

Windows IP Commands – ipconfig-nslookup-ping-tracert etc

As Network Engineers we need to be versatile and troubleshooting-savvy in our work environment.

In addition to having strong knowledge of networking protocols and commands on network devices (routers, switches, firewalls etc) we need also to have very good knowledge of IP and other networking related commands on end-point devices such as Windows computers, Linux servers and workstations etc.

In this article we'll list and describe the most useful and helpful IP Commands on Windows operating system. Most of these commands (with some exceptions and variations) are also available on Linux OS.

I have found myself thousands of times to start troubleshooting network and connectivity problems from an end-point device first (computer, server etc) before moving on to the actual core network devices for further investigation.

Having knowledge of the following IP commands will add a strong array of resources in your troubleshooting arsenal.

Windows IP Commands

Let's now examine the Windows CMD commands (from the DOS prompt) that are related to networking etc:

ipconfig command

This is one of the most useful IP commands on Windows. It displays tons of useful information about the current network settings on the machine such as IPv4 and IPv6 address of all network interface cards (Ethernet adapters, WiFi adapters, virtual network adapters etc), MAC address, default gateway, subnet mask, DNS server, DHCP information etc.

If you want to find the local IP address assigned to your computer or the MAC address of your Ethernet Adapter (shown as “Physical Address” in the command output as shown in the picture below), this is the quickest way to find this information.

Here is a screenshot example of what you can expect as output from ipconfig:

Wireless LAN adapter Wireless Network Connection:

```
Connection-specific DNS Suffix . : 
Description . . . . . : Intel(R) Centrino(R) Advanced-N 6200 AGN
Physical Address . . . . . : 00-23-14-61-D0-58
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::6088:303d:1883:f148%3(Preferred)
IPv4 Address. . . . . : 10.153.34.157(Preferred)
Subnet Mask . . . . . : 255.255.0.0
Lease Obtained. . . . . : Friday, July 20, 2018 7:13:40 AM
Lease Expires . . . . . : Friday, July 20, 2018 12:43:40 PM
Default Gateway . . . . . : 10.153.0.1
DHCP Server . . . . . : 1.1.1.1
DHCPv6 IAID . . . . . : 352330516
DHCPv6 Client DUID. . . . . : 00-01-00-01-13-F4-45-87-A4-BA-DB-BE-0D-C5
DNS Servers . . . . . : 195.1.1.1
                        195.1.1.1
NetBIOS over Tcpip. . . . . : Enabled
```

ipconfig /? : Displays all available options.

ipconfig /all : This will display output as shown on the screenshot above but for ALL network connection adapters of the computer (Wired Ethernet, WiFi, VMware adapters etc).

ipconfig /release : This will release the current IPv4 addresses which were assigned dynamically from a DHCP server. If you specify also a connection name at the end, it will release only the IP of that connection adapter.

ipconfig /release6 : Same as above but for the IPv6 address.

ipconfig /renew : This usually comes after the above command and is used to request a new IP address from a DHCP server.

ipconfig /renew6 : Same as above but for the IPv6 address.

ipconfig /flushdns : This deletes the local DNS resolver cache of the computer. This cache stores DNS entries of frequently accessed internet resources so that the computer will not query an external DNS server every time you try to access an internet resource (website etc). This command is useful when troubleshooting DNS connection problems.

ipconfig /displaydns : It shows the local DNS resolver cache entries as explained above.

ipconfig /registerdns : Refreshes all DHCP addresses and also communicates again with the external DNS server to make sure its reachable etc. Very useful when troubleshooting DNS and network connectivity problems of the local computer.

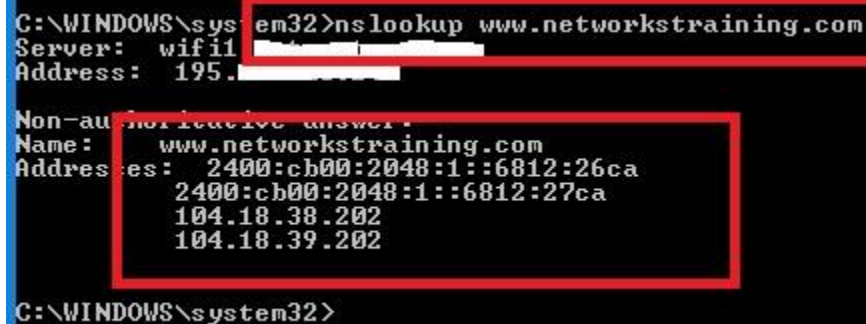
nslookup command

“nslookup” stands for “Name System Lookup” and is very useful in obtaining Domain Name System (DNS) related information about a domain or about an IP address (reverse DNS lookup).

nslookup [domain name]: The most popular usage of this command is to find quickly the IP address of a specific domain name (A-record) as shown below:

Example:

nslookup www.networkstraining.com



```
C:\WINDOWS\system32>nslookup www.networkstraining.com
Server: wifil
Address: 195.166.11.11

Non-authoritative answer:
Name: www.networkstraining.com
Addresses: 2400:cb00:2048:1::6812:26ca
           2400:cb00:2048:1::6812:27ca
           104.18.38.202
           104.18.39.202

C:\WINDOWS\system32>
```

As shown above, the “nslookup” command followed by a domain name will show you the IPv4 and IPv6 addresses (A records and AAAA records) assigned to the specific domain.

nslookup [IP Address]: This will perform a reverse-DNS lookup and will try to match the given IP address in the command with its corresponding domain name.

Example:

nslookup 8.8.8.8



```
C:\WINDOWS\system32>nslookup 8.8.8.8
Server: wifil
Address: 195.166.11.11

Non-authoritative answer:
Name: google-public-dns-a.google.com
Address: 8.8.8.8

C:\WINDOWS\system32>
```

As shown on the screenshot above, the IP address 8.8.8.8 is mapped with the name “**google-public-dns-a.google.com**”. You should note however that not all IP addresses are assigned to a domain name so a lot of times you will not get any information from the command above.

There are several other interesting features of the nslookup command such as finding the authoritative DNS servers of a domain, the SOA and MX records of a domain and much more.

ping command

Now let's examine one of the most popular utilities related to network connectivity.

Probably the first command that every computer user runs on the command line when having connectivity problems is the “**ping**” command.

This will quickly show you if can send and receive packets (**icmp** packets to be exact) from your computer and hence shows whether you have network connectivity or not.

Note also that “ping” is useful for testing connectivity for both the local computer from where you execute the command and also for a remote computer or server which you try to reach.

If for example you try to “ping” your local default gateway IP address and you get replies back (icmp echo replies), this means your local computer is properly connected to the network.

Now, if you “ping” a remote server on the Internet and you get replies back, it means that the remote server is properly connected to its network as well.

ping /? : Displays all available options as shown below:

```
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
```

ping [IP Address] : By default it will send 4 ICMP packets to the stated IP address.

Example:

ping 8.8.8.8

```
C:\WINDOWS\system32>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=92ms TTL=122
Reply from 8.8.8.8: bytes=32 time=80ms TTL=122
Reply from 8.8.8.8: bytes=32 time=52ms TTL=122
Reply from 8.8.8.8: bytes=32 time=53ms TTL=122

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 52ms, Maximum = 92ms, Average = 69ms
```

As you can see from the screenshot above, pinging the IP 8.8.8.8 results in sending 4 packets and then receiving back 4 packets from that IP.

ping [hostname or domain] : When “pinging” a hostname or domain name, the command will resolve first the name to IP address and then send the icmp packets to that IP.

Example:

ping www.networkstraining.com

```
C:\WINDOWS\system32>ping www.networkstraining.com

Pinging www.networkstraining.com [104.18.38.202] with 32 bytes of data:
Reply from 104.18.38.202: bytes=32 time=42ms TTL=59
Reply from 104.18.38.202: bytes=32 time=43ms TTL=59
Reply from 104.18.38.202: bytes=32 time=42ms TTL=59
Reply from 104.18.38.202: bytes=32 time=42ms TTL=59

Ping statistics for 104.18.38.202:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 42ms, Maximum = 43ms, Average = 42ms
```

ping [IP address] -t : This will send ping packets (icmp echo requests) continuously to the target IP.

ping -n 10 [IP address] : This will send 10 ping packets (icmp echo requests) to the target IP.

ping -l 1500 [IP address] : This will send ping packets (icmp echo requests) with size of 1500 bytes length to the target IP.

ping -a [IP address] : The -a switch tells the computer to try to find the hostname assigned to the specific IP address and then ping the IP.

ping -6 [domain or IP] :The -6 switch tells the computer to send IPv6 packets to the target.

tracert command

“**tracert**” in Windows stands for “Trace Route”. In Linux, the same command is “traceroute”.

The command traces the path that a TCP/IP packet takes towards a destination target and shows some information (if available) of the routing nodes within this path.

Just like the “ping” command, “tracert” sends also ICMP echo packets to the destination with varying Time-to-Live (TTL) values.

tracert [domain or IP] : Traces the TCP/IP path to the specified destination target IP or domain.

Example:

tracert www.networkstraining.com

```
C:\WINDOWS\system32>tracert www.networkstraining.com

Tracing route to www.networkstraining.com [104.18.38.202]
over a maximum of 30 hops:

  0  1 ms    2 ms    1 ms   192.168.10.254 [192.168.10.254]
  1  24 ms   23 ms   23 ms   192.168.10.254 [192.168.10.254]
  2  25 ms   24 ms   24 ms   g1-1-0-100-100-100-100 [195.168.100.100]
  3  27 ms   25 ms   28 ms   te0-0-0-3-100-100-100-100 [195.168.100.100]
  4  27 ms   29 ms   27 ms   te0-4-0-6-100-100-100-100 [195.168.100.100]
  5  42 ms   43 ms   42 ms   cloudflare-[195.168.100.100]
  6  43 ms   41 ms   43 ms   104.18.38.202

Trace complete.
```

As shown above, tracing the path to domain **www.networkstraining.com** shows all the intermediary routing nodes (with their hostname and IP address) until the final target destination.

When troubleshooting connection problems in a large network, you can use tracert to see where the packets stop before reaching the target and focus your efforts to find the problem on the node which does not route packets.

netstat command

Another important command is the Network Statistics (“netstat”) utility found in both Windows and Linux OS.

It shows the established network TCP/IP connections of the local computer with remote hosts, open ports on the machine, the process ID (PID) of each connection etc.

Personally I use this command mostly for security forensic purposes to identify if there are backdoors running on the computer, malicious connections to external Command-and-Control servers etc.

Here are some popular usages of this command:

netstat -ano : Displays all connections and listening ports (-a), addresses and ports in numerical form (-n) and also the process ID of each connection (-o).

netstat -vb : Very useful to examine also which executable and which sequence created each connection and each port.

Example:

C:\WINDOWS\system32>**netstat -vb**

```
C:\WINDOWS\system32>netstat -vb

Active Connections

 Proto Local Address           Foreign Address         State
 TCP    192.168.10.5:9492       lg-in-f95:https        CLOSE_WAIT
 [googledrivesync.exe]
 TCP    192.168.10.5:9494       lg-in-f95:https        CLOSE_WAIT
 [googledrivesync.exe]
 TCP    192.168.10.5:9495       lg-in-f95:https        CLOSE_WAIT
 [googledrivesync.exe]
 TCP    192.168.10.5:10853      13.92.210.83:https      ESTABLISHED
 WpnService
 [svchost.exe]
 TCP    192.168.10.5:10854      13.92.210.83:https      ESTABLISHED
 WpnService
 [svchost.exe]
 TCP    192.168.10.5:10886      151.101.122.2:https     ESTABLISHED
 [chrome.exe]
 TCP    192.168.10.5:10892      lq-in-f125:5222        ESTABLISHED
 [googledrivesync.exe]
 TCP    192.168.10.5:10894      r-252-58-45-5:https    CLOSE_WAIT
 Can not obtain ownership information
 TCP    192.168.10.5:10898      ams10-012:http         ESTABLISHED
 Can not obtain ownership information
 TCP    192.168.10.5:10952      104.244.42.136:https    ESTABLISHED
 [chrome.exe]
```

As shown above, for each established connection you can see the executable (e.g **chrome.exe**, **googledrivesync.exe** etc) that created the connection.

netstat -p tcp -f : The “-p tcp” switch will show only TCP connections and the “-f” switch will show the FQDN name of each connection instead of just IP address.

Example:

C:\WINDOWS\system32>netstat -p tcp -f

```
C:\WINDOWS\system32>netstat -p tcp -f

Active Connections

 Proto Local Address           Foreign Address         State
 TCP   192.168.10.5:9492        lg-in-f95.1e100.net:https CLOSE_WAIT
 TCP   192.168.10.5:9494        lg-in-f95.1e100.net:https CLOSE_WAIT
 TCP   192.168.10.5:10853       13.92.210.83:https      ESTABLISHED
 TCP   192.168.10.5:10854       13.92.210.83:https      ESTABLISHED
 TCP   192.168.10.5:10892       lq-in-f125.1e100.net:5222 ESTABLISHED
 TCP   192.168.10.5:10894       r-252-58-45-5.ff.avast.com:https CLOSE_WAIT
 TCP   192.168.10.5:10898       ams10-012.ff.avast.com:http ESTABLISHED
 TCP   192.168.10.5:11358       ams15s32-in-f10.1e100.net:https CLOSE_WAIT
 TCP   192.168.10.5:11360       a23-57-87-114.deploy.static.akamaitechnologies.com:https ESTABLISHED
 TCP   192.168.10.5:11368       a23-57-87-114.deploy.static.akamaitechnologies.com:https ESTABLISHED
 TCP   192.168.10.5:11375       ams15s33-in-f14.1e100.net:https ESTABLISHED
 TCP   192.168.10.5:11379       ams16s30-in-f66.1e100.net:https ESTABLISHED
 TCP   192.168.10.5:11389       ams16s29-in-f3.1e100.net:https ESTABLISHED
 TCP   192.168.10.5:11390       a23-57-85-103.deploy.static.akamaitechnologies.com:http ESTABLISHED
 TCP   192.168.10.5:11391       14-13.li.cytanet.com.cy:http TIME_WAIT
```

route command

The “route” command is used to manipulate the local routing table of the computer. You can print the current routing table, add new static routes, delete entries etc.

Personally, the way I use the “route” command is to add a permanent static route entry in a computer. For example, there might be a specific network subnet which is not accessible via the default gateway of the computer. Instead, this remote subnet might be accessible via a different gateway IP. By adding a static route in the computer’s routing table you will be able to reach that remote subnet from a different gateway.

route PRINT : Displays the current routing table of the computer

route ADD [Destination network] MASK [mask] [gatewayIP]: This adds a static route in the table.

Example:

route ADD 10.10.10.0 MASK 255.255.255.0 192.168.1.2

The above command will add a static route for destination subnet 10.10.10.0/24 via gateway 192.168.1.2

arp command

ARP stands for “Address Resolution Protocol” and is one of the core networking protocols that work in Layer 2 level and facilitate communication in a LAN.

The job of ARP is to find the physical address (MAC address) of the target and map it with its corresponding Layer 3 IP address when communicating in a LAN. The ARP cache table stores mappings of IP addresses with their corresponding MAC address.

arp -a : Displays all ARP cache mappings (IP to MAC address)

Example:

C:\WINDOWS\system32>arp -a

```
C:\WINDOWS\system32>arp -a

Interface: 192.168.10.5 --- 0xb
Internet Address      Physical Address      Type
192.168.10.7          d8-d4-3c-f2-aa-79     dynamic
192.168.10.254        cc-7b-35-2a-35-ae     dynamic
224.0.0.2             01-00-5e-00-00-02     static
224.0.0.22            01-00-5e-00-00-16     static
224.0.0.251           01-00-5e-00-00-fb     static
224.0.0.252           01-00-5e-00-00-fc     static
225.0.11.47           01-00-5e-00-0b-2f     static
239.255.255.250       01-00-5e-7f-ff-fa     static
255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

As you can see from above, the local computer has learned dynamically (type=dynamic) using the ARP protocol two other local devices (192.168.10.7 and 192.168.10.254) and has stored their MAC address (Physical Address) in the ARP table.

arp -d [IP address] : This will delete the arp entry for the specified IP address.

The above is useful when you changed hardware on a specific node (e.g you have changed the default gateway router) and you want to remove old arp entries. Usually it's not needed to do anything in such a case but sometimes its required on some older computers.