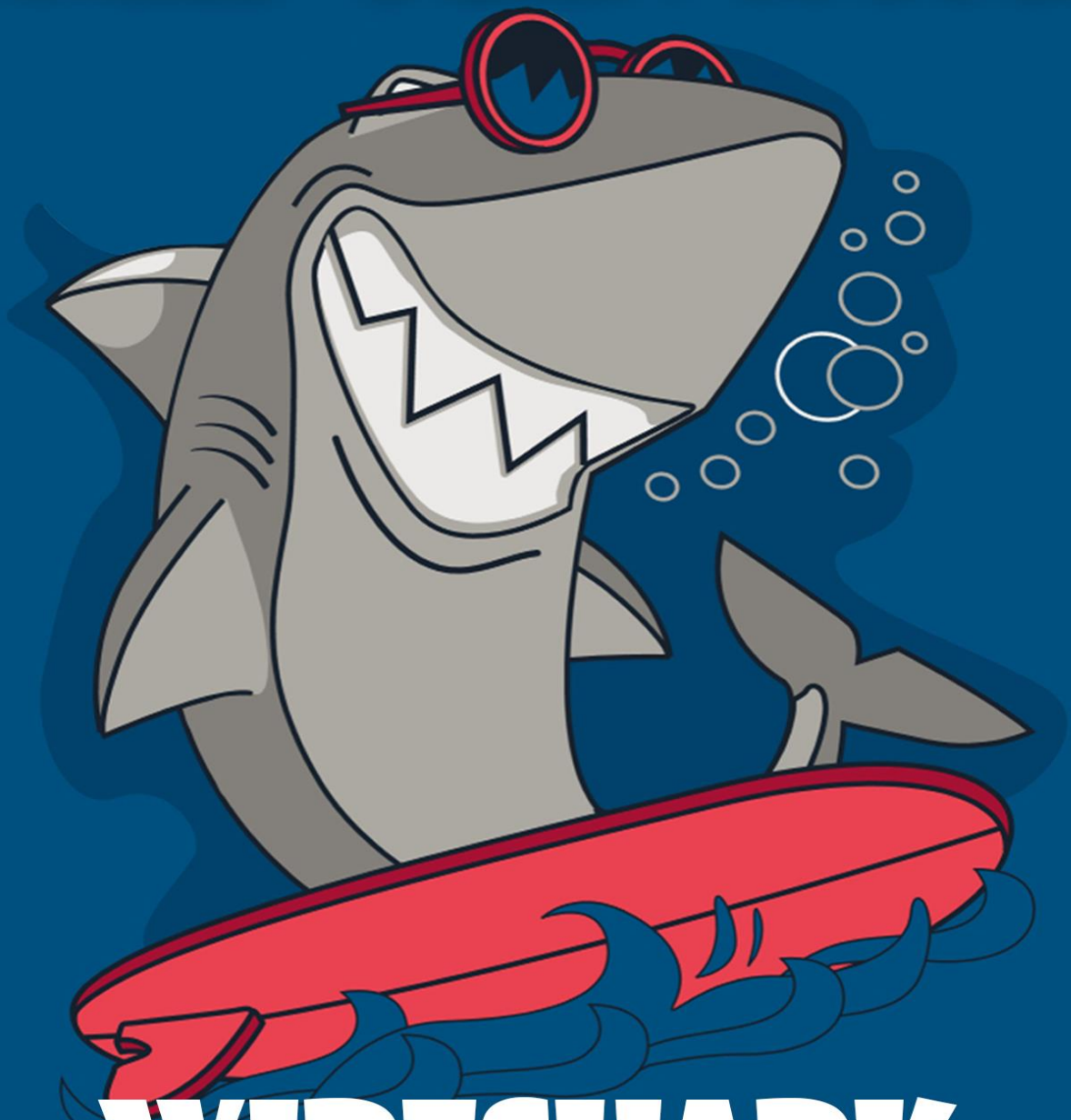
iGNITE Technologies

A DETAILED GUIDE TO NMAP SCAN WITH WIRESHARK



WWW.HACKINGARTICLES.IN

Contents

Introduction

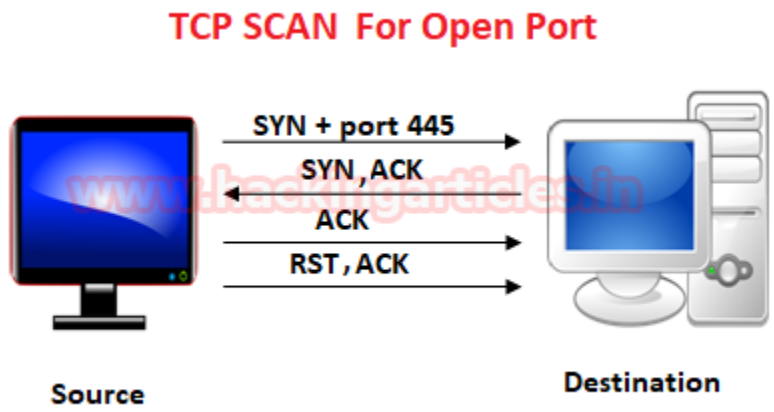
In this post, you will learn how to capture network packets using Wireshark when an attacker is scanning a target using the NMAP port scanning method. Here you will notice how Wireshark captured different network traffic packets for open and closed ports.

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 machines. You may differentiate them by their MAC addresses in this case.

Let's start!!!

TCP Scan

TCP Scan will scan for TCP ports like port 22, 21, 23, 445, etc. and ensure the listening port is open through a 3-way handshake connection between the source and destination port. If the port is open, the source sent an **SYN** packet, the response destination sent an **SYN** packet, the source sent **ACK** packets, and the source sent **RST** and **ACK** packets again.



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

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

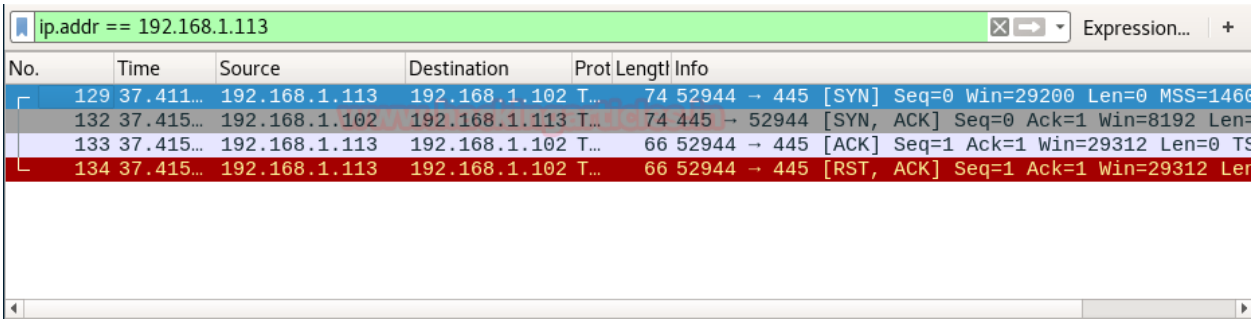
From the given image, you can observe that port **445 is open.**

```
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).

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
```

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

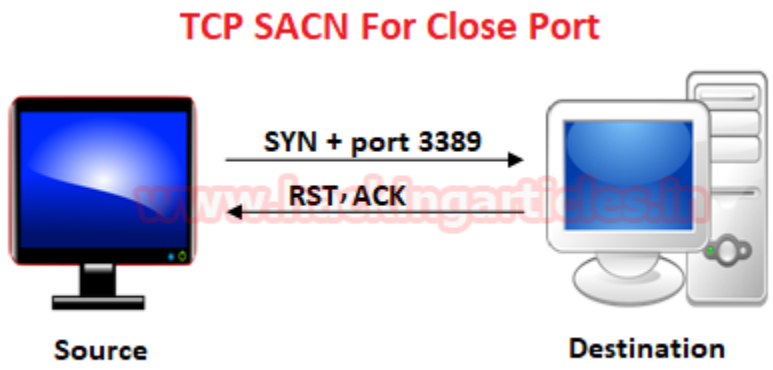
You will notice that it has captured the same sequence of the flag as described above:

- 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



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

The source sent a SYN pack and if the port is closed, the receiver will be sent a response through RST, ACK.



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

nmap -sT -p 3389 192.168.1.102

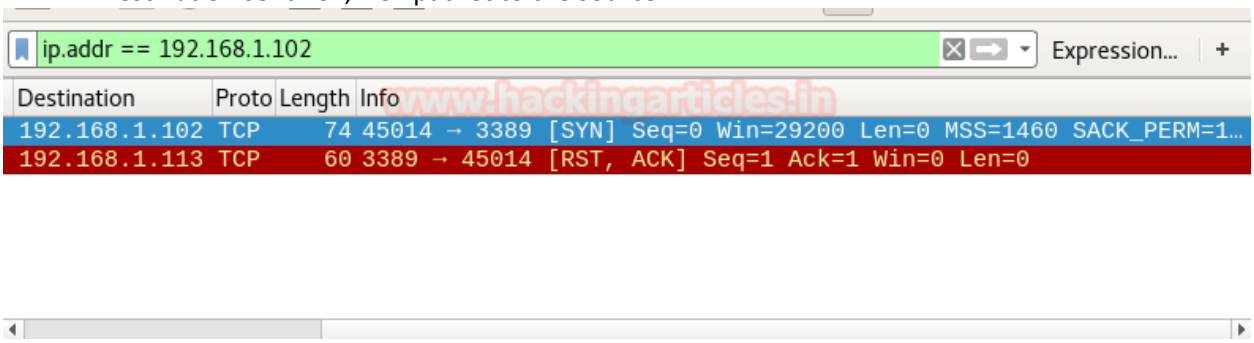
From the given image, you can observe that **port 3389** is closed.

```
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

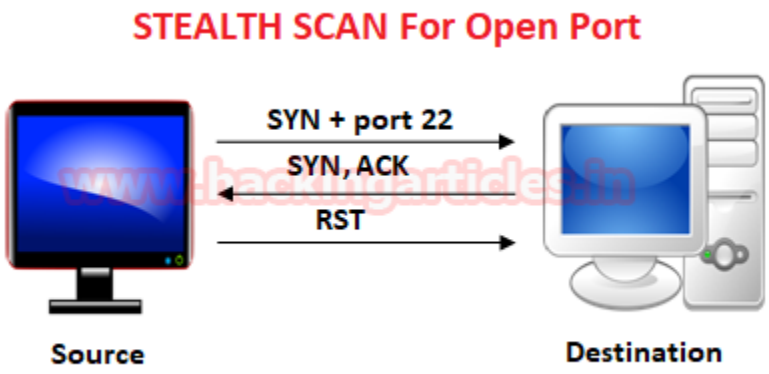


Stealth Scan

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.

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

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're going to establish a real connection, then wait for a response.An SYN, ACK indicates the port is listening (open).



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

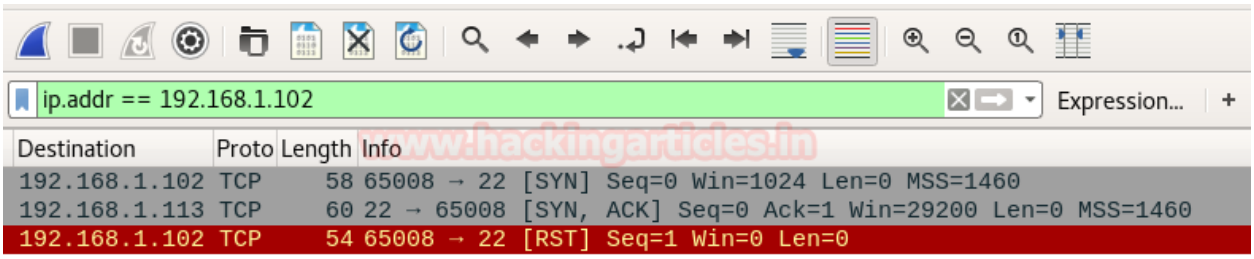
```
nmap -sS -p 22 192.168.1.102
```

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

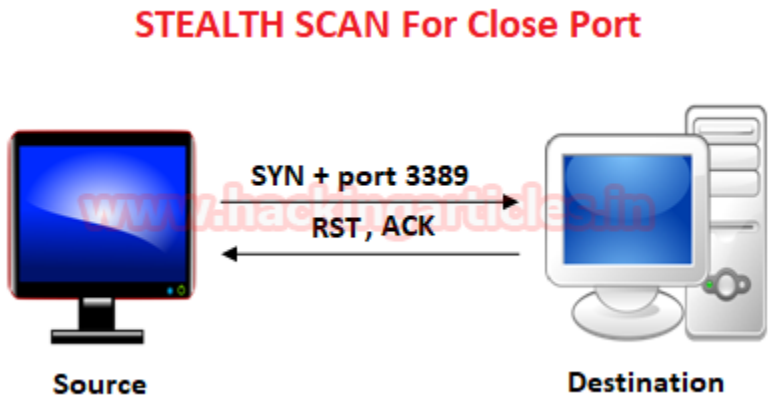
```
root@kali:~# nmap -sS -p 22 192.168.1.102

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



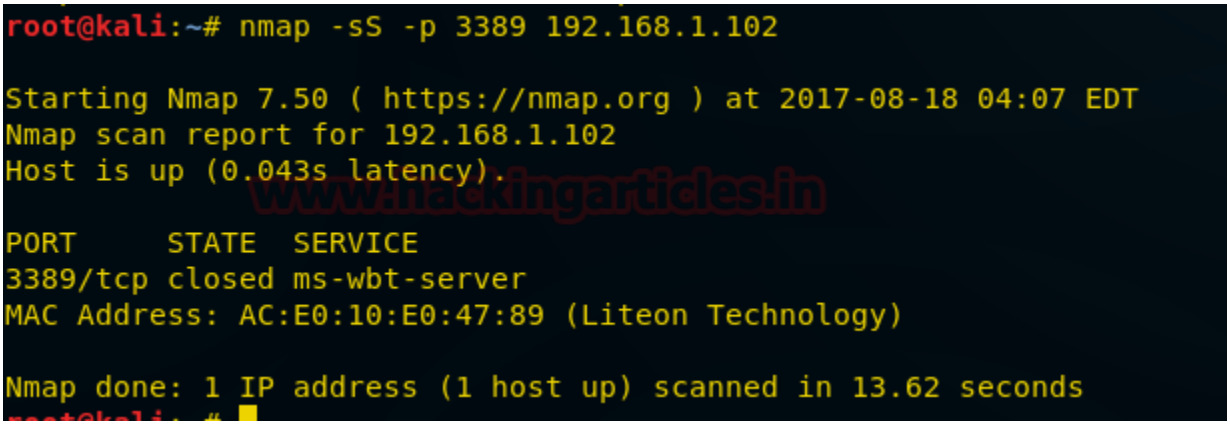
Now figure out traffic for the close port using a stealth scan. When the source sends a SYN packet to the specific port, if the port is closed, the destination will reply by sending an RST packet.



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

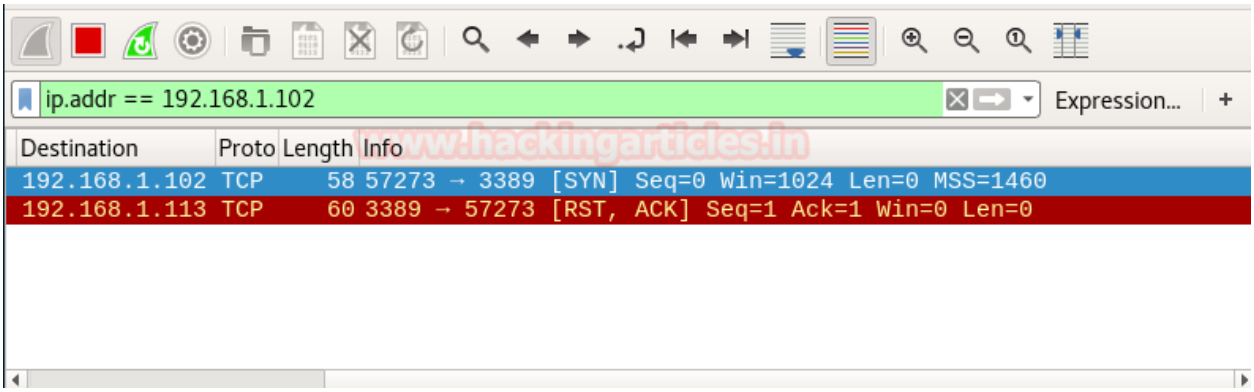
nmap -sS -p 3389 192.168.1.102

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



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

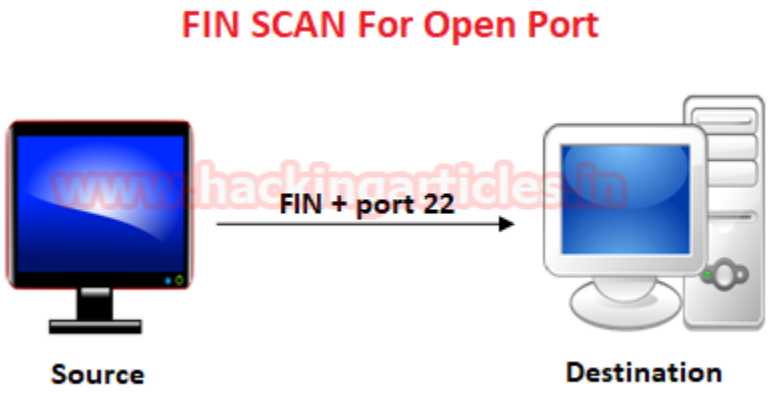
- Source sent SYN packets to the destination
- Destination sent RST, ACK packets to the destination



Fin Scan

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

Fin-Scan is only workable on Linux machines and does not work on the latest version of Windows.



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

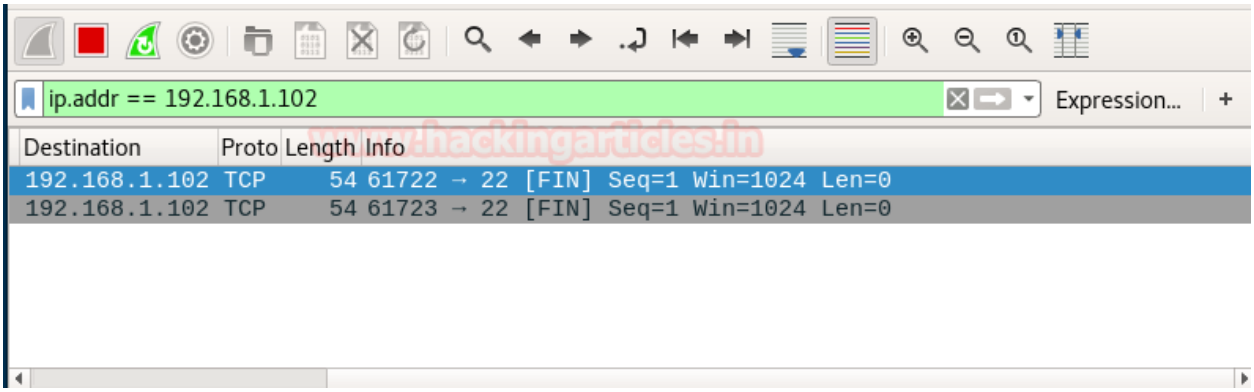
```
nmap -sF -p 22 192.168.1.102
```

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

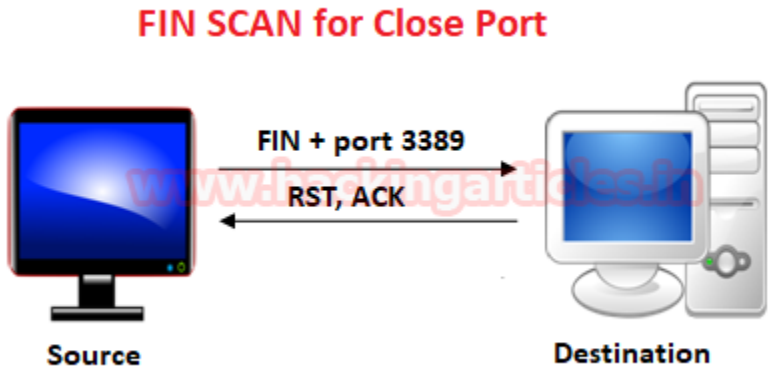
```
root@kali:~# nmap -sF -p 22 192.168.1.102

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 a Fin scan is performed against any close, then the source port will send a FIN packet to the specific port and the destination will reply by sending RST and ACK packets.



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

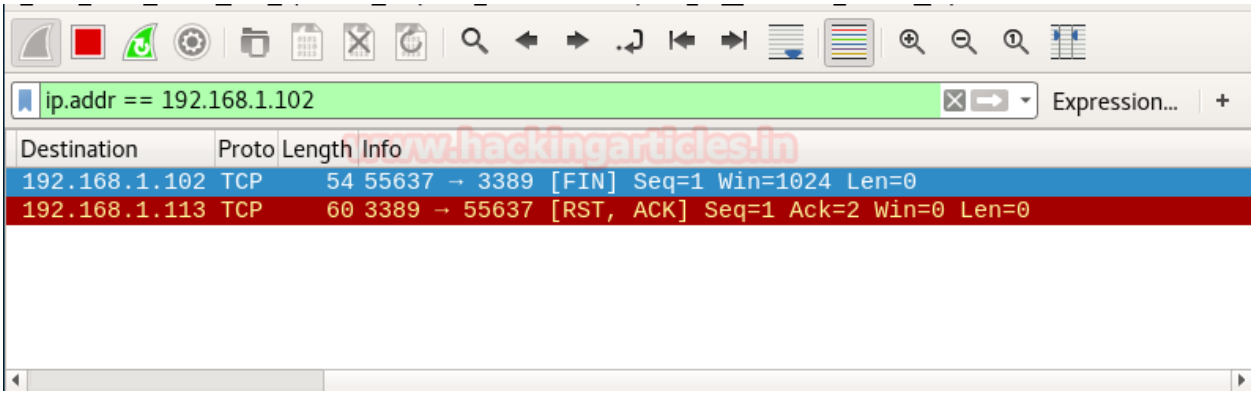
nmap -sF -p 3389 192.168.1.102

From the given image, you can observe 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).

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 transfers between source and destination captured through Wireshark.

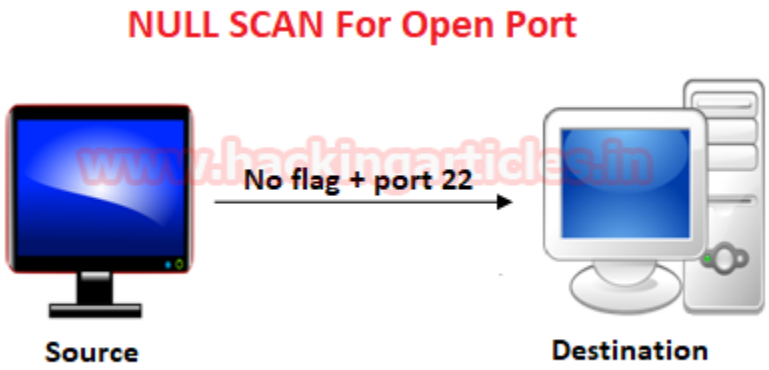
- Source sent SYN packets to the destination
- Destination sent RST packets to the destination



Null Scan

A Null Scan is a series of TCP packets that contain a sequence number of "zeroes" (0000000), and because no flags are set, the destination does not know how to respond to the request. It will discard the packet and no reply will be sent, which indicates that the port is open.

Null Scan is only workable on Linux machines and does not work on the latest version of Windows.



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

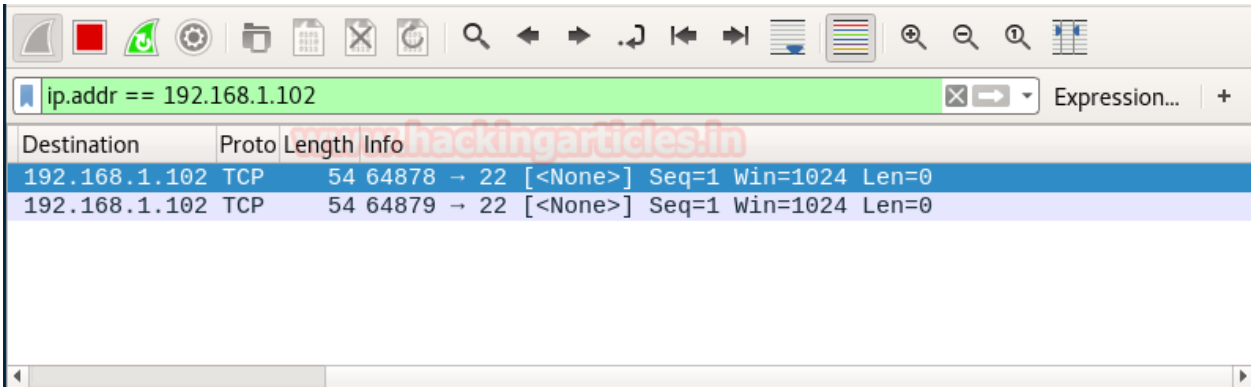
nmap -sN -p 22 192.168.1.102

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

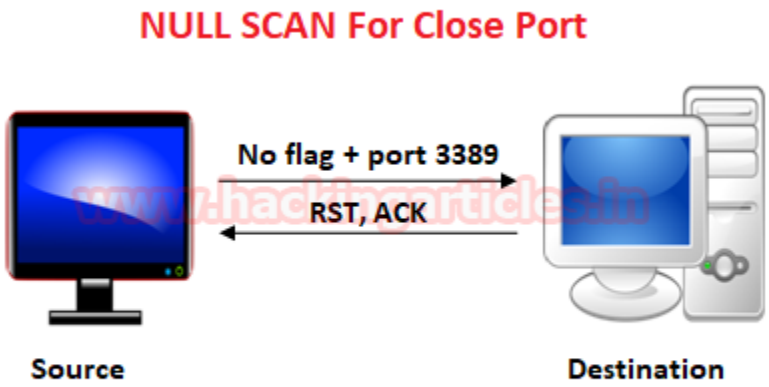
```
root@kali:~# nmap -sN -p 22 192.168.1.102

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 and an ACK packet in response when the source sends null packets on a specific port.



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

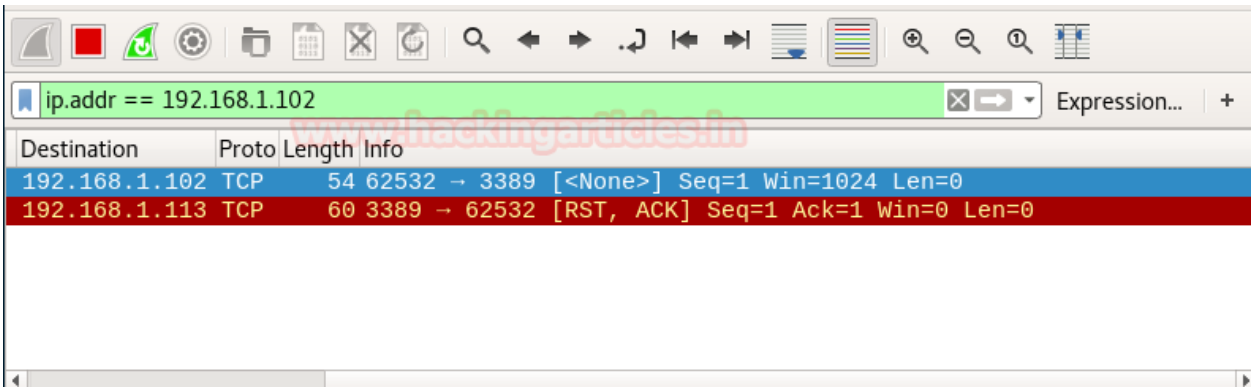
nmap -sN -p 3389 192.168.1.102

From the given image, you can observe 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
```

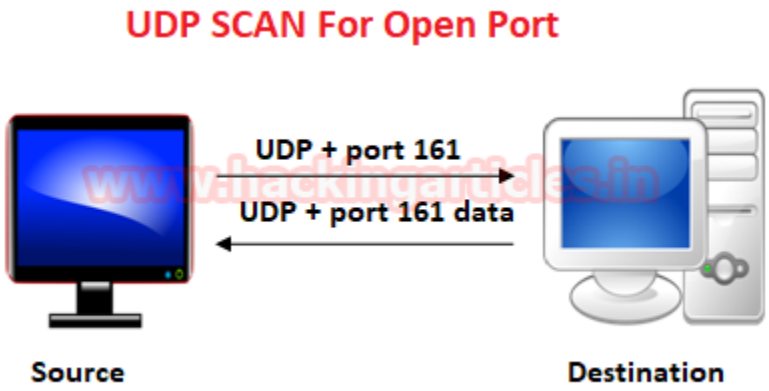
Look over the sequence of packet transfer between source and destination captured through Wireshark

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



UDP Scan

A 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.



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

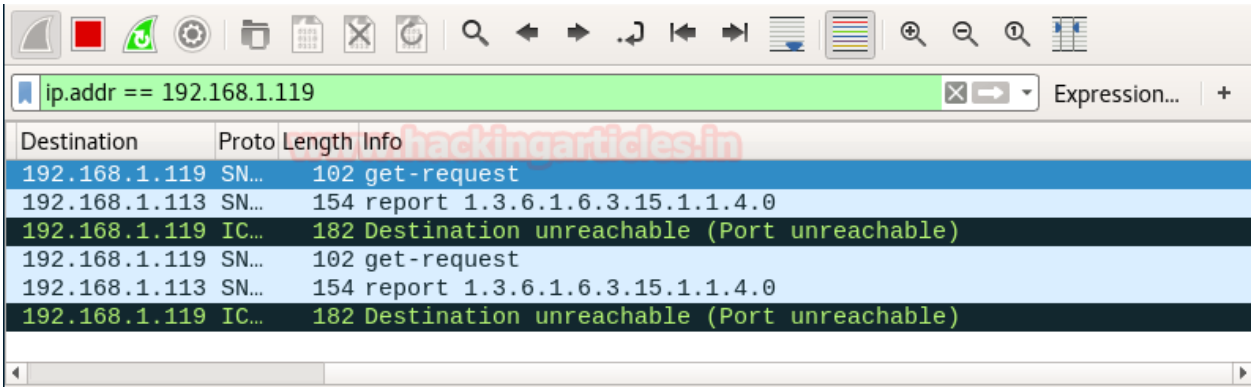
```
nmap -sU -p 161 192.168.1.119
```

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

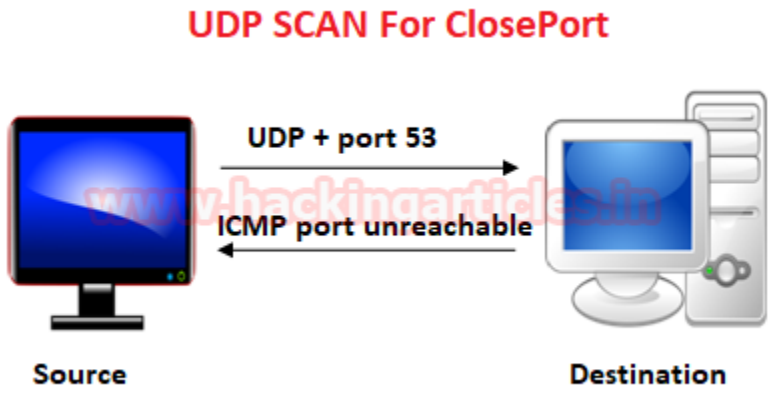
```
root@kali:~# nmap -sU -p 161 192.168.1.119

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).

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



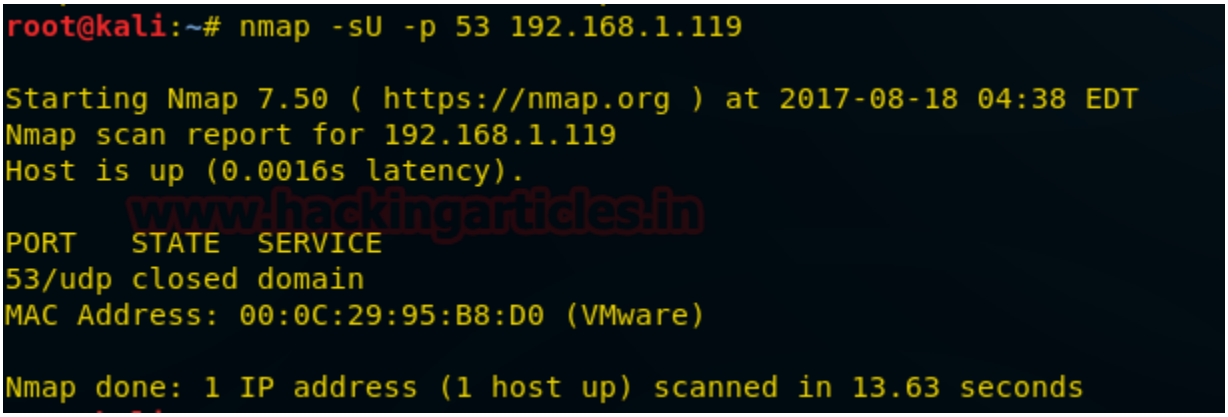
> Similarly, if a source sent an UDP packet on a close port to the destination, the destination would reply with an ICMP packet port unreachable with an appropriate error.



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

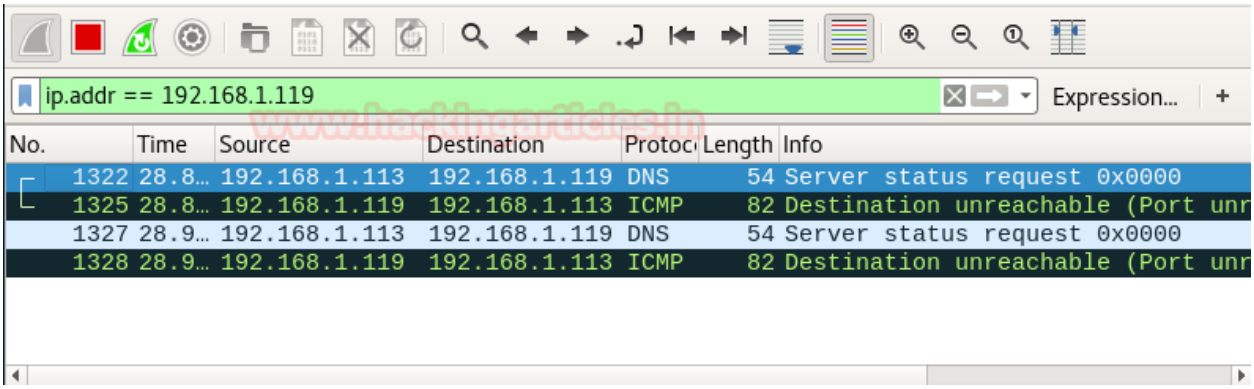
nmap -sU -p 53 192.168.1.119

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



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

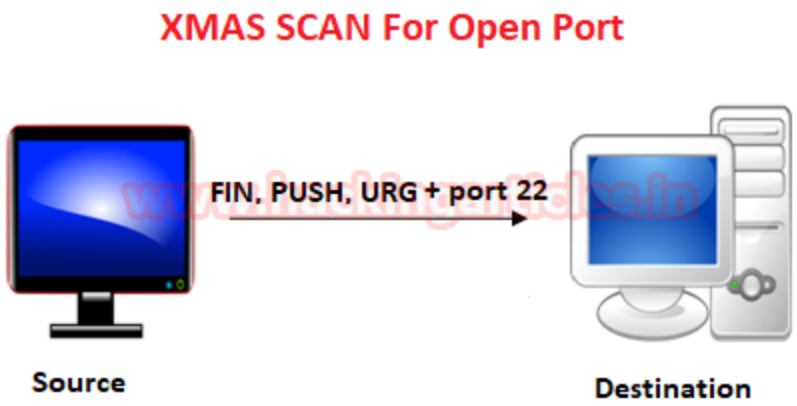
- Source sent UDP packets to the destination
- Destination sent ICMP packet port unreachable to the source



Xmas Scan

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

The Xmas Scan is only workable on Linux machines and does not work on the latest version of Windows.



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

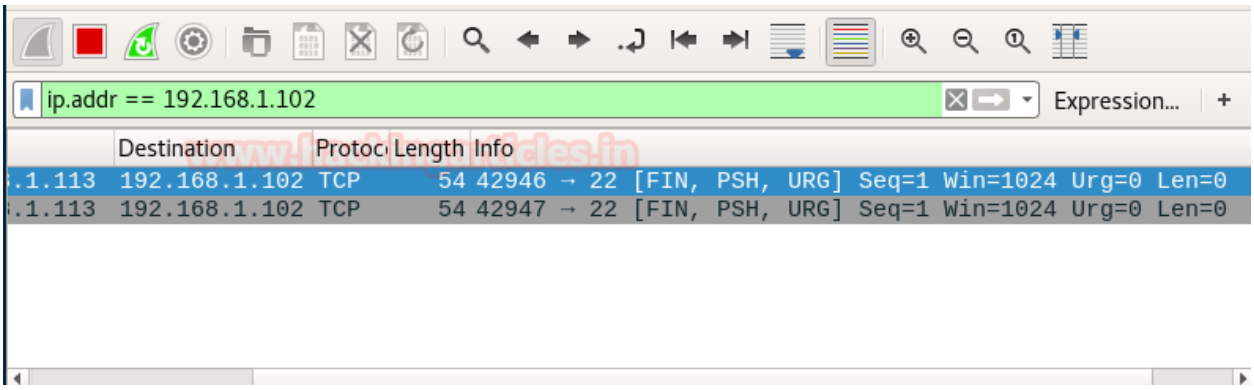
nmap -sX -p 22 192.168.1.102

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

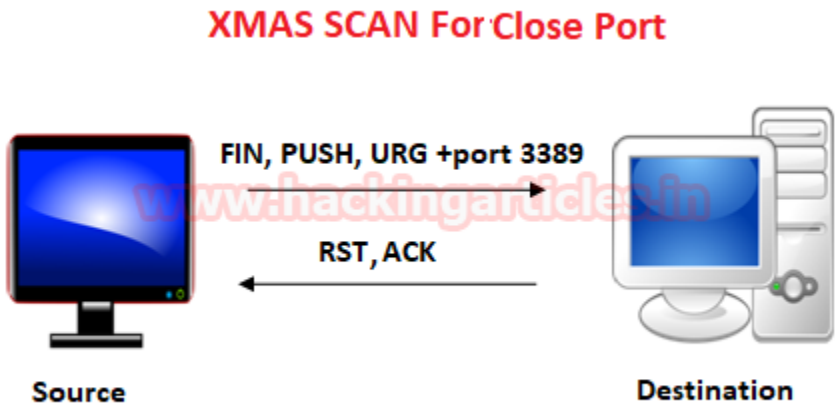
```
root@kali:~# nmap -sX -p 22 192.168.1.102

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 a source sends FIN, PUSH, and URG packets to a specific port and if the port is closed, the destination will send RST and ACK packets to the source.



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

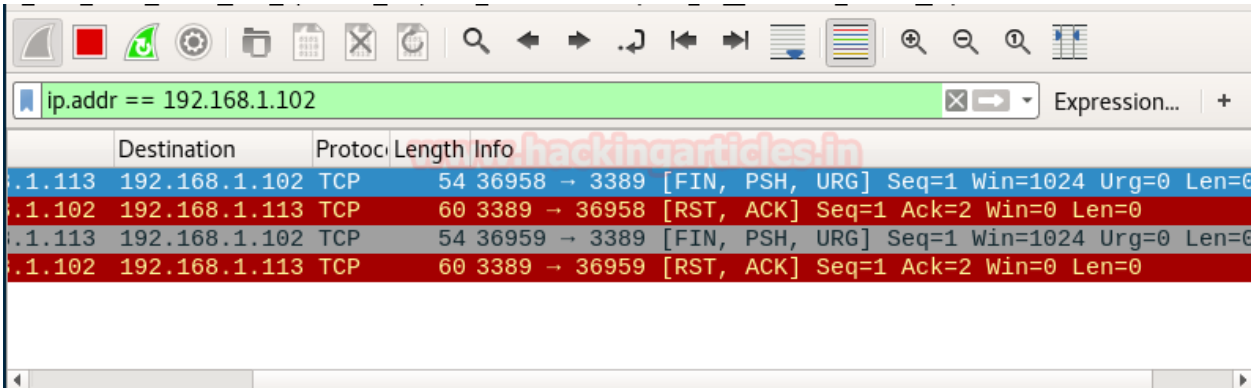
nmap -sX -p 3389 192.168.1.102

From the given image, you can observe 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



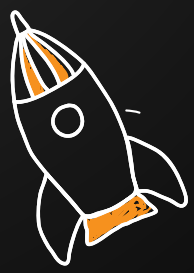
iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

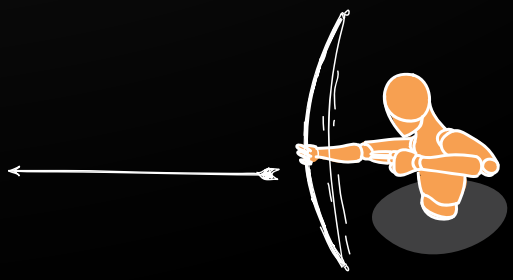
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



ADVANCED

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

EXPERT



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