Host Discovery + OS Fingerprinting: Useful on both local and remote hosts.

First once we identify our target network we should perform a **ping sweep** to locate alive machines on the network. (of course, servers can block external pings but this is a good start regardless)

* There are tools that can be fed a subnet and will systematically ping the entire network. Something like httprobe but instead of subdomains it takes in a subnet.
  + fping -a -g IPRANGE is a cool linux tool for this
  + fping returns a weird error for all hosts you test but that can be suppressed with the following: 

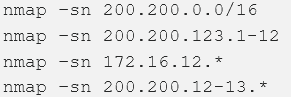
^^this is how the command should be ran (or with CIDR notation ex. /24)

This utility is just for basic how discovery. Once we identify alive hosts, we should run a port scan such as nmap to enumerate our target.

Nmap OS fingerprinting and host discovery:

Run this on localhost (as opposed to VM) nmap takes long as is we don’t need it to take longer.

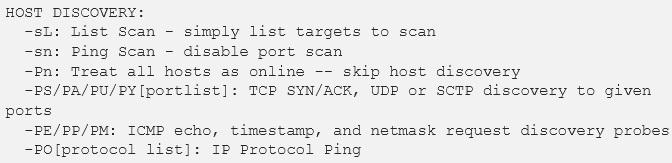
Conveniently, nmap provides its own ping scan feature which can be done with the following:



Alternatively, we can pass in an input list with the -iL switch. This can be useful depending on the format our hosts are in. Eg if theyre all sperate randomish individual IPs that can be thrown in a .txt file.



Nmap can do so much, look over the docs and test stuff out, here are most host discovery tools nmap provides:



Once we have a list of alive IPs we need to being fingerprinting the OS. This can be done offline (passively) with the p0f tool or with an active nmap scan using the -O switch. (use the -Pn switch as well to skip the ping scan this can also come in handy for hosts that block pings!!)  we can also choose a lighter or more aggressive os scan with the following switches

The --osscan-limit is useful when scanning hundreds or thousands of targets, so it will run faster and just show the most promising results.

The -sT switch when used with -O tells nmap to do a TCP scan. This is very useful when doing OS fingerprinting so it’s a good option. Of course, this will add time to the scan and moreover this type of scan makes a full connection so it will be recorded by the application logs.

OS scanning works by sending various requests to the target and comparing the TCP and UDP data it receives from the server to a database. This database contains TCP and UDP information of known OS’s which nmap uses to fingerprint the OS of the target.

Port Scanning:

Port scanning is a process that determines what TCP and UDP ports are open on target hosts. It also tells us which daemon in terms of software and version is listening on each specific port. Similarly, to OS fingerprinting port scanning works by sending different probes to a target and analyzing its response. The goal of port scanning is to identify open ports and the services running on them to identify potential vulnerabilities.

Understanding a TCP port scanner:

A TCP port scanner uses the three-way handshake to detect open ports. When a scanner attempts to connect to a closed port, instead of receiving a SYN + ACK flags back from the server it instead receives a RST + ACK flag. This unique set of flags indicates that port is closed.

The most simple way to do a port scan is to try to connect to every port, if the 3-way handshake can be completed that port is open, if instead a RST + ACK is received, the port is closed. If open, the scanner then quickly sends an RST packet to close that connection and moves on to the next port.

* The RST flag is sent by the server or client to close a connection.

All these incoming TCP requests are stored within the logs of the target server, and since so many requests are being generated it makes it very easy for a sys admin to detect. For this reason, TCP SYN scans were invented, these types of scans are much stealthier.

SYN Scans

SYN scans do not perform a full handshake with the target machine, instead they just send a TCP segment with a SYN flag embed within it. Same concept as the other scan, if an RST flag is sent back the port is closed, if ACK is sent back its open. As no full connection is made to the machine, the daemon logs do not detect this type of behavior.

-sS denotes a SYN stealth scan

Note well configured intrusion detection systems will still pick up this type of scan.

Common Nmap Scans:

**-sV scan** -> version detection scan, combines a TCP scan with some probes. While not very stealthy this can be super useful. This is used to detect what application is listening on a particular port.

Diagram

Description automatically generated

Similar to the other scans, nmap scans the behavior of the application in response to the TCP connections and probes and uses those responses to determine which app is running. The only difference for this scan is the banner which really helps to ID an app.

Nmap is very flexible on the type of input it can take in like Domain name, IP, CIDR notation (subnet), wildcards (\*), ranges of IPs, octets lists and input files.

Or even a mix like this  or this . Any methods can be combined together, in different positions of an IP. Of course, the -iL flag is used for Input list.

If we don’t want to scan all common ports we can use the -p flag to specify the ones we want.

Text

Description automatically generatedSome examples of nmap syntax:

If we think a host is blocking pings (to make us think its not alive), we can use the -Pn to skip the pin scan and treat the host as alive. We can do a port scan on the host we think is alive to check if it as, as some ports are almost always open on certain systems.

Table

Description automatically generatedThe 4 most basic ports. 22,445,80,443 are the best ports to look for when checking if a machine is alive, as these are of the most common.

Identifying firewalls:   
- Identify incomplete results returned by nmpa. If a TCP scan succeedes against a well known service like a web sever nmap should easily be able to fingerprint it with the -sV switch.

Graphical user interface, text, application, email

Description automatically generated- but we could also see something like this  or like this

Both of these results indicate that there is likely a firewall in place.

* If we see this we should run nmap with the --reason switch to show an explanation as to why a port is marked open or closed. This could, for example, return: Text

  Description automatically generated

If we see the state of a port a filtered this signals that nmap doesn’t have enough data to say if its open or closed, this could be from a firewall and we need to test the port further as it could be of interest.

Mass Scan:

Mass scan is a much faster, but often less accurate alternative to nmap. This is useful for huge numbers of targets like for a bug bounty.

We could perform host discovery with mass scan then scan the interesting hosts with nmap.

To idenfity hosts with mass scan we do the following:

./masscan -p22,80,443,53,3389,8080,445 -Pn --rate=800 --banners 192.168.2.0/24

This specifies common ports, makes it so it test all hosts even ones that block ping, limit rate to 800 packets/ second and fingerprint services via banner where possible. Of course, replace IP or subnet as needed.

SQL database enum:

Text, letter

Description automatically generatedOther DB’s also have similar type scripts.