A guide to ethical hacking — Understanding Nmap

Josh Dando Follow

Feb 1 · 9 min read

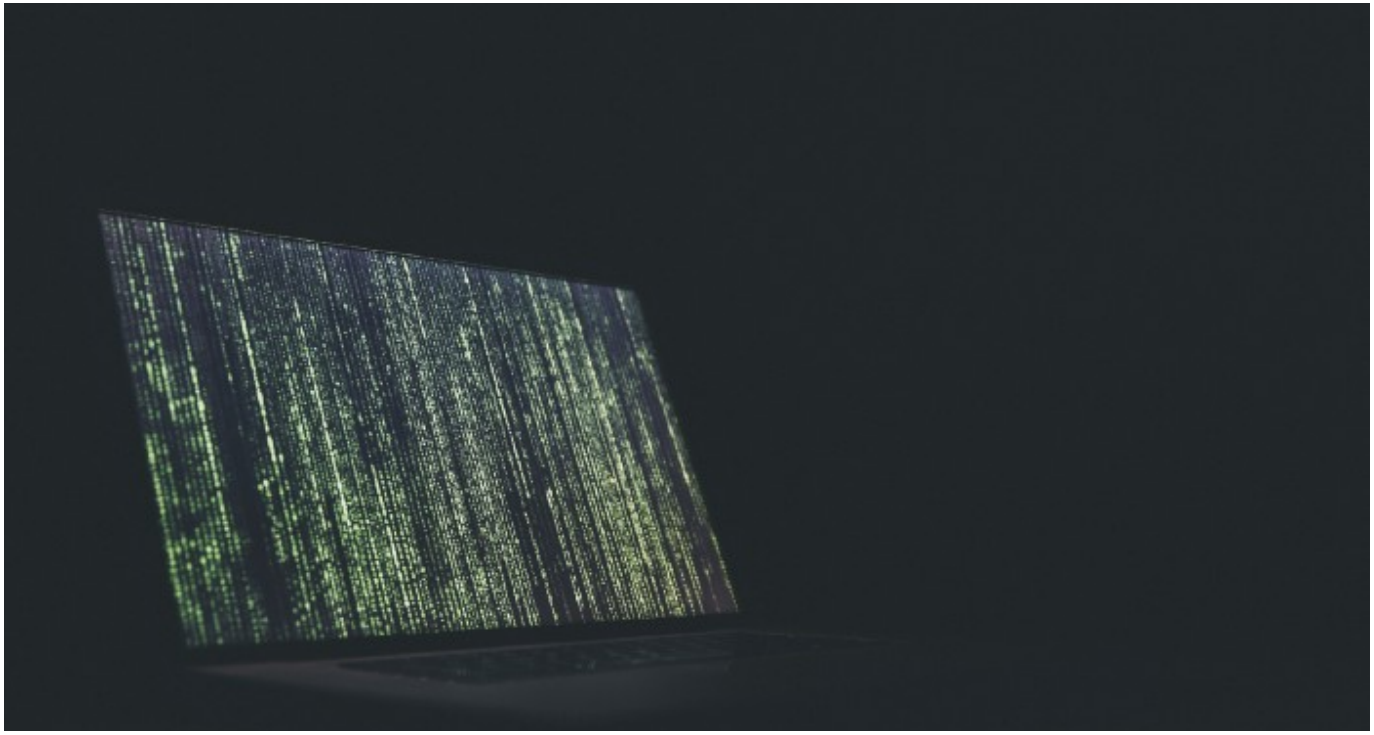


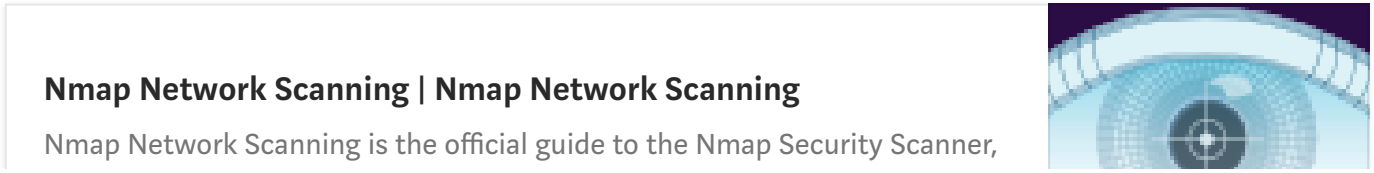
Photo by Markus Spiske on Unsplash

#1 What is Nmap?

Nmap (network mapper) is an open source software that is used to find vulnerabilities on a network by scanning through different ports.

Port scanning is Nmaps core functionality but it also can be used to collect characteristics of a network such as what services are being run and their version number, the operating systems being used and whether there are any firewall rules/packet filters.

Knowing this information as a hacker or security expert is an important step in identifying any weaknesses a system may have and the potential exploits that can be used.



Note: This tool should not be used in any illegal hacking activity. Instead to practice your skills you can use this website as a target: scanme.nmap.org or get involved in CTF (capture the flag) challenges.

#2 Host discovery aka ping scan

One of the first steps in any hack is identifying the target or whats known as the host which can be used for the attack. For a hacker he may be looking for a host that exists outside a set of firewall restrictions.

Usually on a network there are only a few active IPs at any one time. This normally holds true on private networks with large address spaces. E.g. 10.0.0.0/8 has 16 million possible hosts! Host discovery can find which IP's are active on the network.

Below are some options which can be used to control the host discovery. For a full list go to the documentation page here.

The `-sn` option is used to only perform host discovery because Nmap by default will perform port scanning after a host has been found.

#3 Understanding port states

Nmap uses 6 different port states:

Open — An open port is one that is **actively accepting TCP, UDP or SCTP connections**. Open ports are what interests us the most because they are the ones that are vulnerable to attacks. Open ports also show the available services on a network.

Closed — A port that **receives and responds** to Nmap probe packets but there is **no application listening** on that port. Useful for identifying that the host exists and for OS detection.

Filtered — Nmap can't determine whether the port is open because **packet filtering prevents its probes from reaching the port**. Filtering could come from firewalls or router rules. Often little information is given from filtered ports during scans as the filters can drop the probes without responding or respond with useless error messages e.g. destination unreachable.

Unfiltered — Port is accessible but Nmap doesn't know if its open or closed. Only used in ACK scan which is used to map firewall rulesets. Other scan types can be used to identify whether the port is open.

Open/filtered — Nmap is unable to determine between open and filtered. This happens when an open port gives no response. No response could mean that the probe was dropped by a packet filter or any response is blocked.

Closed/filtered — Nmap is unable to determine whether port is closed or filtered. Only used in the IP ID idle scan.

#4 Basic Port scanning techniques

The most basic command for scanning ports is: `nmap <target>` . This command scans 1000 TCP ports on the host. Its a SYN scan (`-sS`)which is very quick and relatively stealthy since it doesn't complete the TCP connection. If the SYN scan is unavailble due to privilages then the TCP scan (-`ST`)will be used by default. The TCP scan is less efficient and offers less control than the SYN scan.

Photo by Chris Nguyen on Unsplash

> *A UDP scan works by sending a UDP packet to every targeted port.- https://nmap.org/book/man-port-scanning-techniques.html*

It is often quite slow but shouldn't be avoided because a lot of services run on these ports. Nmap rate limits the connection to avoid flooding the network. To speed up this search you could look at the most popular ports, scan more ports in parallel or use the `--host-timeout` option to avoid slow hosts.

Note: some ports can be setup to return confusing or misleading results.

SCTP combines the architecture of TCP and UDP and includes congestion avoidance, resistance to flooding and features such as multi streaming and multi homing. Like the SYN scan, SCTP scan is fast, stealthy and clearly defines the port states. It can be performed by using the `-sY` option.

The aggresive scan option (`-A`) combines various different scan types such as operating system detection, script scanning and trace route. Its a good option for when you want a complete scan report without caring how intrusive you are.

TLDR;

```
nmap -sS 192.168.1.1 => SYN scan
nmap -sT 192.168.1.1 => TCP scan
nmap -sU 192.168.1.1 => UDP scan
nmap -sY 192.168.1.1 => SCTP scan
nmap -A 192.168.1.1 => Aggressive scan
```

#5 Advanced Port scanning techniques

You can combine a UDP scan (`-sU`) with any SCTP or TCP scans.

`-sN, -sF, -sX` are used to differentiate between open and closed ports by exploiting a loophole in TCP RCF.

- Port = closed when RST packet is received
- Port = open|filtered when no response
- Port = unfiltered when ICMP error

Ack scan (-sA) is used primarily for mapping out firewall rules by finding out if they are stateful and which ports are filtered.

`-sZ` is a cookie echo scan which is a more obscure method and therefore less likely to be picked up as a port scan or blocked. A port is open if the packet is dropped or closed if the ABORT flag is sent. One disadvantage is that it can't differentiate between open|filtered.

Zombie scan (`-sI<zombiehost>[:<probeport>]`) is the best scan for when you don't want to leave a trace of your IP on a targets system; it's truly the ultimate stealth scan. It also has the added benefit of finding out trusted IP's as it uses the IP of the zombie. Experimenting with different zombie IP's can be useful for finding out which IP's are trusted by a system.

The protocol scan (`-sO`) can be used to identify protocols supported by the targets system.

#6 Port specification

Because a system can contain millions of different ips and thousands of ports it can be useful to specify which ports you want to scan to reduce scanning time. By default nmap scans the top 1000 most likely ports.

- `-p <port-range>` e.g. 1–1023 is used to specify the ports that will be scanned
- Inversely you can specify `--exclude-ports`
- Got to go fast! Use `-F` to reduce the ports scanned to the top 100.

#7 Service and version detection

Knowing the services and the specific version of those services running on a port is valuable information to a hacker. Older versions of software tend to have well known vulnerabilities that a hacker can exploit.

Nmap searches the patterns received from the probes against a known database of services for matches.

Nmap command for version detection: `-sV`

#8 OS detection

Operating System detection is done via TCP/IP fingerprinting. This consists of a bunch of tests including a comparison of the fingerprint to a known list from Nmaps database.

> *"When verbose mode is enabled along with `-O`, IP ID sequence generation is also reported. Most machines are in the "incremental" class, which means that they increment the ID field in the IP header for each packet they send. This makes them vulnerable to several advanced information gathering and spoofing attacks."*

To perform OS detection use `-O` .

Nmap offers some additional options that can be used to speed up or refine your search.

```
--oscan-limit = Limits the OS detection to targets with atleast an
open and closed port resulting in higher chance of success

--fuzzy = Used for when Nmap can't make a clear guess, displays
confidence score

--max-os-tries = The default is 5, set to a lower number to speed up
the scan.
```

#9 Firewall Evasion and Spoofing

This section seperates the amateur hackers from the more experienced ones. Inexperienced hackers will often use the default settings of Nmap and end up getting their IP blocked by the target IDS or their packets dropped by the network firewall.

An experienced hacker will be patient and probe the target using different MAC/IP addresses to gain information about the targets system. Next the

hacker will use his skill to avoid firewall rules using proxies or different routes and evade the IDS by overwhelming it with decoy attacks.

> *Note: "All of the major IDSs ship with rules designed to detect Nmap scans because scans are sometimes a precursor to attacks" — https://nmap.org/book/man-bypass-firewalls-ids.html*

What does IDS stand for?

IDS = Intrusion detection system. It's job is to monitor the network for any suspicious activity and report the logs to the system administrator if any malicious packets are detected. It can also block IPs from accessing the network in the future.

A firewall can be hardware or software. It is a line of defence that either stops or allows packets of data through to the network by using a predefined set of rules.

The one thing going for hackers is that it is difficult to detect malicious activity if the hacker utilizes skill and patience. An IDS may also report a lot of false positives giving the hacker more time before the administrator can

analyze the logs and identify the culprit. Using fake IP or MAC addresses along with proxies can make the search for the real hacker even longer.

Nmap offers various different script options used to bypass IDS and firewalls. It is up to the hacker to decide the correct tool for the job as there is no single command that is guaranteed to work.

List of commands:

- `-f` or `--mtu` = fragment packets and specify the fragment size as a multiple of 8. The idea of this is to split the packets into multiple smaller ones which makes the scan harder to detect and can be used to evade the IDS. Note: *Some programs have trouble handling fragmented packets.*
- `-D` = Decoy. This accepts a list of decoy hosts that will show to the target that you are scanning from. Note: that your IP address will be included amongst the list of decoys!! The position of your address among the decoys can be changed.
- `-S` = Spoof source address
- `-e` = Tell Nmap what interface to send/receive the packets from

- `--source-port` or `-g` = spoof source port number
- `--data` = Append custom binary data to packets (wonder what we can do with this 😉)
- `--data-string` = Send a custom string along with the packet used as a comment e.g. “scan performed by system administrator”
- `--data-length` = Append random data to the packets to make the scan harder to detect
- `--ip-options <S|R [route]|L [route]|T|U ... >;` `--ip-options <hex string>` = Send packets with specified IP options. The options here allow you to record routes taken by packets and specify your route.
- `--ttl` = Time to live for IPv4
- `--randomize-hosts` = Randomize target host order. Makes the scan less obvious
- `--spoof-mac` = Used to spoof the MAC address of the source. If you don’t specify all 12 characters then Nmap will randomly fill in the remaining ones.

- `--proxies<comma seperated list of proxy urls in form proto://host:port` = Relay TCP connections through a chain of proxies. Proxies are used to hide the source of the scan and to evade firewall rules.
- `--badsum` = Uses a malconfigured TCP/UDP checksum. This is used to probe the detection of a firewall or IDS. If a response is received then a firewall/IDS is present.

> The true sign of intelligence is not knowledge but imagination. — Albert Einstein

I hope this section has not only increased your understanding but also opened your mind for the possibilities available to us through these options Nmap offers.

Conclusion

> “Knowledge is power” — Francis Bacon

Every attack starts with gathering intel. Without this information about the target, an educated attack can not be conducted. Which minimizes the hackers chance of success and can even lead to him being caught! Therefore it can not be understated how important the reconnaissance step is in any hack.

On the other end of things if your system is suddenly receiving more packets than normal then this could indicate a potential attack is about to happen. Learning how an attack is coordinated is the 1st step towards prevention. In part 2 we will explore the theory behind some of these script options used for scanning ports and avoiding firewalls/IDS.

. . .

Most of this information came from the original nmaps documentation page but if you did find this blog to be more consise and easier to read then please support this series with a clap.

If you are interested in learning how to become a hacker or how to defend your systems better then give me a **follow!** I also plan to cover and do breakdowns on famous hacks and guides on various capture the flag challenges.

Networking Hacking Security Nmap Pentesting

WRITTEN BY

Josh Dando

Front end engineer with an interest in technology, fitness and travel

Follow

See responses (1)

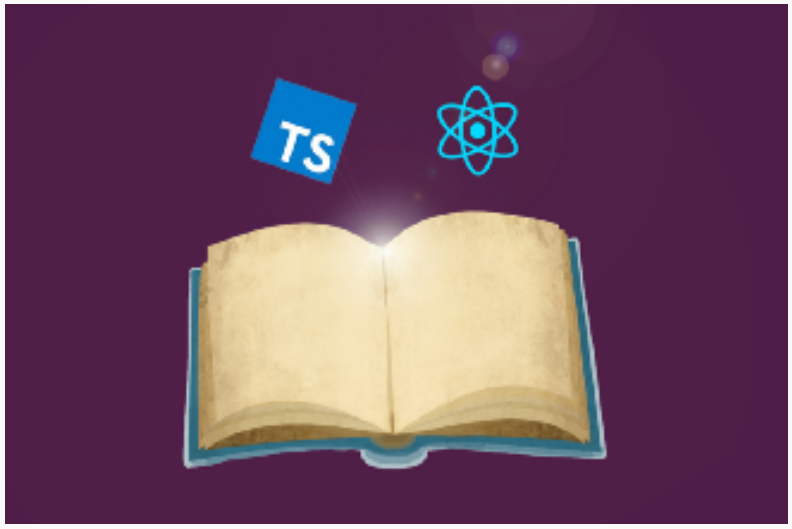
More From Medium

More from Josh Dando

Setting up a React + TypeScript + Storybook project

Josh Dando
Jan 5 · 6 min read ★

431



Related reads

The BatchOverflow Bug and How to Catch All Bugs

Kiran Garimella
May 11, 2018 · 12 min read ★

279

Related reads

The Bugs Are Out There, Hiding in Plain Sight

A Bug'z Life in A Bug'z Life

Jul 15 · 6 min read ★

240