Beginner's Guide to Nmap

https://www.linux.com/tutorials/beginners-guide-nmap/

Ever wondered how attackers know what ports are open on a system? Or how to find out what services a computer is running without just asking the site admin? You can do all this and more with a handy little tool called Nmap.

What is Nmap? Short for "network mapper," nmap is a veritable toolshed of functionality to perform network scans. It can be used for security scans, simply to identify what services a host is running, to "fingerprint" the operating system and applications on a host, the type of firewall a host is using, or to do a quick inventory of a local network. It is, in short, a very good tool to know.

It's famous, too. Once you get to know Nmap a bit, you'll notice that it makes all types of cameo appearances in movies.

In this tutorial, I'll cover some of the basics of using Nmap and provide some examples you can use quickly.

Getting Nmap and Basic Use

You'll find Nmap packaged for most major Linux distros. The most recent release of Nmap came out in early 2010, so the most recent version (5.21) might not be in the current stable releases. You can find the sources and some binaries on the download page.

The basic syntax for **Nmap** is **Nmap *Scan TypeOptionstarget***. Let's say you want to scan a host to see what operating system it is running. To do this, run the following:

```
nmap -O target.host.com
```

Note that Nmap requires root privileges to run this type of scan. The scan might take a minute or so to run, so be patient. When it finishes, you'll see something like this:

```
Starting Nmap 5.21 ( http://nmap.org ) at 2010-02-27 23:52 EST
Nmap scan report for 10.0.0.1
Host is up (0.0015s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
5009/tcp  open  airport-admin
10000/tcp open  snet-sensor-mgmt
MAC Address: 00:11:24:6B:43:E2 (Apple Computer)
Device type: WAP|printer
Running: Apple embedded, Canon embedded, Kyocera embedded, Xerox embedded
OS details: VxWorks: Apple AirPort Extreme v5.7 or AirPort Express v6.3; Canon imageRUNNER
printer (5055, C3045, C3380, or C5185); Kyocera FS-4020DN printer; or Xerox Phaser 8860MFP
printer
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.21 seconds
```

As you can see, **Nmap** provides a lot of data. Here it takes a guess at the operating system that might be running on the system. I ran this particular scan against an Apple Airport Extreme router. As an added bonus, **Nmap** tells me that the device is one hop away, the MAC address of the device and manufacturer of the NIC, the open ports, and how long the scan took.

Here's the result of another scan, against a desktop machine running Ubuntu 9.10:

```
Starting Nmap 5.21 ( http://nmap.org ) at 2010-02-28 00:00 EST
Nmap scan report for 10.0.0.6
Host is up (0.0039s latency).
Not shown: 999 closed ports
```

```
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:17:08:2A:D6:F0 (Hewlett Packard)
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.19 - 2.6.31
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.40 seconds
```

Here we see that the system has an HP NIC (it's an HP workstation), running the Linux kernel somewhere between Linux 2.6.19 and 2.6.31. You may not be able to get an explicit identification of the operating system down to the version of Linux.

Practice Hosts

In the examples above, I chose a local router and one of my workstations in part because I have the permission to scan them. You can use Nmap to scan virtually any host. However, it's a bad idea to run many scans against hosts you're not in control of or don't have permission to scan. The Nmap folks have a test host at scanme.nmap.org that can be used for testing, so long as you're not running any tests of exploits or Denial of Service (DoS) attacks.

Some admins don't appreciate unexpected scans, so use best judgment and restrict scans to hosts that are on your own network or that you have permission to scan. It may also be against your ISP's terms of service to use some of Nmap's more aggressive scan features, so be careful out there!

Multiple Hosts

You can scan more than one host at a time using nmap. If you're using IP addresses, you can specify a range like 10.0.0.1-6 or a range like 10.0.0.0/24. The 10.0.0.1-6 would scan hosts 10.0.0.1, 10.0.0.2, 10.0.0.3 through 10.0.0.6. Using the /24 notation would scan the whole range of hosts from 10.0.0.0 to 10.0.0.255. For example, to scan 10.0.0.1 through 10.0.0.42 to learn what OS they might be running I'd use **nmap -O 10.0.0.1-42**.

If you have hostnames instead of IP addresses, you can separate them with a space on the command line, like so:

```
nmap -O host1.target.com host2.target.com
```

Checking Open Ports

If you give **Nmap** no options at all and just point it at a given host it will scan for open ports and report back those that are open, and what service is running on them. For instance, running **nmap *target.hostname.com*** might yield something like this:

```
Interesting ports on target.hostname.com (10.0.0.88):
Not shown: 1711 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
3306/tcp open  mysql

Nmap done: 1 IP address (1 host up) scanned in 0.228 seconds
```

Here you can see that there are three ports open: 22, 80, and 3306 which run SSH, HTTP, and MySQL respectively. Nmap recognizes six states: open, closed, filtered, unfiltered, open|filtered, and closed|filtered. These are mostly self-explanatory. See the Nmap docs for more on these states. If Nmap can tell what service is running, it will report it under the SERVICE column.

If you'd like a little more information, crank it up a notch by adding one or two **-v** options to the command. For example, using **nmap -vv host.target.com** would produce something like this:

```
Initiating Ping Scan at 11:44
Scanning 10.0.0.28 [1 port]
Completed Ping Scan at 11:44, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:44
Completed Parallel DNS resolution of 1 host. at 11:44, 0.00s elapsed
Initiating Connect Scan at 11:44
Scanning host.target.com (10.0.0.28) [1714 ports]
Discovered open port 22/tcp on 10.0.0.28
Discovered open port 80/tcp on 10.0.0.28
Discovered open port 3306/tcp on 10.0.0.28
Completed Connect Scan at 11:44, 0.08s elapsed (1714 total ports)
Host host.target.com (10.0.0.28) appears to be up ... good.
Interesting ports on host.target.com (10.0.0.28):
Not shown: 1711 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
3306/tcp open  mysql

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.104 seconds
```

Nmap provides a lot more information when using the verbose (**-v**) option.

Service Scans

If you're really curious about what services a host might be running, try the **-sV** options. This will do a more aggressive scan to try to figure out what versions of services are running on a given host, and also might help determine more specifically what OS a host is running. For instance, I ran **nmap -sV** against a test server and got this in response:

Starting Nmap 5.21 (http://nmap.org) at 2010-02-28 00:15 EST

```
Nmap scan report for test.host.net (XX.XXX.XXX.XX)
Host is up (0.090s latency).
Not shown: 965 closed ports, 33 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 4.7p1 Debian 8ubuntu1.2 (protocol 2.0)
80/tcp open  http    Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.10 with Suhosin-Patch)
Service Info: OS: Linux

Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.43 seconds
```

As you can see, Nmap can "fingerprint" the packets and identify the versions of the software running on the SSH and HTTP ports. Here you can see that the system being pinged is a Ubuntu box with Apache 2.2.8 and OpenSSH 4.7p1. This can be useful for a number of reasons. A quick Nmap scan can identify systems that are running unpatched systems and therefore ones that might be vulnerable to known exploits.

What's on My Network?

Not quite sure what might be running on your network? Try using **nmap -sP**, which will run a ping scan on the specified network. For instance, **nmap -sP 10.0.0.0/24** will scan the 256 hosts from 10.0.0.0 through 10.0.0.255 to see if they're available, and report back. You can also use a range, such as **nmap -sP 10.0.0.1-15**.

Zenmap

Finally, if all this command line fun is not your bag, Nmap has a GUI that you can use to build and execute commands. Called Zenmap, the GUI will let you specify targets, run scans, display the results and even save and compare them against one another.

When you open Zenmap, you can give it a target to scan and select one of the profile scans to get started. It includes your basic ping scan, quick scans, some more intense scans that include UDP services, and so forth. The Zenmap GUI is a good way to get acquainted with Nmap, but it's also a good idea to know how to use Nmap from the command line if you're going to be working with it often.

In a future tutorial we'll take a more in-depth look at Nmap and specific tasks you might want to do with Nmap. I hope this overview gave a good sense what Nmap can do and helps you get started working with Nmap.