(I) NMAP Scanning techniques:

1. UDP Scan (-sU)

This technique is used to scan the open UDP ports of the target IP/host. Here, UDP scan sends UDP Packets to every ports of the target and waits till it get response. If, it receives error message stating that the ICMP is unreachable, this means that the port is closed. But, if gets any approachable response, then it means the port is open.

Command: nmap -sU <target>

2. FIN Scan (-sF)

In Fin Scan technique, packets are sent with a Fin Flag. Sometimes, because of firewall, SYN Packets might be blocked. In such case, FIN Scan works by by passing the firewall. FIN packets are send to closed ports, if no response is received, it is because either the packet is dropped by firewall or the port is open.

Command: nmap -sU <target>

3. Ping Scan (-sP)

This technique is only used to find out whether the host is available or not. Ping Scan is not used to detect open ports. It sends ICMP echo request and in return gets ICMP echo reply is the host is alive.

Command: nmap -sP <target>

4. TCP SYN Scan (-sS)

In this technique, Nmap sends SYN packets to the destination, but does not create any session. As a result, target computer wont be able to create any log of interaction as no session was initiated.

Command: nmap -sS <target>

5. TCP Connect() Scan (-sT)

UNIX socket uses a system call named connect() to begin TCP connection and if it succeeds, connection can be made and if it fails, connections cannot be made, basically because the port might be closed.

Command: nmap -sT <target>

6. Version Detection (-sV)

This technique is used to find out about specific service running on open port, its version and product Name. It is not used to detect open ports. However, this scan needs open ports in order to detect the version. It uses TCP SYN scan to know about the open ports.

Command: nmap -sV <target>

7. Idle Scan (-sI)

Idle scan is an advance scan that does not send any packets from your IP address, instead it uses another host from the target network to send the packets.

Command: nmap -sI <target>

(II)SCANNING SPECIFIC PORTS:

```
nmap -p 1-65535 localhost
```

In this example, we scanned all 65535 ports for our localhost computer.
Nmap is able to scan all possible ports, but you can also scan specific ports, which will report faster results. See below:

```
nmap -p 80,443 8.8.8.8
```

(III)MULTIPLE IP ADDRESS SCANNING:

Lets try to scan multiple IP addresses. For this you need to use this syntax:

```
nmap 1.1.1.1 8.8.8.8
```

You can also scan consecutive IP addresses:

```
nmap -p 1.1.1.1,2,3,4
```

This will scan 1.1.1.1, 1.1.1.2, 1.1.1.3 and 1.1.1.4.

(IV)SCANNING MOST POPULAR PORTS:

Using top-ports parameter along with a specific number lets you scan the top X most common ports for that host, as we can see:

```
nmap --top-ports 20 192.168.1.106
```

Replace 20 with the desired number

(V)SCANNING HOSTS AND IP ADDRESSES IN A TEXT FILE:

In this case, Nmap is also useful to read files that contain hosts and IPs inside.

Lets suppose you create a list.txt file that contains these lines inside:

```
192.168.1.106
cloudflare.com
microsoft.com
google.com
```

The -iL parameter lets you read from that file, and scan all those hosts for you:

```
nmap -iL list.txt
```

(VI)SAVE YOUR NMAP SCAN RESULT TO A FILE:

On the other hand, in the following example we will not be reading from a file, but exporting/saving our results into a text file:

```
nmap -oN output.txt google.com
```

Nmap has the ability to export files into XML format as well, see the next example:

```
nmap -oX output.xml google.com
```

(VII)DISABLING DNS NAME RESOLUTION:

If you need to speed up your scans a little bit, you can always choose to disable reverse DNS resolution for all your scans. Just add the -n parameter.

```
nmap -p 80 -n 8.8.8.8
```

See the difference with a normal DNS-resolution enabled scan:

```
nmap -p 80 8.8.8.8
```

(VIII)SCAN + OS AND SERVICE DETECTION WITH FAST EXECUTION:

Using the -A parameter enables you to perform OS and service detection, and at the same time we are combining this with -T4 for faster execution. See the example below:

```
nmap -A -T4 cloudflare.com
```

(IX)DETECT SERVICE_DAEMON VERSION:

This can be done by using -sV parameters

```
nmap -sV localhost
```

(X)SCAN USING TCP OR UDP PROTOCOLS:

One of the things we love most about Nmap is the fact that it works for both TCP and UDP protocols. And while most services run on TCP, you can also get a great advantage by scanning UDP-based services. Lets see some examples.

Standard TCP scanning output:

```
nmap -sT 192.168.1.1
```

UDP scanning results using -sU parameter:

```
nmap -sU localhost
```

(XI)Nmap Scripting Engine (NSE)

The Nmap Scripting Engine (NSE) is one of Nmap's most powerful and flexible features. It allows users to write (and share) simple scripts to automate a wide variety of networking tasks. Those scripts are then executed in parallel with the speed and efficiency you expect from Nmap. Users can rely on the growing and diverse set of scripts distributed with Nmap, or write their own to meet custom needs.
NSE designed to be versatile, with the following tasks in mind:

1. Network discovery

This is Nmap's bread and butter. Examples include looking up whois data based on the target domain, querying ARIN, RIPE, or APNIC for the target IP to determine ownership, performing identd lookups on open ports, SNMP queries, and listing available NFS/SMB/RPC shares and services.

2. More sophisticated version detection

The Nmap version detection system is able to recognize thousands of different services through its probe and regular expression signature based matching system, but it cannot recognize everything. For example, identifying the Skype v2 service requires two independent probes, which version detection isn't flexible enough to handle. Nmap could also recognize more SNMP services if it tried a few hundred different community names by brute force. Neither of these tasks are well suited to traditional Nmap version detection, but both are easily accomplished with NSE. For these reasons, version detection now calls NSE by default to handle some tricky services.

3. Vulnerability detection

When a new vulnerability is discovered, you often want to scan your networks quickly to identify vulnerable systems before the bad guys do. While Nmap isn't a comprehensive vulnerability scanner, NSE is powerful enough to handle even demanding vulnerability checks. When the Heartbleed bug affected hundreds of thousands of systems worldwide, Nmap's developers responded with the ssl-heartbleed detection script within 2 days. Many vulnerability detection scripts are already available and we plan to distribute more as they are written.

4. Backdoor detection

Many attackers and some automated worms leave backdoors to enable later reentry. Some of these can be detected by Nmap's regular expression based version detection, but more complex worms and backdoors require NSE's advanced capabilities to reliably detect. NSE has been used to detect the Double Pulsar NSA backdoor in SMB and backdoored versions of UnrealIRCd, vsftpd, and ProFTPd.

5. Vulnerability exploitation

As a general scripting language, NSE can even be used to exploit vulnerabilities rather than just find them. The capability to add custom exploit scripts may be

valuable for some people (particularly penetration testers), though we aren't planning to turn Nmap into an exploitation framework such as Metasploit.
These listed items were our initial goals, and we expect Nmap users to come up with even more inventive uses for NSE.

NSE is activated with the -sC option (or --script if you wish to specify a custom set of scripts) and results are integrated into Nmap normal and XML output.

(XII) CVE DETECTION USING NMAP:

One of Nmaps greatest features that not all the network and systems administrators know about is something called Nmap Scripting Engine (known as NSE). This scripting engine allows users to use a pre-defined set of scripts, or write their own using Lua programming language.
Using Nmap scripts is crucial in order to automate system and vulnerability scans.
For example, if you want to run a full vulnerability test against your target, you can use these parameters:

```
nmap -Pn --script vuln 192.168.1.105
```

(XIII) DOS USING NMAP:

Nmap features never seem to end, and thanks to the NSE, that even allows us to launch DOS attacks against our network testings.
for example we can try to exploit the slowloris vulnerability by launching a DOS attack in a forever loop:

```
nmap 192.168.1.105 -max-parallelism 800 -Pn --script http-slowloris --script-args
http-slowloris.runforever=true
```

(XIV) LAUNCHING BRUTEFORCE ATTACK:

NSE is really fascinating it contains scripts for everything you can imagine. See the next three examples of BFA against WordPress, MSSQL, and FTP server:
WordPress brute force attack:

```
nmap -sV --script http-wordpress-brute --script-args
'userdb=users.txt,passdb=passwds.txt,http-wordpress-brute.hostname=domain.com,
http-wordpress-brute.threads=3,brute.firstonly=true' 192.168.1.105
```

Brute force attack against MS-SQL:

```
nmap -p 1433 --script ms-sql-brute --script-args
userdb=customuser.txt,passdb=custompass.txt 192.168.1.105
```

FTP brute force attack:

```
nmap --script ftp-brute -p 21 192.168.1.105
```

(XV) DETECTING MALWARE INFECTIONS ON REMOTE HOST:

Nmap is able to detect malware and backdoors by running extensive tests on a few popular OS services like on Identd, Proftpd, Vsftpd, IRC, SMB, and SMTP. It also has a module to check for popular malware signs inside remote servers and integrates Googles Safe Browsing and VirusTotal databases as well.
A common malware scan can be performed by using:

```
nmap -sV --script=http-malware-host 192.168.1.105
```

Or using Googles Malware check:

```
nmap -p80 --script http-google-malware infectedsite.com
```

(XVI) Nmap Firewall and IDS Evasion

1. Fragment Packets

This technique was very effective especially in the old days however you can still use it if you found a firewall that is not properly configured. The Nmap offers that ability to fragment the packets while scanning with the -f option so it can bypass the packet inspection of firewalls.

2. SPECIFY A SPECIFIC MTU

Nmap is giving the option to the user to set a specific MTU (Maximum Transmission Unit) to the packet.This is similar to the packet fragmentation technique that we have explained above.During the scan that size of the nmap will create packets with size based on the number that we will give.In this example we gave the number 24 so the nmap will create 24-byte packets causing a confusion to the firewall.
Have in mind that the MTU number must be a multiple of 8 (8,16,24,32 etc). You can specify the MTU of your choice with the command mtu number target.

3. USE DECOY ADDRESSES

In this type of scan you can instruct Nmap to spoof packets from other hosts.In the firewall logs it will be not only our IP address but also and the IP addresses of the decoys so it will be much harder to determine from which system the scan started.
There are two options that you can use in this type of scan:

```
nmap -D RND:10 [target] (Generates a random number of decoys)
nmap -D decoy1,decoy2,decoy3 etc. (Manually specify the IP addresses of the decoys)
```

4. IDLE ZOMBIE SCAN

This technique allows you to use another host on the network that is idle in order to perform a port scan to another host.The main advantage of this method is that it very stealthy because the firewall log files will record the IP address of the Zombie and not our IP.
However in order to have proper results we must found hosts that are idle on the network.
Metasploit framework has a scanner that can help us to discover hosts that are idle on the network and it can be used while implementing this type of scan.

```
nmap -sI [Zombie IP] [Target IP]
```

5. SOURCE PORT NUMBER SPECIFICATION

A common error that many administrators are doing when configuring firewalls is to set up a rule to allow all incoming traffic that comes from a specific port number.
The source-port option of Nmap can be used to exploit this misconfiguration.Common ports that you can use for this type of scan are: 20,53 and 67.

6. APPEND RANDOM DATA

Many firewalls are inspecting packets by looking at their size in order to identify a potential port scan.This is because many scanners are sending packets that have specific size.

In order to avoid that kind of detection you can use the command data-length to add additional data and to send packets with different size than the default.

7. SCAN WITH RANDOM ORDER

In this technique you can scan a number of hosts in random order and not sequential.The command that you use to instruct Nmap to scan for host in random order is randomize-hosts.
This technique combined with slow timing options in nmap command can be very effective when you dont want to alert firewalls.

8. MAC ADDRESS SPOOFING

Another method for bypassing firewall restrictions while doing a port scan is by spoofing the MAC address of your host.This technique can be very effective especially if there is a MAC filtering rule to allow only traffic from certain MAC addresses so you will need to discover which MAC address you need to set in order to obtain results.
Specifically the spoof-mac option gives you the ability to choose a MAC address from a specific vendor,to choose a random MAC address or to set a specific MAC address of your choice.
Another advantage of MAC address spoofing is that you make your scan more stealthier because your real MAC address it will not appear on the firewall log files.
Specify MAC address from a Vendor -> spoof-mac Dell/Apple/3Com
Generate a random MAC address -> spoof-mac 0
Specify your own MAC address -> spoof-mac 00:01:02:25:56:AE

9. SEND BAD CHECKSUMS

Checksums are used by the TCP/IP protocol to ensure the data integrity.However sending packets with incorrect checksums can help you to discover information from systems that is not properly configured or when you are trying to avoid a firewall.
You can use the command nmap badsum IP in order to send packets with bad checksums to your targets.