

Nmap for Pentester: Ping Scan

February 25, 2018 By Raj Chandel

In this article we are going to scan the target machine with different Nmap ping scans and the response packets of different scans can be confirmed by analysis of Nmap traffic through Wireshark.

Ping scan in nmap is done to check if the target host is alive or not. As we know that ping by default send the ICMP echo request and gets an ICMP echo reply if the system is alive. Ping scan by default send an ARP packet and gets a response to check if the host is up.

Nmap scans changes their behavior according to the network they are scanning.

- Scanning Local Network with Nmap where nmap sends an ARP packet with every scan
- If an external network is to be scanned; Nmap sends following request packets:
 1. ICMP echo request
 2. ICMP timestamp request
 3. TCP SYN to port 443
 4. TCP ACK to port 80

In this article we are using —**disable-arp-ping** attribute for changing the behavior of nmap scans to treat a local network as a public network.

Let's Start!!

Ping Sweep

In order to identify live host without using ARP request packet Nmap utilize —**sP option** which is known as Ping Sweep Scan. We can use —**sn flag** which means no port scan also known as **ping scan**.

```
nmap -sP 192.168.1.104 --disable-arp-ping
or
nmap -sn 192.168.1.104 --disable-arp-ping
```

From given below image you can observe it found **1 Host is up**. Since we have disabled Arp request packet for local network scans by using parameter —**disable-arp-ping** therefore here it will treat it as an external network and behave accordingly that as discussed above.

```
root@kali:~# nmap -sP 192.168.1.104 --disable-arp-ping ↩

Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-24 05:04 EST
Nmap scan report for 192.168.1.104
Host is up (0.00020s latency).
MAC Address: 00:0C:29:6B:71:A7 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds
root@kali:~#
```

Demonstrating working of Ping Sweep using Wireshark

From given below image you can observe following packet of request and reply between both network IP

- 1. ICMP echo request
- 2. TCP SYN to port 443
- 3. TCP ACK to port 80
- 4. ICMP timestamp request
- 5. ICMP echo reply
- 6. TCP RST, ACK to port 443
- 7. TCP RST to port 80
- 8. ICMP Timestamp Reply

ip.addr == 192.168.1.104							
No.	Time	Source	Destination	Protoc	Length	Info	
6	5.6415...	192.168.1.101	192.168.1.104	ICMP	42	Echo (ping) request id:	
7	5.6416...	192.168.1.101	192.168.1.104	TCP	58	53603 → 443 [SYN] Seq=0	
8	5.6416...	192.168.1.101	192.168.1.104	TCP	54	53603 → 80 [ACK] Seq=1	
9	5.6417...	192.168.1.101	192.168.1.104	ICMP	54	Timestamp request id:	
10	5.6417...	192.168.1.104	192.168.1.101	ICMP	60	Echo (ping) reply id:	
11	5.6418...	192.168.1.104	192.168.1.101	TCP	60	443 → 53603 [RST, ACK]	
12	5.6418...	192.168.1.104	192.168.1.101	TCP	60	80 → 53603 [RST] Seq=1	
13	5.6418...	192.168.1.104	192.168.1.101	ICMP	60	Timestamp reply id:	

Block Ping Sweep Scan

Now let's put some firewall rules in IPTABLES to drop ICMP packets, TCP SYN packets on port 443 and TCP ACK on port 80 which will block Ping Sweep scan

```
sudo iptables -I INPUT -p ICMP -j DROP
sudo iptables -I INPUT -p tcp --tcp-flags ALL ACK --dport 80 -j DROP
sudo iptables -I INPUT -p tcp --tcp-flags ALL SYN --dport 443 -j DROP
```

```

pentest@ubuntu:~$ sudo iptables -I INPUT -p ICMP -j DROP
pentest@ubuntu:~$ sudo iptables -I INPUT -p tcp --tcp-flags ALL ACK --dport 80 -j DROP
pentest@ubuntu:~$ sudo iptables -I INPUT -p tcp --tcp-flags ALL SYN --dport 443 -j DROP
pentest@ubuntu:~$

```

Now repeat again ping sweep scan for identifying the state of the live host. From given below image you can observe this time it shows that **0 host is up** which means the firewall has blocked packets send by this scan.

```

root@kali:~# nmap -sP 192.168.1.104 --disable-arp-ping

Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-24 05:06 EST
Note: Host seems down. If it is really up, but blocking our ping probes,
Nmap done: 1 IP address (0 hosts up) scanned in 3.09 seconds
root@kali:~#

```

Again demonstrating request packets of Ping Sweep scan with Wireshark and if you notice given below image then you will found that this time it has not received any reply packet.

ip.addr == 192.168.1.104							Expression...
No.	Time	Source	Destination	Protocol	Length	Info	
2	0.3833...	192.168.1.101	192.168.1.104	ICMP	42	Echo (ping) request id=0xf772, s...	
3	0.3834...	192.168.1.101	192.168.1.104	TCP	58	39937 → 443 [SYN] Seq=0 Win=1024	
4	0.3834...	192.168.1.101	192.168.1.104	TCP	54	39937 → 80 [ACK] Seq=1 Ack=1 Win=...	
5	0.3834...	192.168.1.101	192.168.1.104	ICMP	54	Timestamp request id=0x4a55, s...	
11	2.3860...	192.168.1.101	192.168.1.104	ICMP	54	Timestamp request id=0x1d7b, s...	
12	2.3862...	192.168.1.101	192.168.1.104	TCP	54	39938 → 80 [ACK] Seq=1 Ack=1 Win=...	
13	2.3863...	192.168.1.101	192.168.1.104	TCP	58	39938 → 443 [SYN] Seq=0 Win=1024	
14	2.3864...	192.168.1.101	192.168.1.104	ICMP	42	Echo (ping) request id=0x06a5, s...	

Bypass Ping Sweep Filter using TCP SYN Ping

Now, we'll try to bypass the firewall rules by using ping scan with **TCP SYN packets**, for that we'll use **-PS attribute**. -PS sends TCP SYN packet on port 80 by default; we can change it by specifying the ports with it, like -PS443.

```

nmap -sP -PS 192.168.1.104 --disable-arp-ping

```

From given below image you can observe that observe it found **1 Host is up**.

```

root@kali:~# nmap -sP -PS 192.168.1.104 --disable-arp-ping

Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-24 05:08 EST
Nmap scan report for 192.168.1.104
Host is up (0.00027s latency).
MAC Address: 00:0C:29:6B:71:A7 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds
root@kali:~#

```

From given below image you can observe that it is showing the result which similar to NMAP stealth scan. Here it is following TCP Half connection mechanism where SYN packet is sent on port 80 and received SYN, ACK from port 80 and then RST packet for reset connection

The difference between -sP packet on port 80 and -PS packet on port 80 is as following:

- Ping sweep scan [-sP] send TCP ACK packet on port 80 and hex value of ACK flag is 10, as the reply from host machine it receives RST packet whose hex value is 4.
- TCP SYN Ping scan send TCP SYN packet on port 80 and its hex value is 2, as a reply it received SYN, ACK packet whose value is some of their hex value i.e. $2 + 10 = 12$ and able to bypass above firewall rule applied on port 80 for TCK ACK packet.

ip.addr == 192.168.1.104								Expression...	+
	Time	Source	Destination	Protoc	Length	Info			
4	2.6045...	192.168.1.101	192.168.1.104	TCP	58	64604 → 80	[SYN]	Seq=0	Win=1024 Le
5	2.6048...	192.168.1.104	192.168.1.101	TCP	60	80 → 64604	[SYN, ACK]	Seq=0	Ack=1
6	2.6048...	192.168.1.101	192.168.1.104	TCP	54	64604 → 80	[RST]	Seq=1	Win=0 Len=0

Block TCP SYN Ping Scan

Sometimes network admin applies the filter as given below using Iptables on TCP SYN packet to drop all SYN packet to initiate TCP connection with all TCP Port in their network.

```
sudo iptables -I INPUT -p tcp --tcp-flags ALL SYN -j DROP
```

As result, it blocks the NMAP TCP SYN Ping probes so that it could not identify the state of the live host.

```
pentest@ubuntu:~$ sudo iptables -I INPUT -p tcp --tcp-flags ALL SYN -j DROP
pentest@ubuntu:~$
```

Now repeat again TCP SYN Ping for identifying the state of the live host. From given below image you can observe this time it shows that **0 host is up** which means the firewall has blocked packets send by this scan.

```
root@kali:~# nmap -sP -PS 192.168.1.104 --disable-arp-ping
Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-19 05:16 EST
Note: Host seems down. If it is really up, but blocking our ping probes,
Nmap done: 1 IP address 0 hosts up scanned in 2.07 seconds
```

Bypass TCP SYN Ping using TCP ACK Ping

In order to bypass this, we'll use ping scan using TCP ACK packets, for that we'll use -PA attribute. -PA sends TCP ACK packet on port 80 by default, we can change it by specifying the ports with it, like -PA443

```
nmap -sP -PA 192.168.1.104 --disable-arp-ping
```

From given below image you can observe that it has found **1 Host is up**.

```
root@kali:~# nmap -sP -PA 192.168.1.104 --disable-arp-ping
Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-19 05:25 EST
Nmap scan report for 192.168.1.104
Host is up (0.00031s latency).
MAC Address: 00:0C:29:48:22:FD (VMware)
Nmap done: 1 IP address (1 host up) scanned in 13.12 seconds
```

When you will notice given below packets captured by Wireshark you will found that here ACK packet is sent on port 80 as reply received RST packet from port 80.

Source	Destination	Protocol	Length	Info
192.168.1.103	192.168.1.104	TCP	54	61300 → 80 [ACK] Seq=1 Ac...
192.168.1.104	192.168.1.103	TCP	60	80 → 61300 [RST] Seq=1 Wi...

Block TCP ACK Ping Scan

Sometimes network admin applies the filter as given below using iptables on TCP ACK packet to drop all ACK packet to established TCP connection with all TCP Port in their network.

```
sudo iptables -I INPUT -p tcp --tcp-flags ALL ACK -j DROP
```

As result, it blocks the NMAP TCP ACK Ping probes so that it could not identify the state of the live host.

```
pentest@ubuntu:~$ sudo iptables -I INPUT -p tcp --tcp-flags ALL ACK -j DROP
pentest@ubuntu:~$
```

Now repeat again TCP ACK Ping for identifying the state of the live host. From given below image you can observe this time it shows that **0 host is up** which means the firewall has blocked packets send by this scan.

```
root@kali:~# nmap -sP -PA 192.168.1.104 --disable-arp-ping
Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-24 03:50 EST
Note: Host seems down. If it is really up, but blocking our ping probes,
Nmap done: 1 IP address (0 hosts up) scanned in 2.12 seconds
root@kali:~#
```

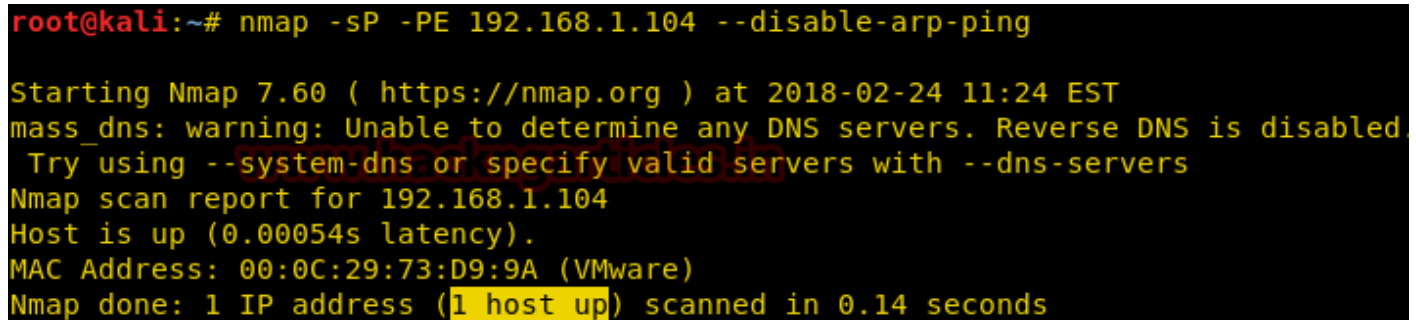
Bypass TCP ACK Ping using ICMP Echo

In some scenarios network, admin apply firewall filter on TCP flag to resist unwanted TCP communication in the network, here let's consider that network admin had blocked TCP communication by applying the filter on SYN as well on ACK flag.

In order to bypass this rule, we'll use ping scan with ICMP packets, for that we'll use **-PE** attribute. **-PE** sends **ICMP echo request** packet [ICMP type 8] and received **ICMP echo reply** packet [ICMP type 0].

```
nmap -sP -PE 192.168.1.104 --disable-arp-ping
```

From given below image you can observe that observe it found **1 Host is up**.



```
root@kali:~# nmap -sP -PE 192.168.1.104 --disable-arp-ping
Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-24 11:24 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled
Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.1.104
Host is up (0.00054s latency).
MAC Address: 00:0C:29:73:D9:9A (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.14 seconds
```

Block ICMP Echo Ping Scan

Usually, most of the network admin apply ICMP filter on their network so that other system or network cannot able to Ping their network.

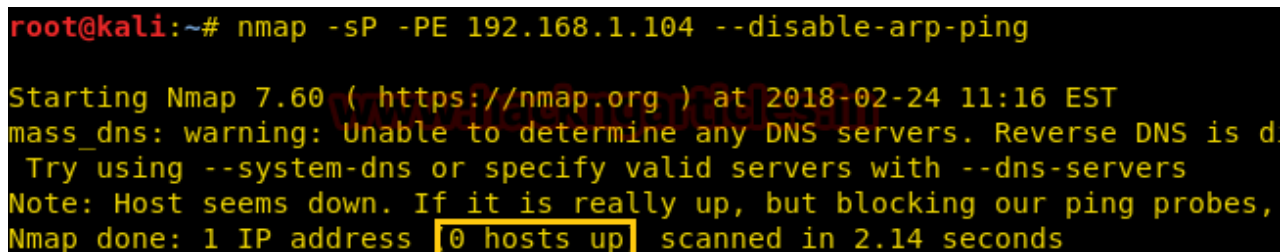
```
sudo iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

As result, it blocks the NMAP ICMP echo Ping probes so that it could not identify the state of the live host.



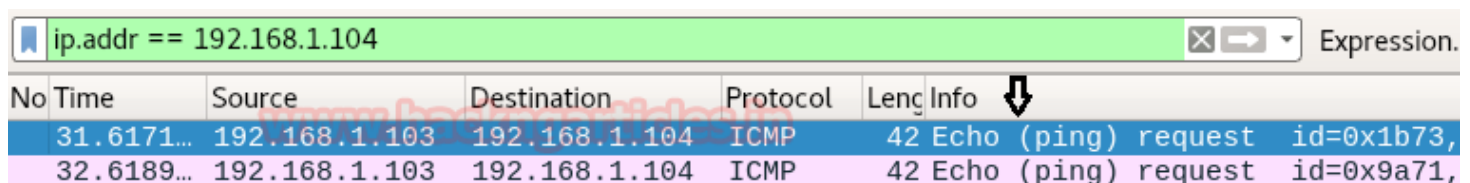
```
sudo iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

Now repeat again TCP ICMP Ping for identifying the state of the live host. From given below image you can observe this time it shows that **0 host is up** which means the firewall has blocked packets send by this scan.



```
root@kali:~# nmap -sP -PE 192.168.1.104 --disable-arp-ping
Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-24 11:16 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is d
Try using --system-dns or specify valid servers with --dns-servers
Note: Host seems down. If it is really up, but blocking our ping probes,
Nmap done: 1 IP address (0 hosts up) scanned in 2.14 seconds
```

Demonstrating NMAP ICMP echo Ping with Wireshark shows only ICMP request packet in the network and didn't receive any reply packet from the host network as shown in the given below image.



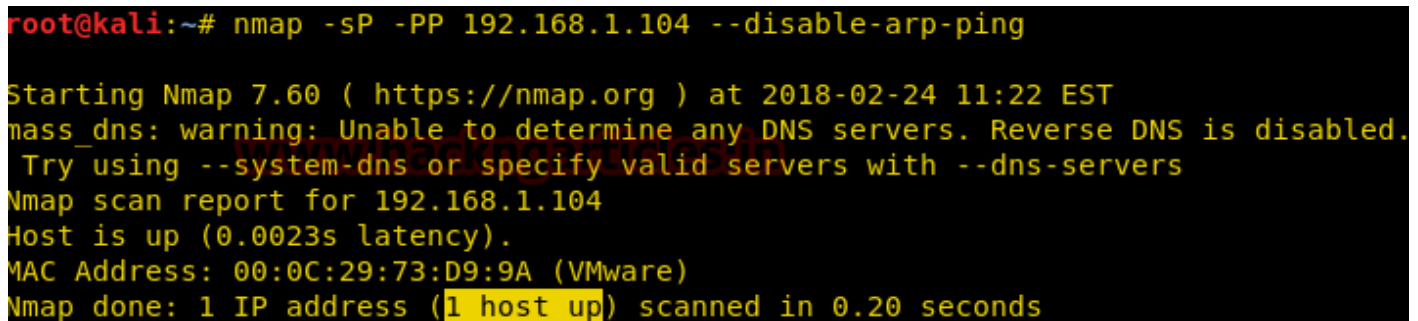
ip.addr == 192.168.1.104							Expression.
No	Time	Source	Destination	Protocol	Length	Info	
31.6171...		192.168.1.103	192.168.1.104	ICMP	42	Echo (ping) request id=0x1b73,	
32.6189...		192.168.1.103	192.168.1.104	ICMP	42	Echo (ping) request id=0x9a71,	

Bypass ICMP Echo Ping using ICMP Timestamp Ping

In order to bypass this rule, we'll use ping scan with ICMP packets, for that we'll use **-PP** attribute. **-PP** sends **ICMP timestamp request** packet [ICMP type 13] and received **ICMP timestamp reply** packet [ICMP type 14].

```
nmap -sP -PP 192.168.1.104 --disable-arp-ping
```

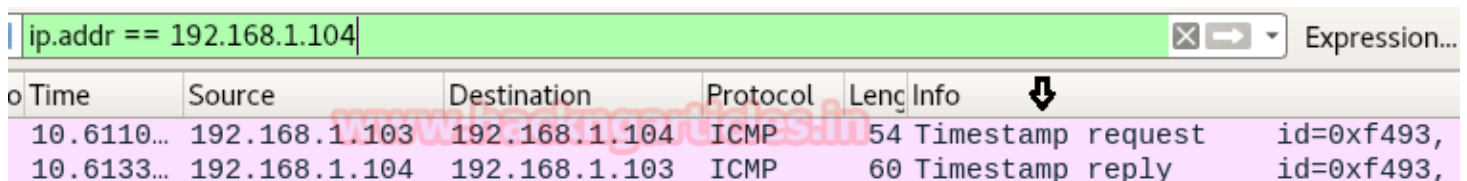
From given below image you can observe that observe it found **1 Host is up**.



```
root@kali:~# nmap -sP -PP 192.168.1.104 --disable-arp-ping

Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-24 11:22 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.1.104
Host is up (0.0023s latency).
MAC Address: 00:0C:29:73:D9:9A (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
```

Demonstrating NMAP ICMP timestamp Ping with Wireshark shows ICMP **timestamp request** packet send in the network and received any **timestamp reply** packet from host network as shown in given below image.



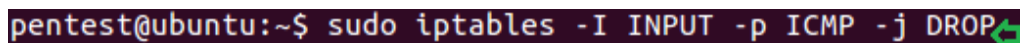
Time	Source	Destination	Protocol	Length	Info
10.6110...	192.168.1.103	192.168.1.104	ICMP	54	Timestamp request id=0xf493,
10.6133...	192.168.1.104	192.168.1.103	ICMP	60	Timestamp reply id=0xf493,

Block ICMP Ping Scan

It might be possible that network admin had block entire types ICMP message by dropping all ICMP packets using following iptables filter.

```
sudo iptables -I INPUT -p ICMP -j DROP
```

As result, it blocks the NMAP ICMP Ping probes so that it could not identify the state of the live host.



```
pentest@ubuntu:~$ sudo iptables -I INPUT -p ICMP -j DROP
```

Now repeat again ICMP Ping either **-PP** or **PE** for identifying the state of the live host. From given below image you can observe this time it shows that **0 host is up** which means the firewall has blocked packets send by this scan.

```
root@kali:~# nmap -sP -PP 192.168.1.104 --disable-arp-ping

Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-24 22:06 EST
Note: Host seems down. If it is really up, but blocking our ping probes,
Nmap done: 1 IP address (0 hosts up) scanned in 2.17 seconds
```

Bypass ICMP Ping Scan using UDP Ping

We have seen multiple ways to check if the system is live. Now, you can determine whether a system is up or not whether it is on the local network or public network.

We had observed that ping scan with ICMP ping is not working or even if TCP packet filter is also enabled in host network then it becomes difficult to identify the live host, now to bypass such types of rules we'll use ping scan with **UDP packets**, for that we'll use **-PU** attribute.

-PU sends UDP packet when no ports are specified, the default is 40125, a reply received ICMP message such as "ICMP destination unreachable" which means the host is live.

```
nmap -sP -PU 192.168.1.104 --disable-arp-ping
```

From given below image you can observe that observe it found **1 Host is up**.

```
root@kali:~# nmap -sP -PU 192.168.1.104 --disable-arp-ping

Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-19 05:30 EST
Nmap scan report for 192.168.1.104
Host is up (0.00040s latency).
MAC Address: 00:0C:29:48:22:FD (VMware)
Nmap done: 1 IP address (1 host up) scanned in 13.11 seconds
```

Demonstrating NMAP UDP Ping with Wireshark shows UDP **request packet** sent on **40125** in the network and received **ICMP destination unreachable** as reply packet from host network as shown in given below image.

Source	Destination	Protocol	Length	Info
192.168.1.103	192.168.1.104	UDP	42	41307 → 40125 Len=0
192.168.1.104	192.168.1.103	ICMP	70	Destination unreachable

Block UDP and Ping Sweep

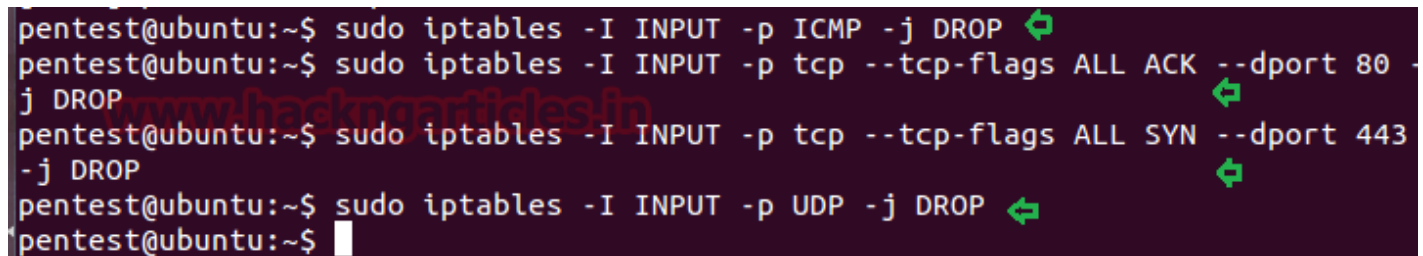
Now let's put some firewall rules in IPTABLES to drop ICMP packets, TCP SYN packets on port 443 and TCP ACK on port 80 which will block Ping Sweep scan as well as Drop UDP packet. Might be network admin had blocked entire TCP packet.


```

sudo iptables -I INPUT -p ICMP -j DROP
sudo iptables -I INPUT -p tcp --tcp-flags ALL ACK --dport 80 -j DROP
sudo iptables -I INPUT -p tcp --tcp-flags ALL SYN --dport 443 -j DROP
sudo iptables -I INPUT -p UDP -j DROP

```

As result, it will resist NMAP for making TCP Ping, ICMP Ping, and UDP ping so that it could not identify the state of the live host.

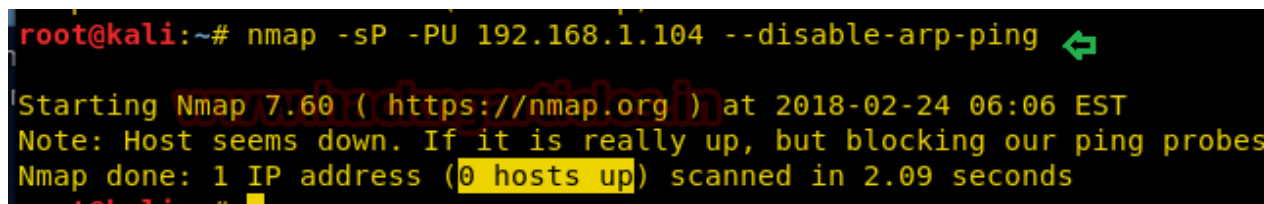


```

pentest@ubuntu:~$ sudo iptables -I INPUT -p ICMP -j DROP
pentest@ubuntu:~$ sudo iptables -I INPUT -p tcp --tcp-flags ALL ACK --dport 80 -j DROP
pentest@ubuntu:~$ sudo iptables -I INPUT -p tcp --tcp-flags ALL SYN --dport 443 -j DROP
pentest@ubuntu:~$ sudo iptables -I INPUT -p UDP -j DROP
pentest@ubuntu:~$

```

Now repeat again UDP Ping for identifying the state of the live host. From given below image you can observe this time it shows that **0 host is up** which means the firewall has blocked packets send by this scan.



```

root@kali:~# nmap -sP -PU 192.168.1.104 --disable-arp-ping
Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-24 06:06 EST
Note: Host seems down. If it is really up, but blocking our ping probes
Nmap done: 1 IP address (0 hosts up) scanned in 2.09 seconds

```

Bypass UDP and Ping Sweep using Protocol Scan

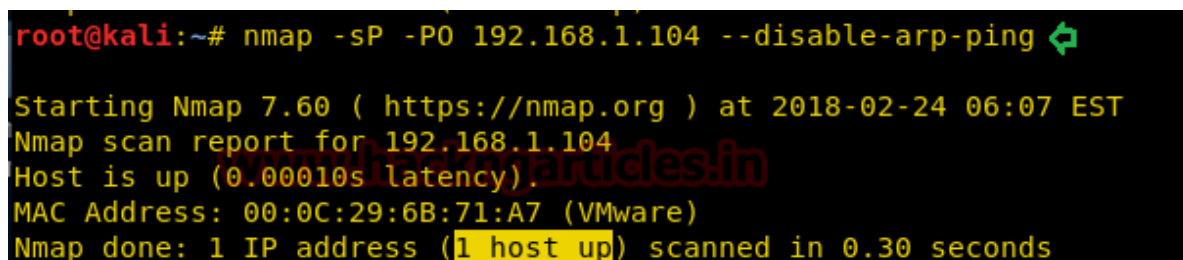
Using Protocol Ping scan we can identify live host when ICMP, TCP, and UDP has been blocked, for that we'll use **-PO attribute**. -PO sends IP packet with the particular protocol number place in their IP header, If no protocols are precise, the default is to send multiple IP packets for ICMP (protocol 1), IGMP (protocol 2), and IP-in-IP (protocol 4).

```

nmap -sP -PO 192.168.1.104 --disable-arp-ping

```

From given below image you can observe that observe it found **1 Host is up**.



```

root@kali:~# nmap -sP -PO 192.168.1.104 --disable-arp-ping
Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-24 06:07 EST
Nmap scan report for 192.168.1.104
Host is up (0.00010s latency).
MAC Address: 00:0C:29:6B:71:A7 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds

```

From given below image of Wireshark we can observe the following mechanism followed by Protocol ping scan.

- Send ICMP Echo to host network
- Send IGMP query to host network
- Send IPv4 (IP-in-IP) to host network
- Received ICMP Destination unreachable as the reply from Host

Source	Destination	Protocol	Length	Info
192.168.1.103	192.168.1.104	ICMP	42	Echo (ping) request id=0..
192.168.1.103	192.168.1.104	IGMPv1	42	Membership Query
192.168.1.103	192.168.1.104	IPv4	34	
192.168.1.104	192.168.1.103	ICMP	62	Destination unreachable (..

Block IP Protocol Ping Scan

Now let's put some firewall rules in iptables to drop ICMP packets, TCP SYN packets on port 443 and TCP ACK on port 80 which will block Ping Sweep scan as well as Drop UDP packet and IP protocol too in the network to prevent the network from any kind of Ping scan. Might be network admin had blocked entire TCP packet.

```
sudo iptables -I INPUT -p ICMP -j DROP
sudo iptables -I INPUT -p tcp --tcp-flags ALL ACK --dport 80 -j DROP
sudo iptables -I INPUT -p tcp --tcp-flags ALL SYN --dport 443 -j DROP
sudo iptables -I INPUT -p UDP -j DROP
sudo iptables -I INPUT -p IP -j DROP
```

As result, it will resist NMAP for making TCP Ping, ICMP Ping, UDP ping and Protocol ping so that it could not identify the state of the live host.

```
pentest@ubuntu:~$ sudo iptables -I INPUT -p ICMP -j DROP
pentest@ubuntu:~$ sudo iptables -I INPUT -p tcp --tcp-flags ALL ACK --dport 80 -j DROP
pentest@ubuntu:~$ sudo iptables -I INPUT -p tcp --tcp-flags ALL SYN --dport 443 -j DROP
pentest@ubuntu:~$ sudo iptables -I INPUT -p UDP -j DROP
pentest@ubuntu:~$ sudo iptables -I INPUT -p IP -j DROP
```

Now repeat again Protocol Ping for identifying the state of the live host. From given below image you can observe this time it shows that **0 host is up** which means the firewall has blocked packets send by this scan.

```
root@kali:~# nmap -sP -P0 192.168.1.104 --disable-arp-ping
Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-24 06:12 EST
Note: Host seems down. If it is really up, but blocking our ping probes,
Nmap done: 1 IP address (0 hosts up) scanned in 3.09 seconds
```

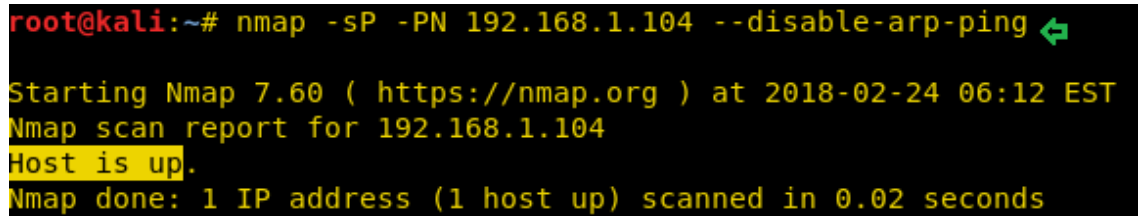
Bypass IP protocol Ping Scan using No Ping Scan

Now when above all Ping scan get failed to identify the state of Host is up or down then we choose the last and best option “No Ping” for we will use **-PN/-P0/-Pn** and basically perform TCP port scan for top 1000 ports.

If you want to prevent Port scan and ping scan use sweep ping with no ping as given below to identify the state of the host is up or down.

```
nmap -sP -PN 192.168.1.104 --disable-arp-ping
```

From given below image you can observe that observe it found **1 Host is up**.

A terminal window screenshot with a black background and yellow text. The prompt is 'root@kali:~#'. The command entered is 'nmap -sP -PN 192.168.1.104 --disable-arp-ping' followed by a green arrow icon. The output shows 'Starting Nmap 7.60 (https://nmap.org) at 2018-02-24 06:12 EST', 'Nmap scan report for 192.168.1.104', 'Host is up.', and 'Nmap done: 1 IP address (1 host up) scanned in 0.02 seconds'.

```
root@kali:~# nmap -sP -PN 192.168.1.104 --disable-arp-ping ↵
Starting Nmap 7.60 ( https://nmap.org ) at 2018-02-24 06:12 EST
Nmap scan report for 192.168.1.104
Host is up.
Nmap done: 1 IP address (1 host up) scanned in 0.02 seconds
```