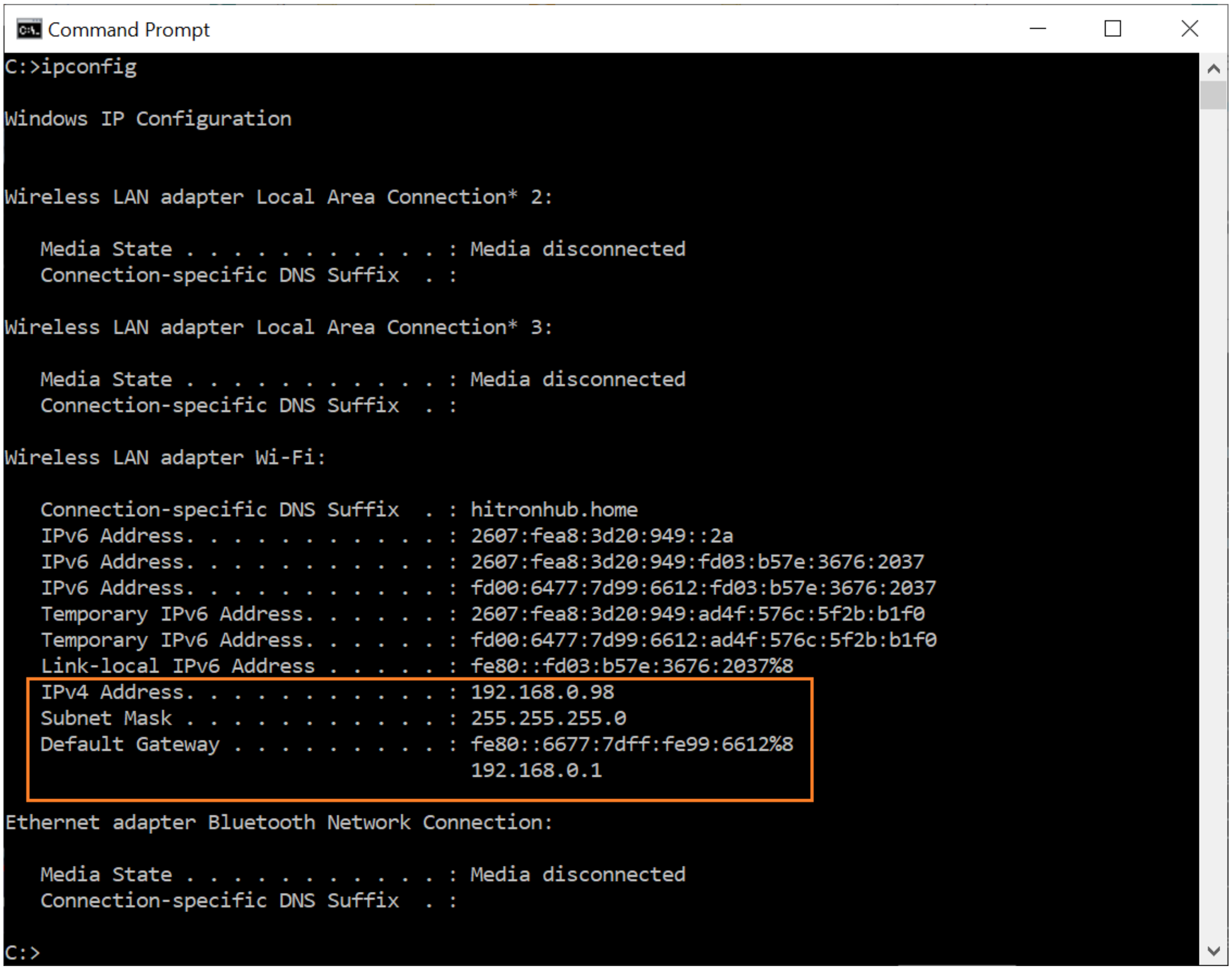
IPconfig - windows

- The ipconfig (short for *IP Configuration*) is a basic, yet popular, *Windows* network command-line utility used to display the TCP/IP network configuration of a computer.
- If you are familiar with Linux, this tool is similiar to *ifconfig*. This tool is often used for troubleshooting network connectivity issues.
- With ipconfig, you can identify the types of network adapaters on your computer, the computer's IP address, the IP addresses of the DNS (Domain Name System) servers being used, and much more.



```
C:>ipconfig

Windows IP Configuration


Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 3:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : hitronhub.home
   IPv6 Address. . . . . . . . . . . : 2607:fea8:3d20:949::2a
   IPv6 Address. . . . . . . . . . . : 2607:fea8:3d20:949:fd03:b57e:3676:2037
   IPv6 Address. . . . . . . . . . . : fd00:6477:7d99:6612:fd03:b57e:3676:2037
   Temporary IPv6 Address. . . . . . : 2607:fea8:3d20:949:ad4f:576c:5f2b:b1f0
   Temporary IPv6 Address. . . . . . : fd00:6477:7d99:6612:ad4f:576c:5f2b:b1f0
   Link-local IPv6 Address . . . . . : fe80::fd03:b57e:3676:2037%8
   IPv4 Address. . . . . . . . . . . : 192.168.0.98
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::6677:7dff:fe99:6612%8
                                       192.168.0.1

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:>
```

ipconfig - Retrieve Basic TCP/IP Network Information

- To get basic network information from your computer, type the following in the command window then press `Enter`: `ipconfig`
- The screenshot example below is the ipconfig output of a particular computer. The output of your ipconfig result will differ depending on your network setup and the type of

network adapters installed on your computer. In our screenshot example, it shows the following basic networking information about the computer from which ipconfig was ran.

IPv4 address: 192.168.0.98

Network subnet mask: 255.255.255.0

Default Gateway: 192.168.0.1

- Please note that unless your computer is connected directly to the Internet (this is rare), the IP address reported by ipconfig will be your local network IP, not your public external IP address.
- While other network details can be retrieved by the ipconfig utility, for most network troubleshooting, this is what is typically needed.

  Note:

  The default gateway is the path used to pass information when the device doesn't know where the destination is. More directly, a default gateway is a router that connects your host to remote network segments. It's the exit point for all the packets in your network that have destinations outside your network.

ipconfig /all - Retrieve All TCP/IP Network Information

Physical Address	This is the MAC address of your network adapter.
DHCP Enabled	Indicates if the network connection is using DHCP or Static IP Address
IPv4 Address	The IP Address of your computer
Default Gateway	The router to which your computer is connected
DHCP Server	Router/server that hands out IP Addresses in your network
DNS Servers	Servers used to translate domain names to IP Addresses
Link-Local IPv6 Address	IPv6 address of your computer (often not used)
Lease Obtained	Date-time when your computer received the IP Address

Note: Subnet mask

A subnet mask is a number that defines a range of IP addresses available within a network. For example, a typical subnet mask for a Class C IP address is:

255.255.255.0

In the example above, the first three sections are full (255 out of 255), meaning the IP addresses of devices within the subnet mask must be identical in the first three sections. The last section of each computer's IP address can be anything from 0 to 255. If the subnet mask is defined as 255.255.255.0, the IP addresses 10.0.1.99 and 10.0.1.100 are in the same subnet, but 10.0.2.100 is not.

Note: On Windows side, The Microsoft ISATAP device is a Inter Site Automatic Tunneling Address Protocol is used to help enterprises transition to an IPv6 infrastructure. The ISATAP adapter encapsulates IPv6 packets by using an IPv4 header.

Note: The Client ID in DHCPv6 consists of two parts: a DHCP Unique Identifier (DUID) and an Identity Association Identifier (IAID). The DUID identifies the client system (rather than just an interface, as in DHCPv4), and the IAID identifies the interface on that system.

Whois – linux, turmux

The whois System

- **The whois system is a listing of records that contains details about both the ownership of domains and the owners.**
- The Internet Corporation for Assigned Names and Numbers (ICANN) regulates domain name registration and ownership, but the list of records is held by many companies, known as registries.
- Anyone can query the list of records. When you do, one of the registries will handle your request and send you details from the appropriate whois record.

  - **Registry:** A company that manages a list containing a set of domain names (there are many of these).
  - **Registrant:** The legal owner of the domain; it’s registered to this person.
  - **Registrar:** A registrant uses a registrar to make his or her registration.

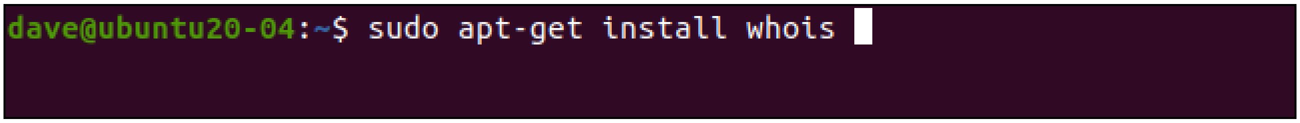
- A whois record contains all the contact information associated with the person, company, or other entity that registered the domain name.
- Some registrations contain more information than others, and some registries return differing amounts of information.
- A typical whois record will contain the following information:

  - **The name and contact information of the registrant:** The owner of the domain.
  - **The name and contact information of the registrar:** The organization that registered the domain name.
  - **The registration date.**
  - **When the information was last updated.**
  - **The expiration date.**

Installing whois

- The `whois` command was already installed on Ubuntu 20.04. If you need to install it on your version of Ubuntu, you can do so with the following command:

```
sudo apt-get install whois
```



```
dave@ubuntu20-04:~$ sudo apt-get install whois
```

Using whois with a Domain Name

```
whois cnn.com
```

Output is as follow:

```
Domain Name: cnn.com

Registry Domain ID: 3269879_DOMAIN_COM-VRSN

Registrar WHOIS Server: whois.corporatedomains.com

Registrar URL: www.cscprotectsbrands.com

Updated Date: 2018-04-10T16:43:38Z

Creation Date: 1993-09-22T04:00:00Z

Registrar Registration Expiration Date: 2026-09-21T04:00:00Z

Registrar: CSC CORPORATE DOMAINS, INC.

Registrar IANA ID: 299

Registrar Abuse Contact Email: domainabuse@cscglobal.com

Registrar Abuse Contact Phone: +1.8887802723

Domain Status: clientTransferProhibited
http://www.icann.org/epp#clientTransferProhibited

Domain Status: serverDeleteProhibited
http://www.icann.org/epp#serverDeleteProhibited
```

```
Domain Status: serverTransferProhibited
http://www.icann.org/epp#serverTransferProhibited

Domain Status: serverUpdateProhibited
http://www.icann.org/epp#serverUpdateProhibited

Registry Registrant ID:

Registrant Name: Domain Name Manager

Registrant Organization: Turner Broadcasting System, Inc.

Registrant Street: One CNN Center

Registrant City: Atlanta

Registrant State/Province: GA

Registrant Postal Code: 30303

Registrant Country: US

Registrant Phone: +1.4048275000

Registrant Phone Ext:

Registrant Fax: +1.4048271995

Registrant Fax Ext:

Registrant Email: tmgroup@turner.com

Registry Admin ID:

Admin Name: Domain Name Manager

Admin Organization: Turner Broadcasting System, Inc.

Admin Street: One CNN Center

Admin City: Atlanta

Admin State/Province: GA

Admin Postal Code: 30303

Admin Country: US

Admin Phone: +1.4048275000

Admin Phone Ext:

Admin Fax: +1.4048271995

Admin Fax Ext:

Admin Email: tmgroup@turner.com

Registry Tech ID:
```

```
Tech Name: TBS Server Operations

Tech Organization: Turner Broadcasting System, Inc.

Tech Street: One CNN Center

Tech City: Atlanta

Tech State/Province: GA

Tech Postal Code: 30303

Tech Country: US

Tech Phone: +1.4048275000

Tech Phone Ext:

Tech Fax: +1.4048271593

Tech Fax Ext:

Tech Email: hostmaster@turner.com

Name Server: ns-576.awsdns-08.net

Name Server: ns-1086.awsdns-07.org

Name Server: ns-47.awsdns-05.com

Name Server: ns-1630.awsdns-11.co.uk

DNSSEC: unsigned
```

- This gives us more or less the same information as the summary, with extra sections about the registrant and their contact details for administrative and technical purposes.
- The registrant name is given as “Domain Name Manager.” Sometimes, for a fee, companies choose to let their registrar register the domain on their behalf under a generic name the registrar maintains for this purpose. That appears to be the case here. However, as the address of the registrant is “1 CCN Center,” it’s obvious who the registrant is.

- This is reasonably self-explanatory. We see various details about the registrar and registry, including contact details, registration dates, and so on. There are a few entries in the list that you might not recognize.
- The Internet Assigned Numbers Authority (IANA) oversees and coordinates things like top-level Domain Name System zones, IP protocol addressing systems, and the list of registries. This registry is number 299, which is indicated in the listing as “IANA ID: 299.”
- The “domain status” lines show the state in which the domain is, and it can be in several simultaneously. The states are defined in the Extensible Provisioning Protocol. Some of these are rarely seen, and others are restricted to certain situations, such as legal disputes.
- The following states are attached to this registration:

- **clientTransferProhibited:** The domain’s registry will reject requests to transfer the domain from the current registrar to another.
- **serverDeleteProhibited:** The domain cannot be deleted.
- **serverTransferProhibited:** The domain cannot be transferred to another registrar.
- **serverUpdateProhibited:** The domain cannot be updated
- The last three are usually enabled at the registrant’s request, or if a legal dispute is in progress. In this case, CNN probably requested these to be enforced to “lock down” the company’s domain.
- “!DNSSEC” stands for Domain Name System Security Extensions, a scheme that allows a DNS name resolver to cryptographically check that the data it received from the DNS zone is valid and hasn’t been tampered with.

Using whois with an IP Address

- Using `whois` with an IP address is just as simple as using it with a domain name. Just specify an IP address after `whois`, like so:

```
whois 205.251.242.103
```

nslookup (Name Server Lookup) – linux, turmux

- nslookup is the name of a program that lets an Internet server administrator or any computer user enter a host name (for example, "whatis.com") and **find out the corresponding IP address or domain name system (DNS) record.** The user can also enter a command for it to do a reverse DNS lookup and find the host name for an IP address that is specified.

Uses of nslookup

- nslookup is used to troubleshoot server connections or for security reasons. Such reasons include guard against phishing attacks, in which a domain name is altered -- for example, by substituting the numeral 1 for a lowercase l -- to make an unfriendly site look friendly and familiar (joes1owerprices.com vs. joeslowerprices.com).

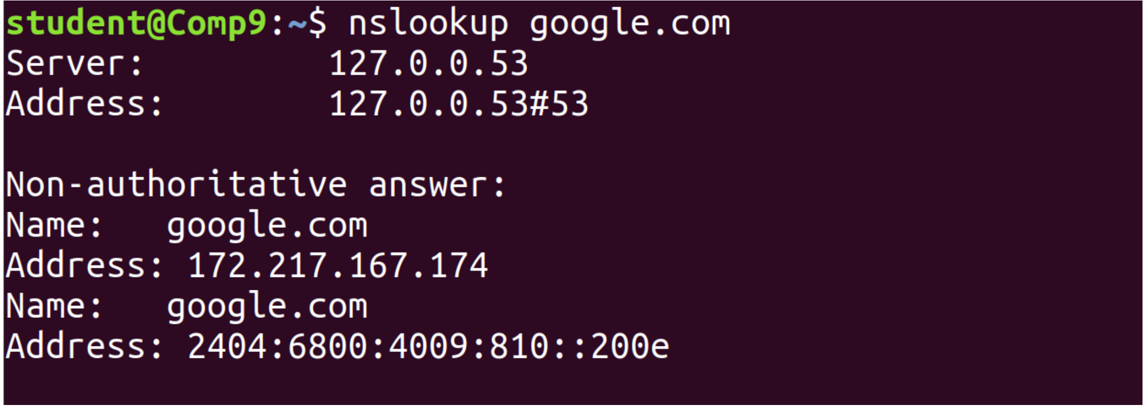
Examples of nslookup commands

- If "WhatIs.com" is entered into a nslookup program, the user would receive the site's IP address as a response, which happens to be 65.214.43.37. If the user enters "65.214.43.37", it would return "sites.techtarget.com".

Syntax:
nslookup [option]

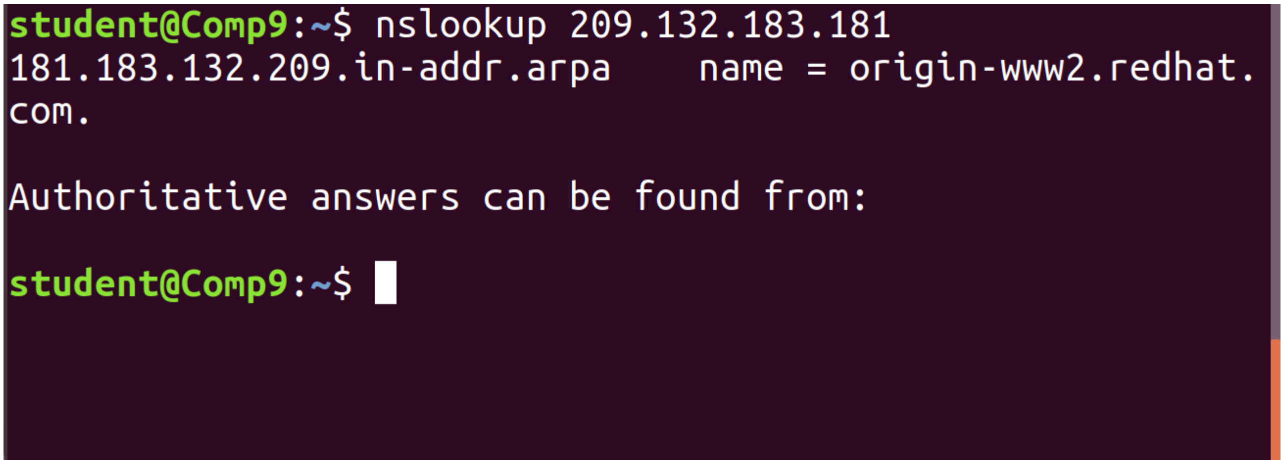
Options of nslookup command:

- **nslookup google.com :**
  nslookup followed by the domain name will display the "A Record" (IP Address) of the domain. Use this command to find the address record for a domain. It queries to domain name servers and gets the details.



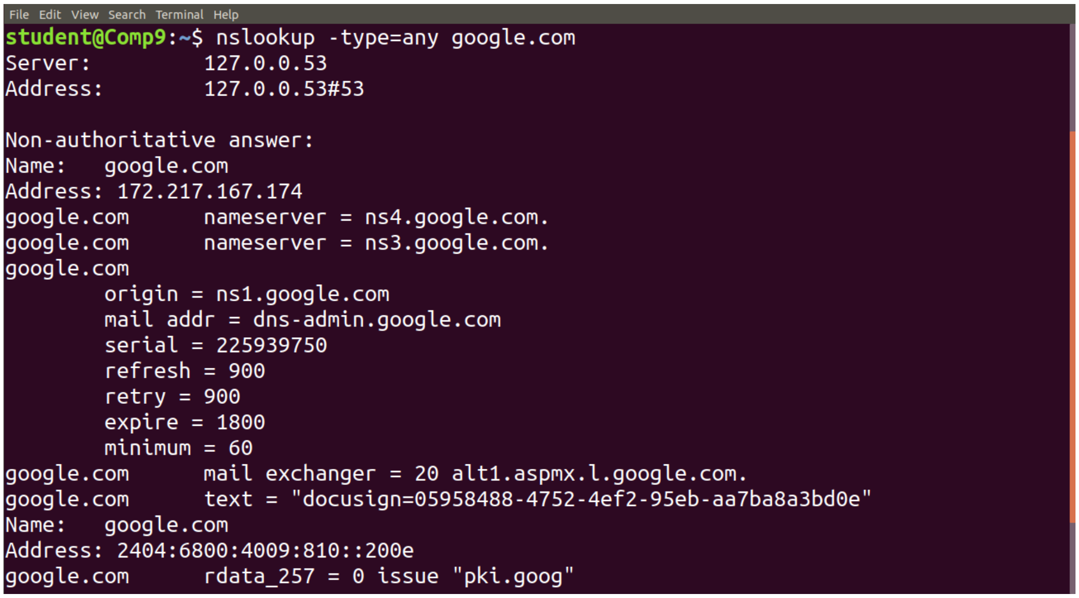
```
student@Comp9:~$ nslookup google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.167.174
Name:   google.com
Address: 2404:6800:4009:810::200e
```

- **nslookup 192.168.0.10:** Reverse DNS lookup
  You can also do the reverse DNS look-up by providing the IP Address as an argument to nslookup.



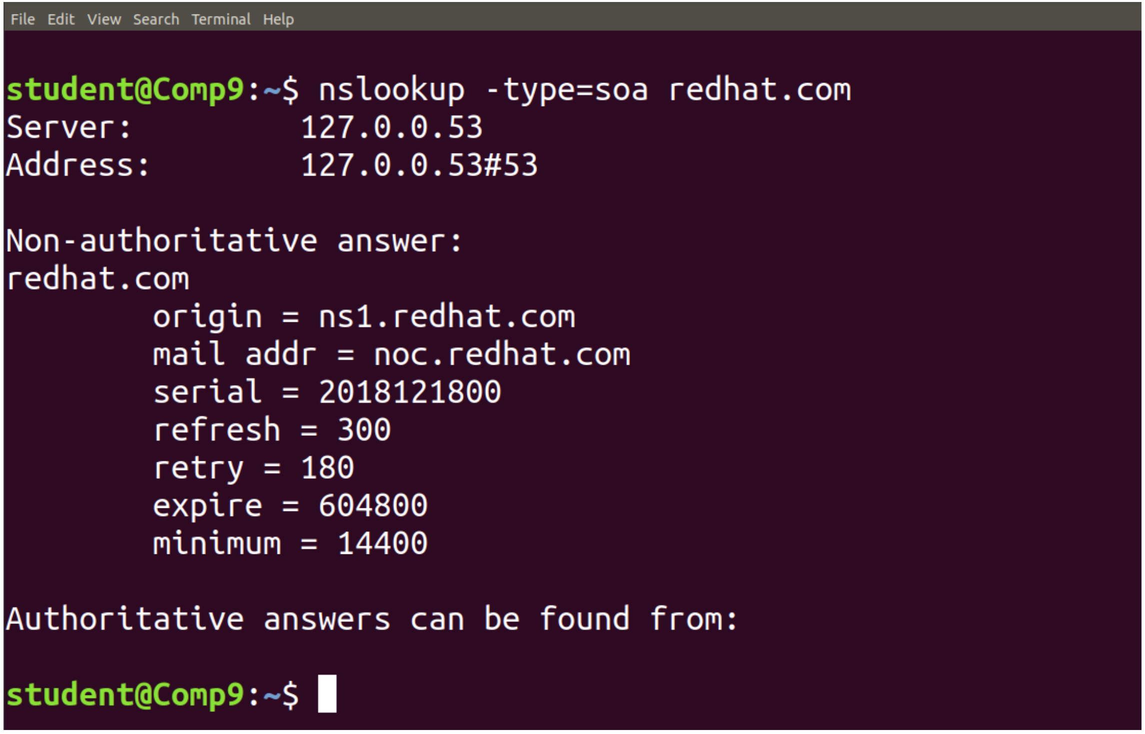
```
student@Comp9:~$ nslookup 209.132.183.181
181.183.132.209.in-addr.arpa    name = origin-www2.redhat.
com.

Authoritative answers can be found from:

student@Comp9:~$
```

- **nslookup -type=any google.com :** Lookup for any record
  We can also view all the available DNS records using the *-type=any* option.



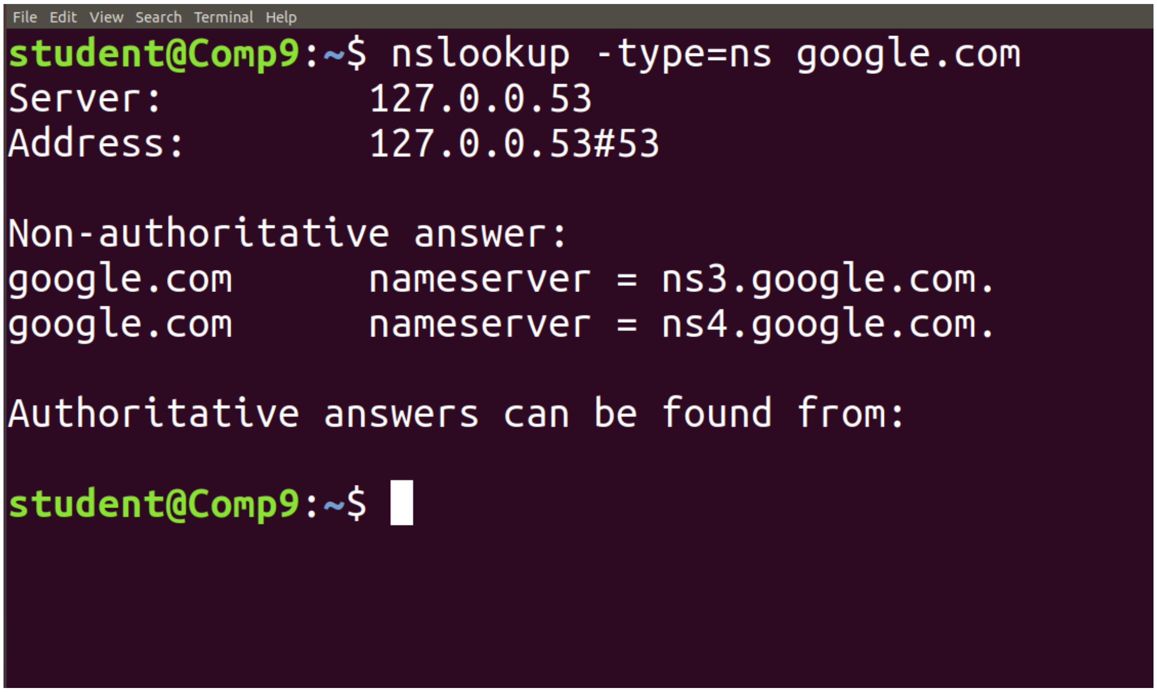
```
student@Comp9:~$ nslookup -type=any google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.167.174
google.com      nameserver = ns4.google.com.
google.com      nameserver = ns3.google.com.
google.com
        origin = ns1.google.com
        mail addr = dns-admin.google.com
        serial = 225939750
        refresh = 900
        retry = 900
        expire = 1800
        minimum = 60
google.com      mail exchanger = 20 alt1.aspmx.l.google.com.
google.com      text = "docusign=05958488-4752-4ef2-95eb-aa7ba8a3bd0e"
Name:   google.com
Address: 2404:6800:4009:810::200e
google.com      rdata_257 = 0 issue "pki.goog"
```

- **nslookup -type=soa redhat.com :** Lookup for an soa record
  SOA record (start of authority), provides the authoritative information about the domain, the e-mail address of the domain admin, the domain serial number, etc…



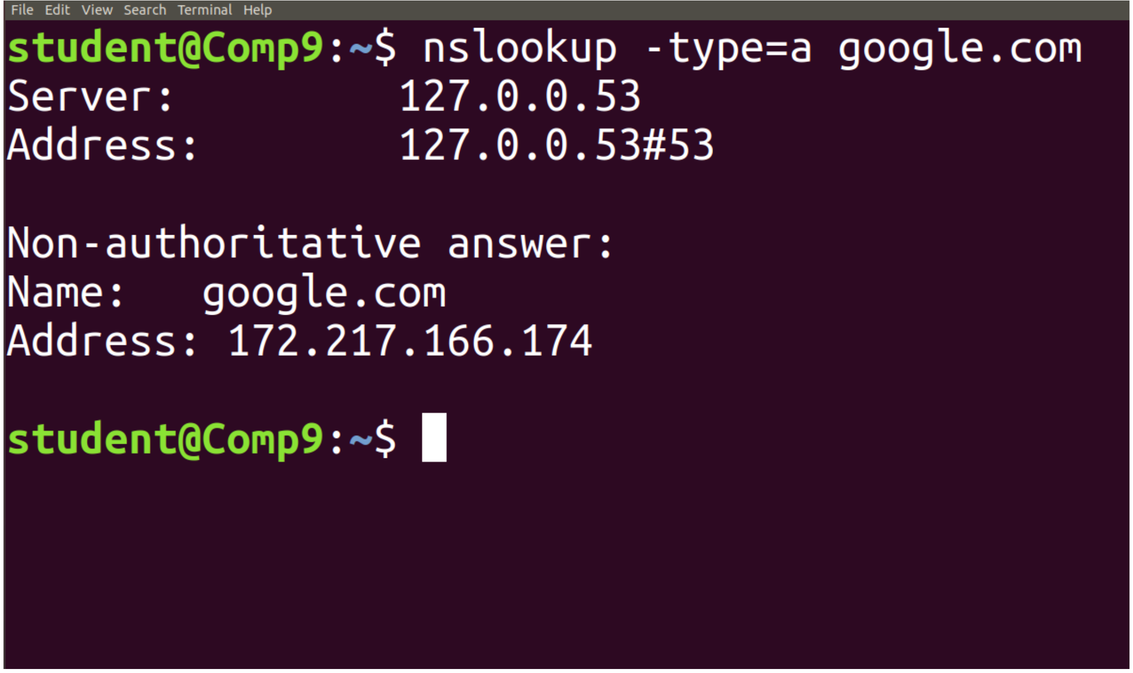
```
student@Comp9:~$ nslookup -type=soa redhat.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
redhat.com
        origin = ns1.redhat.com
        mail addr = noc.redhat.com
        serial = 2018121800
        refresh = 300
        retry = 180
        expire = 604800
        minimum = 14400

Authoritative answers can be found from:

student@Comp9:~$ 
```

- **nslookup -type=ns google.com :** Lookup for an ns record
  NS (Name Server) record maps a domain name to a list of DNS servers authoritative for that domain. It will output the name serves which are associated with the given domain.



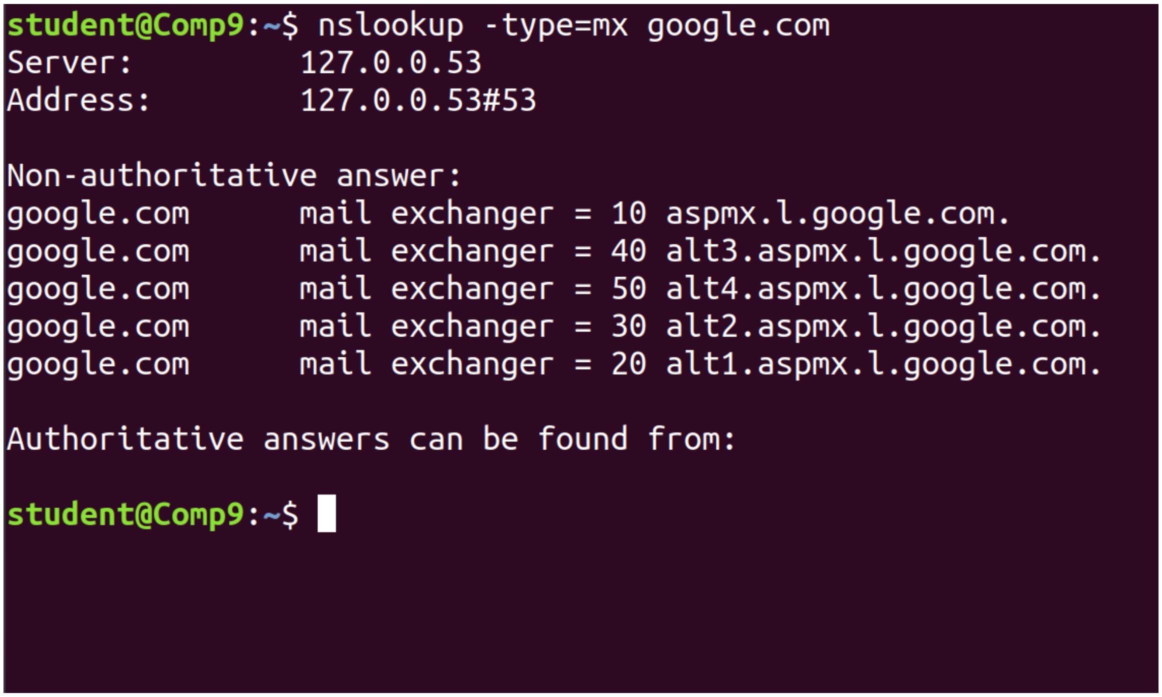
```
student@Comp9:~$ nslookup -type=ns google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
google.com      nameserver = ns3.google.com.
google.com      nameserver = ns4.google.com.

Authoritative answers can be found from:

student@Comp9:~$ 
```

- **nslookup -type=a google.com :** Lookup for an a record
  We can also view all the available DNS records for a particular record using the -*type=a* option.



```
student@Comp9:~$ nslookup -type=a google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.166.174

student@Comp9:~$ 
```

- **nslookup -type=mx google.com :** Lookup for an mx record
  MX (Mail Exchange) record maps a domain name to a list of mail exchange servers for that domain. The MX record tells that all the mails sent to “google.com” should be routed to the Mail server in that domain.



```
student@Comp9:~$ nslookup -type=mx google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
google.com      mail exchanger = 10 aspmx.l.google.com.
google.com      mail exchanger = 40 alt3.aspmx.l.google.com.
google.com      mail exchanger = 50 alt4.aspmx.l.google.com.
google.com      mail exchanger = 30 alt2.aspmx.l.google.com.
google.com      mail exchanger = 20 alt1.aspmx.l.google.com.

Authoritative answers can be found from:

student@Comp9:~$ 
```

Nslookup – windows

There are many reasons why you might need to check the status of your Domain Name System (DNS) records. For example, you might need to verify that updates are correct or troubleshoot issues with accessing a service.

Check a DNS record

To check a specific DNS record, you need to specify the `nslookup` command, an optional record type (for example, `A`, `MX`, or `TXT`), and the host name that you want to check.

Note: If you omit the record type, it defaults to `A`.

The following example shows how to check `A` records for **rackspace.co.uk**:

```
C:\Users\Administrator>nslookup rackspace.co.uk
Server:  cachens1.lon.rackspace.com>
Address:  83.138.151.80

Non-authoritative answer:

Name:    rackspace.co.uk
Address:  212.64.133.165
```

The first two lines of output specify the server to which the request was directed. This server is the default server that your system uses for DNS name resolution.

[**Note**: An authoritative answer comes from a nameserver that is considered authoritative for the domain which it's returning a record for (one of the nameservers in the list for the domain you did a lookup on), and a non-authoritative answer comes from anywhere else (a nameserver not in the list for the domain you did a lookup on).

for example, If I did an nslookup of maps.google.com right now, I would get a response from one of my configured nameservers. (Either from my ISP, or my domain.) It would come back as non-authoritative because neither my ISP's nameservers, nor my own are in the list of nameservers for google.com. They aren't Google's nameservers, so they're not the authoritative source that creates the NS records.]

The second section gives the name of the record and the corresponding Internet Protocol (IP) address. However, the answer in this section is non-authoritative because it originates from a server (**cachens1.lon.rackspace.com**) that isn't the root source for those records.

Get an authoritative answer

To get an authoritative answer you need to specify the authoritative (primary) name server at the end of the request.

Use the `-type=soa` option to tell `nslookup` to display the authoritative name server, as shown in the following example:

```
C:\Users\Administrator>nslookup -type=soa rackspace.co.uk
Server:  cachens1.lon.rackspace.com>
Address:  83.138.151.80

Non-authoritative answer:
rackspace.co.uk
        primary name server = ns.rackspace.com
        responsible mail addr = hostmaster.rackspace.com
        serial  = 1415913000
        refresh = 3600 (1 hour)
        retry   = 300 (5 mins)
        expire  = 1814400 (21 days)
        default TTL = 300 (5 mins)

ns.rackspace.com        internet address = 69.20.95.4
```

The address labeled `primary name server` is the DNS authority for the domain.

If you add the address of the authoritative name server (**ns.rackspace.com**) to the first command, the record is now checked against that name server.

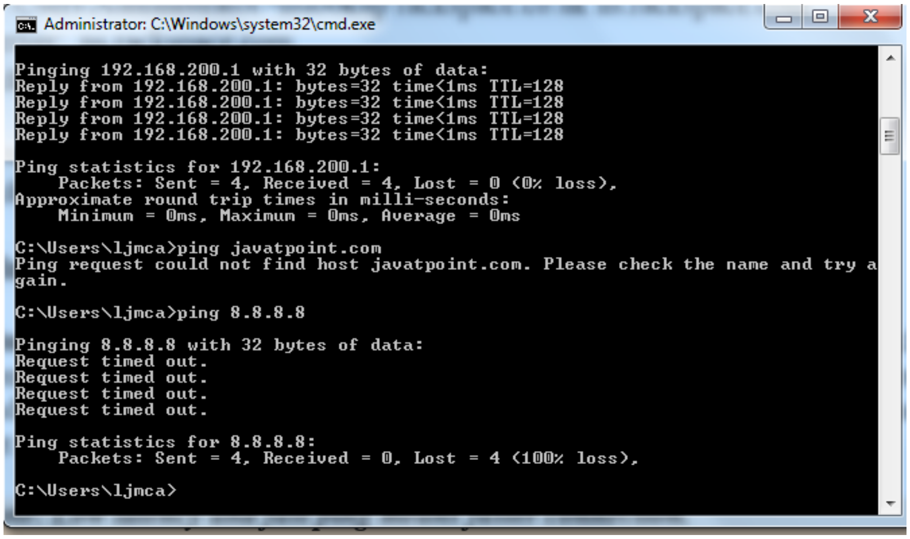
```
C:\Users\Administrator>nslookup rackspace.co.uk ns.rackspace.com
Server:  ns.rackspace.com
Address:  69.20.95.4

Name:    rackspace.co.uk
Address:  212.64.133.165
```

Ping – windows , linux

Ping is short for **Packet Internet Groper**. This command is mainly used for checking the network connectivity among host/server and host. The ping command takes the URL or IP address as input and transfers the data packet to a specified address along with a **"PING"** message. Then, it will get a reply from the host/server. This time is known as **"latency"**.

Note: Low latency and fast ping means faster connection.

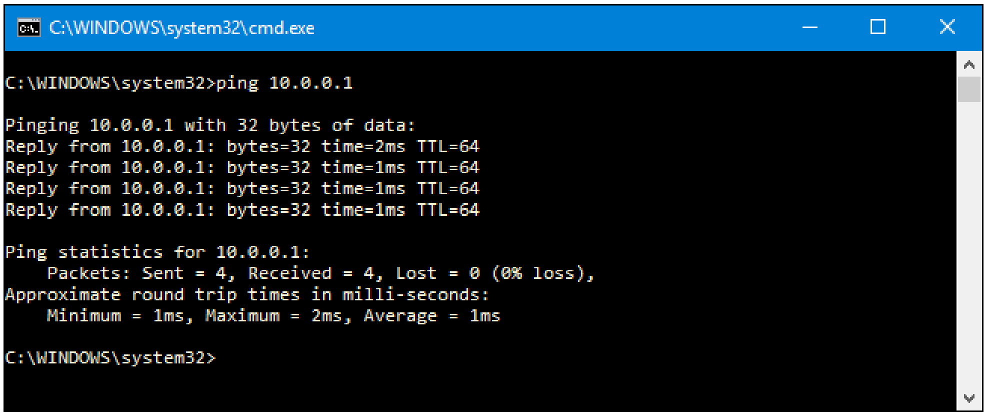
Run on windows:



```
Pinging 192.168.200.1 with 32 bytes of data:
Reply from 192.168.200.1: bytes=32 time<1ms TTL=128
Reply from 192.168.200.1: bytes=32 time<1ms TTL=128
Reply from 192.168.200.1: bytes=32 time<1ms TTL=128
Reply from 192.168.200.1: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.200.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\ljmca>ping javatpoint.com
Ping request could not find host javatpoint.com. Please check the name and try a
gain.

C:\Users\ljmca>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\ljmca>
```

That response shows the URL you're pinging, 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.



```
C:\WINDOWS\system32>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=2ms TTL=64
Reply from 10.0.0.1: bytes=32 time=1ms TTL=64
Reply from 10.0.0.1: bytes=32 time=1ms TTL=64
Reply from 10.0.0.1: bytes=32 time=1ms TTL=64

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms

C:\WINDOWS\system32>
```

The ping command permits us to:

- Test our Internet connection.
- Check if the remote machine is active.
- Analyse when there are network problems such as high latency or dropped packages.

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.howtogeek.com) 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.
- Ping a URL to resolve its IP address. If you want 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.
- 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.

Note: You may not get a ping response from other computers on your local network because the built-in firewalls on those devices prevent them from responding to ping requests. If you want to be able to ping those devices, you'll need to turn off that setting to allow pings through the firewall.

Tracert – windows

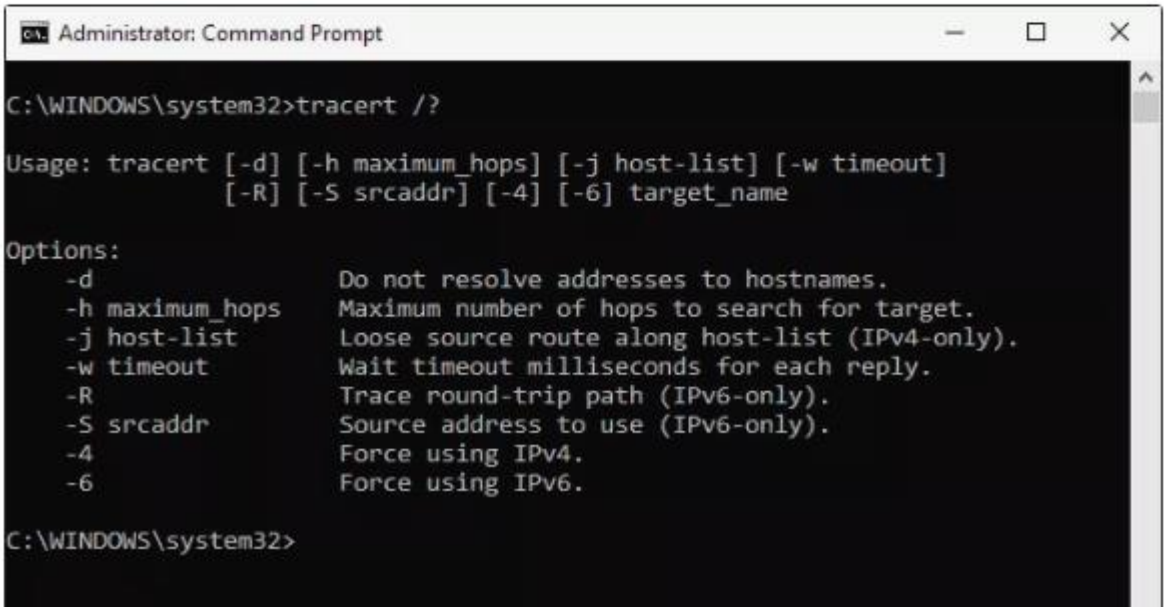
The tracert command is a Command Prompt command that's used to show several details about the path that a packet takes from the computer or device you're on to whatever destination you specify.

You might also sometimes see the tracert command referred to as the *trace route command* or *traceroute command*.

Tracert Command Syntax

If you know how to read command syntax, the syntax for tracert is pretty straight-forward:

tracert [**-d**] [**-h** *MaxHops*] [**-w** *TimeOut*] [**-4**] [**-6**] *target* [/**?**]



```
C:\WINDOWS\system32>tracert /?

Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
               [-R] [-S srcaddr] [-4] [-6] target_name

Options:
    -d                 Do not resolve addresses to hostnames.
    -h maximum_hops    Maximum number of hops to search for target.
    -j host-list       Loose source route along host-list (IPv4-only).
    -w timeout         Wait timeout milliseconds for each reply.
    -R                 Trace round-trip path (IPv6-only).
    -S srcaddr         Source address to use (IPv6-only).
    -4                 Force using IPv4.
    -6                 Force using IPv6.

C:\WINDOWS\system32>
```

Tracert Command Options

Item	**Description**
-d	This option prevents tracert from resolving IP addresses to hostnames, often resulting in much faster results.
-h *MaxHops*	This tracert option specifies the maximum number of hops in the search for the *target*. If you do not specify *MaxHops*, and a *target* has not been found by 30 hops, tracert will stop looking.
-w *TimeOut*	You can specify the time, in milliseconds, to allow each reply before timeout using this tracert option.

Tracert Command Options

-4 This option forces tracert to use IPv4 only.

-6 This option forces tracert to use IPv6 only.

target This is the destination, either an IP address or hostname.

/? Use the help switch with the tracert command to show detailed help about the command's several options.

Tracert Command Examples

```
tracert 192.168.1.1
```

In the above example, the tracert command is used to show the path from the networked computer on which the tracert command is being executed by a network device, in this case, a router on a local network, that's assigned the *192.168.1.1* IP address.

The result displayed on the screen will look something like this:

```
Tracing route to 192.168.1.1 over a maximum of 30 hops
1 <1 ms <1 ms <1 ms 192.168.1.254
2 <1 ms <1 ms <1 ms 192.168.1.1
Trace complete.
```

In this example, you can see that tracert found a network device using the IP address of *192.168.1.254*, let's say a network switch, followed by the destination, *192.168.1.1*, the router.

Output explanation format:

```
Hop   RTT1    RTT2   RTT3   Name [IP Address]

4   13 ms    8 ms    9 ms  pos-0-3-0-0-
cr01.newyork.ny.ibone.comcast.net [68.86.90.57]
```

Hop number: The specific hop number in the path from the sender to the destination.

Round Trip Time (RTT): The time it takes for a packet to get to a hop and back, displayed in milliseconds (ms). By default, tracert sends three packets to each hop, so the output lists three roundtrip times per hop.

If an asterisk (*) appears for RTT, then a packet was not returned within the expected timeframe.

- *One or two asterisks for a hop do not necessarily indicate packet loss at the final destination*. Many Internet routers *intentionally discard ping or traceroute packets*, but this has no bearing on applications that use these routers. This practice is called ICMP Rate Limiting and is used to prevent routers from being impacted by denial-of-service attacks.

- Three asterisks followed by the “Request timed out” message may appear for several reasons.
- Name: The fully qualified domain name (FQDN) of the system. Many times the FQDN may provide an indication of where the hop is physically located. If the Name doesn’t appear in the output, the FQDN wasn’t found. It isn’t necessarily indicative of a problem, if an FQDN isn’t found.

  IP Address: The Internet Protocol (IP) address of that specific router or host associated with the Name.

```
tracert www.google.com
```

With the tracert command shown above, we're asking tracert to show us the path from the local computer all the way to the network device with the hostname *www.google.com*.

```
Tracing route to www.l.google.com [209.85.225.104]
over a maximum of 30 hops:
1 <1 ms <1 ms <1 ms 10.1.0.1
2 35 ms 19 ms 29 ms 98.245.140.1
3 11 ms 27 ms 9 ms te-0-3.dnv.comcast.net [68.85.105.201]
...
13 81 ms 76 ms 75 ms 209.85.241.37
14 84 ms 91 ms 87 ms 209.85.248.102
15 76 ms 112 ms 76 ms iy-f104.1e100.net [209.85.225.104]
Trace complete.
```

In this example, we can see that tracert identified fifteen network devices including our router at *10.1.0.1* and all the way through to the *target* of *www.google.com*, which we now know uses the public IP address of *209.85.225.104*, one of Google's many IP addresses.

Hops 4 through 12 were excluded above just to keep the example simple. If you were executing a real tracert, those results would all show up on screen.

```
tracert -d www.yahoo.com
```

With this tracert command example, we're again requesting the path to a website, this time *www.yahoo.com*, but now we're preventing tracert from resolving hostnames by using the *-d* option.

```
Tracing route to any-fp.wa1.b.yahoo.com [209.191.122.70]
over a maximum of 30 hops:
1 <1 ms <1 ms <1 ms 10.1.0.1
2 29 ms 23 ms 20 ms 98.245.140.1
3 9 ms 16 ms 14 ms 68.85.105.201
...
13 98 ms 77 ms 79 ms 209.191.78.131
14 80 ms 88 ms 89 ms 68.142.193.11
15 77 ms 79 ms 78 ms 209.191.122.70
```

```
Trace complete.
```

We can see that tracert again identified fifteen network devices including our router at *10.1.0.1* and all the way through to the *target* of *www.yahoo.com*, which we can assume uses the public IP address of *209.191.122.70*.

As you can see, tracert didn't resolve any hostnames this time, which significantly sped up the process.

```
tracert -h 3 lifewire.com > z:\tracertresults.txt
```

In this last example of the tracert command in Windows, we're using *-h* to limit the hop count to *3*, but instead of displaying the results in Command Prompt, we'll use the > redirection operator to send it all to a TXT file located on *Z:*, an external hard drive.

Here are some example results of this last command:

```
Tracing route to lifewire.com [151.101.66.114]
over a maximum of 3 hops:
1  <1 ms  <1 ms  <1 ms testwifi.here [192.168.86.1]
2   1 ms   1 ms  <1 ms 192.168.1.1
3  17 ms  16 ms  17 ms giantwls-64-71-222-1.giantcomm.net [64.71.222.1]
Trace complete.
```

Tracepath/traceroute – termux, linux

What Is Traceroute/tracepath?

Traceroute is a network diagnostic tool that tracks the path of a packet of data as it travels from your computer to a destination over the internet. Running a traceroute lets you see where your connection is slow or unresponsive.

You can think of the traceroute tool like a traffic map of your internet connection. When you run a traceroute, you will see all the “hops,” or routers that three separate packets are pushed through on their way to a destination. It will also show you your network’s latency, or how long it took for each packet to travel from one hop to the next.

Note: Each entry, or hop, is **a location that the packet passes through to reach its final destination**. If the trace times out on a certain hop it can mean there is a problem at that location, or that the route is incorrect, preventing the packet from reaching the destination.

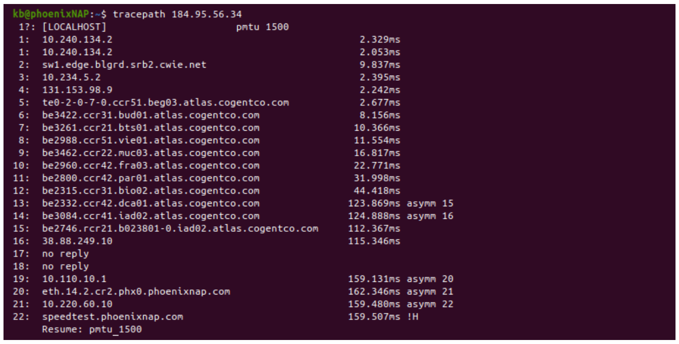
It traces path to *destination* discovering MTU along this path. Every network interface is set with an MTU (**Maximum Transmission Unit**) value that defines the byte size of the largest protocol data unit that is allowed to pass.

The number of probes is **the number of packets that is sent per hop**.

asymm means the **the path to the hop and back have been different (asymmetric)**. This usually happens when there is some link in one direction jammed or the network architecture encourages different paths for the different directions. The number after asymm shows the grade of asymmetry (i.e. how many hops are different)

```
root@mops:~ # tracepath 3ffe:2400:0:109::2
 1?: [LOCALHOST]                         pmtu 1500
 1:  dust.inr.ac.ru                  0.411ms
 2:  dust.inr.ac.ru        asymm  1   0.390ms pmtu 1480
 2:  3ffe:2400:0:109::2              463.514ms reached
     Resume: pmtu 1480 hops 2 back 2
```

- The first column shows TTL of the probe, followed by colon. Usually value of TTL is obtained from reply from network, but sometimes reply does not contain necessary information and we have to guess it. In this case the number is followed by ?.
- The second column shows the network hop, which replied to the probe. It is either address of router or word [LOCALHOST], if the probe was not sent to the network.
- The rest of line shows miscellaneous information about path to the corresponding network hop. As rule it contains value of RTT. Additionally, it can show Path MTU, when it changes. If the path is asymmetric or the probe finishes before it reach prescribed hop, difference between number of hops in forward and backward direction is shown following keyword asymm. This information is not reliable. F.e. the third line shows asymmetry of 1, it is because the first probe with TTL of 2 was rejected at the first hop due to Path MTU Discovery.
- The last line summarizes information about all the path to the destination, it shows detected Path MTU, amount of hops to the destination and our guess about amount of hops from the destination to us, which can be different when the path is asymmetric.
- Another output:



```
kb@phoenixNAP:~$ tracepath 184.95.56.34
 1?: [LOCALHOST]                      pmtu 1500
 1:  10.240.134.2                                          2.329ms
 1:  10.240.134.2                                          2.053ms
 2:  sw1.edge.blgrd.srb2.cwie.net                          9.837ms
 3:  10.234.5.2                                            2.395ms
 4:  131.153.98.9                                          2.242ms
 5:  te0-2-0-7-0.ccr51.beg03.atlas.cogentco.com            2.677ms
 6:  be3422.ccr31.bud01.atlas.cogentco.com                 8.156ms
 7:  be3261.ccr21.bts01.atlas.cogentco.com                10.366ms
 8:  be2988.ccr51.vie01.atlas.cogentco.com                11.554ms
 9:  be3462.ccr22.muc03.atlas.cogentco.com                16.817ms
10:  be2960.ccr42.fra03.atlas.cogentco.com                22.771ms
11:  be2800.ccr42.par01.atlas.cogentco.com                31.998ms
12:  be2315.ccr31.bio02.atlas.cogentco.com                44.418ms
13:  be2332.ccr42.dca01.atlas.cogentco.com               123.869ms asymm 15
14:  be3084.ccr41.iad02.atlas.cogentco.com               124.888ms asymm 16
15:  be2746.rcr21.b023801-0.iad02.atlas.cogentco.com     112.367ms
16:  38.88.249.10                                        115.346ms
17:  no reply
18:  no reply
19:  10.110.10.1                                         159.131ms asymm 20
20:  eth.14.2.cr2.phx0.phoenixnap.com                    162.346ms asymm 21
21:  10.220.60.10                                        159.480ms asymm 22
22:  speedtest.phoenixnap.com                            159.507ms !H
     Resume: pmtu_1500
```

telnet – windows

telnet (**tel**etype **net**work) is a network protocol for two-way text-based communication through a CLI, allowing remote access. Telnet is vulnerable to cybersecurity attacks because it lacks encryption methods compared to the more modern SSH. However, it is still helpful for tasks that do not involve transmitting sensitive information.

Prerequisites

- Windows OS with administrator privileges
- Access to the command prompt
- An address and port to test

What is Telnet?

Telnet is a client-server protocol predating the TCP protocol. The network protocol allows a user to log into another computer within the same network through a TCP/IP connection.

A client machine running the Telnet client connects to a CLI on a remote device, most commonly a dedicated platform. Telnet is lightweight and fast, making it the preferred option in some use cases:

- Initial network hardware configuration.
- Remote access to trusted internal networks.
- Testing for open or used ports.
- Troubleshooting mail and web servers.

How Does Telnet Work?

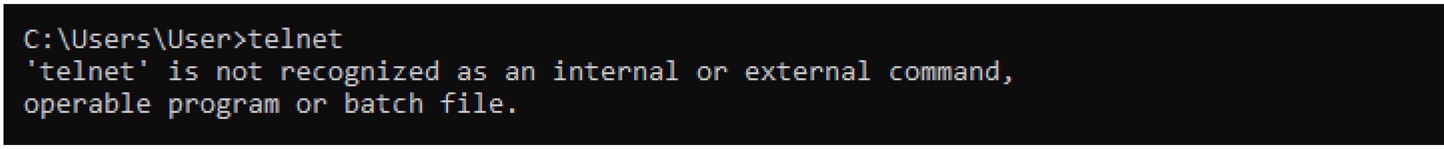
The Telnet protocol creates a communication path through a virtual terminal connection. The data distributes in-band with Telnet control information over the transmission control protocol (TCP).

Unlike other TCP/IP protocols, Telnet provides a log-in screen and allows logging in as the remote device's actual user when establishing a connection on port 23. This type of access grants direct control with all the same privileges as the owner of the credentials.

Telnet comes with a command accessible from the command line in Windows.
The `telnet` command also exists for macOS and Linux operating systems.

How to Enable Telnet on Windows 10?

In Windows systems, Telnet is disabled by default. To check if Telnet is already activated, open your command line, and run `telnet`:



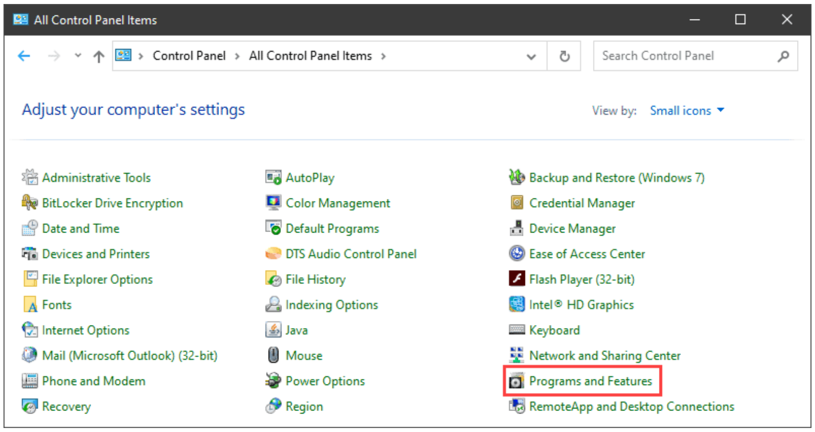
```
C:\Users\User>telnet
'telnet' is not recognized as an internal or external command,
operable program or batch file.
```

If the command prompt does not recognize the command, there are two possible ways to enable the Telnet client in Windows.

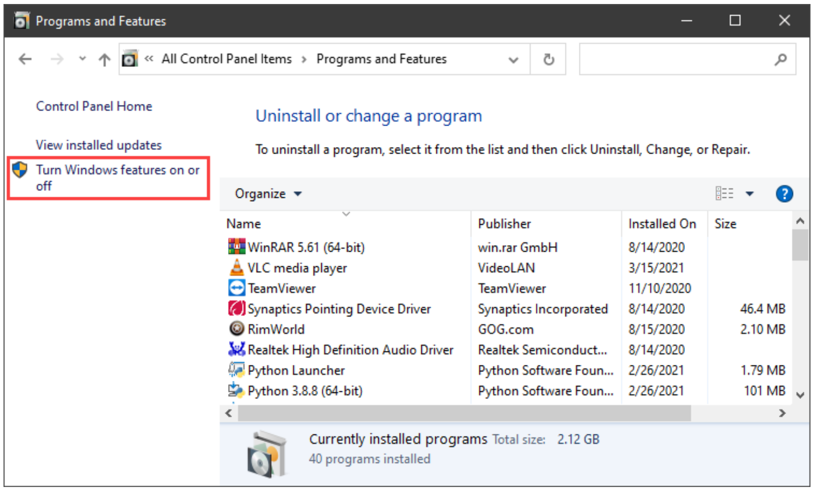
Option 1: Enable Telnet using GUI

To activate the Telnet command using the GUI:

1. Open the **Programs and Features** options in Control Panel:



2. Click the **Turn Windows features on or off** setting:


Programs and Features
« All Control Panel Items > Programs and Features
Control Panel Home
View installed updates
Turn Windows features on or off
Organize

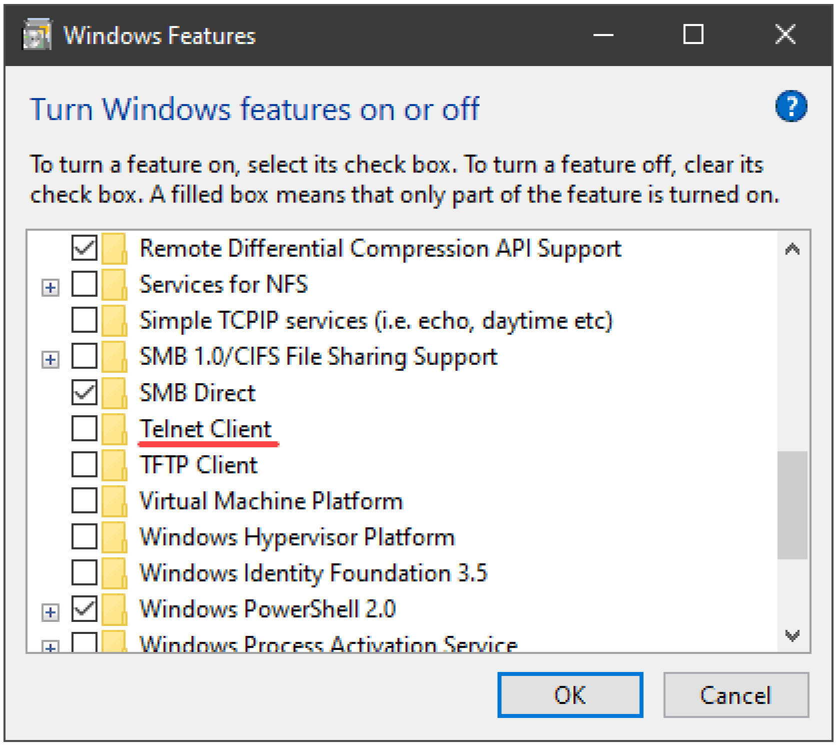
Uninstall or change a program

To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.

Name	Publisher	Installed On	Size
WinRAR 5.61 (64-bit)	win.rar GmbH	8/14/2020	
VLC media player	VideoLAN	3/15/2021	
TeamViewer	TeamViewer	11/10/2020	
Synaptics Pointing Device Driver	Synaptics Incorporated	8/14/2020	46.4 MB
RimWorld	GOG.com	8/15/2020	2.10 MB
Realtek High Definition Audio Driver	Realtek Semiconduct...	8/14/2020	
Python Launcher	Python Software Foun...	2/26/2021	1.79 MB
Python 3.8.8 (64-bit)	Python Software Foun...	2/26/2021	101 MB

Currently installed programs Total size: 2.12 GB
40 programs installed

3. Locate the **Telnet Client** option on the list, select it and click **OK** to install the feature:



Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ☑ Remote Differential Compression API Support
- ⊞ ☐ Services for NFS
- ☐ Simple TCPIP services (i.e. echo, daytime etc)
- ⊞ ☐ SMB 1.0/CIFS File Sharing Support
- ☑ SMB Direct
- ☐ Telnet Client
- ☐ TFTP Client
- ☐ Virtual Machine Platform
- ☐ Windows Hypervisor Platform
- ☐ Windows Identity Foundation 3.5
- ⊞ ☑ Windows PowerShell 2.0
- ⊞ ☐ Windows Process Activation Service

4. When Windows completes the requested change, click **Close**.

5. Open the command prompt and run `telnet` to open the Microsoft Telnet Client:

```
Welcome to Microsoft Telnet Client

Escape Character is 'CTRL+]'

Microsoft Telnet>
```

6. Run `quit` to exit the Telnet client.

Option 2: Enable Telnet Using Command Prompt

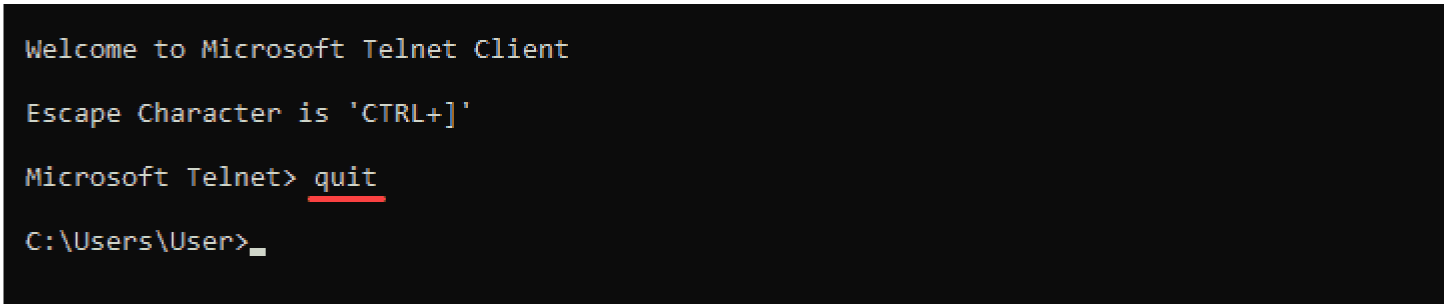
To activate the Telnet client from the command prompt:

1. In the command prompt, run:

```
pkgmgr /iu:"TelnetClient"
```

2. Restart the command prompt and run `telnet` to open the Microsoft Telnet Client.

3. Run `quit` to exit the client:



```
Welcome to Microsoft Telnet Client

Escape Character is 'CTRL+]'

Microsoft Telnet> quit

C:\Users\User>_
```

How to Use Telnet in Windows to Test Open Ports

The Telnet syntax for testing open ports is:

```
telnet <address> <port number>
```

The command accepts both symbolic and numeric addresses. For example:

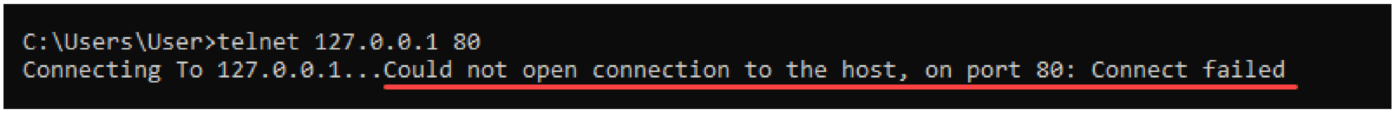
```
telnet towel.blinkenlights.nl 23
```

Or alternatively:

```
telnet 127.0.0.1 80
```

After running the command, one of the following three options happen:

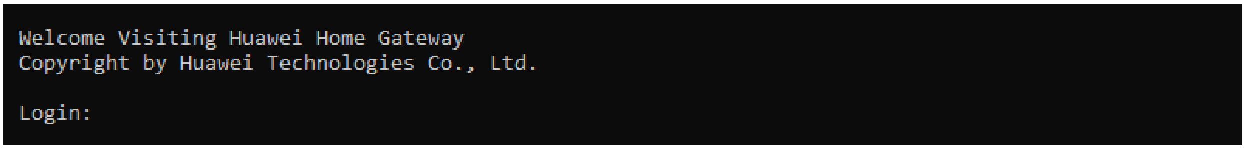
1. The command throws an error, indicating the port is not available for connection:



```
C:\Users\User>telnet 127.0.0.1 80
Connecting To 127.0.0.1...Could not open connection to the host, on port 80: Connect failed
```

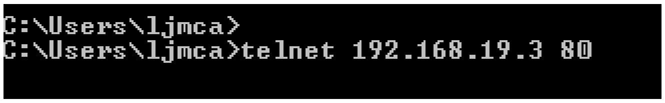
2. The command goes to a blank screen, indicating the port is available.

3. Running the command on an open port 23 displays the screen of the telnet host, confirming an established Telnet connection:



```
Welcome Visiting Huawei Home Gateway
Copyright by Huawei Technologies Co., Ltd.

Login:
```

For checking HTTP connection on port 80



```
C:\Users\ljmca>
C:\Users\ljmca>telnet 192.168.19.3 80
```

telnet www.javatpoint.com 80

If it displays blank screen that means that port is open.

Conclusion

The Telnet communication protocol provides a way to establish a direct connection with a remote host. Although not a secure option for most tasks, there are use cases where Telnet is a viable option.

Running the telent client does not require a telnet server on the client end. If you aren't connecting to port 23 (telnet), a telnet server is not required on the remote end either.

People often use a telnet client to test connectivity to services on remote servers to verify they are not blocked by firewalls or not responding at all. All you are doing is making a TCP connection from a client to a server... it will work on any open TCP port which is not blocked.

Note: Verify you have the permissions to telnet from your computer over the network and to another computer. Verify no firewalls between you and the computer or device you tried connecting to are being blocked. Also, verify telnet service is enabled on both computers.

Secure Shell - linux

There are many ways to establish a connection with a remote machine depending on the operating system you are running, but the two most used protocols are:

- Secure Shell (SSH) for Linux-based machines
- Remote Desktop Protocol (RDP) for Windows-based machines

These tools allow you to gain access and remotely manage other computers, transfer files, and do virtually anything you can do while physically sitting in front of the machine.

Before you can **establish a secure remote desktop protocol** with a remote machine, there are a few basic requirements to meet:

- The remote computer must be turned on at all times and have a network connection.

- The client and server applications need to be installed and enabled.
- You need the IP address or the name of the remote machine you want to connect to.
- You need to have the necessary permissions to access the remote computer.
- Firewall settings need to allow the remote connection.

What is SSH?

Secure Shell, sometimes referred to as **Secure Socket Shell**, is a protocol which allows you to connect securely to a remote computer or a server by using a text-based interface.

When a secure SSH connection is established, a shell session will be started, and you will be able to manipulate the server by typing commands within the client on your local computer.

System and network administrators use this protocol the most, as well as anyone who needs to manage a computer remotely in a highly secure manner.

How Does SSH Work?

In order to establish an SSH connection, you need two components: a client and the corresponding server-side component. An SSH client is an application you install on the computer which you will use to connect to another computer or a server. The client uses the provided remote host information to initiate the connection and if the credentials are verified, establishes the encrypted connection.

On the server's side, there is a component called an SSH daemon that is constantly listening to a specific TCP/IP port for possible client connection requests. Once a client initiates a connection, the SSH daemon will respond with the software and the protocol versions it supports and the two will exchange their identification data. If the provided credentials are correct, SSH creates a new session for the appropriate environment.

The default SSH protocol version for SSH server and SSH client communication is version 2.

How to Enable an SSH Connection

Since creating an SSH connection requires both a client and a server component, you need to make sure they are installed on the local and the remote machine, respectively. An open source SSH tool—widely used for Linux distributions— is OpenSSH. Installing OpenSSH is relatively easy. It requires access to the terminal on the server and the computer that you use for connecting. Note that Ubuntu does not have SSH server installed by default.

How to Install an OpenSSH Client

Before you proceed with installing an SSH client, make sure it is not already installed. Many Linux distributions already have an SSH client. For Windows machines, you can install PuTTY or any other client of your choice to gain access to a server.

To check if the client is available on your Linux-based system, you will need to:

1. Load an SSH terminal. You can either search for "terminal" or press **CTRL** + **ALT** + **T** on your keyboard.
2. Type in `ssh` and press **Enter** in the terminal.
3. If the client is installed, you will receive a response that looks like this:

```
username@host:~$ ssh

usage: ssh [-1246AaCfGgKkMNnqsTtVvXxYy] [-b bind_address] [-c ci
pher_spec]
[-D [bind_address:]port] [-E log_file] [-e escape_char]
[-F configfile] [-I pkcs11] [-i identity_file]
[-J [user@]host[:port]] [-L address] [-l login_name] [-m mac_spe
c] [-O ctl_cmd] [-o option] [-p port] [-Q query_option] [-R addr
ess] [-S ctl_path] [-W host:port] [-w local_tun[:remote_tun]]
[user@]hostname [command]

username@host:~$
```

This means that you are ready to remotely connect to a physical or virtual machine. Otherwise, you will have to install the OpenSSH client:

1. Run the following command to install the OpenSSH client on your computer: `sudo apt-get install openssh-client`
2. Type in your superuser password when asked.
3. Hit Enter to complete the installation.

You are now able to SSH into any machine with the server-side application on it, provided that you have the necessary privileges to gain access, as well as the hostname or IP address.

How to Install an OpenSSH Server

In order to accept SSH connections, a machine needs to have the server-side part of the SSH software toolkit.

If you first want to check if OpenSSH server is available on the Ubuntu system of the remote computer that needs to accept SSH connections, you can try to connect to the local host:

1. Open the terminal on the server machine. You can either search for "terminal" or press **CTRL + ALT + T** on your keyboard.
2. Type in `ssh localhost` and hit enter.
3. For the systems **without** the SSH server installed the response will look similar to this:

```
username@host:~$ ssh localhost
ssh: connect to host localhost port 22: Connection refused usern
ame@host:~$
```

If the above is the case, you will need to install the OpenSSH server. Leave the terminal open and:

1. Run the following command to install the SSH server:

```
sudo apt-get install openssh-server ii.
```

2. Type in your **superuser password** when asked.
3. **Enter** and **Y** to allow the installation to continue after the disk space prompt.

The required support files will be installed, and then you can check if the SSH server is running on the machine by typing this command:

```
sudo service ssh status
```

The response in the terminal should look similar to this if the SSH service is now running properly:

```
username@host:-$ sudo service ssh status
• ssh.service - OpenBSD Secure Shell server
Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor
preset: enab
Active: active (running) since Fr 2018-03-12 10:53:44 CET; 1min
22s ago Process: 1174 ExecReload=/bin/kill -HUP $MAINPID (code=e
xited, status=0/SUCCES

Main PID: 3165 (sshd)
```

Another way to test if the OpenSSH server is installed properly and will accept connections is to try running the **`ssh localhost`** command again in your terminal prompt. The response will look similar to this screen when you run the command for the first time:

```
username@host:~$ ssh localhost

The authenticity of host 'localhost (127.0.0.1)' can't be establ
ished. ECDSA key fingerprint is SHA256:9jqmhko9Yo1EQAS1QeNy9xKce
HFG5F8W6kp7EX9U3Rs. Are you sure you want to continue connecting
(yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of kn
own hosts.

username@host:~$
```

Enter **yes** or **y** to continue.

Congratulations! You have set up your server to accept SSH connection requests from a different computer using an SSH client.

TIP

You can now edit the SSH daemon configuration file, for example, you can change the default port for SSH connections. In the terminal prompt, run this command:

```
sudo nano /etc/ssh/sshd_config
```

The configuration file will open in the editor of your choice. In this case, we used Nano.

If you need to install Nano, run this command:

```
sudo apt-get install nano
```

Please note that you need to restart SSH service every time you make any changes to the `sshd_config` file by running this command:

```
sudo service ssh restart
```

How to Connect via SSH

Now that you have the OpenSSH client and server installed on every machine you need, you can establish a secure remote connection with your servers. To do so:

1. Open the SSH terminal on your machine and run the following command: `ssh your_username@host_ip_address`

   If the username on your local machine matches the one on the server you are trying to connect to, you can just type: `ssh host_ip_address` And hit **Enter**.
2. Type in your password and hit **Enter**. Note that you will not get any feedback on the screen while typing. If you are pasting your password, make sure it is stored safely and not in a text file.
3. When you are connecting to a server for the very first time, it will ask you if you want to continue connecting. Just type yes and hit **Enter**. This message appears only this time since the remote server is not identified on your local machine.
4. An ECDSA key fingerprint is now added and you are connected to the remote server.

If the computer you are trying to remotely connect to is on the same network, then it is best to use the private IP address instead of the public IP address. Otherwise, you will have to use the public IP address only. Additionally, make sure that you know the correct TCP port OpenSSH is listening to for connection requests and that the port forwarding settings are correct. The default port is 22 if nobody changed configuration in the sshd_config file. You may also just append the port number after the host IP address.

Here is the example of a connection request using the OpenSSH client. We will specify the port number as well:

```
username@machine:~$ ssh phoenixnap@185.52.53.222 –p7654 phoenixn
ap@185.52.53.222’s password:

The authenticity of host '185.52.53.222 (185.52.53.222)' can't b
e established. ECDSA key fingerprint is SHA256:9lyrpzo5Yo1EQAS2Q
eHy9xKceHFH8F8W6kp7EX2O3Ps. Are you sure you want to continue co
nnecting (yes/no)? yes
Warning: Permanently added ' 185.52.53.222' (ECDSA) to the list
of known hosts.

username@host:~$
```

You are now able to manage and control a remote machine using your terminal. If you have trouble connecting to a remote server, make sure that:

- The IP address of the remote machine is correct.
- The port SSH daemon is listening to is not blocked by a firewall or forwarded incorrectly.

- Your username and password are correct.
- The SSH software is installed properly.

For Windows – installation

Install openssh client

Download open ssh from https://github.com/PowerShell/Win32-OpenSSH/releases for windows.

Extract that folder and copy it to c:\

Now right click on ssh.exe and copy location and click on edit system variables

Now edit path variable copy that path and put semicolon at end.

Now go to ssh folder, press shift and right click and open command prompt.

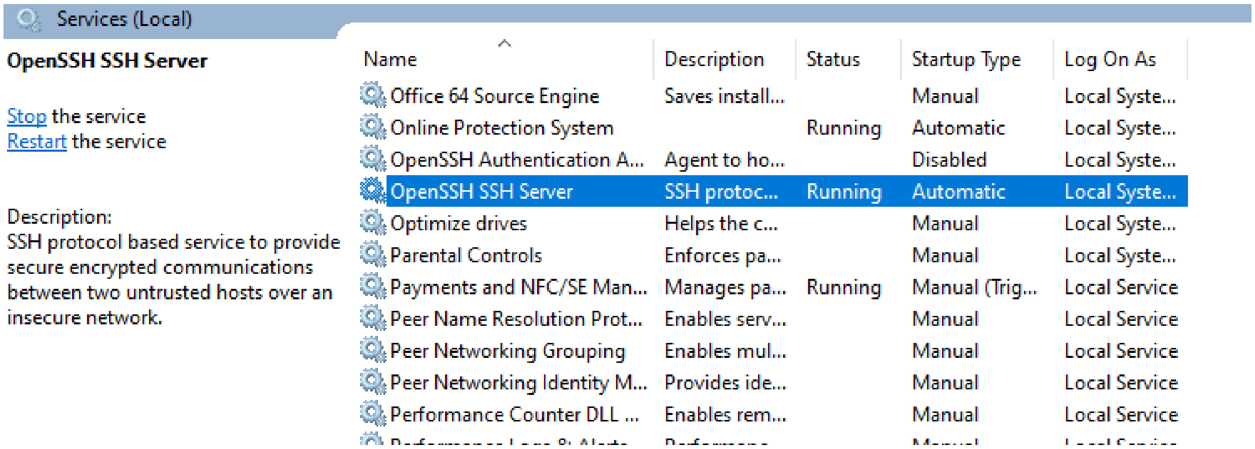
Now type ssh then it will run.

Install openssh server

Type optional features in search bar

Then install openssh server.

Type services in run and then select openssh server and right click on it and select properties.



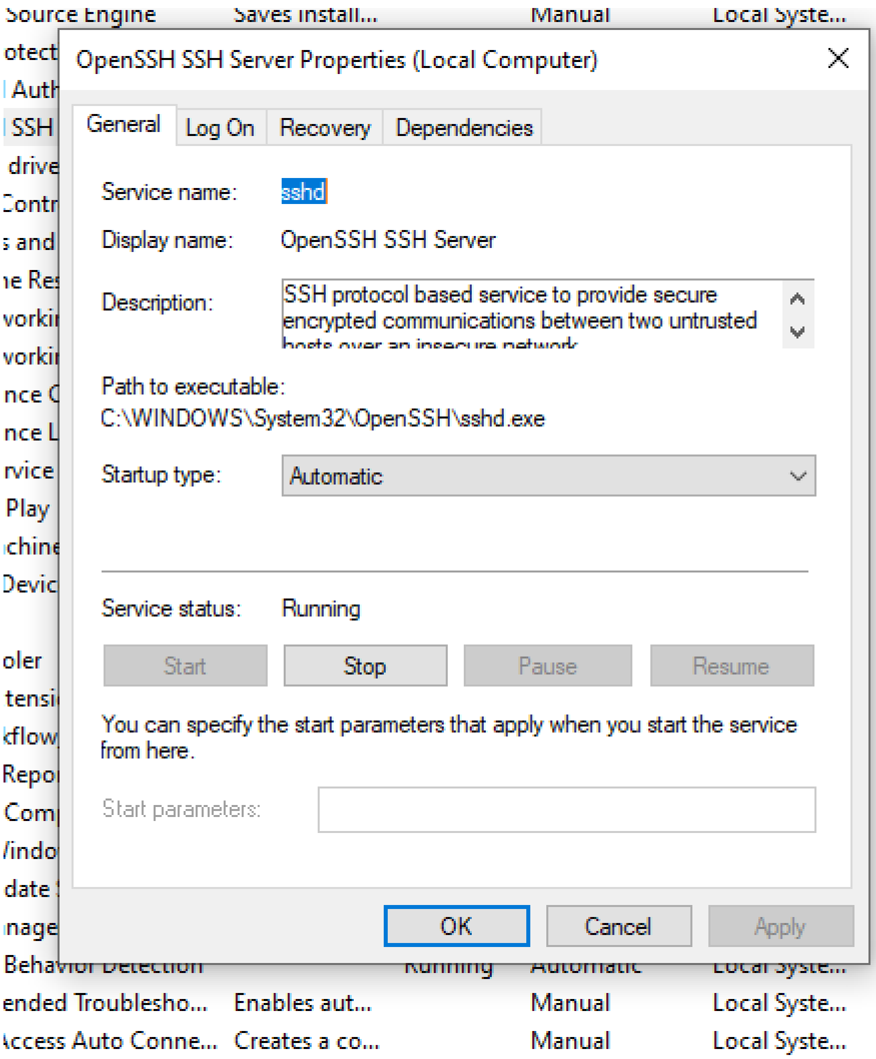
Services (Local)

OpenSSH SSH Server

Stop the service
Restart the service

Description:
SSH protocol based service to provide secure encrypted communications between two untrusted hosts over an insecure network.

Name	Description	Status	Startup Type	Log On As
Office 64 Source Engine	Saves install...		Manual	Local Syste...
Online Protection System		Running	Automatic	Local Syste...
OpenSSH Authentication A...	Agent to ho...		Disabled	Local Syste...
OpenSSH SSH Server	SSH protoc...	Running	Automatic	Local Syste...
Optimize drives	Helps the c...		Manual	Local Syste...
Parental Controls	Enforces pa...		Manual	Local Syste...
Payments and NFC/SE Man...	Manages pa...	Running	Manual (Trig...	Local Service
Peer Name Resolution Prot...	Enables serv...		Manual	Local Service
Peer Networking Grouping	Enables mul...		Manual	Local Service
Peer Networking Identity M...	Provides ide...		Manual	Local Service
Performance Counter DLL ...	Enables rem...		Manual	Local Service



Service name: sshd

Display name: OpenSSH SSH Server

Description: SSH protocol based service to provide secure encrypted communications between two untrusted hosts over an insecure network

Path to executable:
C:\WINDOWS\System32\OpenSSH\sshd.exe

Startup type: Automatic

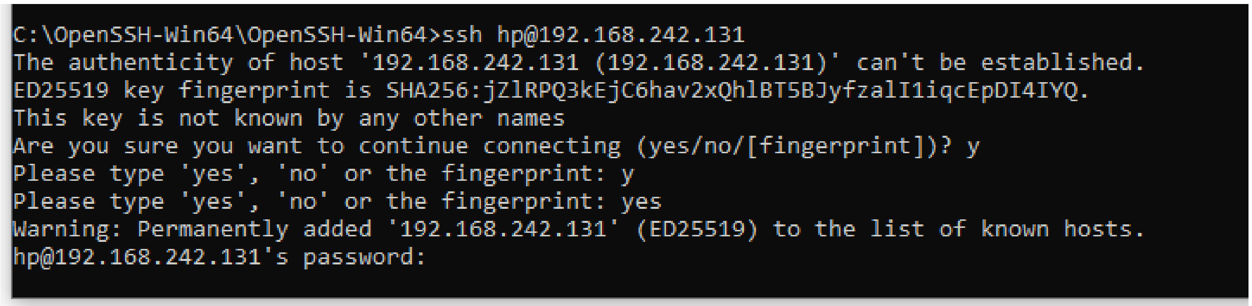
Service status: Running

You can specify the start parameters that apply when you start the service from here.

Start parameters:

Select status type to automatic and click on start then click on apply.

Now open command prompt and run following command. Check also if firewall settings allowed ssh request or not



```
C:\OpenSSH-Win64\OpenSSH-Win64>ssh hp@192.168.242.131
The authenticity of host '192.168.242.131 (192.168.242.131)' can't be established.
ED25519 key fingerprint is SHA256:jZlRPQ3kEjC6hav2xQhlBT5BJyfzalI1iqcEpDI4IYQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.242.131' (ED25519) to the list of known hosts.
hp@192.168.242.131's password:
```

Introduction

SSH is a network tool used for remote, command-line login to systems that have the server enabled. It has sibling applications named SFTP and SCP that can be used to copy files.

Windows 10 systems with build 1803 or newer and Windows Server 2019 come with an implementation of OpenSSH that's enabled by default. Some older versions may have this as an optional component that needs to be installed before it can be used.

Checking for Installation

The easiest way to determine if OpenSSH is installed is to open a command window (go to the Start Menu, look under "Windows System", and pick "Command Prompt"). Type "ssh" and you should see something like this:

```
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface]

        [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]

        [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]

        [-i identity_file] [-J [user@]host[:port]] [-L address]

        [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]

        [-Q query_option] [-R address] [-S ctl_path] [-W host:port]

        [-w local_tun[:remote_tun]] destination [command]
```

If it returns:

```
'ssh' is not recognized as an internal or external command,

operable program or batch file.
```

Then it may need to be installed (see if the folder "C:\Windows\System32\OpenSSH" exists and has files such as "ssh.exe" in it). If it is installed, then that directory needs to be added to the search path (type "echo %PATH%" in the command prompt window to see if it's in there). If not, changing the system path requires administrative privileges so contact SENS for assistance.

Installation

Click on the "Settings" gear in the left pane of the Start Menu.

Click on "Apps".

Click on "Optional features".

Look in the list. If you do not see "OpenSSH Client", click "Add a feature".

Select "OpenSSH Client" and click "Install".

Usage

Once it's installed, using it is as simple as typing something such as:

> ssh user@linux.sens.buffalo.edu

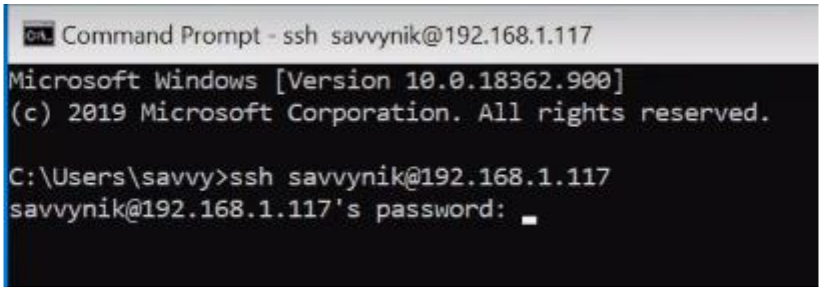
To copy files using Secure FTP:

> sftp *.doc user@linux.sens.buffalo.edu

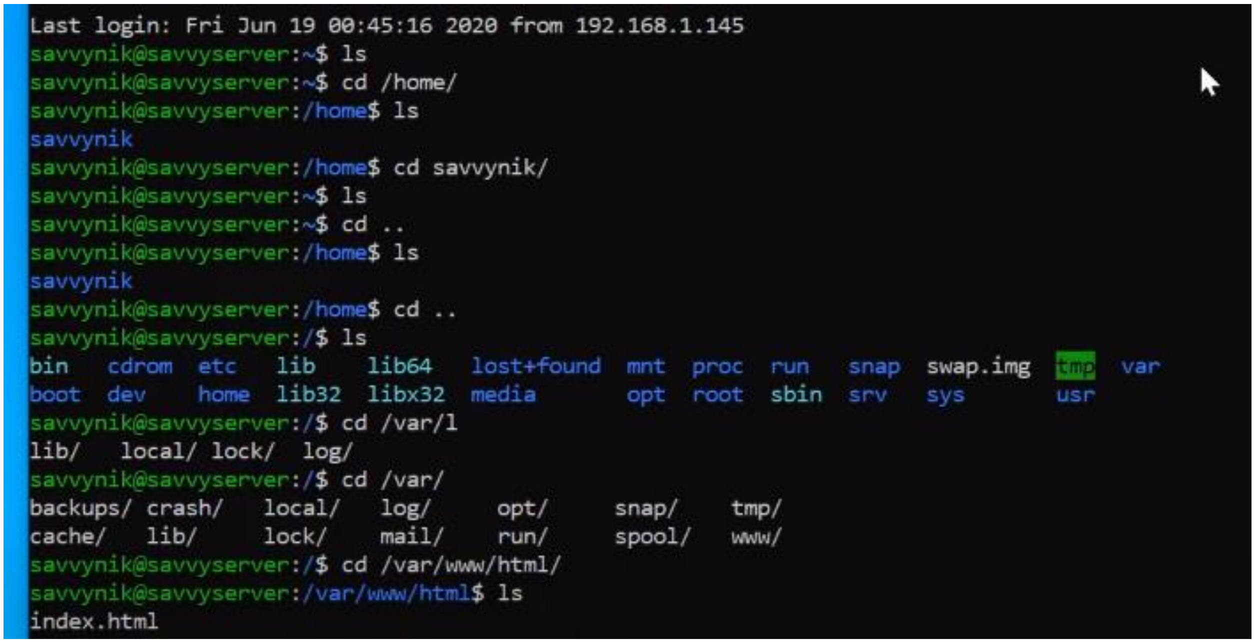
To copy files using Secure Copy, preserving original timestamps:

> scp -p *.doc user@linux.sens.buffalo.edu

Example



```
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\savvy>ssh savvynik@192.168.1.117
savvynik@192.168.1.117's password:
```



```
Last login: Fri Jun 19 00:45:16 2020 from 192.168.1.145
savvynik@savvyserver:~$ ls
savvynik@savvyserver:~$ cd /home/
savvynik@savvyserver:/home$ ls
savvynik
savvynik@savvyserver:/home$ cd savvynik/
savvynik@savvyserver:~$ ls
savvynik@savvyserver:~$ cd ..
savvynik@savvyserver:/home$ ls
savvynik
savvynik@savvyserver:/home$ cd ..
savvynik@savvyserver:/$ ls
bin   cdrom  etc   lib    lib64   lost+found  mnt  proc  run   snap  swap.img  tmp  var
boot  dev    home  lib32  libx32  media       opt  root  sbin  srv   sys       usr
savvynik@savvyserver:/$ cd /var/l
lib/   local/ lock/  log/
savvynik@savvyserver:/$ cd /var/
backups/ crash/   local/   log/     opt/     snap/    tmp/
cache/   lib/     lock/    mail/    run/     spool/   www/
savvynik@savvyserver:/$ cd /var/www/html/
savvynik@savvyserver:/var/www/html$ ls
index.html
```