# NMAP CHEAT SHEET AND TUTORIAL

https://www.networkstraining.com/

## PART A: NMAP CHEAT SHEET

### Scan IP address (Targets)

| Command | Description |
|---|---|
| nmap 10.0.0.1 | Scan a single host IP |
| nmap 192.168.10.0/24 | Scan a Class C subnet |
| nmap 10.1.1.5-100 | Scan the range of IPs between 10.1.1.5 up to 10.1.1.100 |
| nmap -iL hosts.txt | Scan the IP addresses listed in text file “hosts.txt” |
| nmap 10.1.1.3 10.1.1.6 10.1.1.8 | Scan the 3 specified IPs only |
| nmap www.somedomain.com | First resolve the IP of the domain and then scan its IP address |

**NOTE:**

Because we have not specified any other switches on the commands above (except the target IP address), the command will perform first host discovery by default and then scan the most common 1000 TCP ports by default.

### Port Related Commands

On the section above we have not specified any ports which means the tool will scan the 1000 most common ports. However, in real engagements you should specify port numbers as well as shown below.

| Command | Description |
|---|---|
| nmap -p80 10.1.1.1 | Scan only port 80 for specified host |
| nmap -p20-23 10.1.1.1 | Scan ports 20 up to 23 for specified host |
| nmap -p80,88,8000 10.1.1.1 | Scan ports 80,88,8000 only |
| nmap -p- 10.1.1.1 | Scan ALL ports for specified host |
| nmap -sS -sU -p U:53,T:22 10.1.1.1 | Scan ports UDP 53 and TCP 22 |
| nmap -p http,ssh 10.1.1.1 | Scan http and ssh ports for specified host |

## Different Scan Types

Nmap is able to use various different techniques to identify live hosts, open ports etc. The following are the most popular scan types.

| Command | Description |
|---|---|
| nmap -sS 10.1.1.1 | TCP SYN Scan (best option) |
| nmap -sT 10.1.1.1 | Full TCP connect scan |
| nmap -sU 10.1.1.1 | Scan UDP ports |
| nmap -sP 10.1.1.0/24 | Do a Ping scan only |
| nmap -Pn 10.1.1.1 | Don't ping the hosts, assume they are up. |

There are some more scan types supported by nmap but we have listed the most useful ones above. Here is an overview of the most popular scan types:

- **-sS**: This sends only a TCP SYN packet and waits for a TCP ACK. If it receives an ACK on the specific probed port, it means the port exist on the machine. This is fast and pretty accurate.
- **-sT**: This creates a full TCP connection with the host (full TCP handshake). This is considered more accurate than SYN scan but slower and noisier.
- **-sP**: This is for fast checking which hosts reply to ICMP ping packets (useful if you are on the same subnet as the scanned range and want a fast result about how many live hosts are connected).

## Identify Versions of Services and Operating Systems

Another important feature of NMAP is to give you a wealth of information about what versions of services and Operating Systems are running on the remote hosts.

| Command | Description |
|---|---|
| nmap -sV 10.1.1.1 | Version detection scan of open ports (services) |
| nmap -O 10.1.1.1 | Identify Operating System version |
| nmap -A 10.1.1.1 | This combines OS detection, service version detection, script scanning and traceroute. |

## Scan Timings

These switches have to do with how fast or slow the scan will be performed.

| Command | Description |
|---|---|
| nmap -T0 10.1.1.1 | Slowest scan (to avoid IDS) |
| nmap -T1 10.1.1.1 | Sneaky (to avoid IDS) |
| nmap -T2 10.1.1.1 | Polite (10 times slower than T3) |
| nmap -T3 10.1.1.1 | Default scan timer (normal) |
| nmap -T4 10.1.1.1 | Aggressive (fast and fairly accurate) |
| nmap -T5 10.1.1.1 | Very Aggressive (might miss open ports) |

## Output Types

For each scan we recommend outputting the results in a file for further evaluation later on. Nmap supports 3 main output formats as below:

| Command | Description |
|---|---|
| nmap -oN [filename] [IP hosts] | Normal text format |
| nmap -oG [filename] [IP hosts] | Grepable file (useful to search inside file) |
| nmap -oX [filename] [IP hosts] | XML file |
| nmap -oA [filename] [IP hosts] | Output in all 3 formats supported |

Example:

**nmap -oN scan.txt 192.168.0.0/24** (this will scan the subnet and output the results in text file “scan.txt”)

## Discover Live Hosts

There are various techniques that can be used to discover live hosts in a network with nmap. Depending on whether you are scanning from the same LAN subnet or outside of a firewall, different live host identifications can be used (we will discuss this later).

| Command | Description |
|---|---|
| nmap -PS22-25,80 10.1.1.0/24 | Discover hosts by TCP SYN packets to specified ports (in our example here the ports are 22 to 25 and 80) |
| nmap -Pn 10.1.1.0/24 | Disable port discovery. Treat all hosts as online. |
| nmap -PE 10.1.1.0/24 | Send ICMP Echo packets to discover hosts. |
| nmap -sn 10.1.1.0/24 | Ping scan. |

## NSE Scripts

Did you know that nmap is not only a port scanner? Actually, there are hundreds of included scripts that you can use with nmap to scan for all sorts of vulnerabilities, brute force login to services, check for well-known weaknesses on services etc.

| Command | Description |
|---|---|
| nmap --script=”name of script” 10.1.1.0/24 | Run the specified script towards the targets. |
| nmap --script=”name of script” --script-args=”argument=arg” 10.1.1.0/24 | Run the script with the specified arguments. |
| nmap --script-updatedb | Update script database |

## Other Useful Commands

Some other miscellaneous but useful commands

| Command | Description |
|---|---|
| nmap -6 [IP hosts] | Scan IPv6 hosts |
| nmap --proxies url1,url2 | Run the scan through proxies |
| nmap --open | Only show open ports |
| nmap --script-help=”script name” | Get info and help for the specified script |
| nmap -V | Show currently installed version |
| nmap -S [IP address] | Spoof source IP |
| nmap --max-parallelism [number] | Maximum parallel probes/connections |
| nmap --max-rate [number] | Maximum packets per second |

# PART B: NMAP Tutorial and Examples

This is the second part of this article where I'll show you some examples, use cases and techniques of using nmap in practical penetration testing and security assessment engagements.

## #1 My personal favourite way of using Nmap

Whenever I start a penetration test, I follow the steps below with nmap.

### Step 1a: Host Discovery with well knows ports

**nmap -PS21-25,80,88,111,135,443,445,3306,3389,8000-8080 -T4 -oA hostdiscovery 100.100.100.0/24**

The above will perform host discovery to identify live hosts using some well-known ports (21-25, 80, 443 etc). The output will be 3 files (gnmap, xml, txt) with filename "hostdiscovery". We assume the target network range is 100.100.100.0/24

With the above technique, if at least one of the above TCP ports is open on a target host in the IP range then nmap will know that the host is alive.

The above technique is efficient if you are scanning a large public IP range and you know there is a firewall in front and that only limited ports are visible because of the firewall. The above ports will most probably be visible on public hosts.

### Step 1b: Host Discovery with ICMP

**nmap -PE -oA hostdiscovery 192.168.1.0/24**

The above is a variation of previous step (Step 1a) whereby nmap sends ICMP packets to discover live hosts.

This technique is effective if you are scanning from the same LAN subnet as the target range and there is no firewall in front of the hosts and also ICMP ping is not blocked from the hosts.

The end result is the same as the previous step. Live hosts will be recorded in filename "hostdiscovery" with several ports marked as open for each IP address.

### Step 2: Filter Above Files to Create a Clean Live Hosts Lists

The filename created above ("hostdiscovery") will contain hosts with open ports. We can filter all IP addresses in the file above that have at least one open port and create a clean list of live host IPs.

I use the linux "**awk**" command for this task as shown below:

**# awk '/open/{print $2}' hostdiscovery.gnmap > livehosts.txt**

From Step 1 before, there are three files created and one of them is a greppable format file with extension gnmap ("**hostdiscovery.gnmap**").

We run awk to search for open ports in that file and then redirect the output to another file "**livehosts.txt**". This file will only contain a list of IP addresses that correspond to live hosts in the target network.

## Step 3: Perform Full Port Scan using the Live Hosts List

Now after identifying the live hosts in the whole subnet, we can perform full port scan with nmap towards these hosts only.

By doing this, we managed to be more efficient and perform scans faster than doing full port scan on the whole target range from the beginning.

**nmap -p- -Pn -sS -A -T4 -iL livehosts.txt -oA fullscan**

-p- : This scans all ports

-Pn : Do not perform host discovery again

-sS : Perform TCP SYN scan

-A : This combines OS detection, service version detection, script scanning and traceroute

-T4 : Pretty fast and accurate scanning

-iL livehosts.txt : Scan the IPs contained in file “livehosts.txt”

-oA : Export the results in file “fullscan”

## #2 Scan network for EternalBlue (MS17-010) Vulnerability

In 2017 a huge zero-day vulnerability in Windows SMB was leaked to the public with the name “EternalBlue” (reference code MS17-010 from Microsoft). This is a critical risk vulnerability that allows easy compromise of remote Windows machines.

You must scan your networks to find out if you have Windows machines that are not patched for this and the following nmap script is very useful for this task.

**nmap -Pn -p445 --script=smb-vuln-ms17-010 192.168.1.0/24 -oN eternalblue-scan.txt**

The command above will scan the whole Class C network 192.168.1.0/24 on port 445 (SMB port) for the EternalBlue vulnerability and will write the results in file “eternalblue-scan.txt”

## #3 Find HTTP servers and then run nikto against them

The following scans the target range (100.100.100.0/24) for HTTP servers (ports 80 and 443) and then pipes the result to “Nikto” for further HTTP scans. Nikto is an open source tool for identifying well known HTTP vulnerabilities.

**nmap -p80,443 100.100.100.0/24 -oG - | nikto.pl -h –**

## #4 Find Servers running Netbios (ports 137,139, 445)

**nmap -sV -v -p 137,139,445 192.168.1.0/24**

## #5 Find Geo Location of a specific IP address

The following command uses geolocation script "**ip-geolocation-ipinfodb**" to find the geographic location of a specific IP address. To use the above script you need to create a free account at https://ipinfodb.com/register.php and get an API key to use in the command as shown below (in script-args)

**nmap --script=ip-geolocation-ipinfodb --script-args=ip-geolocation-ipinfodb.apikey=[APIKEY] 8.8.8.8**

Nmap scan report for google-public-dns-a.google.com (8.8.8.8)
Host is up (0.0097s latency).
Not shown: 998 filtered ports
PORT STATE SERVICE
53/tcp open domain
443/tcp open https

Host script results:
| ip-geolocation-ipinfodb:
| 8.8.8.8
| coordinates (lat,lon): 37.406,-122.079
|_ city: Mountain View, California, United States

## #6 Detect if a Website is protected by WAF

A WAF (Web Application Firewall) can be a software or hardware device in front of webservers to protect from HTTP web application attacks.

The following command uses a script to detect if the target website is protected by a Web Application Firewall (WAF). The **http-waf-detect** script uses two arguments to try the tool's built-in attack vectors for evaluating if the target web domain is protected by a WAF.

**# nmap -p80,443 --script http-waf-detect --script-args="http-waf-detect.aggro,http-waf-detect.detectBodyChanges" www.networkstraining.com**

Nmap scan report for www.networkstraining.com (104.18.38.202)
Host is up (0.011s latency).
PORT STATE SERVICE
80/tcp open http
443/tcp open https
| http-waf-detect: IDS/IPS/WAF detected:
|_www.networkstraining.com:443/?p4yl04d=hostname%00

## #7 Find well known vulnerabilities related to an open port

Let's say you have scanned a target host and found several open services/ports running on the host. With nmap you can query public vulnerability databases to find out if there are any known published vulnerabilities related to the services running.

### Step 1:

First you need to download the "nmap-vulners" script from Git and place it under the script directory of nmap:

**# cd /pentest/vulnerability-analysis/nmap/scripts** (or whatever the scripts directory is)

**# git clone https://github.com/vulnersCom/nmap-vulners.git**

### Step 2:

Since the script needs to know the exact version of the remote scanned service, you must use the **-sV** key when using the **vulners** script:

**# nmap -Pn -sV -p80 --script=vulners scanme.nmap.org**

```
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
| vulners:
|   cpe:/a:apache:http_server:2.4.7:
|       CVE-2017-7679          7.5          https://vulners.com/cve/CVE-2017-7679
|       CVE-2018-1312          6.8          https://vulners.com/cve/CVE-2018-1312
|       CVE-2014-0226          6.8          https://vulners.com/cve/CVE-2014-0226
|       CVE-2017-15715         6.8          https://vulners.com/cve/CVE-2017-15715
|       CVE-2017-9788          6.4          https://vulners.com/cve/CVE-2017-9788
|       CVE-2013-6438          5.0          https://vulners.com/cve/CVE-2013-6438
|       CVE-2014-0231          5.0          https://vulners.com/cve/CVE-2014-0231
|       CVE-2017-9798          5.0          https://vulners.com/cve/CVE-2017-9798
|       CVE-2016-8743          5.0          https://vulners.com/cve/CVE-2016-8743
|       CVE-2017-15710         5.0          https://vulners.com/cve/CVE-2017-15710
|       CVE-2016-0736          5.0          https://vulners.com/cve/CVE-2016-0736
|       CVE-2014-3523          5.0          https://vulners.com/cve/CVE-2014-3523
|       CVE-2016-2161          5.0          https://vulners.com/cve/CVE-2016-2161
|       CVE-2018-17199         5.0          https://vulners.com/cve/CVE-2018-17199
|       CVE-2014-0098          5.0          https://vulners.com/cve/CVE-2014-0098
|       CVE-2016-4975          4.3          https://vulners.com/cve/CVE-2016-4975
|       CVE-2014-0117          4.3          https://vulners.com/cve/CVE-2014-0117
|       CVE-2014-8109          4.3          https://vulners.com/cve/CVE-2014-8109
|       CVE-2015-3185          4.3          https://vulners.com/cve/CVE-2015-3185
|       CVE-2014-0118          4.3          https://vulners.com/cve/CVE-2014-0118
|       CVE-2018-1283          3.5          https://vulners.com/cve/CVE-2018-1283
|_      CVE-2016-8612          3.3          https://vulners.com/cve/CVE-2016-8612
```

As you can see from above, we have scanned port 80 (with -sV switch) and used the **vulners** script to get all known public vulnerabilities of the specific service (Apache httpd 2.4.7).

**Written By:** https://www.networkstraining.com/