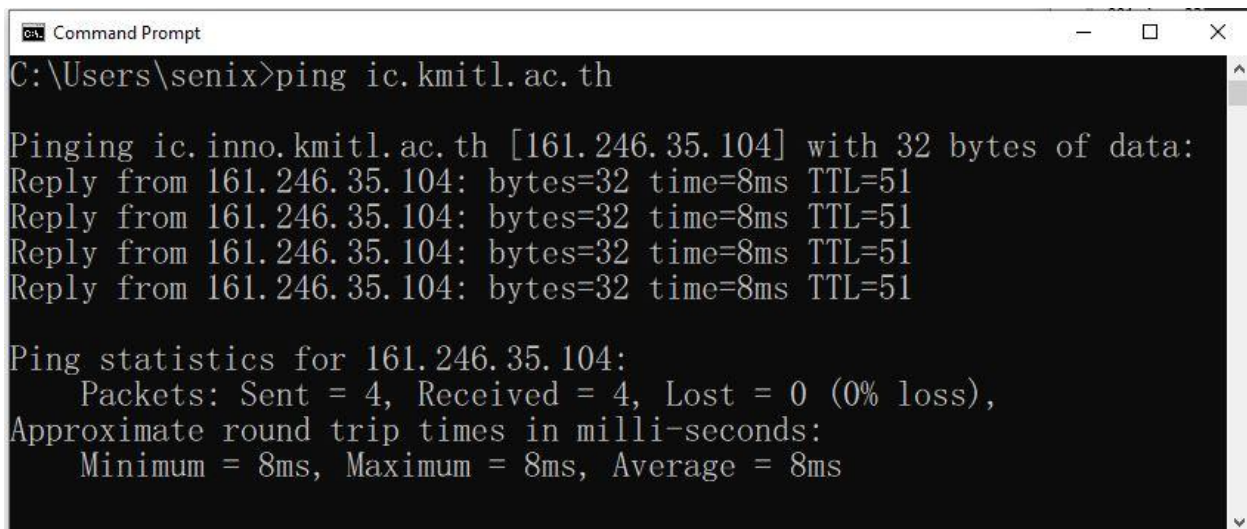


Name Audthaneer Supeeramongkolkul Student-ID 66011525

## Chapter 3 (Week 3-4-5)

### Ping

The ping command sends packets of data to a specific IP address on a network, and then lets you know how long it took to transmit that data and get a response. It's a handy tool that you can use to quickly test various points of your network. Here's how to use it. let's look at an example.” ping ic.kmitl.ac.th “.



```
ca Command Prompt
C:\Users\senix>ping ic.kmitl.ac.th

Pinging ic.inno.kmitl.ac.th [161.246.35.104] with 32 bytes of data:
Reply from 161.246.35.104: bytes=32 time=8ms TTL=51
Reply from 161.246.35.104: bytes=32 time=8ms TTL=51
Reply from 161.246.35.104: bytes=32 time=8ms TTL=51
Reply from 161.246.35.104: bytes=32 time=8ms TTL=51

Ping statistics for 161.246.35.104:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 8ms, Maximum = 8ms, Average = 8ms
```

### How Does Ping Work?

Ping comes from a term used in sonar technology that sends out pulses of sound, and then listens for the echo to return. On a computer network, a ping tool is built into most operating systems that works in much the same way. You issue the ping command along with a specific URL or IP address. Your computer sends several packets of information out to that device, and then waits for a response. When it gets the response, the ping tool shows you how long each packet took to make the round trip—or tells you there was no reply.

It sounds simple, and it is. But you can use it to good effect. You can test whether your computer can reach another device—like your router—on your local network, or whether it can reach a device on the Internet. This can help you determine if a network problem is somewhere on your local network, or somewhere beyond. The time it takes packets to return to you can help you identify a slow connection, or if you're experiencing packet loss.

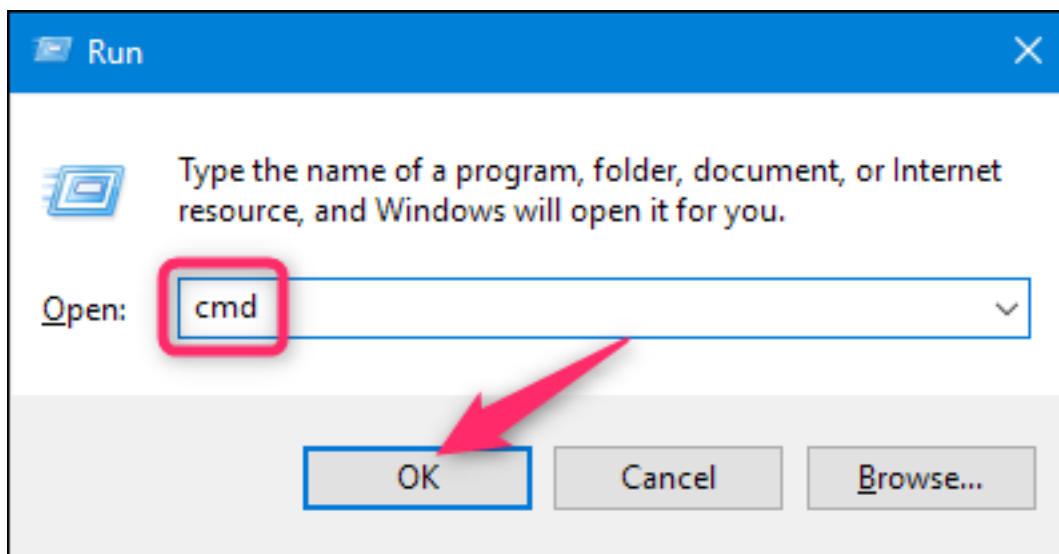
**Name** \_\_\_\_\_ **Student-ID** \_\_\_\_\_

And it pretty much doesn't matter what operating system you're using. Pull up a terminal or Command Prompt window, and you can use ping on macOS, Linux, or any version of Windows.

#### How to Use Ping

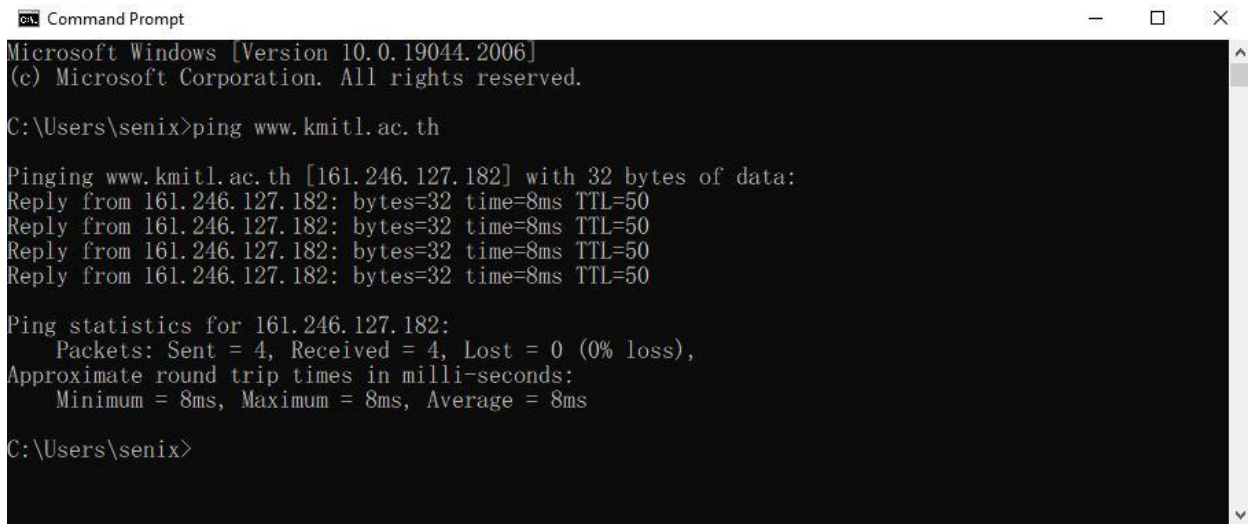
We're going to use the Windows Command Prompt in our example here. But you can also use the ping command in Windows PowerShell, or in the Terminal app on macOS or any Linux distro. Once you get to using the actual command, it works the same everywhere.

In Windows, hit Windows+R. In the Run window, type "cmd" into the search box, and then hit Enter.



At the prompt, type "ping" along with the URL or IP address you want to ping, and then hit Enter. In the image below, we're pinging ww.kmitl.ac.th and getting a normal response.

Name \_\_\_\_\_ Student-ID \_\_\_\_\_



```
Command Prompt
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\senix>ping www.kmitl.ac.th

Pinging www.kmitl.ac.th [161.246.127.182] with 32 bytes of data:
Reply from 161.246.127.182: bytes=32 time=8ms TTL=50
Reply from 161.246.127.182: bytes=32 time=8ms TTL=50
Reply from 161.246.127.182: bytes=32 time=8ms TTL=50
Reply from 161.246.127.182: bytes=32 time=8ms TTL=50

Ping statistics for 161.246.127.182:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 8ms, Maximum = 8ms, Average = 8ms

C:\Users\senix>
```

That response shows the URL you're ping, the IP address associated with that URL, and the size of the packets being sent on the first line. The next four lines show the replies from each individual packet, including the time (in milliseconds) it took for the response and the time-to-live (TTL) of the packet, which is the amount of time that must pass before the packet is discarded.

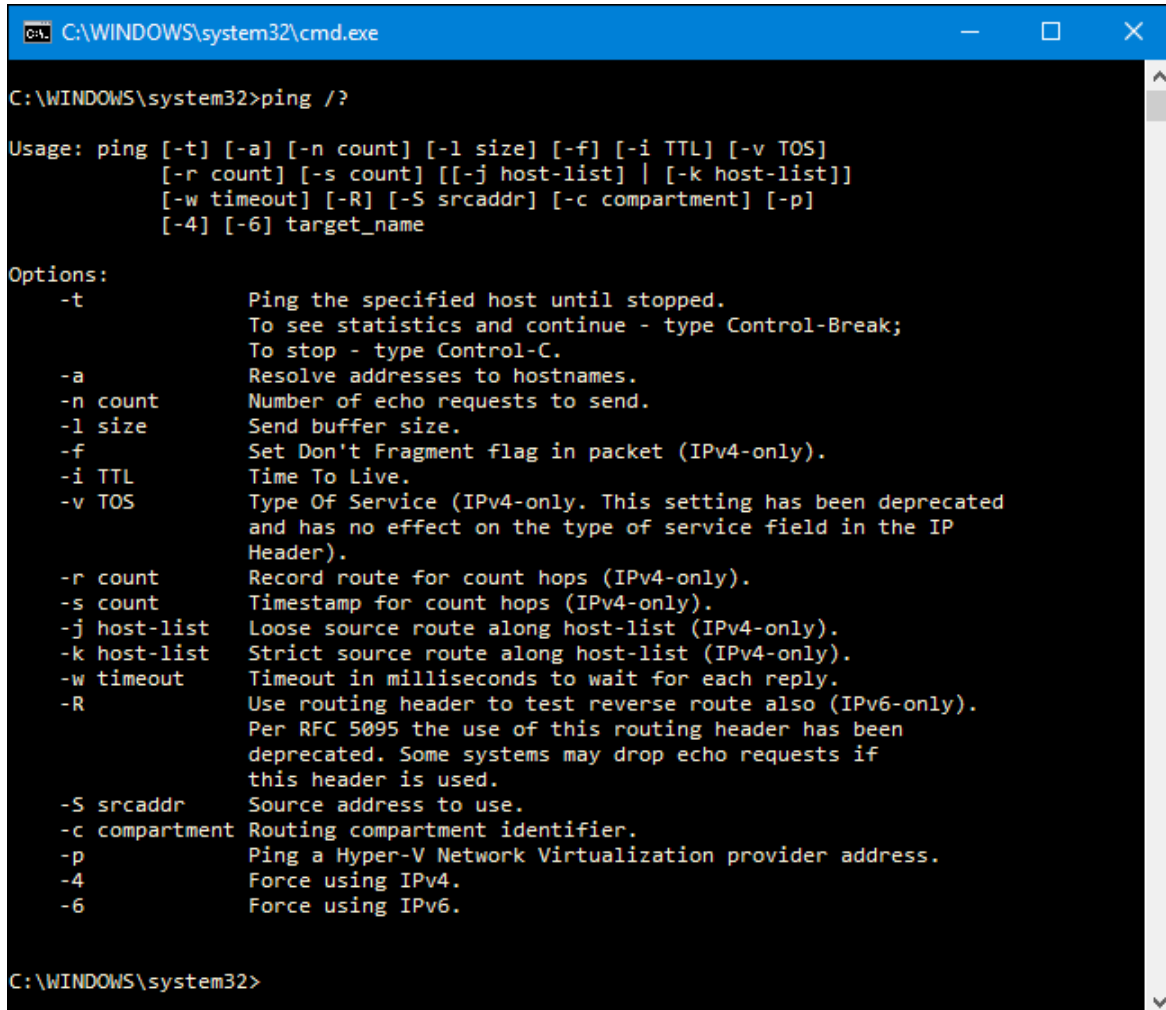
At the bottom, you'll see a summary that shows how many packets were sent and received, as well as the minimum, maximum, and average response time.

And in the next image, we're ping, the router on our local network using its IP address. We're also getting a normal response from it.

And that's how to use ping at its most basic. Of course, like most commands, there are some advanced switches you can use to make it behave a bit differently. For example, you can have it keep ping, specify the number of times you want it to ping, set how often it should ping, and more. But unless you're doing some very specific types of troubleshooting, you won't need to worry much about those advanced switches.

If you're curious about them, though, just type "ping /?" at the Command Prompt to see a list.

Name \_\_\_\_\_ Student-ID \_\_\_\_\_



```
C:\WINDOWS\system32\cmd.exe

C:\WINDOWS\system32>ping /?

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
          [-r count] [-s count] [[-j host-list] | [-k host-list]]
          [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]
          [-4] [-6] target_name

Options:
    -t                Ping the specified host until stopped.
                      To see statistics and continue - type Control-Break;
                      To stop - type Control-C.
    -a                Resolve addresses to hostnames.
    -n count           Number of echo requests to send.
    -l size            Send buffer size.
    -f                Set Don't Fragment flag in packet (IPv4-only).
    -i TTL             Time To Live.
    -v TOS             Type Of Service (IPv4-only. This setting has been deprecated
                      and has no effect on the type of service field in the IP
                      Header).
    -r count           Record route for count hops (IPv4-only).
    -s count           Timestamp for count hops (IPv4-only).
    -j host-list       Loose source route along host-list (IPv4-only).
    -k host-list       Strict source route along host-list (IPv4-only).
    -w timeout         Timeout in milliseconds to wait for each reply.
    -R                Use routing header to test reverse route also (IPv6-only).
                      Per RFC 5095 the use of this routing header has been
                      deprecated. Some systems may drop echo requests if
                      this header is used.
    -S srcaddr         Source address to use.
    -c compartment     Routing compartment identifier.
    -p                Ping a Hyper-V Network Virtualization provider address.
    -4                Force using IPv4.
    -6                Force using IPv6.

C:\WINDOWS\system32>
```

So, What Can You Do With Ping?

Now that you know how to use the command, here are some interesting things you can do with it:

Ping a URL (like [www.kmitl.ac.th](http://www.kmitl.ac.th)) or IP address to see if you can reach an internet destination. If you get a successful response, you know that all the networking devices between you and that destination are working, including the network adapter in your computer, your router, and whatever devices exist on the internet between your router and the destination. And if you're interested in exploring those routes further, you can use another networking tool named `tracert` to do just that.

Ping a URL to resolve its IP address. If you want to know the IP address for a particular URL, you can ping the URL. The ping tool shows you right at the top the IP address it's working with.

Ping your router to see if you can reach it. If you can't successfully ping an internet location, you can then try pinging your router. A successful response lets you know that your local network is working okay, and that the problem reaching the internet location is somewhere out of your control.

Name Audthaneer Supeeramongkolkul Student-ID 66011525

Ping your loopback address (127.0.0.1). If you can't successfully ping your router, but your router appears to be turned on and working, you can try pinging what's known as a loopback address. That address is always 127.0.0.1, and pinging it successfully lets you know that the network adapter on your computer (and the networking software in your OS) is working properly.

### Lab Ping explains what they means. Number 1 – 12

```

C:\Users\senix>ping ic.kmitl.ac.th

Pinging ic.inno.kmitl.ac.th [161.246.35.104] with 32 bytes of data:
Reply from 161.246.35.104: bytes=32 time=8ms TTL=51
Reply from 161.246.35.104: bytes=32 time=8ms TTL=51
Reply from 161.246.35.104: bytes=32 time=8ms TTL=51
Reply from 161.246.35.104: bytes=32 time=8ms TTL=51

Ping statistics for 161.246.35.104:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 8ms, Maximum = 8ms, Average = 8ms
  
```

1. IP of the domain you want to ping.
2. The amount of data that sent to the target in each sending.
3. Tell that you can ping the target or it reachable.
4. Amount of data that target receive.
5. The time that cost from receiving the packet.
6. The value of the hops to the target IP.
7. Tell how many packets sent.
8. Tell how many packets that target receive.
9. Tell how many packets that target didn't receive or loss.
10. The least time cost in sending the packet.

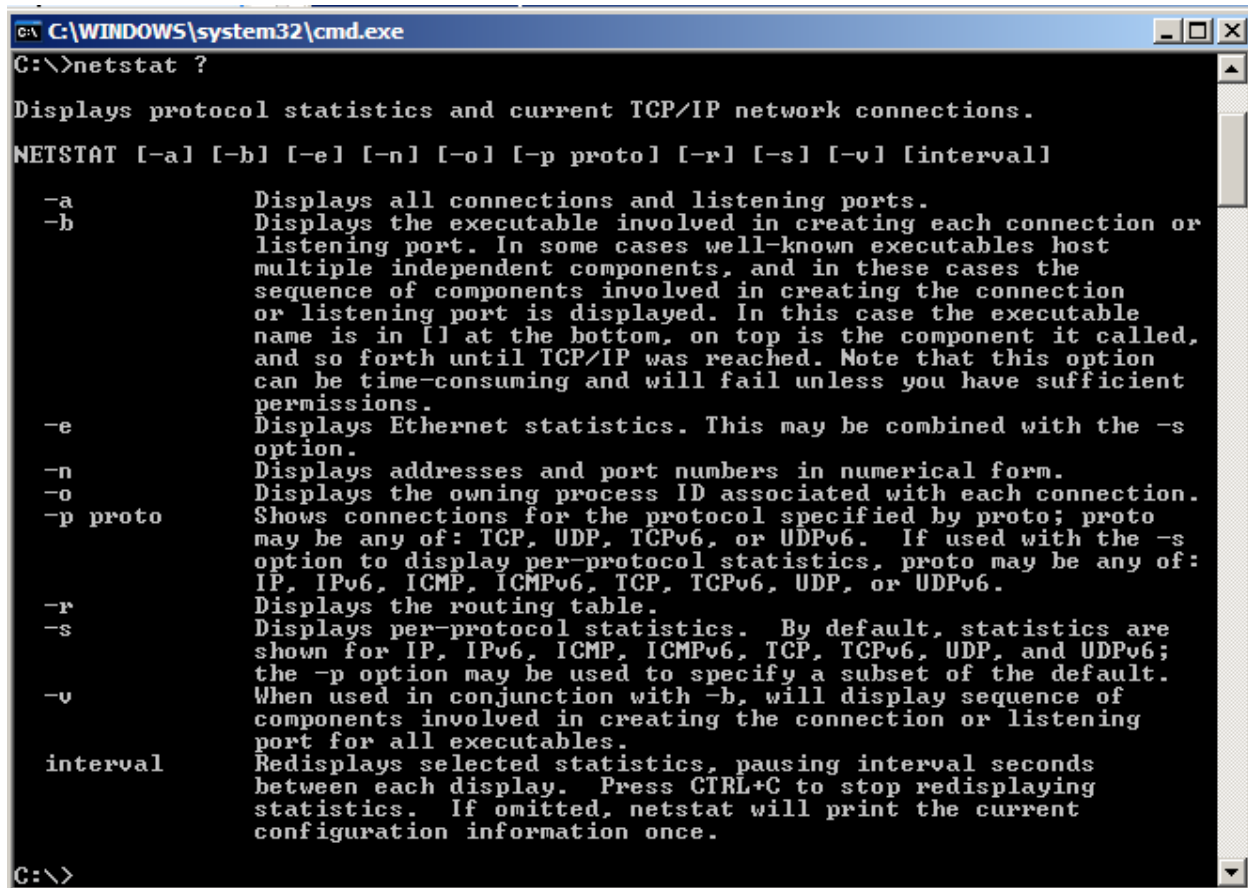
**Name** Audthanee Supeeramongkolkul **Student-ID** 66011525

11. The highest time cost in sending the packet.
12. The least time cost in sending the packet.

Name \_\_\_\_\_ Student-ID \_\_\_\_\_

## How to use netstat command

You can use the netstat command to monitor and troubleshoot many network problems, and in this guide, you'll get the knowledge to get started with the tool on Windows 10.



```
C:\WINDOWS\system32\cmd.exe
C:\>netstat ?

Displays protocol statistics and current TCP/IP network connections.

NETSTAT [-a] [-b] [-e] [-n] [-o] [-p proto] [-r] [-s] [-v] [interval]

-a          Displays all connections and listening ports.
-b          Displays the executable involved in creating each connection or
           listening port. In some cases well-known executables host
           multiple independent components, and in these cases the
           sequence of components involved in creating the connection
           or listening port is displayed. In this case the executable
           name is in [] at the bottom, on top is the component it called,
           and so forth until TCP/IP was reached. Note that this option
           can be time-consuming and will fail unless you have sufficient
           permissions.
-e          Displays Ethernet statistics. This may be combined with the -s
           option.
-n          Displays addresses and port numbers in numerical form.
-o          Displays the owning process ID associated with each connection.
-p proto    Shows connections for the protocol specified by proto; proto
           may be any of: TCP, UDP, TCPv6, or UDPv6. If used with the -s
           option to display per-protocol statistics, proto may be any of:
           IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, or UDPv6.
-r          Displays the routing table.
-s          Displays per-protocol statistics. By default, statistics are
           shown for IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, and UDPv6;
           the -p option may be used to specify a subset of the default.
-v          When used in conjunction with -b, will display sequence of
           components involved in creating the connection or listening
           port for all executables.
interval   Redisplays selected statistics, pausing interval seconds
           between each display. Press CTRL+C to stop redisplaying
           statistics. If omitted, netstat will print the current
           configuration information once.

C:\>
```

On [Windows 10](#), netstat (network statistics) has been around for a long time, and it's a command-line tool that you can use in Command Prompt to display statistics for all network connections. It allows you to understand open and connected ports to monitor and troubleshoot networking problems for system or applications.

When using this tool, you can list active networks (incoming and outgoing) connections and listening ports. You can view network adapter statistics as well as statistics for protocols (such as IPv4 and IPv6). You can even display the current routing table, and much more.



Name \_\_\_\_\_ Student-ID \_\_\_\_\_

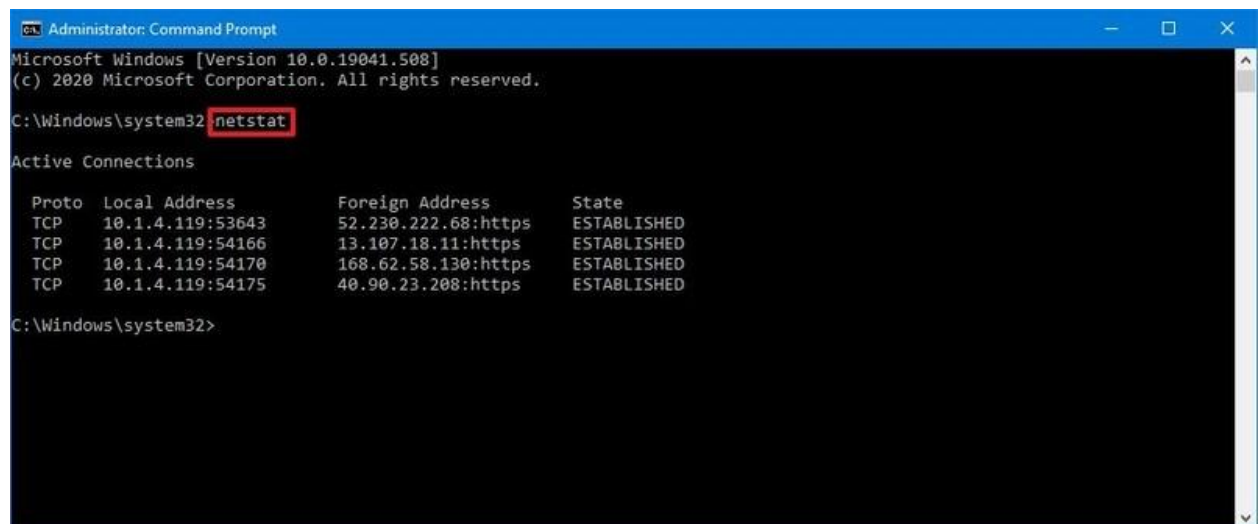
In this [Windows 10 guide](#), we'll walk you through the steps to use the netstat command to examine connections to discover open and connected network ports.

How to use netstat on Windows 10

To get started with netstat, use these steps:

1. Open **Start**.
2. Search for **Command Prompt**, right-click the top result, and select the **Run as administrator** option.
3. Type the following command to show all active TCP connections and press **Enter**:

netstat



The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The text inside the window is as follows:

```
Microsoft Windows [Version 10.0.19041.508]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Windows\system32 netstat

Active Connections

Proto Local Address           Foreign Address         State
TCP    10.1.4.119:53643         52.230.222.68:https     ESTABLISHED
TCP    10.1.4.119:54166         13.107.18.11:https      ESTABLISHED
TCP    10.1.4.119:54170         168.62.58.130:https     ESTABLISHED
TCP    10.1.4.119:54175         40.90.23.208:https      ESTABLISHED

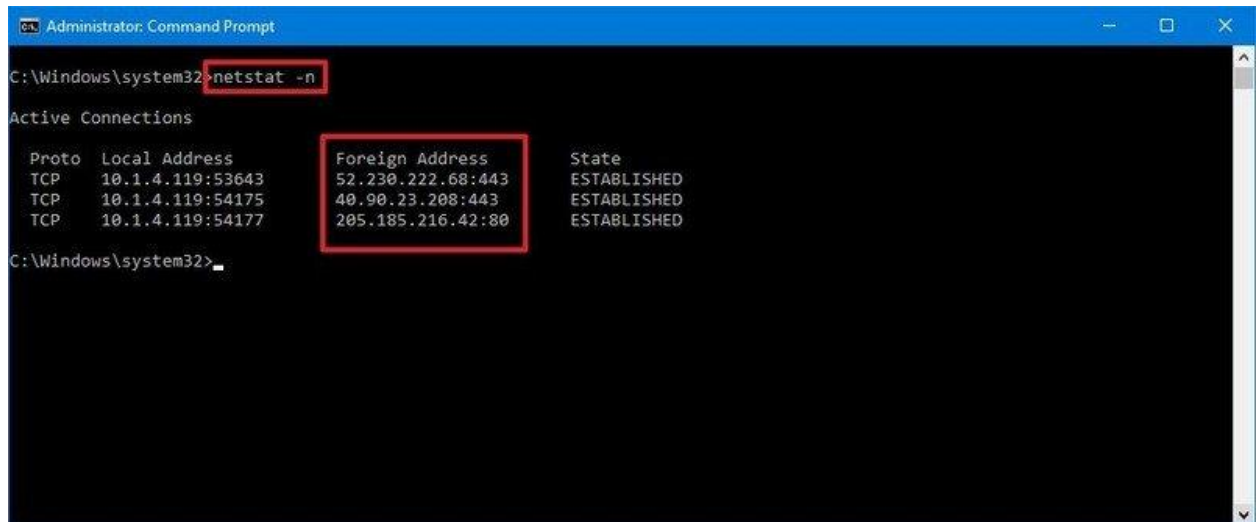
C:\Windows\system32>
```

4. (Optional) Type the following command to display active connections showing numeric IP address and port number instead of trying to determine the names and press **Enter**:

netstat -n



Name \_\_\_\_\_ Student-ID \_\_\_\_\_



```
Administrator: Command Prompt
C:\Windows\system32>netstat -n

Active Connections

Proto Local Address          Foreign Address         State
TCP    10.1.4.119:53643        52.230.222.68:443      ESTABLISHED
TCP    10.1.4.119:54175        40.90.23.208:443       ESTABLISHED
TCP    10.1.4.119:54177        205.185.216.42:80      ESTABLISHED

C:\Windows\system32>
```

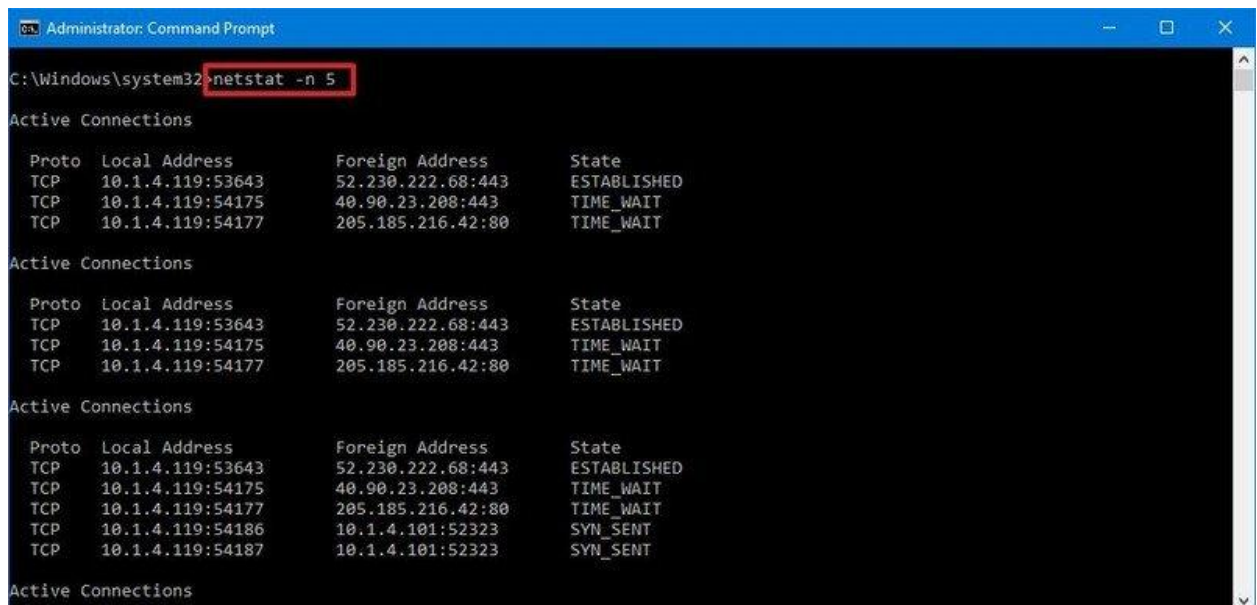
5. (Optional) Type the following command to refresh the information at a specific interval and press **Enter**:

`netstat -n INTERVAL`

In the command, make sure to replace **INTERVAL** for the number (in seconds) you want to redisplay the information.

This example refreshes the command in question every five seconds:

`netstat -n 5`



```
Administrator: Command Prompt
C:\Windows\system32>netstat -n 5

Active Connections

Proto Local Address          Foreign Address         State
TCP    10.1.4.119:53643        52.230.222.68:443      ESTABLISHED
TCP    10.1.4.119:54175        40.90.23.208:443       TIME_WAIT
TCP    10.1.4.119:54177        205.185.216.42:80      TIME_WAIT

Active Connections

Proto Local Address          Foreign Address         State
TCP    10.1.4.119:53643        52.230.222.68:443      ESTABLISHED
TCP    10.1.4.119:54175        40.90.23.208:443       TIME_WAIT
TCP    10.1.4.119:54177        205.185.216.42:80      TIME_WAIT

Active Connections

Proto Local Address          Foreign Address         State
TCP    10.1.4.119:53643        52.230.222.68:443      ESTABLISHED
TCP    10.1.4.119:54175        40.90.23.208:443       TIME_WAIT
TCP    10.1.4.119:54177        205.185.216.42:80      TIME_WAIT
TCP    10.1.4.119:54186        10.1.4.101:52323       SYN_SENT
TCP    10.1.4.119:54187        10.1.4.101:52323       SYN_SENT

Active Connections
```

**Quick note:** When using the interval parameter, you can terminate the command using the **Ctrl + C** keyboard shortcut in the console.

**Name** \_\_\_\_\_ **Student-ID** \_\_\_\_\_

Once you execute the command, it'll return a list of all active connections in four columns, including:

- **Proto:** Shows the connection protocol (TCP or UDP).
- **Local Address:** Shows the computer's IP address followed by a semicolon with a port number of the connection. The double-semicolon inside brackets indicates the local IPv6 address, and "0.0.0.0" refers to the local address too.
- **Foreign Address:** Lists the remote device's IP (or FQDN) address with the port number after semicolon port name (for example, https, http, microsoft-ds, wsd).
- **State:** Indicates where the connection is active (established), the local port has been closed (time\_wait), and the program hasn't closed the port (close\_wait). Other status include, closed, fin\_wait\_1, fin\_wait\_2, last\_ack, listen, syn\_received, syn\_send, and timed\_wait.

## How to search netstat details on Windows 10

In addition to displaying all the available statistic information, you can also output only the certain details you need using these steps:

1. Open **Start**.
2. Search for **Command Prompt**, right-click the top result, and select the **Run as administrator** option.
3. Type the following command to list all the connections that have the state set to LISTENING and press **Enter**:

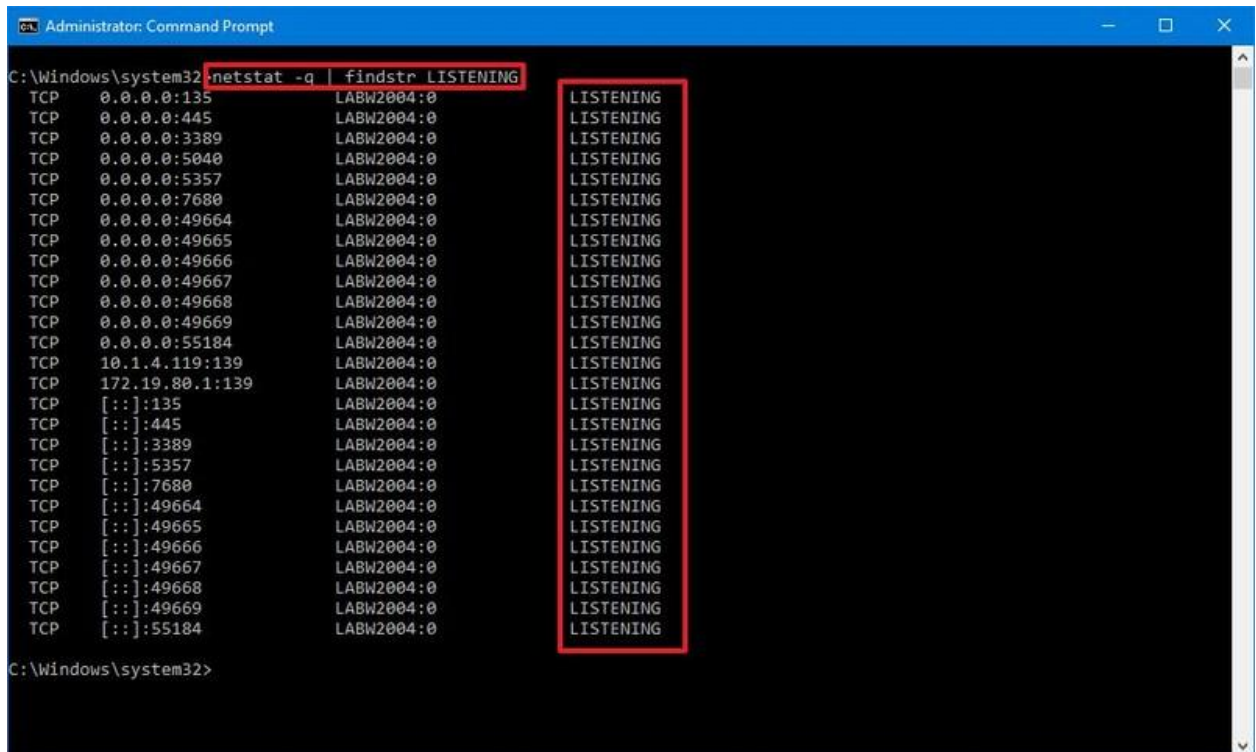
```
netstat -q | findstr STRING
```

In the command, make sure to replace STRING for the information you want to list. Also, the **findstr** option is case sensitive, which means that you must enter the string you want to find with the exact casing.

This example lists all the connections that have the state set to "LISTENING."

```
netstat -q | findstr LISTENING
```

Name \_\_\_\_\_ Student-ID \_\_\_\_\_



```
Administrator: Command Prompt
C:\Windows\system32>netstat -q | findstr LISTENING
TCP    0.0.0.0:135      LABW2004:0      LISTENING
TCP    0.0.0.0:445      LABW2004:0      LISTENING
TCP    0.0.0.0:3389     LABW2004:0      LISTENING
TCP    0.0.0.0:5040     LABW2004:0      LISTENING
TCP    0.0.0.0:5357     LABW2004:0      LISTENING
TCP    0.0.0.0:7680     LABW2004:0      LISTENING
TCP    0.0.0.0:49664    LABW2004:0      LISTENING
TCP    0.0.0.0:49665    LABW2004:0      LISTENING
TCP    0.0.0.0:49666    LABW2004:0      LISTENING
TCP    0.0.0.0:49667    LABW2004:0      LISTENING
TCP    0.0.0.0:49668    LABW2004:0      LISTENING
TCP    0.0.0.0:49669    LABW2004:0      LISTENING
TCP    0.0.0.0:55184    LABW2004:0      LISTENING
TCP    10.1.4.119:139   LABW2004:0      LISTENING
TCP    172.19.80.1:139  LABW2004:0      LISTENING
TCP    [::]:135         LABW2004:0      LISTENING
TCP    [::]:445         LABW2004:0      LISTENING
TCP    [::]:3389        LABW2004:0      LISTENING
TCP    [::]:5357        LABW2004:0      LISTENING
TCP    [::]:7680        LABW2004:0      LISTENING
TCP    [::]:49664       LABW2004:0      LISTENING
TCP    [::]:49665       LABW2004:0      LISTENING
TCP    [::]:49666       LABW2004:0      LISTENING
TCP    [::]:49667       LABW2004:0      LISTENING
TCP    [::]:49668       LABW2004:0      LISTENING
TCP    [::]:49669       LABW2004:0      LISTENING
TCP    [::]:55184       LABW2004:0      LISTENING
C:\Windows\system32>
```

This other example shows all the connections from a foreign server FQDN, in this case, Amazon:

```
netstat -f | findstr amazon
```

As you can see, you only need to type part of the string to return a result.

The **findstr** command isn't part of the **netstat** tool. It's a simple command to search for a text string in a file, but you can use it with many of the netstat commands to make more sense of the information you're viewing.

The netstat command is available on Windows 10, but you can also find it on Windows Server, Windows 8.x, Windows 7, and older versions. The tool is not exclusive to Windows either, as it's also available across platforms, including Linux and macOS. Even though the parameters and syntax may be different, they all are very similar.

Name \_\_\_\_\_ Student-ID \_\_\_\_\_

## Monitoring internet connected programs by using Netstat -b

(Need to run cmd as admin)

The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The command "netstat -b" has been executed, displaying a list of active network connections. Each entry includes the protocol (Proto), local address, foreign address, state, and the name of the application (exe) that established the connection. The connections are listed in a table-like format.

Proto	Local Address	Foreign Address	State	Application
TCP	127.0.0.1:5354	192.49675	ESTABLISHED	[mDNSResponder.exe]
TCP	127.0.0.1:5354	192.49676	ESTABLISHED	[mDNSResponder.exe]
TCP	127.0.0.1:49675	192.5354	ESTABLISHED	[AppleMobileDeviceService.exe]
TCP	127.0.0.1:49676	192.5354	ESTABLISHED	[AppleMobileDeviceService.exe]
TCP	192.168.1.138:2869	192.168.1.108:33346	TIME WAIT	
TCP	192.168.1.138:14168	20.198.2.181:https	ESTABLISHED	CDPUserSvc_7890d
TCP	192.168.1.138:14171	13.107.4.254:https	ESTABLISHED	[svchost.exe]
TCP	192.168.1.138:14174	204.79.197.222:https	ESTABLISHED	[SearchApp.exe]
TCP	192.168.1.138:14196	204.79.197.254:https	ESTABLISHED	[SearchApp.exe]
TCP	192.168.1.138:49709	20.25.241.18:https	ESTABLISHED	[SearchApp.exe]
TCP	192.168.1.138:49806	192.168.1.119:8009	ESTABLISHED	WpnService
TCP	192.168.1.138:49807	192.168.1.119:32252	ESTABLISHED	[svchost.exe]
TCP	[2405:9800:bc00:3e7d:9513:1cce:c0d8:eb6f]:1351	edge-star6-shv-02-xspl:https	ESTABLISHED	[chrome.exe]
TCP	[2405:9800:bc00:3e7d:9513:1cce:c0d8:eb6f]:2072	edge-star6-shv-02-xspl:https	ESTABLISHED	[chrome.exe]
TCP	[2405:9800:bc00:3e7d:9513:1cce:c0d8:eb6f]:13755	[2603:1046:1406:1::3]:https	ESTABLISHED	[chrome.exe]
TCP	[2405:9800:bc00:3e7d:9513:1cce:c0d8:eb6f]:13823	edge-msgr-latest6-shv-02-xspl:https	ESTABLISHED	[WINWORD.EXE]
TCP	[2405:9800:bc00:3e7d:9513:1cce:c0d8:eb6f]:13857	mc:https	ESTABLISHED	[chrome.exe]
TCP	[2405:9800:bc00:3e7d:9513:1cce:c0d8:eb6f]:13944	edge-msgr-latest6-shv-03-xspl:https	ESTABLISHED	[chrome.exe]
TCP	[2405:9800:bc00:3e7d:9513:1cce:c0d8:eb6f]:13946	edge-msgr-latest6-shv-03-xspl:https	ESTABLISHED	[chrome.exe]
TCP	[2405:9800:bc00:3e7d:9513:1cce:c0d8:eb6f]:14027	ku109s02-in-x0a:https	ESTABLISHED	[chrome.exe]
TCP	[2405:9800:bc00:3e7d:9513:1cce:c0d8:eb6f]:14033	edge-msgr-latest6-shv-03-xspl:https	ESTABLISHED	[GoogleDriveFS.exe]

Name \_\_\_\_\_ Student-ID \_\_\_\_\_

Let's look at what libraries each internet-connected program details.

```
C:\>netstat -bv

Active Connections

Proto Local Address          Foreign Address         State       PID
TCP   coco:1325              localhost:1326          ESTABLISHED 564
C:\WINDOWS\system32\WS2_32.dll
C:\Program Files\Mozilla Firefox\nspr4.dll
C:\Program Files\Mozilla Firefox\xul.dll
-- unknown component(s) --
[firefox.exe]

TCP   coco:1326              localhost:1325          ESTABLISHED 564
C:\WINDOWS\system32\mswsock.dll
C:\PROGRA~1\Google\GOOGLE~1\GOEC62~1.DLL
C:\WINDOWS\system32\WS2_32.dll
C:\Program Files\Mozilla Firefox\nspr4.dll
C:\Program Files\Mozilla Firefox\xul.dll
-- unknown component(s) --
[firefox.exe]

TCP   coco:1327              localhost:1328          ESTABLISHED 564
C:\WINDOWS\system32\WS2_32.dll
C:\Program Files\Mozilla Firefox\nspr4.dll
C:\Program Files\Mozilla Firefox\xul.dll
C:\Program Files\Mozilla Firefox\nspr4.dll
C:\Program Files\Mozilla Firefox\xul.dll
-- unknown component(s) --
[firefox.exe]

TCP   coco:1328              localhost:1327          ESTABLISHED 564
C:\WINDOWS\system32\mswsock.dll
C:\PROGRA~1\Google\GOOGLE~1\GOEC62~1.DLL
C:\WINDOWS\system32\WS2_32.dll
C:\Program Files\Mozilla Firefox\nspr4.dll
C:\Program Files\Mozilla Firefox\xul.dll
-- unknown component(s) --
[firefox.exe]

TCP   coco:4196              localhost:30606         ESTABLISHED 740
C:\WINDOWS\system32\mswsock.dll
C:\PROGRA~1\Google\GOOGLE~1\GOEC62~1.DLL
C:\WINDOWS\system32\WS2_32.dll
C:\Program Files\MSN Messenger\msnmsgr.exe
[msnmsgr.exe]
```

TIP : which we will now read, and it will be possible to guess what each line that the program displays means. But if you use option as shown above (per Internet), type the netstat command -a to display as a name, perhaps in the ESTABLISHED state. imply There's another machine that's already entered us. It could be the website we're downloading. The netstat -abv command is displayed as various web names. Which, if using the netstat -an command , is displayed as an IP number , hard to guess. And to really look at it, we have to observe that port. That's what port we're used to do.

**Name** \_\_\_\_\_ **Student-ID** \_\_\_\_\_

## NMAP

### What is Nmap?

At its core, Nmap is a network scanning tool that uses IP packets to identify all the devices connected to a network and to provide information on the services and operating systems they are running.

The program is most commonly used via a command-line interface (though GUI front-ends are also available) and is available for many different operating systems such as Linux, Free BSD, and Gentoo. Its popularity has also been bolstered by an active and enthusiastic user support community.

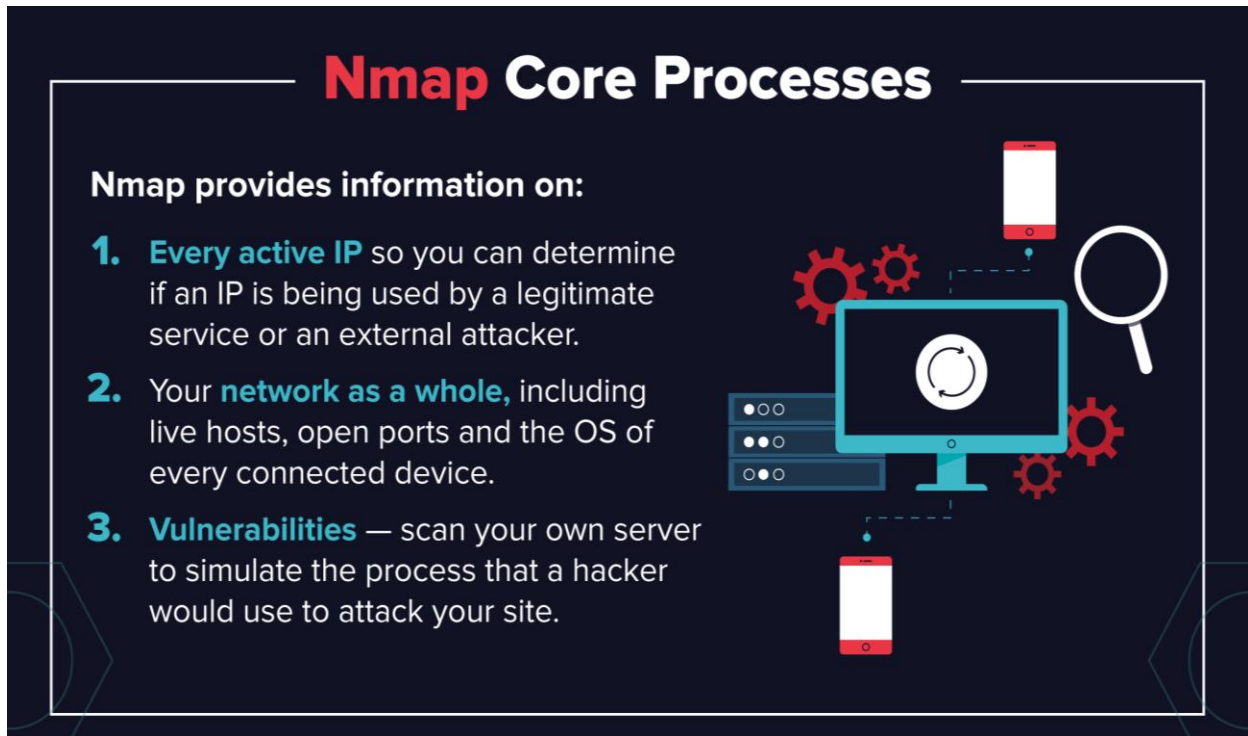
Nmap was developed for enterprise-scale networks and can scan through thousands of connected devices. However, in recent years Nmap is being increasingly used by smaller companies. The rise of the IoT, in particular, now means that the networks used by these companies have become more complex [and therefore harder to secure](#).

This means that Nmap is now [used in many website monitoring tools](#) to audit the traffic between web servers and IoT devices. The recent emergence of [IoT botnets, like Mirai](#), has also stimulated interest in Nmap, not least because of its ability to interrogate [devices connected via the UPnP protocol](#) and to highlight any devices that may be malicious.



Name \_\_\_\_\_ Student-ID \_\_\_\_\_

What Does Nmap Do?



At a practical level, Nmap is used to provide detailed, real-time information on your networks, and on the devices connected to them.

The primary uses of Nmap can be broken into three core processes. First, the program gives you detailed information on every IP active on your networks, and each IP can then be scanned. This allows administrators to check whether an IP is being used by a legitimate service, or by an external attacker.

Secondly, Nmap provides information on your network as a whole. It can be used to provide a list of live hosts and open ports, as well as identifying the OS of every connected device. This makes it a valuable tool in ongoing system monitoring, as well as a critical part of pentesting. Nmap can be used alongside [the Metasploit framework](#), for instance, to probe and then repair network vulnerabilities.

Thirdly, Nmap has also become a valuable tool for users looking to protect personal and business websites. Using Nmap to scan your own web server, particularly if you are hosting your website from home, is essentially simulating the process that a hacker would use to attack your site. “Attacking” your own site in this way is a powerful way of identifying security vulnerabilities.



**Name** \_\_\_\_\_ **Student-ID** \_\_\_\_\_

## How To Use Nmap

Nmap is straightforward to use, and most of the tools it provides are familiar to system admins from other programs. The advantage of Nmap is that it brings a wide range of these tools into one program, rather than forcing you to skip between separate and discrete network monitoring tools.

In order to use Nmap, you need to be familiar with command-line interfaces. Most advanced users are able to write scripts to automate common tasks, but this is not necessary for basic network monitoring.

## How To Install Nmap

The process for installing Nmap is easy but varies according to your operating system. The Windows, Mac, and Linux versions of the [program can be downloaded here](#).

- For Windows, Nmap comes with a custom installer (nmap<version>setup.exe). Download and run this installer, and it automatically configures Nmap on your system.
- On Mac, Nmap also comes with a dedicated installer. Run the Nmap-<version>mpkg file to start this installer. On some recent versions of macOS, you might see a warning that Nmap is an “unidentified developer”, but you can ignore this warning.
- Linux users can either compile Nmap from source or use their chosen package manager. To use apt, for instance, you can run `Nmap --version` to check if Nmap is installed, and `sudo apt-get install Nmap` to install it.

## Nmap Tutorial and Examples

Once you’ve installed Nmap, the best way of learning how to use it is to perform some basic network scans.

### How To Run a Ping Scan

One of the most basic functions of Nmap is to identify active hosts on your network. Nmap does this by using a ping scan. This identifies all of the IP addresses that are currently online without sending any packets to these hosts.

To run a ping scan, run the following command:

```
# nmap -sp 192.100.1.1/24
```

This command then returns a list of hosts on your network and the total number of assigned IP addresses. If you spot any hosts or IP addresses on this list that you cannot account for, you can then run further commands (see below) to investigate them further.

**Name** \_\_\_\_\_ **Student-ID** \_\_\_\_\_

### How To Run A Host Scan

A more powerful way to scan your networks is to use Nmap to perform a host scan. Unlike a ping scan, a host scan actively sends ARP request packets to all the hosts connected to your network. Each host then responds to this packet with another ARP packet containing its status and MAC address.

To run a host scan, use the following command:

```
# nmap -sp <target IP range>
```

This returns information on every host, their latency, their MAC address, and also any description associated with this address. This can be a powerful way of spotting suspicious hosts connected to your network.

If you see anything unusual in this list, you can then run a DNS query on a specific host, by using:

```
# nmap -sL <IP address>
```

This returns a list of names associated with the scanned IP. This description provides information on what the IP is actually for.

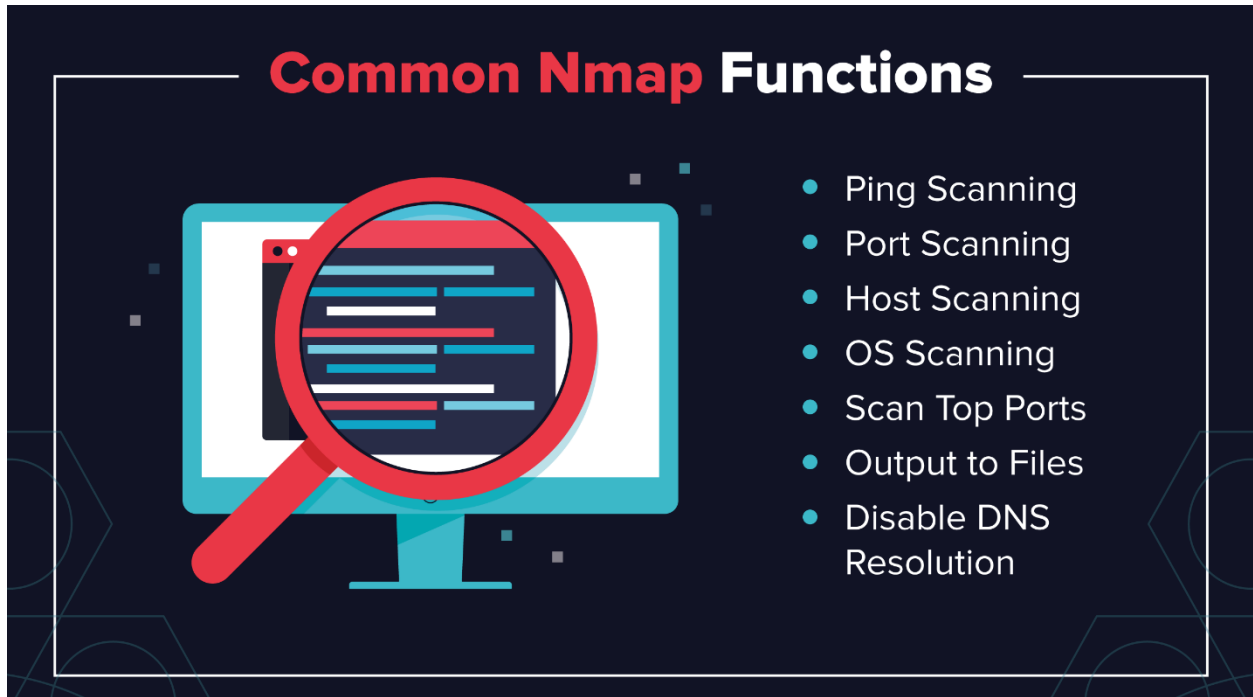
### How To Use Nmap in Kali Linux

Using Nmap in Kali Linux can be done in an identical way to running the program on any other flavor of Linux.

That said, there are advantages to using Kali when running Nmap scans. Most modern distros of Kali now come with a fully-features Nmap suite, which includes an advanced GUI and results viewer (Zenmap), a flexible data transfer, redirection, and debugging tool (Ncat), a utility for comparing scan results (Ndiff), and a packet generation and response analysis tool (Nping).

Name \_\_\_\_\_ Student-ID \_\_\_\_\_

## Nmap Commands



Most of the common functions of Nmap can be executed using a single command, and the program also uses a number of 'shortcut' commands that can be used to automate common tasks.

Here is a quick run-down:

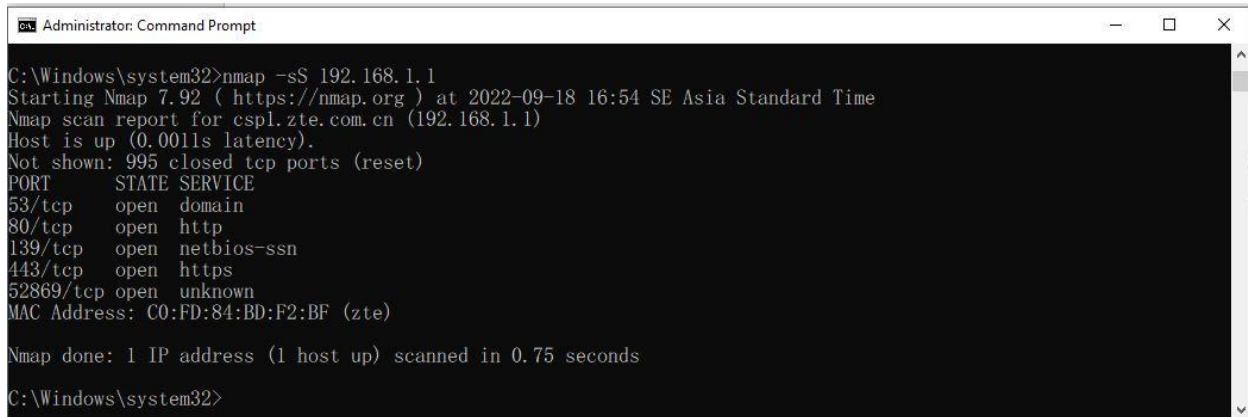
### 1. Ping Scanning

As mentioned above, a ping scan returns information on every active IP on your network. You can execute a ping scan using this command:

```
# nmap -sp 192.100.1.1/24
```

Name \_\_\_\_\_ Student-ID \_\_\_\_\_

## 2. Port Scanning



```
Administrator: Command Prompt
C:\Windows\system32>nmap -sS 192.168.1.1
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-18 16:54 SE Asia Standard Time
Nmap scan report for cspl.zte.com.cn (192.168.1.1)
Host is up (0.0011s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
139/tcp   open  netbios-ssn
443/tcp   open  https
52869/tcp open  unknown
MAC Address: C0:FD:84:BD:F2:BF (zte)

Nmap done: 1 IP address (1 host up) scanned in 0.75 seconds
C:\Windows\system32>
```

There are several ways to execute port scanning using Nmap. The most used are these:

# sS TCP SYN scan

# sT TCP connect scan

# sU UDP scans

# sY SCTP INIT scan

# sN TCP NULL

The major differences between these types of scans are whether they cover TCP or UDP ports and whether they execute a TCP connection. Here are the basic differences:

- The most basic of these scans is the sS TCP SYN scan, and this gives most users all the information they need. It scans thousands of ports per second, and because it doesn't complete a TCP connection it does not arouse suspicion.
- The main alternative to this type of scan is the TCP Connect scan, which actively queries each host, and requests a response. This type of scan takes longer than a SYN scan, but can return more reliable information.
- The UDP scan works in a similar way to the TCP connect scan but uses UDP packets to scan DNS, SNMP, and DHCP ports. These are the ports most frequently targeted by hackers, and so this type of scan is a useful tool for checking for vulnerabilities.
- The SCTP INIT scan covers a different set of services: SS7 and SIGTRAN. This type of scan can also be used to avoid suspicion when scanning an external network because it doesn't complete the full SCTP process.
- The TOP NULL scan is also a very crafty scanning technique. It uses a loophole in the TCP system that can reveal the status of ports without directly querying them, which means that you can see their status even where they are protected by a firewall.

Name \_\_\_\_\_ Student-ID \_\_\_\_\_

### 3. Host Scanning

Host scanning returns more detailed information on a particular host or a range of IP addresses. As mentioned above, you can perform a host scan using the following command:

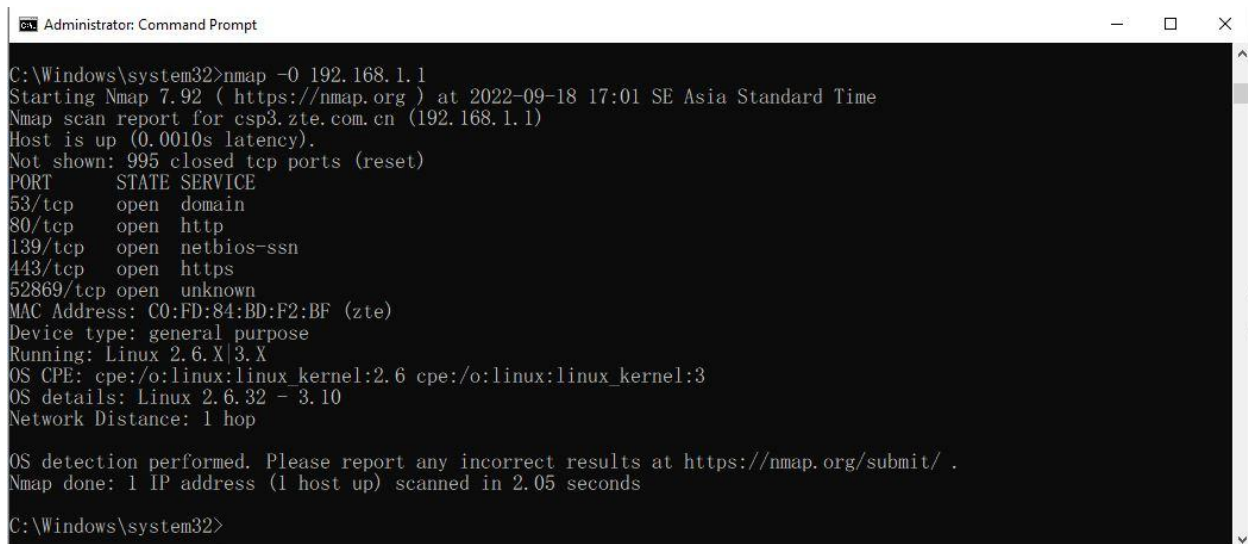
```
# nmap -sp <target IP range>
```

### 4. OS Scanning

OS scanning is one of the most powerful features of Nmap. When using this type of scan, Nmap sends TCP and UDP packets to a particular port, and then analyze its response. It compares this response to a database of 2600 operating systems, and return information on the OS (and version) of a host.

To run an OS scan, use the following command:

```
# nmap -O <target IP>
```



```
Administrator: Command Prompt
C:\Windows\system32>nmap -O 192.168.1.1
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-18 17:01 SE Asia Standard Time
Nmap scan report for csp3.zte.com.cn (192.168.1.1)
Host is up (0.0010s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
139/tcp   open  netbios-ssn
443/tcp   open  https
52869/tcp open  unknown
MAC Address: C0:FD:84:BD:F2:BF (zte)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
OS details: Linux 2.6.32 - 3.10
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.05 seconds
C:\Windows\system32>
```

Name \_\_\_\_\_ Student-ID \_\_\_\_\_

## 5. Scan The Most Popular Ports

```
Administrator: Command Prompt
C:\Windows\system32>nmap --top-ports 20 192.168.1.1
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-18 17:03 SE Asia Standard Time
Nmap scan report for cspl.zte.com.cn (192.168.1.1)
Host is up (0.0011s latency).

PORT      STATE SERVICE
21/tcp    closed ftp
22/tcp    closed ssh
23/tcp    closed telnet
25/tcp    closed smtp
53/tcp    open  domain
80/tcp    open  http
110/tcp   closed pop3
111/tcp   closed rpcbind
135/tcp   closed msrpc
139/tcp   open  netbios-ssn
143/tcp   closed imap
443/tcp   open  https
445/tcp   closed microsoft-ds
993/tcp   closed imaps
995/tcp   closed pop3s
1723/tcp  closed pptp
3306/tcp  closed mysql
3389/tcp  closed ms-wbt-server
5900/tcp  closed vnc
8080/tcp  closed http-proxy
MAC Address: C0:FD:84:BD:F2:BF (zte)

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
C:\Windows\system32>
```

If you are running Nmap on a home server, this command is very useful. It automatically scans a number of the most ‘popular’ ports for a host. You can run this command using:

```
nmap --top-ports 20 192.168.1.1
```

Replace the “20” with the number of ports to scan, and Nmap quickly scans that many ports. It returns a concise output that details the status of the most common ports, and this lets you quickly see whether you have any unnecessarily open ports.

## 6. Output to a File

If you want to output the results of your Nmap scans to a file, you can add an extension to your commands to do that. Simply add:

```
-oN output.txt
```

To your command to output the results to a text file, or:

```
-oX output.xml
```

**Name** \_\_\_\_\_ **Student-ID** \_\_\_\_\_

To output to an XML.

#### 7. Disable DNS Name Resolution

Finally, you can speed up your Nmap scans by using the `-n` parameter to disable reverse DNS resolution. This can be extremely useful if you want to scan a large network. For example, to turn off DNS resolution for the basic ping scan mentioned above, add `-n`:

```
# nmap -sp -n 192.1.0.1/24
```



Name \_\_\_\_\_ Student-ID \_\_\_\_\_

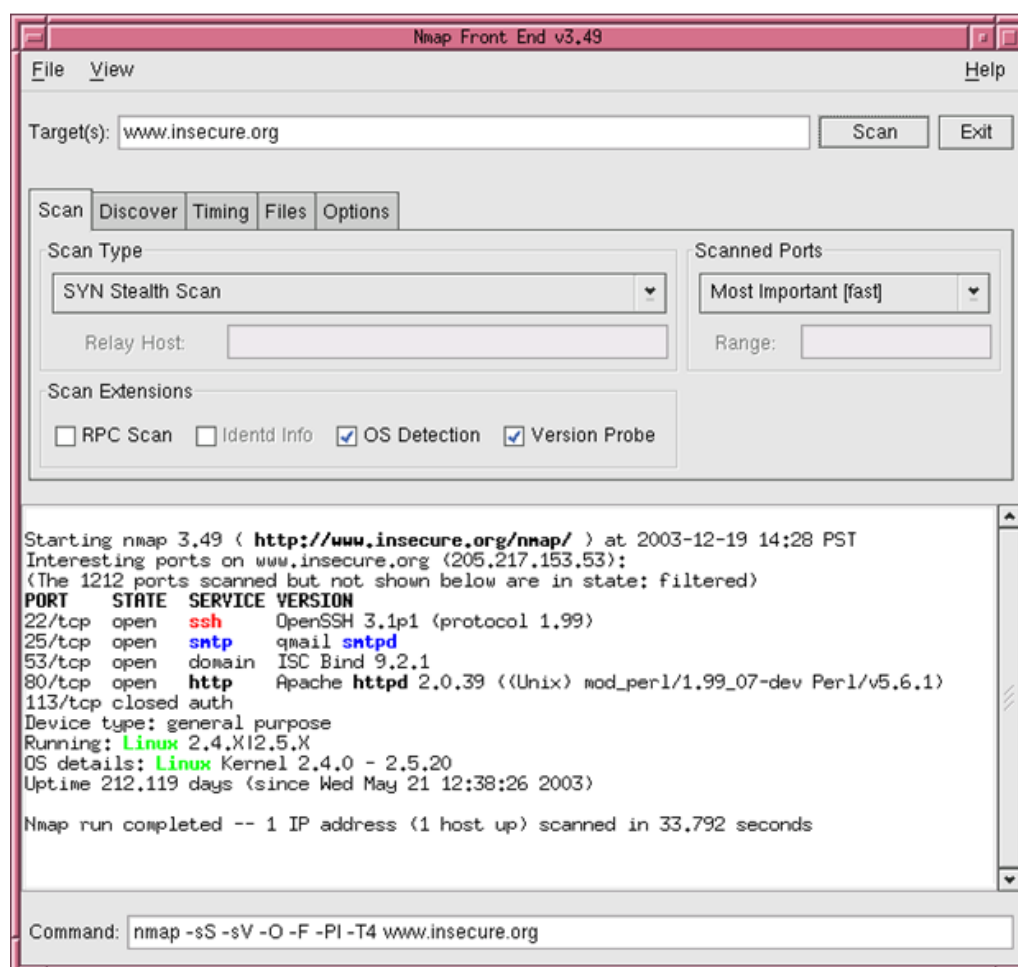
## Nmap FAQ

The commands above cover most of the basic functionality of Nmap. You might still have some questions though, so let's run through the most common ones.

### Q: What Are Some Nmap Alternatives?

There are some [alternatives to Nmap](#), but most of them are focused on providing specific, niche functionality that the average system administrator does need frequently. MASSCAN, for instance, is much faster than Nmap but provides less detail. Umit, by contrast, allows you to run several scans at once.

In reality, however, Nmap provides all the functionality and speed that the average user requires, especially when used alongside other similarly popular tools like [NetCat](#) (which can be used to manage and control network traffic) and [ZenMap](#) (which provides a GUI for Nmap)



Let install some namp frontends and alternatives (Lab 30 Mins) and try some nmap commands.

**Name** \_\_\_\_\_ **Student-ID** \_\_\_\_\_

Q: Is Nmap Legal?

Yes. If used properly, Nmap helps protect your network from hackers, because it allows you to quickly spot any security vulnerabilities in your systems.

Whether port scanning on external servers is legal is another issue. The legislation in this area is complex and varies by territory. Using Nmap to scan external ports can lead to you being banned by your ISP, so make sure you research the [legal implications of using the program](#) before you start using it more widely.

Name \_\_\_\_\_ Student-ID \_\_\_\_\_

### How Does Nmap Work? (Week 4)

Nmap builds on previous network auditing tools to provide quick, detailed scans of network traffic. It works by using IP packets to identify the hosts and IPs active on a network and then analyze these packets to provide information on each host and IP, as well as the operating systems they are running.

Nmap is an incredibly useful tool, but it's even more useful if you understand the results of an Nmap scan. After a scan is complete, Nmap will categorize each scanned port into one of six states; open, closed, filtered, open|filtered, closed|filtered, and unfiltered.

The use of the vertical bar (|) or "pipe" is used to denote the use of "or." The vertical bar is often used in programming to designate a logical "or" operation, and Nmap has borrowed this nomenclature for use in its output.

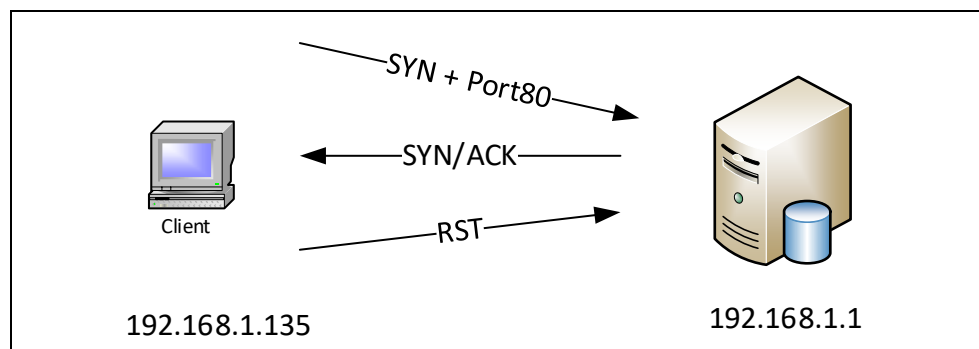
Some scan methods can only identify if a port is open, closed, or filtered, while other scan types might only be able to identify filtered or unfiltered ports. Where applicable, we've specified the scan types that apply to the port dispositions.

### Nmap Port Dispositions

#### open

When Nmap interrogates a port and receives a positive response, the port is assigned the state of "open." This is such a valuable state that Nmap even includes a special command line option, `--open`, to filter out all of the other port states.

For example, Nmap's TCP SYN scan (`-sS`) will receive a SYN/ACK from a remote device if a port is open:



Name \_\_\_\_\_ Student-ID \_\_\_\_\_

This is the Nmap output from a TCP SYN scan to port 25 of a device: # nmap 192.168.1.1 -p 25

```
Administrator: Command Prompt

C:\Windows\system32>nmap 192.168.1.1 -p 80
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-18 17:16 SE Asia Standard Time
Nmap scan report for cspl.zte.com.cn (192.168.1.1)
Host is up (0.00013s latency).

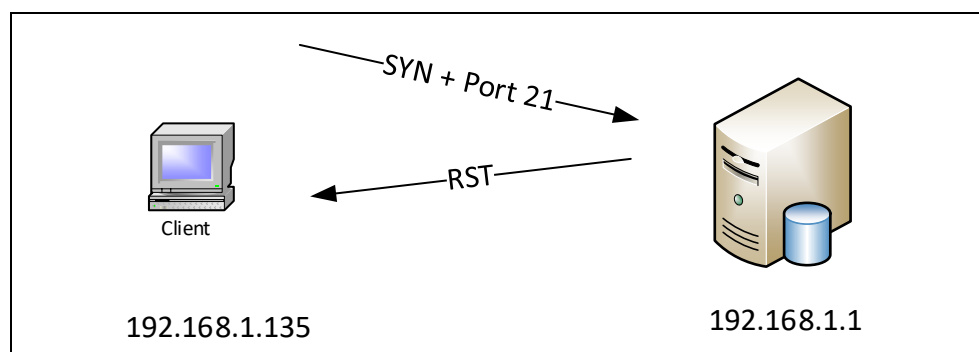
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: C0:FD:84:BD:F2:BF (zte)

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
C:\Windows\system32>
```

### closed

If Nmap determines that a port is not available, it assigns it the "closed" state. This signifies that Nmap has interrogated the port and has received a result that unequivocally shows that the port is closed.

A TCP SYN scan (-sS) receiving a RST in response to a port query is an example of a closed port:



Name \_\_\_\_\_ Student-ID \_\_\_\_\_

This is the output of an Nmap to two ports on a device, where one is open and the other is closed:

```
Administrator: Command Prompt
C:\Windows\system32>nmap 192.168.1.1 -p 21,25
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-18 17:27 SE Asia Standard Time
Nmap scan report for csp3.zte.com.cn (192.168.1.1)
Host is up (0.00076s latency).

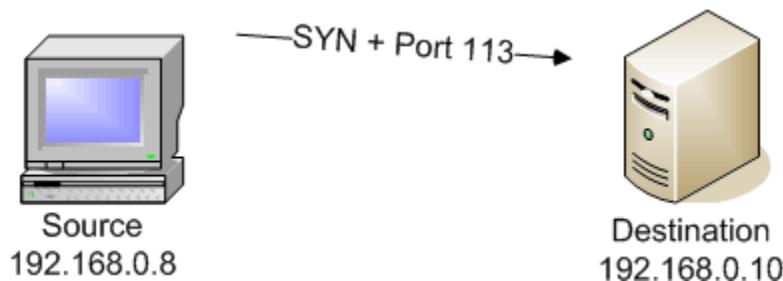
PORT      STATE SERVICE
21/tcp    closed ftp
25/tcp    closed smtp
MAC Address: C0:FD:84:BD:F2:BF (zte)

Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
C:\Windows\system32>
```

## filtered

Filtered ports are the result of a packet filter or firewall. When no response at all is received from the remote device, the port is "filtered." Since a response isn't received from the port, Nmap often retries communication to the port to ensure that the packet wasn't simply dropped due to error or congestion. Due to this retransmission process, filtered ports often cause delays during extensive Nmap scans.

This Nmap SYN scan sends a probe to a remote device, but a response is never received. Since this is a SYN scan, this response is categorized as "filtered."



Notice that this type of response is categorized differently if this is a different scan type, such as a UDP scan or a FIN scan (see open|filtered, below).

The Nmap output will clearly show filtered ports if the remote device does not respond to the scan. This is the output from an Nmap scan where one port is open and the other is filtered:

Name \_\_\_\_\_ Student-ID \_\_\_\_\_

```
# nmap scanme.insecure.org -p80,8088
```

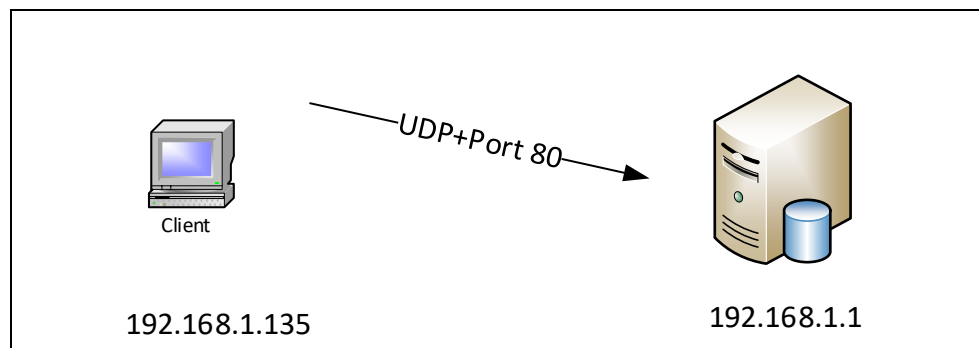
```
C:\Windows\system32>nmap scanme.insecure.org -p80,8088
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-18 17:32 SE Asia Standard Time
Nmap scan report for scanme.insecure.org (45.33.49.119)
Host is up (0.0060s latency).
rDNS record for 45.33.49.119: ack.nmap.org

PORT      STATE      SERVICE
80/tcp    open       http
8088/tcp   filtered   radan-http

Nmap done: 1 IP address (1 host up) scanned in 2.07 seconds
C:\Windows\system32>
```

### open|filtered

In some cases, the lack of a response may not necessarily mean that a port is filtered. In some cases, lack of a response might mean that the port might also be open. In these situations, Nmap signifies that the port is either filtered or open. The FIN scan (-sF), Xmas tree scan (-sX), Null scan (-sN) and UDP scan (-sU) can't definitively determine an open port, so they always specify that the port is open|filtered.



This is the resulting Nmap output from a UDP scan. Since UDP ports don't necessarily return any packets, Nmap categorizes them as open|filtered:

```
# nmap -sU -v 192.168.1.1
```

Name \_\_\_\_\_ Student-ID \_\_\_\_\_

```

Administrator: Command Prompt

C:\Windows\system32>nmap -sU -v 192.168.1.139
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-18 17:48 SE Asia Standard Time
Initiating ARP Ping Scan at 17:48
Scanning 192.168.1.139 [1 port]
Completed ARP Ping Scan at 17:48, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 17:48
Completed Parallel DNS resolution of 1 host. at 17:48, 0.00s elapsed
Initiating UDP Scan at 17:48
Scanning 192.168.1.139 (192.168.1.139) [1000 ports]
Discovered open port 137/udp on 192.168.1.139
Increasing send delay for 192.168.1.139 from 0 to 50 due to max_successful_ryno increase to 4
Increasing send delay for 192.168.1.139 from 50 to 100 due to max_successful_ryno increase to 5
Increasing send delay for 192.168.1.139 from 100 to 200 due to max_successful_ryno increase to 6
Increasing send delay for 192.168.1.139 from 200 to 400 due to max_successful_ryno increase to 7
Increasing send delay for 192.168.1.139 from 400 to 800 due to max_successful_ryno increase to 8
UDP Scan Timing: About 4.19% done; ETC: 18:00 (0:11:49 remaining)
Increasing send delay for 192.168.1.139 from 800 to 1000 due to 11 out of 24 dropped probes since last increase.
UDP Scan Timing: About 7.11% done; ETC: 18:02 (0:13:17 remaining)
^C
C:\Windows\system32>nmap -sU -v 192.168.1.1
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-18 17:49 SE Asia Standard Time
Initiating ARP Ping Scan at 17:49
Scanning 192.168.1.1 [1 port]
Completed ARP Ping Scan at 17:49, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 17:49
Completed Parallel DNS resolution of 1 host. at 17:49, 0.00s elapsed
Initiating UDP Scan at 17:49
Scanning csp3.zte.com.cn (192.168.1.1) [1000 ports]
Increasing send delay for 192.168.1.1 from 0 to 50 due to 11 out of 17 dropped probes since last increase.
Increasing send delay for 192.168.1.1 from 50 to 100 due to 11 out of 24 dropped probes since last increase.
UDP Scan Timing: About 26.02% done; ETC: 17:51 (0:01:28 remaining)
UDP Scan Timing: About 52.82% done; ETC: 17:51 (0:00:54 remaining)
Discovered open port 137/udp on 192.168.1.1
Discovered open port 53/udp on 192.168.1.1
Completed UDP Scan at 17:51, 112.38s elapsed (1000 total ports)
Nmap scan report for csp3.zte.com.cn (192.168.1.1)
Host is up (0.00066s latency).
Not shown: 994 closed udp ports (port-unreach)
PORT      STATE      SERVICE
53/udp    open       domain
67/udp    open|filtered dhcp
137/udp   open       netbios-ns
138/udp   open|filtered netbios-dgm
1900/udp  open|filtered upnp
5355/udp  open|filtered llmnr
MAC Address: C0:FD:84:BD:F2:BF (zte)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 112.56 seconds
Raw packets sent: 1073 (50.567KB) | Rcvd: 1069 (88.777KB)

C:\Windows\system32>

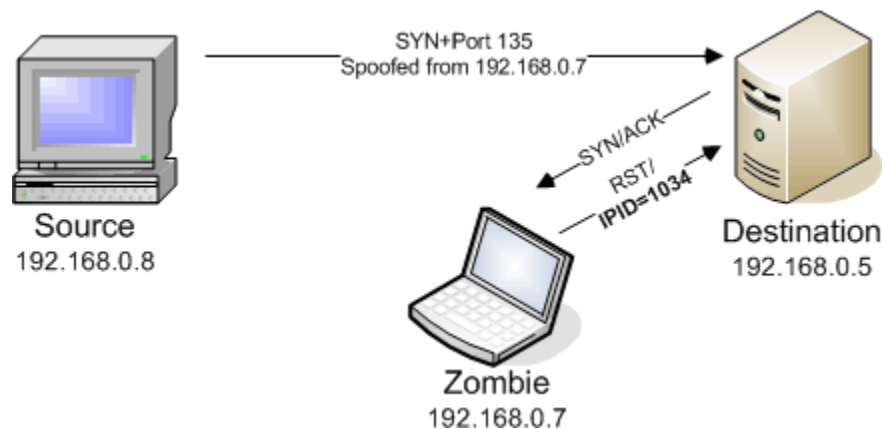
```



Name \_\_\_\_\_ Student-ID \_\_\_\_\_

### closed|filtered

There's only one scan that identifies ports as either closed or filtered. Nmap's `idlescan` (`-sI`) operates by spoofing a zombie's IP address and querying the IPID of the zombie to determine if a response was received. If the IPID increments, then the port is open.

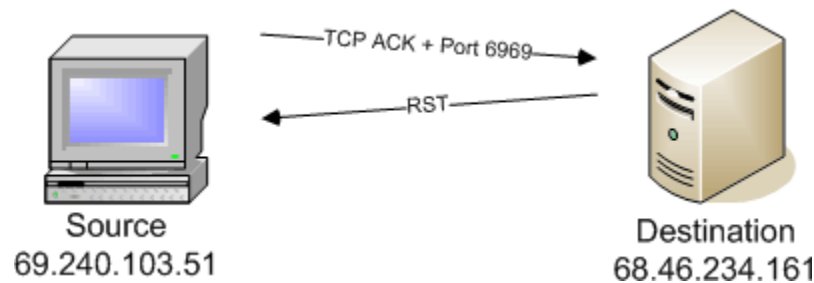


If the IPID does not increment, then Nmap cannot determine if the port was filtered or if it was closed. To be as specific as possible, Nmap categorizes this port as either closed or filtered.

Name \_\_\_\_\_ Student-ID \_\_\_\_\_

### unfiltered

The TCP ACK scan (-sA) is often used to determine the availability of ports on a firewall or packet filter. The response to an out-of-sequence ACK will return a RST, which also signifies that the port is unfiltered.



This TCP ACK scan focuses on ports 80 and 8088, and finds that one is filtered and the other is unfiltered: # nmap scanme.insecure.org -sA -p80,8088

```
Administrator: Command Prompt
C:\Windows\system32>nmap scanme.insecure.org -sA -p80,8088
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-18 17:56 SE Asia Standard Time
Nmap scan report for scanme.insecure.org (45.33.49.119)
Host is up (0.0068s latency).
rDNS record for 45.33.49.119: ack.nmap.org

PORT      STATE      SERVICE
80/tcp    unfiltered http
8088/tcp  unfiltered radan-http

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
C:\Windows\system32>
```

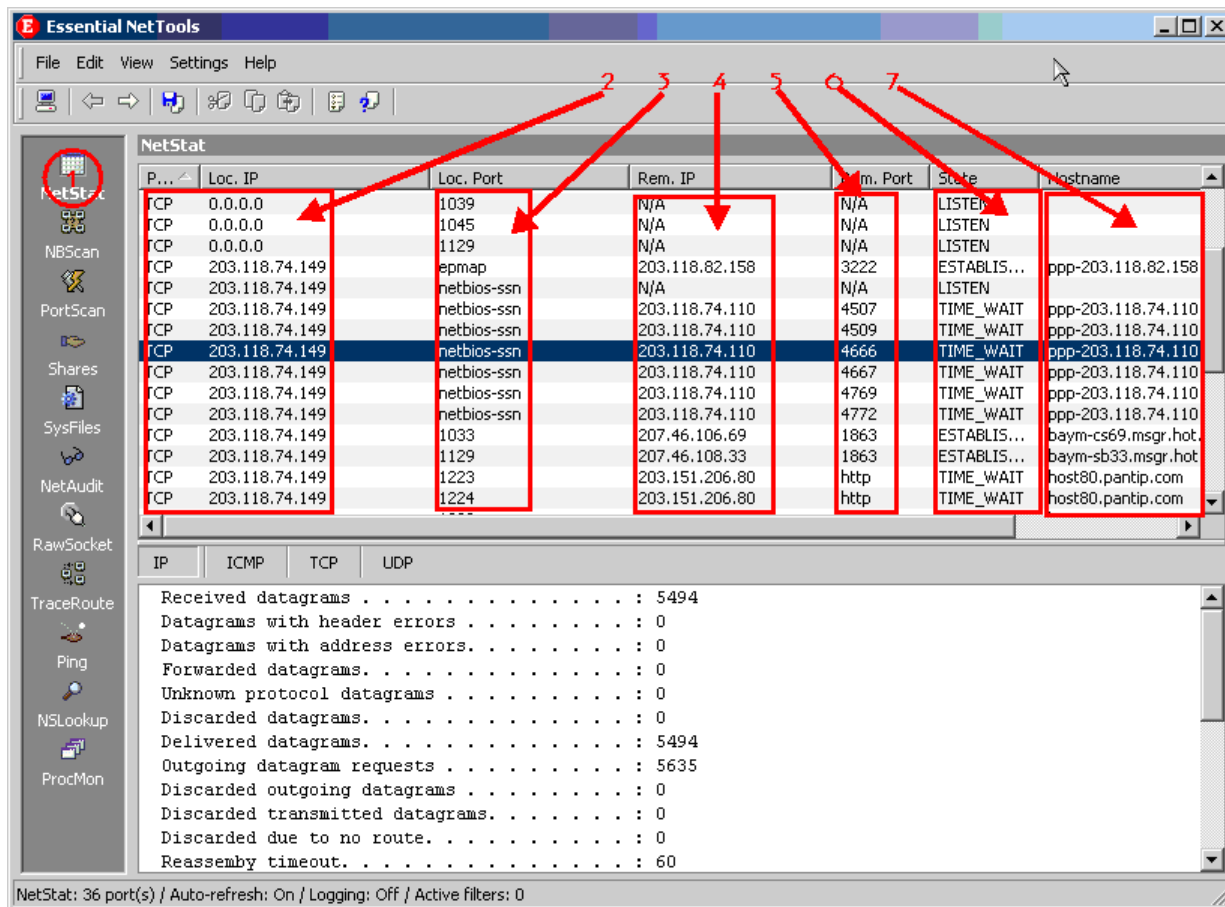
### Conclusion

Nmap is very descriptive in its port dispositions, so be sure to pay very close attention to the information in the output. When multiple scan methods are used on single scan, the results can be a mix of many different port designations.

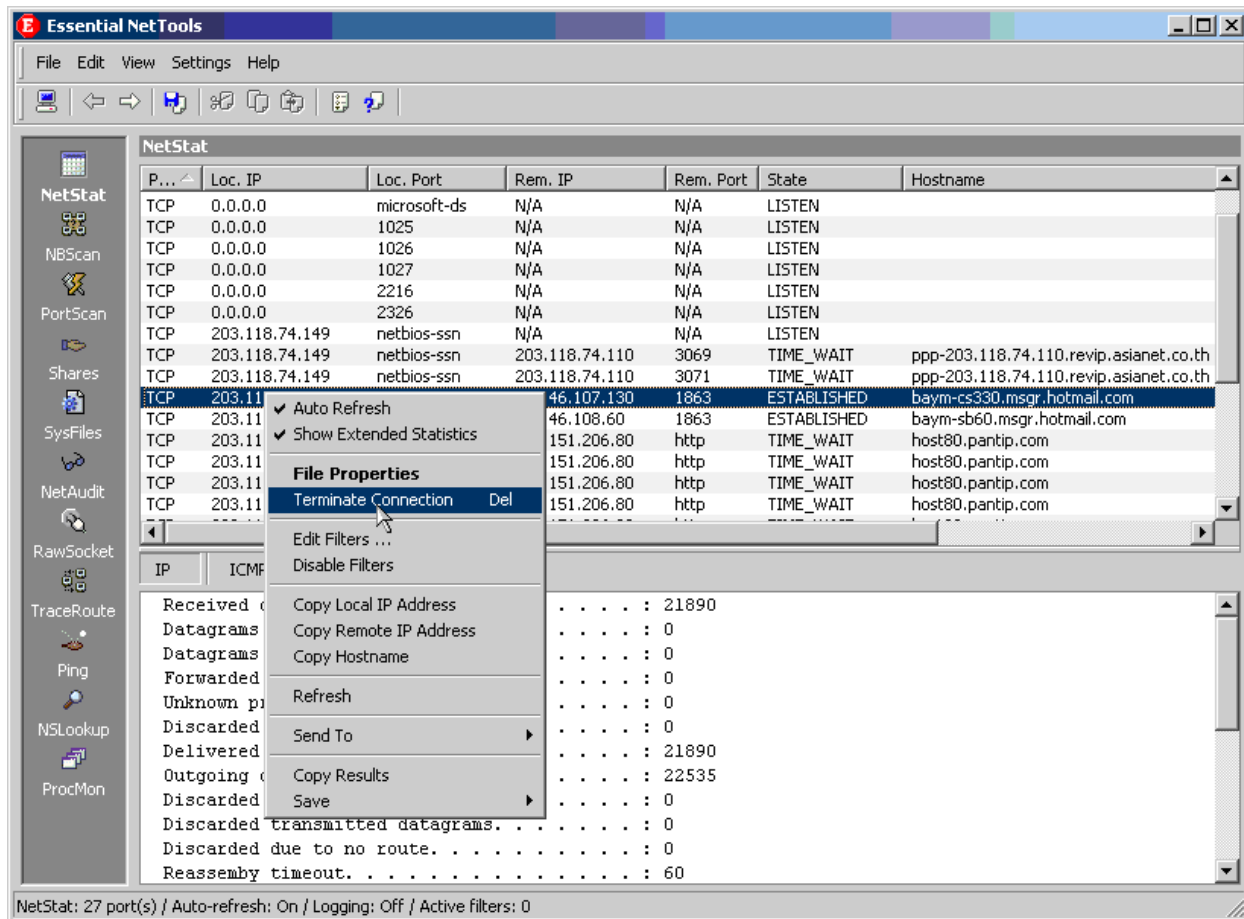
Name \_\_\_\_\_ Student-ID \_\_\_\_\_

## Essential Net Tool

### Using the software



Name \_\_\_\_\_ Student-ID \_\_\_\_\_



**Install and Try the software. (LAB 30 mins)**