Nmap Commands – 17 Basic Commands for Linux Network

May 14, 2019 COMMANDS LINUX NMAP

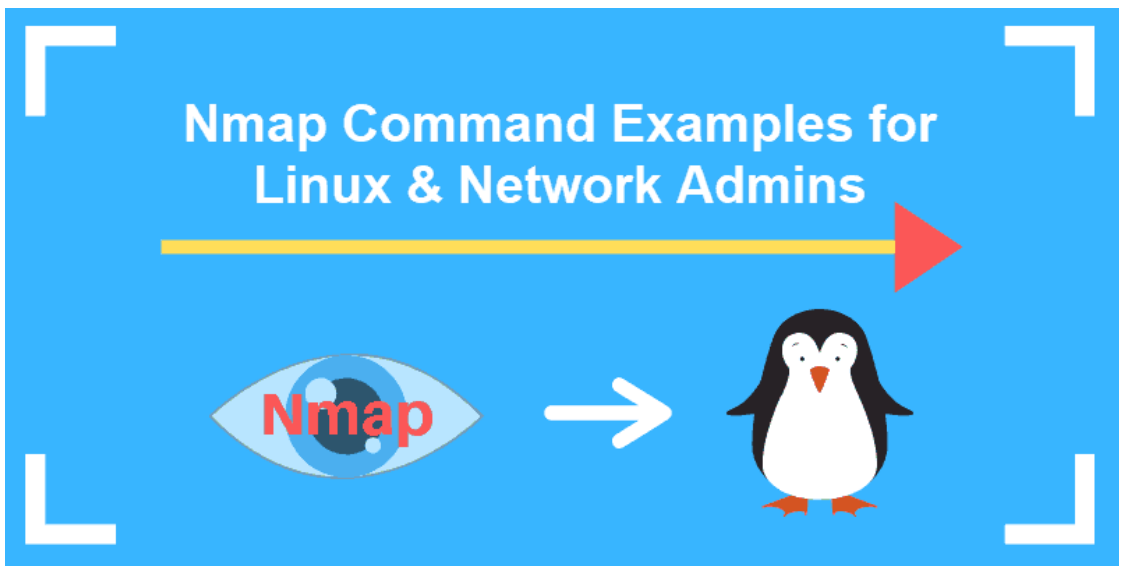
Contents

Updated on August 26, 2022.

Introduction

Nmap is one of the oldest and most flexible networking tools. Network administrators use Nmap to discover, analyze, and map networks under various conditions. The feature-rich command-line tool is essential from a security and troubleshooting perspective.

This article explains what Nmap is and showcases 17 basic commands for Linux.



What is Nmap?

Nmap (**N**etwork **map**per) is an open-source Linux tool for network and security auditing. The tool helps network administrators reveal hosts and services on various systems.

Nmap works both locally and remotely. Typical uses include scanning for open ports, discovering vulnerabilities in a network, network mapping, and maintenance. The tool is valuable from both a security and networking standpoint.

Nmap Commands

The `nmap` command comes with many options and use cases depending on the situation at hand. Below are some of the most common and useful `nmap` **commands in Linux with examples**.

Note: If you don't have Network Mapper, you can install the software by following our guide on how to install NMAP on Ubuntu.

1. Nmap Command to Scan for Open Ports

When scanning hosts, Nmap commands can use server names, IPV4 addresses or IPV6 addresses. A basic Nmap command will produce information about the given host.

```
nmap subdomain.server.com
```

Without flags, as written above, Nmap reveals open services and ports on the given host or hosts.

```
nmap 192.168.0.1
```

Nmap can reveal open services and ports by IP address as well as by domain name.

```
nmap -F 192.168.0.1
```

If you need to perform a scan quickly, you can use the `-F` flag. The `-F` flag will list ports on the *nmap-services* files. Because the `-F` "Fast Scan" flag does not scan as many ports, it isn't as thorough.

Note: Learn about other methods you can use to check for open ports in Linux.

2. Scan Multiple Hosts

Nmap can scan multiple locations at once rather than scanning a single host at a time. This is useful for more extensive network infrastructures. There are several ways to scan numerous locations at once, depending on how many locations you need to examine.

Add multiple domains or multiple IP addresses in a row to scan multiple hosts at the same time.

```
nmap 192.168.0.1 192.168.0.2 192.168.0.3
```

Use the `*` wildcard to scan an entire subnet at once.

```
nmap 192.168.0.*
```

Separate different address endings with commas rather than typing out the entire IP address.

```
nmap 192.168.0.1,2,3
```

Use a hyphen to scan a range of IP addresses.

```
nmap 192.168.0.1-4
```

3. Excluding Hosts from Search

When scanning a network, you may want to select an entire group (such as a whole subnet) while excluding a single host.

```
nmap 192.168.0.* --exclude 192.168.0.2
```

You can exclude certain hosts from your search using the **`-exclude`** flag.

```
nmap 192.168.0.* --excludefile /file.txt
```

You can also exclude a list of hosts from your search using the **`-exclude`** flag and linking to a specific file. This is the easiest way to exclude multiple hosts from your search.

4. Scan to Find out OS Information

In addition to general information, Nmap can also provide operating system detection, script scanning, traceroute, and version detection. It's important to note that Nmap will do its best to identify things like operating systems and versions, but it may not always be entirely accurate.

Add in the **`-A`** flag on your Nmap command, so you can discover the operating system information of the hosts that are mapped.

```
nmap -A 192.168.0.1
```

The **`-A`** flag can be used in combination with other Nmap commands.

Using the **`-O`** flag on your Nmap command will reveal further operating system information of the mapped hosts. The **`-O`** flag enables OS detection.

```
nmap -O 192.168.0.1
```

Additional tags include **`-osscan-limit`** and **`-osscan-guess`**.

The **`-osscan-limit`** command will only guess easy operating system targets. The **`-osscan-guess`** command will be more aggressive about guessing operating systems. Again, operating systems are detected based on certain hallmarks: it isn't a certainty that the information is accurate.

5. Scan to Detect Firewall Settings

Detecting firewall settings can be useful during penetration testing and vulnerability scans. Several functions can be used to detect firewall settings across the given hosts, but the **`-sA`** flag is the most common.

```
nmap -sA 192.168.0.1
```

Using the **`-sA`** flag will let you know whether a firewall is active on the host. This uses an ACK scan to receive the information.

Note: Learn more about penetration testing types and methodologies and penetration testing software in our guides.

6. Find Information About Service Versions

At times, you may need to detect service and version information from open ports. This is useful for troubleshooting, scanning for vulnerabilities, or locating services that need to be updated.

```
nmap -sV 192.168.0.1
```

This will give you the necessary information regarding the services across the given host.

You can use `--version-intensity` level from `0` to `9` to determine the intensity level of this search. You can also use `--version-trace` to show more detailed information of the scan if the scan does not come out with the results that you would ordinarily expect.

7. Scan for Ports

Port scanning is one of the basic utilities that Nmap offers and consequently, there are a few ways that this command can be customized.

With the `-p` flag followed by a port, you can scan for information regarding a specific port on a host.

```
nmap -p 443 192.168.0.1
```

By adding a type of port before the port itself, you can scan for information regarding a specific type of connection.

```
nmap -p T:8888,443 192.168.0.1
```

You can scan for multiple ports with the `-p` flag by separating them with a comma.

```
nmap -p 80,443 192.168.0.1
```

You can also scan for multiple ports with the `-p` flag by marking a range with the hyphen.

```
nmap -p 80-443 192.168.0.1
```

To scan ports in order rather than randomly, add the flag `-r` to the command. You can also use the command `--top-ports` followed by a number to find the most common ports, up to that amount.

8. Complete a Scan in Stealth Mode

If it is necessary to complete a stealthy scan, use the following Nmap command:

```
nmap -sS 192.168.0.1
```

Using the `-sS` flag will initiate a stealth scan with TCP SYN. The `-sS` flag can be used in conjunction with other types of Nmap commands. However, this type of scan

is slower and may not be as aggressive as other options.

9. Identify Hostnames

There are a few ways you can implement host discovery through Nmap. The most common of which is through `-sL`. For example:

```
nmap -sL 192.168.0.1
```

The `-sL` flag will find the hostnames for the given host, completing a DNS query for each one. Additionally, the `-n` option can be used to skip DNS resolution, while the `-R` flag can be used to always resolve DNS. The `-Pn` flag will skip host discovery entirely, instead of treating hosts as though they are online regardless.

10. Scan from a File

If you have a long list of addresses that you need to scan, you can import a file directly through the command line.

```
nmap -iL /file.txt
```

This will produce a scan for the given IP addresses. In addition to scanning those IP addresses, you can also add other commands and flags. This is useful if there is a set of hosts that you often need to reference.

11. Get More Information with Verbose

A verbose output generally gives you far more information regarding a command. Sometimes this output is unnecessary. However, if you're debugging a particularly tricky situation or you want more information, you can set the given command to verbose mode.

```
nmap -v 192.168.0.1
```

The `-v` flag will provide additional information about a completed scan. It can be added to most commands to give more information. Without the `-v` flag, Nmap will generally return only the critical information available.

12. Scan IPv6 Addresses

IPv6 is becoming more commonplace, and Nmap supports it just as it supports domains and older IP addresses. IPv6 works with any of the available Nmap commands. But, a flag is required to tell Nmap that an IPv6 address is being referenced.

```
nmap -6 ::ffff:c0a8:1
```

Use the `-6` option with other flags to perform more complicated Nmap functions with IPv6.

13. Scan to See Which Servers are Active

One of the most simple abilities for Nmap is the ability to ping active machines. The `-sP` command locates machines, make sure that machines are responding, or identifies unexpected machines across a network.

```
nmap -sP 192.168.0.0/24
```

The `-sP` command will produce a list of which machines are active and available.

14. Find Host Interfaces, Routes, and Packets

It may become necessary to find host interfaces, print interfaces, and routes to debug.

To do this, use the `--iflist` command:

```
nmap --iflist
```

The `--iflist` command will produce a list of the relevant interfaces and routes.

```
nmap --packet-trace
```

Similarly, `--packet-trace` will show packets sent and received, providing similar value for debugging.

15. Aggressive Scans and Timings

Sometimes you may need to scan more aggressively or want to run a quick scan. You can control this through the use of the timing mechanisms. In Nmap, timing controls both the speed and the depth of the scan.

```
nmap -T5 192.168.0.1
```

An aggressive scan is going to be faster, but it also could be more disruptive and inaccurate. There are other options such as `T1`, `T2`, `T3`, and `T4` scans. For most scans, `T3` and `T4` timings are sufficient.

16. Get Some Help

If you have any questions about Nmap or any of the given commands, you can use a tag to get context-based information.

```
nmap -h
```

The `-h` tag will show the help screen for Nmap commands, including giving information regarding the available flags.

17. Create Decoys While Scanning

Nmap can also be used to create decoys, which are intended to fool firewalls. While decoys can be used for nefarious purposes, it's generally used to debug.

```
nmap -D 192.168.0.1,192.168.0.2,...
```

When using the `-D` command, you can follow the command with a list of decoy addresses. These decoy addresses will also show as though they are scanning the network, to obfuscate the scan that is actually being done.

Similarly, it's possible to use commands such as `--spoof-mac` to spoof an Nmap MAC address, as well as the command `-S` to spoof a source address.

Conclusion

With the right Nmap commands, you can quickly find out information about ports, routes, and firewalls.

Nmap has several settings and flags for a system administrator to explore. In addition to being able to run in a cloaked mode, initiate decoys, and aggressively and quickly scan for potential vulnerabilities.

Dejan Tucakov

Dejan is the Head of Content at phoenixNAP with over 8 years of experience in Web publishing and technical writing. Prior to joining PNAP, he was Chief Editor of several websites striving to advocate for emerging technologies. He is dedicated to simplifying complex notions and providing meaningful insight into data center and cloud technology.

Next you should read

Was this article helpful? Yes No

Security, SysAdmin

How to Install and Use Nmap Network Scanner on Linux

April 8, 2024

Learn how to install Nmap on a Linux based system and use it to retrieve valuable information from remote ...

READ MORE

Networking, SysAdmin

Linux Ping Command Tutorial with Examples

April 2, 2024

Most Linux users are familiar with the ping command and know how to use it in its basic form. However, there ...

READ MORE

SysAdmin

How to Use mkdir Command to Make or Create a Linux Directory

December 1, 2023

The mkdir command in Linux allows users to create or make new directories. mkdir stands for make directory ...

READ MORE

SysAdmin

How To Use grep Command In Linux/UNIX

February 29, 2024

This guide details the most useful grep commands for Linux / Unix systems. After going through all the ...

READ MORE