Nmap (“Network Mapper”) is a free and open-source utility for network exploration and security auditing. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, and what type of packet filters/firewalls are in use.

The first step in a vulnerability assessment is network discovery. This reconnaissance stage determines what IP address ranges the target is using, what hosts are available, what services those hosts are offering, general network topology details, and what firewall/filtering policies are in effect.

Scans proceed in phases, with each phase finishing before the next one begins. As you can see from the phase descriptions below, there is far more to Nmap than just port scanning.

Target enumeration. In this phase, Nmap researches the host specifiers provided by the user, which may be a combination of host DNS names, IP addresses, CIDR network notations, and more. You can even use (-iR) to ask Nmap to choose your targets for you! Nmap resolves these specifiers into a list of IPv4 or IPv6 addresses for scanning.

Network scans usually begin by discovering which targets on the network are online and thus worth deeper investigation. This process is called host discovery or ping scanning. Nmap offers many host discovery techniques, ranging from quick ARP requests to elaborate combinations of TCP, ICMP, and other types of probes.

Reverse-DNS. 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. This step may be skipped with the -n (no resolution) option.

Port scanning. This is Nmap's core 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.

Version detection. If any ports are found to be open, Nmap may be able to determine what server software is running on the remote system. It does this by sending a variety of probes to the open ports and matching any responses against a database of thousands of more than 6,500 known service signatures. Version detection is enabled with the -sV option

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.

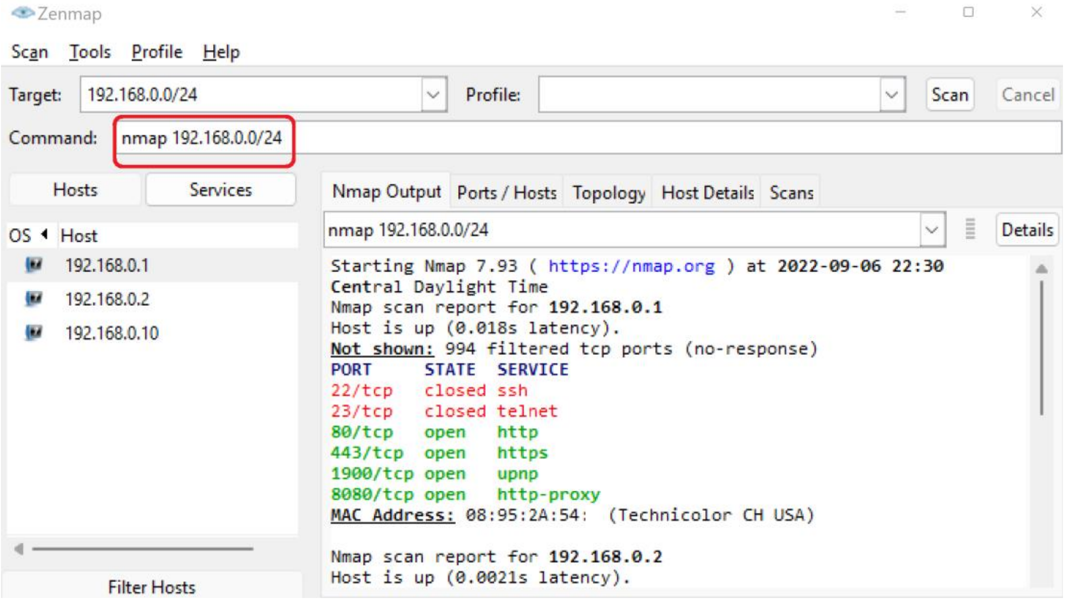
Traceroute. Nmap contains an optimized traceroute implementation, enabled by the --traceroute option. It can find the network routes to many hosts in parallel, using the best available probe packets as determined by Nmap's previous discovery phases. Traceroute usually involves another round of reverse-DNS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

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. They commonly perform tasks such as detecting service vulnerabilities, malware discovery, collecting more information from databases and other network services, and advanced version detection

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.

An administrator may be comfortable using just an ICMP ping to locate hosts on his internal network, while an external penetration tester may use a diverse set of dozens of probes in an attempt to evade firewall restrictions. Nmap offers many ping techniques because it often takes carefully crafted combinations to get through a series of firewalls and router filters leading to a target network.

Sometimes you wish to scan a whole network of adjacent hosts. For this, Nmap supports CIDR-style addressing. For example, 192.168.0.0/24 would scan the 256 hosts between 192.168.0.0 and 192.168.0.255



Target: 192.168.0.0/24

Command: nmap 192.168.0.0/24

OS	Host
	192.168.0.1
	192.168.0.2
	192.168.0.10

nmap 192.168.0.0/24

```
Starting Nmap 7.93 ( https://nmap.org ) at 2022-09-06 22:30
Central Daylight Time
Nmap scan report for 192.168.0.1
Host is up (0.018s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE  SERVICE
22/tcp    closed ssh
23/tcp    closed telnet
80/tcp    open   http
443/tcp   open   https
1900/tcp  open   upnp
8080/tcp  open   http-proxy
MAC Address: 08:95:2A:54:  (Technicolor CH USA)

Nmap scan report for 192.168.0.2
Host is up (0.0021s latency).
```

The specification scanme.nmap.org/16 would scan the 65,536 IP addresses between 64.13.0.0 and 64.13.255.255. The smallest allowed value is /0, which targets the whole Internet. The largest value is /32, which scans just the named host or IP address because all address bits are fixed.

Nmap supports this through octet range addressing. Rather than specify a normal IP address, you can specify a comma-separated list of numbers or ranges for each octet. For example, 192.168.0-255.1-254 will skip all addresses in the range that end in .0 or .255.

Ranges need not be limited to the final octets: the specifier 0-255.0-255.13.37 will perform an Internet-wide scan for all IP addresses ending in 13.37. This sort of broad sampling can be useful for Internet surveys and research. IPv6 addresses can only be specified by their fully qualified IPv6 address or hostname.

You can generate a list of hosts to scan and pass that filename to Nmap as an argument via the -iL option. Entries can be in any of the formats accepted by Nmap on the command line (IP address, hostname, CIDR, IPv6, or octet ranges). Each entry must be separated by one or more spaces, tabs, or newlines. You can specify a hyphen (-) as the filename if you want Nmap to read hosts from standard input rather than an actual file.

You can exclude hosts or entire networks with the --exclude option. Simply pass the option a comma-separated list of excluded targets and netblocks using the normal Nmap syntax. Alternatively, you can create a file of excluded hosts/networks and pass that to Nmap with the --excludefile option.

Nmap offers a dry run using the list scan (-sL option). Simply execute nmap -sL -n <targets> to see which IPs would be scanned before you actually do it.

> *nmap 64.13.134.52/24 --exclude scanme.nmap.org,insecure.org*
> Tells Nmap to scan the class C around 64.13.134.52, but to skip scanme.nmap.org and insecure.org if they are found within that address range.
>
> *nmap 10.0.0.0/8 --exclude 10.6.0.0/16,ultra-sensitive-host.company.com*
> Tells Nmap to scan the whole private 10 range except that it must skip anything starting with 10.6 as well as ultra-sensitive-host.company.com.
>
> *egrep '^lease' /var/lib/dhcp/dhcpd.leases | awk '{print $2}' | nmap -iL -*
> Obtain the list of assigned DHCP IP addresses and feed them directly to Nmap for scanning. Note that a hyphen is passed to -iL to read from standard input.

Web databases can also be used to find hostnames under a given domain. For example, Netcraft has a web site DNS search feature at http://searchdns.netcraft.com/?host. Google can also be used for this purpose with queries such as site:target.com.

After a set of initial “seed” IPs are discovered, they must be researched to ensure they belong to the company you expect and to determine what netblocks they are part of. Small and mid-sized companies normally don't have IP space allocated by the likes of ARIN. Instead, they are delegated netblocks from their ISPs. Fortunately, many ISPs now subdelegate customer ranges using Shared Whois (SWIP) or Referral Whois (RWhois). If the ISP has done this, you learn the customer's exact netblock size.

The core routing protocol of the Internet is the Border Gateway Protocol (BGP). When scanning mid-sized and large organizations, BGP routing tables can help you find their IP subnets all over the world.

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. While the defaults generally work well, Nmap offers four options for controlling DNS resolution.

> -n (No DNS resolution)
> Tells Nmap to never do reverse DNS resolution on the active IP addresses it finds.
>
> -R (DNS resolution for all targets)
> Tells Nmap to always do reverse DNS resolution on the target IP addresses.
>
> --system-dns (Use system DNS resolver)
> By default, Nmap resolves IP addresses by sending queries directly to the name servers configured on your host and then listening for responses. Specify this option to use your system resolver instead.
>
> --dns-servers <server1>[,<server2>[,...]]
> By default, Nmap determines your DNS servers (for rDNS resolution) from your resolv.conf file (Unix) or the Registry (Win32). Alternatively, you may use this option to specify alternate servers. This option can also improve stealth, as your requests can be bounced off just about any recursive DNS server on the Internet.

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.

(-sL) List scan is a good sanity check to ensure that you have proper IP addresses for your targets. If the hosts sport domain names you do not recognize, it is worth investigating further to prevent scanning the wrong company's network.

Another reason for an advance list scan is stealth. In some cases, you do not want to begin with a full-scale assault on the target network that is likely to trigger IDS alerts and bring unwanted attention. A list scan is unobtrusive and provides information that may be useful in choosing which individual machines to target.

(-sP) This option tells Nmap to only perform a ping scan, then print out the available hosts that responded to the scan. No further testing (such as port scanning or OS detection) is performed. This is one step more intrusive than a list scan, and can often be used for the same purposes. It performs light re connai ssance of a target network quickly and without attrac ting 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.

(-PN) 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 option causes Nmap to attempt the requested scanning functions against -PN 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.

While specifying -PN is rarely helpful as a time saver, it is important if some of the machines on your list block all of the discovery techniques that would otherwise be specified. Users must strike a balance between scan speed and the possibility of missing heavily cloaked machines.

The -PS option sends an empty TCP packet with the SYN flag set. The default destination port is 80, but an alternate port can be specified as a parameter. A list of ports may be specified, in which case probes will be attempted against each port in parallel.

The SYN flag suggests to the remote system that you are attempting to establish a connection. Normally the destination port will be closed, and a RST (reset) packet will be sent back. If the port happens to be open, the target will take the second step of a TCP three-way-handshake by responding with a SYN/ACK TCP packet. The machine running Nmap then tears down the nascent connection by responding with a RST rather than sending an ACK packet which would complete the three-way-handshake and establish a full connection.

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. . The reason for offering both SYN and ACK ping probes is to maximize the chances of bypassing firewalls.

Another host discovery option is the UDP ping, which sends an empty UDP packet to the given ports. Upon hitting a closed port on the target machine, the UDP probe should elicit an ICMP port unreachable packet in return. This signifies to Nmap that the machine is up and available. The primary advantage of this scan type is that it bypasses firewalls and filters that only screen TCP.

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

(-v) By default, Nmap usually only prints active, responsive hosts. Verbose mode causes Nmap to print down hosts, as well as extra information about active ones.

- - source-port <port num>
Some naive firewall administrators make a ruleset exception in order to keep DNS (port 53) or FTP-DATA (port 20) working. Of course this opens a hole big enough to drive an Nmap ping scan through.

-n, - R
The - n option disables all DNS resolution, while the -R option enables DNS queries for all hosts, even down ones. The default behavior is to limit DNS resolution to active hosts.

--data-length <length>
This option adds <length > random bytes of data to every packet, and works with the TCP, UDP, and ICMP ping scan types. This helps make the scan less conspicuous. Several intrusion detection systems (IDS), including Snort, have alerts for zero-byte ping packets. This option evades those alerts.

--ttl <value>
Setting the outgoing TTL is supported for privileged users doing IPv4 ping scans. This can be useful as a safety precaution to ensure a scan does not propagate beyond the local network.

Input options (-iL <filename>, -iR <number>)
Host input options are supported as in the rest of Nmap. Users often combine the input-from-list (-iL) option with -Pn to avoid ping-scanning hosts that are already known to be up.

--reason
The normal Nmap output indicates whether a host is up or not, but does not describe which discovery test(s) the host responded to. For this detail, add the --reason option.

- s <source IP address>
As with other functions of Nmap, the source address and sending device can be specified with these options.

The TCP ping options are some of the most powerful discovery techniques in Nmap. An administrator may be able to get away with blocking ICMP echo request packets without affecting most users, but a server absolutely must respond to SYN packets sent to the public services it provides.

For security scans of target networks over the Internet, adding more probes is usually advisable. Try to include a diverse set of the techniques discussed previously. Here is a set of ping options that should catch the vast majority of hosts: -PE -PP -PS21,22,23,25,80,113,443,31339 -PA80,113,443,10042. Adding in --source-port 53 might be worthwhile as well.

```
nmap -n -sn -PE -PP -PS21,22,23,25,80,113,443,31339 -PA80,113,443,10042 -T4 --source-port 53
```

When performing security audits for clients, I normally start TCP analysis with a port scan against the most common 1000 ports (the default) with comprehensive ping scan options like those shown in Example 3.14

Ports are simply a software abstraction, used to distinguish between channels. Similar to the way IP addresses are used to identify machines on networks, ports identify specific applications in use on a single machine.

Nmap works with two protocols that use ports: TCP and UDP. A connection for each protocol is uniquely identified by four elements: source and destination IP addresses and corresponding source and destination ports. Because most popular services are registered to a well-known port number, one can often guess what services open ports represent.

While port zero is invalid, nothing stops someone from specifying it in the header field. Some malicious trojan backdoors listen on port zero of compromised systems as a stealthy way to offer illegitimate access without appearing on most port scans. To combat this, Nmap does allow scanning of port zero when it is specified explicitly (e.g. -p0-65535).

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

The six port states recognized by Nmap

open - An application is actively accepting TCP connections or UDP packets on this port. Finding these is often the primary goal of port scanning. Security-minded people know that each open port is an avenue for attack. Attackers and pen-testers want to exploit the open ports, while administrators try to close or protect them with firewalls without thwarting legitimate users.

closed - A closed port is accessible (it receives and responds to Nmap probe packets), but there is no application listening on it. They can be helpful in showing that a host is on line and using an IP address (host discovery, or ping scanning), and as part of OS detection. Because closed ports are reachable, they may be worth scanning later in case some open up.

filtered - Nmap cannot determine whether the port is open because packet filtering prevents its probes from reaching the port. The filtering could be from a dedicated firewall device, router rules, or host-based firewall software. These ports frustrate attackers because they provide so little information.

unfiltered - The unfiltered state means that a port is accessible, but Nmap is unable to determine whether it is open or closed. Only the ACK scan, which is used to map firewall rulesets, classifies ports into this state. Scanning unfiltered ports with other scan types such as Window scan, SYN scan, or FIN scan, may help resolve whether the port is open.

open|filtered Nmap places ports in this state when it is unable to determine whether a port is open or filtered. This occurs for scan types in which open ports give no response.

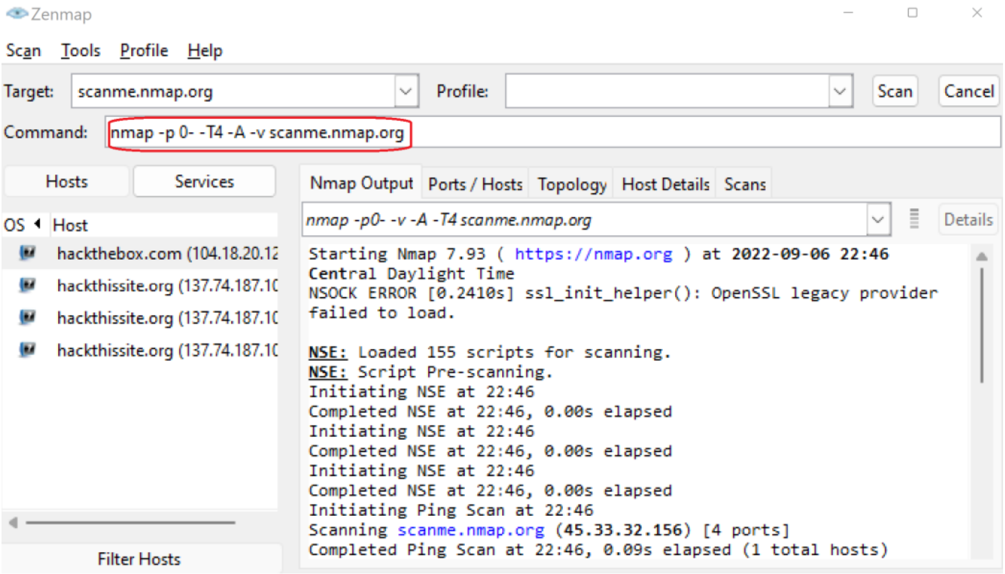
closedlfiltered This state is used when Nmap is unable to determine whether a port is closed or filtered.

One of the central tenets of network security is that reducing the number and complexity of services offered reduces the opportunity for attackers to break in. Most remote network compromises come from exploiting a server application listening on a TCP or UDP port. In many cases, the exploited application is not even used by the targeted organization, but was enabled by default when the machine was set up.

nmap <target> does the following:

- Converts <target> from a hostname into an IPv4 address using DNS. If an IP address is specified instead of a hostname this lookup is skipped.
- Pings the host, by default with an ICMP echo request packet and a TCP ACK packet to port 80, to determine whether it is up and running. If not, Nmap reports that fact and exits.
- Converts the target IP address back to the name using a reverse-DNS query.
- Launches a TCP port scan of the most popular 1,000 ports listed in nmap-services.
- Prints the results to standard output in normal human-readable format, and exits.

Example 4.3: *nmap -p0- -v -A -T4 scanme.nmap.org*
In Example 4.3, the scan is modified with four options. -p0- asks Nmap to scan every possible TCP port, -v asks Nmap to be verbose about it, -A enables aggressive tests such as remote OS detection, service/version detection, and the Nmap Scripting Engine (NSE). Finally, -T4 enables a more aggressive timing policy to speed up the scan.



Target: scanme.nmap.org

Command: nmap -p 0- -T4 -A -v scanme.nmap.org

hackthebox.com (104.18.20.12
hackthissite.org (137.74.187.1C
hackthissite.org (137.74.187.1C
hackthissite.org (137.74.187.1C

nmap -p0- -v -A -T4 scanme.nmap.org

```
Starting Nmap 7.93 ( https://nmap.org ) at 2022-09-06 22:46
Central Daylight Time
NSOCK ERROR [0.2410s] ssl_init_helper(): OpenSSL legacy provider
failed to load.

NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 22:46
Completed NSE at 22:46, 0.00s elapsed
Initiating NSE at 22:46
Completed NSE at 22:46, 0.00s elapsed
Initiating NSE at 22:46
Completed NSE at 22:46, 0.00s elapsed
Initiating Ping Scan at 22:46
Scanning scanme.nmap.org (45.33.32.156) [4 ports]
Completed Ping Scan at 22:46, 0.09s elapsed (1 total hosts)
```

TCP SYN Stealth (-sS) This is far and away the most popular scan type because it the fastest way to scan ports of the most popular protocol (TCP).

TCP Connect (-sT) Connect scan uses the system call of the same name to scan machines, rather than relying on raw packets as most of the other methods do.

UDP (-sU) Scans UDP ports.

TCP FIN, Xmas, and Null (-sF, -sX, -sN) These special purpose scan types are adept at sneaking past firewalls to explore the systems behind them.

TCP ACK (-sA) scan is commonly used to map out firewall rulesets. In particular, it helps understand whether firewall rules are stateful or not. The downside is that it cannot distinguish open from closed ports.

Protocol (-sO) Protocol scan determines which IP protocols (TCP, ICMP, IGMP, etc.) are supported by the target machine.

Port selection examples with the option -p

(-p 22) Scan a single port (in this case port 22) by specifying just that number as the -p argument.

(-p ssh) Port names may be specified rather than numbers. Note that a name may match multiple ports.

(-p 22,25,80) Multiple ports may be separated with commas.

(-p 80-85,443,8000-8005) Port ranges may be specified by separating the beginning and end port with a hyphen.

-p http * Wildcards may b e used to match ports with similar names. This expression matches eight port numbers, including http (80), http-mgmt (280), https (443), and http-proxy (8080).

Top Nmap output options applicable to port scans

-v
Increases the verbosity level, causing Nmap to print more information about the scan in progress.

-oN <filename> (normal output)
Write output in Nmap's normal format to <filename>.

-oG <filename> (grepable format output)
Write output in Nmap's so-called grepable format to <filename>. This tabular format fits the output of each host on a single line, making it easy to grep for open ports, certain operating systems, application names, or other data.

The Nmap version scanning subsystem obtains all of this data by connecting to open ports and interrogating them for further information using probes that the specific services understand. This allows Nmap to give a detailed assessment of what is really running, rather than just what port numbers are open.

To enable version detection, just add -sV to whatever Nmap flags you normally use. Or use the -A option, which turns on version detection and other Advanced and Aggressive features later.

Finer grained detection (such as distinguishing Mac OS X 10.4 from 10.3) is useful for determining vulnerability to specific flaws and for tailoring effective exploits for those vulnerabilities.

The surest way to verify that a vulnerability is real is to exploit it, but that risks crashing the service and can lead to wasted hours or even days of frustrating exploitation efforts if the service turns out to be patched. OS detection can help reduce these false positives. Simply add -O to your scan options. You may want to also increase the verbosity with -v for even more OS-related details.

Device type. All fingerprints are classified with one or more high-level device types, such as router, printer, firewall, or (as in this case) general purpose.

Running. It shows the OS Family (Linux, Windows) and OS generation (2.6.X) if available. If there are multiple OS families, they are separated by commas.

While the Device type and Running lines are from predefined enumerated lists that are easy to parse by a computer, the OS details line contains free-form data which is useful to a human reading the report. This can include more exact version numbers, device models, and architectures specific to a given fingerprint.

With two effective OS detection methods available, which one should you use? The best answer is usually both. TCP/IP fingerprinting will identify the proxy while version scanning will generally detect the server running the proxied application. If they come out the same, that makes the results more credible. If they come out wildly different, investigate further to determine what is going on before relying on either. Since OS and version detection go together so well, the -A option enables them both.

Like just about every other part of Nmap, results ultimately come from the target machine itself. While rare, systems are occasionally configured to confuse or mislead Nmap. Several programs have even been developed specifically to trick Nmap OS detection. Your best bet is to use numerous reconnaissance methods to explore a network, and don't trust any one of them.

Nmap OS fingerprinting works by sending up to 16 TCP, UDP, and ICMP probes to known open and closed ports of the target machine. These probes are specially designed to exploit various ambiguities in the standard protocol RFCs. Then Nmap listens for responses. Dozens of attributes in those responses are analyzed and combined to generate a fingerprint.

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.

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. NSE supports four types of scripts, which are distinguished by the kind of targets they take and the scanning phase in which they are run. Individual scripts may support multiple types of operation.

Prerule scripts. These scripts run before any of Nmap's scan phases, so Nmap has not collected any information about its targets yet. They can be useful for tasks which don't depend on specific scan targets, such as performing network broadcast requests to query DHCP and DNS SD servers.

Host scripts. Scripts in this phase run during Nmap's normal scanning process after Nmap has performed host discovery, port scanning, version detection, and OS detection against the target host. Examples are whois-ip, which looks up ownership information for a target IP, and path-mtu which tries to determine the maximum IP packet size which can reach the target without requiring fragmentation.

Service scripts. These scripts run against specific services listening on a target host. For example, Nmap includes more than 15 http service scripts to run against web servers.

Postrule scripts. These scripts run after Nmap has scanned all of its targets. They can be useful for formatting and presenting Nmap output.

NSE scripts define a list of categories they belong to. Currently defined categories are auth, broadcast, brute, default. discovery, dos, exploit, external, fuzzer, intrusive, malware, safe, version, and vuln. The following list describes each category.

auth These scripts deal with authentication credentials (or bypassing them) on the target system.

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

brute These scripts use brute force attacks to guess authentication credentials of a remote server. Nmap contains scripts for brute forcing dozens of protocols

default These scripts are the default set and are run when using the -sC or -A options rather than listing scripts with --script.

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

dos Scripts in this category may cause a denial of service.

exploit These scripts aim to actively exploit some vulnerability.

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

intrusive These are scripts that cannot be classified in the category because the risks are too high that they safe will crash the target system, use up significant resources on the target host (such as bandwidth or CPU time), or otherwise be perceived as malicious by the target's system administrators.

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

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