NMAP

Network Discovering and Security Scanning

BEU CYBER TEAM

Osama Ahmed

Hey everyone! Today I will be talking about what Nmap is, how to get started with Nmap, some advanced scanning techniques, tips and tricks, and a deeper understanding of Nmap. Whether you're a beginner or an expert, I hope that you will find this informative and engaging. So, grab a cup of coffee, sit back, and let's dive into Nmap together!

https://www.linkedin.com/in/sharkv
https://www.linkedin.com/company/beu-cyber

Table of Contents

Introduction

Nmap was created by Gordon Lyon (Fyodor) and was released in 1997, and through the power of open-source development, Nmap grew into the world's most popular network security scanner. It was even featured in nine movies!

Nmap ("Network Mapper") is a free, open-source, powerful network exploration, and auditing tool. It has been widely used by network administrators and security professionals. It allows users to scan networks and hosts, services they are offering and operating systems they are running on. It can also perform various tasks such as detecting security vulnerabilities, version detection, fingerprinting of operating systems, determining open ports, detecting firewalls, and tracking network inventory.

Nmap supports a wide range of platforms including Windows, Linux, macOS and BSD with both console and graphical interfaces. It also integrates with other tools and technologies such as scripting, VPNs, and database systems.

This makes Nmap an essential tool for network administrators and security professionals as it provides a comprehensive view of network infrastructure and helps to ensure the security of sensitive data.

While Nmap is extremely powerful, it is also complex. More than 100 command-line options add expressiveness for network gurus.

The Phases of a Nmap Scan

Let's look at what happens when a Nmap scan runs. Scans proceed in phases, with each phase finishing before the next one begins.

1. **Script pre-scanning**

   The Nmap Scripting Engine uses a collection of special-purpose scripts to gain more information about remote systems.

   It is not executed unless you request it with options such as `--script` or `-sC`, and the pre-scanning phase only happens when scripts that need it are selected.

2. **Target enumeration**

   In this phase, Nmap researches the host specifiers provided by the user, which can be a combination of host DNS names, IP addresses, CIDR network notations, and more. This phase is essential so that it can't be skipped.

3. **Host discovery**

   Network scans usually begin by discovering which targets on the network are online and thus worth deeper investigation.

   This process is also called ***ping scanning***.

4. **Reverse-DNS lookup**

   Once Nmap has determined which hosts to scan, it looks up the reverse-DNS names of all hosts found online by the ping scan.

5. **Port scanning**

   This is the core operation of Nmap. Probes are sent, and the responses to those probes are used to classify remote ports into states such as open, closed, or filtered.

6. Version detection

If any ports are found to be open, Nmap may be able to determine what server software is running on the remote system.

7. OS detection

If requested with -O option, Nmap proceeds to OS detection. Different operating systems implement network standards in subtly different ways.

8. Traceroute

Nmap contains an optimized traceroute implementation, enabled by the --traceroute options. It can find the network routes to many hosts in parallel.

9. Scripts scanning

Nmap Scripting Engine is powered by the Lua programming language and a standard library designed for network information gathering.

Scripts running during this phase generally run once for each target host and port number that they interact with

10. Output

Finally, Nmap collects all information it has gathered and writes it to the screen or to a file.

Nmap can write output in several formats.

. . .

Before we start our first Nmap application, let's talk about some legal issues. Although Nmap helps you protect your network from invaders, it can be used improperly.

Nmap in rare cases can get you fired, expelled, banned, or jailed So, reduce your risk by reading this legal guide.

Legal Guide

You must ensure that you have permission to scan. There are many good and bad reasons for doing this sort of network exploration. But the best approach is to get permission first. So whenever possible, obtain written authorization before scanning a network. If you try to test a company's security, then report them with the flaws afterward, be prepared for some negative answers and legal issues.

Target you scan as tightly as possible, scanning enough networks or executing very intrusive scans increases the probability of generating complaints.

You always have too much to lose when someone in power decides you are a malicious cracker, even though your intentions of scanning are not bad.

Remember that there is always a trade-off, even if you use options like IP spoofing or decoy scanning through a chain of multiple open proxies, you can always be tracked and there would be suspicion of your intentions.

. . .

As I mentioned before, you can install Nmap on your Windows, MacOS, or on your Linux machine.

Make sure that you have it installed, then you are ready to go.

Target Enumeration

There are different techniques that you can use to scan, but before I talk about each in detail and put it into use against alive targets, you need to specify the targets you want to scan. You can provide a list, a range, or a subnet to scan.

Examples of target specification:

- **List**: `Your_IP example.com` will scan 2 IP addresses.
- **Range**: `14.9.11.11-20` will scan 10 IP addresses.
- **Subnet**: `14.9.11.0/24` will scan 254 IP addresses.

You can also provide a file as input with `-iL` option for the list of targets. For example, `nmap -iL List_Of_Targets`.

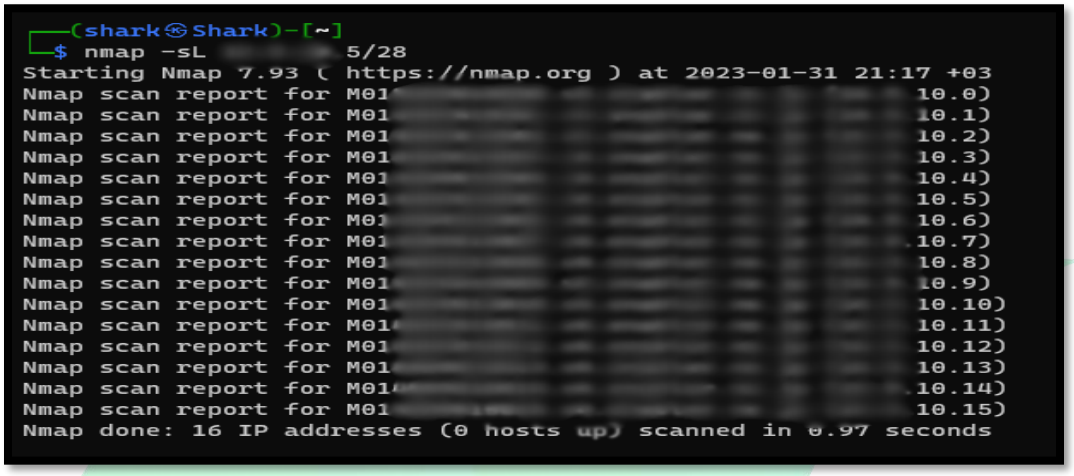
Let's start with checking the list of hosts that Nmap will scan by using `nmap -sL Targets` and this option will give you a detailed list of the host that Nmap will scan without performing scanning then; However, Nmap will attempt a reverse-DNS resolution on all the targets to obtain their names. Names might reveal a lot of information. If you don't prefer Nmap to perform the reverse-DNS resolution, you can use `-n` option.

For Internet-wide surveys and other research, you may want to choose targets at random. This is done with `-iR` option, which takes as an argument the number of IPs to generate.

Sometimes it can be so critical that you won't take a risk scanning a specific machine, you can exclude hosts or entire networks with the `--exclude` option then the IP or IP range that you want to exclude, also you can exclude hosts with an input file, this done with `--excludefile` option with the input filename.

This feature of Nmap helps you for better understanding of the target network, all without raising alarm bells. You must ensure that you are scanning the right network, otherwise scanning the wrong network would be a disaster.

Sometimes the best way to understand things is to see them in action, so here is an example of listing our target hosts:



...

Host Discovery

By default, Nmap will include a ping scanning stage prior to more intrusive probes such as port scan, OS detection, or version detection.

You can control your scan to save substantial time and bandwidth compared to performing in full scan against every single IP address.

List scan can be done with `-sL` option. This option is a good sanity check to ensure that you have proper IP addresses for your targets. IF the hosts display domain names you do not recognize, it is worth investigating further to prevent scanning the wrong company's network.

Disable Port Scan can be done with `-sn` option. This option makes Nmap do host discovery, then print out the available hosts that responded to the scan. Even though no port scanning is done. This scan helps to easily count available machines on a network or monitor server availability.

Disable Ping scan can be done with `-pn` option. This option makes Nmap skips discovery stage altogether. One of the most common reasons to use

this option is intrusive vulnerability assessments, or if you already know the online hosts, it also might be helpful while scanning large networks as it tells Nmap not to send ICMP echo requests.

No ping option can be done with `-PN`. This option tells Nmap to assume that the host is online. This option saves time as the ping scan is skipped, but it also means that Nmap might return incorrect results if the target host is actually down.

You can control the length of the packet by using `--data-length` then a value of the packet bytes. This option works with TCP, UDP, and ICMP ping scan types. This helps make the scan less conspicuous and more like the packets generated by the ubiquitous ping diagnostics program.

Setting the outgoing TTL value can be useful as a safety precaution to ensure a scan does not propagate beyond the local network. It can also be used to simulate a native ping program even more convincingly. This option is done with `--ttl` followed by a value.

Verbose mode causes Nmap not only to print live hosts, but it also prints down hosts, as well as extra information about active ones, you can use verbose mode by using `-v` option.

By default, Nmap uses a ping scan to find live hosts, then proceeds to scan only the live hosts. If you want to discover online hosts without post-scanning the live services, you can use `-sn` option.

Now let's gain more understanding of the different techniques you can use, you can leverage the protocols and use them to discover the live hosts, let's start by the order from the bottom of TCP/IP layers to the top.

- ARP from Link Layer
- ICMP from Network Layer
- TCP from Transport Layer
- UDP from Transport Layer

...

HOST DISCOVERY USING ARP

Isn't it time wasting scanning offline hosts or IP address not in use? There are several ways to discover online hosts.

You can use ARP scan technique only if the target is on the same subnet as yours, and you need to know the MAC address of any system before you can communicate with it. To get the MAC address, the OS sends an ARP query. All hosts reply to ARP queries, are live.

If you want Nmap to perform an ARP scan and skip the port-scanning, you can use `nmap -PR -sn Targets`, where `-PR` indicates that you only want and ARP scan, and `-sn` indicates that you want to skip the port-scanning.

Here is an example using ARP scan:

```
┌──(shark㉿Shark)-[~]
└─$ nmap -PR -sn scanme.nmap.org
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-02 17:38 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.21s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Nmap done: 1 IP address (1 host up) scanned in 13.22 seconds
```

...

HOST DISCOVERY USING ICMP

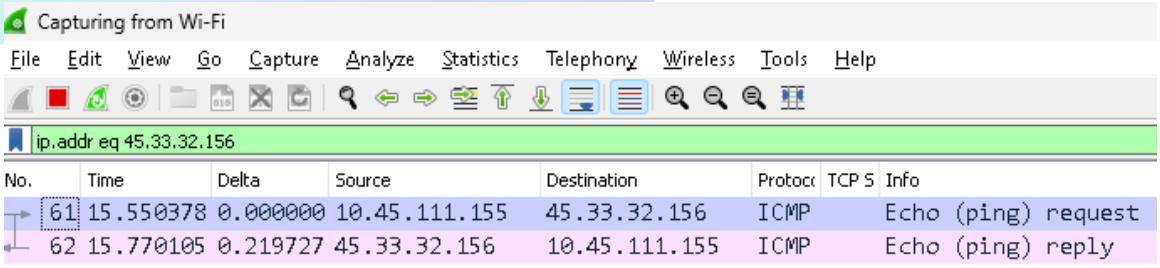
Using ICMP Type 8/Echo requests you can ping every IP address on a target network and see who response with ICMP Type 0/reply. It is the

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -PE -sn scanme.nmap.org
[sudo] password for shark:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-01 05:44 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.22s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Nmap done: 1 IP address (1 host up) scanned in 13.43 seconds
```

most straightforward approach but many firewalls block ICMP echo requests. You can use ICMP echo request to discover live hosts by using `-PE` and the live hosts will reply with ICMP echo reply.

Here is an ICMP ping scan and Nmap found 1 live host then I skipped the port scanning using `-sn` option. If the target is on the same subnet as our system, expect to see the MAC address of the target.

Let's look at the ICMP echo ping scan packet traffic.



If ICMP echo requests tend to be blocked, you might consider ICMP Timestamp or ICMP Address Mask requests to tell if a system is online. Adding `-PP` or `-PM` will send ICMP Timestamp and ICMP Address Mask requests to the target.

It is essential to learn multiple approaches to achieve the same result. If one type of packet is being blocked, you can always choose another to discover the target network and services.

HOST DISCOVERY USING TCP AND UDP

Using TCP, you can send a packet with flags set to help us discover live hosts. Here I don't focus on port scanning, just to see which hosts are up.

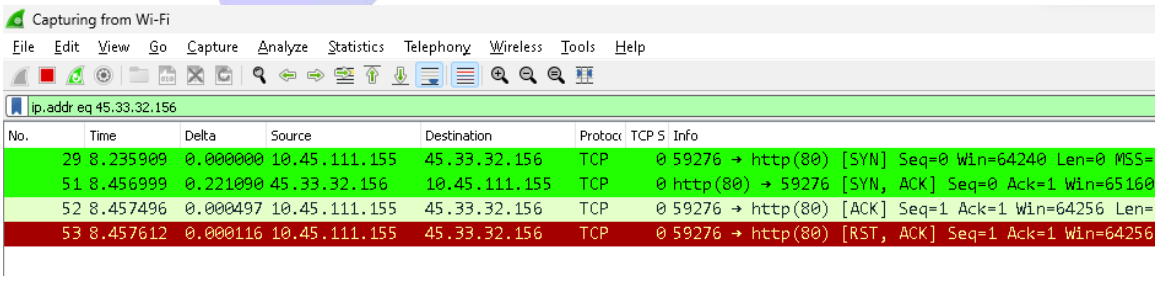
TCP connection ping

Using TCP 3-way handshake connection, Nmap can scan for live hosts on target's network, this scan done with `-PT` option, which tells Nmap to start a TCP connection by sending TCP packet with SYN flag set, live hosts should respond with TCP packet with SYN/ACK flags set, then the connection is done with Nmap sending TCP packet with ACK flag set, afterwards the connection is ended by Nmap sending TCP packet with RST/ACK flags set.

An example of TCP connection ping scan

```
┌──(shark㉿Shark)-[~]
└─$ nmap -PT -sn scanme.nmap.org
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-03 16:10 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.22s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Nmap done: 1 IP address (1 host up) scanned in 13.23 seconds

┌──(shark㉿Shark)-[~]
└─$
```

Let's take a look at TCP connection ping scan network traffic.



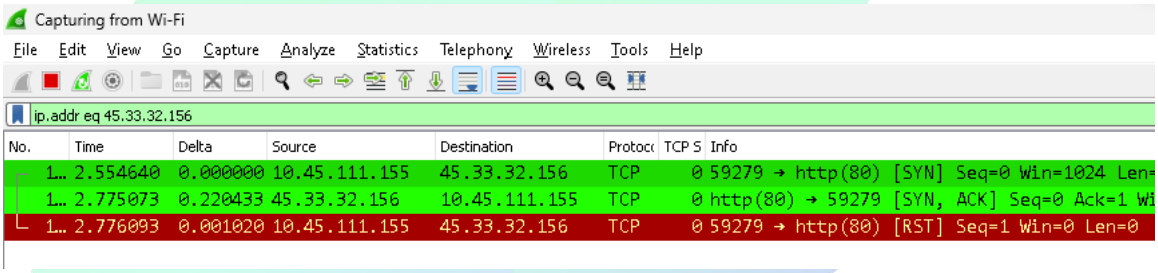
TCP SYN ping

To send TCP packet with SYN flag set, you will use `-PS` followed by the port number, range, or a list of them. For example `-PS80` `-PS21-25` `-PS23, 53, 80`. Then the live hosts should respond with SYN/ACK, then

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -PS -sn scanme.nmap.org
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-01 06:23 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.21s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Nmap done: 1 IP address (1 host up) scanned in 13.39 seconds
```

Nmap ends the connection with sending a TCP packet with RST flag set.

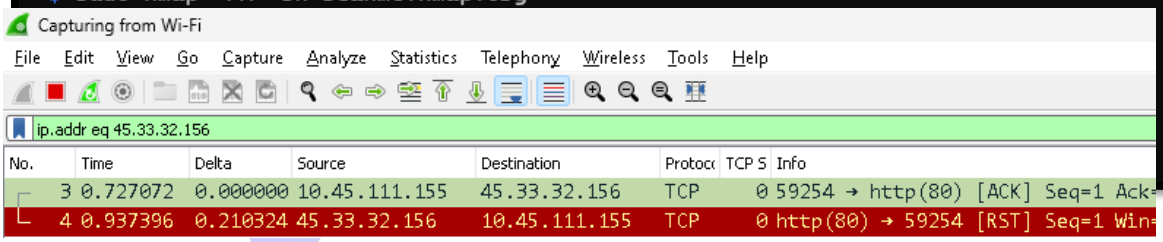
Let's take a look at the TCP SYN ping scan network traffic.



TCP ACK ping

To send TCP packet with ACK flag set, we will use `-PA` and the same as SYN flag syntax, followed by ports. Then the live hosts should respond with RST flag.

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -PA -sn scanme.nmap.org
```



Let's take a look at the TCP ACK ping scan network traffic.

UDP ping

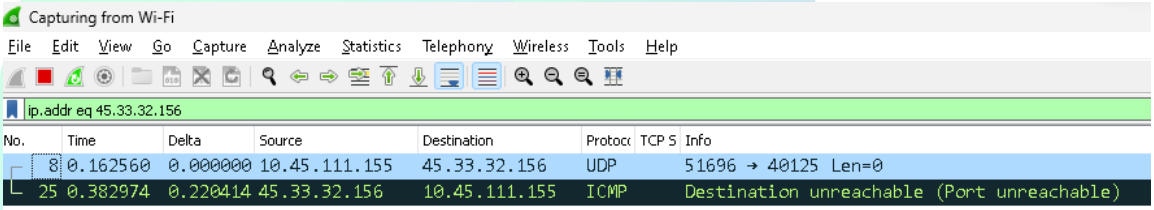
Let's talk about the usage of UDP to discover live hosts. Sending a UDP packet to an open port is not expected to lead to any reply.

But when Nmap sends UDP packet to a closed UDP port, you expect to get an ICMP port unreachable packet; this indicates that the target system is up and available, this scan is done by using `-PU` option.

An example of UDP ping scan

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -PU -sn scanme.nmap.org
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-01 06:47 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.22s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Nmap done: 1 IP address (1 host up) scanned in 13.60 seconds
```

Take a look at the UDP scan network traffic.



IP PROTOCOL PING

The newest host discovery option is IP protocol ping, which sends IP packets with the specified protocol number set in their IP header. If no protocols are specified, the default is to send multiple packets to ICMP (protocol 1), IGMP (protocol 2), and IP-in-IP (protocol 4).

This host discovery method looks for either responses using the same protocol as a probe, or ICMP protocol unreachable messages which signify that the destination host does not support the given protocol.

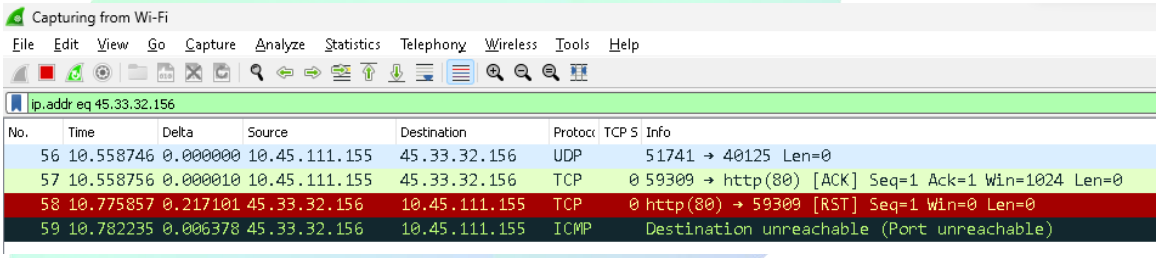
This scan is done with `-PO` option followed by protocol list.

An example of IP Protocol Ping:

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -PO6,17 -sn scanme.nmap.org
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-02 15:52 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.22s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Nmap done: 1 IP address (1 host up) scanned in 13.40 seconds

┌──(shark㉿Shark)-[~]
└─$
```

In this example I used protocol 6, which is TCP, and protocol 17, which is UDP to discover live hosts.

Let's take a look at IP protocol ping scan network traffic.



To perform a fast and efficient scan to check if a target host is up or not you can use `-sP` option. This option won't provide any information about the services or open ports on the target host.

. . .

Reverse-DNS Lookup

Nmap uses reverse-DNS live hosts by default. The host names can be helpful and can reveal a lot. However, if you don't prefer to send DNS queries, we use `-n` to skip the reverse-DNS.

Also, Nmap looks up online hosts; however, you can use `-R` to query DNS server even for the hosts which are offline.

An example of revere-DNS `-R` option

```
┌──(shark㉿Shark)-[~]
└─$ nmap -sn -R 45.33.32.156
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-03 16:47 +03
Nmap scan report for 45.33.32.156
Host is up (0.21s latency).
Nmap done: 1 IP address (1 host up) scanned in 13.22 seconds

┌──(shark㉿Shark)-[~]
└─$
```

Port scanning

Port scanning is the act of remotely testing numerous ports to determine what state they are in. The most interesting state is usually open, which means that an application is listening and accepting connections on the port.

Nmap offers a lot of port scanning methods, only one scan may be used at a time, except UDP scan, may be combined with any one of the TCP scan types.

Port Scanning Methods

- **TCP SYN Scan**

  The most popular scan type because it is the fastest way to scan ports of the most popular protocol TCP.

- **TCP ACK Scan**

  ACK scan helps to understand whether firewall rules are stateful or not.

- **TCP Connect Scan**

  Connect scan uses the system call of the same name to scan machines.

- **UDP Scan**

  UDP ports offer plenty of security holes too!

- **TCL FIN, NULL, and Xmas Scans**

  These special purpose scan types are adept at sneaking past firewalls to explore the systems behind them.

- **TCP Window Scan**

  Window scan is like ACK, except that it is able to detect open versus closed ports against certain machines.

- **TCP Maimon Scan**

  This obscure firewall-evading scan type is similar to FIN scan but includes the ACK flag as well. This allows it to get by more packet filtering firewalls.

- **TCP Idle Scan**

  This scan can exploit trusted IP address relationships. Also, it is slow and complex.

TCP AND UDP PORTS

A TCP port or UDP port is used to identify a network service running on that host, it is provided by a server, and it adheres to a specific network protocol, and there only can be one service that can listen on any TCP or UDP port on the same IP address.

Scanning a host's ports is going to tell us which ports have service listening on it, Nmap shows that as "open" or "closed" port, but we need to consider the impact of firewalls, where firewalls can block our sent packets, therefore we won't be able to determine whether port is open or closed. Here are the six states that Nmap considers:

1. Open: indicates that a service is listening on the specified port.
2. Closed: indicates that no service is listening on the specified port and the port is reachable and is not blocked by a firewall or other security applications.

3. Filtered: means that Nmap can't determine if the port is open or closed because the port is not reachable, and this usually happens due to a firewall blocking the sent packets.
4. Unfiltered: means that Nmap can't determine if the port is open or closed, although the port is reachable.
5. Open|Filtered: this indicates that Nmap can't determine whether a port is open or filtered.
6. Closed|Filtered: this indicates that Nmap can't determine whether a port is closed or unfiltered.

PORT SCANNING USING TCP

First let's talk about TCP flags used at port scanning:

1. URG : Urgent flag indicates that the urgent pointer filed is significant. The urgent pointer indicates that the incoming data is urgent, and that a TCP segment with the URG flag set is processed immediately without consideration of having to wait on previously sent TCP segments.
2. ACK : Acknowledgement flag indicates that the acknowledgement number is significant. It is used to acknowledge the receipt of a TCP segment.
3. PSH : Push flag asking TCP to pass the data to the application promptly.
4. RST : Reset flag is used to reset the connection. Another device, such as a firewall, might send it to tear a TCP connection. This flag is also used when data is sent to a host and there is no service on the receiving end to answer.
5. SYN : Synchronize flag is used to initiate a TCP 3-way handshake and synchronize sequence numbers with the other host. The sequence number should be set randomly during TCP connection establishment.
6. FIN : The sender has no more data to send.

TCP CONNECT SCAN

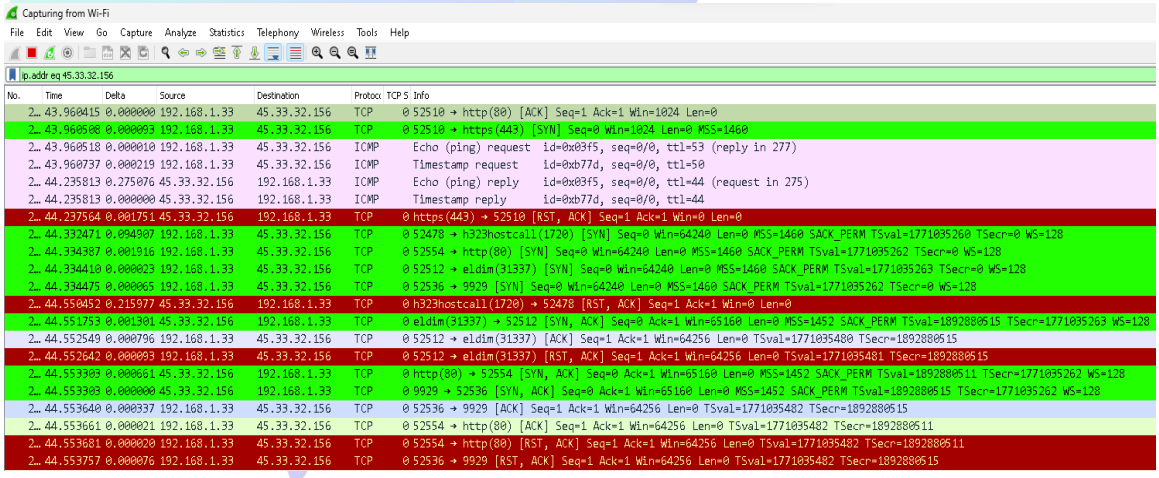
TCP Connect Scan work by completing the TCP 3-way handshake, but here Nmap sends RST/ACK as a fourth packet to tear the connection down after the TCP 3-way handshake.

We can perform this scan using `-sT` , and by default, Nmap is attempting to connect to the 1000 most common ports. A closed port responds to TCP packet with RST/ACK to indicate that it is not open.

An example of TCP Connect Scan

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sT scanme.nmap.org
[sudo] password for shark:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-01 14:24 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.23s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 854 closed tcp ports (conn-refused), 142 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
1720/tcp  open  h323q931
9929/tcp  open  nping-echo
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 34.47 seconds
```

The network traffic using TCP Connect Scan



You can also use `-F` to enable fast mode and decrease the number of scanned ports from 100 to most common ports as so

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sT -F scanme.nmap.org
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-01 14:31 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.31s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 64 closed tcp ports (conn-refused), 34 filtered tcp ports (no-response)
PORT     STATE SERVICE
80/tcp   open  http
1720/tcp open  h323q931

Nmap done: 1 IP address (1 host up) scanned in 21.38 seconds

┌──(shark㉿Shark)-[~]
└─$
```

You can perform the port scan using `-r` to consecutive order port scanning instead of random order.

TCP SYN SCAN

TCP SYN Scan doesn't complete the TCP 3-way handshake; instead, it tears down the connection once it receives a response from the server. Not establishing TCP connection decreases the chances of the scan being logged.

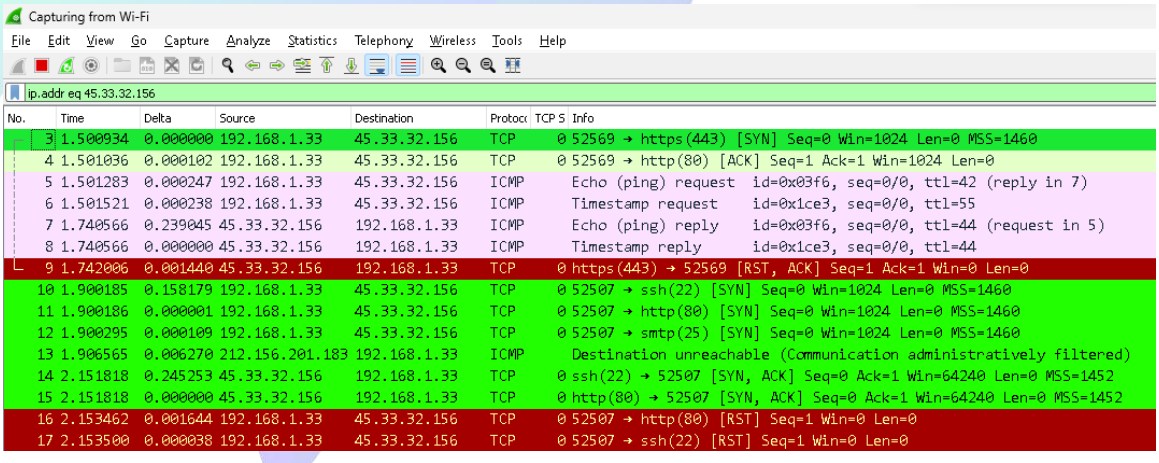
To perform a TCP SYN Scan you will use `-sS` .You will expect to receive SYN/ACK from open ports, and then Nmap sends TCP RST packet.

An example of TCP SYN Scan, but this time we only scan 3 TCP ports 22, 25, 80:

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sS -p22,25,80 scanme.nmap.org
[sudo] password for shark:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-01 14:47 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.32s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f

PORT   STATE    SERVICE
22/tcp filtered ssh
25/tcp filtered smtp
80/tcp open     http

Nmap done: 1 IP address (1 host up) scanned in 17.48 seconds
```

And Nmap got the TCP 22, 25 port with filtered state which means that our TCP SYN packet has been blocked by a firewall or a security application and TCP 80 port with open state.

The network traffic using TCP SYN Scan



UDP PORT SCAN

The open ports don't respond to our UDP scan packets because UDP is a connectionless protocol, and it doesn't require handshake for connection establishment. In this case closed ports will respond with ICMP unreachable error.

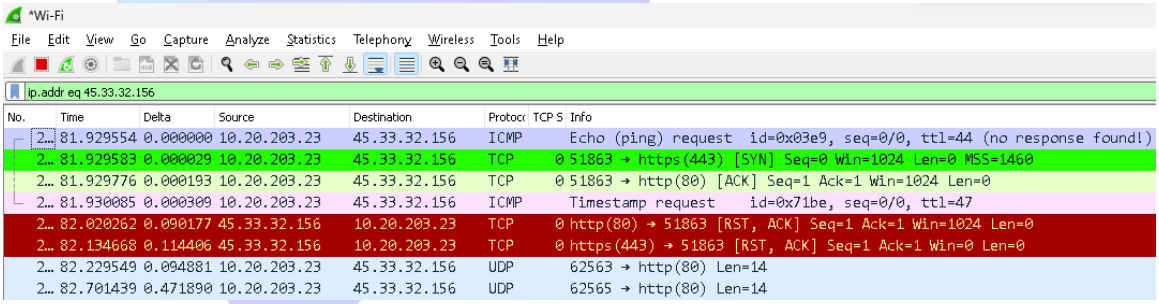
Use `-sU` to perform UDP port scan.

An example of UDP port scan and we are going to scan the most common 100 ports:

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sU -F scanme.nmap.org
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-01 15:10 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.22s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 60 closed udp ports (port-unreach), 39 open|filtered udp ports (no-response)
PORT    STATE SERVICE
123/udp open  ntp

Nmap done: 1 IP address (1 host up) scanned in 110.05 seconds
```

As shown, Nmap got 60 closed UDP ports, 39 `open|filtered` UDP ports, and the UDP port 123 open and have service listening on it.

The network traffic using UPD Port Scan

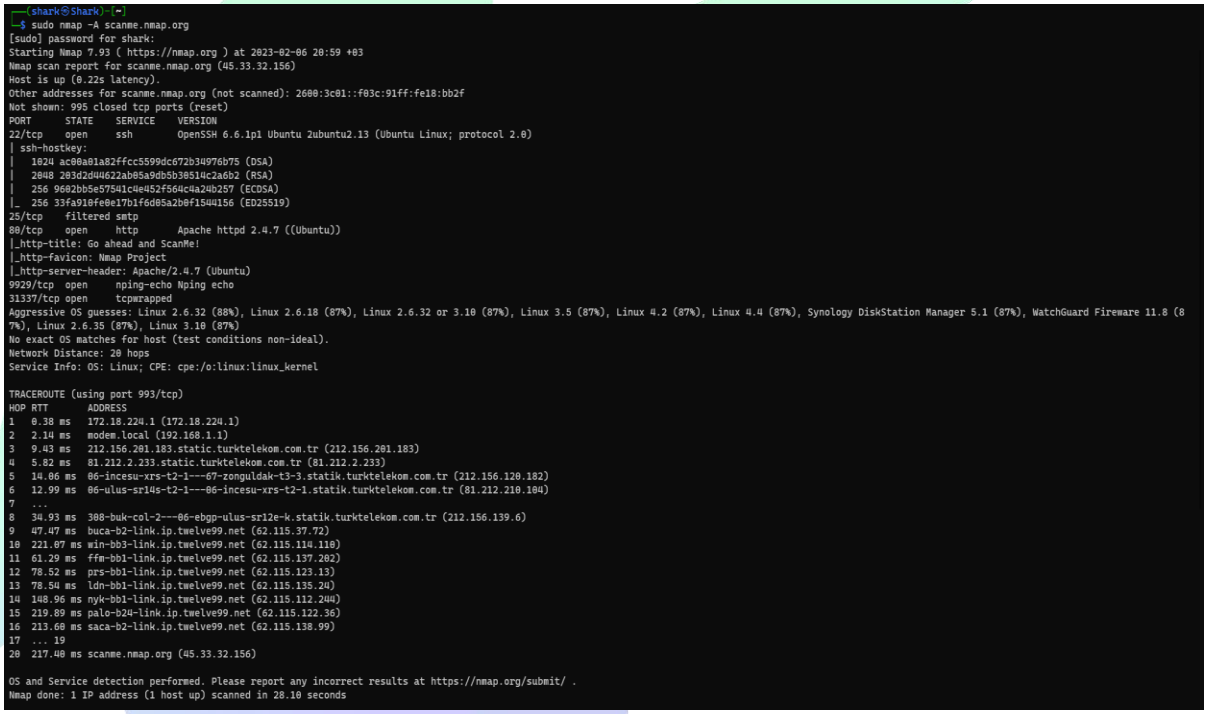


AGGRESSIVE SCAN

Nmap has an aggressive scan mode that enables OS detection, version detection, script scanning, and traceroute. This type of scan provides far better information than regular scans.

You can use `-A` to perform an aggressive scan.

An example of an aggressive scan



```
┌──(shark㉿shark)-[~]
└─$ sudo nmap -A scanme.nmap.org
[sudo] password for shark:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-06 20:59 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.22s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 995 closed tcp ports (reset)
PORT      STATE    SERVICE    VERSION
22/tcp    open     ssh        OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 ac00a81a82ffcc5599dc672b34976b75 (DSA)
|   2048 203d2d44622ab05a9db5b30514c2a6b2 (RSA)
|   256 9602bb5e57541c4e452f564c4a24b257 (ECDSA)
|_  256 33fa910fe0e17b1f6d05a2b0f1544156 (ED25519)
25/tcp    filtered smtp
80/tcp    open     http       Apache httpd 2.4.7 ((Ubuntu))
|_http-title: Go ahead and ScanMe!
|_http-favicon: Nmap Project
|_http-server-header: Apache/2.4.7 (Ubuntu)
9929/tcp  open     nping-echo Nping echo
31337/tcp open     tcpwrapped
Aggressive OS guesses: Linux 2.6.32 (88%), Linux 2.6.18 (87%), Linux 2.6.32 or 3.10 (87%), Linux 3.5 (87%), Linux 4.2 (87%), Linux 4.4 (87%), Synology DiskStation Manager 5.1 (87%), WatchGuard Fireware 11.8 (8
7%), Linux 2.6.35 (87%), Linux 3.10 (87%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 20 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 993/tcp)
HOP RTT       ADDRESS
1   0.38 ms   172.18.224.1 (172.18.224.1)
2   2.14 ms   modem.local (192.168.1.1)
3   9.43 ms   212.156.201.183.static.turktelekom.com.tr (212.156.201.183)
4   5.82 ms   81.212.2.233.static.turktelekom.com.tr (81.212.2.233)
5   14.06 ms  86-incesu-xrs-t2-1---67-zonguldak-t3-3.statik.turktelekom.com.tr (212.156.120.182)
6   12.99 ms  86-ulus-sr14s-t2-1---86-incesu-xrs-t2-1.statik.turktelekom.com.tr (81.212.210.184)
7   ...
8   34.93 ms  308-buk-col-2---86-ebgp-ulus-sr12e-k.statik.turktelekom.com.tr (212.156.139.6)
9   47.47 ms  buca-b2-link.ip.twelve99.net (62.115.37.72)
10  221.07 ms win-bb3-link.ip.twelve99.net (62.115.114.118)
11  61.29 ms  ffm-bb1-link.ip.twelve99.net (62.115.137.202)
12  78.52 ms  prs-bb1-link.ip.twelve99.net (62.115.123.13)
13  78.54 ms  ldn-bb1-link.ip.twelve99.net (62.115.135.24)
14  148.96 ms nyk-bb1-link.ip.twelve99.net (62.115.112.244)
15  219.89 ms palo-b24-link.ip.twelve99.net (62.115.122.36)
16  213.60 ms saca-b2-link.ip.twelve99.net (62.115.138.99)
17  ... 19
20  217.40 ms scanme.nmap.org (45.33.32.156)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 28.10 seconds
```

• • •

BAD CHECKSUMS SCAN

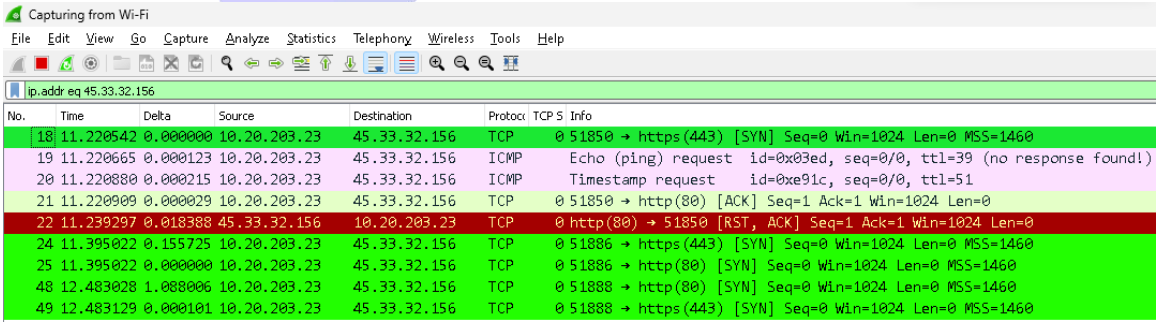
Bad checksum is sending the packet with bad or bogus TCP/UDP checksums to the intended target to avoid certain firewall rulesets.

Use --badsum option to perform bad checksums scan.

An example of using Bad Checksum Scan with verbose option performed.

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap --badsum scanme.nmap.org -v
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-09 20:28 +03
Initiating Ping Scan at 20:28
Scanning scanme.nmap.org (45.33.32.156) [4 ports]
Completed Ping Scan at 20:28, 0.13s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 20:28
Completed Parallel DNS resolution of 1 host. at 20:28, 0.00s elapsed
Initiating SYN Stealth Scan at 20:28
Scanning scanme.nmap.org (45.33.32.156) [1000 ports]
Completed SYN Stealth Scan at 20:28, 4.39s elapsed (1000 total ports)
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.020s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
All 1000 scanned ports on scanme.nmap.org (45.33.32.156) are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 4.78 seconds
           Raw packets sent: 2007 (88.272KB) | Rcvd: 4 (160B)

┌──(shark㉿Shark)-[~]
└─$
```

The network traffic using Bad Checksum Scan



. . .

Now I will show you some cool options to use while performing port scanning, first we can use `-p-` to scan all 65,535 ports, as I mentioned before `-F` for the most common 100 ports, well, let's customize that by using `--top-ports` and the number we prefer to scan for example `--top-ports 77` and this will scan the most common 77 ports, you can control the timing of our scan by using `-T` and then a number from `0` to `5` , while `-T0` is the slowest and `-T5` is the fastest scan, also, you can control the packet rate using `--min-rate` or `--max-rate` then the number of packet we want to send per second.

For example, `--max-rate 10` will ensure that the scanner is not sending more than 10 packets per second, and finally you can control probing parallelization using `--min-parallelism` or `--max-parallelism` then the number of probes. For example `--min-parallelism 200` pushes Nmap to maintain at least 200 probes in parallel, where these 200 probes are related to host discovery and open ports.

The `--packet-trace` option is used to display the low-level details of the packets sent and received during a scan. This option is primarily used for debugging and troubleshooting purposes, as it provides a detailed view of the packets and their contents, including IP headers, TCP flags, and payload data.

Using the `--stats-every` option followed by time value is going to set periodic timing stats. For example, using `--stats-every 2s` is going to print the scan's status every two seconds.

. . .

Advanced Port Scans

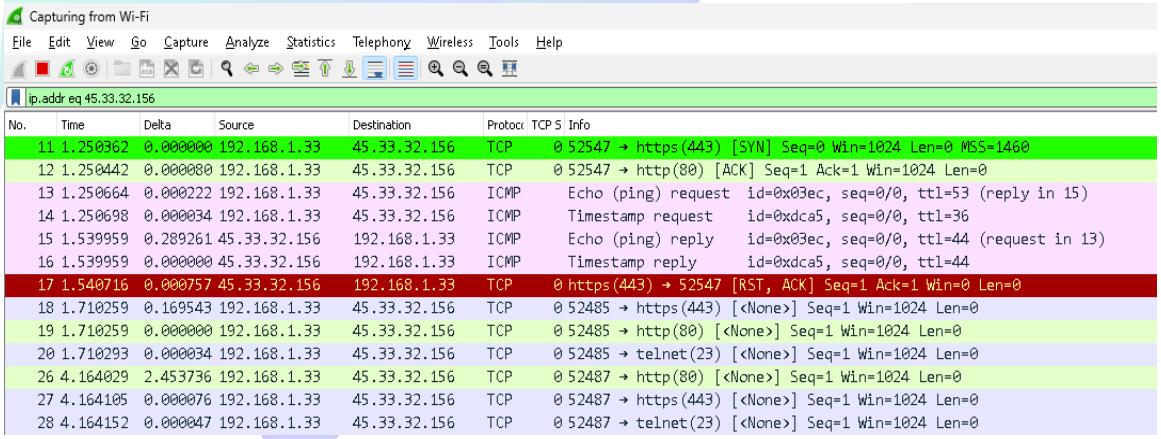
TCP NULL SCAN

The null scan does not set any flag, all six flag bits are set to zero. You can choose this scan using `-sN` option. A TCP packet with no flags set won't trigger any response when it reaches an open port. However, a closed port should respond with an RST packet.

An example of TCP NULL Scan

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sN -p23,80,443 45.33.32.156
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-05 22:32 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.29s latency).

PORT    STATE         SERVICE
23/tcp  open|filtered telnet
80/tcp  open|filtered http
443/tcp open|filtered https

Nmap done: 1 IP address (1 host up) scanned in 4.59 seconds
```

The packet traffic using TCP NULL Scan



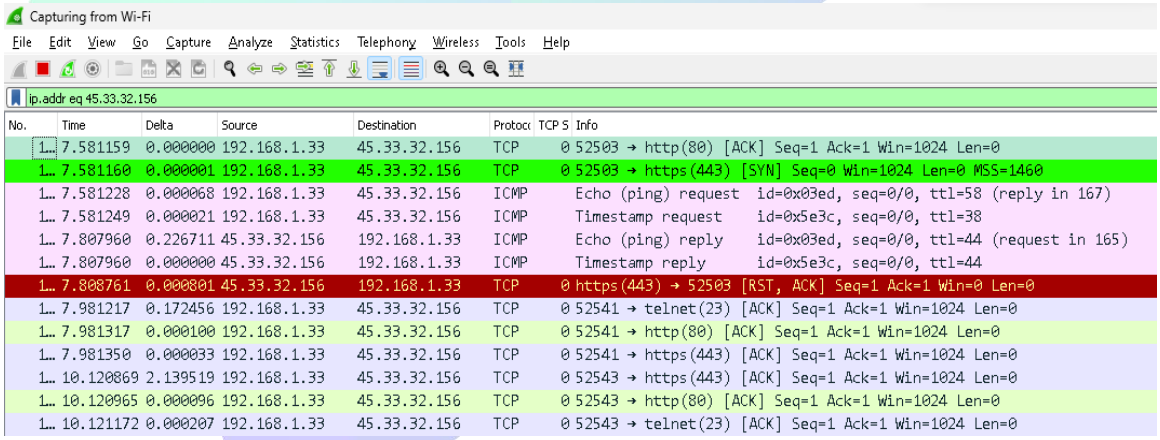
TCP ACK SCAN

This scan sends a TCP packet with ACK flag set . To choose this scan use `-sA` option. All target ports respond with RST TCP packet, this scan is helpful if there is a firewall in front of the target.

An example of TCP ACK Scan

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sA -p23,80,443 45.33.32.156
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-05 22:35 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.23s latency).

PORT    STATE    SERVICE
23/tcp  filtered telnet
80/tcp  filtered http
443/tcp filtered https

Nmap done: 1 IP address (1 host up) scanned in 3.89 seconds
```

The packet traffic using TCP ACK Scan



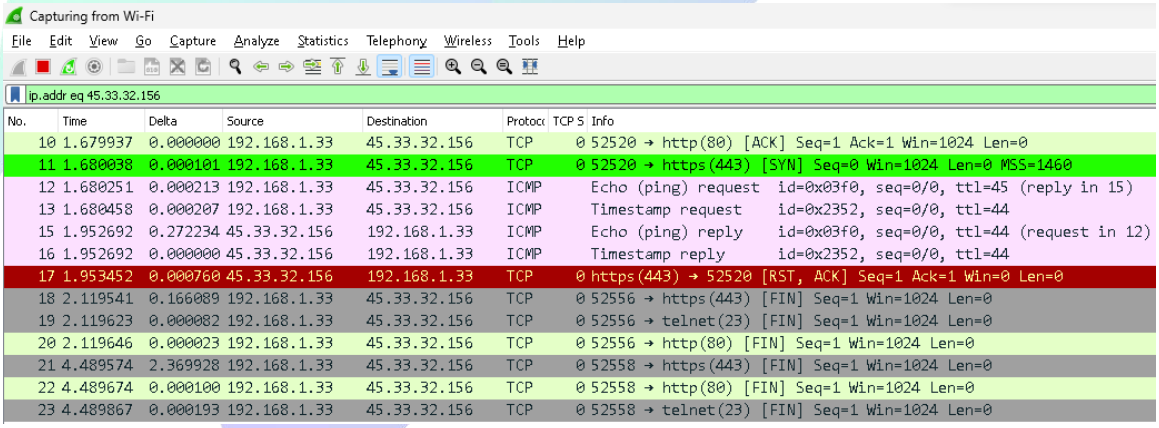
TCP FIN SCAN

The FIN scan sends a TCP packet with FIN flag set. You can choose this scan type using `-sF` option. The open ports won't respond to any packets Nmap sends.

An example of TCP FIN Scan

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sF -p23,80,443 45.33.32.156
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-05 22:38 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.27s latency).

PORT    STATE         SERVICE
23/tcp  open|filtered telnet
80/tcp  open|filtered http
443/tcp open|filtered https

Nmap done: 1 IP address (1 host up) scanned in 4.39 seconds
```

The packet traffic using TCP FIN Scan



TCP XMAS SCAN

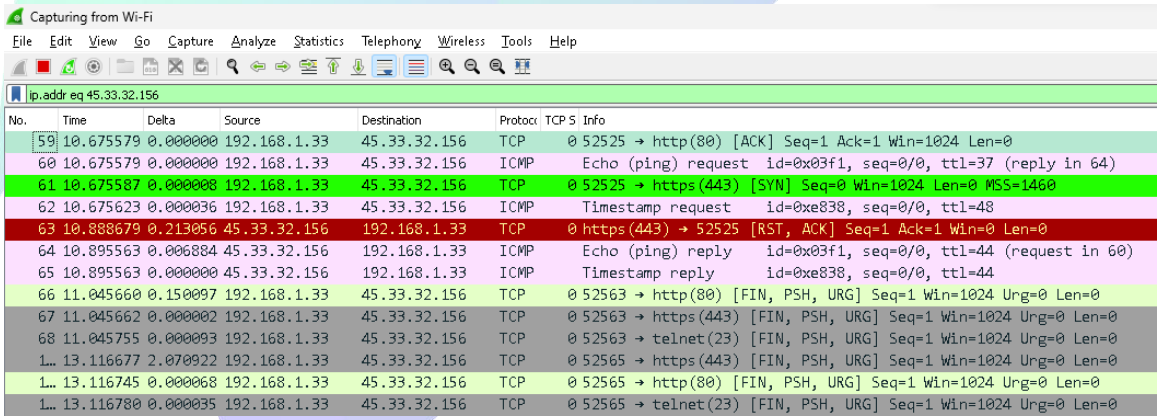
The Xmas scan gets its name after Christmas tree. An Xmas scan sets the FIN, PSH, and URG flags to TCP packets. It's done with `-sX` option

No packets are received from open ports, closed ports respond with RST TCP packet.

An example of TCP Xmas Scan

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sX -p23,80,443 45.33.32.156
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-05 22:41 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.21s latency).

PORT    STATE         SERVICE
23/tcp  open|filtered telnet
80/tcp  open|filtered http
443/tcp open|filtered https

Nmap done: 1 IP address (1 host up) scanned in 3.70 seconds
```

The packet traffic using TCP Xmas Scan



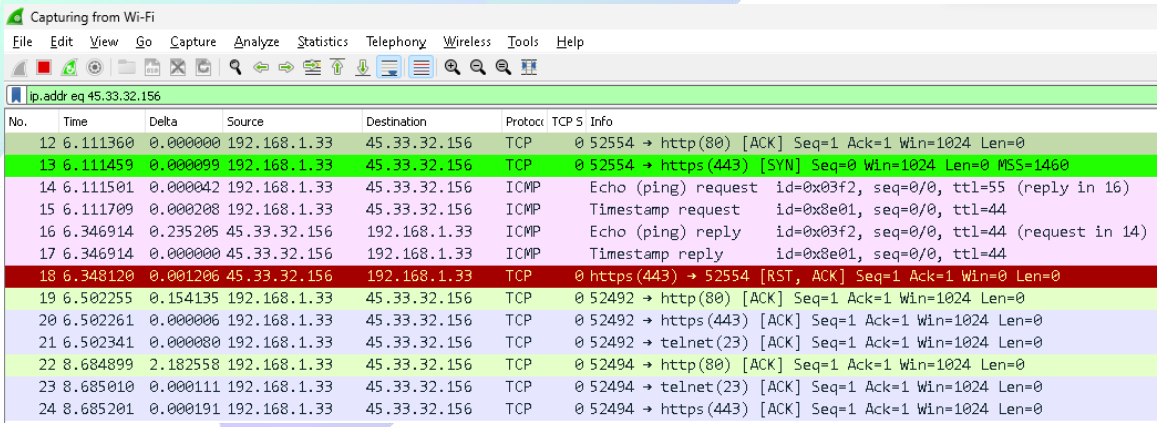
TCP WINDOW SCAN

The TCP window scan is similar to the TCP ACK scan; however, it examines the TCP window field of the RST packets returned, which can reveal that the port is open. It is done with `-sW` option.

An example of TCP Window Scan

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sW -p23,80,443 45.33.32.156
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-05 22:43 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.24s latency).

PORT    STATE    SERVICE
23/tcp  filtered telnet
80/tcp  filtered http
443/tcp filtered https

Nmap done: 1 IP address (1 host up) scanned in 3.97 seconds
```

The packet traffic using TCP Window Scan



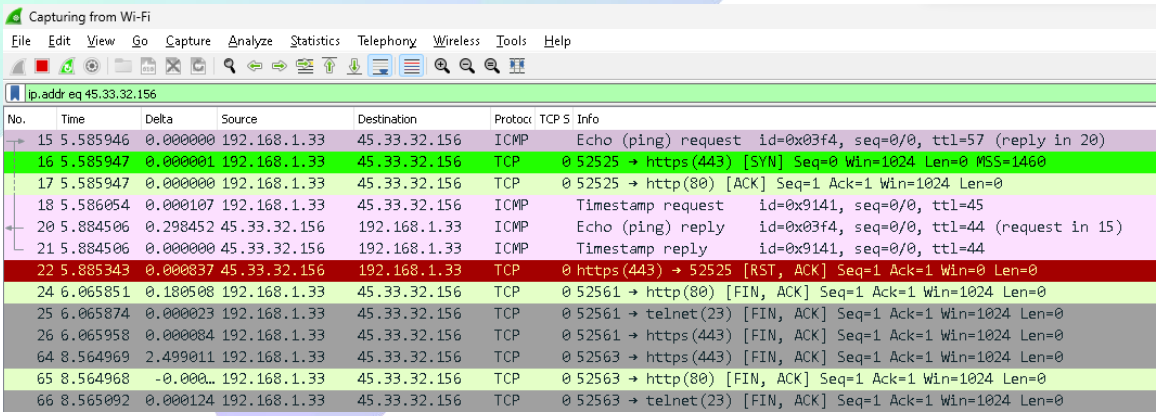
TCP MAIMON SCAN

This scan doesn't work on targets encountered in modern networks, this scan sets FIN and ACK flags to TCP packets. It's done using -sM option, most of targets respond with a RST TCP packet whether the port is open or closed, so, it is hard to determine whether the port is open or closed.

An example of TCP Maimon Scan

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sM -p23,80,443 45.33.32.156
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-05 22:47 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.30s latency).

PORT    STATE         SERVICE
23/tcp  open|filtered telnet
80/tcp  open|filtered http
443/tcp open|filtered https

Nmap done: 1 IP address (1 host up) scanned in 4.67 seconds
```

The packet traffic using TCP Maimon Scan



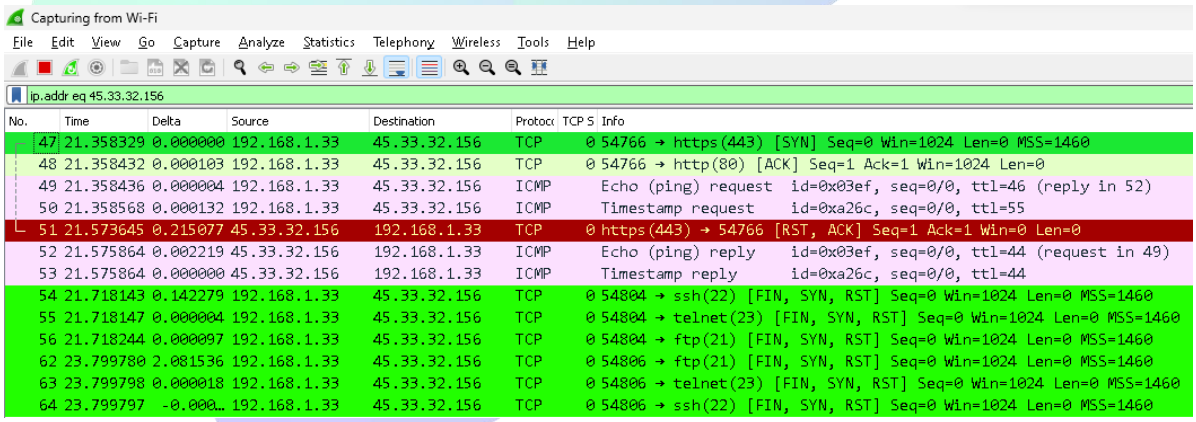
CUSTOM SCAN

You can custom a TCP packet to send using `--scanflags` option .For example, if you want to send a FIN, RST TCP packet, you can use `--scanflags FINRST` option. You need to know how ports behave with these flags to be able to determine the port's state.

An example of TCP Custom Scan

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -scanflags SYNFINRST -p21,22,23 scanme.nmap.org
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-06 21:19 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.22s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f

PORT   STATE    SERVICE
21/tcp filtered ftp
22/tcp filtered ssh
23/tcp filtered telnet

Nmap done: 1 IP address (1 host up) scanned in 3.74 seconds
```

The packet traffic using TCP Custom Scan



SPOOFING AND DECOYS

Sometimes you can scan networks with spoofed IP address and even a spoofed MAC address. You should capture the response using this scan, because you won't have any response routed from targets. It's done with `-S` followed by the spoofed IP address.

In this scan there are three steps:

1. Nmap sends a packet with a spoofed IP address to targets.
2. Targets reply to the spoofed IP address.
3. The attacker captures the reply to figure out open ports.

To specify a network interface use `-e` option

An example of spoofing an IP address to scan

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sS -Pn -S 172.29.142.47 -p 80,443 scanme.nmap.org
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-09 15:57 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.24s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f

PORT    STATE  SERVICE
80/tcp  open   http
443/tcp closed https

Nmap done: 1 IP address (1 host up) scanned in 0.48 seconds
```

You are able to use decoys to scan the target's network, the scan will be going from many IP addresses so that your IP address would be lost among them. It is done with `-D` option followed by the IP addresses, you can write `RND` to have a random IP address and `ME` instead of writing your IP address.

An example of using decoys to scan

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sS -Pn -D 10.10.10.1,10,10,10,2,ME -p 80,443 scanme.nmap.org
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-09 15:52 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.25s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f

PORT    STATE  SERVICE
80/tcp  open   http
443/tcp closed https

Nmap done: 1 IP address (1 host up) scanned in 0.49 seconds
```

. . .

PACKET FRAGMENTATION

Nmap provides the option `-f` to fragment packets. By default, the packets are being divided into 8 bytes, you can add another one to divide the packets into 16 bytes. (`-f -f` or `-ff`). You can use `--mtu` to change the default value.

This option is very helpful while there is a firewall blocking the sent packets.

An example of packet fragmentation

. . .

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sT -f -p21,22,23 scanme.nmap.org
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-06 21:35 +03
You have specified some options that require raw socket access.
These options will not be honored for TCP Connect scan.
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.22s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f

PORT   STATE  SERVICE
21/tcp closed ftp
22/tcp open   ssh
23/tcp closed telnet

Nmap done: 1 IP address (1 host up) scanned in 0.67 seconds
```

IDLE/ZOMBIE SCAN

The idle or zombie scan requires an idle system connected to the network that you can communicate with.

In this scan Nmap checks the IP identification value (IP ID) in the IP header. This scan is done using -sI option followed by the zombie IP address.

This scan has three steps:

1. First Nmap records the current IP ID value on the idle host by sending it SYN/ACK TCP packet.
2. Nmap sends SYN packets the target's TCP ports using the idle host IP address.
3. Finally, Nmap triggers the idle host to respond so that you can compare the new IP ID value with the one received earlier.

If the port is closed, the target responds to the idle host with RST TCP packet, if the port is open, target responds with SYN/ACK to the idle host, then idle host responds with RST TCP packet.

Sometimes the target won't respond at all due to firewall rules.

. . .

Firewall Evasion

When performing network reconnaissance with Nmap, one of the key challenges is to avoid detection by firewalls.

Scanning a virtual machine, I'm going to try to bypass some firewall rules.

First, I'm going to discover whether the host is running or not.

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -PE -sn 192.168.144.130
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-23 04:10 +03
Nmap scan report for 192.168.144.130 (192.168.144.130)
Host is up (0.0012s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
```

Using ICMP Echo ping scan, I found the target host up.

Some firewalls may block these kinds of scans, here is an example of the same scan type blocked by a firewall.

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -PE -sn 192.168.144.130
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-23 04:08 +03
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 2.18 seconds
```

Usually, firewalls block these kinds of scans, so It's not common to discover hosts on the target's network.

I'm going to change the scanning technique to bypass that firewall.

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -PT -sn 192.168.144.130
[sudo] password for shark:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-23 06:10 +03
Nmap scan report for 192.168.144.130 (192.168.144.130)
Host is up (0.0023s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.24 seconds
```

Using TCP Ping Scan , Nmap is able to discover that host. Sometimes it can be blocked as so

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -PT -sn 192.168.144.130
[sudo] password for shark:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-23 06:56 +03
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 2.18 seconds
```

There is no Nmap option for detecting and subverting firewalls and IDS systems. It takes skill and experience.

Let's perform some port scanning to discover vulnerabilities.

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sT -p80,2220 192.168.144.130
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-23 07:06 +03
Nmap scan report for 192.168.144.130 (192.168.144.130)
Host is up (0.0049s latency).

PORT     STATE    SERVICE
80/tcp   filtered http
2220/tcp filtered netiq

Nmap done: 1 IP address (1 host up) scanned in 1.50 seconds
```

Nmap determined these ports as filtered which means that there is a firewall dropping sent packets.

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sS -sV -p80,2220 192.168.144.130
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-23 07:21 +03
Nmap scan report for 192.168.144.130 (192.168.144.130)
Host is up (0.0016s latency).

PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.54 ((Debian))
2220/tcp open  ssh     OpenSSH 9.1p1 Debian 1 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.81 seconds
```

Using TCP SYN Scan I'm able to see the ports as open. If there is a firewall rule blocking this scan type, the output is going to be as so

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sS -sV -p80,2220 192.168.144.130
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-23 07:31 +03
Nmap scan report for 192.168.144.130 (192.168.144.130)
Host is up (0.0047s latency).

PORT     STATE    SERVICE VERSION
80/tcp   filtered http
2220/tcp filtered netiq

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.91 seconds
```

Using a scan type like Xmas or TCP Window Scan, I will expect to find these ports as open, although these scans can be blocked.

Sometimes the best result can be `open|filtered` like this scan

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sX -sV -p80,2220 192.168.144.130
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-23 07:35 +03
Nmap scan report for 192.168.144.130 (192.168.144.130)
Host is up (0.0010s latency).

PORT     STATE         SERVICE    VERSION
80/tcp   open|filtered tcpwrapped
2220/tcp open|filtered tcpwrapped

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.94 seconds
```

Which means that Nmap can't determine whether the port is open or filtered.

Fragmenting the packets is an IDS evading way, splitting the TCP header over several packets makes it harder for packet filtering. Using `-f` or `--mtu` will perform packet fragmentation.

An example of packet fragmentation scan

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -mtu8 -sV -p80,2220 192.168.144.130
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-23 07:51 +03
Stats: 0:00:00 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Ping Scan Timing: About 100.00% done; ETC: 07:51 (0:00:00 remaining)
Nmap scan report for 192.168.144.130 (192.168.144.130)
Host is up (0.0012s latency).

PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.54 ((Debian))
2220/tcp open  ssh     OpenSSH 9.1p1 Debian 1 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.79 seconds
```

Asking Nmap to use an invalid TCP checksum for sent packets also is a firewall evading way, it's performed using `--badsum` option.

An example of bad checksums scan

```
┌──(shark㉿Shark)-[~]
└─$ nmap --badsum -sV -p80,2220 192.168.144.130
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-23 08:11 +03
You have specified some options that require raw socket access.
These options will not be honored without the necessary privileges.
Nmap scan report for 192.168.144.130 (192.168.144.130)
Host is up (0.0038s latency).

PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.54 ((Debian))
2220/tcp open  ssh     OpenSSH 9.1p1 Debian 1 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.35 seconds
```

Spoofing an IP address is a firewall evading way, this makes the target think that someone else is scanning them, so it's hard for firewalls to block the sent packets. But usually, Nmap doesn't receive reply packets back and you should capture the packets from the spoofed Ip address to determine if you could bypass the firewall.

An example of spoofing an Ip address

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -Pn -S 192.168.1.33 -e eth0 -p80,2220 192.168.144.130
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-23 08:29 +03
setup_target: failed to determine route to 192.168.144.130
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.05 seconds
```

Decoying IP addresses to scan a target host is a firewall evading way, it makes the hosts you specify appear to the target host are scanning the target too, the target's firewall won't know which IP address is scanning and which is not. It is a good way to hide your IP address while scanning.

It is important to use up hosts as decoys, otherwise it would be to the target host to determine which host is performing the scan.

An example of decoying IP addresses to perform a scan

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -D 192.168.1.33,RND,RND,RND,ME -p80,2220 192.168.144.130
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-23 08:45 +03
Nmap scan report for 192.168.144.130 (192.168.144.130)
Host is up (0.0041s latency).

PORT     STATE SERVICE
80/tcp   open  http
2220/tcp open  netiq

Nmap done: 1 IP address (1 host up) scanned in 0.33 seconds
```

Specifying a TTL value to the sent packets may be helpful to bypass some firewalls, it's performed by using `--ttl` option followed by a value.

An example of specifying a TTL value

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap --ttl 3 -p80,2220 192.168.144.130
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-23 08:34 +03
Nmap scan report for 192.168.144.130 (192.168.144.130)
Host is up (0.0060s latency).

PORT     STATE SERVICE
80/tcp   open  http
2220/tcp open  netiq

Nmap done: 1 IP address (1 host up) scanned in 0.40 seconds
```

...

Version detection

After discovering the open ports at the target's network, you can detect the running service on a specific port.

Adding `-sV` option is going to determine services and version information on the open ports by making a TCP 3-way handshake connection.

As I mentioned before TCP SYN Scan ends the connection by sending RST packet before completing the 3-way handshake, so that, TCP SYN Scan is not possible when `-sV` option is chosen.

An example of version detection

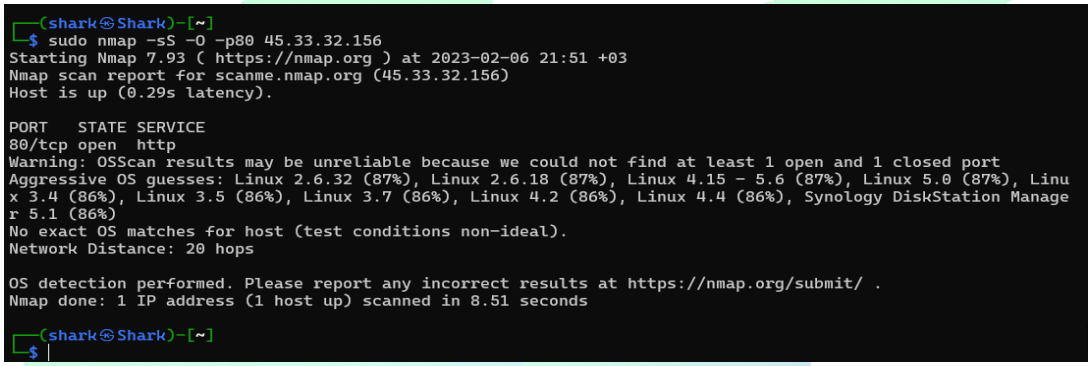
```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sS -sV -p21,22,23 45.33.32.156
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-06 21:49 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.24s latency).

PORT   STATE  SERVICE VERSION
21/tcp closed ftp
22/tcp open   ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
23/tcp closed telnet
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.77 seconds
```

...

OS Detection

Nmap can detect the Operating System based on its behavior and any signs in its responses. It is done with `-O` option.

An example of OS detection



...

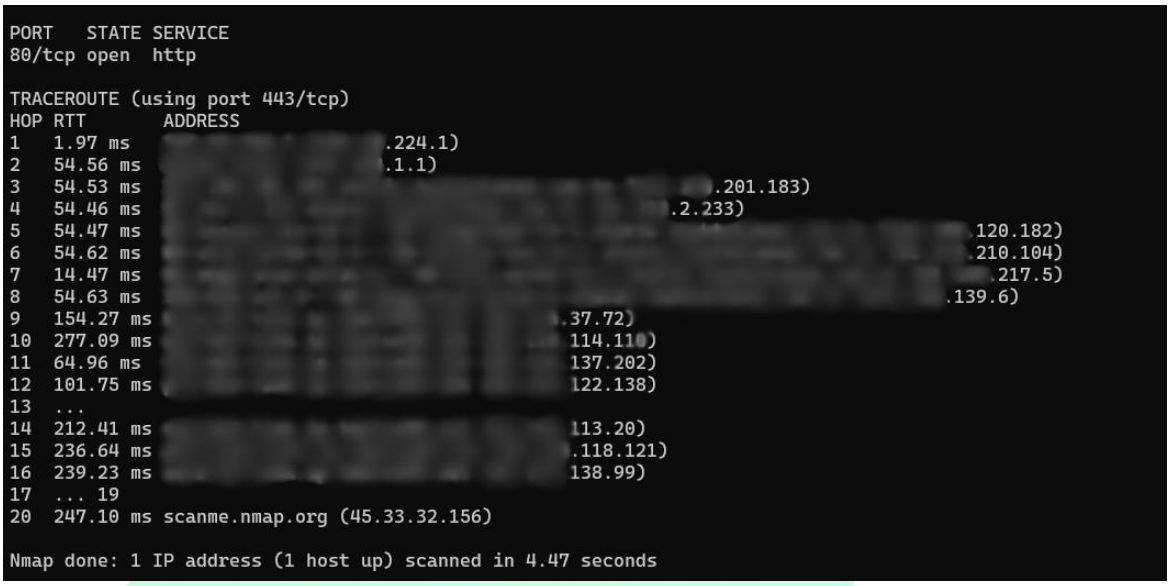
Traceroute

Nmap can find the routers between you and the target by using `--traceroute` option.

This option is the opposite of `traceroute` Linux command and macOS or `tracert` MS Windows command.

Nmap starts with a packet of high TTL and keeps decreasing it until it reaches the target.

An example of traceroute



. . .

Nmap Scripting Engine

The Nmap Scripting Engine is a powerful feature which allows users to write custom scripts to automate tasks and gather information from network devices.

Use `-sC` option to enable the most common scripts, or use `--script` option to choose your own scripts to execute by providing categories, script file names, or the name of directories full of scripts you with to execute.

Check the files at `/usr/share/nmap/scripts`, and you will see hundreds of scripts conveniently named starting with the protocol they target.

You can specify to use one of them of a group of these scripts by using --script followed by a script name for a single script execution. For example, use `-script “http-date”` to make Nmap get the date from HTTP-like services and print how much the date differs from local time, or use `--script “ftp*”` to make Nmap use all scripts starting with ftp.

SCRIPTING CATEGORIES

Nmap Scripting Engine scripts define a list of categories they belong to, category names are not case sensitive. The following list describes each category.

- Auth

  These scripts deal with authentication credentials on the target system.

- Broadcast

  Scripts in this category typically do discovery of hosts not listed on the command line by broadcasting on the local network.

- Brute

  These scripts use brute force attack to guess authentication credentials of a remote server.

- Default

  These scripts are the default set and are run when using `-sC` or `-A` options rather than listing scripts with `--script` . This category can also be specified explicitly like any other using `--script=default`.

- Discovery

  These scripts try to actively discover more about the network by querying public registries, SNMP-enabled devices, directory services , and the like.

- Dos

  Scripts in this category may cause a denial of service. Sometimes this is done to test vulnerability to a denial-of-service method, but more commonly it is an undesired by necessary side effect of testing for traditional vulnerability. These tests sometimes crash vulnerable services.

- Exploit

These scripts aim to actively exploit some vulnerability.

- External

  Scripts in this category may send data to a third-party database or other network resource. An example of this is `whois-ip` , which makes a connection to `whois` servers to learn about the address of the target.

- Fuzzer

  This category contains scripts which are designed to send server software unexpected or randomized fields in each packet.

- Intrusive

  There are scripts that cannot be classified in the safe category because the risks are too high that they will crash the target system, use up significant resources on the target host, or otherwise be perceived as malicious by the target's system administrator.

- Malware

  These scripts test whether the target platform is infected by malware or backdoors.

- Safe

  Scripts which weren't designed to crash services, use large amounts of network bandwidth or other resources, or exploit security holes are categorized as safe. These are less likely to offend remote administrators, though we can't guarantee that they won't ever cause adverse reactions. Most of these perform general network discovery.

- Version

  The scripts in this special category are an extension to the version detection feature and can't be selected explicitly. They are selected to run only if version detection `-sV` was requested.

- Vuln

  These scripts check for specific known vulnerabilities and generally only report results if they are found.

Check NSE Libraries for more detailed information
https://nmap.org/nsedoc/lib/

. . .

Saving the output

Whenever you run a Nmap scan, you can save the result into a text file. The three main formats are:

- Normal
- Grepable
- XML

NORMAL

The normal format is similar to the output on the screen you get after scanning targets. To save your output into a file using the normal format use `-oN` option followed by a file name.

An example of saving output in normal format

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sS -p53,80 scanme.nmap.org -oN output.txt
┌──(shark㉿Shark)-[~]
└─$ cat output.txt
# Nmap 7.93 scan initiated Mon Feb  6 22:01:41 2023 as: nmap -sS -p53,80 -oN output.txt scanme.nmap.org
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.26s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f

PORT   STATE  SERVICE
53/tcp closed domain
80/tcp open   http

# Nmap done at Mon Feb  6 22:01:42 2023 -- 1 IP address (1 host up) scanned in 1.21 seconds
```

Let's read the file.

GREPABLE

The grepable format got this name from the command `grep`. You can save your output in grapable format using `-oG` option followed by a file name. It makes the file easier for users to search for specific data from the output using `grep` command.

An example of saving output in grepable format

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sS -p53,80 scanme.nmap.org -oG output.txt
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-06 22:04 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.28s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f

PORT   STATE  SERVICE
53/tcp closed domain
80/tcp open   http

Nmap done: 1 IP address (1 host up) scanned in 0.97 seconds
```

Let's read the file

```
┌──(shark㉿Shark)-[~]
└─$ cat output.txt
# Nmap 7.93 scan initiated Mon Feb  6 22:04:24 2023 as: nmap -sS -p53,80 -oG output.txt scanme.nmap.org
Host: 45.33.32.156 (scanme.nmap.org)    Status: Up
Host: 45.33.32.156 (scanme.nmap.org)    Ports: 53/closed/tcp//domain///, 80/open/tcp//http///
# Nmap done at Mon Feb  6 22:04:25 2023 -- 1 IP address (1 host up) scanned in 0.97 seconds
```

XML

The XML format would be the most convenient to process the output in other programs. To save your output in XML format use `-oX` option followed by a file name.

An example of saving output in XML format

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sS -p53,80 scanme.nmap.org -oX output.txt
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-06 22:05 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.26s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f

PORT   STATE  SERVICE
53/tcp closed domain
80/tcp open   http

Nmap done: 1 IP address (1 host up) scanned in 0.91 seconds
```

Let's read the file

```
┌──(shark㉿Shark)-[~]
└─$ cat output.txt
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nmaprun>
<?xml-stylesheet href="file:///usr/bin/../share/nmap/nmap.xsl" type="text/xsl"?>
<!-- Nmap 7.93 scan initiated Mon Feb  6 22:05:56 2023 as: nmap -sS -p53,80 -oX output.txt scanme.nmap.org -->
<nmaprun scanner="nmap" args="nmap -sS -p53,80 -oX output.txt scanme.nmap.org" start="1675710356" startstr="Mon Feb  6 22:05:56 2023" version="7.93" xmloutputversion="1.05">
<scaninfo type="syn" protocol="tcp" numservices="2" services="53,80"/>
<verbose level="0"/>
<debugging level="0"/>
<hosthint><status state="up" reason="unknown-response" reason_ttl="0"/>
<address addr="45.33.32.156" addrtype="ipv4"/>
<hostnames>
<hostname name="scanme.nmap.org" type="user"/>
</hostnames>
</hosthint>
<host starttime="1675710356" endtime="1675710356"><status state="up" reason="echo-reply" reason_ttl="43"/>
<address addr="45.33.32.156" addrtype="ipv4"/>
<hostnames>
<hostname name="scanme.nmap.org" type="user"/>
<hostname name="scanme.nmap.org" type="PTR"/>
</hostnames>
<ports><port protocol="tcp" portid="53"><state state="closed" reason="reset" reason_ttl="44"/><service name="domain" method="table" conf="3"/></port>
<port protocol="tcp" portid="80"><state state="open" reason="syn-ack" reason_ttl="43"/><service name="http" method="table" conf="3"/></port>
</ports>
<times srtt="258307" rttvar="154700" to="877107"/>
</host>
<runstats><finished time="1675710357" timestr="Mon Feb  6 22:05:57 2023" summary="Nmap done at Mon Feb  6 22:05:57 2023; 1 IP address (1 host up) scanned in 0.91 seconds" elapsed="0.91" exit="success"/><hosts up="1" down="0" total="1"/>
</runstats>
</nmaprun>
```

You can save your output in all formats at once using `-oA` option followed by a file name.

An example of saving output into all three formats at once

```
┌──(shark㉿Shark)-[~]
└─$ sudo nmap -sS -p53,80 scanme.nmap.org -oA output.txt
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-06 22:07 +03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.23s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f

PORT   STATE  SERVICE
53/tcp closed domain
80/tcp open   http

Nmap done: 1 IP address (1 host up) scanned in 0.85 seconds
```

This is going to make three files similar to the previous ones.

. . .

Summary

Below is a summary of Nmap scan types and options I have covered.

Scan Type	Command
Scan random hosts	**-iR**
Ping Scan	**-sP**
ARP Scan	**-PR**
ICMP Echo Scan	**-PE**
ICMP Timestamp Scan	**-PP**
ICMP Address Mask Scan	**-PM**
TCP SYN Ping Scan	**-PS**
TCP ACK Ping Scan	**-PA**

TCP Connect Ping Scan	-PT
IP Ping Scan	-PO
TCP Connect port Scan	-sT
TCP Xmas Port Scan	-sX
TCP SYN Port Scan	-sS
UDP Port Scan	-sU
TCP FIN Port Scan	-sF

Scan Type	Command
TCP Xmas Port Scan	-sX
TCP Null Port Scan	-sN
IP Port Scanning	-sO
TCP Maimon Port Scan	-sM
TCP ACK Port Scan	-sA
TCP Window Port Scan	-sW
TCP Custom Port Scan	--scan-flags
Spoofed IP Scan	-S
Decoy Scan	-D
Idle/Zombie Scan	-sI
Bad checksums Scan	--badsum

Option	Command
Using list as input	-iL
Target listing	-sL
No DNS lookup	-n

Reverse-DNS lookup for all hosts	-R
100 most common ports	-F
No ping	-PN
Skip ping step	-Pn
Option	**Command**
100 most common ports	-F
All ports	-p-
Consecutively port scanning	-r
Only host discovery	-sn
Verbose mode	-v
Very verbose	-vv
Provide more details	--reason
Version detection	-sV
OS detection	-O
Traceroute	-traceroute
Execute scanning script	--script
Default scanning scripts	-sC
Packet fragmentation	-f
Specify fragmentation size	--mtu
Specify an interface	-e
Scan timing	-T<0-5>
Aggressive Scan	-A
Save output in normal format	-oN
Save output in grepable format	-oG

Save output in XML format	-oX
Option	**Command**
Save output in all three formats	-oA
Specify a value to the TTL filed	--ttl

. . .

References

1. "Nmap Scripting Engine: Introduction". Nmap.org.
2. "The History and the Future of Nmap". Nmap.org.
3. "Host Discovery (Ping Scanning)". Nmap.org.
4. "DNS Resolution". Nmap.org.
5. "Port Scanning Techniques and Algorithms". Nmap.org.
6. "Service and Application Version Detection". Nmap.org.
7. "Remote OS Detection". Nmap.org.
8. "Firewall/IDS Evasion and Spoofing". Nmap.org.
9. "Nmap Scripting Engine". Nmap.org.
10. "Nmap Reference Guide: Output". Nmap.org.
11. "Nmap Reference Guide". Nmap.org.
12. Joshi, Sager (2021-02-25). "What is Nmap And Why You Should Use It?". The Hack Report.

. . .

Done.