

Understanding Nmap Scan with Wireshark

August 20, 2017 By Raj Chandel

In this article, you will learn how to capture network packet using Wireshark when an attacker is scanning target using NMAP port scanning method. Here you will notice that how Wireshark captured different network traffic packet for open and close ports.

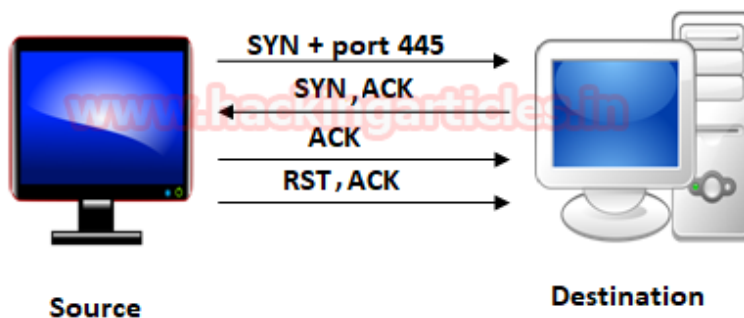
Note: The Below Practical is performed with the same IP address (192.168.1.102), which you will notice is common for our Windows and Linux Machine, you may differentiate them by their MAC addresses in this case.

Let's start!!!

TCP Scan

Tcp scan will scan for TCP port like port 22, 21, 23, 445 etc and ensure for listening port (open) through 3-way handshake connection between the source and destination port. If the port is open then source made request with **SYN** packet, a response destination sent **SYN, ACK** packet and then source sent **ACK** packets, at last source again sent **RST, ACK** packets.

TCP SCAN For Open Port



Type following NMAP command for TCP scan as well as start Wireshark on another hand to capture the sent Packet.

```
nmap -sT -p 445 192.168.1.102
```

From the given image you can observe the result that port **445** is **open**.

```
root@kali:~# nmap -sT -p 445 192.168.1.102

Starting Nmap 7.50 ( https://nmap.org ) at 2017-08-18 02:05 EDT
Nmap scan report for 192.168.1.102
Host is up (0.087s latency).
www.hackingarticles.in
PORT      STATE SERVICE
445/tcp   open  microsoft-ds
MAC Address: 0C:D2:92:82:EE:02 (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 13.39 seconds
root@kali:~#
```

Look over the sequence of packet transfer between source and destination captured through Wireshark.

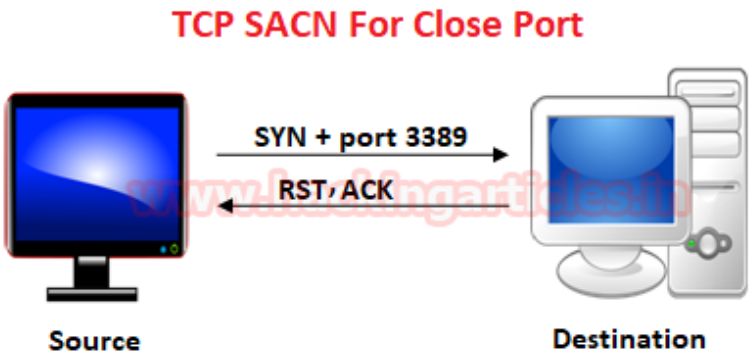
You will notice that it has captured the same sequence of the flag as described above:

- Source sent SYN packet to the destination
- Destination sent SYN, ACK to source
- Source sent ACK packet to the destination
- Source again sent RST, ACK to destination

| ip.addr == 192.168.1.113 | | | | | | | Expression... | + |
|--------------------------|-----------|---------------|---------------|------|--------|-----------------------------------------------------|---------------|---|
| No. | Time | Source | Destination | Prot | Length | Info | | |
| 129 | 37.411... | 192.168.1.113 | 192.168.1.102 | T... | 74 | 52944 → 445 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 | | |
| 132 | 37.415... | 192.168.1.102 | 192.168.1.113 | T... | 74 | 445 → 52944 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 | | |
| 133 | 37.415... | 192.168.1.113 | 192.168.1.102 | T... | 66 | 52944 → 445 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TS... | | |
| 134 | 37.415... | 192.168.1.113 | 192.168.1.102 | T... | 66 | 52944 → 445 [RST, ACK] Seq=1 Ack=1 Win=29312 Len=0 | | |

Let’s figure out network traffic for the close port. According to a given image, it is showing if scanning port is closed then 3-way handshake connection would be not possible between source and destination.

Source sent SYN pack and if the port is close the receiver will be sent a response through RST, ACK.



Type following NMAP command for TCP scan as well as start Wireshark on another hand to capture the sent Packet.

```
nmap -sT -p 3389 192.168.1.102
```

From the given image you can observe the result that port **3389** is **closed**.

```
root@kali:~# nmap -sT -p 3389 192.168.1.102

Starting Nmap 7.50 ( https://nmap.org ) at 2017-08-18 03:54 EDT
Nmap scan report for 192.168.1.102
Host is up (0.049s latency).

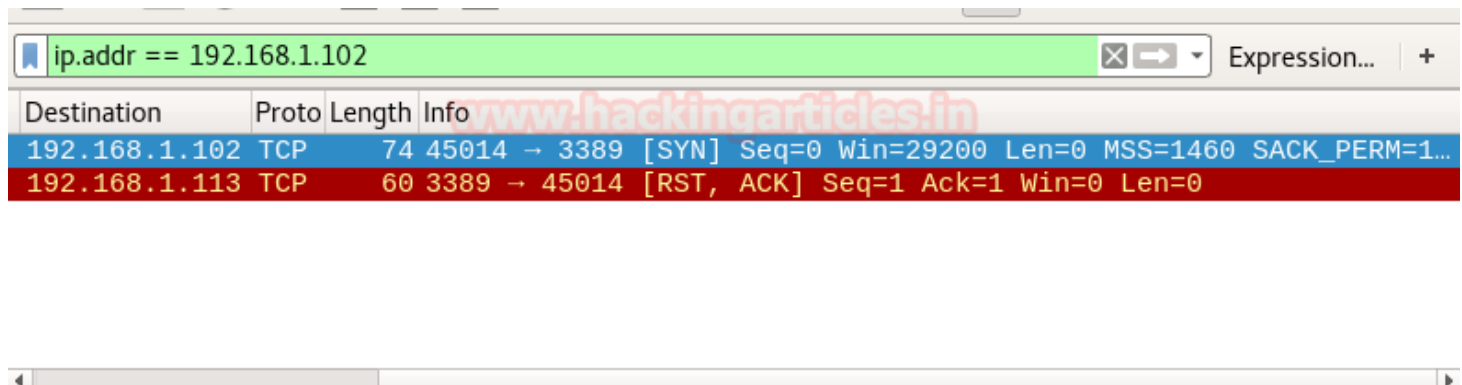
PORT      STATE SERVICE
3389/tcp  closed ms-wbt-server
MAC Address: 0C:D2:92:82:EE:02 (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 13.59 seconds
```

Look over the sequence of packet transfer between source and destination captured through Wireshark.

You will notice that it has captured the same sequence of the flag as described above:

- Source sent SYN packet to the destination
- Destination sent RST, ACK packet to the source



| Destination | Proto | Length | Info |
|---------------|-------|--------|------------------------------------------------------------------|
| 192.168.1.102 | TCP | 74 | 45014 → 3389 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1... |
| 192.168.1.113 | TCP | 60 | 3389 → 45014 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0 |

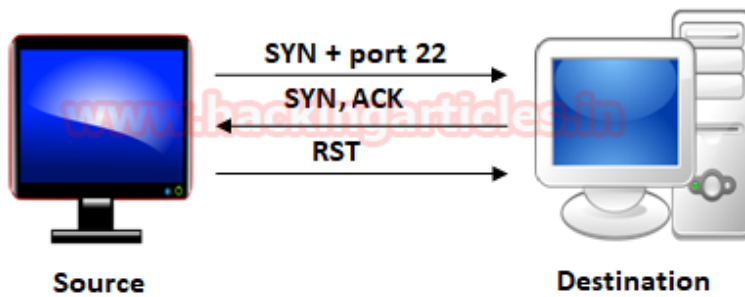
Stealth Scan

SYN scan is the default and most popular scan option for good reasons. It can be performed quickly, scanning thousands of ports per second on a fast network not hampered by restrictive firewalls. It is also relatively typical and stealthy since it never completes TCP connections.

The port is also considered open if an SYN packet (without the ACK flag) is received in response.

This technique is often referred to as half-open scanning because you don't open a full TCP connection. You send an SYN packet as if you are going to open a real connection and then wait for a response. An SYN, ACK indicates the port is listening (open)

STEALTH SCAN For Open Port



Type following NMAP command for TCP scan as well as start Wireshark on another hand to capture the sent Packet.

```
nmap -sS -p 22 192.168.1.102
```

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

```
root@kali:~# nmap -sS -p 22 192.168.1.102
Starting Nmap 7.50 ( https://nmap.org ) at 2017-08-18 04:10 EDT
Nmap scan report for 192.168.1.102
Host is up (0.046s latency).
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: AC:E0:10:E0:47:89 (Liteon Technology)
Nmap done: 1 IP address (1 host up) scanned in 13.63 seconds
```

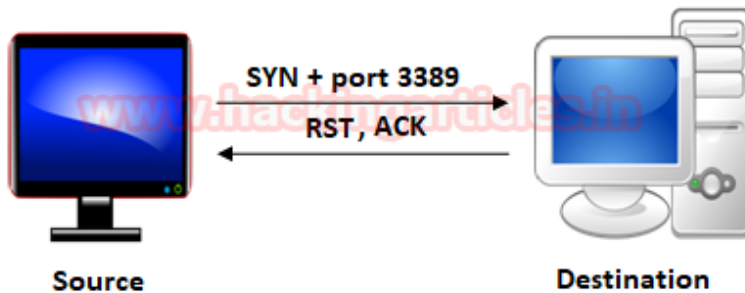
Look over the sequence of packet transfer between source and destination captured through Wireshark

- Source sent SYN packets to the destination
- Destination sent SYN, ACK packets to the source
- Source sent RST packets to the destination

| Destination | Proto | Length | Info |
|---------------|-------|--------|------------------------------------------------------------|
| 192.168.1.102 | TCP | 58 | 65008 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |
| 192.168.1.113 | TCP | 60 | 22 → 65008 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 |
| 192.168.1.102 | TCP | 54 | 65008 → 22 [RST] Seq=1 Win=0 Len=0 |

Now figure out traffic for close port using stealth scan. When source sent SYN packet on the specific port then if the port is closed then the destination will reply by sending RST packet.

STEALTH SCAN For Close Port



Type following NMAP command for TCP scan as well as start Wireshark on another hand to capture the sent Packet.

```
nmap -sS -p 3389 192.168.1.102
```

From the given image you can observe the result that port **3389** is **closed**.

```
root@kali:~# nmap -sS -p 3389 192.168.1.102

Starting Nmap 7.50 ( https://nmap.org ) at 2017-08-18 04:07 EDT
Nmap scan report for 192.168.1.102
Host is up (0.043s latency).
www.hackingarticles.in
PORT      STATE SERVICE
3389/tcp  closed ms-wbt-server
MAC Address: AC:E0:10:E0:47:89 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 13.62 seconds
root@kali:~#
```

Look over the sequence of packet transfer between source and destination captured through Wireshark

- Source sent SYN packets to the destination
- Destination sent RST, ACK packets to the destination

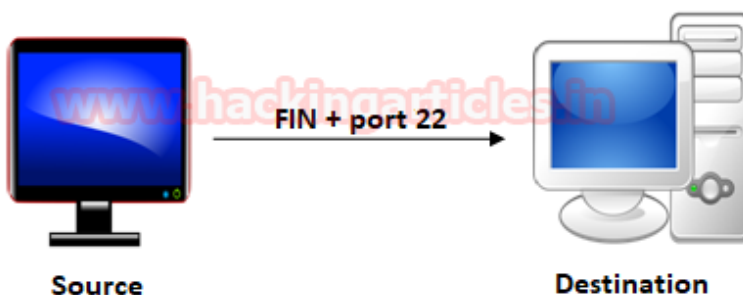
| ip.addr == 192.168.1.102 | | | | Expression... | + |
|--------------------------|-------|--------|--------------|-------------------------------------|---|
| Destination | Proto | Length | Info | | |
| 192.168.1.102 | TCP | 58 | 57273 → 3389 | [SYN] Seq=0 Win=1024 Len=0 MSS=1460 | |
| 192.168.1.113 | TCP | 60 | 3389 → 57273 | [RST, ACK] Seq=1 Ack=1 Win=0 Len=0 | |

Fin Scan

A FIN packet is used to terminate the TCP connection between the source and destination port typically after the data transfer is complete. In the place of an SYN packet, Nmap starts a FIN scan by using a FIN packet. If the port is open then no response will come from destination port when FIN packet is sent through source port.

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

FIN SCAN For Open Port



Type following NMAP command for TCP scan as well as start Wireshark on another hand to capture the sent Packet.

```
nmap -sF -p 22 192.168.1.102
```

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

```
root@kali:~# nmap -sF -p 22 192.168.1.102

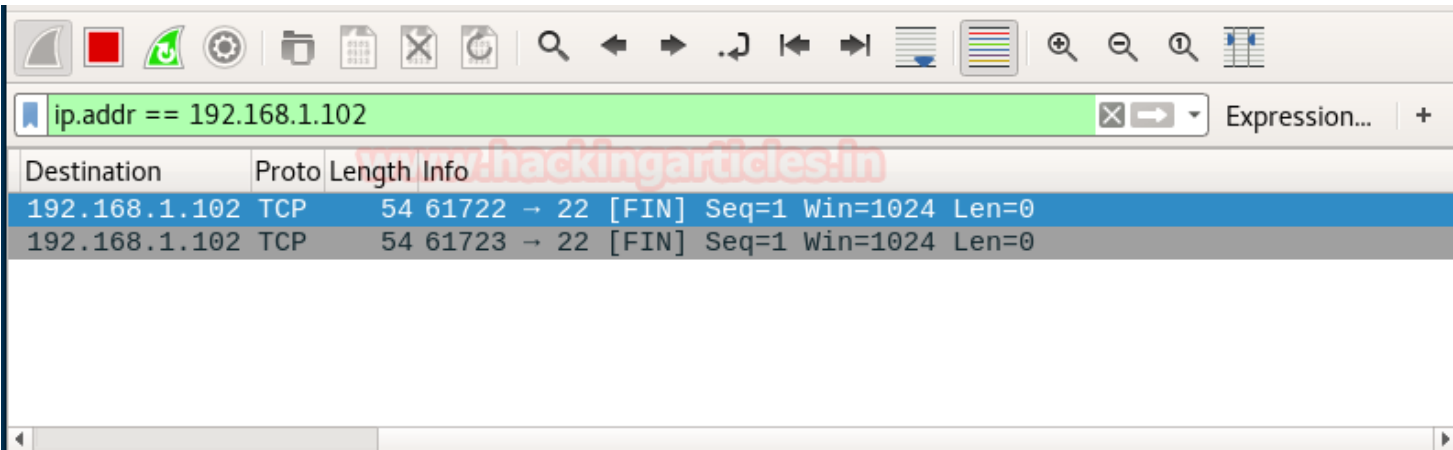
Starting Nmap 7.50 ( https://nmap.org ) at 2017-08-18 04:20 EDT
Nmap scan report for 192.168.1.102
Host is up (0.085s latency).

PORT      STATE SERVICE
22/tcp    open|filtered ssh
MAC Address: AC:E0:10:E0:47:89 (Liteon Technology)

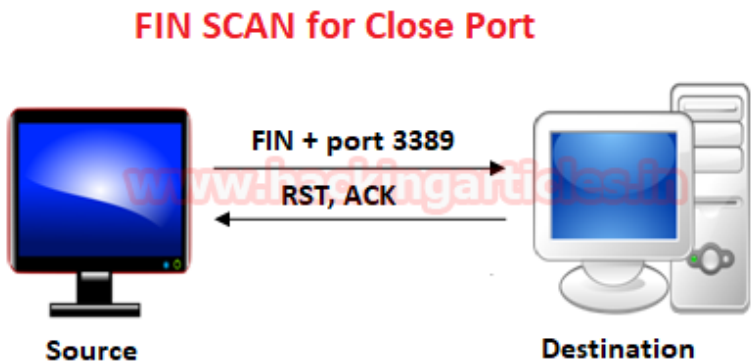
Nmap done: 1 IP address (1 host up) scanned in 14.29 seconds
```

Look over the sequence of packet transfer between source and destination captured through Wireshark

- Source sent FIN packets to the destination
- Destination sent no reply to the source



Similarly, if Fin scan is performed against any close then source port will be sent FIN packet to specific port and destination will reply by sending RST, ACK packets.



Type following NMAP command for TCP scan as well as start Wireshark on another hand to capture the sent Packet.

```
nmap -sF -p 3389 192.168.1.102
```

From the given image you can observe the result that port **3389** is **closed**.

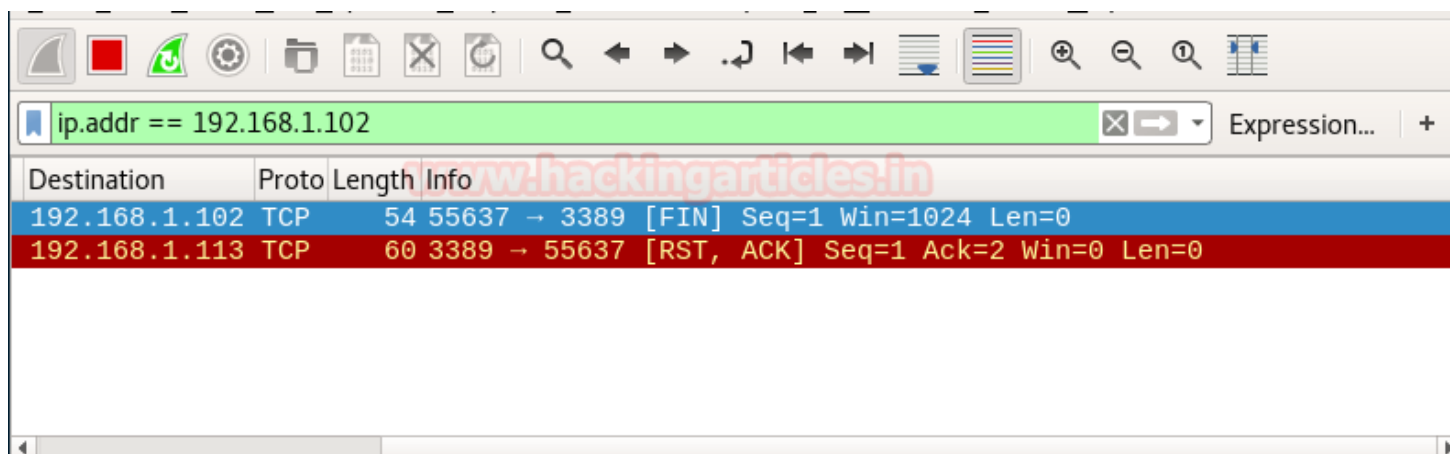
```
root@kali:~# nmap -sF -p 3389 192.168.1.102

Starting Nmap 7.50 ( https://nmap.org ) at 2017-08-18 04:22 EDT
Nmap scan report for 192.168.1.102
Host is up (0.065s latency).
www.hackingarticles.in
PORT      STATE SERVICE
3389/tcp  closed ms-wbt-server
MAC Address: AC:E0:10:E0:47:89 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 13.62 seconds
```

Look over the sequence of packet transfer between source and destination captured through Wireshark

- Source sent SYN packets to the destination
- Destination sent RST packets to the destination

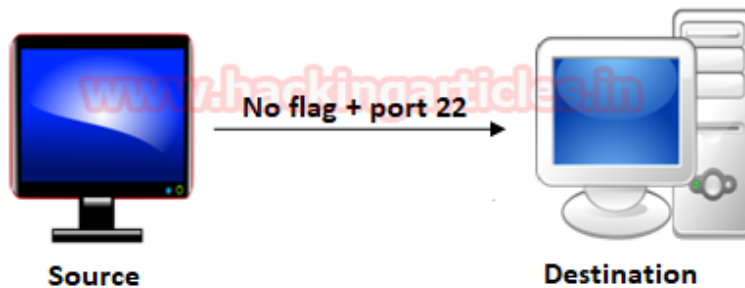


Null Scan

A Null Scan is a series of TCP packets which hold a sequence number of “zeros” (00000000) and since there are none flags set, the destination will not know how to reply the request. It will discard the packet and no reply will be sent, which indicate that the port is open.

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

NULL SCAN For Open Port



Type following NMAP command for TCP scan as well as start Wireshark on another hand to capture the sent Packet.

```
nmap -sN -p 22 192.168.1.102
```

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

```
root@kali:~# nmap -sN -p 22 192.168.1.102

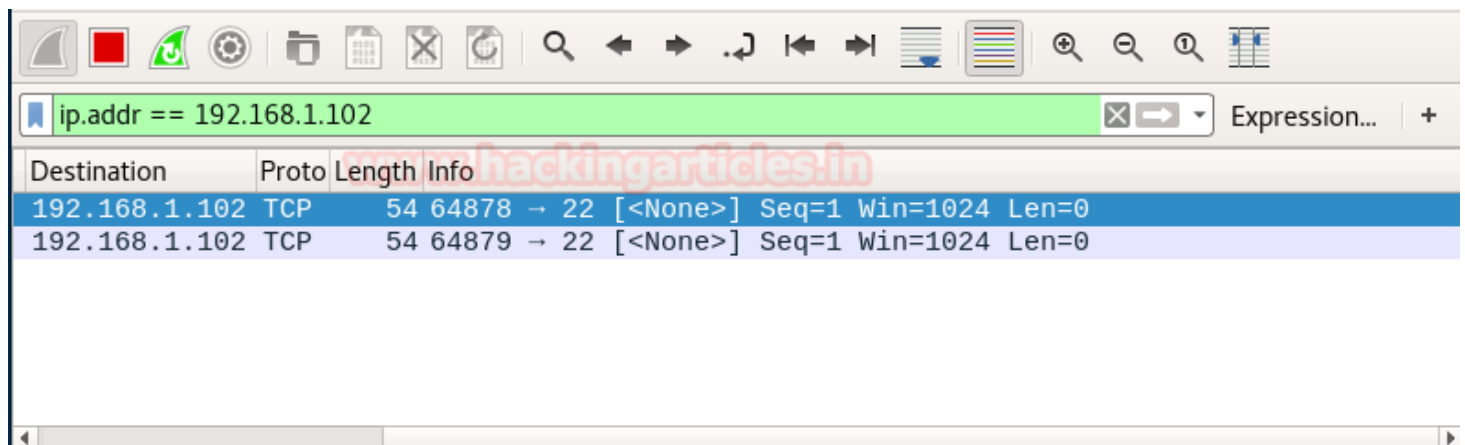
Starting Nmap 7.50 ( https://nmap.org ) at 2017-08-18 04:26 EDT
Nmap scan report for 192.168.1.102
Host is up (0.071s latency).

PORT      STATE      SERVICE
22/tcp    open|filtered ssh
MAC Address: AC:E0:10:E0:47:89 (Liteon Technology)

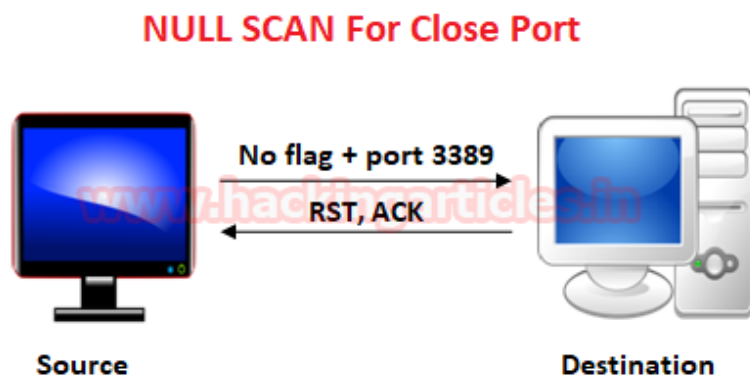
Nmap done: 1 IP address (1 host up) scanned in 14.17 seconds
```

Look over the sequence of packet transfer between source and destination captured through Wireshark

- Source sent Null packets to the destination
- Destination sent no reply to the source



If the port is closed, the Destination will send an RST, ACK packet in response when source send null packets on a specific port



Type following NMAP command for TCP scan as well as start Wireshark on another hand to capture the sent Packet.

```
nmap -sN -p 3389 192.168.1.102
```

From the given image you can observe the result that port **3389** is **closed**.

```
root@kali:~# nmap -sN -p 3389 192.168.1.102

Starting Nmap 7.50 ( https://nmap.org ) at 2017-08-18 04:30 EDT
Nmap scan report for 192.168.1.102
Host is up (0.063s latency).

PORT      STATE SERVICE
3389/tcp  closed ms-wbt-server
MAC Address: AC:E0:10:E0:47:89 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 13.63 seconds
```

Look over the sequence of packet transfer between source and destination captured through Wireshark

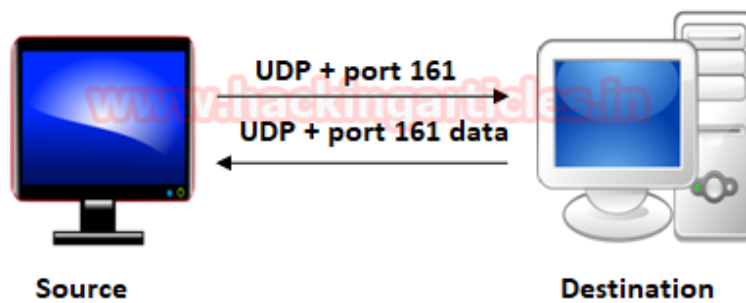
- Source sent Null (none) packets to the destination
- Destination sent RST, ACK to source

| ip.addr == 192.168.1.102 | | | | | Expression... | + |
|--------------------------|-------|--------|--------------|------------------------------------|---------------|---|
| Destination | Proto | Length | Info | | | |
| 192.168.1.102 | TCP | 54 | 62532 → 3389 | [<None>] Seq=1 Win=1024 Len=0 | | |
| 192.168.1.113 | TCP | 60 | 3389 → 62532 | [RST, ACK] Seq=1 Ack=1 Win=0 Len=0 | | |

UDP Scan

UDP scan works by sending a UDP packet to every destination port; it is a connectionless protocol. For some common ports such as 53 and 161, a protocol-specific payload is sent to increase the response rate, a service will respond with a UDP packet, proving that it is open. If no response is received after retransmissions, the port is classified as open|filtered. This means that the port could be open, or perhaps packet filters are blocking the communication.

UDP SCAN For Open Port



Type following NMAP command for TCP scan as well as start Wireshark on another hand to capture the sent Packet.

```
nmap -sU -p 161 192.168.1.119
```

From the given image you can observe the result that port **161** is **open**.

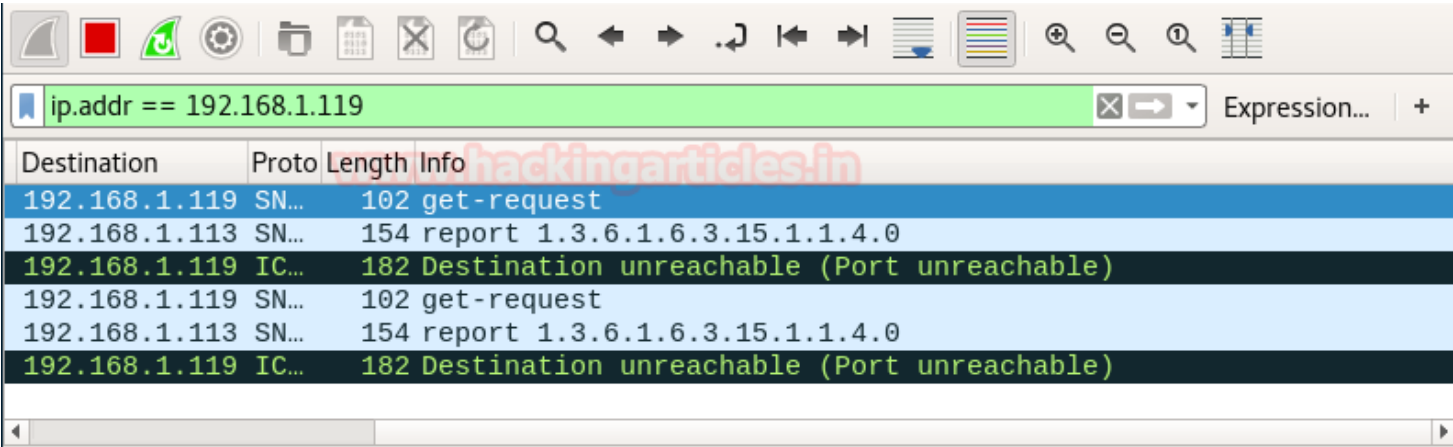
```
root@kali:~# nmap -sU -p 161 192.168.1.119

Starting Nmap 7.50 ( https://nmap.org ) at 2017-08-18 04:32 EDT
Nmap scan report for 192.168.1.119
Host is up (0.0013s latency).
www.hackingarticles.in
PORT      STATE SERVICE
161/udp   open  snmp
MAC Address: 00:0C:29:95:B8:D0 (VMware)

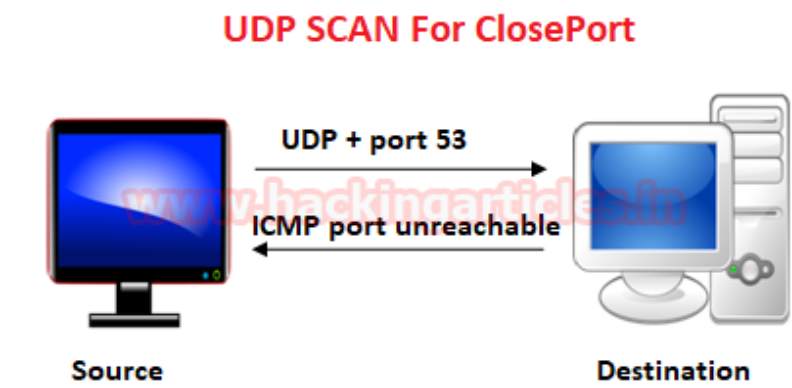
Nmap done: 1 IP address (1 host up) scanned in 13.64 seconds
```

Look over the sequence of packet transfer between source and destination captured through Wireshark

- Source sent UDP packets to the destination
- Destination sent UDP packet with some data to the source



Similarly, if source sent UDP packet on a close port to the destination then destination sent a reply with ICMP packet port unreachable with an appropriate error.



Type following NMAP command for TCP scan as well as start Wireshark on another hand to capture the sent Packet.

```
nmap -sU -p 53 192.168.1.119
```

From the given image you can observe the result that port **53** is **closed**.

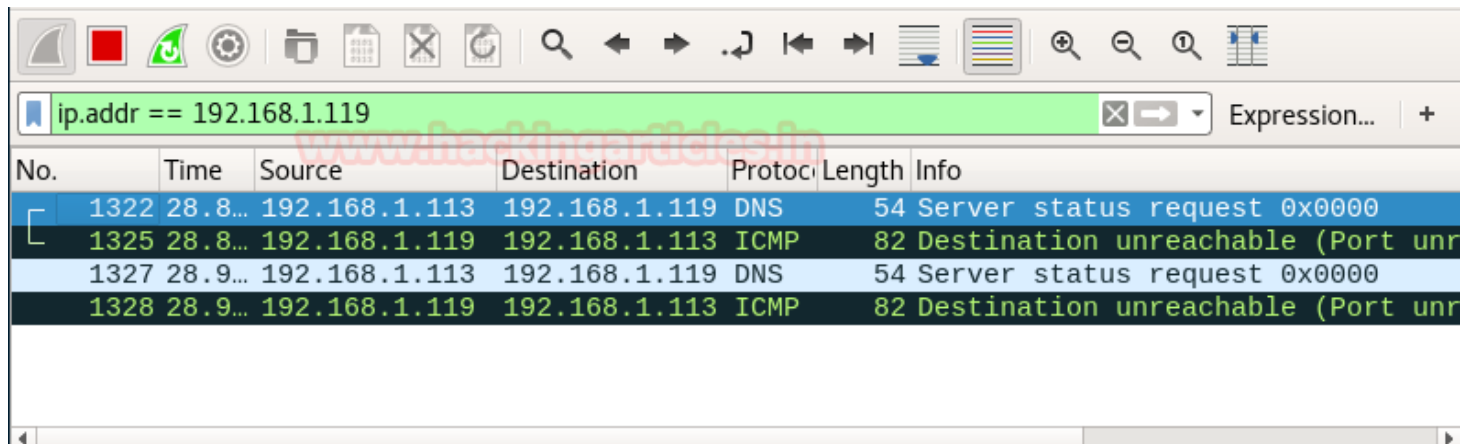
```
root@kali:~# nmap -sU -p 53 192.168.1.119

Starting Nmap 7.50 ( https://nmap.org ) at 2017-08-18 04:38 EDT
Nmap scan report for 192.168.1.119
Host is up (0.0016s latency).
www.hackingarticles.in
PORT      STATE SERVICE
53/udp    closed domain
MAC Address: 00:0C:29:95:B8:D0 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 13.63 seconds
```

Look over the sequence of packet transfer between source and destination captured through Wireshark

- Source sent UDP packets to the destination
- Destination sent ICMP packet port unreachable to the source

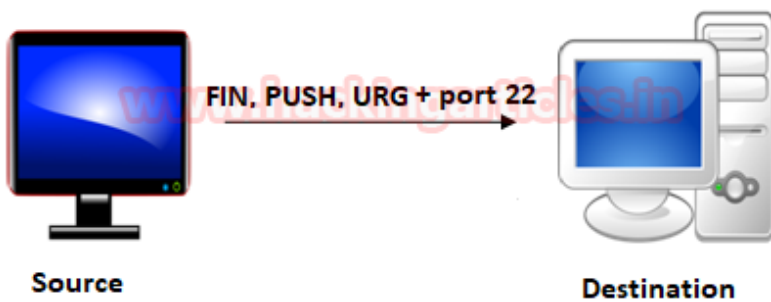


Xmas Scan

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

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

XMAS SCAN For Open Port



Type following NMAP command for TCP scan as well as start Wireshark on another hand to capture the sent Packet.

```
nmap -sX -p 22 192.168.1.102
```

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

```
root@kali:~# nmap -sX -p 22 192.168.1.102

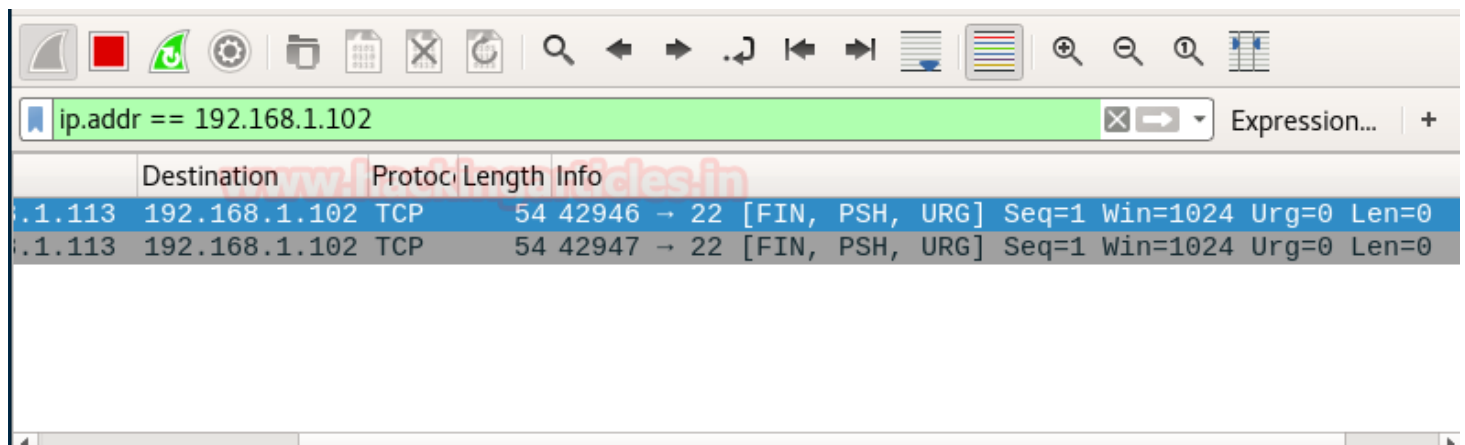
Starting Nmap 7.50 ( https://nmap.org ) at 2017-08-18 04:42 EDT
Nmap scan report for 192.168.1.102
Host is up (0.028s latency).

PORT      STATE      SERVICE
22/tcp    open|filtered ssh
MAC Address: AC:E0:10:E0:47:89 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 13.74 seconds
```

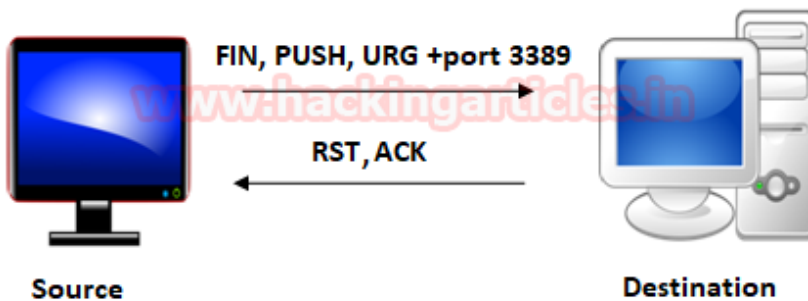
Look over the sequence of packet transfer between source and destination captured through Wireshark

- Source sent FIN, PUSH and URG packets to the destination
- Destination sent no reply to the source



Similarly, if source sent FIN, PUSH and URG packets to a specific port and if the port is closed then destination will be sent RST, ACK packets to the source.

XMAS SCAN For Close Port



Type following NMAP command for TCP scan as well as start Wireshark on another hand to capture the sent Packet.

```
nmap -sX -p 3389 192.168.1.102
```

From the given image you can observe the result that port **3389** is **closed**.

```
root@kali:~# nmap -sX -p 3389 192.168.1.102

Starting Nmap 7.50 ( https://nmap.org ) at 2017-08-18 04:44 EDT
Nmap scan report for 192.168.1.102
Host is up (0.020s latency).
PORT      STATE SERVICE
3389/tcp  closed ms-wbt-server
MAC Address: AC:E0:10:E0:47:89 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 13.84 seconds
```

Look over the sequence of packet transfer between source and destination captured through Wireshark

- Source sent FIN, PUSH and URG packets to the destination
- Destination RST, ACK packet to the source

| | | | | | ip.addr == 192.168.1.102 | | Expression... | + |
|---------------|---------------|----------|--------|---------------------------------------------------------|--------------------------|--|---------------|---|
| | Destination | Protocol | Length | Info | | | | |
| 192.168.1.113 | 192.168.1.102 | TCP | 54 | 36958 → 3389 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0 | | | | |
| 192.168.1.102 | 192.168.1.113 | TCP | 60 | 3389 → 36958 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0 | | | | |
| 192.168.1.113 | 192.168.1.102 | TCP | 54 | 36959 → 3389 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0 | | | | |
| 192.168.1.102 | 192.168.1.113 | TCP | 60 | 3389 → 36959 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0 | | | | |