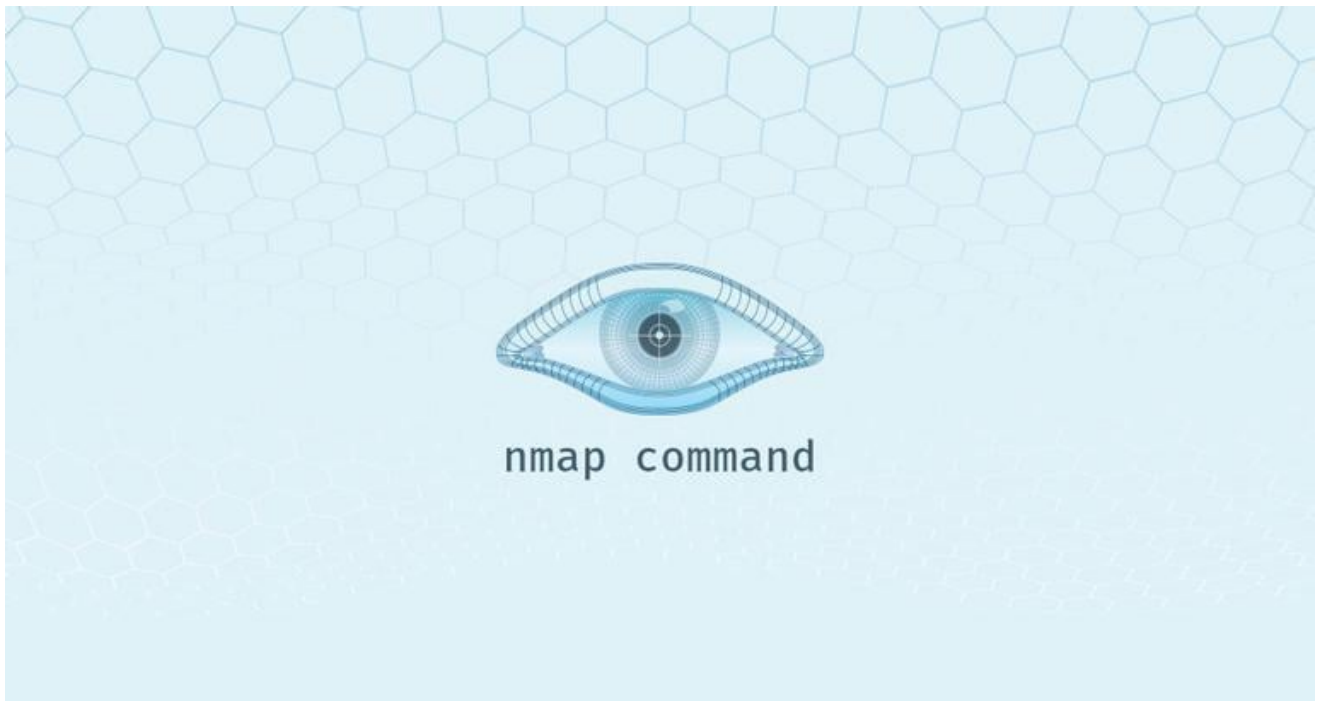
How to Use the nmap Command

Published on Dec 16, 2020 • 8 min read



RELATED TUTORIALS

How to Check (Scan) for Open Ports in Linux

Bash read Command

Listing Linux Services with Systemctl

Chattr Command in Linux (File Attributes)

Basename Command in Linux

who Command in Linux

Rmmod Command in Linux

Nmap is a powerful network scanning tool for security audits and penetration testing. It is one of the essential tools used by network administrators to troubleshooting network connectivity issues and port scanning .

Nmap can also detect the Mac address, OS type , service version, and much more.

This article explains the basics of how to use the `nmap` command to perform various network tasks.

Installing Nmap

Nmap is a multi-platform program that can be installed on all major operating systems. It was initially released as a Linux-only tool, and later it was ported to other systems such as BSD, Windows, and macOS.

If you prefer a GUI over the command line, Nmap also has a graphical user interface called Zenmap .

The official binary packages are available for download from the Nmap download page .

The installation procedure is straightforward and varies according to your operating system.

Installing Nmap on Ubuntu and Debian

Nmap is available from the default Ubuntu and Debian repositories. To install it, run:

```
$ sudo apt update
$ sudo apt install nmap
```

Installing Nmap on CentOS and Fedora

```
$ sudo dnf install nmap
```

Installing Nmap on macOS

macOS users can install Nmap by downloading the ".dmg" installation package from the Nmap site or via Homebrew:

```
$ brew install nmap
```

Installing Nmap on Windows

The Windows version of Nmap has some limitations, and it is generally a little slower than the UNIX version.

The easiest option to install Nmap on Windows is to download and run the self-installation exe file.

You can run Nmap on Windows either from the command line or by launching the Zenmap program. For more information about how to use Nmap on Windows, check the post-install usage instructions .

Using Nmap

Nmap is typically used to audit network security, network mapping, identify open ports, and search for online devices.

The simplified syntax of the `nmap` command is as follows:

```
nmap [Options] [Target...]
```

The most basic example of using Nmap is to scan a single target as a standard user without specifying any options:

```
$ nmap scanme.nmap.org
```

When invoked as a non-root user that does not have raw packet privileges, `nmap` runs TCP connect scan. The (`-sT`) is turned on by default in unprivileged mode.

The output will look something like this, including basic information about the scan and a list of open and filtered TCP ports.

```
Output

Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-16 20:19 CET
Nmap scan report for cast.lan (192.168.10.121)
Host is up (0.048s latency).
Not shown: 981 closed ports
PORT     STATE    SERVICE
21/tcp   open     ftp
22/tcp   open     ssh
25/tcp   open     smtp
53/tcp   open     domain
```

```
443/tcp  open     https
587/tcp  open     submission
993/tcp  open     imaps
995/tcp  open     pop3s
1025/tcp open     NFS-or-IIS
1080/tcp open     socks
8080/tcp open     http-proxy
8081/tcp open     blackice-icecap

Nmap done: 1 IP address (1 host up) scanned in 1.78 seconds
```

The most popular scan option is the TCP SYN scan (`-sS`) that is faster than the connect option and works against all compliant TCP stacks.

`-sS` is turned on by default when `nmap` is invoked as a user with administrative privileges:

```
$ sudo nmap 192.168.10.121
```

For more detailed output, use the increase the verbosity with `-v` or `-vv` :

```
$ sudo nmap -vv 192.168.10.121
```

To perform a UDP scan, invoke the command with the (`-sU`) option as a root user:

```
$ sudo nmap -sU 192.168.10.121
```

For a complete list of port scanning methods, visit the Nmap documentation page .

Nmap also supports IPv6 addresses. To specify an IPv6 host use the `-6` option:

```
$ sudo nmap -6 fd12:3456:789a:1::1
```

Specifying Target Hosts

Nmap treats all arguments that are not options as target hosts.

> Arguments are considered options if they begin with a single or double dash (`-` , `--`).

The simplest option is to pass one or more target addresses or domain names:

```
$ nmap 192.168.10.121 host.to.scan
```

You can use the CIDR notation to specify a network range:

```
$ nmap 192.168.10.0/24
```

```
$ nmap 192.168.10-12.1
```

Another character you can use to specify the targets is the comma. The following command targets the same hosts as the one above:

```
$ nmap 192.168.10,11,12.1
```

You can combine all forms:

```
$ nmap 10.8-10.10,11,12.0/28  192.168.1-2.100,101
```

To make sure you specified the correct hosts before scanning, use the list scan option (`-sL`), which only lists the targets without running a scan:

```
$ nmap -sL 10.8-10.10,11,12.0/28  192.168.1-2.100,101
```

If you want to exclude targets that are included in the range you specified, use the `--exclude` option:

```
$ nmap 10.8-10.10,11,12.0/28 --exclude 10.10.12.12
```

Specifying and Scanning Ports

By default, Nmap performs a quick scan for the 1000 most popular ports. These ports are not the first 1000 consecutive ports, but the 1000 most commonly used ports ranging from 1 to 65389.

To scan for all ports from 1 through 65535, use the `-p-` option:

```
$ nmap -p- 192.168.10.121
```

Each port can be in one of the following states:

- open - The program running on the port responds to request.
- closed - No program runs on the port, and the host reply to requests.
- filtered - The host doesn’t reply to the request.
-

Ports and port ranges are specified with the `-p` option.

For example, to scan only port 443, you would use the following command:

```
$ nmap -p 443 192.168.10.121
```

To specify more than one port, separate the target ports with a comma:

```
$ nmap -p 80,443 192.168.10.121
```

```
$ sudo nmap -sU -p 1-1024 192.168.10.121
```

All combined:

```
$ nmap -p 1-1024,8080,9000 192.168.10.121
```

Ports can also be specified using the port name. For example, to scan for port 22, ssh, you can use:

```
$ nmap -p ssh 192.168.10.121
```

Ping Scanning

To perform a ping scanning or host discovery, invoke the `nmap` command with the `-sn` option:

```
$ sudo nmap -sn 192.168.10.0/24
```

The `-sn` option tells Nmap only to discover online hosts and not to do a port scan. This is useful when you want to quickly determine which of the specified host are up and running.

Disabling DNS Name Resolution

Nmap's default behavior is to perform reverse-DNS resolution for each discovered host, which increases the scan time.

When scanning large networks, it is a good idea to disable reverse-DNS resolution and speed up the scans. To do that, invoke the command with the `-n` option:

```
$ sudo nmap -n 192.168.10.0/16
```

OS, Service and Version Detection

Nmap can detect the remote host operating system using TCP/IP stack fingerprinting. To run OS detection, invoke the command with the `-O` option:

```
$ sudo nmap -O scanme.nmap.org
```

If Nmap can detect the host OS, it will print something like below:

```
Output

...
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5
```

```
OS detection performed. Please report any incorrect results at http
Nmap done: 1 IP address (1 host up) scanned in 26.47 seconds
```

Typically, system services listen on standard ports that are well known and reserved for them. For example, if port 22 that corresponds to the SSH service is open, you'll assume that an SSH server runs on the host. However, you cannot be absolutely sure because people can run services on whatever ports they want.

With service and version detection, Nmap will show you what program listens on the port and the program version.

To scan for service and version, use the `-sV` option:

```
$ sudo nmap -sV scanme.nmap.org
```

```
Output
...
PORT      STATE    SERVICE      VERSION
19/tcp    filtered chargen
22/tcp    open     ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13
80/tcp    open     http         Apache httpd 2.4.7 ((Ubuntu))
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
9929/tcp  open     nping-echo   Nping echo
31337/tcp open     tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
...
```

You can also scan for OS, Versions, and run traceroute in one command using the `-A` option:

```
$ sudo nmap -A 192.168.10.121
```

Nmap Output

By default, Nmap prints the information to standard output (stdout).

If you scan a large network or need the information for later usage, you can save the output to a file.

Nmap provides several output types. To save the output in normal format, use the `-oN` option followed by the file name:

```
$ sudo nmap -sU -p 1-1024 192.168.10.121 -oN output.txt
```

The most popular option is to save the output in XML format. To do so, use the `-oX` option:

```
$ sudo nmap -sU -p 1-1024 192.168.10.121 -oX output.xml
```

with the `-oG` option:

```
$ sudo nmap -sU -p 1-1024 192.168.10.121 -oG output
```

Nmap Scripting Engine

One of the most powerful features of Nmap is its scripting engine. Nmap ships with hundreds of scripts , and you can also write your own scripts in the Lua language.

You can use scripts to detect malware and backdoors, perform brute-force attacks, and more.

For example, to check if a given host is compromised you can use:

```
$ nmap -sV --script http-malware-host scanme.nmap.org
```

Conclusion

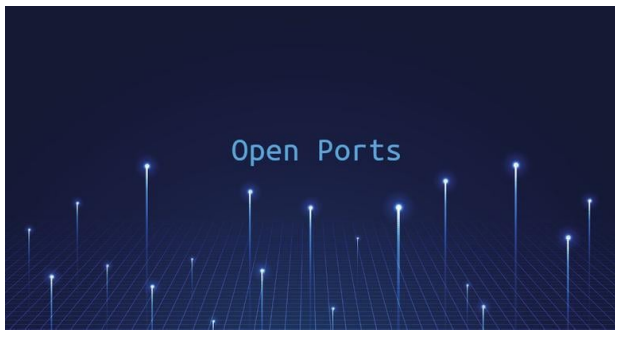
Nmap is an open-source tool that is used primarily by network administrators to discover host and scan ports.

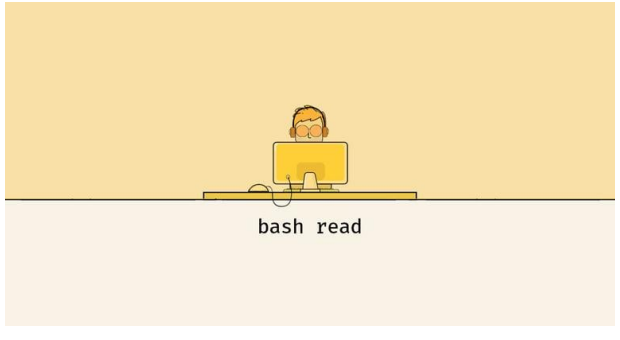
Please note that in some countries, it is not legal to scan networks without authorization.

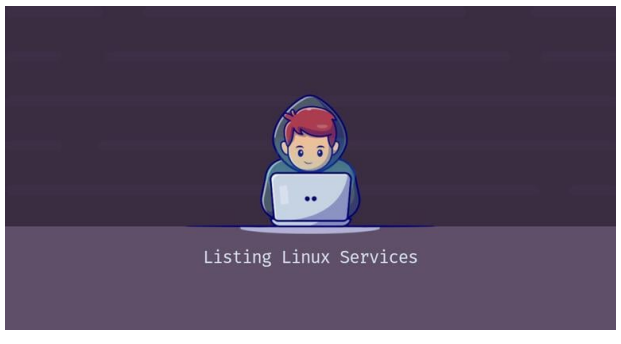
If you have any questions or remarks, please leave a comment below.

terminal nmap

Related Articles







Write a comment