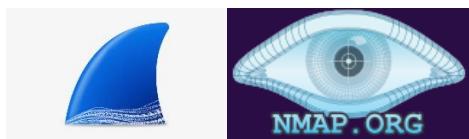


## WIRESHARK ANALYSIS OF VARIOUS NMAP SCANS



**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.

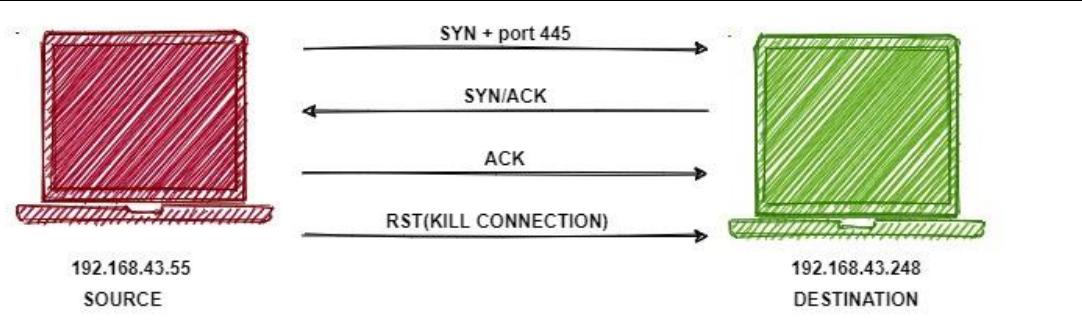
#### **Advantage:**

- 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
```

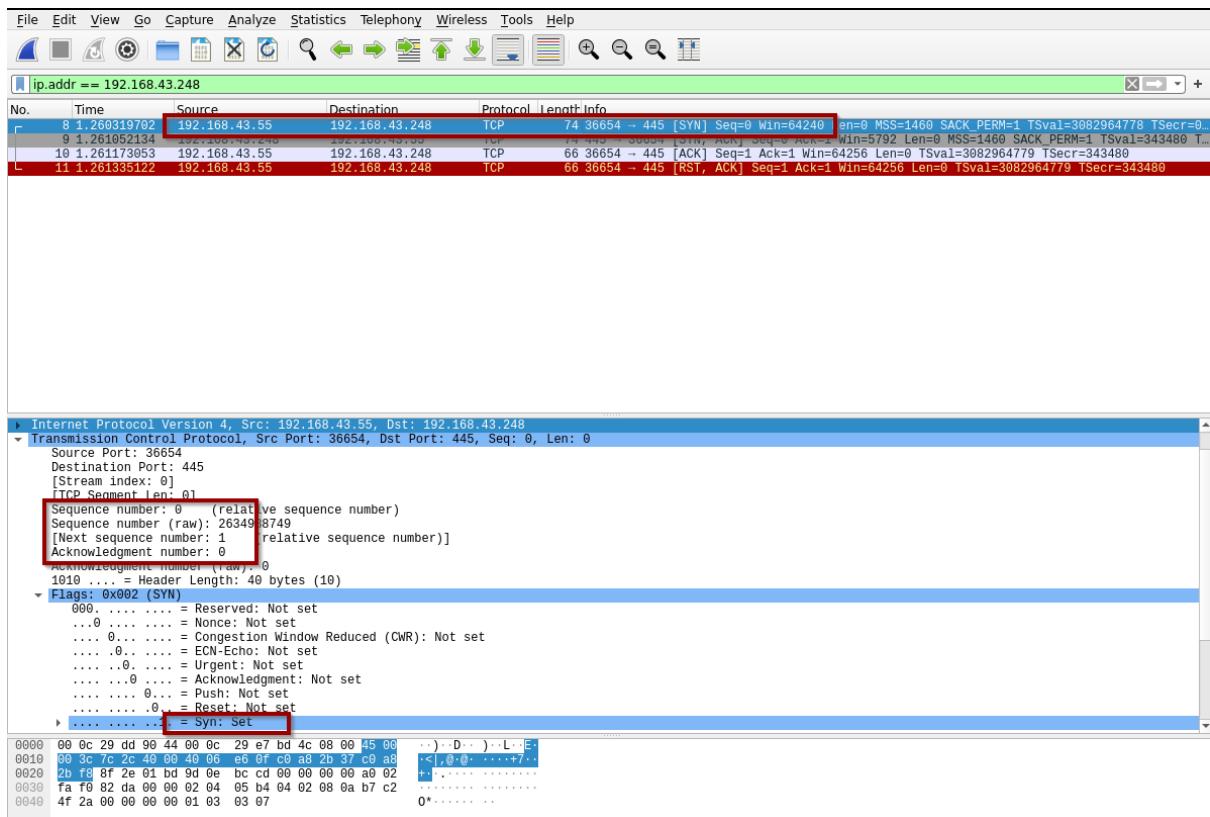
```
kali@kali:~$ sudo nmap -sT -p 445 192.168.43.248
[sudo] password for kali:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-19 19:46 IST
Nmap scan report for 192.168.43.248
Host is up (0.00088s latency).

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.

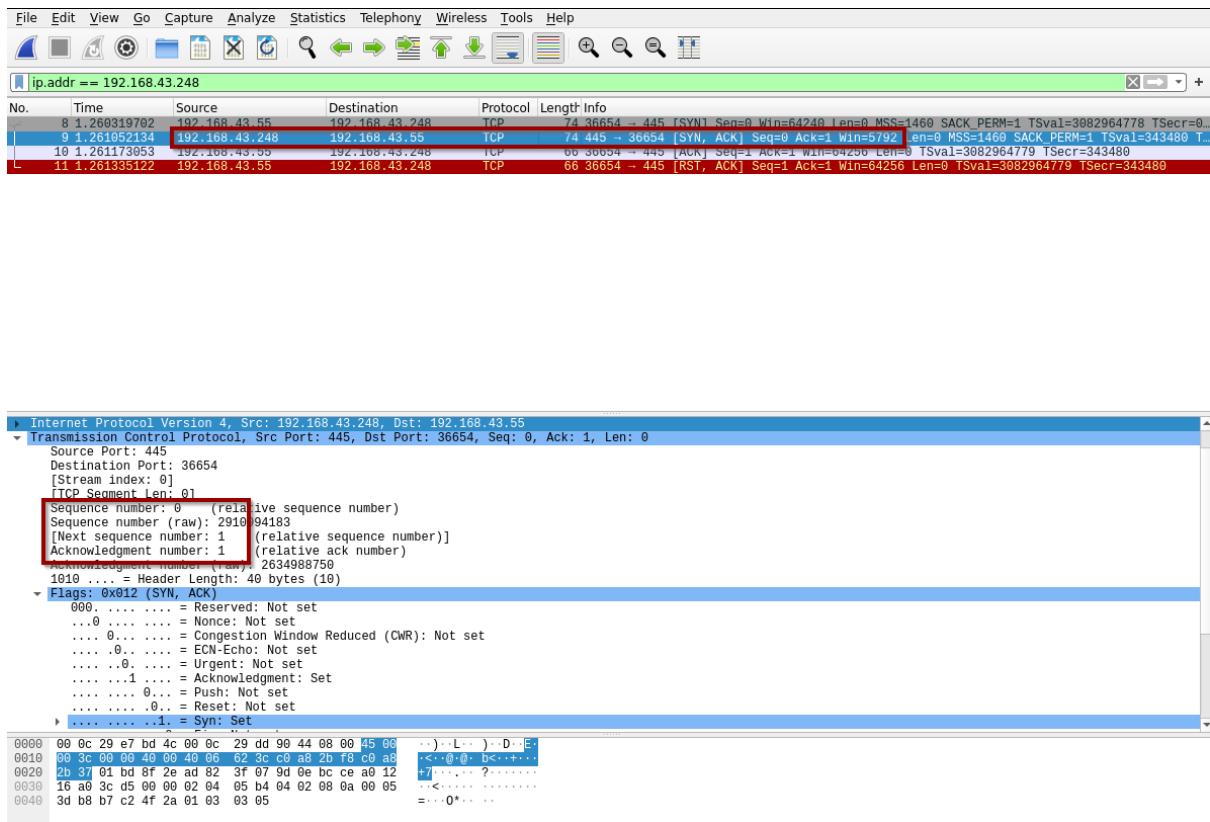


We added a Wireshark filter `ip.addr == 192.168.43.248` to filter packets which deals with 192.168.43.248.

So total of 4 packets are captured, that's what we expected.

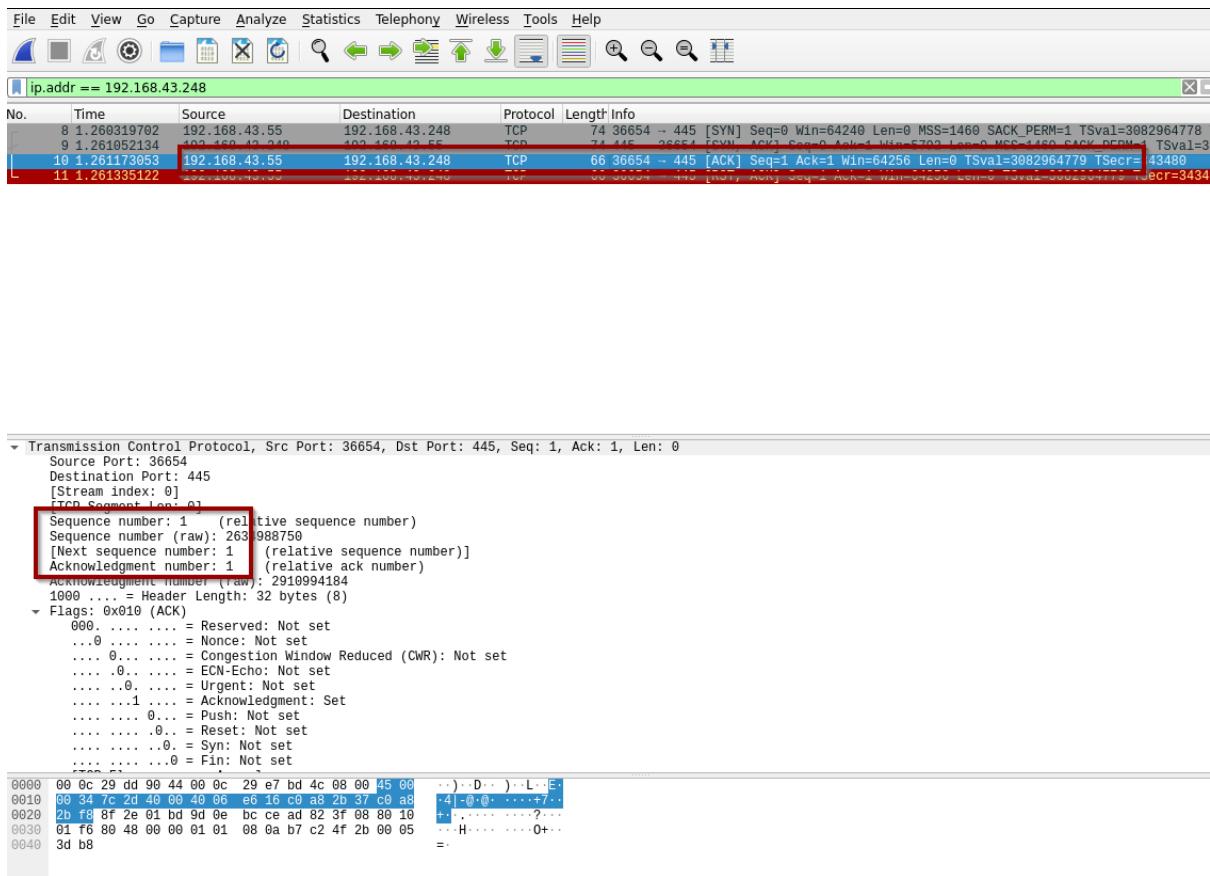
Let's analyze the first packet, we see that SYN packet is sent to the destination, in the flag section we see SYN flag is set.

Let's see if the server responds.



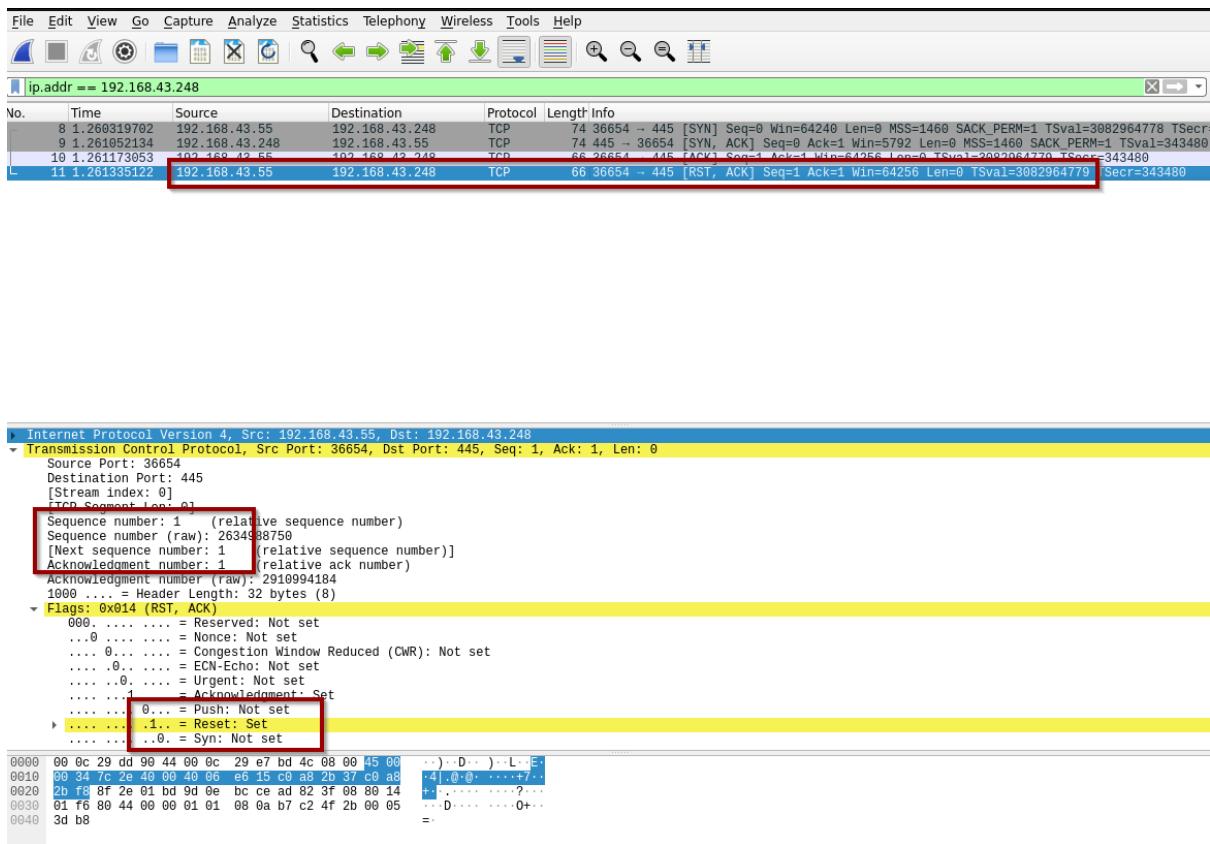
In the second packet the server sends an ACK (responds for SYN packet 1) and send SYN packet to the source for his validation. We can see the sequence number is 0 and Acknowledgement number is 1.

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.

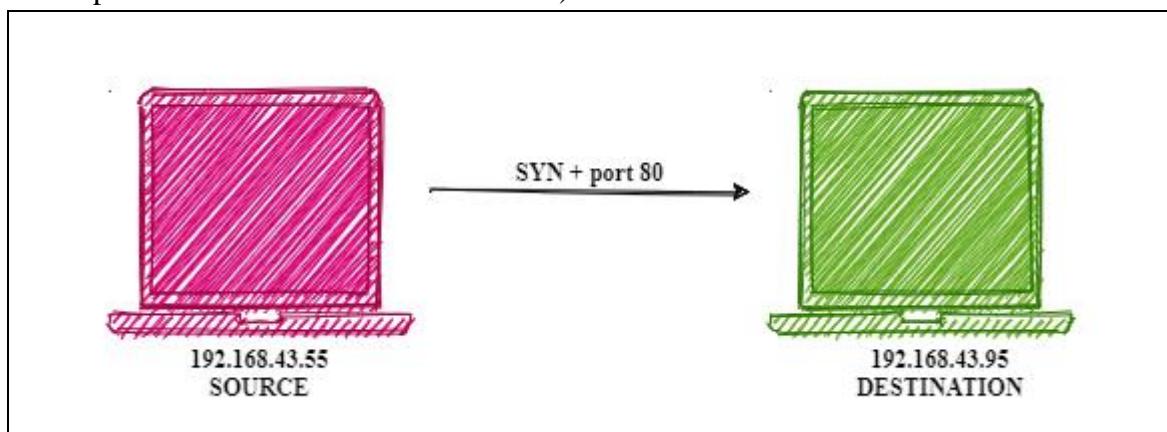
The last packet completes the connection and then resets it.



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>**

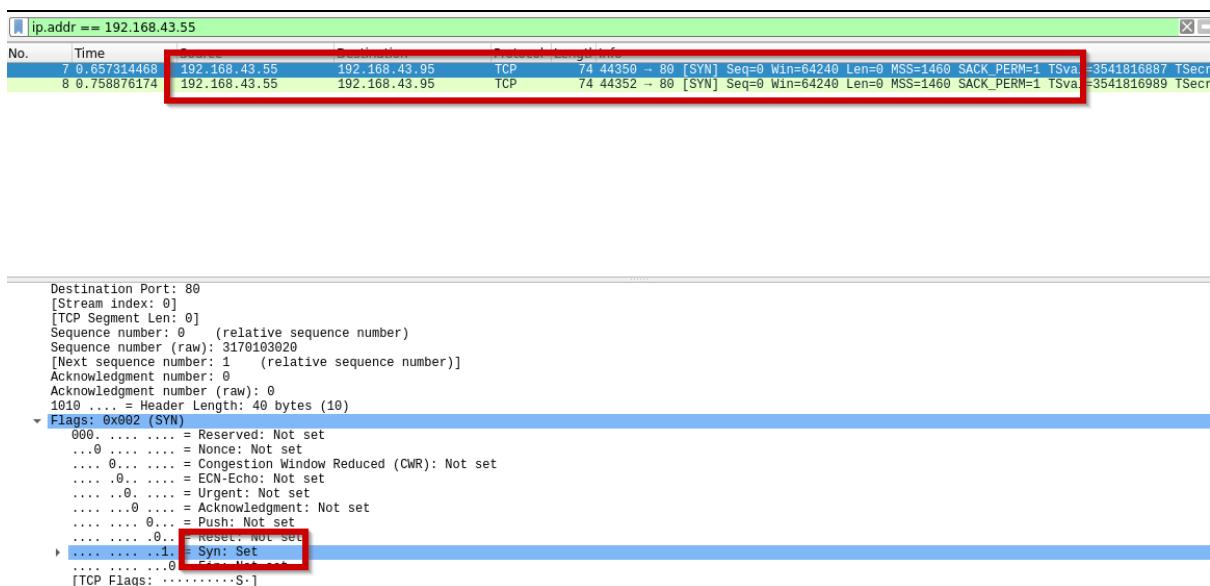
## Nmap command:

**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).

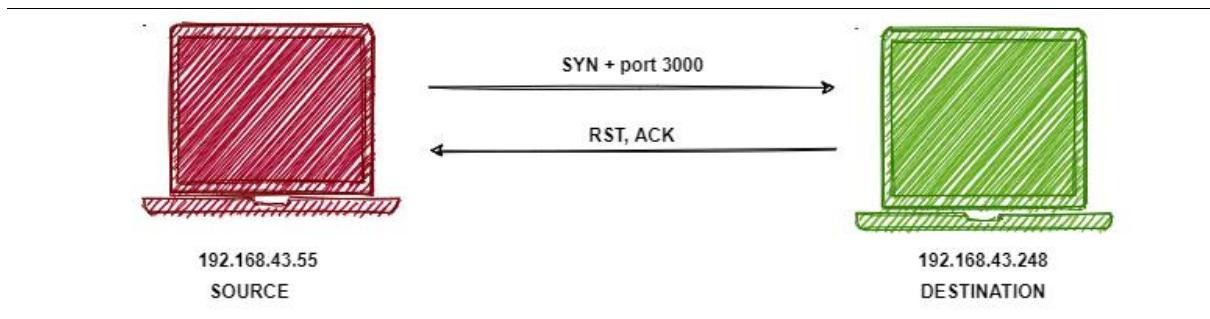
PORT      STATE      SERVICE
80/tcp    filtered  http
```

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`

```

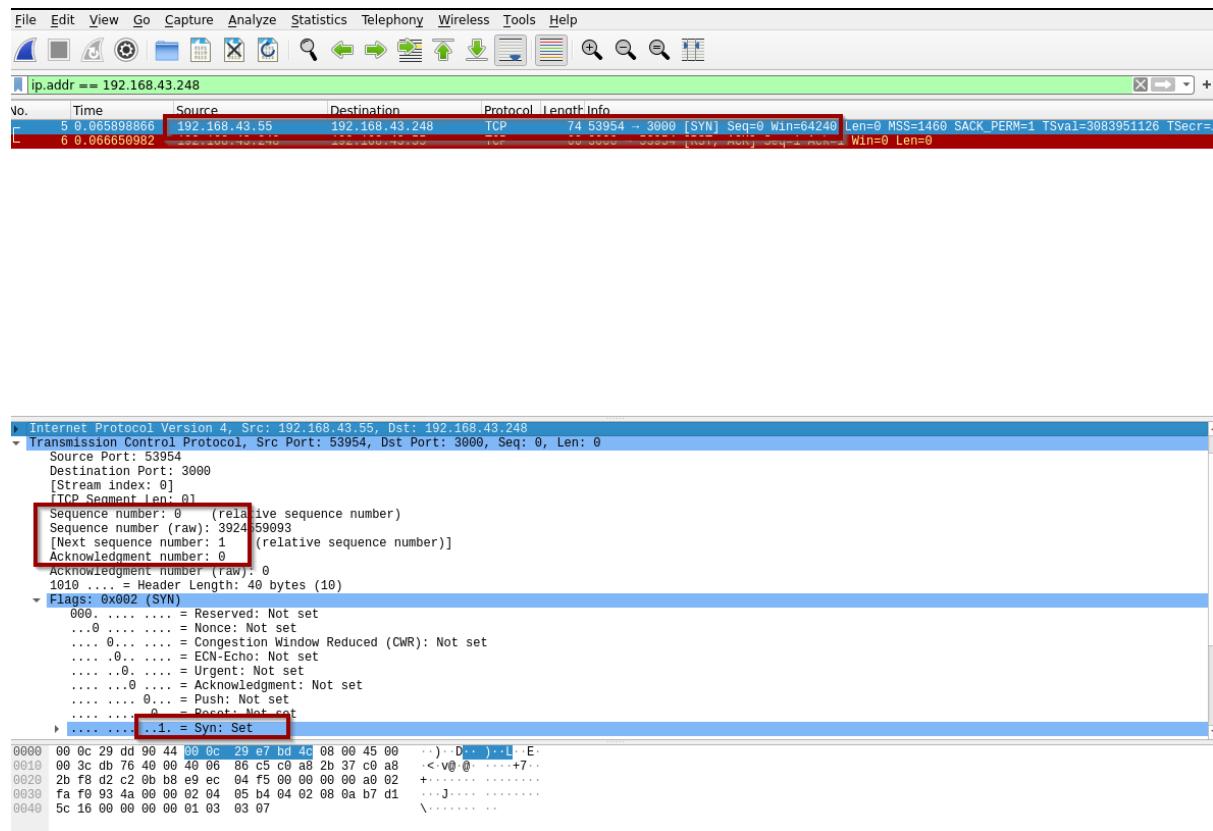
kali@kali:~$ sudo nmap -sT -p 3000 192.168.43.248
[sudo] password for kali:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-19 20:02 IST
Nmap scan report for 192.168.43.248
Host is up (0.00092s latency).

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

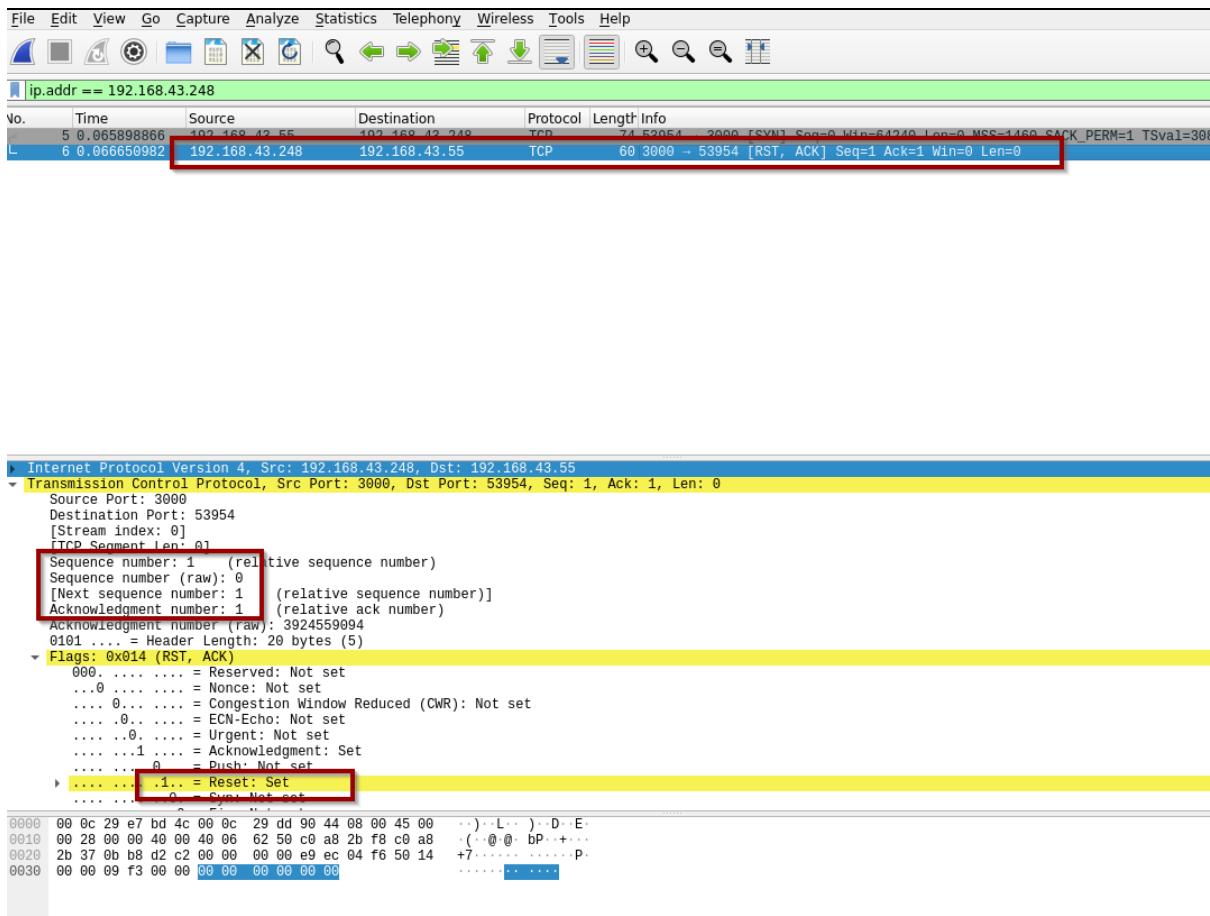
Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
kali@kali:~$ █

```

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.

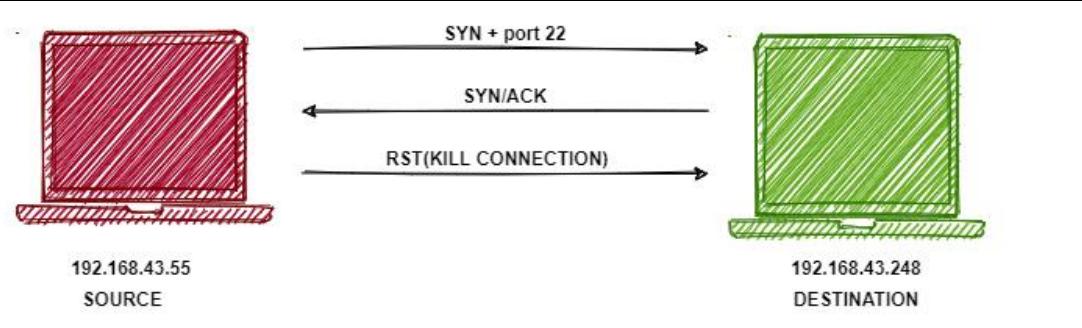
### Advantage:

- 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**

```

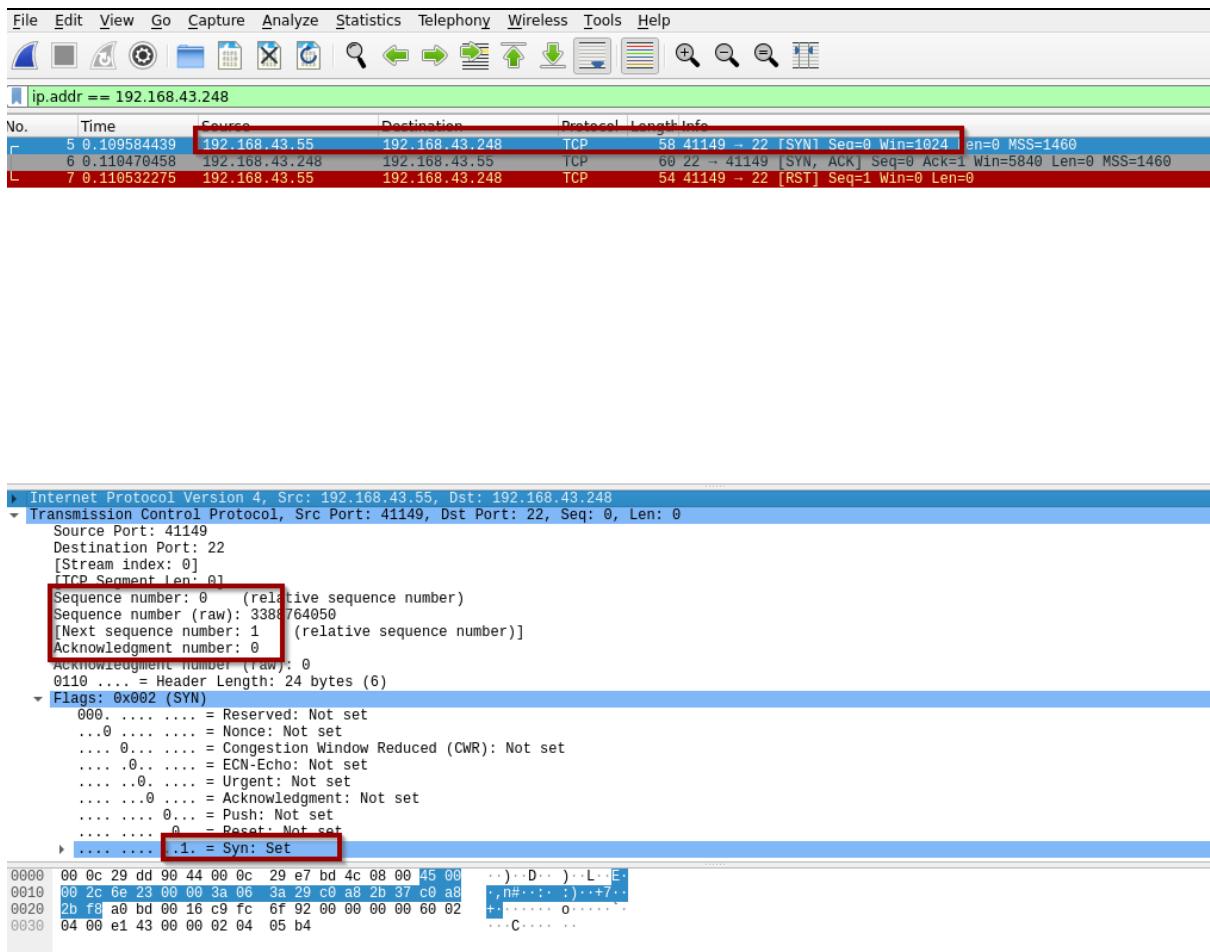
kali@kali:~$ sudo nmap -sS -p 22 192.168.43.248
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 07:51 IST
Nmap scan report for 192.168.43.248
Host is up (0.00091s latency).

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:~$ █

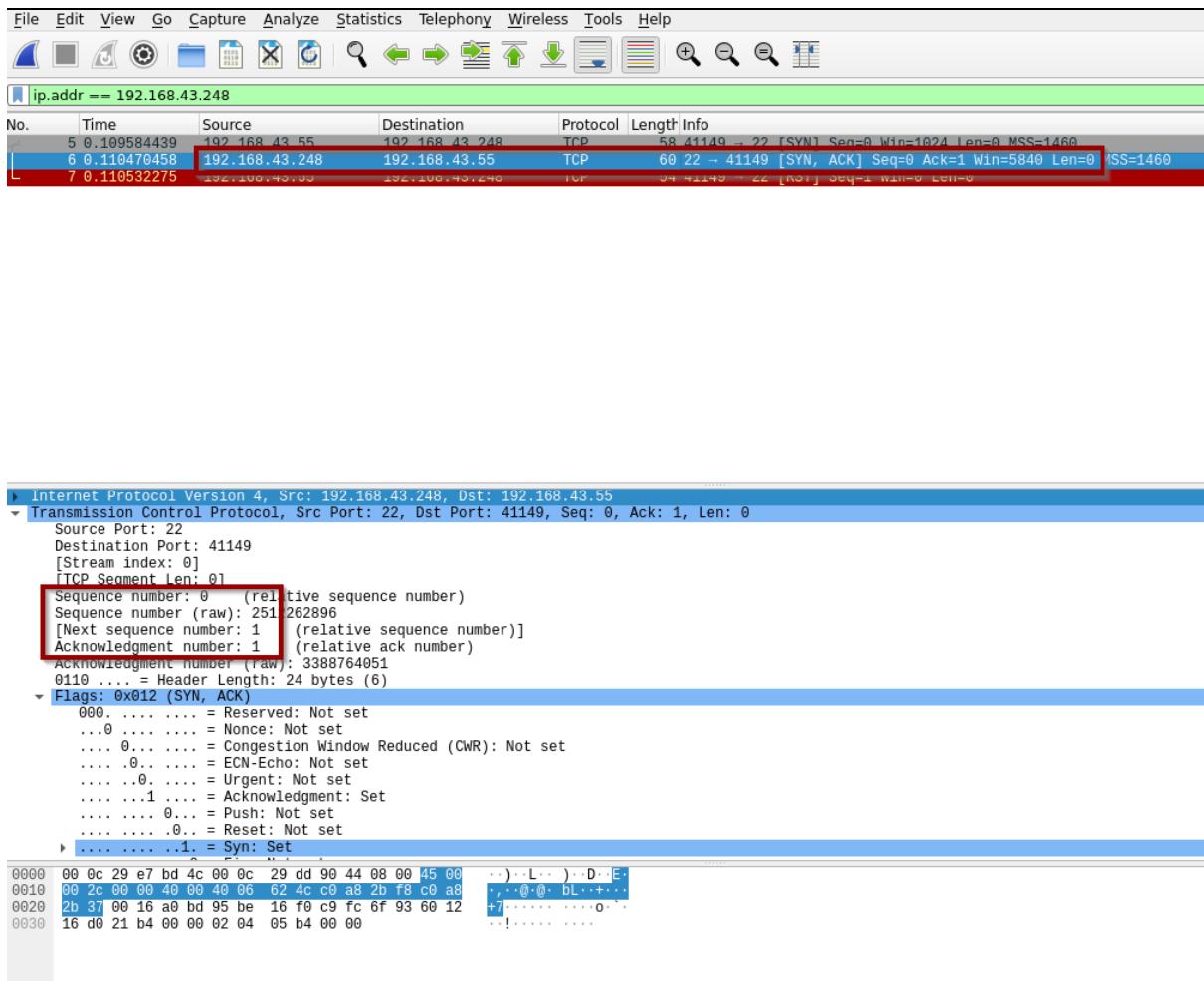
```

Port is open, lets analyze it in Wireshark.



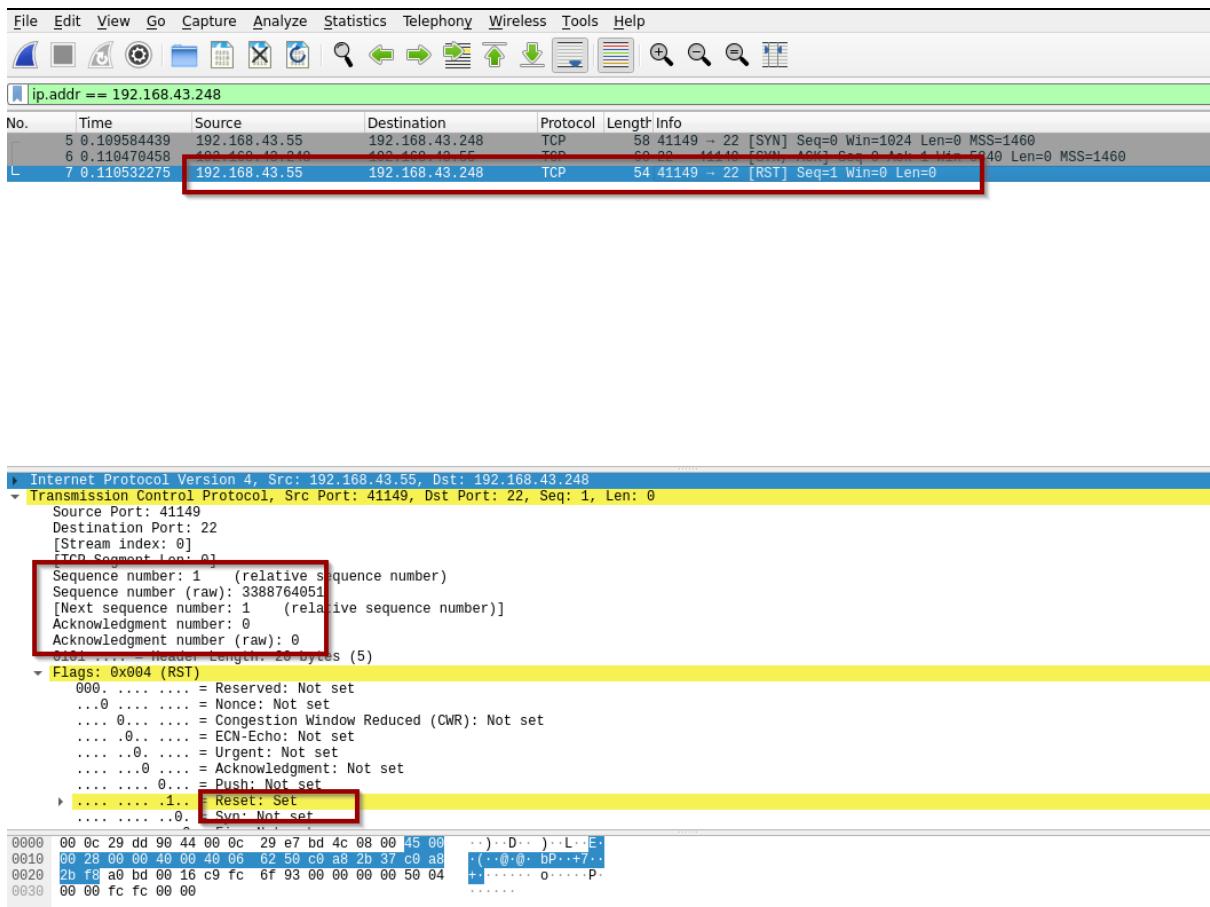
Let's analyze the first packet, we see that SYN packet is sent to the destination, in the flag section we see SYN is in Set mode.

Let's see if the server responds.



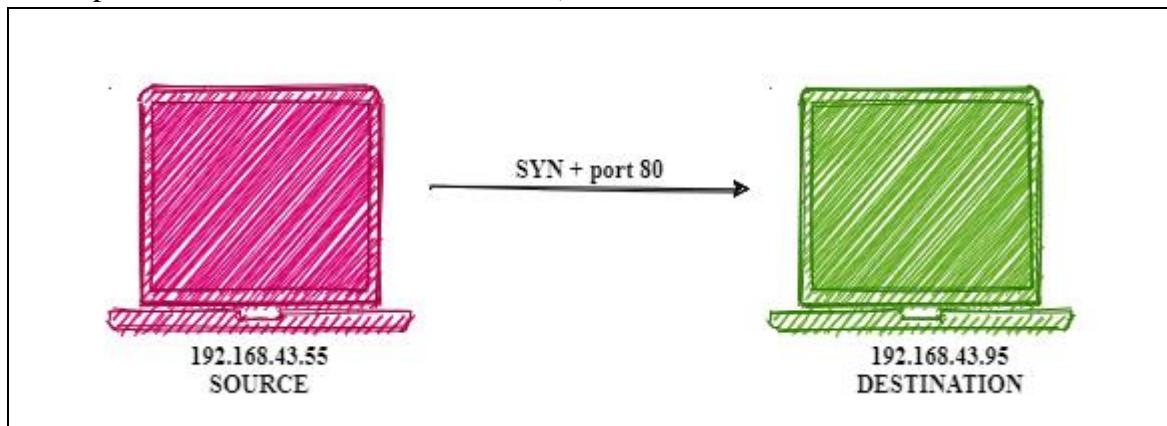
In the second packet the server sends an ACK (responds for SYN packet 1) and send SYN packet to the source for his validation. We can see the sequence number is 0 and Acknowledgement number is 1.

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.

ip.addr == 192.168.43.55

No.	Time	Source	Destination	Protocol	Length
10	2.102235058	192.168.43.55	192.168.43.95	TCP	58
11	2.204690629	192.168.43.55	192.168.43.95	TCP	58

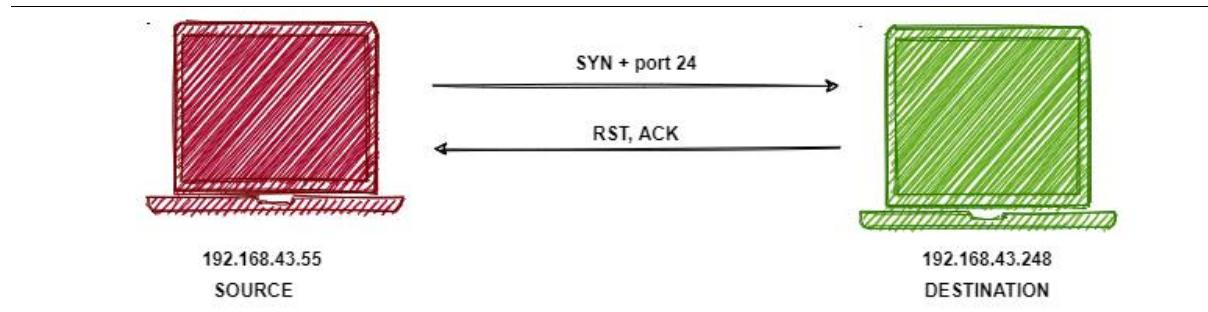
Transmission Control Protocol, Src Port: 48355, Dst Port: 80, Seq: 0, Len: 0

Source Port: 48355  
 Destination Port: 80  
 [Stream index: 0]  
 [TCP Segment Len: 0]  
 Sequence number: 0 (relative sequence number)  
 Sequence number (raw): 3474478607  
 [Next sequence number: 1 (relative sequence number)]  
 Acknowledgment number: 0  
 Acknowledgment number (raw): 0  
 0110 .... = Header Length: 24 bytes (6)  
 Flags: 0x002 (SYN)  
 000.... .... = Reserved: Not set  
 ...0.... .... = Nonce: Not set  
 ....0.... .... = Congestion Window Reduced (CWR): Not set  
 ....0.... .... = ECN-Echo: Not set  
 ....0.... .... = Urgent: Not set  
 ....0.... .... = Acknowledgment: Not set  
 ....0.... .... = Push: Not set  
 ....0.... .... = Reserve: Not set  
 ....0.... ....1= Syn: Set

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.

The screenshot shows a Wireshark capture window. A single TCP packet is selected, originating from 192.168.43.55 (port 43286) to 192.168.43.248 (port 24). The packet details pane displays the following information:

- Source: 192.168.43.55
- Destination: 192.168.43.248
- Protocol: TCP
- Length: 24
- Info: [SYN] Seq=0 Win=1024

The packet bytes pane shows the raw hex and ASCII data of the packet. In the packet details pane, the "Flags" section is expanded, showing the SYN flag is set (value 0x002). A red box highlights this entry.

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.

Screenshot of Wireshark showing a TCP RST/ACK packet. The packet details show the sequence number (1), acknowledgement number (1), and flags (RST, ACK). The hex dump shows the raw bytes of the packet.

```

Internet Protocol Version 4, Src: 192.168.43.248, Dst: 192.168.43.55
Transmission Control Protocol, Src Port: 24, Dst Port: 43286, Seq: 1, Ack: 1, Len: 0
Source Port: 24
Destination Port: 43286
[Stream index: 0]
[TCP Segment len: 0]
Sequence number: 1 (relative sequence number)
Sequence number (raw): 0
[Next sequence number: 1 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
Acknowledgment number (raw): 344244938
0101 .... = Header Length: 20 bytes (5)
Flags: 0x014 (RST, ACK)
    000.... = Reserved: Not set
    ...0.... = Nonce: Not set
    ....0.... = Congestion Window Reduced (CWR): Not set
    ....0.... = ECN-Echo: Not set
    ....0.... = Urgent: Not set
    ....1.... = Acknowledgment: Set
    ....0.... = Push: Not set
    ....1.. = Reset: Set
    ....0.. = Syn: Not set
0000 00 0c 29 e7 bd 4c 00 0c 29 dd 90 44 08 00 45 00  ..L..)D.E.
0010 00 28 00 00 40 00 40 06 62 50 c0 a8 b2 f8 c0 a8  .@.D.P.+
0020 2b 37 00 18 a9 16 00 00 00 00 14 84 c2 ca 50 14  +P....P.
0030 00 00 56 d3 00 00 00 00 00 00 00 00 00 00 00 00 00  ..V.....

```

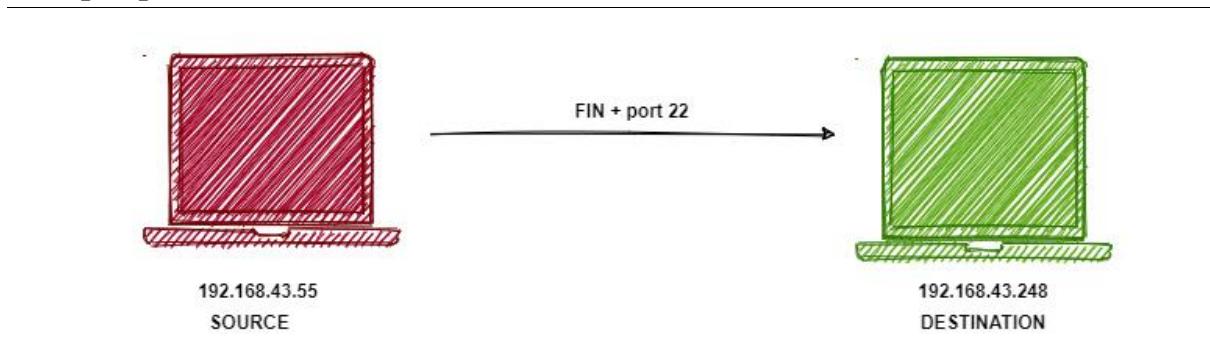
We get an RST and ACK packet, as the port is closed.

## 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.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 192.168.43.248

No.	Time	Source	Destination	Protocol	Length	Info
5	0.130019275	192.168.43.55	192.168.43.248	TCP	54	51004 → 22 [FIN] Seq=1 Win=1024 Len=0
6	0.234064789	192.168.43.55	192.168.43.248	TCP	54	51005 → 22 [FIN] Seq=1 Win=1024 Len=0

Internet Protocol Version 4, Src: 192.168.43.55, Dst: 192.168.43.248

Transmission Control Protocol, Src Port: 51004, Dst Port: 22, Seq: 1, Len: 0

Source Port: 51004  
Destination Port: 22  
[Stream index: 0]  
[TCP Segment Len: 0]

Sequence number: 1 (relative sequence number)  
Sequence number (raw): 4164097312  
[Next sequence number: 2 (relative sequence number)]  
Acknowledgment number: 0

Acknowledgment number (raw): 0  
0101 .... = Header Length: 20 bytes (5)

Flags: 0x001 (FIN)

000. .... = Reserved: Not set  
...0 .... =Nonce: Not set  
.... 0.... = Congestion Window Reduced (CWR): Not set  
.... .0.... = ECN-Echo: Not set  
.... ..0.... = Urgent: Not set  
.... ...0.... = Acknowledgment: Not set  
.... ....0.... = Push: Not set  
.... .....0.... = Reset: Not set  
.... .....0.... = Syn: Not set

0000 00 0c 29 dd 90 44 00 0c 29 e7 bd 4c 08 00 45 00 ..D..L..E..  
0010 00 28 96 bb 00 00 35 06 16 95 c0 a8 2b 37 c0 a8 ..(....5.....+7..  
0020 2b f8 c7 3c 00 16 f8 33 15 20 00 00 00 00 50 01 +.<...3.....P.  
0030 04 00 fe bc 00 00 .....

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.

The screenshot shows a Wireshark capture window. The top part displays a list of network packets. The fifth packet (Source: 192.168.43.55, Destination: 192.168.43.248, Protocol: TCP, Length: 54, Info: 54 51005 → 22 [FIN] Seq=1 Win=1024 Len=0) is highlighted with a red box. The bottom part shows the detailed analysis for this packet:

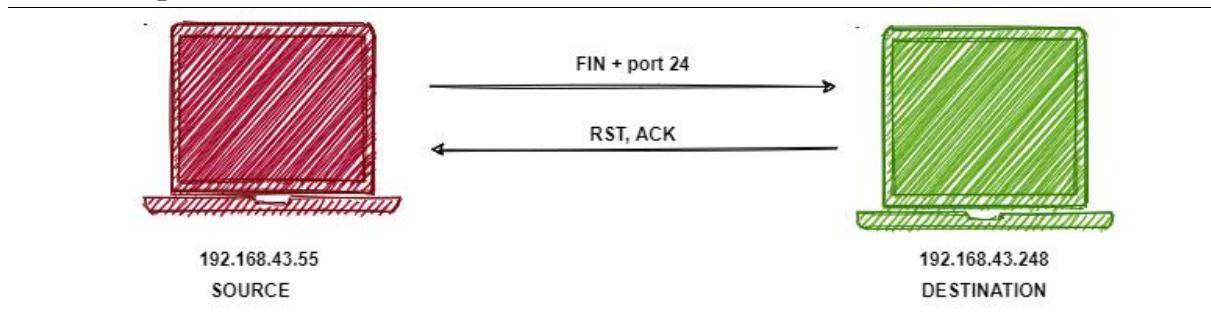
- Internet Protocol Version 4, Src: 192.168.43.55, Dst: 192.168.43.248**
- Transmission Control Protocol, Src Port: 51005, Dst Port: 22, Seq: 1, Len: 0**
- Source Port: 51005  
Destination Port: 22  
[Stream index: 1]  
[TCP Segment Len: 0]
- Sequence number: 1 (relative sequence number)  
Sequence number (raw): 4164031777  
[Next sequence number: 2 (relative sequence number)]  
Acknowledgment number: 0
- Acknowledgment number (raw): 0  
0101 .... = Header Length: 20 bytes (5)
- Flags: 0x001 (FIN)**
  - 000. .... = Reserved: Not set
  - ...0 .... = Nonce: Not set
  - .... 0... = Congestion Window Reduced (CWR): Not set
  - .... .0.. = ECN-Echo: Not set
  - .... ..0. = Urgent: Not set
  - .... .0.... = Acknowledgment: Not set
  - .... ....0... = Push: Not set
  - .... ....0.. = Reset: Not set
  - .... ....0. = Syn: Not set
- Hex dump:

0000	00	0c	29	dd	90	44	00	0c	29	e7	bd	4c	08	00	45	00	..)	..D	..)	..L	..E
0010	00	28	c5	62	00	00	35	06	e7	ed	c0	a8	2b	37	c0	a8	.(.	b..5	..	+7..	
0020	2b	f8	c7	3d	00	16	f8	32	15	21	00	00	00	00	50	01	+	=	..2	..!	p..
0030	04	00	fe	bb	00	00											.....				
- ASCII dump:

0000	00	0c	29	dd	90	44	00	0c	29	e7	bd	4c	08	00	45	00	..)	..D	..)	..L	..E
0010	00	28	c5	62	00	00	35	06	e7	ed	c0	a8	2b	37	c0	a8	.(.	b..5	..	+7..	
0020	2b	f8	c7	3d	00	16	f8	32	15	21	00	00	00	00	50	01	+	=	..2	..!	p..
0030	04	00	fe	bb	00	00											.....				

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

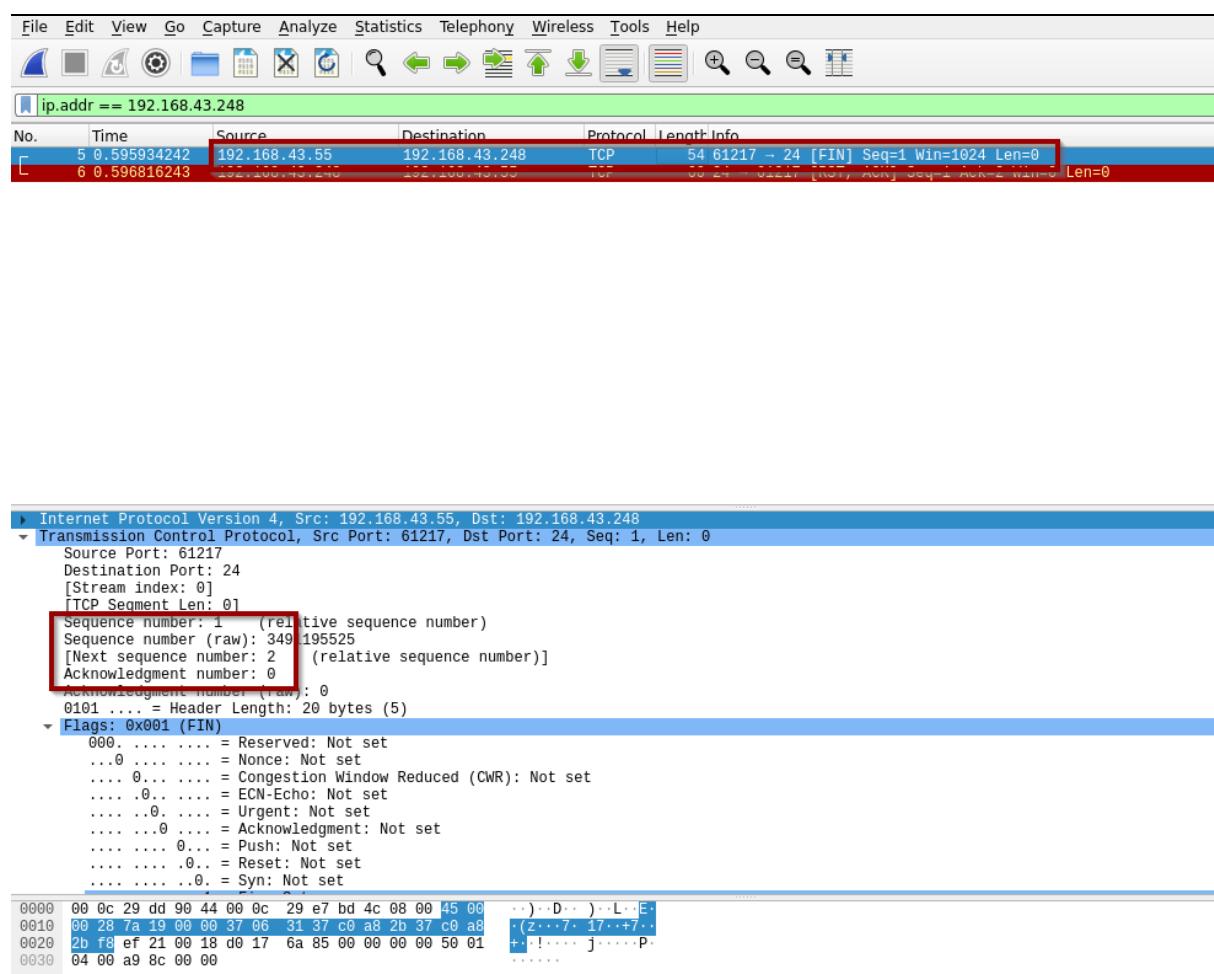
```
nmap -sF -p 24 192.168.43.248
```

```
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).

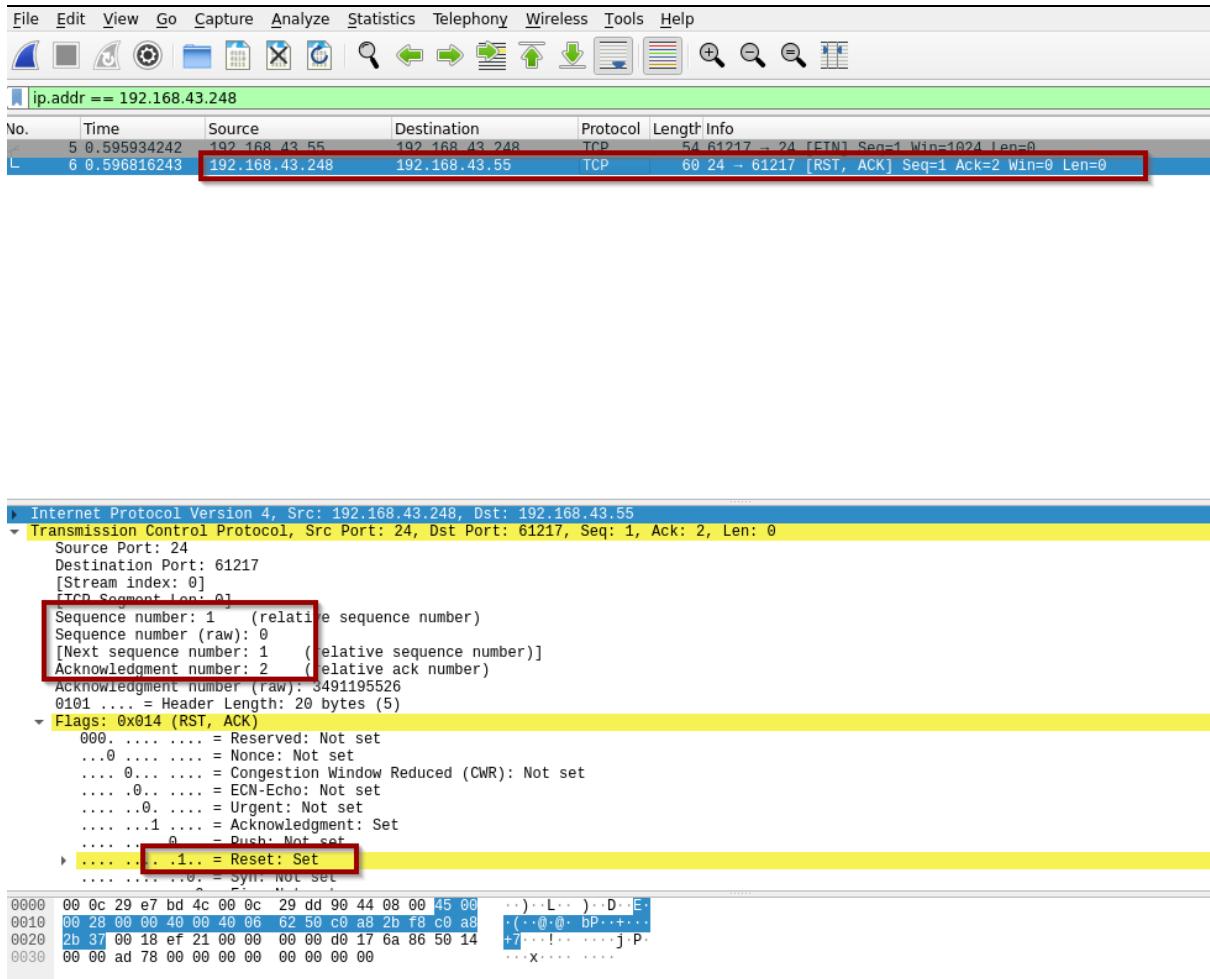
PORT      STATE SERVICE
24/tcp    closed  priv-mail
MAC Address: 00:0C:29:DD:90:44 (VMware)

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.



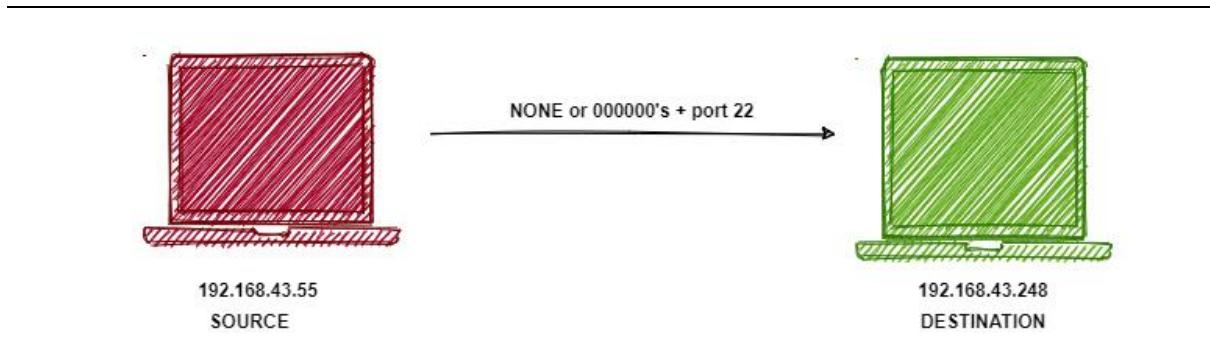
Since the port is closed the server sends the responds as RST (reset), ACK packet.

## 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.

Screenshot of Wireshark showing network traffic capture and packet details.

**Network Statistics:**

- No. 5 0.098424706 192.168.43.55 → 192.168.43.248 [TCP] Seq=1 Win=1024 Len=0
- No. 6 0.200903113 192.168.43.55 → 192.168.43.248 [TCP] Seq=1 Win=1024 Len=0

**Selected Packet Details:**

Internet Protocol Version 4, Src: 192.168.43.55, Dst: 192.168.43.248

Transmission Control Protocol, Src Port: 62728, Dst Port: 22, Seq: 1, Len: 0

Source Port: 62728  
 Destination Port: 22  
 [Stream index: 0]  
 [TCP Segment len: 0]  
 Sequence number: 1 (relative sequence number)  
 Sequence number (raw): 4265613432  
 [Next sequence number: 1 (relative sequence number)]  
 Acknowledgment number: 0  
 Acknowledgment number (raw): 0  
 0101 .... = Header Length: 20 bytes (5)  
 Flags: 0x000 (<None>)  
 000. .... = Reserved: Not set  
 ...0 .... = Nonce: Not set  
 .... 0... = Congestion Window Reduced (CWR): Not set  
 .... .0... = ECN-Echo: Not set  
 .... .0... = Urgent: Not set  
 .... ..0... = Acknowledgment: Not set  
 .... ....0... = Push: Not set  
 .... ....0.. = Reset: Not set  
 .... ....0. = Syn: Not set

**Hex and ASCII Data:**

0000	00 0c 29 dd 90 44 00 0c 29 e7 bd 4c 08 00 45 00	..)·D.. )..L..E..
0010	00 28 ae f1 00 00 35 06 fe 5e c0 a8 2b 37 c0 a8	.(....5. .^...+7..
0020	2b f8 f5 08 00 16 fe 40 18 78 00 00 00 00 50 00	+.....@ x...P.
0030	04 00 c7 8c 00 00	.....

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.

Wireshark screenshot showing a packet capture for IP address 192.168.43.248. The selected packet (No. 6) is highlighted in red.

No.	Time	Source	Destination	Protocol	Length	Info
5	0.098424706	192.168.43.55	192.168.43.248	TCP	54	62728 → 22 [ <None> ] Seq=1 Win=1024 Len=0
6	0.200903113	192.168.43.55	192.168.43.248	TCP	54	62729 → 22 [ <None> ] Seq=1 Win=1024 Len=0

Packet details for No. 6:

```

> Internet Protocol Version 4, Src: 192.168.43.55, Dst: 192.168.43.248
  Transmission Control Protocol, Src Port: 62729, Dst Port: 22, Seq: 1, Len: 0
    Source Port: 62729
    Destination Port: 22
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence number: 1 (relative sequence number)
    Sequence number (raw): 4265678969
    [Next sequence number: 1 (relative sequence number)]
    Acknowledgment number: 0
    Acknowledgment number (raw): 0
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x000 (<None>)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0.... .... = Congestion Window Reduced (CWR): Not set
      .... .0.... .... = ECN-Echo: Not set
      .... ..0.... .... = Urgent: Not set
      .... ...0.... .... = Acknowledgment: Not set
      .... ...0... .... = Push: Not set
      .... ....0.... .... = Reset: Not set
      .... ....0.... .... = Syn: Not set

```

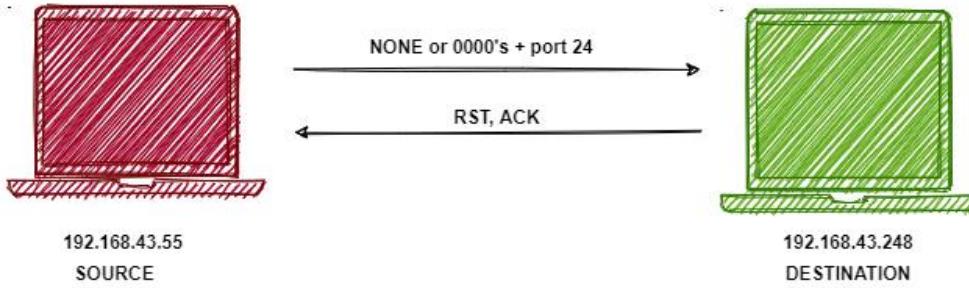
Hex dump for No. 6:

```

0000  00 0c 29 dd 90 44 00 0c 29 e7 bd 4c 08 00 45 00  ..)..D.. )..L..E..
0010  00 28 f7 1c 00 00 30 06 bb 33 c0 a8 2b 37 c0 a8  ((...0, .3..+7..
0020  2b f8 f5 09 00 16 fe 41 18 79 00 00 00 00 50 00  +.....A.y....P.
0030  04 00 c7 89 00 00

```

## 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**

```

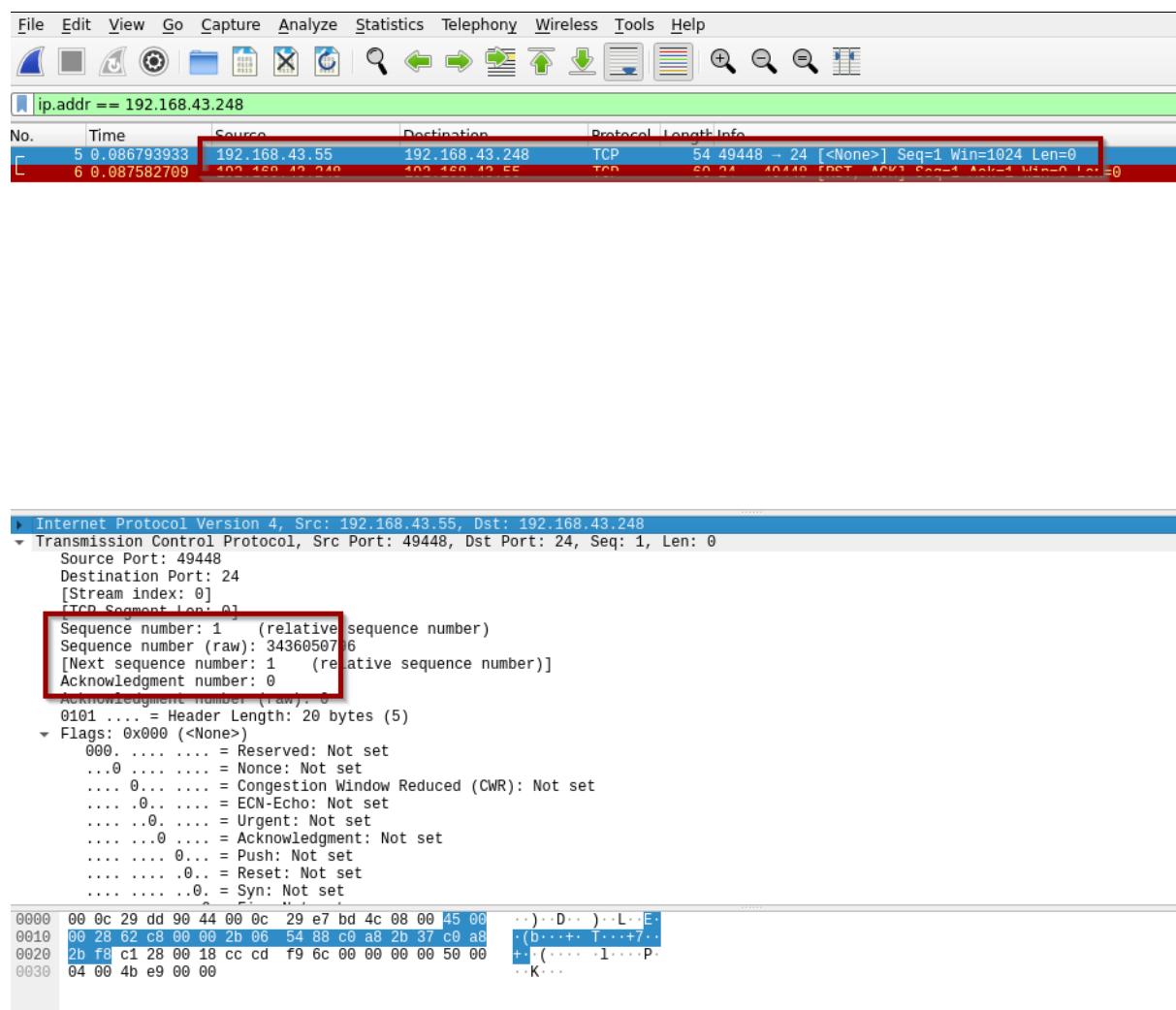
kali@kali:~$ sudo nmap -sN -p 24 192.168.43.248
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 09:13 IST
Nmap scan report for 192.168.43.248
Host is up (0.00085s latency).

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:~$ █

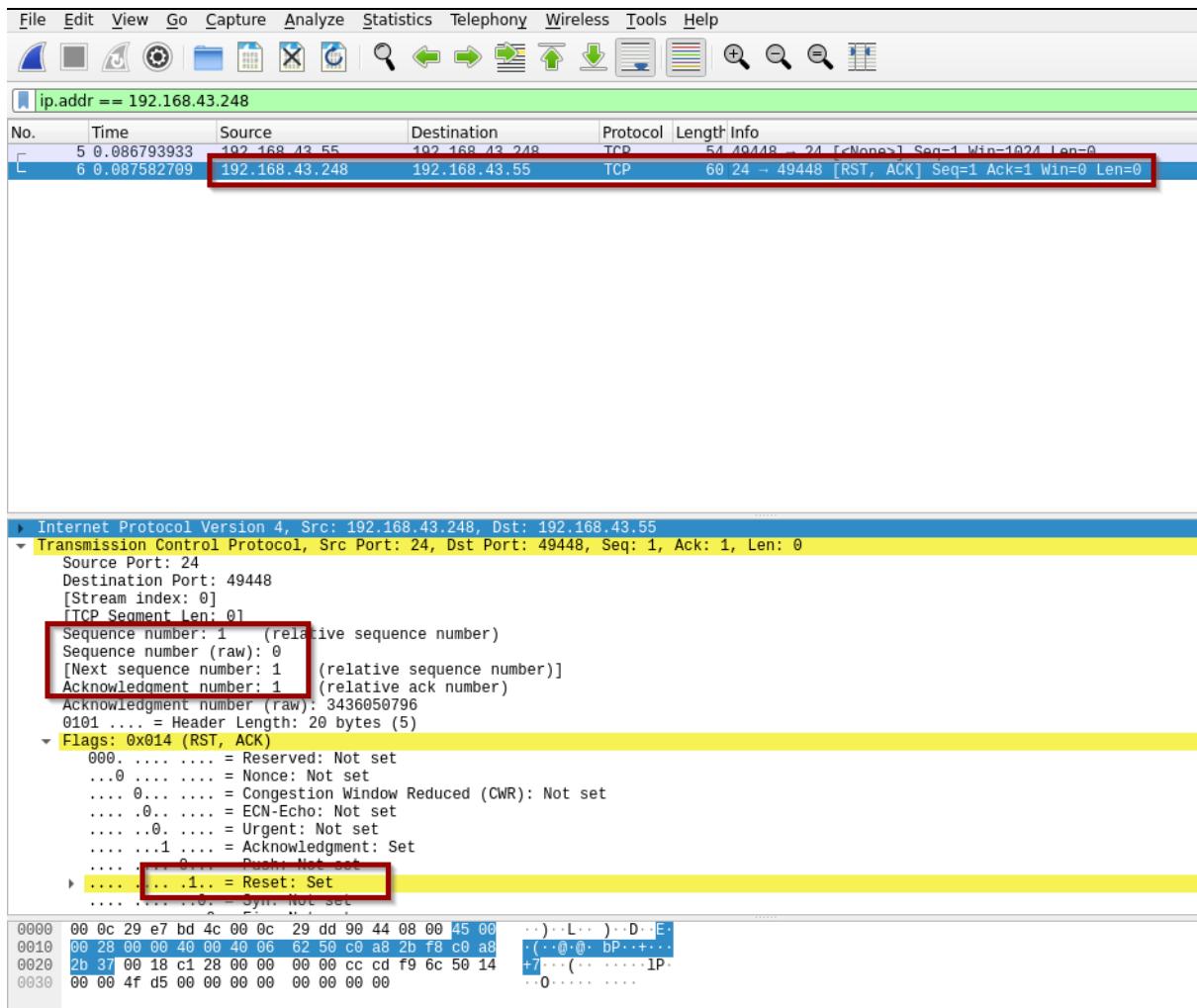
```

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.



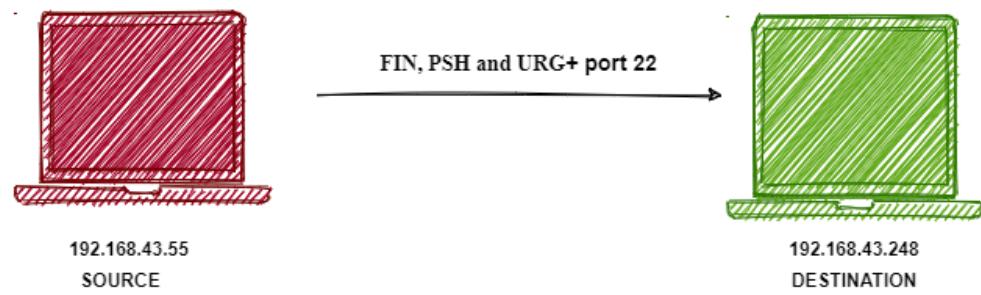
We got a reset flag from the destination which means port is closed.

## 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

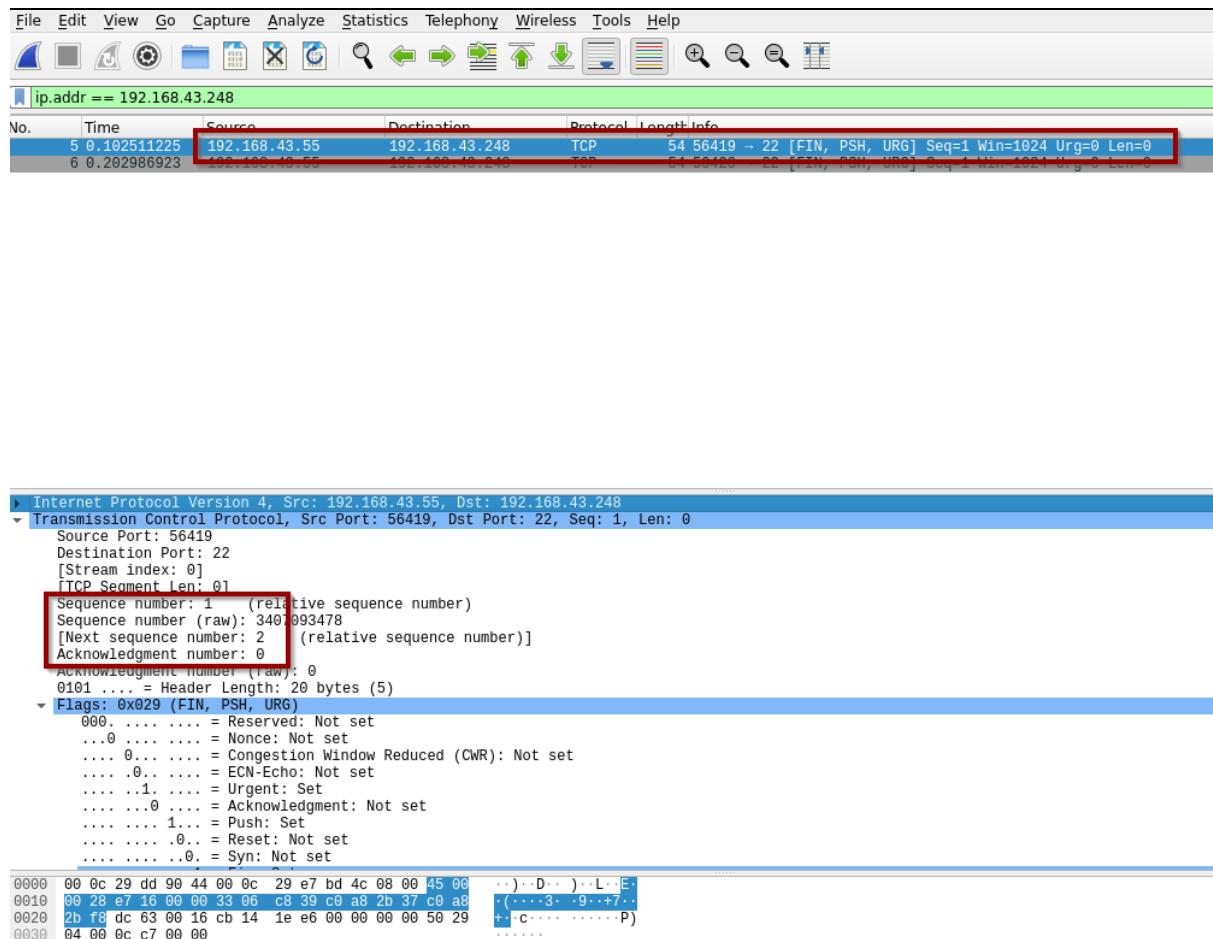
```
nmap -sX -p 22 192.168.43.248
```

```
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).

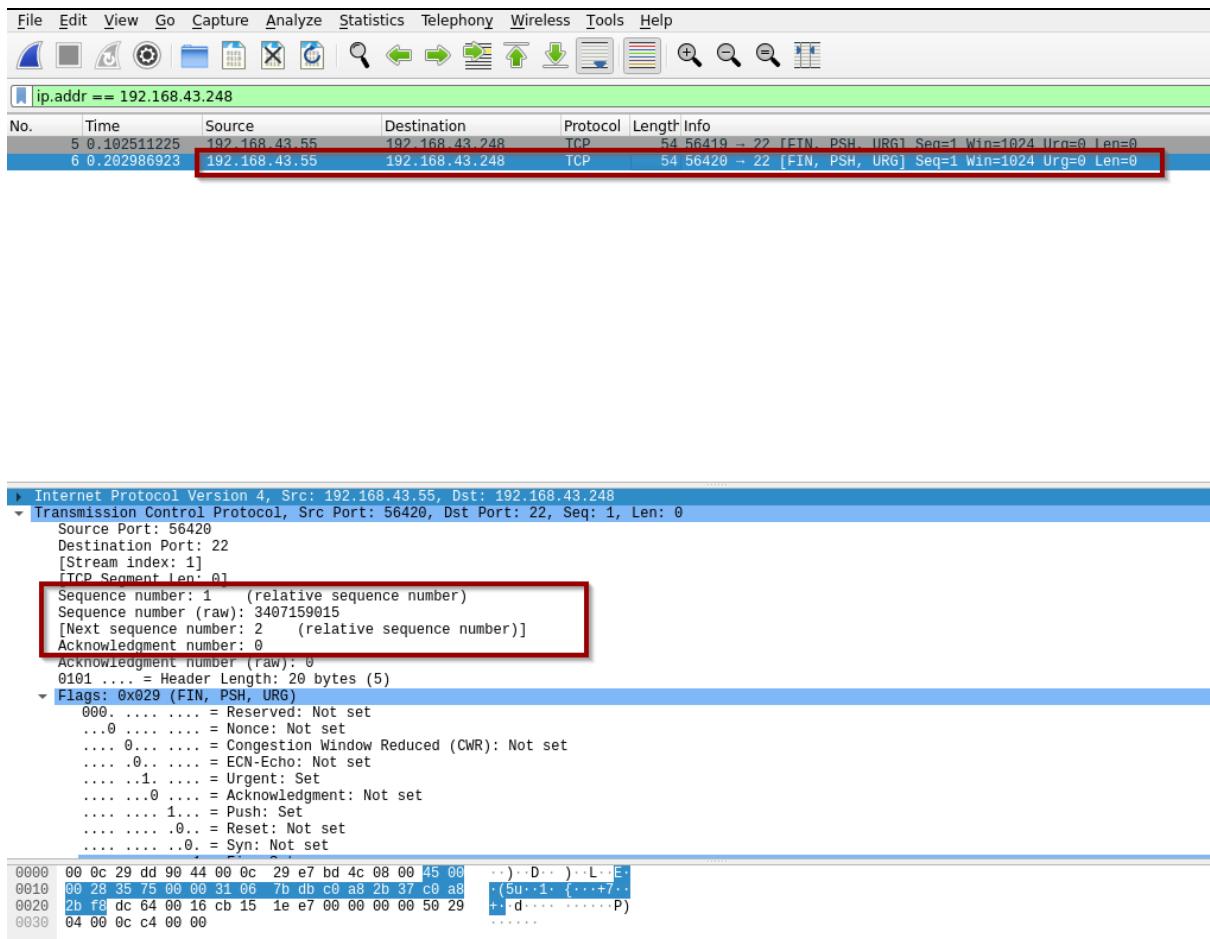
PORT      STATE            SERVICE
22/tcp    open|filtered  ssh
MAC Address: 00:0C:29:DD:90:44 (VMware)

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.

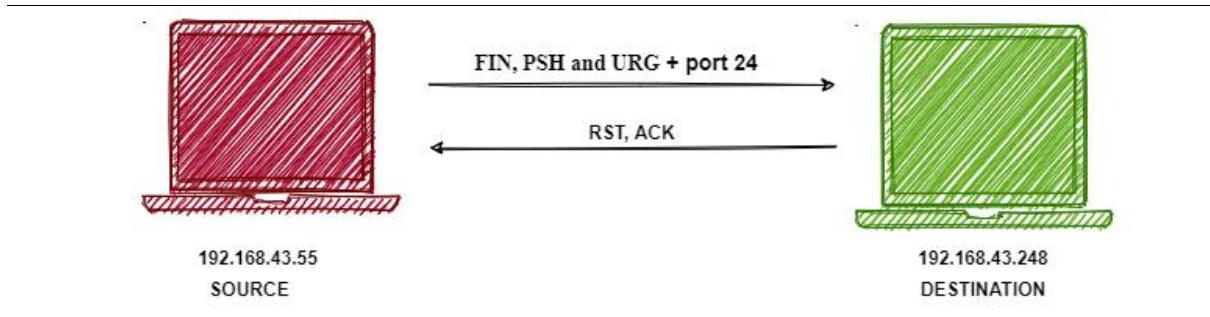


Source also send a second packet (as part of filtered port)



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`

```

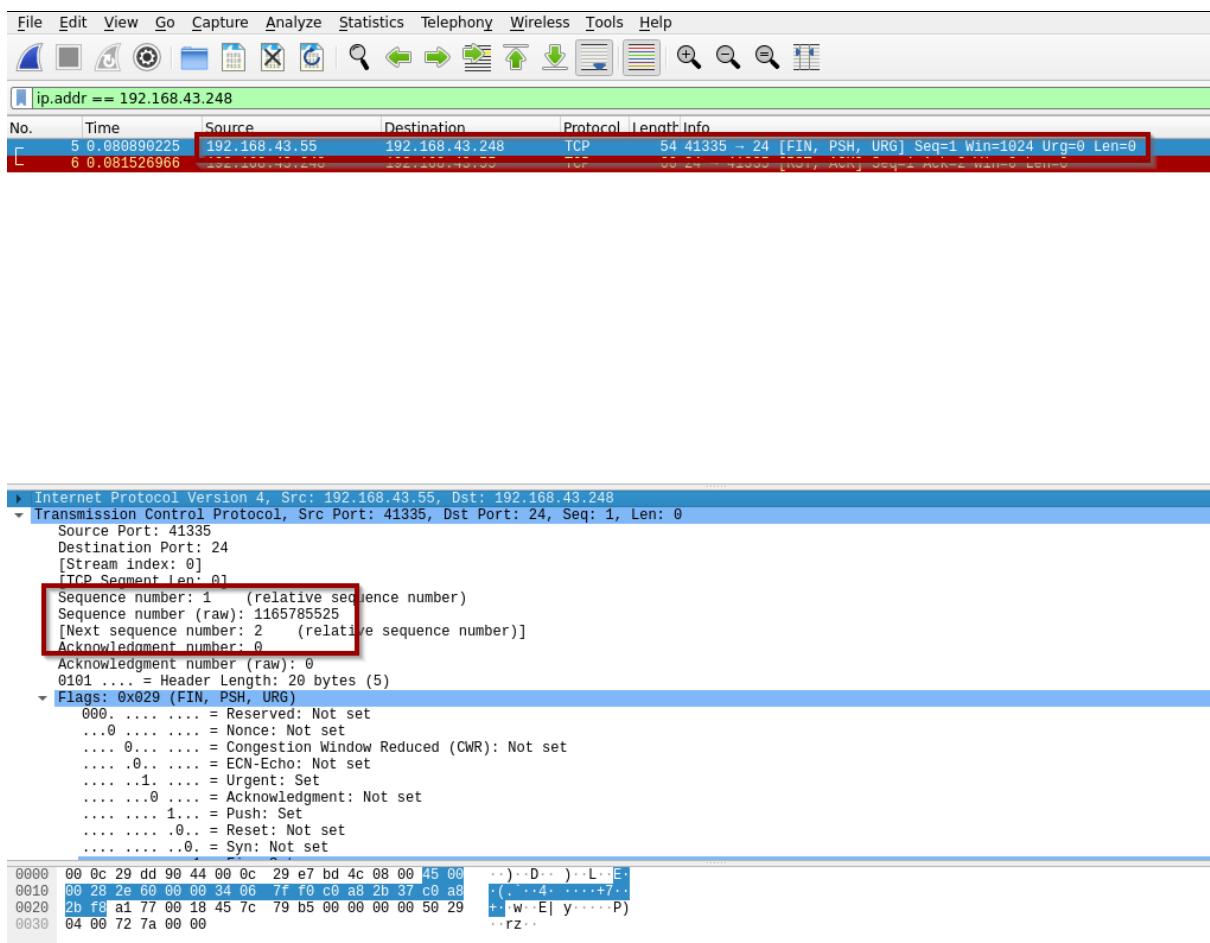
kali@kali:~$ sudo nmap -sX -p 24 192.168.43.248
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-20 09:02 IST
Nmap scan report for 192.168.43.248
Host is up (0.00100s latency).

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.

Wireshark screenshot showing a TCP connection. A red box highlights the second packet (TCP ACK) from the source.

Internet Protocol Version 4, Src: 192.168.43.248, Dst: 192.168.43.55

Transmission Control Protocol, Src Port: 24, Dst Port: 41335, Seq: 1, Ack: 2, Len: 0

Source Port: 24  
 Destination Port: 41335  
 [Stream index: 0]  
 [TCP Segment len: 0]

Sequence number: 1 (relative sequence number)  
 Sequence number (raw): 0  
 [Next sequence number: 1 (relative sequence number)]  
 Acknowledgment number: 2 (relative ack number)  
 Acknowledgment number (raw): 1165785526  
 0101 .... = Header Length: 20 bytes (5)

Flags: 0x014 (RST, ACK)

- 000 .... .... = Reserved: Not set
- ....0 .... .... = Nonce: Not set
- ....0.... .... = Congestion Window Reduced (CWR): Not set
- ....0.... .... = ECN-Echo: Not set
- ....0.... .... = Urgent: Not set
- ....1.... .... = Acknowledgment: Set
- ....0.... .... = Push: Not set
- ....0.... .... = Reset: Set

0000 00 0c 29 e7 bd 4c 00 0c 29 dd 90 44 08 00 45 00 ..L.. )-D-E.  
 0010 00 28 00 00 40 00 40 06 62 50 c0 a8 2b f8 c0 a8 .(0:0:0:0:P+...  
 0020 2b 37 00 18 a1 77 00 00 00 45 7c 79 b6 50 14 +7:...w...E/y/P.  
 0030 00 00 76 8e 00 00 00 00 00 00 00 00 00 00 00 00 ..V.....

## UDP SCAN (-sU)

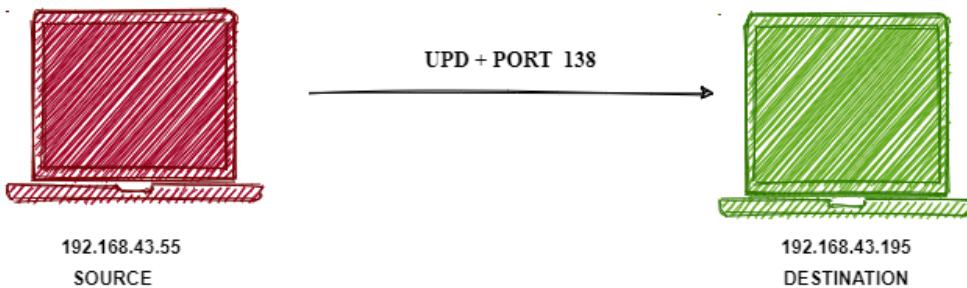
This scan works by sending UDP packets to the destination port, if the target port is open then there is no response, if port is closed then destination sends an ICMP packet saying unreachable.

This scan is also called connection less protocol.

### Advantage:

- 1) More control over data sent out.
- 2) UDP 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.

The Wireshark interface shows a list of network captures. A red box highlights the first two entries:

No.	Time	Source	Destination	Protocol	Length	Info
5	0.086820542	192.168.43.55	192.168.43.195	UDP	42	35096 → 138 Len=0
6	0.188698020	192.168.43.55	192.168.43.195	UDP	42	35097 → 138 Len=0

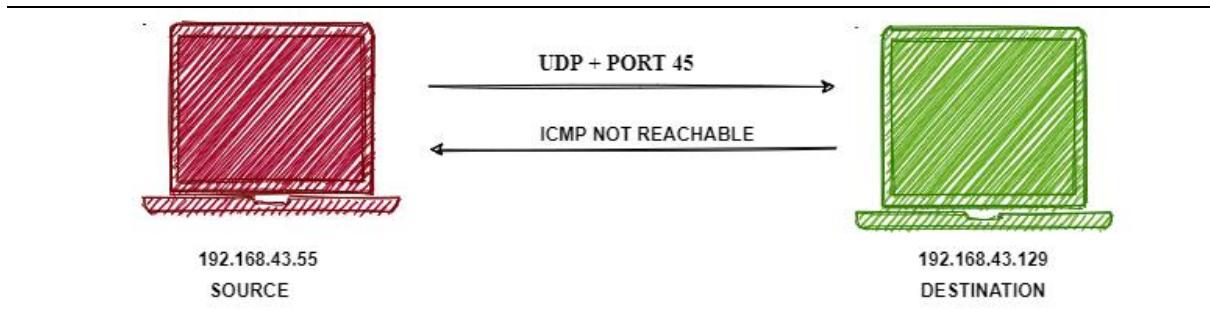
The details pane for the first packet (No. 5) displays the following information:

- Ethernet II, Src: VMware\_e7:bd:4c (00:0c:29:e7:bd:4c), Dst: VMware\_e9:aa:f0 (00:0c:29:e9:aa:f0)
- Internet Protocol Version 4, Src: 192.168.43.55, Dst: 192.168.43.195
- Version: 4
- Header Length: 20 bytes (5)
- Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
- Total Length: 28
- Identification: 0xf798 (63384)
- Flags: 0x0000
  - 0... .... .... = Reserved bit: Not set
  - .0.. .... .... = Don't fragment: Not set
  - ..0. .... .... = More fragments: Not set
- Fragment offset: 0
- Time to live: 49
- Protocol: UDP (17)
- Header checksum: 0xb9ed [validation disabled]

The bytes pane shows the raw hex and ASCII data for the selected packet.

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.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 192.168.43.129

No.	Time	Source	Destination	Protocol	Length	Info
7	0.001637615	192.168.43.95	192.168.43.129	UDP	70	51318 → 2054 Len=28
8	0.001639722	192.168.43.129	192.168.43.95	ICMP	98	Destination unreachable (Port unreachable)
13	0.840191258	192.168.43.55	192.168.43.129	UDP	42	56825 → 45 Len=0
14	0.841275405	192.168.43.129	192.168.43.55	ICMP	70	Destination unreachable (Port unreachable)

```

▶ Frame 7: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface eth0, id 0
▶ Ethernet II, Src: IntelCor_63:61:9f (1c:1b:b5:63:61:9f), Dst: VMWare_de:ad:cc (00:0c:29:de:ad:cc)
└ Internet Protocol Version 4, Src: 192.168.43.95, Dst: 192.168.43.129
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 56
    Identification: 0x4219 (16921)
    ▶ Flags: 0x0000
      0... .... .... = Reserved bit: Not set
      .0.. .... .... = Don't fragment: Not set
      ... . .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: UDP (17)
  0000  00 0c 29 de ad cc 1c 1b b5 63 61 9f 08 00 45 00  ..).... ca .. E.
  0010  00 38 42 19 00 00 80 11 20 6b c0 a8 2b 5f c0 a8  .8B..... k..+...
  0020  2b 81 c8 76 08 06 00 24 3d a2 00 01 08 00 06 04  +..v...$ =.....
  0030  00 01 1c 1b b5 63 61 9f c0 a8 2b 5f ff ff ff ff  .....ca. ..+....
  0040  ff ff c0 a8 2b 81  .....+.

```

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

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 192.168.43.129

No.	Time	Source	Destination	Protocol	Length	Info
7	0.001637615	192.168.43.95	192.168.43.129	UDP	70	51318 → 2054 Len=28
8	0.001639722	192.168.43.129	192.168.43.95	ICMP	98	Destination unreachable (Port unreachable)
13	0.840191258	192.168.43.55	192.168.43.129	UDP	42	56825 → 45 Len=0
14	0.841275405	192.168.43.129	192.168.43.55	ICMP	70	Destination unreachable (Port unreachable)

```

▶ Frame 8: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth0, id 0
▶ Ethernet II, Src: VMWare_de:ad:cc (00:0c:29:de:ad:cc), Dst: IntelCor_63:61:9f (1c:1b:b5:63:61:9f)
└ Internet Protocol Version 4, Src: 192.168.43.129, Dst: 192.168.43.95
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x86b1 (34481)
    ▶ Flags: 0x0000
      0... .... .... = Reserved bit: Not set
      .0.. .... .... = Don't fragment: Not set
      ... . .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)

```

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 respond with RST packet.
- 3) Each IP packet has a fragment Identity Number (IP ID), each packet sent results in increment of 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 powerful tool. It becomes even more powerful 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**

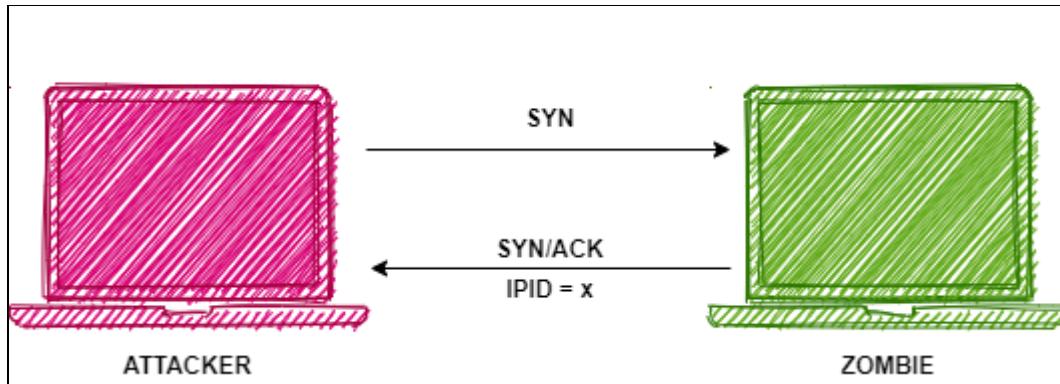
Attacker IP 192.168.43.55

Zombie IP 192.168.43.195

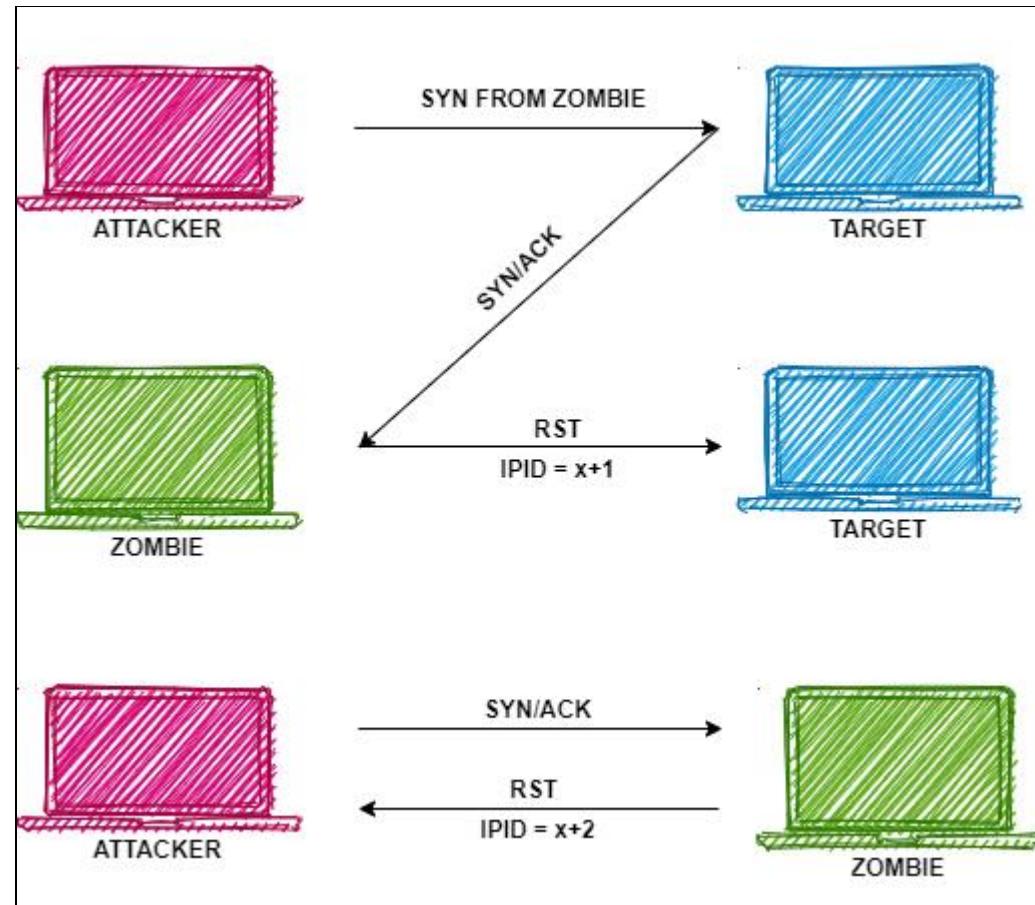
Target IP 192.168.43.248

**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 allow 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.

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	3.556300971	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	3.551105359	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
130	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	4.041912899	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	4.251867883	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.

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	3.78389945	192.168.43.195	192.168.43.55	TCP	60	443 - 42984 [RST] Seq=1 Win=0 Len=0
107	3.902677511	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	3.965342430	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	3.997686641	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	4.031680822	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=1460
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=1460
122	4.135381599	192.168.43.195	192.168.43.248	TCP	60	443 - 42983 [RST] Seq=1 Win=0 Len=0
123	4.185159691	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=1460
124	4.185159877	192.168.43.195	192.168.43.248	TCP	60	443 - 42983 [RST] Seq=1 Win=0 Len=0
125	4.485665294	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	4.486829591	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=1 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	4.538607891	192.168.43.55	192.168.43.195	TCP	58	[TCP Retransmission] 443 - 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
131	4.539231033	192.168.43.195	192.168.43.55	TOP	60	80 - 443 [SYN, ACK] Seq=1 Ack=1 Win=1024 Len=0 MSS=1460
132	4.599046937	192.168.43.55	192.168.43.195	TOP	58	43232 - 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
133	4.599675394	192.168.43.195	192.168.43.55	TOP	60	443 - 43232 [RST] Seq=1 Win=0 Len=0
134	4.5990776563	192.168.43.195	192.168.43.248	TOP	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	TOP	60	[TOP Previous segment not captured] [TCP Port numbers reused] 80 - 443 [SYN, ACK]
136	4.591502995	192.168.43.195	192.168.43.248	TOP	60	443 - 80 [RST] Seq=1 Win=0 Len=0
137	4.643513377	192.168.43.55	192.168.43.195	TOP	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	TOP	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.

No.	Time	Source	Destination	Protocol	Length	Info	
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	4.486506194	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	4.538607884	192.168.43.55	192.168.43.195	TCP	58	43206 - 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	4.539946033	192.168.43.195	192.168.43.55	TCP	58	43232 - 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460	
L	133	4.590675394	192.168.43.195	192.168.43.55	TCP	60	443 - 43232 [RST] Seq=1 Win=0 Len=0
134	4.591502995	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] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460	
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	
142	4.696588075	192.168.43.55	192.168.43.195	TCP	58	43177 - 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460	
						Frame 133: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0	
						Ethernet II, Src: VMware_e9:aa:f0 (00:0c:29:e9:aa:f0), Dst: VMware_e7:bd:4c (00:0c:29:e7:bd:4c)	
						Internet Protocol Version 4, Src: 192.168.43.195, Dst: 192.168.43.55	
						0100 ... = Version: 4	
						.... 0101 = Header Length: 20 bytes (5)	
						Differential Services Field: 0x00 (DSRP: CS0, ECN: Not-ECT)	
						Total Length: 40	
						Identification: 0x000b (186)	
						Flags: 0x4000, Don't Fragment	
						Fragment offset: 0	
						Time to live: 128	
						Protocol: TCP (6)	
						Header checksum: 0x21c9 [validation disabled]	
						[Header checksum status: Unverified]	
						Source: 192.168.43.195	
						Destination: 192.168.43.55	
						Transmission Control Protocol, Src Port: 443, Dst Port: 43232, Seq: 1, Len: 0	
						Source Port: 443	
						Destination Port: 43232	
						[Stream index: 12]	
						[TCP Segment Len: 0]	
						Sequence number: 1 (relative sequence number)	

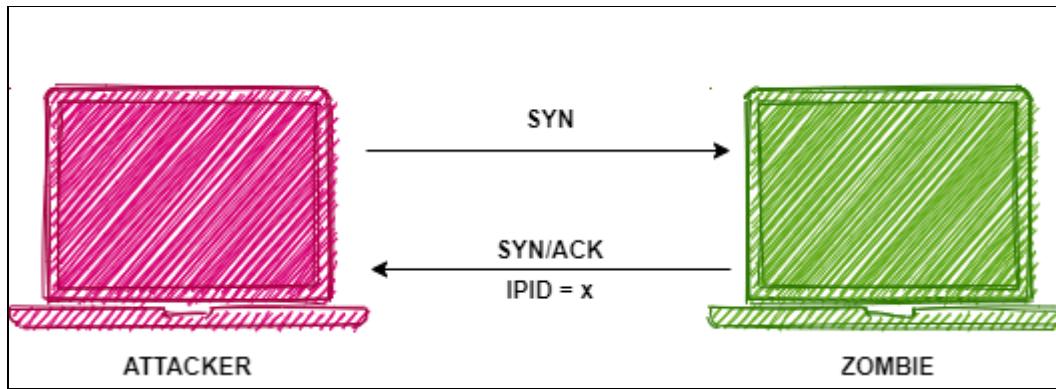
**Step 2 and 3:** After the RST packet to attacker the zombie scans the target (Retransmission step). After successful scan the zombie returns to communicate with attacker. Note the Identification number again. See the increase from previous number, its 188-186=2

No.	Time	Source	Destination	Protocol	Length	Info	
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	4.486506194	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	4.538607884	192.168.43.55	192.168.43.195	TCP	58	43206 - 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	4.539946033	192.168.43.55	192.168.43.195	TCP	58	43232 - 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460	
L	133	4.590675394	192.168.43.195	192.168.43.55	TCP	60	443 - 43232 [RST] Seq=1 Win=0 Len=0
134	4.591502995	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] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460	
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	
142	4.696588075	192.168.43.55	192.168.43.195	TCP	58	43177 - 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460	
						Frame 138: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0	
						Ethernet II, Src: VMware_e9:aa:f0 (00:0c:29:e9:aa:f0), Dst: VMware_e7:bd:4c (00:0c:29:e7:bd:4c)	
						Internet Protocol Version 4, Src: 192.168.43.195, Dst: 192.168.43.55	
						0100 ... = Version: 4	
						.... 0101 = Header Length: 20 bytes (5)	
						Differential Services Field: 0x00 (DSRP: CS0, ECN: Not-ECT)	
						Total Length: 40	
						Identification: 0x000b (188)	
						Flags: 0x4000, Don't Fragment	
						Fragment offset: 0	
						Time to live: 128	
						Protocol: TCP (6)	
						Header checksum: 0x21c9 [validation disabled]	
						[Header checksum status: Unverified]	
						Source: 192.168.43.195	
						Destination: 192.168.43.55	
						Transmission Control Protocol, Src Port: 443, Dst Port: 43216, Seq: 1, Len: 0	
						Source Port: 443	
						Destination Port: 43216	
						[Stream index: 13]	
						[TCP Segment Len: 0]	
						Sequence number: 1 (relative sequence number)	
						.... 0100 = Header Length: 20 bytes (5)	
						Differential Services Field: 0x00 (DSRP: CS0, ECN: Not-ECT)	
						Total Length: 40	
						Identification: 0x000b (188)	
						Flags: 0x4000, Don't Fragment	
						Fragment offset: 0	
						Time to live: 128	
						Protocol: TCP (6)	
						Header checksum: 0x21c9 [validation disabled]	
						[Header checksum status: Unverified]	
						Source: 192.168.43.195	
						Destination: 192.168.43.55	

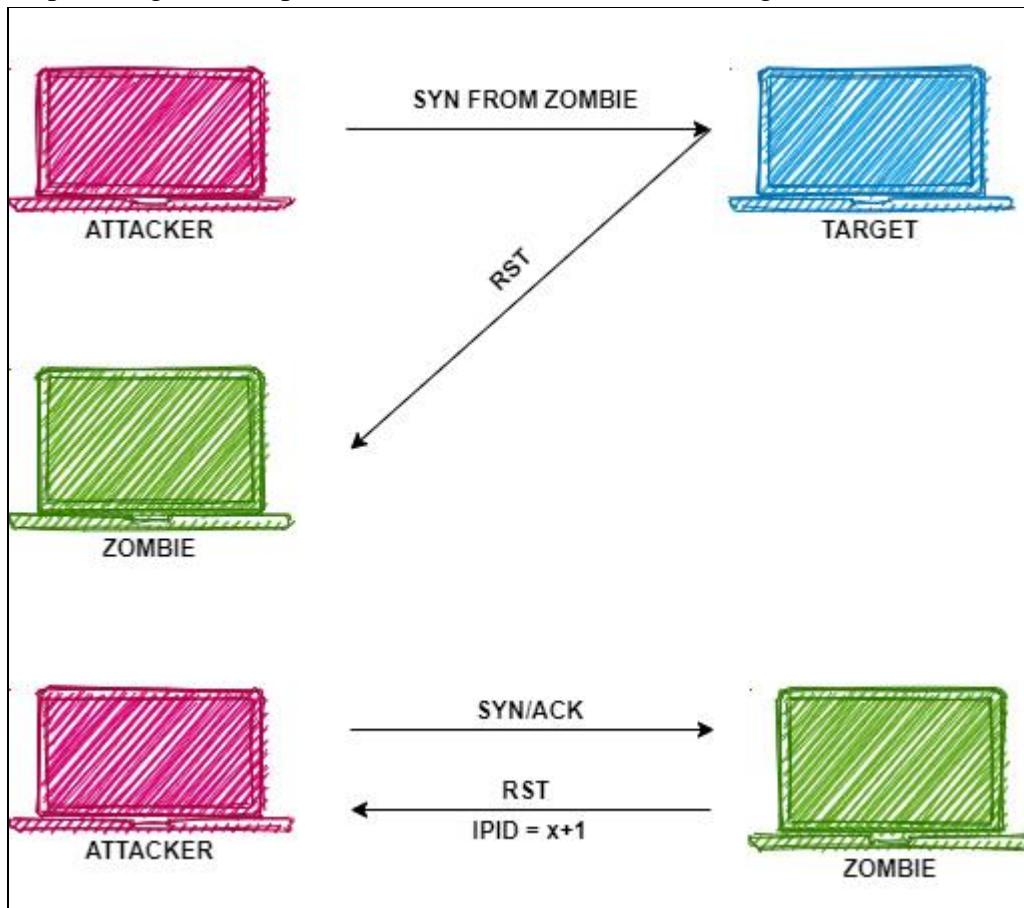
This difference signifies that the port is open for connection.

### 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 allow 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  sysstat
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.

No.	Time	Source	Destination	Protocol	Length	Info
88	5.687442855	192.168.43.55	192.168.43.195	TCP	58	42149 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
89	5.710505694	192.168.43.195	192.168.43.55	TCP	60	443 → 42149 [RST] Seq=1 Win=0 Len=0
90	5.714890952	192.168.43.55	192.168.43.195	TCP	58	42078 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
91	5.761447883	192.168.43.195	192.168.43.55	TCP	60	443 → 42078 [RST] Seq=1 Win=0 Len=0
92	5.812391615	192.168.43.55	192.168.43.195	TCP	58	42100 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
93	5.822172243	192.168.43.195	192.168.43.55	TCP	60	443 → 42199 [RST] Seq=1 Win=0 Len=0
94	5.822289765	192.168.43.195	192.168.43.248	TCP	58	42149 → 11 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
95	5.874895887	192.168.43.55	192.168.43.195	TCP	58	42148 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
96	5.878887608	192.168.43.248	192.168.43.195	TCP	60	11 → 443 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
97	5.878894395	192.168.43.195	192.168.43.55	TCP	60	443 → 42148 [RST] Seq=1 Win=0 Len=0
98	5.898270172	192.168.43.55	192.168.43.195	TCP	58	42155 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
99	5.918429733	192.168.43.195	192.168.43.55	TCP	60	443 → 42155 [RST] Seq=1 Win=0 Len=0
100	5.972712764	192.168.43.55	192.168.43.195	TCP	58	42154 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
101	5.975253896	192.168.43.195	192.168.43.55	TCP	60	443 → 42154 [RST] Seq=1 Win=0 Len=0

Frame 93: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0
 Ethernet II, Src: VMware\_e9:aa:f0 (00:0c:29:e9:aa:f0), Dst: VMware\_e7:bd:4c (00:0c:29:e7:bd:4c)
 Internet Protocol Version 4, Src: 192.168.43.195, Dst: 192.168.43.55
 0100 .... = Version: 4
 .... 0101 = Header Length: 20 bytes (5)
 Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 40
 Identification: 0x0005 (88) (88)
 Flags: 0x4000, Don't fragment
 Fragment offset: 0
 Time to live: 128
 Protocol: TCP (6)
 Header checksum: 0x222d [validation disabled]
 [Header checksum status: Unverified]
 Source: 192.168.43.195
 Destination: 192.168.43.55
 Transmission Control Protocol, Src Port: 443, Dst Port: 42190, Seq: 1, Len: 0
 Source Port: 443
 Destination Port: 42190
 [Stream index: 14]
 [TCP Segment Len: 0]
 Sequence number: 1 (relative sequence 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

ip.addr == 192.168.43.55 or ip.addr == 192.168.43.248 and ip.addr == 192.168.43.195						
No.	Time	Source	Destination	Protocol	Length	Info
88	5.687442855	192.168.43.55	192.168.43.195	TCP	58	42149 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
89	5.710505694	192.168.43.195	192.168.43.55	TCP	60	443 → 42149 [RST] Seq=1 Win=0 Len=0
90	5.714890952	192.168.43.55	192.168.43.195	TCP	58	42078 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
91	5.761447883	192.168.43.195	192.168.43.55	TCP	60	443 → 42078 [RST] Seq=1 Win=0 Len=0
92	5.812391615	192.168.43.55	192.168.43.195	TCP	58	42199 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
93	5.822172243	192.168.43.195	192.168.43.55	TCP	60	443 → 42190 [RST] Seq=1 Win=0 Len=0
94	5.822289705	192.168.43.195	192.168.43.248	TCP	58	[TCP Retransmission] 443 → 11 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
95	5.874895807	192.168.43.55	192.168.43.195	TCP	58	42148 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
96	5.878887608	192.168.43.248	192.168.43.195	TCP	60	443 → 443 [RST] Seq=1 Win=0 Len=0
97	5.878894395	192.168.43.195	192.168.43.55	TCP	60	443 → 42148 [RST] Seq=1 Win=0 Len=0
98	5.898276172	192.168.43.55	192.168.43.195	TCP	58	42155 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
99	5.918429733	192.168.43.195	192.168.43.55	TCP	60	443 → 42155 [RST] Seq=1 Win=0 Len=0
100	5.972712764	192.168.43.55	192.168.43.195	TCP	58	42154 → 443 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0 MSS=1460
101	5.975253896	192.168.43.195	192.168.43.55	TCP	60	443 → 42154 [RST] Seq=1 Win=0 Len=0

```

▶ Frame 97: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0
▶ Ethernet II, Src: VMware_e9:aa:f0 (00:0c:29:e9:aa:f0), Dst: VMware_e7:bd:4c (00:0c:29:e7:bd:4c)
▶ Internet Protocol Version 4, Src: 192.168.43.195, Dst: 192.168.43.55
    0100 . . . . . Version: 4
    ... 0101 = Header Length: 20 bytes (5)
    ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
    Identification: 0x005 (89)
    ▶ Flags: 0x4000, Don't fragment
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (6)
    Header checksum: 0x222c [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.43.195
    Destination: 192.168.43.55
    ▶ Transmission Control Protocol, Src Port: 443, Dst Port: 42148, Seq: 1, Len: 0
        Source Port: 443
        Destination Port: 42148
        [Stream index: 15]

```

This difference of 1 signifies 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.