Contents

Intro

This is notes from the Nmap rooms on tryhackme's JR pentest path

Summary of basic nmap commands

Summary for this room: https://tryhackme.com/room/nmap01
Scan Type | Example Command ———–|———————– ARP Scan | sudo nmap -PR -sn MACHINE_IP/24 ICMP Echo Scan | sudo nmap -PE -sn MACHINE_IP/24 ICMP Timestamp Scan | sudo nmap -PP -sn MACHINE_IP/24 ICMP Address Mask Scan | sudo nmap -PM -sn MACHINE_IP/24 TCP SYN Ping Scan | sudo nmap -PS22,80,443 -sn MACHINE_IP/30 TCP ACK Ping Scan | sudo nmap -PA22,80,443 -sn MACHINE_IP/30 UDP Ping Scan |sudo nmap -PU53,161,162 -sn MACHINE_IP/30

Remember to add -sn if you are only interested in host discovery without port-scanning. Omitting -sn will let Nmap default to port-scanning the live hosts. Option | Purpose ——–|——— -n | no DNS lookup -R | reverse-DNS lookup for all hosts -sn | host discovery only

Summary of more nmap commands

Port Scan Type	Example Command
TCP Null Scan	sudo nmap -sN 10.10.164.191
TCP FIN Scan	sudo nmap -sF 10.10.164.191
TCP Xmas Scan	sudo nmap -sX 10.10.164.191

Port Scan Type	Example Command
TCP Maimon Scan	sudo nmap -sM 10.10.164.191
TCP ACK Scan	sudo nmap -sA 10.10.164.191
TCP Window Scan	sudo nmap -sW 10.10.164.191
Custom TCP Scan	sudo nmap –scanflags URGACKPSHRSTSYNFIN 10.10.164.191
Spoofed Source IP	sudo nmap -S SPOOFED_IP 10.10.164.191
Spoofed MAC Address	–spoof-mac SPOOFED_MAC
Decoy Scan	nmap -D DECOY_IP,ME 10.10.164.191
Idle (Zombie) Scan	sudo nmap -sI ZOMBIE_IP 10.10.164.191
Fragment IP data into 8 bytes	-f
Fragment IP data into 16 bytes	-ff

Option	Purpose
–source-port PORT_NUM	specify source port number
–data-length NUM	append random data to reach given length

These scan types rely on setting TCP flags in unexpected ways to prompt ports for a reply. Null, FIN, and Xmas scan provoke a response from closed ports, while Maimon, ACK, and Window scans provoke a response from open and closed ports.

Option	Purpose

–reason explains how Nmap made its conclusion -v | verbose -vv | very verbose -d | debugging -dd | more details for debugging

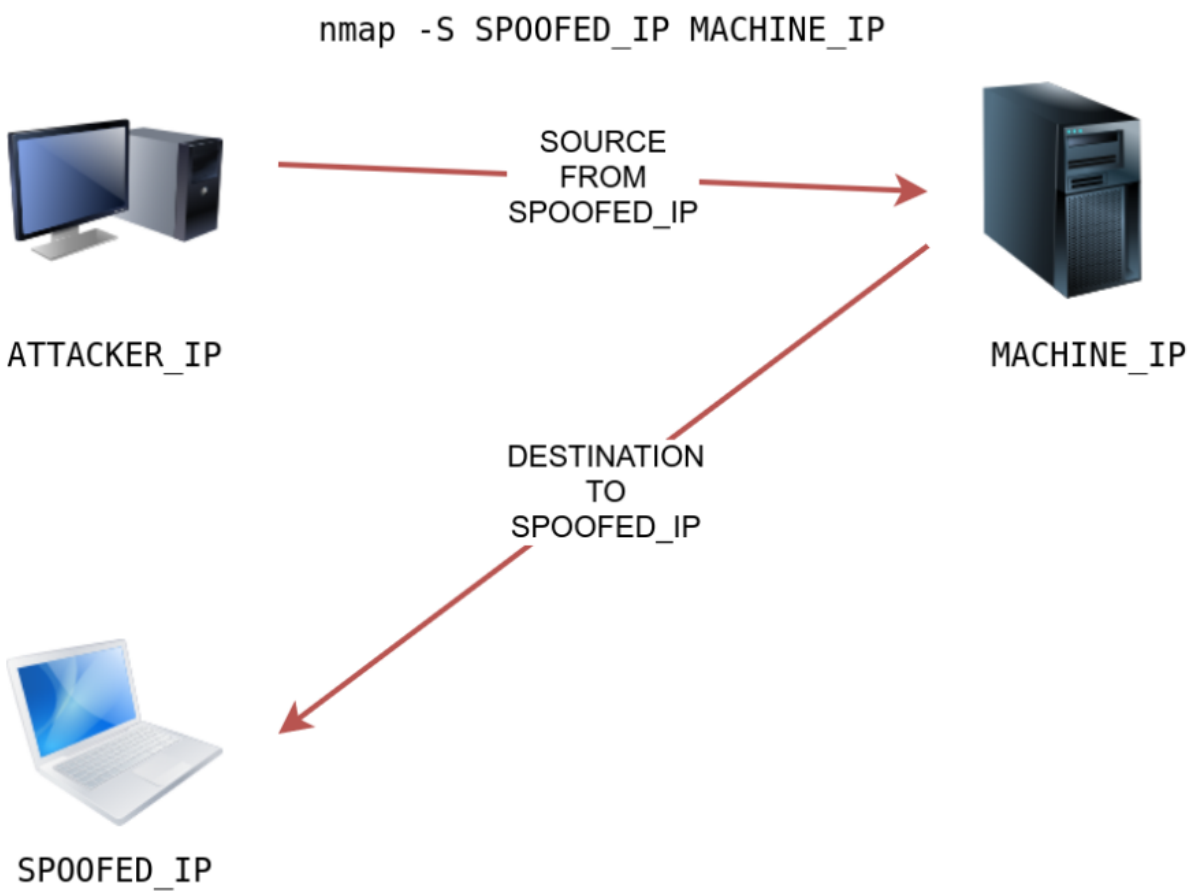
Nmap post port scans

Option	Meaning
-sV	determine service/version info on open ports
-sV –version-light	try the most likely probes (2)
-sV –version-all	try all available probes (9)
-O	detect OS
–traceroute	run traceroute to target
–script=SCRIPTS	Nmap scripts to run
-sC or –script=default	run default scripts
-A	equivalent to -sV -O -sC –traceroute
-oN	save output in normal format
-oG	save output in grepable format
-oX	save output in XML format
-oA	save output in normal, XML and Grepable formats

Spoofing and decoys

In brief, scanning with a spoofed IP address is three steps:
1. Attacker sends a packet with a spoofed source IP address to the target machine. 2. Target machine replies to the spoofed IP address as the destination. 3. Attacker captures the replies to figure out open ports.

In general, you expect to specify the network interface using `-e` and to explicitly disable ping scan `-Pn`. Therefore, instead of `nmap -S SPOOFED_IP 10.10.37.119`, you will need to issue nmap `-e NET_INTERFACE -Pn -S SPOOFED_IP 10.10.37.119` to tell Nmap explicitly which network interface to use and not to expect to receive a ping reply. It is worth repeating that this scan will be useless if the attacker system cannot monitor the network for responses.



Using SPOOFED_IP as the source IP address for the scan.

{: .width-25}

When you are on the same subnet as the target machine, you would be able to spoof your MAC address as well. You can specify the source MAC address using `--spoof-mac SPOOFED_MAC`. This address spoofing is only possible if the attacker and the target machine are on the same Ethernet (802.3) network or same WiFi (802.11).

You can launch a decoy scan by specifying a specific or random IP address after `-D`. For example, `nmap -D 10.10.0.1,10.10.0.2,ME 10.10.37.119` will make the scan of `10.10.37.119` appear as coming from the IP addresses 10.10.0.1, 10.10.0.2, and then ME to indicate that your IP address should appear in the third order. Another example command would be `nmap -D 10.10.0.1,10.10.0.2,RND,RND,ME 10.10.37.119`, where the third and fourth source IP addresses are assigned randomly, while the fifth source is going to be the attacker's IP address. In other words, each time you execute the latter command, you would expect two new random IP addresses to be the third and fourth decoy sources.