1.Basic scan using nmap

```
nmap 1.1.1.1
```

Now, if you want to scan a hostname, simply replace the IP for the host, as you see below:

```
nmap cloudflare.com
```

These kinds of basic scans are perfect for your first steps when starting with Nmap.

2. Scan specific ports or scan entire port ranges on a local or remote server

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

In this example, we scanned all 65535 ports for our localhost computer.

Nmap can scan all ports, but you can also scan specific ports, which will report faster results. See below:

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

3. Scan multiple IP addresses

Let us 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`.

4. Scan IP ranges

You can also use Nmap to scan entire CIDR IP ranges, for example:

```
nmap -p 8.8.8.0/28
```

This will scan 14 consecutive IP ranges, from 8.8.8.1 to 8.8.8.14.

An alternative is to simply use this kind of range:

```
nmap 8.8.8.1-14
```

You can even use wildcards to scan the entire C class IP range, for example:

```
nmap 8.8.8.*
```

This will scan 256 IP addresses from 8.8.8.1 to 8.8.8.256.

If you ever need to exclude certain IPs from the IP range scan, you can use the "–exclude" option, as you see below:

```
nmap -p 8.8.8.* --exclude 8.8.8.1
```

5. Scan the 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
```

where 20 can be replaced by desired port number.

6. Scan hosts and IP addresses reading from a text file

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

Let us suppose you create a list.txt file that has these lines inside:

```
192.168.1.106

www.google.com

www.demo.testfire.com
```

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

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

7. Save your Nmap scan results 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 securitytrails.com
```

Nmap can export files into XML format as well, see the next example:

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

8. 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.

```
[root@kali:~]nmap -p 80 -n 8.8.8.8
Starting Nmap 7.60 ( https://nmap.org ) at 2018-10-01 09:15 -03
Nmap scan report for 8.8.8.8
Host is up (0.014s latency).
PORT   STATE    SERVICE
```

```
80/tcp filtered http
```

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

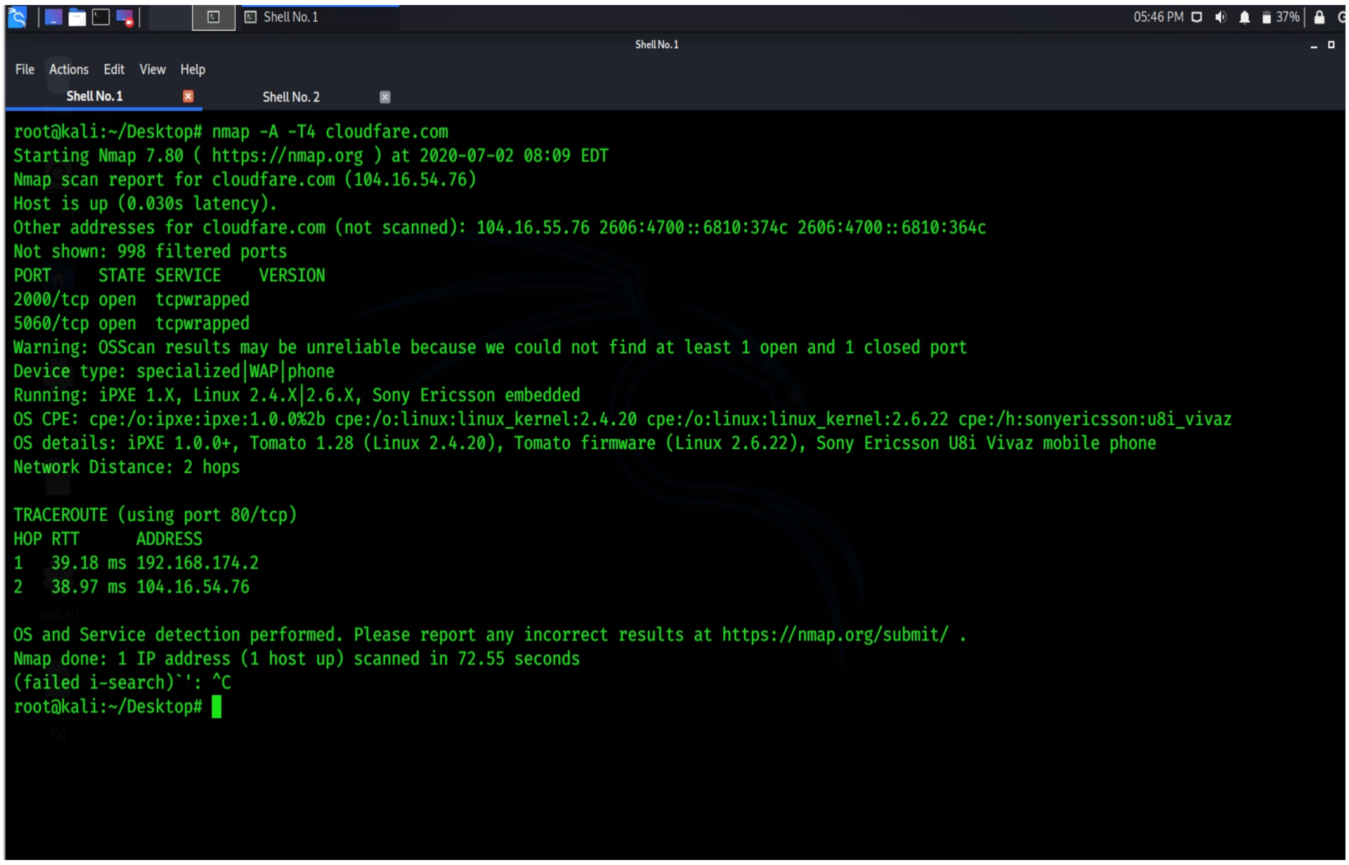
```
[root@kali:~]nmap -p 80 8.8.8.8
Starting Nmap 7.60 ( https://nmap.org ) at 2018-10-01 09:15 -03
Nmap scan report for google-public-dns-a.google.com (8.8.8.8)
Host is up (0.014s latency).
PORT   STATE    SERVICE
80/tcp filtered http
```

9. 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
```

This is the output we got for this test:



```
root@kali:~/Desktop# nmap -A -T4 cloudfare.com
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-02 08:09 EDT
Nmap scan report for cloudfare.com (104.16.54.76)
Host is up (0.030s latency).
Other addresses for cloudfare.com (not scanned): 104.16.55.76 2606:4700::6810:374c 2606:4700::6810:364c
Not shown: 998 filtered ports
PORT     STATE SERVICE    VERSION
2000/tcp open  tcpwrapped
5060/tcp open  tcpwrapped
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: specialized|WAP|phone
Running: iPXE 1.X, Linux 2.4.X|2.6.X, Sony Ericsson embedded
OS CPE: cpe:/o:ipxe:ipxe:1.0.0%2b cpe:/o:linux:linux_kernel:2.4.20 cpe:/o:linux:linux_kernel:2.6.22 cpe:/h:sonyericsson:u8i_vivaz
OS details: iPXE 1.0.0+, Tomato 1.28 (Linux 2.4.20), Tomato firmware (Linux 2.6.22), Sony Ericsson U8i Vivaz mobile phone
Network Distance: 2 hops

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   39.18 ms 192.168.174.2
2   38.97 ms 104.16.54.76

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 72.55 seconds
(failed i-search)`': ^C
root@kali:~/Desktop# 
```

10. Detect service/daemon versions

This can be done by using -sV parameters

nmap -sV localhost

As you can see here:

```
[root@kali:~]nmap -sV localhost
Starting Nmap 7.60 ( https://nmap.org ) at 2018-10-01 09:28 -03
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000020s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 997 closed ports
PORT STATE SERVICE VERSION
111/tcp open rpcbind 2-4 (RPC #100000)
631/tcp open ipp CUPS 2.2
902/tcp open ssl/vmware-auth VMware Authentication Daemon 1.10 (U
ses VNC, SOAP)
```

```
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.96 seconds
```

11. 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. Let us see some examples.

Standard TCP scanning output:

```
[root@kali:~]nmap -sT 192.168.1.1
Starting Nmap 7.60 ( https://nmap.org ) at 2018-10-01 09:33 -03
Nmap scan report for 192.168.1.1
Host is up (0.58s latency).
Not shown: 995 closed ports
PORT STATE SERVICE
80/tcp open http
1900/tcp open upnp
20005/tcp open btx
49152/tcp open unknown
49153/tcp open unknown
Nmap done: 1 IP address (1 host up) scanned in 1.43 seconds
```

UDP scanning results using “-sU” parameter:

```
[root@kali:~]nmap -sU localhost
Starting Nmap 7.60 ( https://nmap.org ) at 2018-10-01 09:37 -03
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000021s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 997 closed ports
PORT STATE SERVICE
68/udp open|filtered dhcpc
111/udp open rpcbind
```

```
5353/udp open|filtered zeroconf
```

12. CVE detection using Nmap

One of Nmap’s 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 NSE is crucial 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
```

Output example:

```
[root@kali:~]nmap -Pn --script vuln 192.168.1.105
Starting Nmap 7.60 ( https://nmap.org ) at 2018-10-01 09:46 -03
Pre-scan script results:
| broadcast-avahi-dos:
| Discovered hosts:
| 224.0.0.251
| After NULL UDP avahi packet DoS (CVE-2011-1002).
|_ Hosts are all up (not vulnerable).
Nmap scan report for 192.168.1.105
Host is up (0.00032s latency).
Not shown: 995 closed ports
PORT STATE SERVICE
80/tcp open http
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-DOM-based-xss: Couldn't find any DOM based XSS.
| http-slowloris-check:
| VULNERABLE:
| Slowloris DOS attack
| State: LIKELY VULNERABLE
| IDs: CVE:CVE-2007-6750
| Slowloris tries to keep many connections to the target web server op
en and hold
| them open if possible. It carries out this by opening connections to
| the target web server and sending a partial request. By doing so, it st
arves
| the http server's resources causing Denial Of Service.
|
| Disclosure date: 2009-09-17
| References:
| http://ha.ckers.org/slowloris/
|_ https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to d
ebug)
1900/tcp open upnp
20005/tcp open btx
49152/tcp open unknown
49153/tcp open unknown
```

As you can see, in this vulnerability test we were able to detect one CVE (Slowloris DOS attack).

13. Launching DOS with Nmap

Nmap features never seem to end, and thanks to the NSE, that even allows us to launch DOS attacks against our network testings.

In our earlier example (#12) we found the host was vulnerable to Slowloris attack, and now we will try to exploit that vulnerability by launching a DOS attack in a forever loop:

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

14. Launching brute force attacks

NSE is really fascinating – it has 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
```

15. Detecting malware infections on remote hosts

Nmap can 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 Google's 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 Google's Malware check:

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

Output example:

```
80/tcp open  http
|_http-google-malware.nse: Host is known for distributing malware.
```

ADVANCE N-MAP COMMANDS

(Note: - For this section we will be using diffrent IP addresses as our target.)

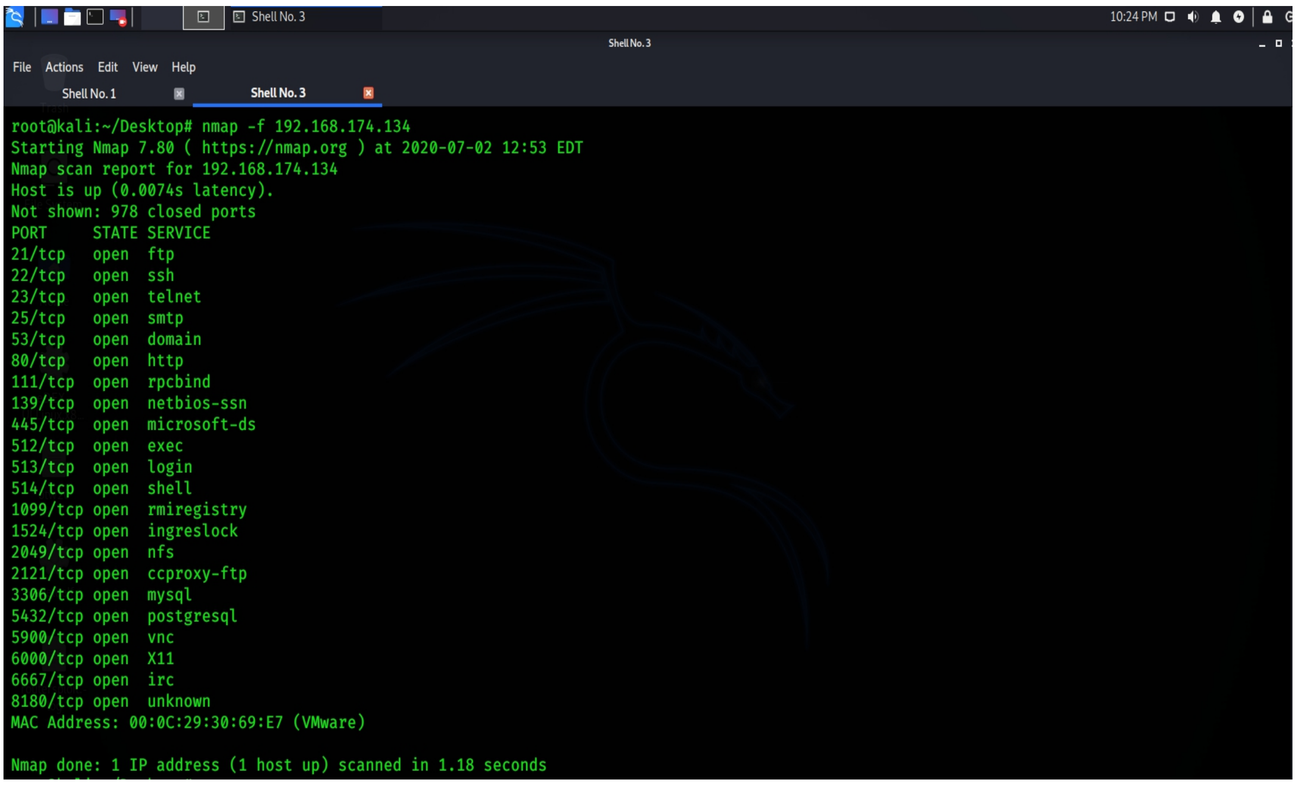
Firewall Evasion Commands

1.Fragment Packets

We can just send small chunks of packets usually of 8 bytes, by using this command will just evade the firewall, Also this is going to scan the most targeted top 1000 ports, so this is not helpful if our target has done port forwarding for the standard ports.

nmap -f 192.168.174.134

Output for the above command: -



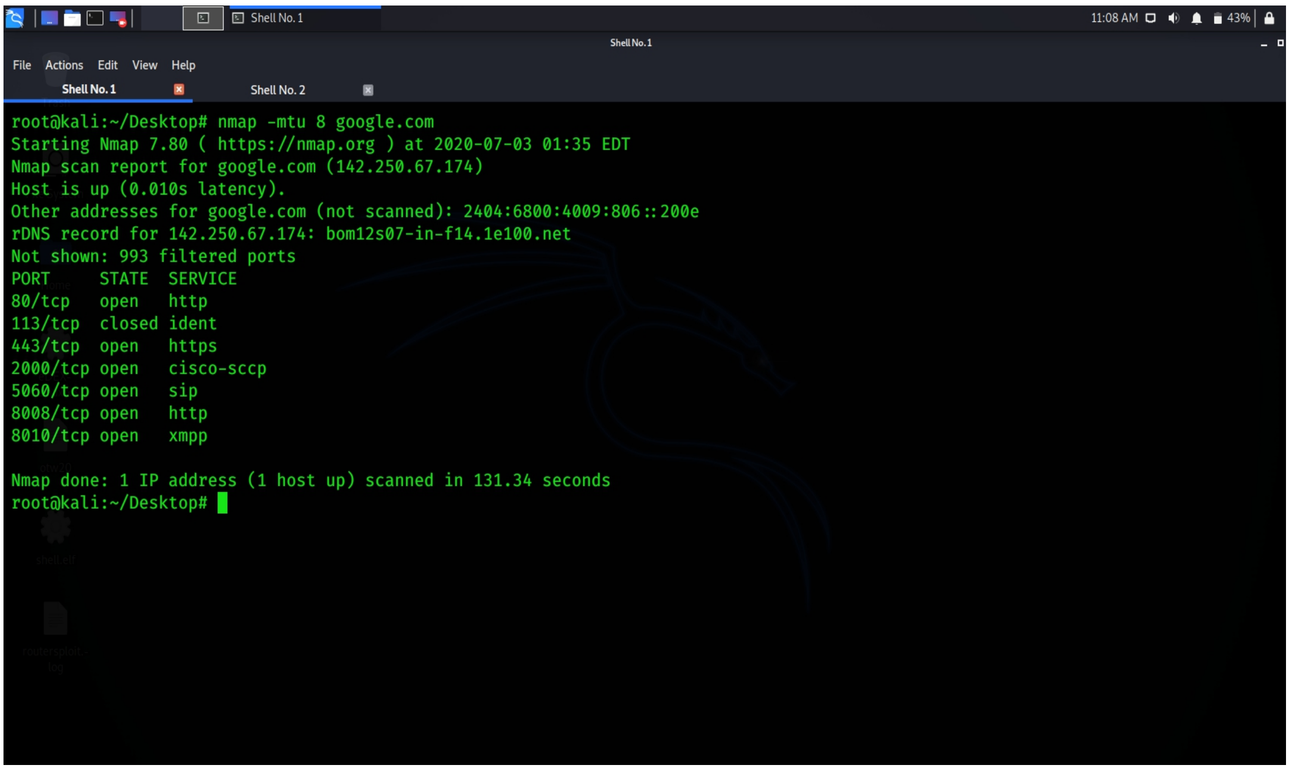
```
root@kali:~/Desktop# nmap -f 192.168.174.134
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-02 12:53 EDT
Nmap scan report for 192.168.174.134
Host is up (0.0074s latency).
Not shown: 978 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8180/tcp open  unknown
MAC Address: 00:0C:29:30:69:E7 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.18 seconds
```

2. Specify a specific MTU

Now here MTU stands for maximum transmission units, this command is pretty simmilar to the previous command i.e -f, here we will send 8 bytes of small bin packets.Basically we can send any packets of size of multiple of 8, insted of a large size of a big packets now we will send some small chunks and hence evade the WAF. Now for this command we will scan google.com and see the diffrence.

nmap -mtu 8 google.com

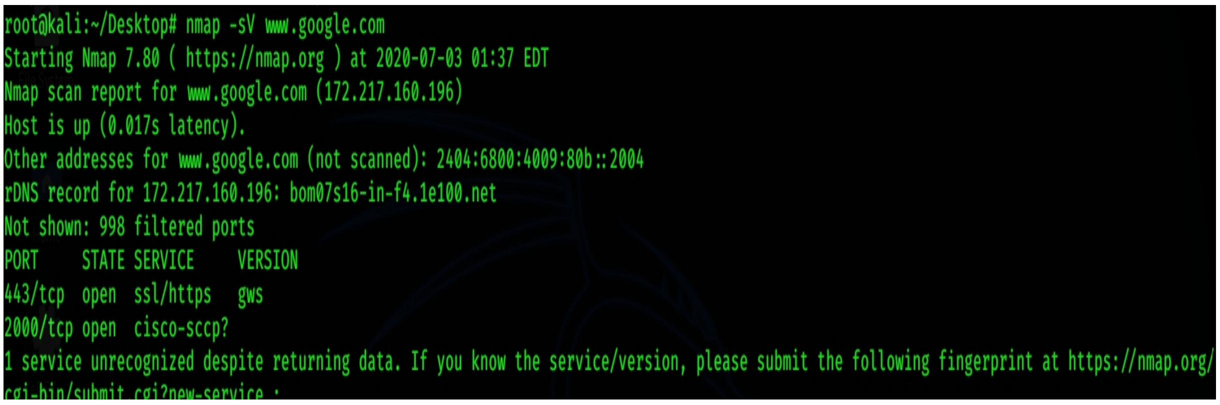
Output for the above command: -



```
root@kali:~/Desktop# nmap -mtu 8 google.com
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-03 01:35 EDT
Nmap scan report for google.com (142.250.67.174)
Host is up (0.010s latency).
Other addresses for google.com (not scanned): 2404:6800:4009:806::200e
rDNS record for 142.250.67.174: bom12s07-in-f14.1e100.net
Not shown: 993 filtered ports
PORT     STATE  SERVICE
80/tcp   open   http
113/tcp  closed ident
443/tcp  open   https
2000/tcp open   cisco-sccp
5060/tcp open   sip
8008/tcp open   http
8010/tcp open   xmpp

Nmap done: 1 IP address (1 host up) scanned in 131.34 seconds
root@kali:~/Desktop# 
```

Now to see the diffrence you will see if you use –sV to scan google then firewall will drop the packets and you will not recieve info about all the open ports, which you can get if you use –mtu [mtu] tag.

Output for simple –sV commnand:-

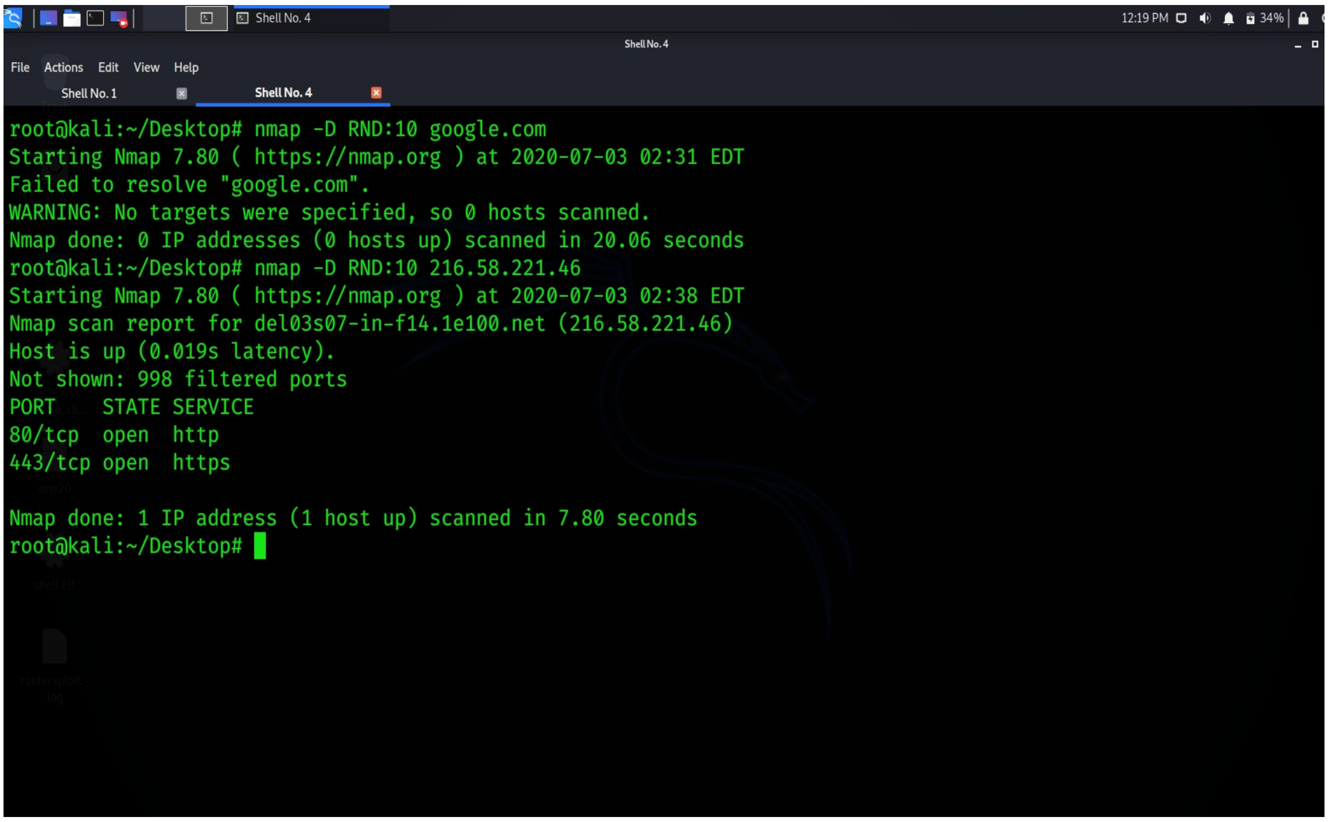

```
root@kali:~/Desktop# nmap -sV www.google.com
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-03 01:37 EDT
Nmap scan report for www.google.com (172.217.160.196)
Host is up (0.017s latency).
Other addresses for www.google.com (not scanned): 2404:6800:4009:80b::2004
rDNS record for 172.217.160.196: bom07s16-in-f4.1e100.net
Not shown: 998 filtered ports
PORT     STATE SERVICE     VERSION
443/tcp  open  ssl/https   gws
2000/tcp open  cisco-sccp?
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/
cgi-bin/submit.cgi?new-service :
```

3.Use a decoy

Now, here we will see the use of a decoy, decoys are scanning the target network too. Thus their IDS might report 5–10 port scans from unique IP Designer's excellent Scanlogd) are unlikely to show your IP address at all. If you don't use ME, Nmap will put you in a random position. You can also use RND to generate a random, non-reserved IP address, or RND:number to Decoys are used both in the initial ping scan (using ICMP, SYN, ACK, or whatever) and during the actual port scanning phase. Decoys are also used during remote OS detection (-O).

nmap –D RND:10 google.com

Command output: -



```
root@kali:~/Desktop# nmap -D RND:10 google.com
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-03 02:31 EDT
Failed to resolve "google.com".
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 20.06 seconds
root@kali:~/Desktop# nmap -D RND:10 216.58.221.46
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-03 02:38 EDT
Nmap scan report for del03s07-in-f14.1e100.net (216.58.221.46)
Host is up (0.019s latency).
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 7.80 seconds
root@kali:~/Desktop# 
```

4. Idle Zombie scan

Predictable IP fragmentation ID sequence generation on the zombie host to glean information about the open ports on the target. IDS systems will display the scan as coming from the zombie machine perspective of the zombie host. So you can try scanning a target using various zombies that You can add a colon followed by a port number to the zombie host if you wish to probe a particular port on the zombie for IP ID changes. Otherwise Nmap will use the port it uses by default for TCP.

nmap –sI 192.168.174.131:4444