


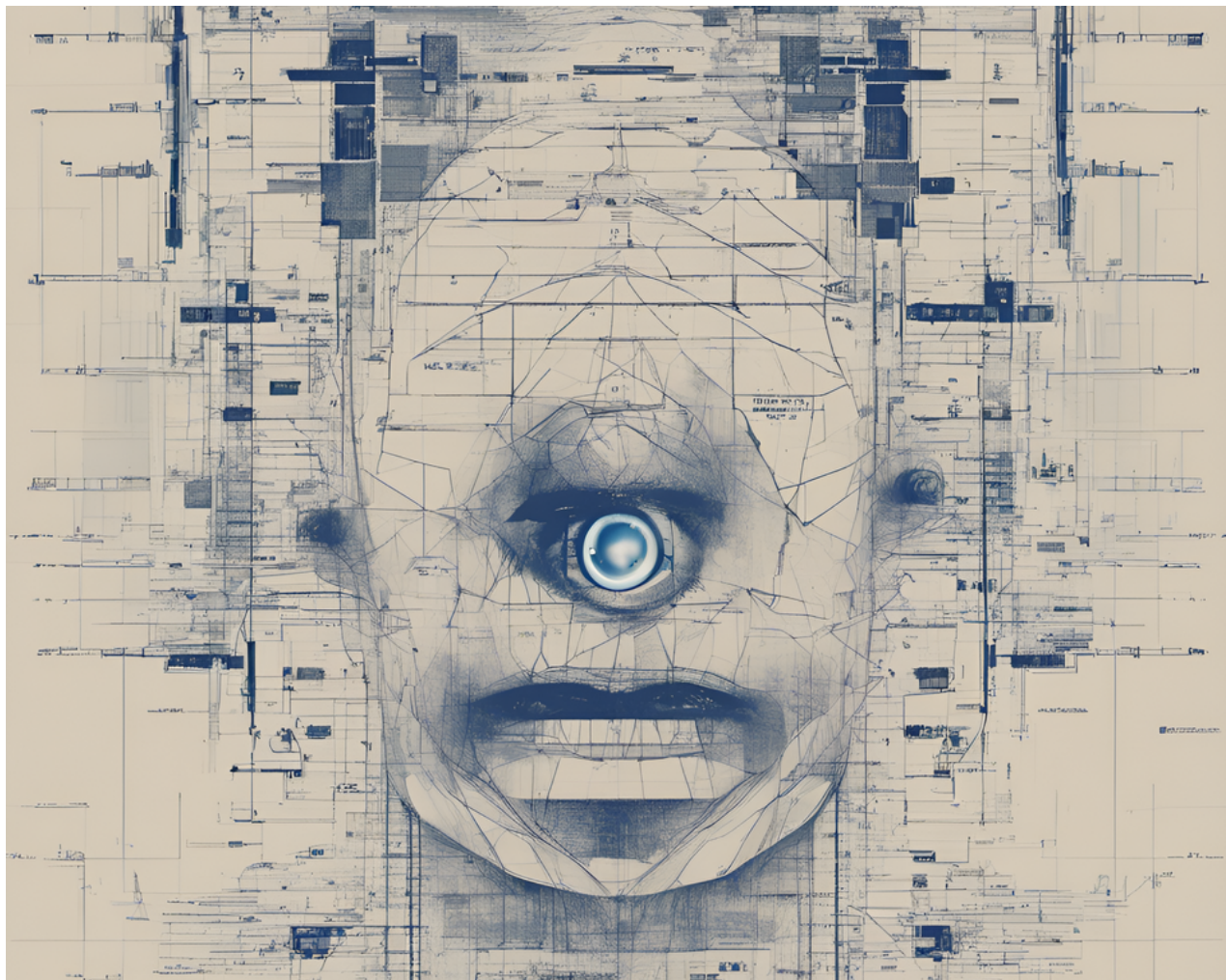
# Do You Really Know Nmap? Think Again! Part 1

 [sud0ru.ghost.io/do-you-really-know-nmap-think-again](https://sud0ru.ghost.io/do-you-really-know-nmap-think-again)

Sud0Ru

March 24, 2024

Mar 24, 2024 19 min read



Do you really think you know Nmap well? After reading this blog series, you might reconsider. Nmap, created by Gordon Lyon, isn't just a simple tool for checking services under specific ports. It employs sophisticated network techniques.

You can find a comprehensive guide to Nmap's capabilities for network discovery and security scanning at <https://nmap.org/book/toc.html>. This guide goes beyond being a mere manual, offering insights into every network technique implemented by Nmap, ranging from host discovery to port scanning and firewall bypass techniques.

In this blog series, I'll delve into all the topics mentioned in the guide and uncover the network secrets it holds. You'll be surprised by the brilliance behind these network techniques.

From the perspective of a red teamer, it's crucial to understand the network traffic generated by Nmap, how to fine-tune its operations, and how to remain as stealthy as possible.

Throughout this series, I'll cover these aspects along with analyzing Nmap's activities using Wireshark to reveal what goes on under the hood. Join me on this intriguing journey.

## The Phases of an Nmap Scan:

---

**A. Script pre-scanning:** Nmap Scripting Engine (NSE) uses specialized scripts to gather remote system information. NSE runs only with options like `--script` or `-sC`. Pre-scanning occurs for scripts needing it, running once per Nmap execution. Examples: `dhcp-discover`, `broadcast-dns-service-discovery`.

**B. Target enumeration:** In this phase, Nmap processes user-provided host specifiers, which can include DNS names, IP addresses, CIDR notations, and more. Nmap resolves these into a list of IPv4 or IPv6 addresses. This phase is crucial and cannot be skipped, but you can simplify it by providing only IP addresses. Using `-sL` `-n` options will print targets without reverse-DNS resolution and perform no further scanning.

**C. Host discovery (ping scanning):** In this phase Nmap discovers which targets on the network are online and thus worth deeper investigation. Nmap offers many host discovery techniques, ranging from quick ARP requests to elaborate combinations of TCP, ICMP, and other types of probes.

**D. Reverse-DNS resolution:** Once Nmap has determined which hosts to scan, it looks up the reverse-DNS names of all hosts found online by the ping scan. Sometimes a host's name provides clues to its function, and names make reports more readable than providing only IP numbers.

**E. Port scanning:** This is Nmap's fundamental operation. Probes are sent, and the responses (or non-responses) to those probes are used to classify remote ports into states such as `open`, `closed`, or `filtered`. Port scanning is performed by default.

**F. Version detection:** If some ports are found to be open, Nmap may be able to determine what server software is running on the remote system. It does this by sending a variety of probes and matching the responses against a database of thousands of known service signature. version detection is not performed by default and you should enable it by the `-sV` option

**G. OS detection.** If requested with the `-O` option, Nmap proceeds to OS detection. Different operating systems implement network standards in subtly different ways. By measuring these differences it is often possible to determine the operating system running on a remote host. Nmap matches responses to a standard set of probes against a database of more than a thousand known operating system responses.

**H. Traceroute:** Nmap contains an optimized traceroute implementation, enabled by the `-traceroute` option. It can find the network routes to many hosts in parallel, using the best available probe packets as determined by Nmap's previous discovery phases. Traceroute usually involves another round of reverse-DNS resolution for the intermediate hosts.

**I. Script scanning:** The Nmap Scripting Engine (NSE) uses a collection of special-purpose scripts to gain even more information about remote systems. NSE is powered by the Lua programming language and a standard library designed for network information gathering. NSE is not executed unless you request it with options such as `--script` or `-sC`. **J. Output:** Finally, Nmap collects all the information it has gathered and writes it to the screen or to a file. Nmap can write output in several formats.

## Host Discovery (“Ping Scanning”)

---

One of the very first steps in any network reconnaissance mission is to reduce a (sometimes huge) set of IP ranges into a list of active or interesting hosts. Scanning every port of every single IP address is slow and usually unnecessary. Of course what makes a host interesting depends greatly on the scan purposes.

Because host discovery needs are so diverse, Nmap offers a wide variety of options for customizing the techniques used. Despite the name ping scan, this goes well beyond the simple ICMP echo request packets associated with the ubiquitous ping tool. Users can skip the ping step entirely with a list scan (`-sL`) or by disabling ping (`-PN`), or engage the network with arbitrary combinations of multi-port TCP SYN/ACK, UDP, and ICMP probes. The goal of these probes is to solicit responses which demonstrate that an IP address is actually active (is being used by a host or network device).

### 1- Specifying Target Hosts and Networks:

As first step you need tell Nmap about which hosts should be scanned, The simplest case is to specify a target IP address or hostname for scanning, or you can use CIDR-style addressing or specifying ranges for each octet, like "192.168.0-255.1-254".

you can use `-iL` option to input from list of hosts/networks, you can also exclude some hosts by `--exclude`, `--excludefile` options or even tell the nmap choose a random host for u.

### 2- DNS Resolution:

You wouldn't date girls just because they're breathing, and selecting boxes on the network to penetrate deserves special care too. A great source of information is DNS

By default, Nmap performs reverse-DNS resolution for every IP which responds to host discovery probes (i.e. those that are online). If host discovery is skipped with `-Pn`, resolution is performed for all IPs. Rather than use the slow standard DNS resolution

libraries, Nmap uses a custom stub resolver which performs dozens of requests in parallel.

Time	Source	Destination	Protocol	Length	Destination Port	Sec Info
1 0.000000000	192.168.129.152	192.168.129.2	DNS	84		Standard query 0xf029 PTR 0.1.168.192.in-addr.arpa
2 0.000165242	192.168.129.152	192.168.129.2	DNS	84		Standard query 0xf02a PTR 1.1.168.192.in-addr.arpa
3 0.000188647	192.168.129.152	192.168.129.2	DNS	84		Standard query 0xf02b PTR 2.1.168.192.in-addr.arpa
4 0.000288738	192.168.129.152	192.168.129.2	DNS	84		Standard query 0xf02c PTR 3.1.168.192.in-addr.arpa
5 0.000381147	192.168.129.152	192.168.129.2	DNS	84		Standard query 0xf02d PTR 4.1.168.192.in-addr.arpa
6 0.000475182	192.168.129.152	192.168.129.2	DNS	84		Standard query 0xf02e PTR 5.1.168.192.in-addr.arpa
7 0.000581125	192.168.129.152	192.168.129.2	DNS	84		Standard query 0xf02f PTR 6.1.168.192.in-addr.arpa
8 0.000782819	192.168.129.152	192.168.129.2	DNS	84		Standard query 0xf030 PTR 7.1.168.192.in-addr.arpa
9 0.001097792	192.168.129.152	192.168.129.2	DNS	84		Standard query 0xf031 PTR 8.1.168.192.in-addr.arpa
10 0.001206118	192.168.129.152	192.168.129.2	DNS	84		Standard query 0xf032 PTR 9.1.168.192.in-addr.arpa
11 0.011900079	192.168.129.2	192.168.129.152	DNS	84		Standard query response 0xf029 No such name PTR 0.1.168.192.in-addr.arpa
12 0.012872842	192.168.129.152	192.168.129.2	DNS	85		Standard query 0xf033 PTR 10.1.168.192.in-addr.arpa

Nmap parallel reverse DNS resolution

From the photo above, it's evident that there is a short duration between DNS requests. and Nmap did not wait for the response before sending the next query.

Worth to note that Nmap does DNS resolution for online hosts only, unless **-R** option specified. You can also disable the parallel DNS resolution by using **--system-dns** option which makes the system do the job (one IP at a time via **getnameinfo** call) in this case the Nmap wait for the response before sending the next query and performance will be slower. As a final note on DNS resolution, if you provide Nmap with hostnames instead of IP addresses, Nmap will begin by performing a standard DNS query like any other application to determine the IP address associated with the hostname. Additionally, Nmap will still conduct reverse DNS resolution for the IP address obtained.

### 3- Host Discovery Controls:

By default, Nmap will include a ping scanning stage prior to more intrusive probes such as port scans, OS detection, Nmap Scripting Engine, or version detection. Nmap usually only performs intrusive scans on machines that are shown to be available during the ping scan stage. This saves substantial time and bandwidth compared to performing full scans against every single IP address.

Maybe you want to check to ensure that you have proper IP addresses for your targets, and do dns resolution without any further operations, in this case you can use list scan (**-sL**) . List scan is a degenerate form of host discovery that simply lists each host on the network(s) specified, without sending any packets to the target hosts.

Also maybe you need not to run a port scan after host discovery, in this case you can use (**-sn**) option. When used by itself, it makes Nmap do host discovery, then print out the available hosts that responded to the scan. This is often called a “ping scan”. Even though no port scanning is done, you can still request Nmap Scripting Engine (**--script**) host scripts and traceroute probing (**--traceroute**). A ping-only scan is one step more intrusive than a list scan, and can often be used for the same purposes. It performs light reconnaissance of a target network quickly and without attracting much attention. Knowing how many hosts are up is more valuable to attackers than the list of every single IP and host name provided by list scan.

The **-sn** option sends an ICMP echo request, a TCP SYN packet to port 443, a TCP ACK packet to port 80, and an ICMP timestamp request by default. Since unprivileged Unix users (or Windows users without Npcap installed) cannot send these raw packets, only



SYN packets are sent instead in those cases. The SYN packet is sent using a TCP **connect** system call to ports 80 and 443 of the target host. When a privileged user tries to scan targets on a local ethernet network, ARP requests (**-PR**) are used unless the **--send-ip** option is specified.

let's start with privileged user, when the privileged user want to use default host discovery techniques against non local ethernet IP address, the traffic will look like the photo below

No.	Time	Source	Destination	Protocol	Length	Destination Port	Sec Info
1	0.000000000	192.168.129.152	192.168.177.153	ICMP	42		Echo (ping) request id=0x5d2b, seq=0/0, ttl=37 (reply in 8)
2	0.000054639	192.168.129.152	192.168.177.153	TCP	58	443	0 51508 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3	0.000451355	192.168.129.152	192.168.177.153	TCP	54	80	1 51508 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
4	0.000527574	192.168.129.152	192.168.177.153	ICMP	54		Timestamp request id=0xc409, seq=0/0, ttl=39
9	0.036238992	192.168.129.152	192.168.129.2	DNS	88		Standard query 0x02b2 PTR 153.177.168.192.in-addr.arpa

Default host discovery probes for privileged user in different broadcast segment

It's evident that Nmap probes were sent to a host with the IP address **192.168.177.153**, which resides in a separate broadcast segment. The sequence of probes initiated includes an ICMP echo request followed by a SYN packet to port 443, an ACK packet to port 80, and finally an ICMP timestamp request.

Now let's see the traffic for local ethernet IP address, assuming we run host discovery against host **192.168.129.1** and our IP is **192.168.129.152**:

No.	Time	Source	Destination	Protocol	Length	Destination Port	Sec Info
1	0.000000000	VMware_de:08:ae	Broadcast	ARP	42		Who has 192.168.129.1? Tell 192.168.129.152
2	0.000582664	VMware_c0:00:08	VMware_de:08:ae	ARP	60		192.168.129.1 is at <span style="border: 1px solid red; padding: 0 2px;">08:00:00:00:00:00</span>

Default host discovery probes for privileged user in local ethernet

we can notices that only ARP request goes from Nmap that asks about the MAC address that's related to host **192.168.129.1**

let's try the same scenario but with unprivileged user:

1	0.000000000	192.168.129.152	192.168.177.153	TCP	74	80	0 56306 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1683396026 TSecr=0 WS=128
2	0.00011943	192.168.129.152	192.168.129.1	TCP	74	443	0 42494 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1683396027 TSecr=0 WS=128
3	0.000660230	192.168.129.1	192.168.129.152	TCP	60	56306	1 80 → 56306 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4	0.001209234	192.168.129.1	192.168.129.152	TCP	60	42494	1 443 → 42494 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Default host discovery probes for unprivileged user in different broadcast segment

we can see only two SYN packets were sent to ports 80 and 443

for the host in the same network segment:

1	0.000000000	192.168.129.152	192.168.129.1	TCP	74	80	0 56306 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1683396026 TSecr=0 WS=128
2	0.00011943	192.168.129.152	192.168.129.1	TCP	74	443	0 42494 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1683396027 TSecr=0 WS=128
3	0.000660230	192.168.129.1	192.168.129.152	TCP	60	56306	1 80 → 56306 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4	0.001209234	192.168.129.1	192.168.129.152	TCP	60	42494	1 443 → 42494 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Default host discovery probes for unprivileged user in local ethernet

The absence of ARP requests is notable due to lack of permission. Additionally, two SYN packets were sent to the remote host in this scenario.

It's worth noting that determining the host's liveness doesn't depend on whether ports are open. Any response from the host, regardless of the port status, signifies its availability. In

the final photo, ports 443 and 80 were closed on the remote host (RST packets came from host), the critical factor at this stage is confirming the host's liveliness, which any response indicates.

Another option is to skip the Nmap discovery stage altogether. Normally, Nmap uses this stage to determine active machines for heavier scanning. By default, Nmap only performs heavy probing such as port scans, version detection, or OS detection against hosts that are found to be up. Disabling host discovery with the `-Pn` option causes Nmap to attempt the requested scanning functions against every target IP address specified.

There are many reasons for disabling the Nmap ping tests. One of the most common is intrusive vulnerability assessments. One can specify dozens of different ping probes in an attempt to elicit a response from all available hosts, but it is still possible that an active yet heavily firewalled machine might not reply to any of those probes. So to avoid missing anything, auditors frequently perform intense scans, such as for all 65,536 TCP ports, against every IP on the target network. It may seem wasteful to send hundreds of thousands of packets to IP addresses that probably have no host listening, and it can slow scan times by an order of magnitude or more. Nmap must send retransmissions to every port in case the original probe was dropped in transit, and Nmap must spend substantial time waiting for responses because it has no round-trip-time (RTT) estimate for these non-responsive IP addresses.

Another frequent reason given for using `-Pn` is that the tester has a list of machines that are already known to be up. So the user sees no point in wasting time with the host discovery stage. This strategy is rarely beneficial from a time-saving perspective. Due to the retransmission and RTT estimate issues discussed in the previous paragraph, even one unresponsive IP address in a large list will often take more time to scan than a whole ping scanning stage would have. In addition, the ping stage allows Nmap to gather RTT samples that can speed up the following port scan, particularly if the target host has strict firewall rules.

In simple terms, pentesters usually turn off the ping scan when they're scanning IPs they already know are active to save time. However, this can end up slowing down the process instead.

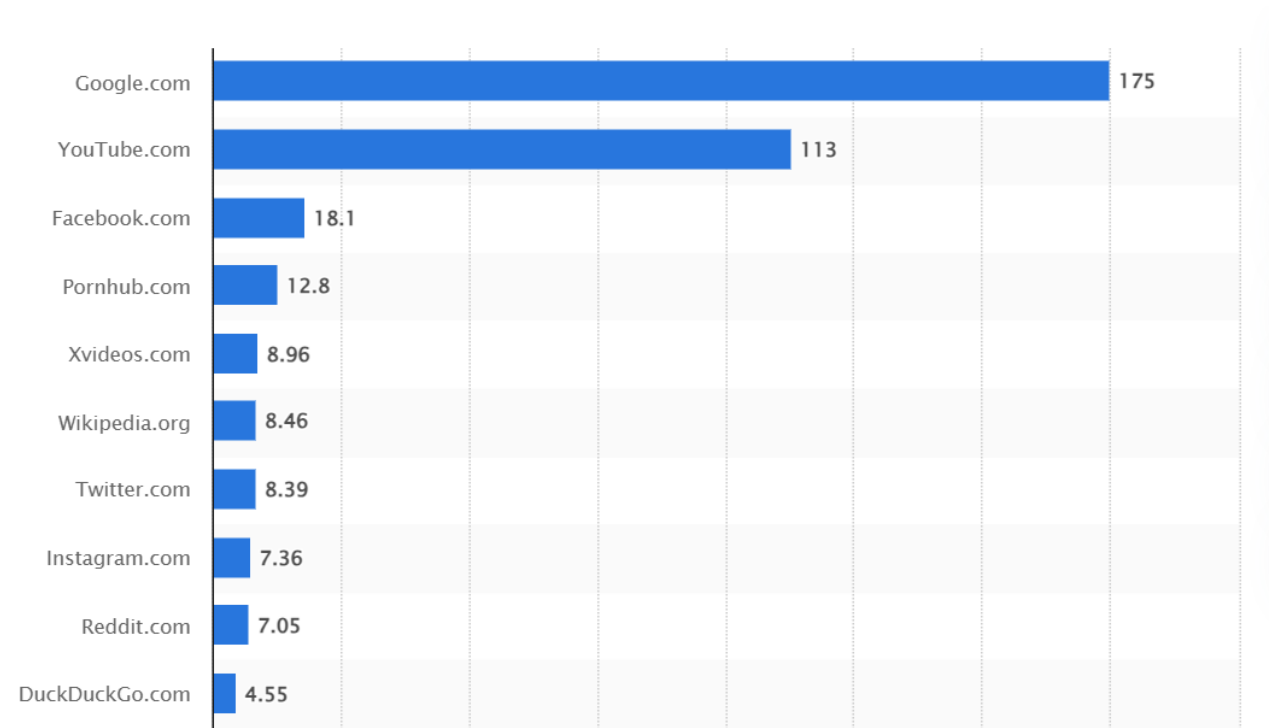
#### **4- Host Discovery Techniques:**

There was a day when finding whether an IP address was registered to an active host was easy. Simply send an ICMP echo request (*ping*) packet and wait for a response. Firewalls rarely blocked these requests, and the vast majority of hosts obediently responded. Such a response has been required since 1989 by [RFC 1122](#), which clearly states that "Every host MUST implement an ICMP Echo server function that receives Echo Requests and sends corresponding Echo Replies".

Unfortunately for network explorers, many administrators have decided that security concerns trump RFC requirements and have blocked ICMP ping messages

Fortunately, Nmap offers a wide variety of host discovery techniques beyond the standard ICMP echo request. They are described in the following sections. Note that if you specify any of the **-P** options discussed in this section, they *replace* the default discovery probes rather than adding to them.

I'll now attempt to show results for regular ping technique (ICMP echo request) by testing it against the top 10 visited websites in November 2023, as reported by [Statista](#).



Most popular websites worldwide as of November 2023, by total visits(*in billions*)

The Nmap output:

```

$ nmap -sn -PE -n google.com youtube.com facebook.com pornhub.com xvideos.com wikipedia.org twitter.com instagram.com reddit.com duckduckgo.com
Starting Nmap 7.94 ( https://nmap.org ) at 2024-03-17 13:10 EDT
Nmap scan report for google.com (142.250.74.174)
Host is up (0.096s latency).
Other addresses for google.com (not scanned): 2a00:1450:400f:805::200e
Nmap scan report for youtube.com (142.250.74.110)
Host is up (0.096s latency).
Other addresses for youtube.com (not scanned): 2a00:1450:400f:803::200e
Nmap scan report for facebook.com (157.240.205.1)
Host is up (0.11s latency).
Other addresses for facebook.com (not scanned): 2a03:2880:f013:0:face:b00c:0:2
Nmap scan report for pornhub.com (66.254.114.41)
Host is up (0.073s latency).
Nmap scan report for wikipedia.org (185.15.59.224)
Host is up (0.078s latency).
Other addresses for wikipedia.org (not scanned): 2a02:ec80:300:ed1a::1
Nmap scan report for twitter.com (104.244.42.1)
Host is up (0.072s latency).
Nmap scan report for instagram.com (157.240.205.174)
Host is up (0.11s latency).
Other addresses for instagram.com (not scanned): 2a03:2880:f213:e4:face:b00c:0:4420
Nmap scan report for reddit.com (151.101.1.140)
Host is up (0.073s latency).
Other addresses for reddit.com (not scanned): 151.101.193.140 151.101.65.140 151.101.129.140 2a04:4e42::396 2a04:4e42:600::396 2a04:4e42:400::396 2a04:4e42:00::396
Nmap scan report for duckduckgo.com (40.114.177.156)
Host is up (0.080s latency).
Nmap done: 10 IP addresses (9 hosts up) scanned in 0.81 seconds

```

Nmap ICMP echo request ping

Nmap using ICMP echo request identified 9 out of 10 hosts as up, completing the scan in 0.81 seconds.

#### 4.1 TCP SYN Ping (**-PS***<port list>*):

The -PS option in Nmap sends an empty TCP packet with the SYN flag set. By default, it targets port 80, but you can specify an alternate port. Multiple ports can be specified, and probes will be sent to each port simultaneously.

When the SYN flag is set, it indicates to the remote system an attempt to establish a connection. Typically, if the destination port is closed, the remote system sends back a RST (reset) packet. If the port is open, the remote system responds with a SYN/ACK packet, initiating the second step of a TCP three-way handshake. However, Nmap immediately terminates this potential connection by responding with a RST, rather than completing the handshake with an ACK packet.

Nmap doesn't distinguish between open or closed ports; both the RST and SYN/ACK responses indicate the host's availability and responsiveness.

On Unix systems, only the privileged user "root" can usually send and receive raw TCP packets. For unprivileged users, Nmap uses a workaround where it initiates a connect system call to each target port. This effectively sends a SYN packet to the target host in an attempt to establish a connection. If the connect call returns quickly or with an ECONNREFUSED failure, indicating receipt of a SYN/ACK or RST, the host is marked as available. If the connection attempt times out, the host is marked as down. This workaround is also applied for IPv6 connections, as Nmap lacks raw IPv6 packet building support.

Now let's conduct the same scan against the previous websites

```
└─$ nmap -sn -PS443 -n google.com youtube.com facebook.com pornhub.com xvideos.com wikipedia.org twitter.com instagram.com reddit.com duckduckgo.com
Starting Nmap 7.94 ( https://nmap.org ) at 2024-03-17 13:37 EDT
Nmap scan report for google.com (142.250.74.174)
Host is up (0.12s latency).
Other addresses for google.com (not scanned): 2a00:1450:400f:805::200e
Nmap scan report for youtube.com (142.250.74.110)
Host is up (0.12s latency).
Other addresses for youtube.com (not scanned): 2a00:1450:400f:803::200e
Nmap scan report for facebook.com (157.240.205.1)
Host is up (0.14s latency).
Other addresses for facebook.com (not scanned): 2a03:2880:f013:0:face:b00c:0:2
Nmap scan report for pornhub.com (66.254.114.41)
Host is up (0.094s latency).
Other addresses for pornhub.com (not scanned): 185.88.181.11
Nmap scan report for xvideos.com (185.88.181.11)
Host is up (0.11s latency).
Other addresses for xvideos.com (not scanned): 185.88.181.6 185.88.181.2 185.88.181.4 185.88.181.5 185.88.181.7 185.88.181.8 185.88.181.3 185.88.181.9 185.88.181.10
Nmap scan report for wikipedia.org (185.15.59.224)
Host is up (0.11s latency).
Other addresses for wikipedia.org (not scanned): 2a02:ec80:300:ed1a::1
Nmap scan report for twitter.com (104.244.42.129)
Host is up (0.094s latency).
Nmap scan report for instagram.com (157.240.205.174)
Host is up (0.12s latency).
Other addresses for instagram.com (not scanned): 2a03:2880:f213:e4:face:b00c:0:4420
Nmap scan report for reddit.com (151.101.65.140)
Host is up (0.094s latency).
Other addresses for reddit.com (not scanned): 151.101.129.140 151.101.1.140 151.101.193.140 2a04:4e42::396 2a04:4e42:200::396 2a04:4e42:600::396 2a04:4e42:400::396
Nmap scan report for duckduckgo.com (40.114.177.156)
Host is up (0.11s latency).
Nmap done: 10 IP addresses (10 hosts up) scanned in 0.29 seconds
```

### TCP SYN ping

in addition to detecting all 10 websites, the second run is much faster because the machines are scanned in parallel and the scan never times out waiting for a response. This test is not entirely fair because these are all popular web servers and thus can be expected to listen on TCP port 443. However, it still demonstrates the point that different types of hosts respond to different probe types. Nmap supports the usage of many scan types in parallel to enable effective scanning of diverse networks.



## 4.2 TCP ACK Ping (-PA<port list>):

The TCP ACK ping is quite similar to the SYN ping. The difference, as you could likely guess, is that the TCP ACK flag is set instead of the SYN flag. Such an ACK packet purports to be acknowledging data over an established TCP connection, but no such connection exists. So remote hosts should always respond with a RST packet, disclosing their existence in the process.

The -PA option uses the same default port as the SYN probe (80) and can also take a list of destination ports in the same format. If an unprivileged user tries this, or an IPv6 target is specified, the connect workaround discussed previously is used. This workaround is imperfect because connect is actually sending a SYN packet rather than an ACK.

The reason for offering both SYN and ACK ping probes is to maximize the chances of bypassing firewalls. Many administrators configure routers and other simple firewalls to block incoming SYN packets except for those destined for public services like the company web site or mail server. This prevents other incoming connections to the organization, while allowing users to make unobstructed outgoing connections to the Internet. This non-stateful approach takes up few resources on the firewall/router and is widely supported by hardware and software filters.

When firewall rules such as this are in place, SYN ping probes (-PS) are likely to be blocked when sent to closed target ports. In such cases, the ACK probe excels by cutting right through these rules.

Another common type of firewall uses stateful rules that drop unexpected packets. This feature was initially found mostly on high-end firewalls, though it has become much more common over the years.

The ACK probe is unlikely to work against firewalls taking this approach, as such an unexpected packet will be classified in the INVALID state and probably dropped

Let's try ACK ping on windows machine. I will try it against open port (TCP 88) and closed one (TCP 80) on both cases when the windows firewall is turned off and turned on, and I will compare the response. The target Host is run windows server 2019 with default configuration (default windows firewall settings)

The two Nmap commands that used:

```
sudo nmap -sn -PA -n 192.168.177.153 --send-ip
```

-sn: meaning just do ping scan and stop

-n: for no DND resolution

-PA : ACK ping to port 80

--send-ip: treat the host as remote host not as host in the local ethernet segment; I will talk about this option later when we are talking about ARP ping

```
sudo nmap -sn -PA88 -n 192.168.177.153 --send-ip
```

The same but here we send the ACK packet to port 88

first trial when the firewall is on:

No.	Time	Source	Destination	Protocol	Length	Destination Port	Sec Info
1	0.000000000	192.168.177.2	192.168.177.153	TCP	54	88	1 57464 → 88 [ACK] Seq=1 Ack=1 Win=1024 Len=0
2	1.002105892	192.168.177.2	192.168.177.153	TCP	54	88	1 57466 → 88 [ACK] Seq=1 Ack=1 Win=1024 Len=0
3	3.975644625	192.168.177.2	192.168.177.153	TCP	54	80	1 35774 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
4	4.976786842	192.168.177.2	192.168.177.153	TCP	54	80	1 35776 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0

ACK ping against firewalled windows machine

We can notice there is no response from the target host for the both ports (80,88)

Second when the firewall is off:

No.	Time	Source	Destination	Protocol	Length	Destination Port	Sec Info
3	1.634478992	192.168.177.2	192.168.177.153	TCP	54	88	1 46648 → 88 [ACK] Seq=1 Ack=1 Win=1024 Len=0
4	1.635193411	192.168.177.153	192.168.177.2	TCP	60	46648	1 88 → 46648 [RST] Seq=1 Win=0 Len=0
9	5.982181470	192.168.177.2	192.168.177.153	TCP	54	80	1 33408 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
10	5.982863821	192.168.177.153	192.168.177.2	TCP	60	33408	1 80 → 33408 [RST] Seq=1 Win=0 Len=0

ACK ping against windows machine with no firewall

we can see that after turned off the windows firewall the we have RST packets from the both ports indicating that the host is a live

ACK packets it's very interesting not only for host discovery but usually it used to determine if there is firewall or not

for now I won't go deeper and explain the details of using ACK for firewall testing. I'll cover it later because it requires creating raw packets and compare different header in addition to use other tools like hping3.

#### 4.3 UDP Ping (-PU<port list>):

Another option for host discovery is UDP ping, which sends a UDP packet to specified ports. The default ports, if not specified, are 40,125 but can be changed at compile-time. This default is chosen to avoid sending to open ports

When hitting a closed port, Nmap expects an ICMP port unreachable packet in return, signifying the host is up. Other ICMP errors or lack of response indicate a down or unreachable host. If an open port is reached, most services ignore the empty packet, making the default probe port highly unlikely to be in use

let's try to do the same thing that we did with TCP ACK ping, let's try to do it with firewall and without

The nmap command that I want to use:

```
sudo nmap -sn -PU -n 192.168.177.153 --send-ip
```

-PU: stands for UDP ping

With firewall

No.	Time	Source	Destination	Protocol	Length	Destination Port	Sec Info
1	0.000000000	192.168.177.2	192.168.177.153	UDP	82		33886 → 40125 Len=40
2	1.001492241	192.168.177.2	192.168.177.153	UDP	82		33888 → 40125 Len=40

UPD ping against firewalled windows machine

we can see as expected there is no response

let's try it without firewall:

No.	Time	Source	Destination	Protocol	Length	Destination Port	Sec Info
2	1.082398541	192.168.177.2	192.168.177.153	UDP	82	82	61683 → 40125 Len=40
3	1.082731253	192.168.177.153	192.168.177.2	ICMP	110		Destination unreachable (Port unreachable)

UDP ping against windows machine without firewall.

We can see that the host responded with an ICMP message indicating that the port is unreachable. Nmap will successfully identify that the host is active

This scan type's primary advantage is bypassing firewalls and filters TCP only.

#### 4.4 ICMP Ping Types (-PE, -PP, and -PM):

In addition to the unusual TCP and UDP host discovery types discussed previously, Nmap can send the standard packets sent by the ubiquitous ping program. Nmap sends an ICMP type 8 (echo request) packet to the target IP addresses, expecting a type 0 (echo reply) in return from available hosts. As noted at the beginning of this chapter, many hosts and firewalls now block these packets, rather than responding as required by [RFC 1122](#). For this reason, ICMP-only scans are rarely reliable enough against unknown targets over the Internet. But for system administrators monitoring an internal network, this can be a practical and efficient approach. Use the -PE option to enable this echo request behavior.

While echo request is the standard ICMP ping query, Nmap does not stop there. The ICMP standards ([RFC 792](#) and [RFC 950](#)) also specify timestamp request, information request, and address mask request packets as codes 13, 15, and 17, respectively. While the ostensible purpose for these queries is to learn information such as address masks and current times, they can easily be used for host discovery. Nmap does not currently implement information request packets, as they are not widely supported (RFC 1122 insists that “a host SHOULD NOT implement these messages”). Timestamp and address mask queries can be sent with the -PP and -PM options, respectively. A timestamp reply (ICMP code 14) or address mask reply (code 18) discloses that the host is available. These two queries can be valuable when administrators specifically block echo request packets, but forget that other ICMP queries can be used for the same purpose.

Now let's try these three types against our test host, first let's start with firewall:

No.	Time	Source	Destination	Protocol	Len	De	Sec Info
125	87.383888677	192.168.177.2	192.168.177.153	ICMP	42		Echo (ping) request id=0x8074, seq=0/0, ttl=59 (reply in 130)
126	87.384168165	192.168.177.2	192.168.177.153	ICMP	54		Timestamp request id=0xea4b, seq=0/0, ttl=44
127	87.384336739	192.168.177.2	192.168.177.153	ICMP	46		Address mask request id=0x16a8, seq=0/0, ttl=51
130	87.385799201	192.168.177.153	192.168.177.2	ICMP	60		Echo (ping) reply id=0x8074, seq=0/0, ttl=128 (request in 125)

ICMP ping against firewalled windows machine

we can see that the host responded to the echo request only.

Now let's turn off the firewall:

No.	Time	Source	Destination	Protocol	Length	Destination Port	Sec Info
3	1.758183545	192.168.177.111	192.168.177.153	ICMP	42		Echo (ping) request id=0xac66, seq=0/0, ttl=44 (reply in 5)
4	1.758467596	192.168.177.111	192.168.177.153	ICMP	54		Timestamp request id=0x2434, seq=0/0, ttl=39
5	1.758628679	192.168.177.153	192.168.177.111	ICMP	60		Echo (ping) reply id=0xac66, seq=0/0, ttl=128 (request in 3)
6	1.758632810	192.168.177.153	192.168.177.111	ICMP	60		Timestamp reply id=0x2434, seq=0/0, ttl=128
7	1.758677556	192.168.177.111	192.168.177.153	ICMP	46		Address mask request id=0x3856, seq=0/0, ttl=46

ICMP ping against windows machine without firewall.

we can see in addition to ICMP echo request, the host responded to ICMP timestamp request.

#### 4.5 IP Protocol Ping (-P0<protocol list>):

The newest host discovery option is the IP protocol ping, which sends IP packets with the specified protocol number set in their IP header. The protocol list takes the same format as do port lists in the previously discussed TCP and UDP host discovery options. If no protocols are specified, the default is to send multiple IP packets for ICMP (protocol 1), IGMP (protocol 2), and IP-in-IP (protocol 4). The default protocols can be configured at compile-time. Note that for the ICMP, IGMP, TCP (protocol 6), and UDP (protocol 17), the packets are sent with the proper protocol headers while other protocols are sent with no additional data beyond the IP header (unless the --data-length option is specified).

This host discovery method looks for either responses using the same protocol as a probe, or ICMP protocol unreachable messages which signify that the given protocol isn't supported by the destination host. Either type of response signifies that the target host is alive.

Again let's try it against our test host:

No.	Time	Source	Destination	Protocol	Length	Destination Port	Sec Info
1	0.000000000	192.168.177.2	192.168.177.153	ICMP	42		Echo (ping) request id=0xd8ea, seq=0/0, ttl=55 (reply in 3)
2	0.000287948	192.168.177.2	192.168.177.153	IGMPv1	42		Membership Query id=0xd8ea, seq=0/0, ttl=128 (request in 1)
3	0.000601269	192.168.177.153	192.168.177.2	ICMP	60		Echo (ping) reply id=0xd8ea, seq=0/0, ttl=128 (request in 1)
4	0.000743494	192.168.177.2	192.168.177.153	IPv4	34		

IP protocol ping against firewalled windows machine

we can see the host response to ICMP echo request only.

Now let's try after we turned off the firewall:

Time	Source	Destination	Protocol	Length	Destination Port	Sec Info
1	0.000000000	192.168.177.111	192.168.177.153	ICMP	42	Echo (ping) request id=0xd330, seq=0/0, ttl=43 (reply in 2)
2	0.001692897	192.168.177.153	192.168.177.111	ICMP	60	Echo (ping) reply id=0xd330, seq=0/0, ttl=128 (request in 1)
3	0.005359694	192.168.177.111	192.168.177.153	IGMPv1	42	Membership Query
4	0.005514155	192.168.177.111	192.168.177.153	IPv4	34	
5	0.005735613	192.168.177.153	192.168.177.111	ICMP	62	Destination unreachable (Protocol unreachable)

IP protocol against windows machine without firewall.



```

Frame 5: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface eth0, id 0
Ethernet II, Src: VMware_10:d9:a2 (00:0c:29:10:d9:a2), Dst: VMware_de:08:ae (00:0c:29:de:08:ae)
Internet Protocol Version 4, Src: 192.168.177.153, Dst: 192.168.177.111
Internet Control Message Protocol
  Type: 3 (Destination unreachable)
  Code: 2 (Protocol unreachable)
  Checksum: 0xfcfd [correct]
  [Checksum Status: Good]
  Unused: 00000000
Internet Protocol Version 4, Src: 192.168.177.111, Dst: 192.168.177.153
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 20
  Identification: 0x36de (14046)
  000. .... = Flags: 0x0
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 53
  Protocol: IPIP (4)
  Header Checksum: 0x6aae [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 192.168.177.111
  Destination Address: 192.168.177.153

```

ICMP protocol unreachable packet.

We can notice that the host responded to the ICMP echo request as usual but in this time it also responded to IPIP packet by sent ICMP protocol unreachable packet.

## 4.6 ARP Scan (-PR):

Nmap commonly scans local Ethernet LANs. When sending raw IP packets, the OS must resolve destination hardware addresses (MAC) through ARP requests, causing delays if hosts don't respond promptly. ARP scanning, which bypasses the OS ARP cache, resolves this issue and is faster and more accurate than raw IP scanning. Nmap defaults to ARP scanning for local Ethernet networks, improving efficiency and accuracy. Additionally, ARP scanning allows Nmap to control the source MAC address, useful for maintaining anonymity in certain scenarios

now let's see example when you try to scan local ethernet host. I'm connected to network (192.168.177.0/24) and I will scan 192.168.177.153 by running the nmap without any additional flags:

Time	Source	Destination	Protocol	Length	Destination Port	Sec Info
1 0.000000000	VMware_de:08:ae	Broadcast	ARP	42		who has 192.168.177.153? Tell 192.168.177.111
2 0.000455924	VMware_10:d9:a2	VMware_de:08:ae	ARP	60		192.168.177.153 is at 00:0c:29:10:d9:a2

ARP scan against windows machine

```

# nmap -sn -n 192.168.177.153
Starting Nmap 7.94 ( https://nmap.org ) at 2024-03-21 04:23 EDT
Nmap scan report for 192.168.177.153
Host is up (0.00052s latency).
MAC Address: 00:0C:29:10:D9:A2 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.14 seconds

```

NMAP host scan output for local ethernet host

from above two photos Nmap identified successfully that the host is part of local ethernet and sent ARP request and got the response.

So now let's try to use `--send-ip` option:

Time	Source	Destination	Protocol	Length	Destination Port	Sec Info
1 0.000000000	Intel_44:aa:48	Broadcast	ARP	60		Who has 10.121.122.254? Tell 10.121.122.146
2 1.174749918	192.168.177.111	192.168.177.153	ICMP	42		Echo (ping) request id=0x0300, seq=0/0, ttl=59 (reply in 6)
3 1.175093645	192.168.177.111	192.168.177.153	TCP	58	443	0 57489 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4 1.175288141	192.168.177.111	192.168.177.153	TCP	54	80	1 57489 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
5 1.175421360	192.168.177.111	192.168.177.153	ICMP	54		Timestamp request id=0x58a1, seq=0/0, ttl=48
6 1.175913354	192.168.177.153	192.168.177.111	ICMP	60		Echo (ping) reply id=0x0300, seq=0/0, ttl=128 (request in 2)

Network traffic when `--send-ip` option used

```
# nmap -sn -n --send-ip 192.168.177.153
Starting Nmap 7.94 ( https://nmap.org ) at 2024-03-21 04:26 EDT
Nmap scan report for 192.168.177.153
Host is up (0.00055s latency).
MAC Address: 00:0C:29:10:D9:A2 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

NMAP host scan output for local ethernet host with `--send-ip` option

In the second example, we use the `--send-ip` argument with NMAP. This instructs NMAP to treat the host as a non-local IP address. As a result, we encounter the usual host scanning probes.

#### 4.7 SCTP Ping (-PY):

PY (SCTP INIT Ping) sends SCTP packets with minimal INIT chunks. Default port is 80. Multiple probes sent in parallel. INIT chunk suggests association establishment. Closed ports trigger ABORT chunk; open ports, INIT-ACK chunk. Nmap doesn't distinguish between open or closed ports, The host's response confirms its availability; ABORT/INIT-ACK signals host availability. Unix requires root privilege for sending/receiving raw SCTP packets. Unprivileged users cannot use SCTP INIT Pings.

let's try it with the remote host with firewall enabled:

No.	Time	Source	Destination	Protocol	Length	Destination Port	Sec Info
3 0.867025941	192.168.177.111	192.168.177.153	SCTP	66		INIT	
6 1.869262235	192.168.177.111	192.168.177.153	SCTP	66		INIT	

SCTP ping against firewalled windows machine

we can see that there is no response from the host

Now let's try it after we turned off the firewall:

Time	Source	Destination	Protocol	Length	Destination Port	Sec Info
2 0.749289976	192.168.177.111	192.168.177.153	SCTP	66		INIT
3 0.749698259	192.168.177.153	192.168.177.111	ICMP	94		Destination unreachable (Protocol unreachable)

SCTP against windows machine without firewall

we can notice that the host responded with ICMP protocol unreachable and that's enough for NMAP to guess that the system is alive.

At this point, I will conclude this section because I don't want it to become too lengthy. Even though this part of host discovery isn't finished yet, there are still interesting things to explore. Let's save them for the next part. Until then, happy deep learning 🐱, and I hope you enjoyed it!