Hacker tools: Nmap – Next level port scanning

BY ANNA HAMMOND · JUNE 14, 2021 · LAST UPDATED ON MARCH 6, 2025

It's a new week and we have a new tool. This week we will review Nmap, the port scanner of choice for every security researcher. In this article, we will discuss some of the less known features of Nmap. Read on to know more.

Nmap is an open-source network mapper that uses various techniques to discover hosts and services on a network. This is the most well-known tool out there, and the one tool that every security researcher should know. Nmap can do OS detection, scan for services, check for vulnerabilities, evade firewalls, and much more.

Nmap is developed by Gordon Lyon, a respected security expert. You can find all the official documentation on his website https://nmap.org.

The installation

Nmap can simply be installed with a packet manager by doing ***sudo apt-get install nmap***. But to make sure we have the latest version, I will install it from the source. Follow the steps below if you want to do the same.

First, we need to download the latest Nmap package and extract the content. You can find the latest version by going to https://nmap.org/download.html and scroll down to the source packages. From there copy the link to the latest package (I will use the **tar.bz2** package).

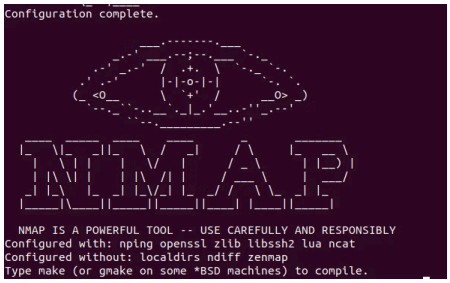
```
wget https://nmap.org/dist/nmap-7.91.tar.bz2
bzip2 -cd nmap-7.91.tar.bz2 | tar xvf –
cd nmap-7.91
```

Nmap uses a couple of extra packages to function in full. Install the following packages to make full use of Nmap.

```
sudo apt-get install g++
sudo apt-get install libssl-dev
sudo apt-get install libssh2-1-dev
```

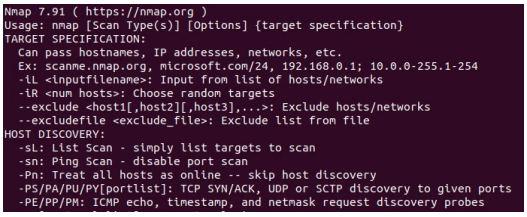
Now we can start building Nmap from the source. By default, it will install Nmap and the GUI Zenmap, If you don't want Zenmap to be installed we can add a flag to our configure command to exclude it.

```
./configure --without-zenmap
```



Now let's make and install our configured source files.

```
make
sudo make install
```



The Basics

Nmap has lots of flags and options to play with, and I will go over some of them. The most basic command is nmap <IP>. This will do a TCP SYN scan of the top 1000 ports. You can provide domain names, IP addresses, IP ranges, and entire subnets.

```
nmap 10.10.10.1        # Single IP
nmap 10.10.10.1-11     # IP range
nmap 10.10.10.0/24     # Subnet
nmap domain.com        # Hostnames
```

Providing a target list:

One of the forgotten flags is the (**-iL**) flag. With this option, we can provide a target list that nmap needs to scan.

```
nmap -iL hostnames.txt   # Provide target list
```

Host discovery techniques:

Nmap has a couple of techniques to check if hosts are online. The 2 most useful are the ping scan (**-sn**) and the "skip host discovery" (**-Pn**).

```
nmap -sn 10.10.10.0/24    # Ping sweep
nmap -Pn 10.10.10.0/24    # Don't Ping, only port scan
```

Scanning techniques:

Nmap can discover open ports with several techniques. The most useful is the stealth scan (**-sS**) (this will require root privileges). This type of scan will not finish the 3-way handshake, and therefore be more stealthy.

```
nmap -sS 10.10.10.0/24    # TCP Stealth scan
nmap -sT 10.10.10.0/24    # TCP connect scan (noisy)
nmap -sU 10.10.10.0/24    # UDP scan (most forgotten)
```

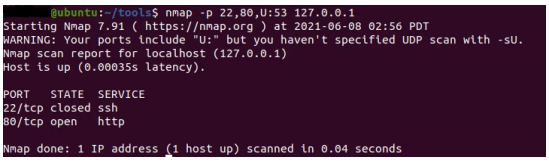
If the above scans don't give any results you can try the FIN scan. With the FIN scan, you sometimes can trick an IDS or firewall not to block your scan.

```
nmap -sF 10.10.10.0/24    # TCP FIN scan
```

Specifying ports :

Like I described before, by default Nmap only scant the top 1000 most common ports. If we want to extend the port range or scan for fewer ports we can do this with the (**-p**) flag.

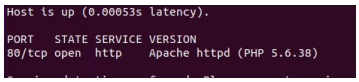
```
namp -p 22,80,443 10.10.10.1    # Scan port 22 80 and 443
nmap -p 22-80,U:53 10.10.10.1  # Scan port range and udp port 53
nmap -p- 10.10.10.1     # Scan all ports 1-65535
```



Version and Service detection:

With the version detection flag (**-sV**) we can identify service software versions. This will make life much easier as a security researcher. Keep in mind this is not always 100% correct. You can also define the intensity to detect the software version. A higher number indicates more intensive scanning.

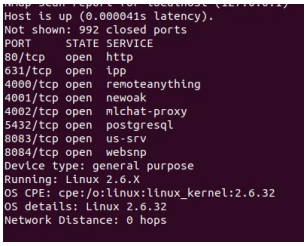
```
nmap -sV 10.10.10.0/24 --version-intensity 5    # service detection
```



OS detection:

A very powerful scan is the Operating Detection Scan (**-O**). This will try to determine the operating system behind the host. This will require root privileges.

```
nmap -O 10.10.10.1    # OS detection
```



Timings and scan speeds:

Nmap has pre-defined timing templates that you can use. They go from 0 to 5 where 5 is faster and less accurate.

```
nmap -T3 10.10.10.0/24    # Timing
```

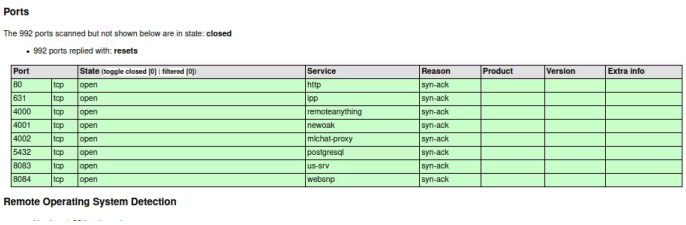
Outputting results:

Nmap can output the results in different formats. This can come in handy when you are scripting automation or need to provide a scan report.

```
nmap -oA file       # Output in the 3 major formats
nmap -oX file       # Output in xml format
nmap -oG file         # Output in greppable format
nmap -oN file         # Output in normal format
```

As an extra, you can convert the XML output to an HTML page. To do this we need to install **xsltproc**.

```
sudo apt-get install xsltproc
xsltproc export.xml -o scan.html
```



Ports

The 992 ports scanned but not shown below are in state: **closed**

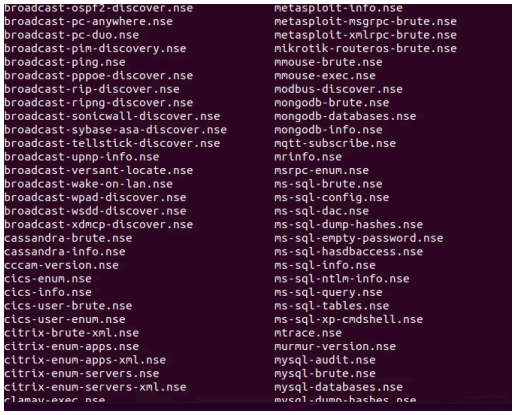
- 992 ports replied with: **resets**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
80	tcp	open	http	syn-ack			
631	tcp	open	ipp	syn-ack			
4000	tcp	open	remoteanything	syn-ack			
4001	tcp	open	newoak	syn-ack			
4002	tcp	open	mlchat-proxy	syn-ack			
5432	tcp	open	postgresql	syn-ack			
8083	tcp	open	us-srv	syn-ack			
8084	tcp	open	websnp	syn-ack			

Remote Operating System Detection

Advanced features

Nmap has a scripting engine that can run Lua scripts. This is the part where Nmap gets interesting. There are lots of pre-defined scripts that you can use. They are located in the ***/scripts*** directory.



First update the script database so we have all the latest nse scripts.

```
sudo nmap --script-updatedb
```

We can run Lua scripts by providing the (**–script**) flag following by the script or script location. It is also possible to use wildcards or comma's to run multiple scripts at once.

```
nmap –script=smb-vuln-cve-2017–7494 10.10.10.1
nmap –script=smb-vuln* 10.10.10.1
```

There are also some pre-defined keywords that group scripts together. More on this on https://nmap.org/book/nse-usage.html. Some examples are:

```
Safe
intrusive
discovery
vuln

nmap –script=vuln 10.10.10.1    # Runs all the vuln scripts
```

You can also write your own NSE scripts, but that's for another time. More information on writing your own scripts can be found on https://nmap.org/book/nse-tutorial.html.

Conclusion

Nmap is one of those tools that have been around for a long time and still bring value to your research and information gathering. Nmap has lots of options and features that probably most of you didn't know yet. It is also a perfect tool to put in your automation scripts. I hope you learned something by reading this article. See you all at the next one.