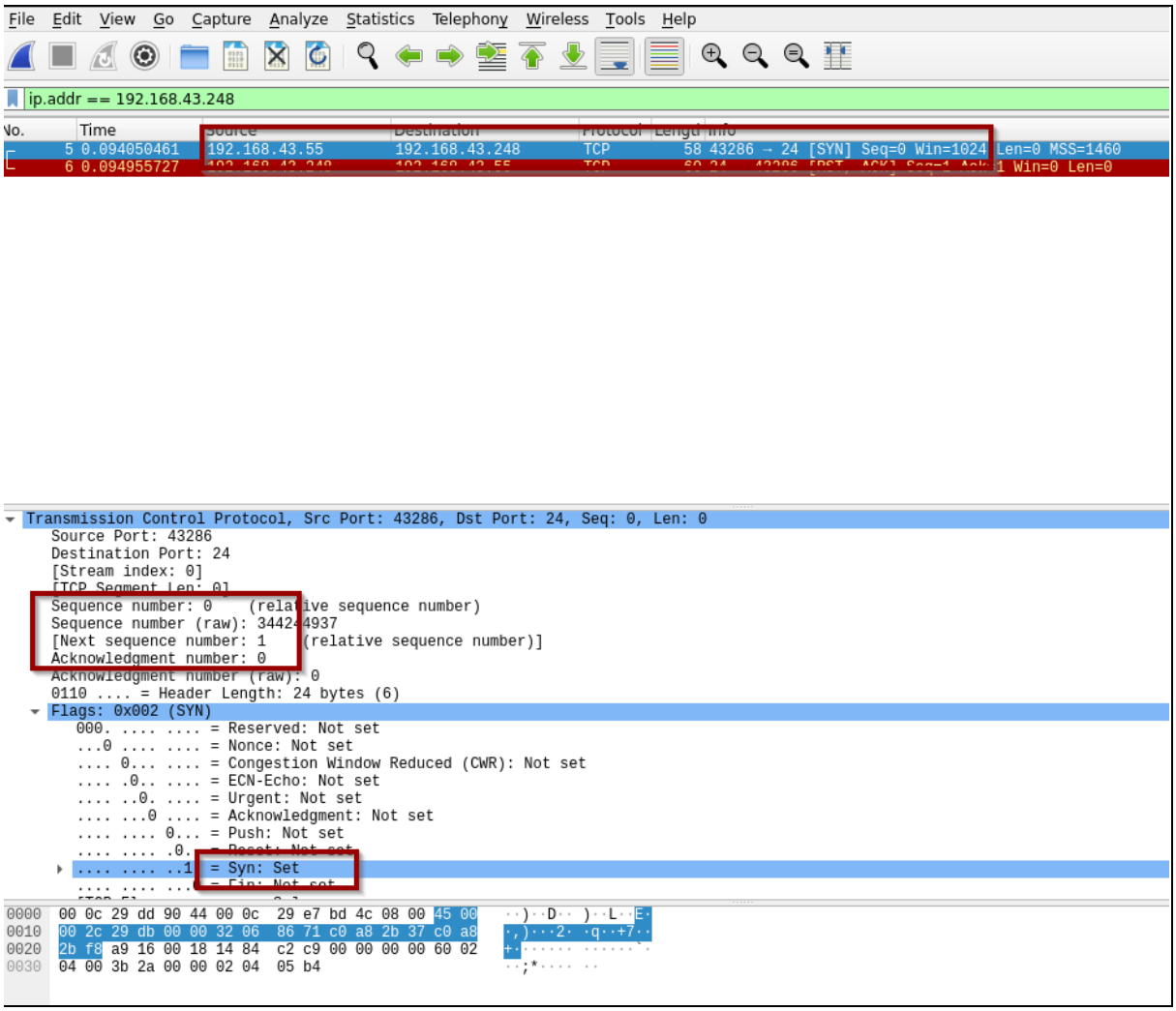
nmap -sS -p <port number> <destination IP>

Nmap scan command

nmap -sS -p 24 192.168.43.248

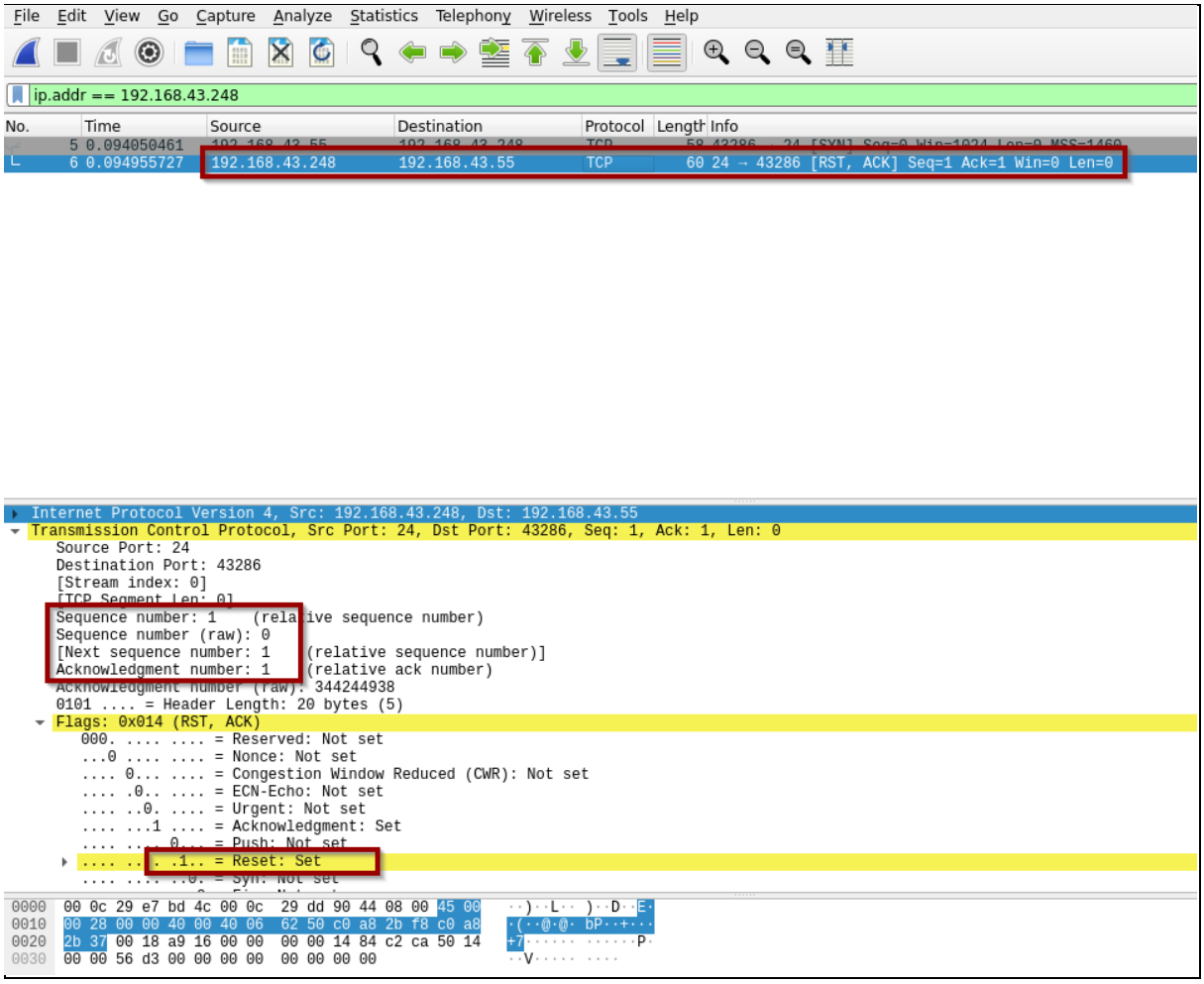
```
kali@kali:~$ sudo nmap -sS -p 24 192.168.43.248
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 07:57 IST
Nmap scan report for 192.168.43.248
Host is up (0.0015s latency).

PORT   STATE  SERVICE
24/tcp closed priv-mail
MAC Address: 00:0C:29:DD:90:44 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.52 seconds
kali@kali:~$
```

Let’s analyze this scan in Wireshark.



We sent a SYN packet to the server and we can see in flag that SYN is Set.

Let’s see the response from the server.

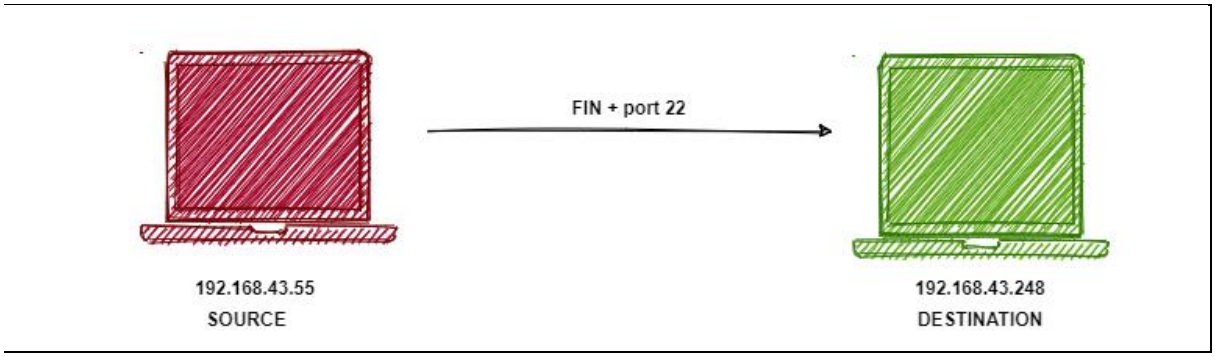


We get an RST and ACK packet, as the port is closed.

TCP FIN SCAN (-sF)

This scan is simple. We send a FIN packet and if the port is open then there is no response from the server, if port is closed then the server responds with a RST, ACK packet.
Drawback: Fin Scans only work for Linux machines and cannot be run against latest Windows machines.

For open port



Let’s perform a FIN scan through nmap

Syntax

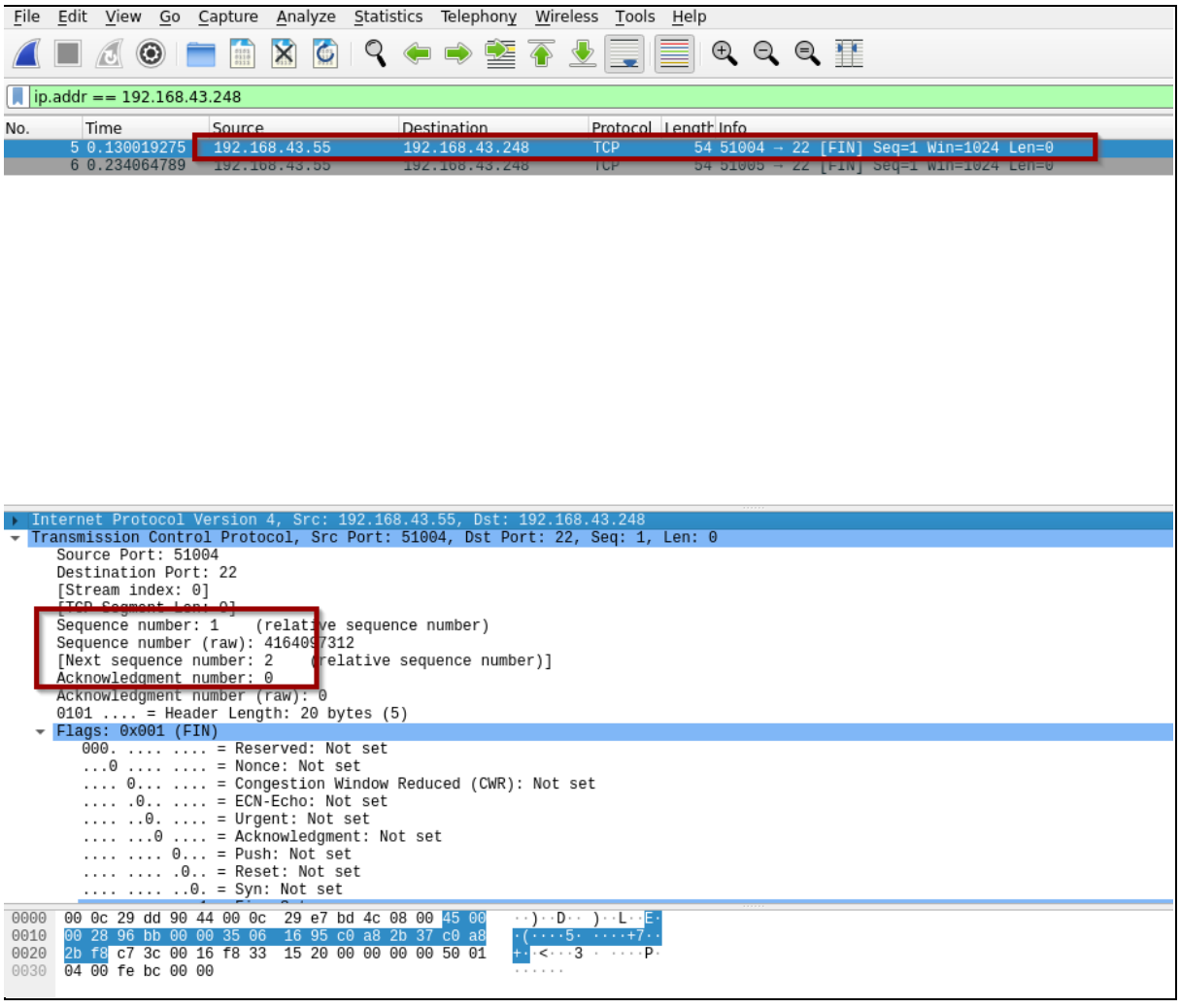
nmap -sF -p <port number> <destination IP>

Nmap scan command

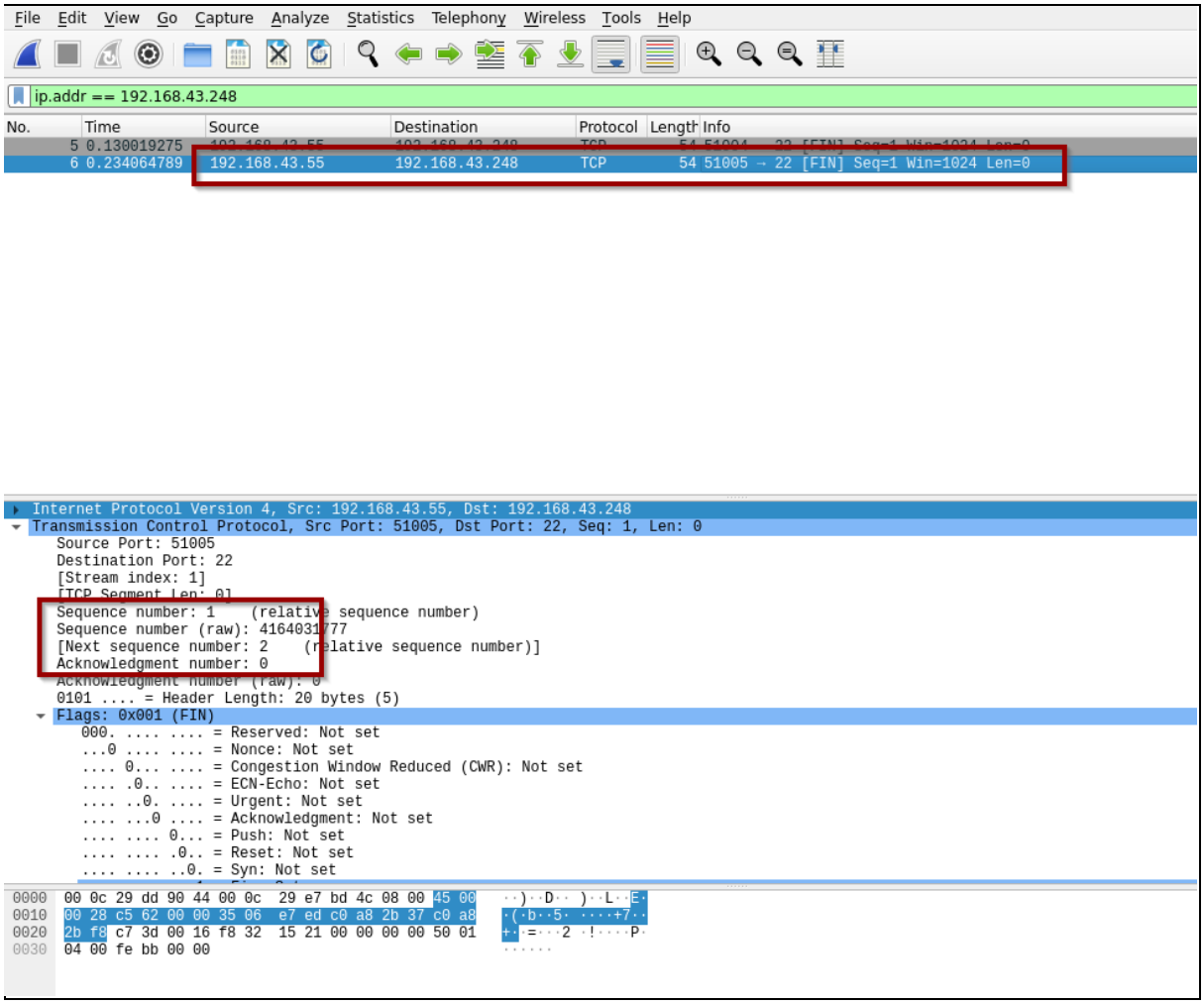
nmap -sF -p 22 192.168.43.248

```
kali@kali:~$ sudo nmap -sF -p 22 192.168.43.248
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 08:43 IST
Nmap scan report for 192.168.43.248
Host is up (0.00074s latency).

PORT   STATE         SERVICE
22/tcp open|filtered ssh
MAC Address: 00:0C:29:DD:90:44 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.05 seconds
kali@kali:~$
```

Let’s analyze this scan through Wireshark.

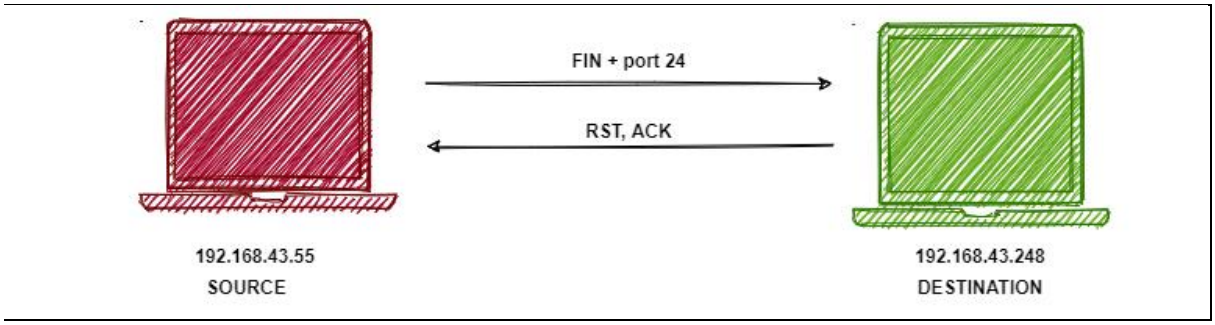


FIN packet is sent from the source , you can see the flag section(there's no SYN, ACK or RST packet SET)
The next packet is also sent from the source.



Again, it's SYN packet sent. We did not receive any packet from destination or server , but that's what this scan is about. If we don't receive any response from the server that means the port is open for connection.

For closed port



Let's scan a closed port.

Syntax

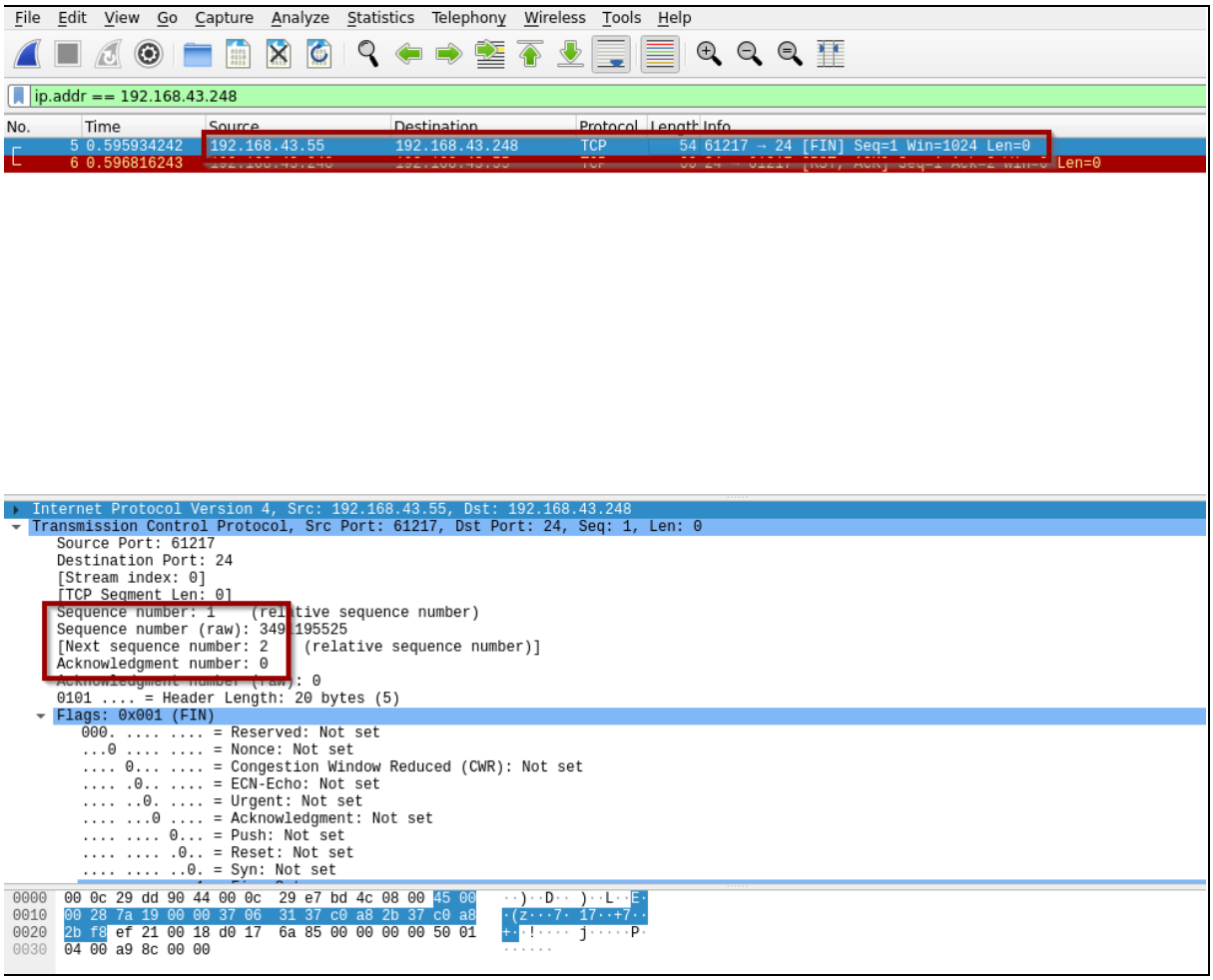
nmap -sF -p <port number> <destination IP>

Nmap scan command

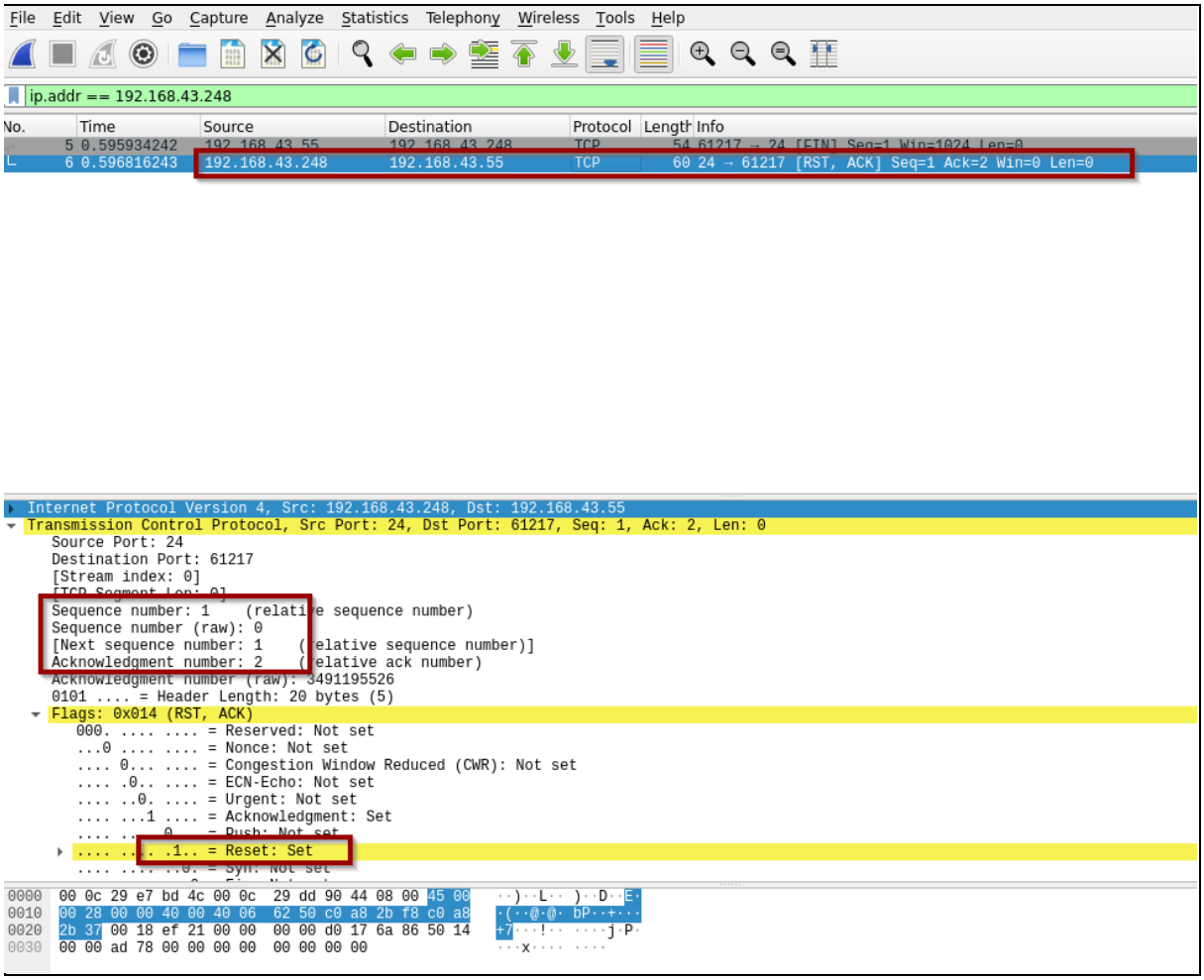
nmap -sF -p 24 192.168.43.248

```
kali@kali:~$ sudo nmap -sF -p 24 192.168.43.248
[sudo] password for kali:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 08:32 IST
Nmap scan report for 192.168.43.248
Host is up (0.00091s latency).

PORT   STATE  SERVICE
24/tcp closed priv-mail
MAC Address: 00:0C:29:DD:90:44 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.24 seconds
kali@kali:~$
```

Port is closed that means we should see a response from the server, let's verify it in Wireshark.



First packet is sent from us and it's a FIN packet.



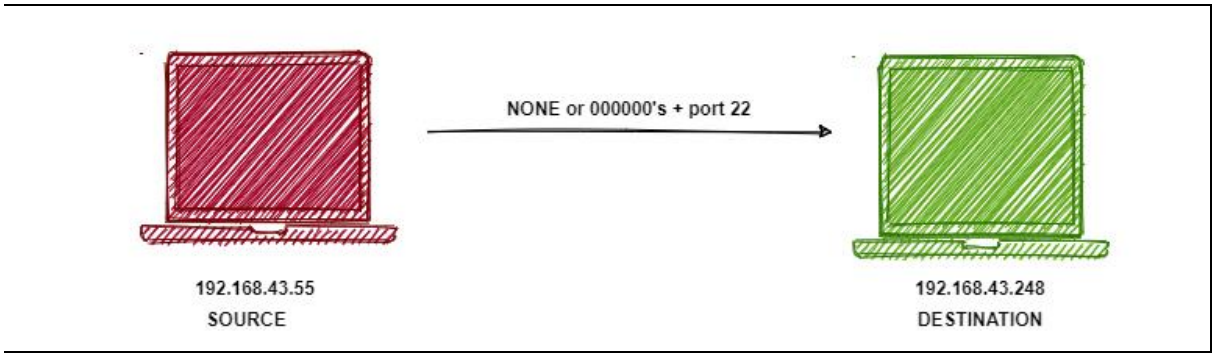
Since the port is closed the server sends the responds as RST (reset), ACK packet.

NULL SCAN (-sN)

In this scan the source sends TCP packets that contain a series of zeros “00000000”, and since no flags are set , the destination doesn’t know how to process the request and thus discards the packets. This means that port is open. If the server responds to the packet, then it means those ports are closed.

Drawback: Null Scans only work for Linux machines and cannot be run against latest Windows machines.

For open ports (we will see two packets sent when state of port is open as well as filtered)



Let’s scan an open port and capture the packets in Wireshark.

Syntax

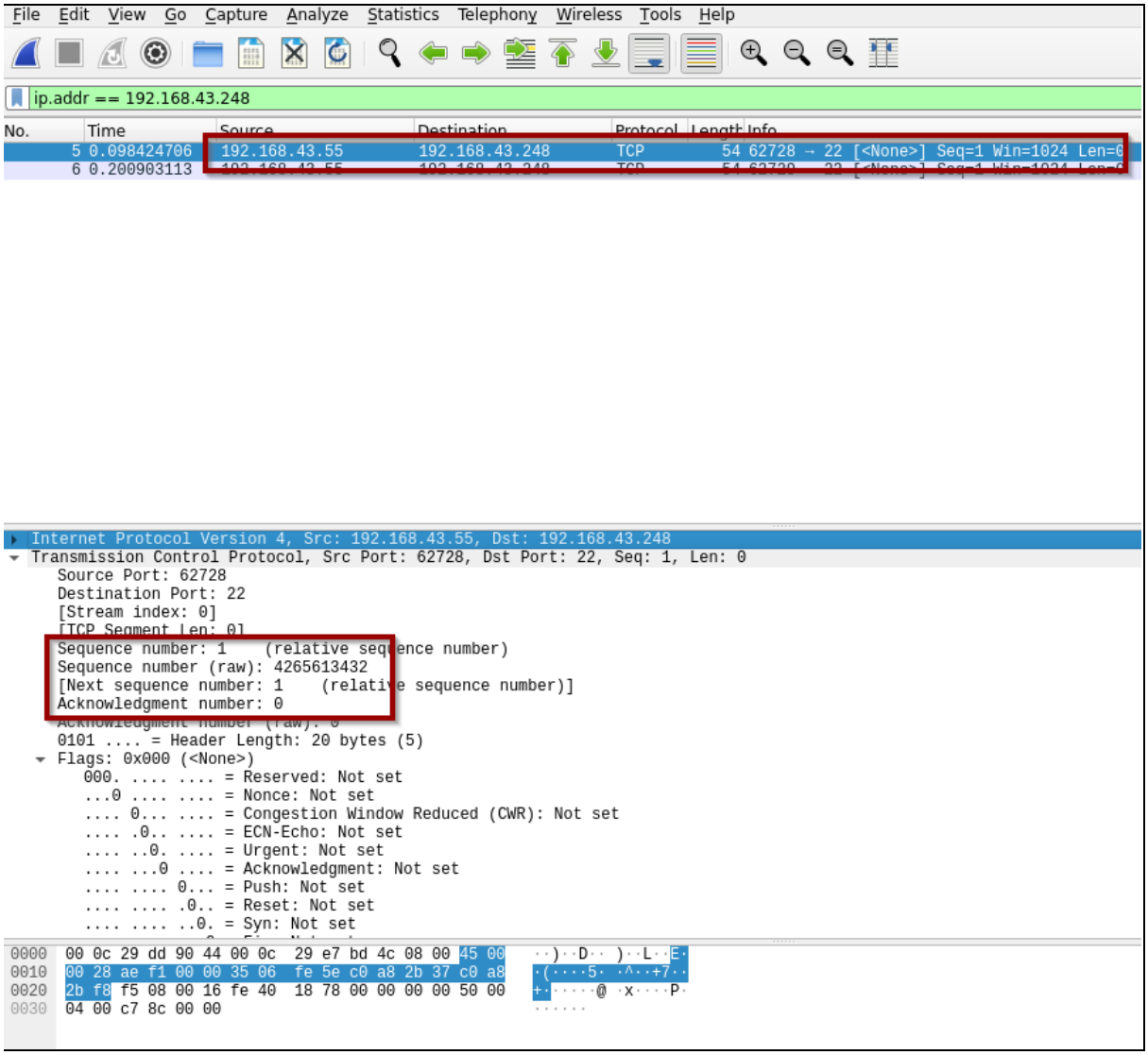
nmap -sN -p <port number> <destination IP>

Nmap scan command

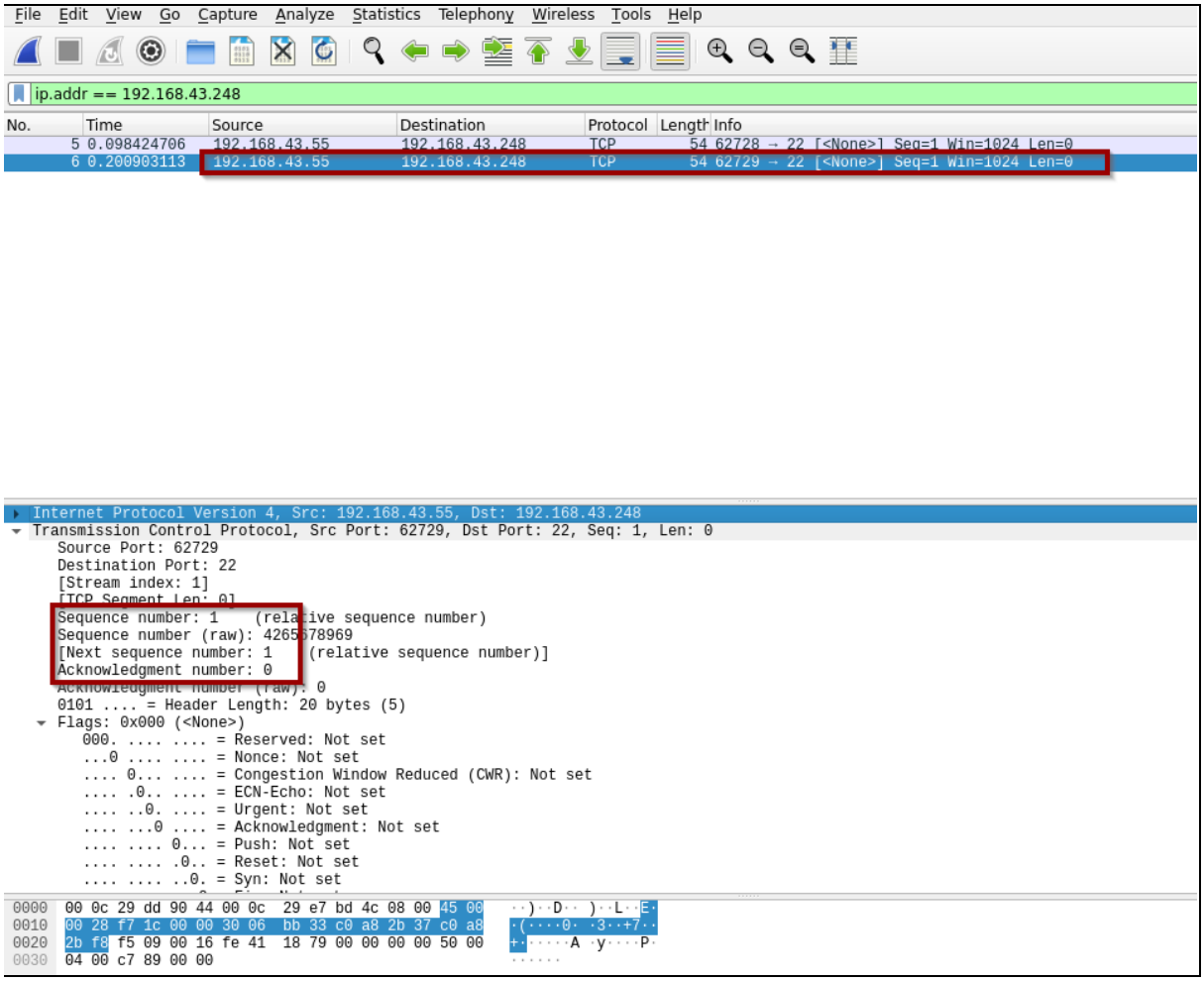
nmap -sN -p 22 192.168.43.248

```
kali@kali:~$ sudo nmap -sN -p 22 192.168.43.248
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 09:17 IST
Nmap scan report for 192.168.43.248
Host is up (0.00093s latency).

PORT   STATE         SERVICE
22/tcp open|filtered ssh
MAC Address: 00:0C:29:DD:90:44 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.73 seconds
kali@kali:~$
```

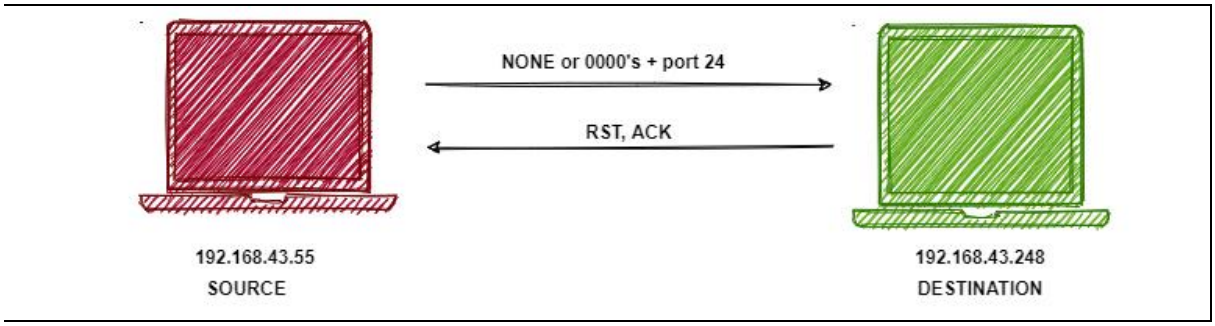
Port is open , this means we didn’t get any response from the destination.



Packet one is sent with no flags Set and we did not receive the response in second packet as that packet is also sent from the source.



For closed port



Let’s scan a closed port and capture the packets in Wireshark.

Syntax

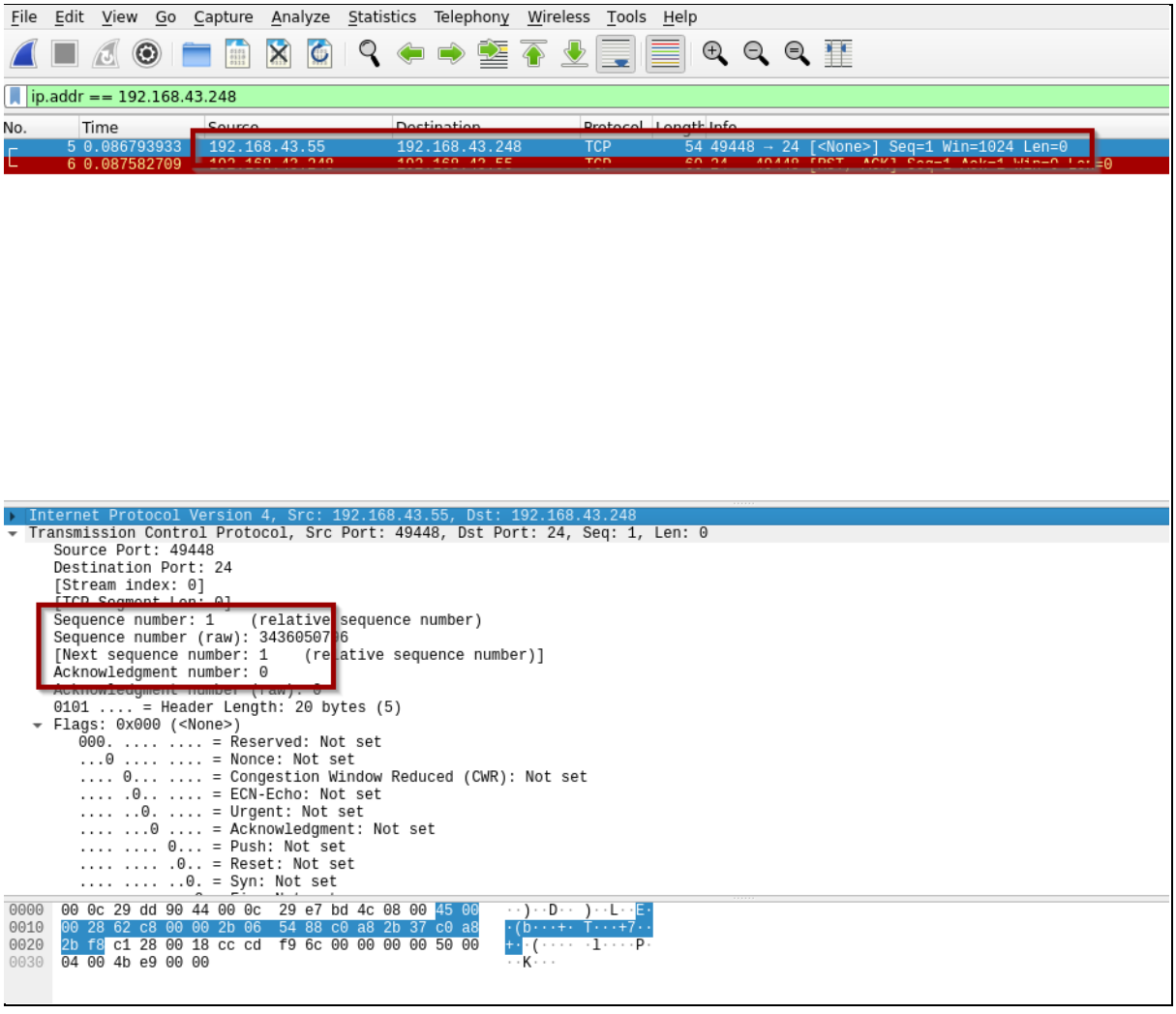
nmap -sN -p <port number> <destination IP>

Nmap scan command

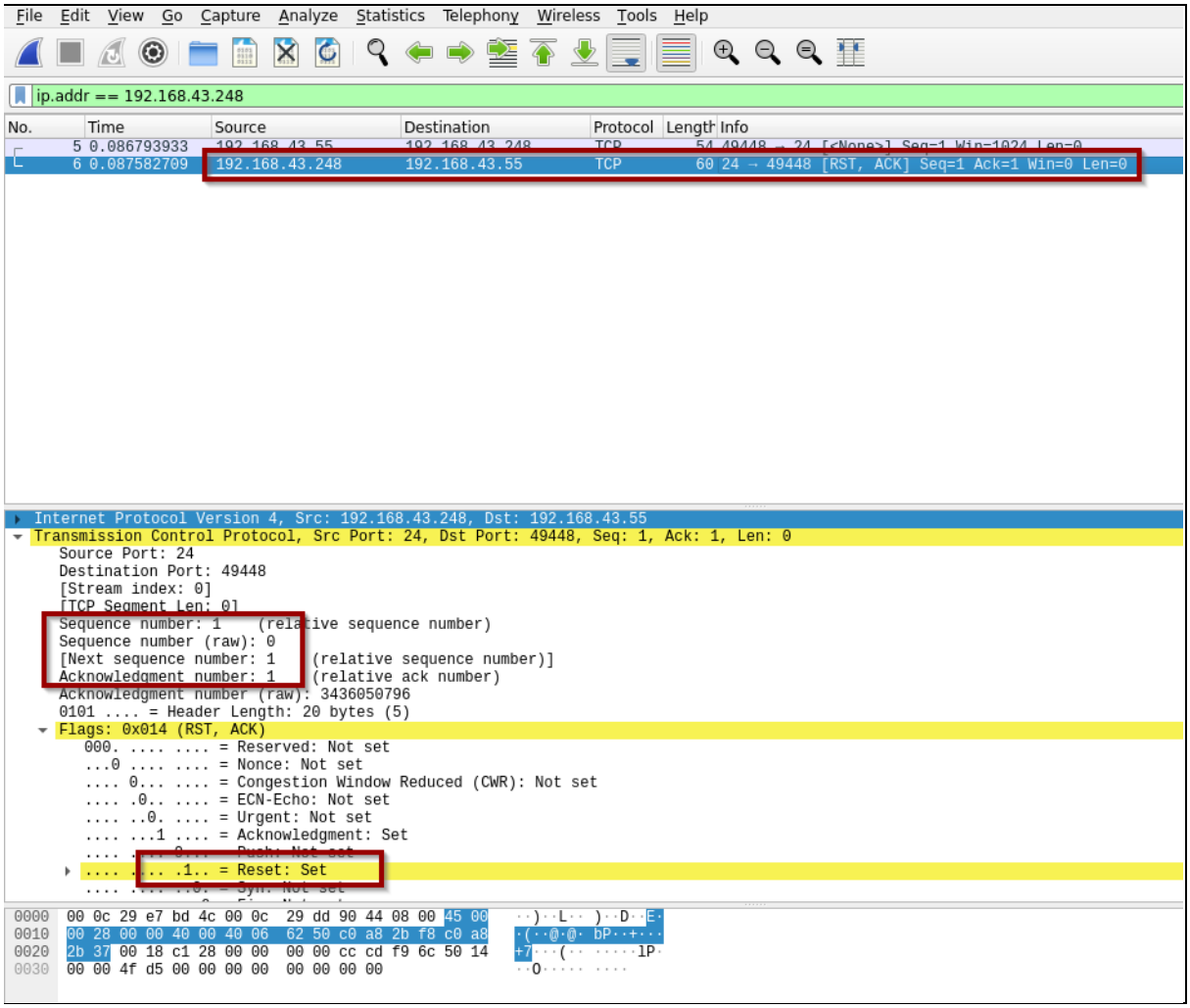
nmap -sN -p 24 192.168.43.248

```
kali@kali:~$ sudo nmap -sN -p 24 192.168.43.248
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 09:13 IST
Nmap scan report for 192.168.43.248
Host is up (0.00085s latency).

PORT   STATE  SERVICE
24/tcp closed priv-mail
MAC Address: 00:0C:29:DD:90:44 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.57 seconds
kali@kali:~$
```

Port is closed , it means we got a response from the destination.



First packet is sent from the source with no flags.
In second packet we get a response from the destination as RST(reset) , ACK packet.



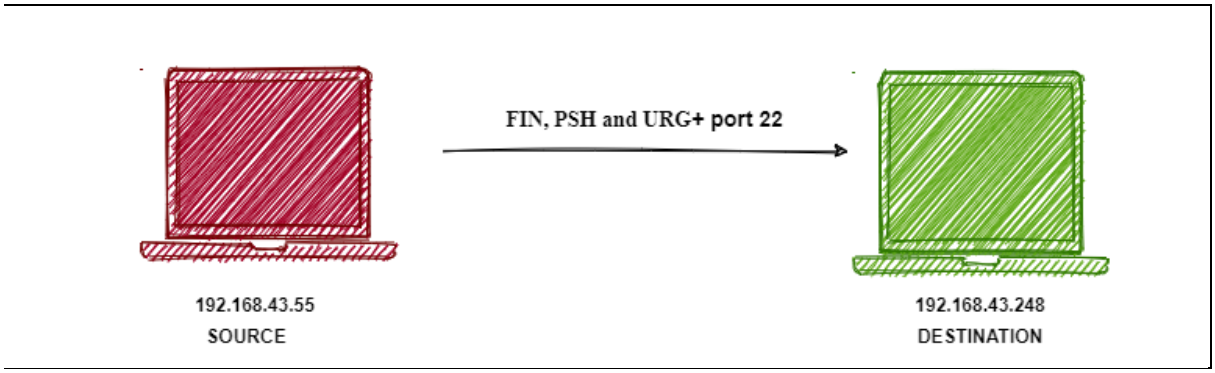
We got a reset flag from the destination which means port is closed.

TCP XMAS SCAN (-sX)

In this source send packets with flags like FIN, PSH and URG, this lights up the packet like a Christmas tree thus names XMAS scan, if the port is open then the destination doesn't respond and discards the packets. If the port is closed, then the destination responds with RSR (reset) and ACK packet.

Drawback: XMAS Scans only work for Linux machines and cannot be run against latest Windows machines.

For open ports



Let’s scan an open port.

Syntax

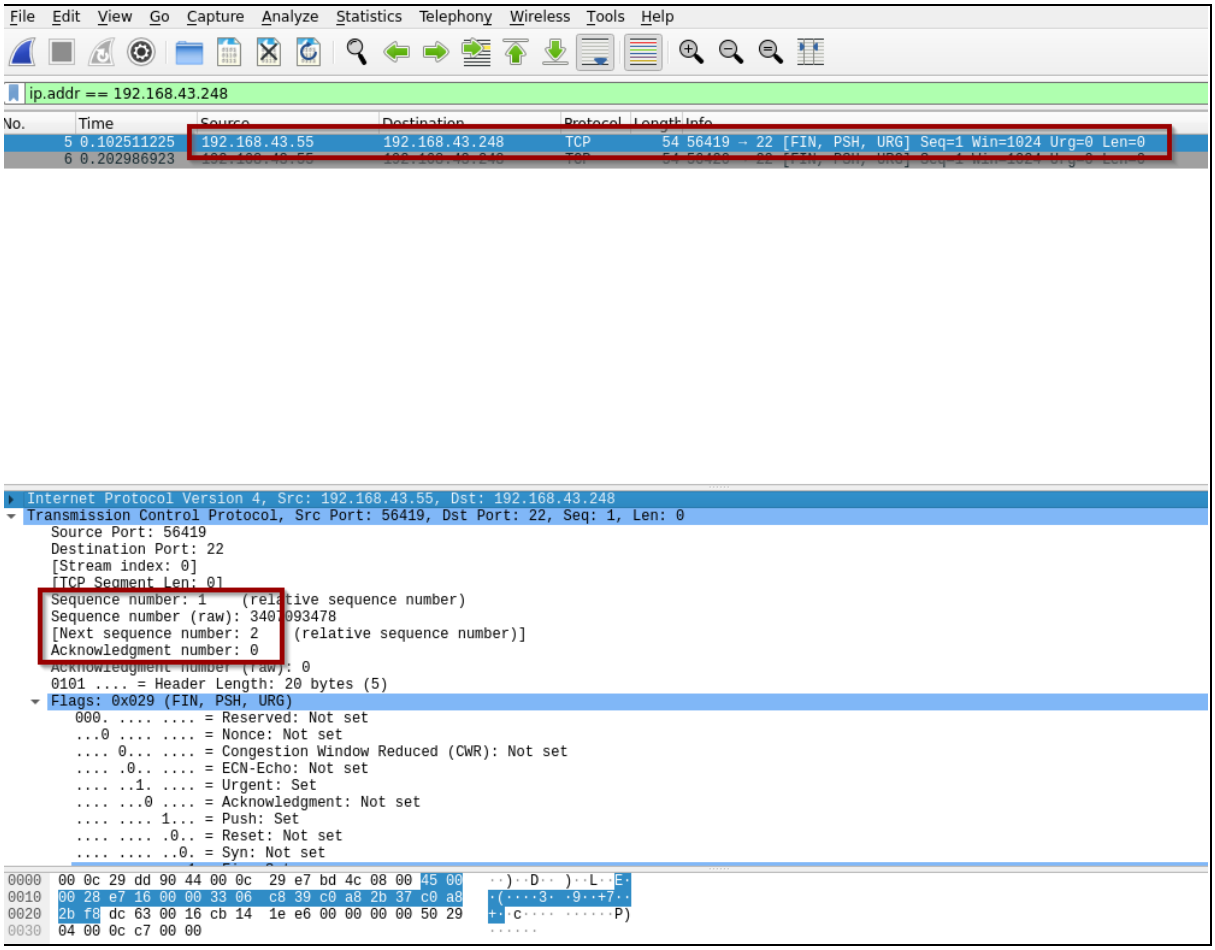
nmap -sX -p <port number> <destination IP>

Nmap scan command

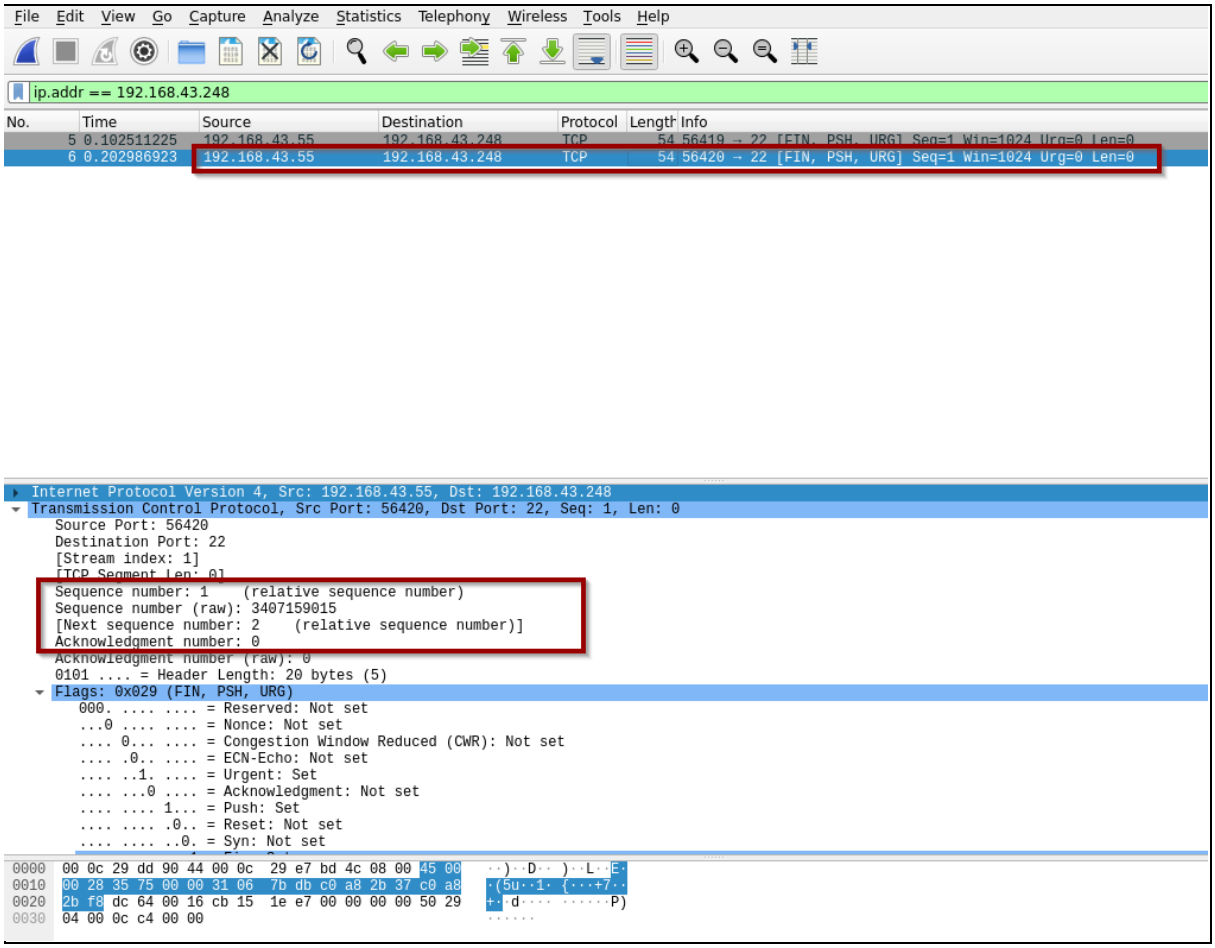
nmap -sX -p 22 192.168.43.248

```
kali@kali:~$ sudo nmap -sX -p 22 192.168.43.248
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 08:59 IST
Nmap scan report for 192.168.43.248
Host is up (0.00067s latency).

PORT   STATE         SERVICE
22/tcp open|filtered ssh
MAC Address: 00:0C:29:DD:90:44 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.80 seconds
kali@kali:~$
```

Port is open , it means we do not get any response from the destination.

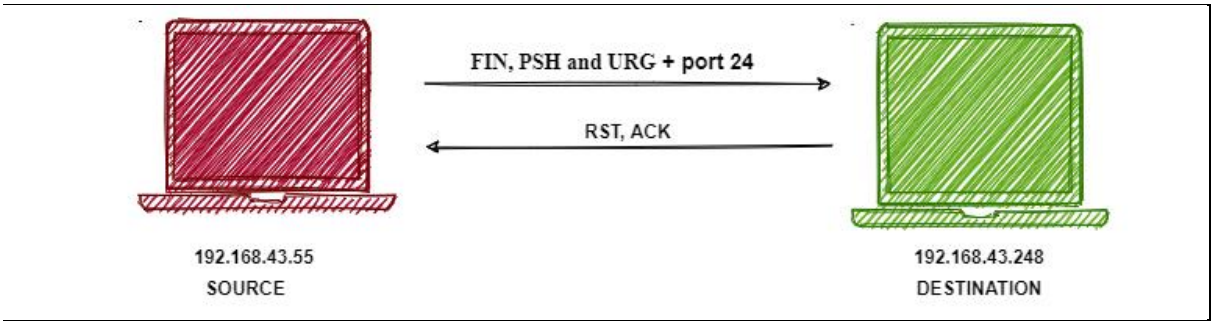


Source also send a second packet (as part of filtered port)



And still there is no response from the destination. This means that port is open for connection.

For closed port



Let's scan a closed port.

Syntax

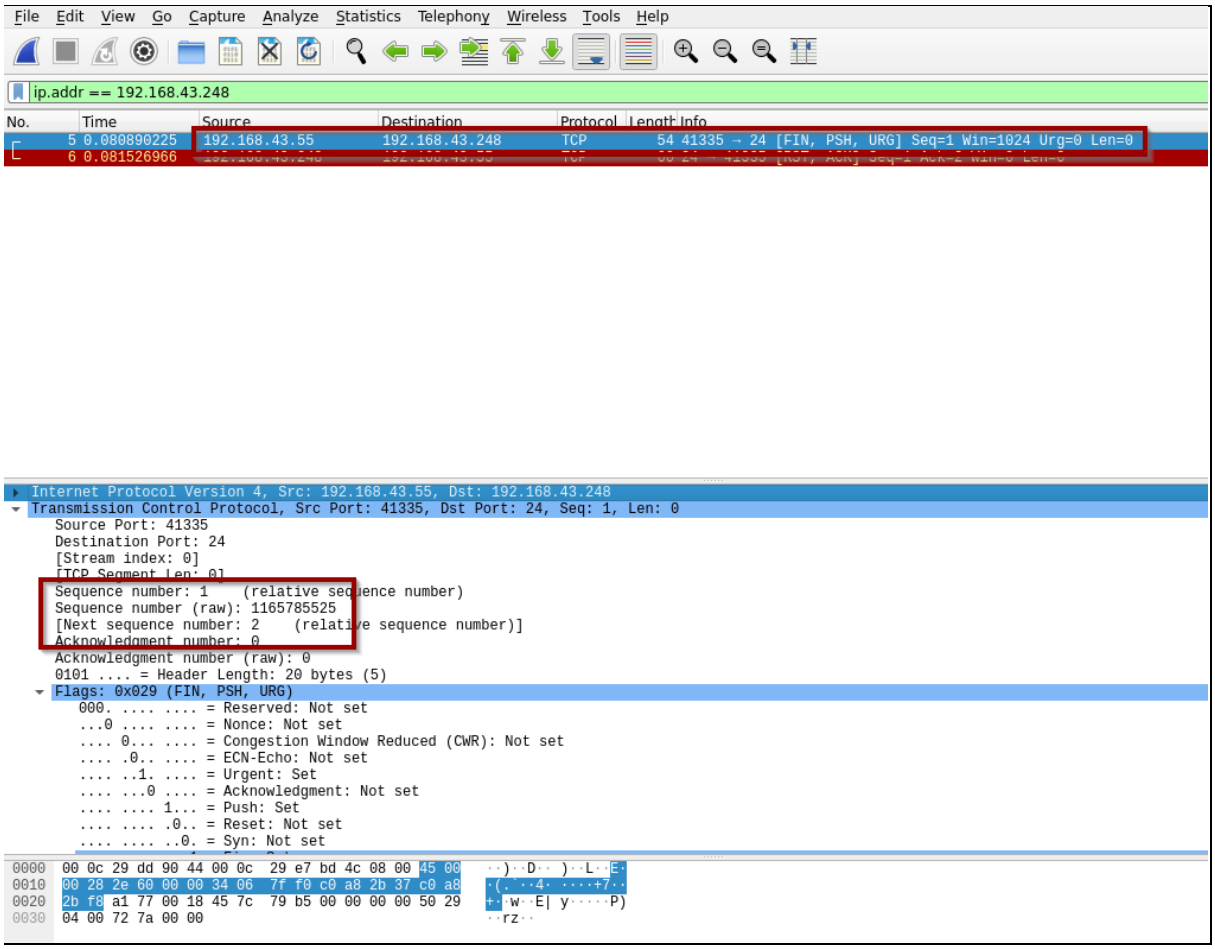
nmap -sX -p <port number> <destination IP>

Nmap scan command

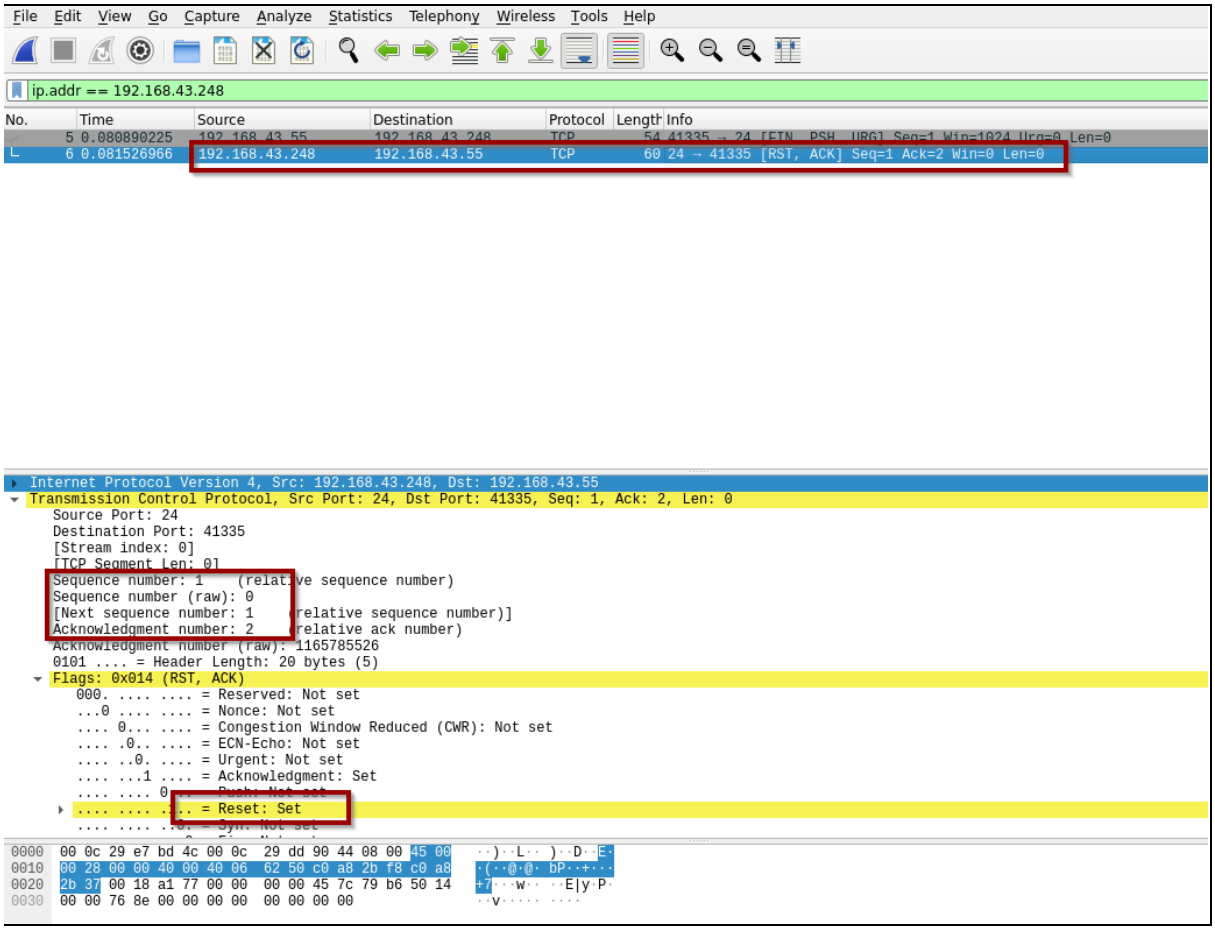
nmap -sX -p 24 192.168.43.248

```
kali@kali:~$ sudo nmap -sX -p 24 192.168.43.248
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 09:02 IST
Nmap scan report for 192.168.43.248
Host is up (0.00100s latency).

PORT   STATE  SERVICE
24/tcp closed priv-mail
MAC Address: 00:0C:29:DD:90:44 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
kali@kali:~$
```

Port is closed which means destination send a response.



First packet contains FIN, PSH and URG flags.
Since the port is closed the destination responds.



UDP SCAN (-sU)

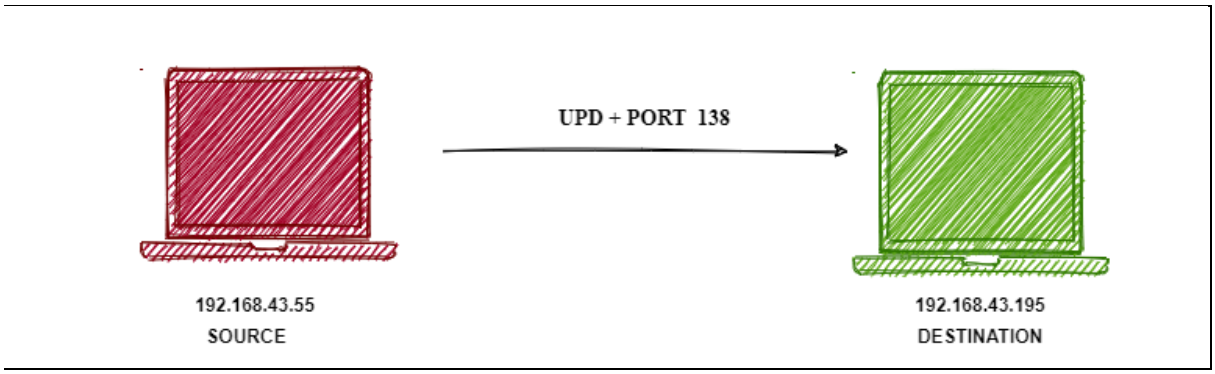
This scan works by sending UDP packets to the destination port, if the target port is open then there is no response, if port is closed then destination sends an ICMP packet saying unreachable.

This scan is also called connection less protocol.

Advantage:

1) More control over data sent out.
2) UPD header is 20 bytes while TCP header is 80 bytes.

For open port



Scan an open port and capture the packets in Wireshark.

Syntax

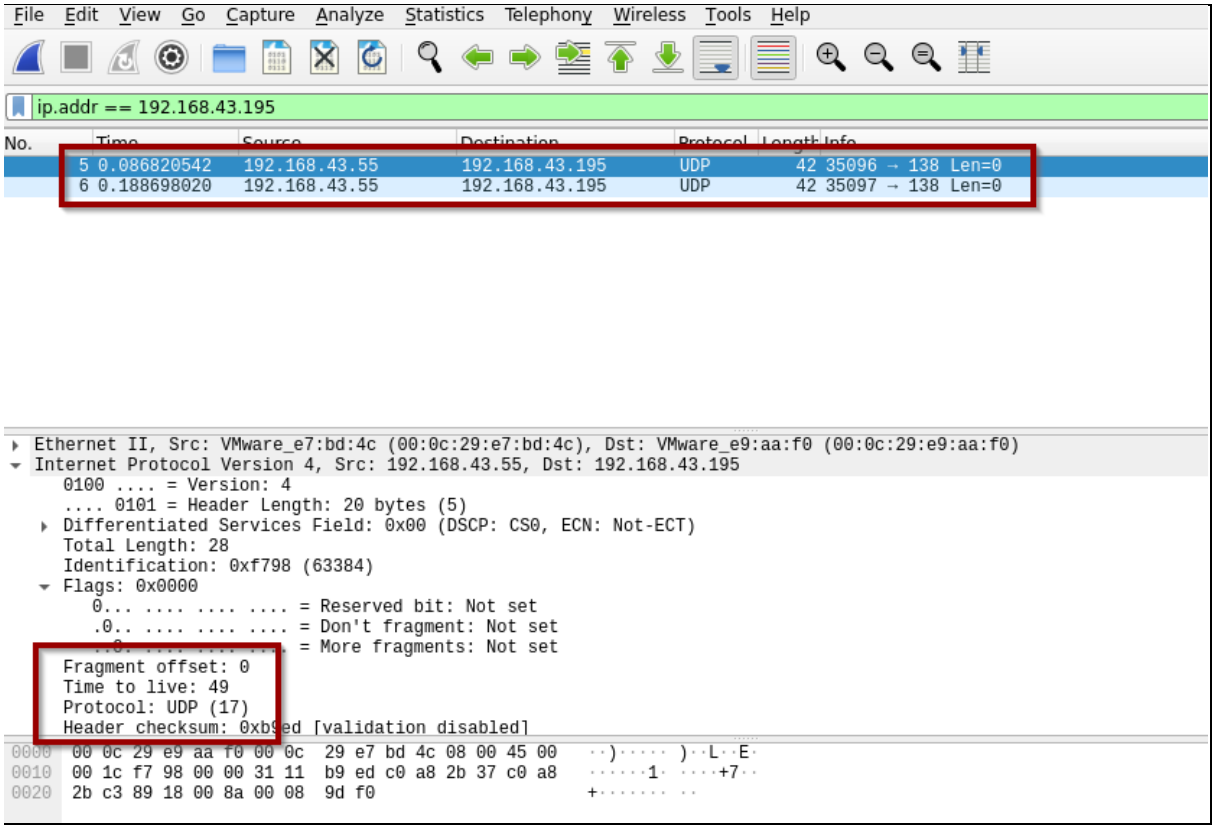
nmap -sU -p <port number> <destination IP>

Nmap scan command

nmap -sU -p 138 192.168.43.195

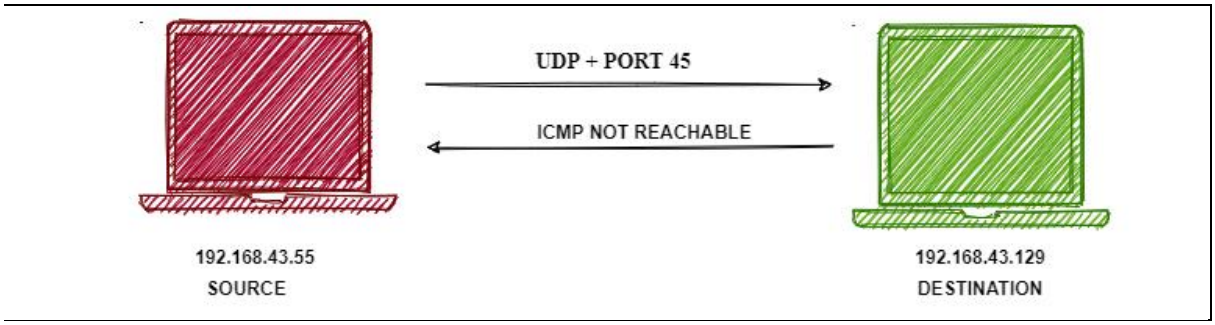
```
kali@kali:~$ sudo nmap -sU -p 138 192.168.43.195
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-27 15:35 IST
Nmap scan report for IRONMAN (192.168.43.195)
Host is up (0.00047s latency).

PORT    STATE         SERVICE
138/udp open|filtered netbios-dgm
MAC Address: 00:0C:29:E9:AA:F0 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.59 seconds
kali@kali:~$
```

Let's see the Wireshark.



We can see that the server is not responding to our UDP packet, in the flag section we can see that protocol used is UDP.

For closed port



Let’s scan a closed port through nmap and analyze the packets through Wireshark.

Syntax

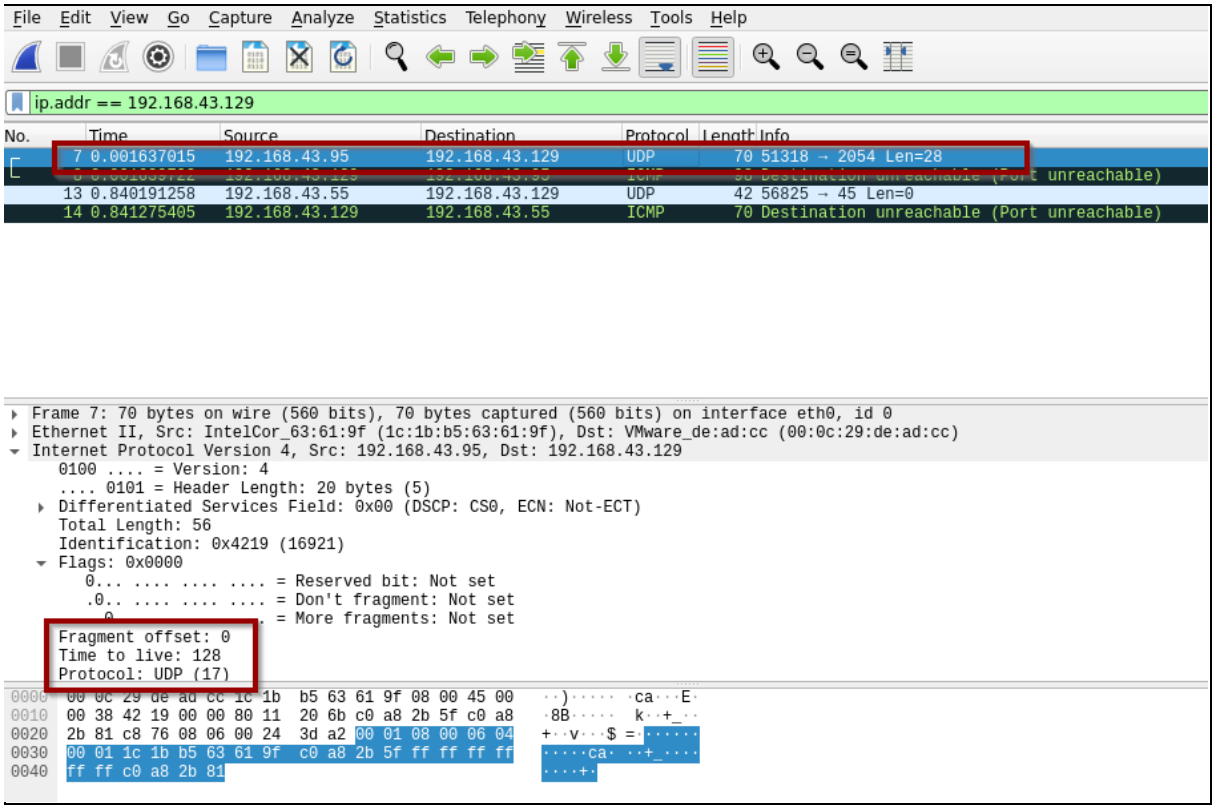
nmap -sU -p <port number> <destination IP>

Nmap command

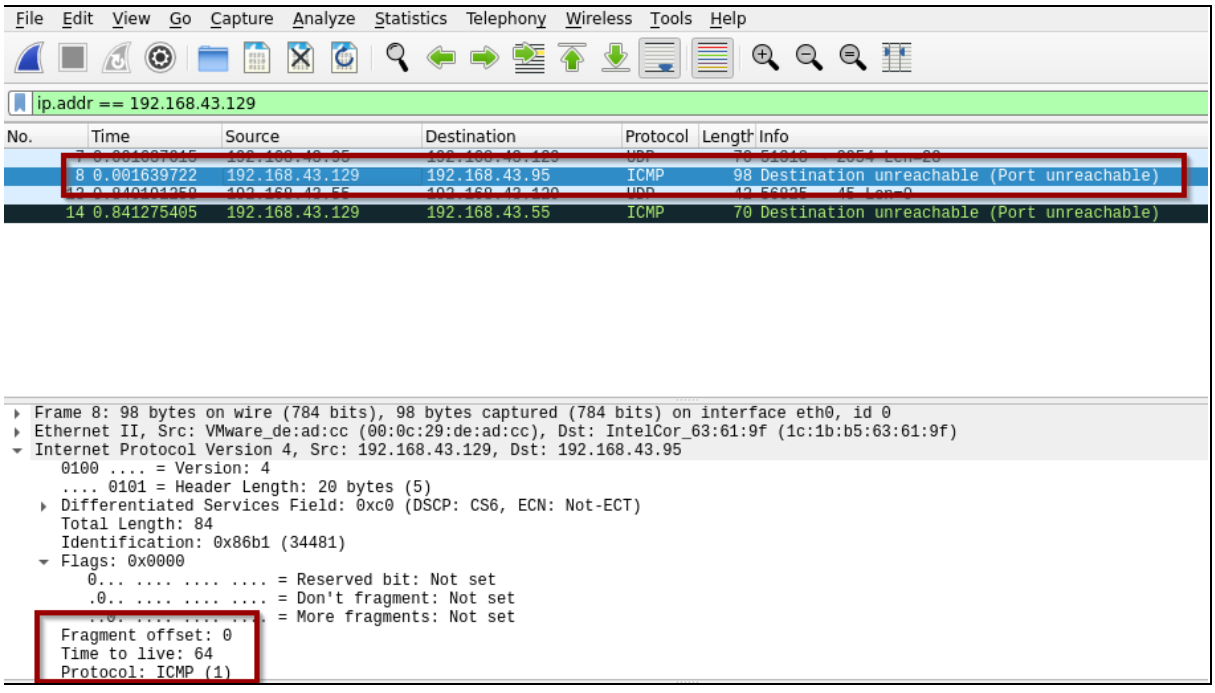
nmap -sU -p 45 192.168.43.129

```
kali@kali:~$ sudo nmap -sU -p 45 192.168.43.129
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-27 16:12 IST
Nmap scan report for owaspbwa (192.168.43.129)
Host is up (0.00088s latency).

PORT   STATE  SERVICE
45/udp closed mpm
MAC Address: 00:0C:29:DE:AD:CC (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
kali@kali:~$
```

Port is closed, let’s see Wireshark.



First UDP packet is sent by us, since the port is closed, we should receive ICMP packet from the server.



We can see the server responds with ICMP packet, and tells us that destination is unreachable.

Idle Scan /Zombie Scan (-sI)

This scan is far more complex as compared to other scans. Idle scan allows complete blind port scanning. The scan is carried out using another host known as “zombie host”. This is somewhat similar to identity theft. The idea is to prevent the attackers IP address to be logged in victims Intrusion Detection System (IDS). This scan involves an attacking machine, a target, a zombie.

Basic understanding: -

1) TCP scan involves attacker sending SYN scan to determine if open is open or not. If port is open the target machine sends SYN/ACK packet, else sends an RST packet.
2) Machine that receives an unsolicited SYN/ACK packet will responds with RST packet.
3) Each IP packet has a fragment Identity Number (IP ID), each packet sent results in increment if IP ID, probing/inspecting these IP ID the attacker can tell how many packets have been sent since last probe. This also allows attacker to forge his identity and perform blind scan.

This scan is carried out in three steps: -

1) Probe the zombie IPID and note it.
2) Forge a SYN packet from the zombie and send it to the desired port to the target. The target will respond on basis of open or close port which may or may not cause the zombie IPID to be incremented (Increase in IPID is because of SYN/ACK packet received by zombie and encouraging the zombie to send RST packet).
3) Probe the zombie IPID again. The port state is determined by comparing the current IPID with the probe IPID record in step 1.

Finding zombie/idle machine:

Nmap is a very power full tool. It becomes even more power full when you use nmap scripts (.nse) effectively. This script detects vulnerable zombie devices. Idle scan uses port 80 as default port but you can also manually set the port too, we also need to provide range/number of machines we want to scan (high range means greater scan time).

Nmap scan

Syntax:

nmap -p<port scan> --script ipidseq -iR <range>

Nmap Scan Command

nmap -p80 --script ipidseq -iR 100

(-iR → Random IP address scans)

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -p80 --script ipidseq -iR 1000
Starting Nmap 7.91 ( https://nmap.org ) at 2020-11-01 11:51 IST
Nmap scan report for 177-22-12-48.alfanetwork.com.br (177.22.12.48)
Host is up (0.58s latency).

PORT   STATE    SERVICE
80/tcp filtered http

Nmap scan report for softbank126125080030.bbtec.net (126.125.80.30)
Host is up (0.079s latency).

PORT   STATE    SERVICE
80/tcp filtered http

Nmap scan report for 100.55.62.203
Host is up (0.066s latency).

PORT   STATE    SERVICE
80/tcp filtered http

Nmap scan report for 149.204.140.35
Host is up (0.071s latency).

PORT   STATE    SERVICE
80/tcp filtered http

Nmap scan report for c-67-188-37-175.hsd1.ca.comcast.net (67.188.37.175)
Host is up (0.32s latency).
```

The scan gives out a lot of result but what we are looking for is results in which ipidseq was detected as incremental and port is open.

```
PORT   STATE    SERVICE
80/tcp filtered http

Nmap scan report for 40.69.66.10
Host is up (0.30s latency).

PORT   STATE SERVICE
80/tcp open  http

Host script results:
|_ipidseq: Incremental!

Nmap scan report for 16.63.149.98
Host is up (0.11s latency).

PORT   STATE    SERVICE
80/tcp filtered http
```

Note down the corresponding IP address and use the as zombie.
The downside is

1) Most of the time the ipidseq shows Random, All zeros..etc, this maybe be due to reasons like the firewall or not vulnerable..etc.
2) Most of Operating Systems assign IP ID randomly.

3) Well configured firewalls and honeypots may return false positive.

For explanation I used the following IP address

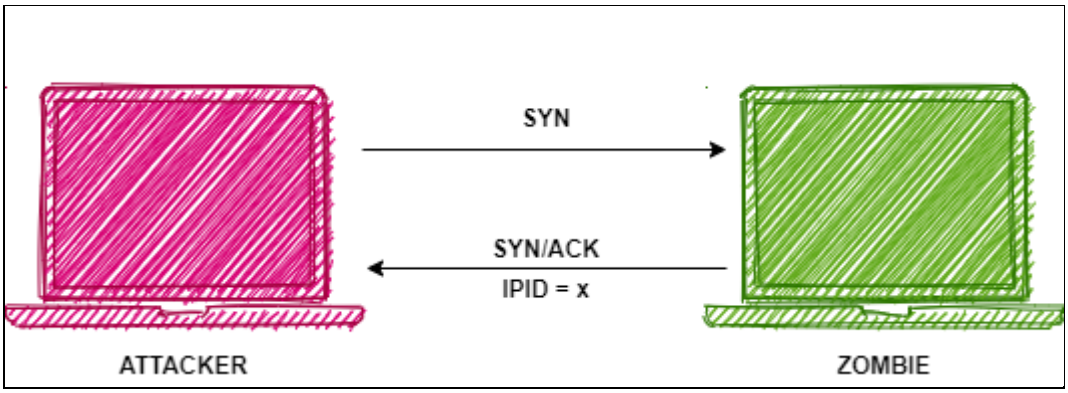
Attacker IP 192.168.43.55

Zombie IP 192.168.43.195

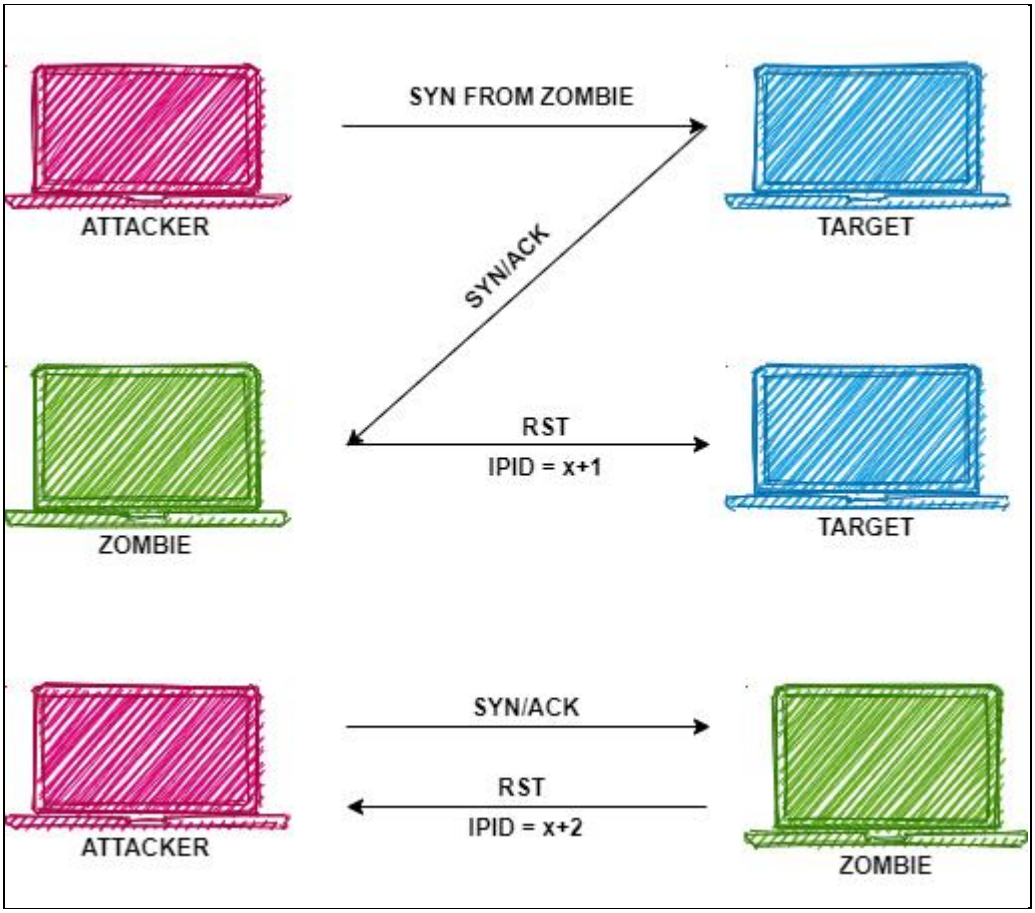
Target IP 192.168.43.248

For open ports

Step 1: Probe zombie IPID



Step 2: Forge a SYN request from zombie and send it to target.



Step 3: Probe the zombie IPID again. If the IPID got increased by 2 from the previous IPID, then it suggests that the port was open.

Syntax:

nmap -sI <zombie IP address> -p<port number> <target machine IP>

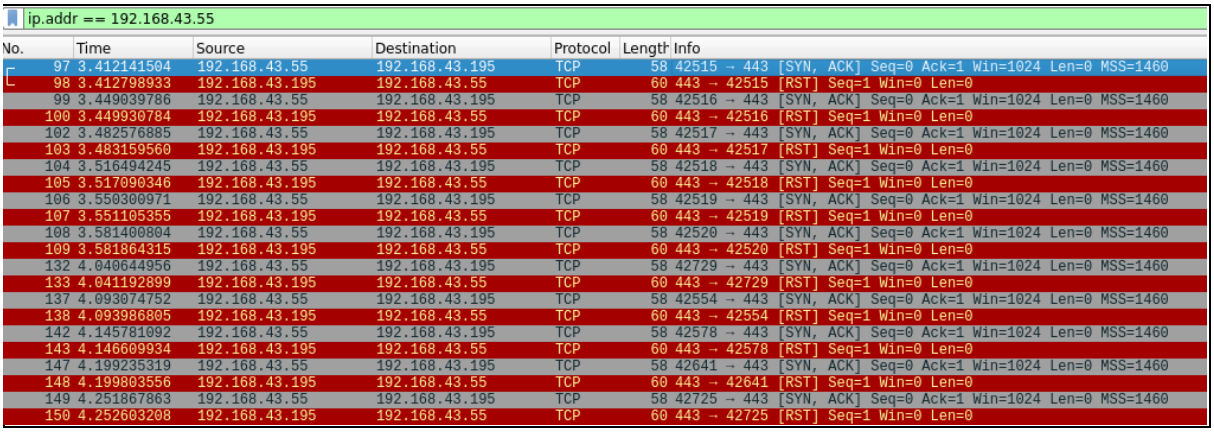
Nmap Scan Command

nmap -sI 192.168.43.195 -p80 192.168.43.248

```
└─$ sudo nmap -sI 192.168.43.195 -p80 192.168.43.248
WARNING: Many people use -Pn w/Idlescan to prevent pings from their true IP.  On the other hand, timing info Nmap gains from pings can a
llow for faster, more reliable scans.
Starting Nmap 7.91 ( https://nmap.org ) at 2020-10-31 17:54 IST
Idle scan using zombie 192.168.43.195 (192.168.43.195:443); Class: Incremental
Nmap scan report for 192.168.43.248
Host is up (0.0073s latency).

PORT   STATE SERVICE
80/tcp open  http
MAC Address: 00:0C:29:DD:90:44 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 3.08 seconds
```

Wireshark analysis

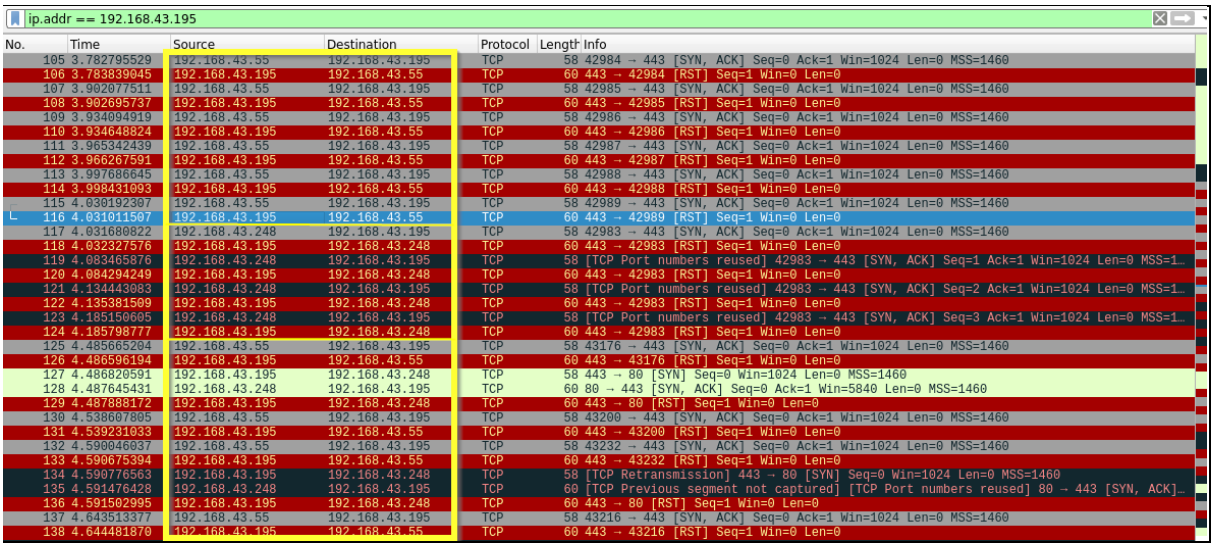
The screenshot given below shows an open port scan. I filtered the scan to show that the attacker never directly communicates with the target.



ip.addr == 192.168.43.55

No.	Time	Source	Destination	Protocol	Length	Info
97	3.412141504	192.168.43.55	192.168.43.195	TCP	58	42515 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
98	3.412798933	192.168.43.195	192.168.43.55	TCP	60	443 → 42515 [RST] Seq=1 Win=0 Len=0
99	3.449039786	192.168.43.55	192.168.43.195	TCP	58	42516 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
100	3.449930784	192.168.43.195	192.168.43.55	TCP	60	443 → 42516 [RST] Seq=1 Win=0 Len=0
102	3.482576885	192.168.43.55	192.168.43.195	TCP	58	42517 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
103	3.483159560	192.168.43.195	192.168.43.55	TCP	60	443 → 42517 [RST] Seq=1 Win=0 Len=0
104	3.516494245	192.168.43.55	192.168.43.195	TCP	58	42518 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
105	3.517090346	192.168.43.195	192.168.43.55	TCP	60	443 → 42518 [RST] Seq=1 Win=0 Len=0
106	3.550300971	192.168.43.55	192.168.43.195	TCP	58	42519 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
107	3.551105355	192.168.43.195	192.168.43.55	TCP	60	443 → 42519 [RST] Seq=1 Win=0 Len=0
108	3.581400804	192.168.43.55	192.168.43.195	TCP	58	42520 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
109	3.581864315	192.168.43.195	192.168.43.55	TCP	60	443 → 42520 [RST] Seq=1 Win=0 Len=0
132	4.040644956	192.168.43.55	192.168.43.195	TCP	58	42729 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
133	4.041192899	192.168.43.195	192.168.43.55	TCP	60	443 → 42729 [RST] Seq=1 Win=0 Len=0
137	4.093074752	192.168.43.55	192.168.43.195	TCP	58	42554 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
138	4.093986805	192.168.43.195	192.168.43.55	TCP	60	443 → 42554 [RST] Seq=1 Win=0 Len=0
142	4.145781092	192.168.43.55	192.168.43.195	TCP	58	42578 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
143	4.146609934	192.168.43.195	192.168.43.55	TCP	60	443 → 42578 [RST] Seq=1 Win=0 Len=0
147	4.199235319	192.168.43.55	192.168.43.195	TCP	58	42641 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
148	4.199803556	192.168.43.195	192.168.43.55	TCP	60	443 → 42641 [RST] Seq=1 Win=0 Len=0
149	4.251867863	192.168.43.55	192.168.43.195	TCP	58	42725 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
150	4.252603208	192.168.43.195	192.168.43.55	TCP	60	443 → 42725 [RST] Seq=1 Win=0 Len=0

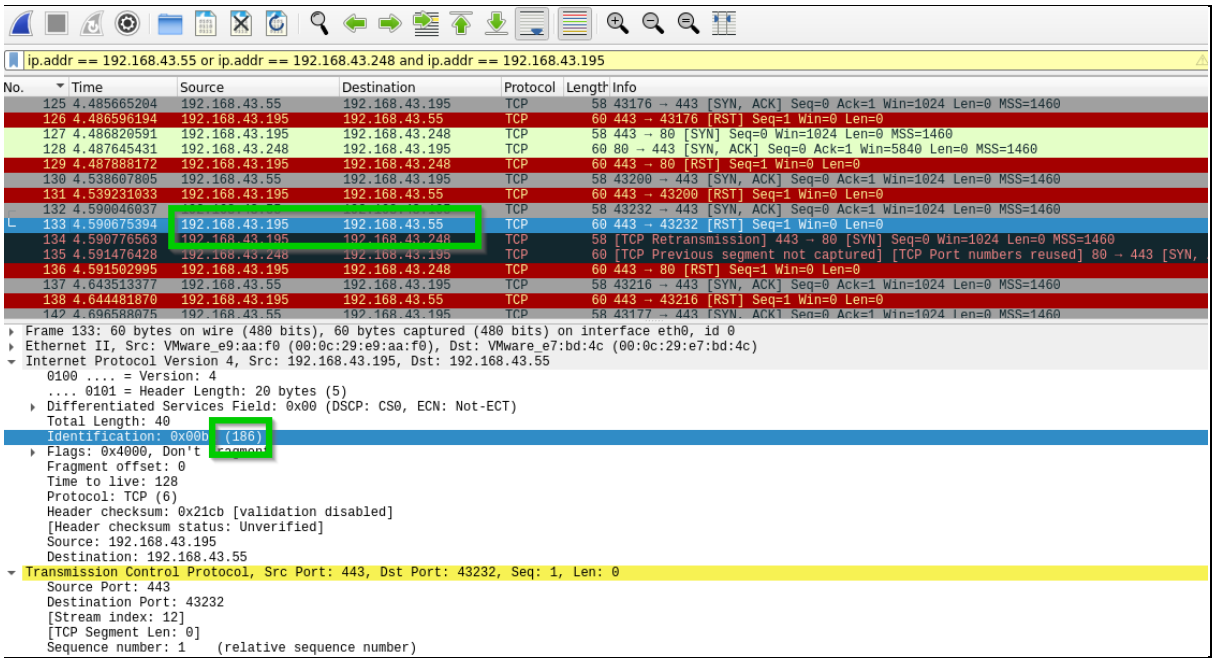
The screenshot below shows the phases in which scan happens.
The first slot show communication between attacker and zombie.
The second slot is where the actual scan takes place. The zombie communicates with the target.
The third slot is communication between zombie and attacker to verify port status.



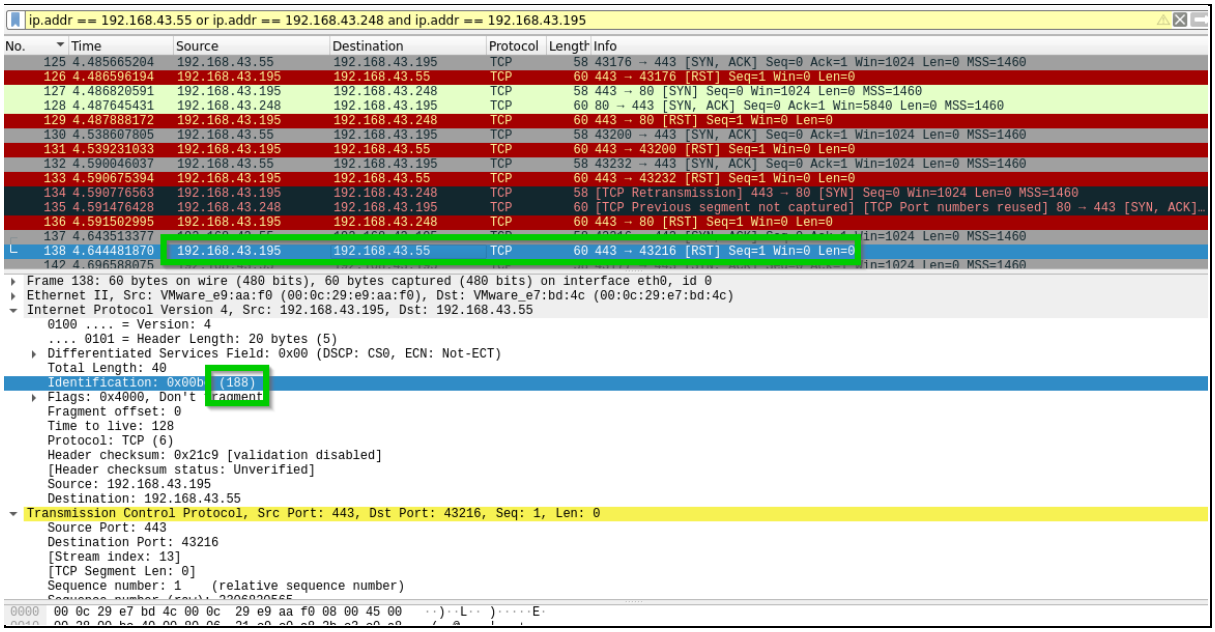
ip.addr == 192.168.43.195

No.	Time	Source	Destination	Protocol	Length	Info
105	3.782795529	192.168.43.55	192.168.43.195	TCP	58	42984 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
106	3.783839045	192.168.43.195	192.168.43.55	TCP	60	443 → 42984 [RST] Seq=1 Win=0 Len=0
107	3.902077511	192.168.43.55	192.168.43.195	TCP	58	42985 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
108	3.902695737	192.168.43.195	192.168.43.55	TCP	60	443 → 42985 [RST] Seq=1 Win=0 Len=0
109	3.934094919	192.168.43.55	192.168.43.195	TCP	58	42986 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
110	3.934648824	192.168.43.195	192.168.43.55	TCP	60	443 → 42986 [RST] Seq=1 Win=0 Len=0
111	3.965342439	192.168.43.55	192.168.43.195	TCP	58	42987 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
112	3.966267591	192.168.43.195	192.168.43.55	TCP	60	443 → 42987 [RST] Seq=1 Win=0 Len=0
113	3.997686645	192.168.43.55	192.168.43.195	TCP	58	42988 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
114	3.998431093	192.168.43.195	192.168.43.55	TCP	60	443 → 42988 [RST] Seq=1 Win=0 Len=0
115	4.030192307	192.168.43.55	192.168.43.195	TCP	58	42989 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
116	4.031011507	192.168.43.195	192.168.43.55	TCP	60	443 → 42989 [RST] Seq=1 Win=0 Len=0
117	4.031680922	192.168.43.248	192.168.43.195	TCP	58	42983 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
118	4.032327576	192.168.43.195	192.168.43.248	TCP	60	443 → 42983 [RST] Seq=1 Win=0 Len=0
119	4.083465876	192.168.43.248	192.168.43.195	TCP	58	[TCP Port numbers reused] 42983 → 443 [SYN, ACK] Seq=1 Ack=1 Win=1024 Len=0 MSS=1…
120	4.084294249	192.168.43.195	192.168.43.248	TCP	60	443 → 42983 [RST] Seq=1 Win=0 Len=0
121	4.134443083	192.168.43.248	192.168.43.195	TCP	58	[TCP Port numbers reused] 42983 → 443 [SYN, ACK] Seq=2 Ack=1 Win=1024 Len=0 MSS=1…
122	4.135381509	192.168.43.195	192.168.43.248	TCP	60	443 → 42983 [RST] Seq=1 Win=0 Len=0
123	4.185150605	192.168.43.248	192.168.43.195	TCP	58	[TCP Port numbers reused] 42983 → 443 [SYN, ACK] Seq=3 Ack=1 Win=1024 Len=0 MSS=1…
124	4.185798777	192.168.43.195	192.168.43.248	TCP	60	443 → 42983 [RST] Seq=1 Win=0 Len=0
125	4.485665204	192.168.43.55	192.168.43.195	TCP	58	43176 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
126	4.486596194	192.168.43.195	192.168.43.55	TCP	60	443 → 43176 [RST] Seq=1 Win=0 Len=0
127	4.486820591	192.168.43.195	192.168.43.248	TCP	58	443 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
128	4.487645431	192.168.43.248	192.168.43.195	TCP	60	80 → 443 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
129	4.487888172	192.168.43.195	192.168.43.248	TCP	60	443 → 80 [RST] Seq=1 Win=0 Len=0
130	4.538607805	192.168.43.55	192.168.43.195	TCP	58	43200 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
131	4.539231033	192.168.43.195	192.168.43.55	TCP	60	443 → 43200 [RST] Seq=1 Win=0 Len=0
132	4.590046037	192.168.43.55	192.168.43.195	TCP	58	43232 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
133	4.590675394	192.168.43.195	192.168.43.55	TCP	60	443 → 43232 [RST] Seq=1 Win=0 Len=0
134	4.590776563	192.168.43.195	192.168.43.248	TCP	58	[TCP Retransmission] 443 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
135	4.591476428	192.168.43.248	192.168.43.195	TCP	60	[TCP Previous segment not captured] [TCP Port numbers reused] 80 → 443 [SYN, ACK]…
136	4.591502995	192.168.43.195	192.168.43.248	TCP	60	443 → 80 [RST] Seq=1 Win=0 Len=0
137	4.643513377	192.168.43.55	192.168.43.195	TCP	58	43216 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
138	4.644481870	192.168.43.195	192.168.43.55	TCP	60	443 → 43216 [RST] Seq=1 Win=0 Len=0

Step 1: Record the Identification number present in Internet Protocol Version Section. Pay close attention to which Request you should analyze. The marked one is the last request that communicates with the attacker and sends the RST packet and this should be used for analysis.



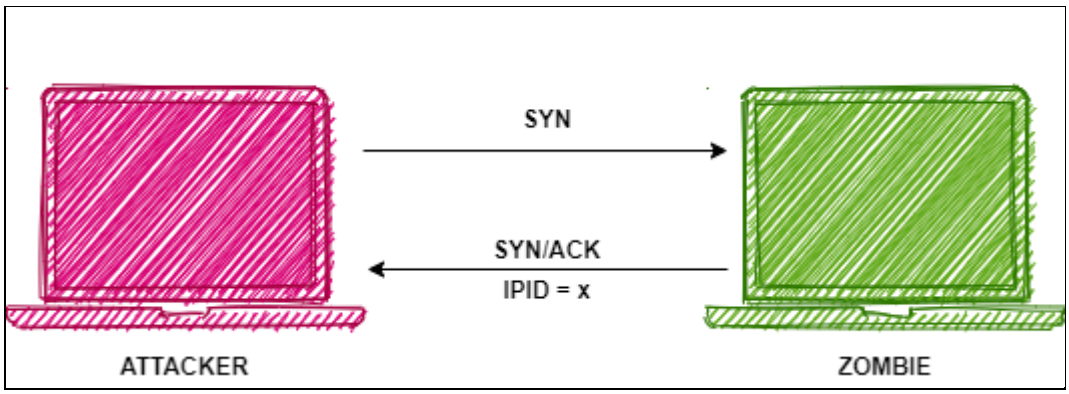
Step 2 and 3: After the RST packet to attacker the zombie scans the target (Retransmission step). After successful scan the zombie returns to communicate with attacker. Note the Identification number again. See the increase from previous number, its 188-186=2



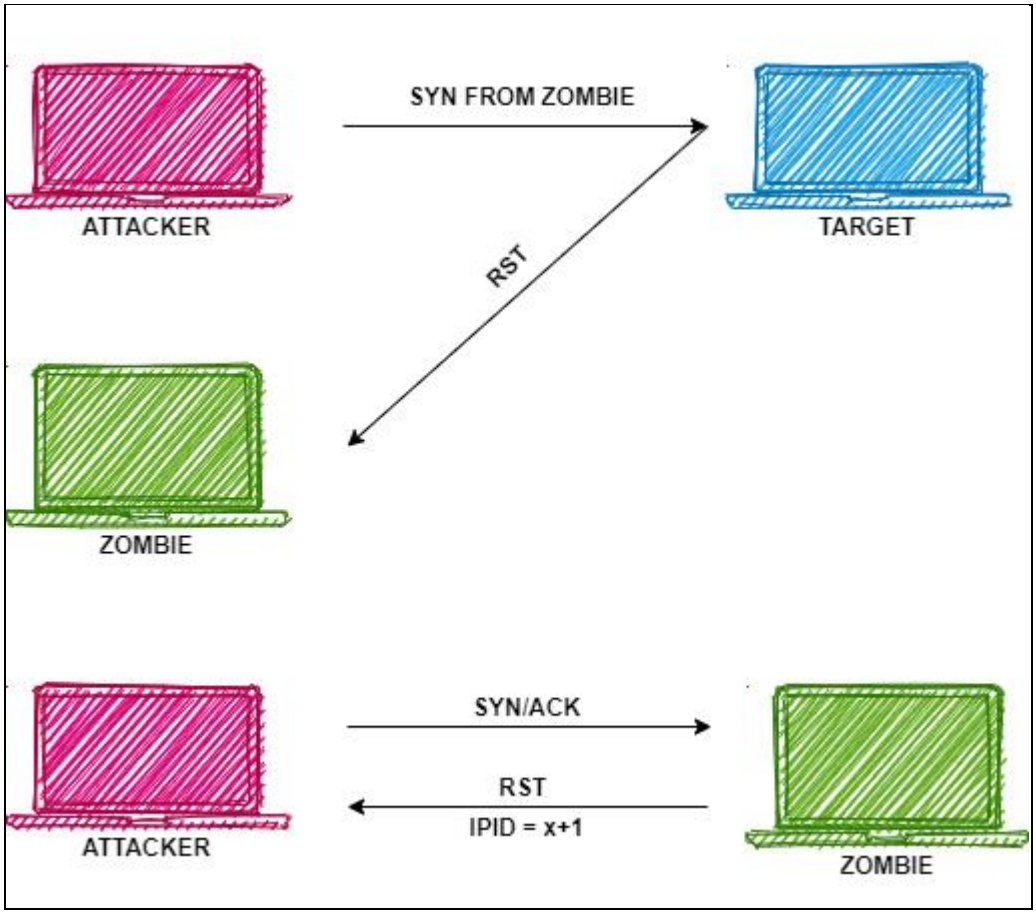
This difference signifies that the port is open for connection.

For closed port

Step 1: Probe the zombie IPID.



Step 2: Forge a SYN packet from zombie and send it to target.



Step 3: Probe the zombie IPID again. If the IPID got increased by 1 from the previous IPID, then it suggests that the port was closed.

Syntax:

nmap -sI <zombie IP> -p<port number> <target IP>

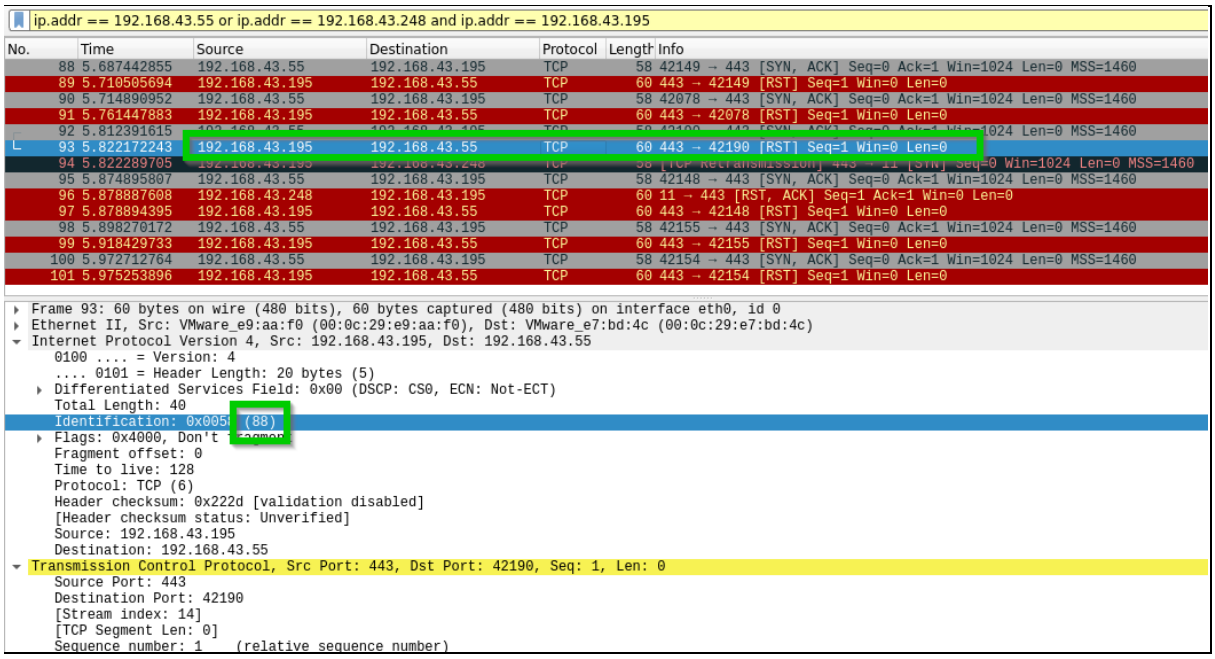
Nmap scan:

nmap -sI 192.168.43.195 -p11 192.168.43.248

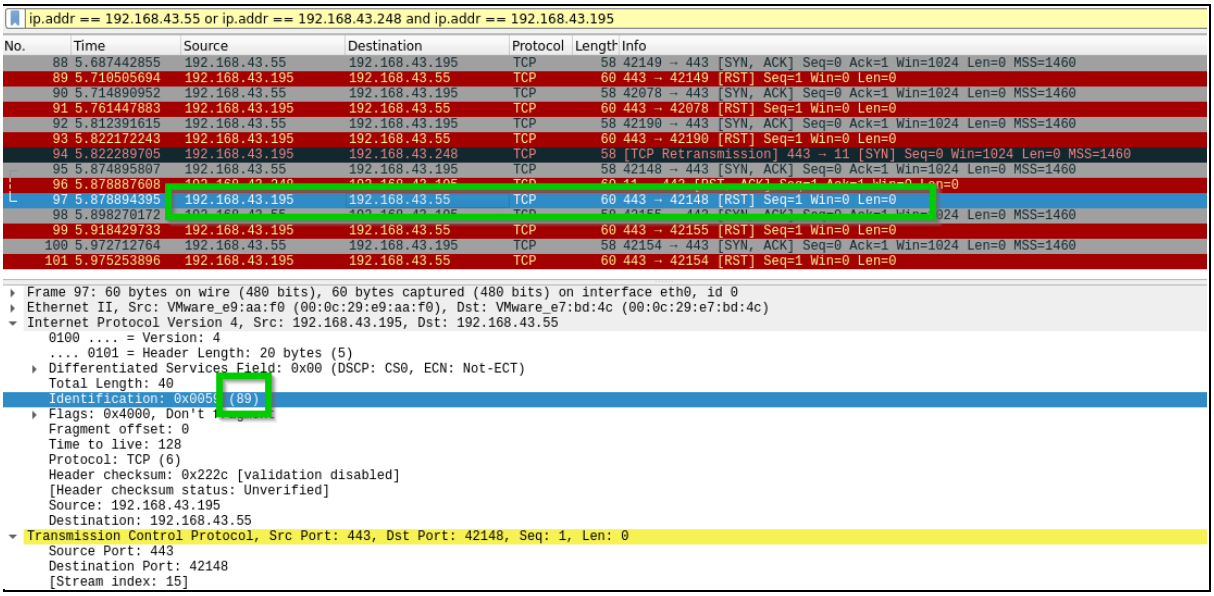
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sI 192.168.43.195 -p11 192.168.43.248
WARNING: Many people use -Pn w/Idlescan to prevent pings from their true IP.  On the other hand, timing info Nmap gains from pings can a
llow for faster, more reliable scans.
Starting Nmap 7.91 ( https://nmap.org ) at 2020-10-31 17:55 IST
Idle scan using zombie 192.168.43.195 (192.168.43.195:443); Class: Incremental
Nmap scan report for 192.168.43.248
Host is up (0.011s latency).

PORT   STATE           SERVICE
11/tcp closed|filtered systat
MAC Address: 00:0C:29:DD:90:44 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 4.23 seconds
```

Wireshark analysis:

Step 1: Similar to the previous one, analysis the request in which the zombie sends the last RST packet to attacker. Note down the Identification number.



Step 2 and 3: The next request is communication between zombie and target. After the scan is performed the zombie communicates with attacker and we note the Identification Number again. Calculate the difference between both the numbers i.e. 89-88=1



This difference of 1signifies that the port is closed.

Advantage:

1) The attacker IP never gets logged in victims IDS (Intrusion Detection System).
2) This is very stealthy scan.

Disadvantage:

1) Complex scan.
2) The attacker IP is still logged in zombie so there's always a chance of traceback.
3) Not easy to find zombie/idle machines.