WINDOWS NETWORK ESSENTIALS & TROUBLESHOOTING

Pavlovsky Anton

System engineer at EPAM
Certificates: CCNA Enterprise, CCNP Enterprise, CCNP Service Provider

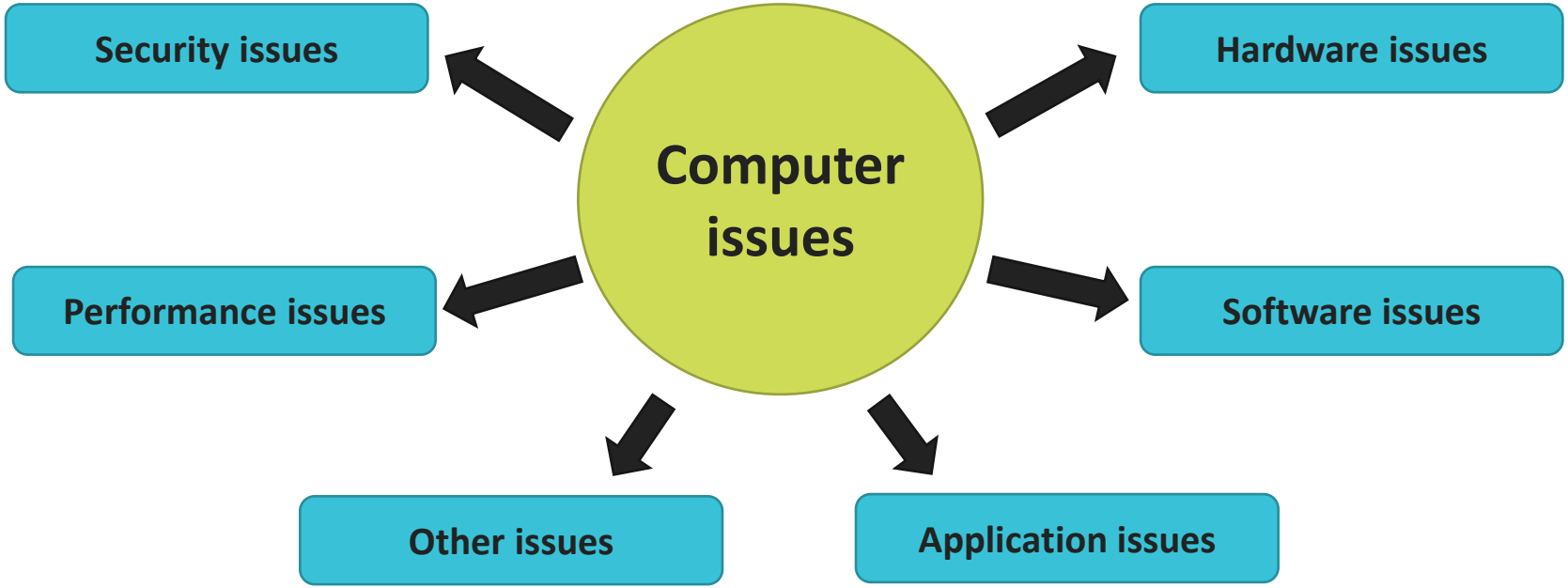
TRAINING CENTER

Agenda

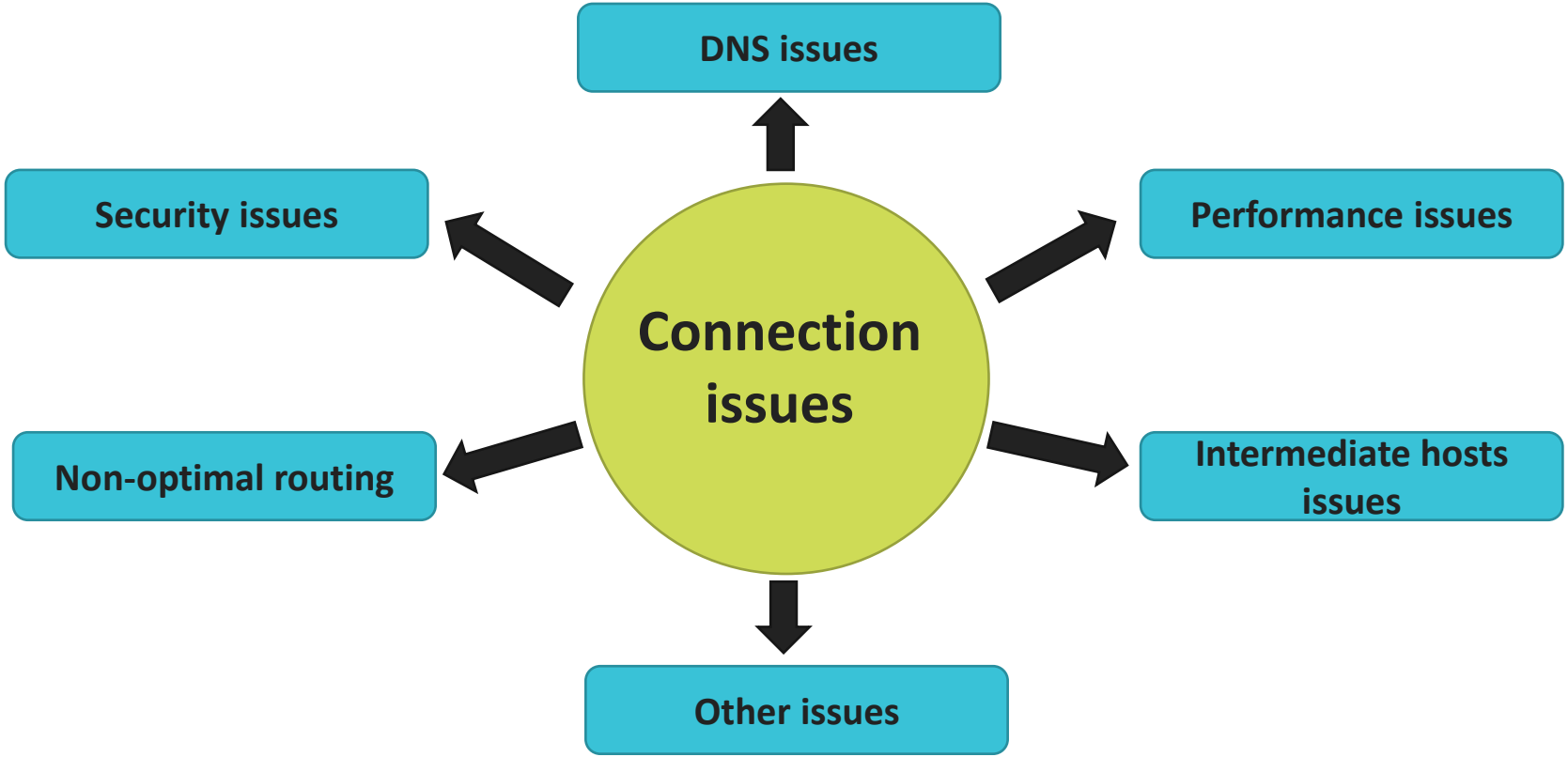
- Windows CLI
- Common computer and network connectivity issues
- Troubleshooting tools (Debug common problems)
- Examples

COMMON COMPUTER AND NETWORK CONNECTIVITY ISSUES

Computer local issues



Network connectivity issues



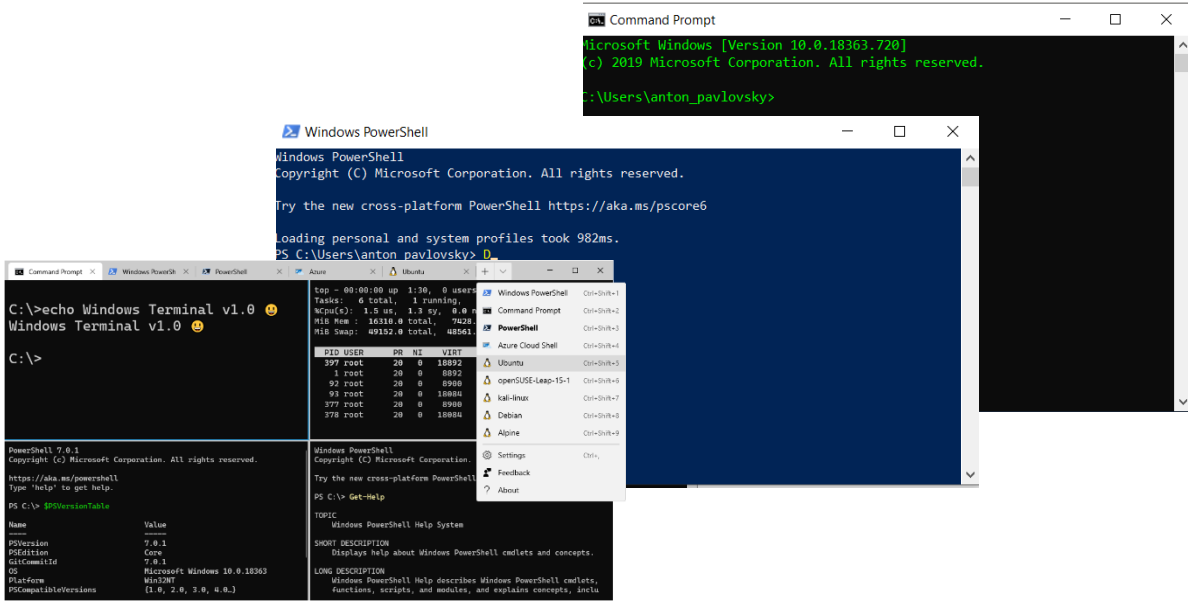
WINDOWS CLI

Windows Command Line Interface (CLI)

A **command-line interface** (**CLI**) processes commands to a computer program in the form of lines of text. The program which handles the interface is called a **command-line interpreter** or **command-line processor**.

Operating systems implement a command-line interface in a shell for interactive access to operating system functions or services.

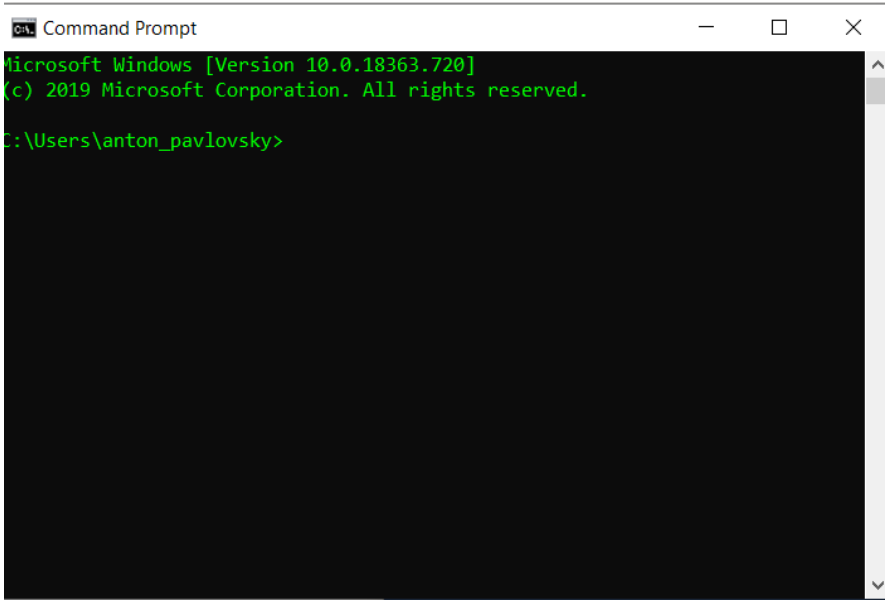
Format:
name_of_command parameter_1 [parameter_2 […]



Windows Command Prompt (CMD)

cmd.exe is the default command-line interpreter for Microsoft Windows operating systems.

cmd.exe interacts with the user through a command-line interface. On Windows, this interface is implemented through the Win32 console.



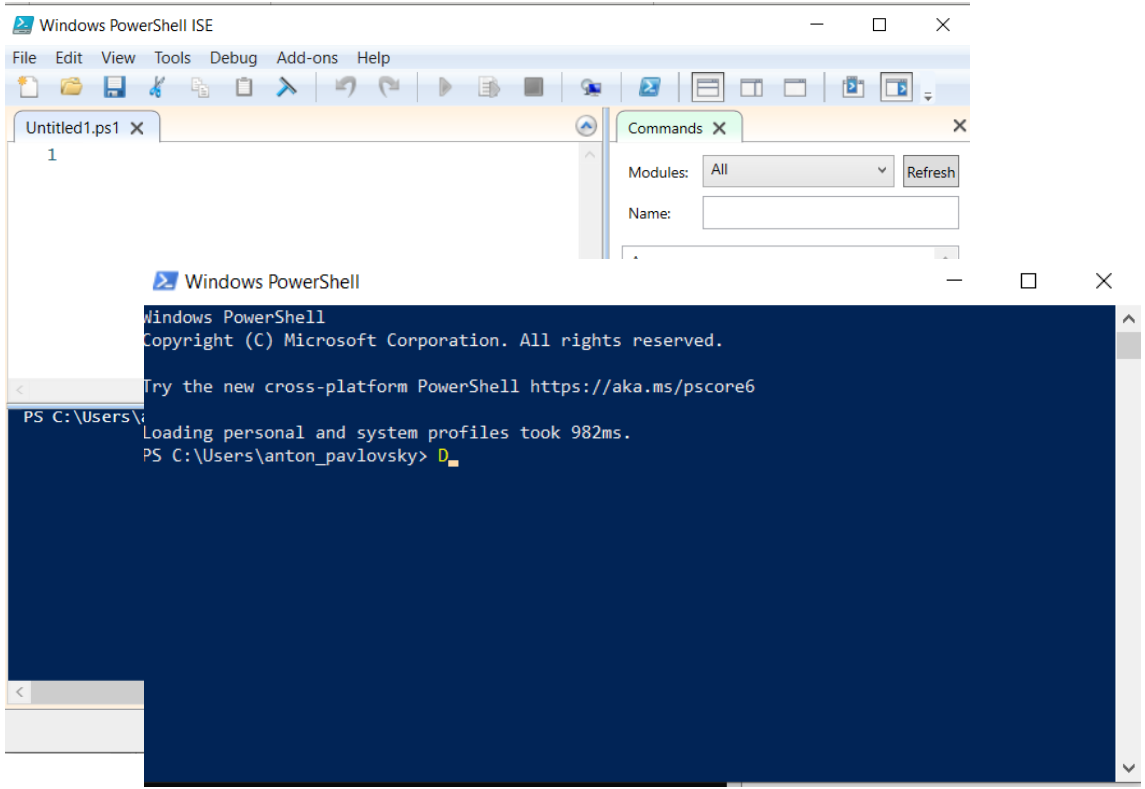
shutdown -r -f -t 30 -c - «мягкая» перезагрузка компьютера через 30 сек
xcopy "C:\folder1" "D:\folder2" /e - копирование содержимого из одной папки в другую
control userpasswords - вызов окна учетных записей пользователей
ping -t "8.8.8.8" - запуск утилиты ping в непрерывном режиме
ipconfig /all - просмотр полных сведений о сетевом интерфейсе

Windows PowerShell

PowerShell is a cross-platform task automation and configuration management framework, consisting of a command-line shell and scripting language.

PowerShell uses cmdlets, which are self-contained programming objects that expose the underlying administration options inside of Windows.

The Windows PowerShell Integrated Scripting Environment (ISE) is a host application for Windows PowerShell. In the ISE, you can run commands and write, test, and debug scripts in a single Windows-based graphic user interface.

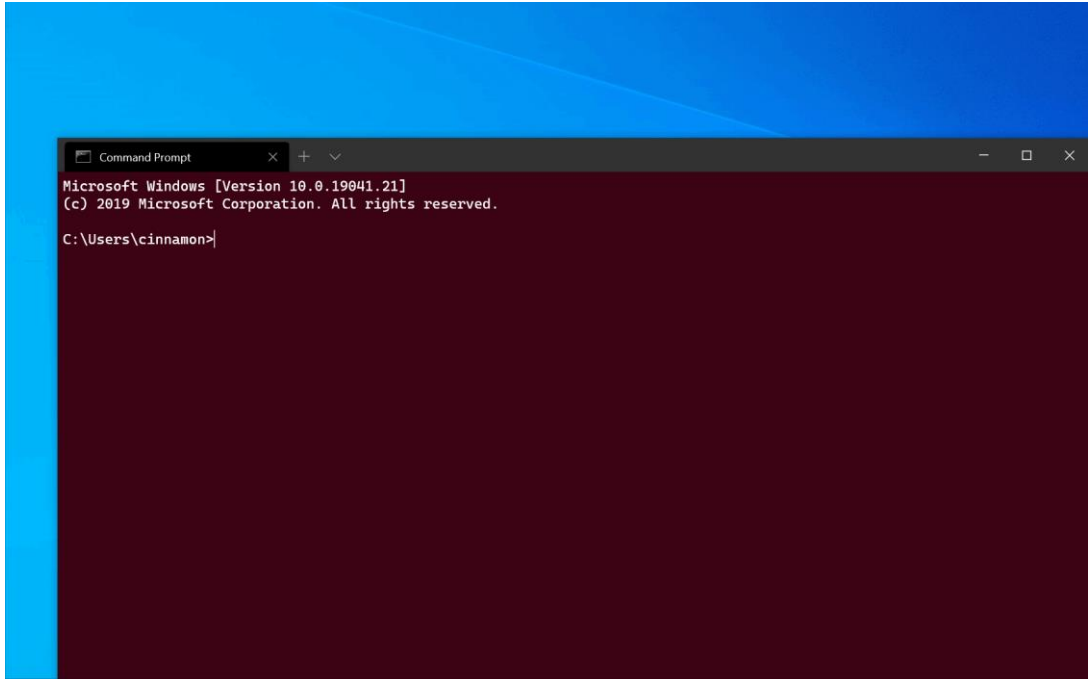


Windows Terminal

Windows Terminal (codenamed *Cascadia*) is a terminal emulator for Windows 10. It includes support for the Command Prompt, PowerShell, WSL and SSH.

The initial source code release preview release was first published to the Microsoft Store on June 21, 2019.

more documentation



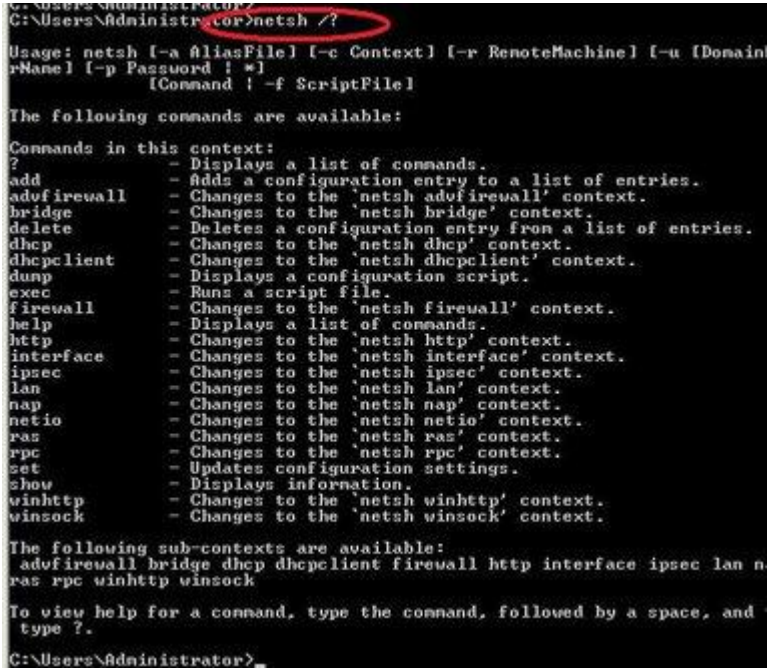
NETWORK TOOLS (DEBUG COMMON PROBLEMS)

Network shell (Netsh)

Netsh is a command-line scripting utility that allows you to, either locally or remotely, display or modify the network configuration of a computer that is currently running.

Netsh also provides a scripting feature that allows you to run a group of commands in batch mode against a specified computer.

It is recommended that you use Windows PowerShell to manage networking technologies in Windows Server 2016 and Windows 10 rather than Network Shell. Network Shell is included for compatibility with your scripts, however, and its use is supported.



```
C:\Users\Administrator>netsh /?

Usage: netsh [-a AliasFile] [-c Context] [-r RemoteMachine] [-u [Domain
rName] [-p Password | *]
             [Command | -f ScriptFile]

The following commands are available:

Commands in this context:
?               - Displays a list of commands.
add             - Adds a configuration entry to a list of entries.
advfirewall     - Changes to the 'netsh advfirewall' context.
bridge          - Changes to the 'netsh bridge' context.
delete          - Deletes a configuration entry from a list of entries.
dhcp            - Changes to the 'netsh dhcp' context.
dhcpclient      - Changes to the 'netsh dhcpclient' context.
dump            - Displays a configuration script.
exec            - Runs a script file.
firewall        - Changes to the 'netsh firewall' context.
help            - Displays a list of commands.
http            - Changes to the 'netsh http' context.
interface       - Changes to the 'netsh interface' context.
ipsec           - Changes to the 'netsh ipsec' context.
lan             - Changes to the 'netsh lan' context.
nap             - Changes to the 'netsh nap' context.
netio           - Changes to the 'netsh netio' context.
ras             - Changes to the 'netsh ras' context.
rpc             - Changes to the 'netsh rpc' context.
set             - Updates configuration settings.
show            - Displays information.
winhttp         - Changes to the 'netsh winhttp' context.
winsock         - Changes to the 'netsh winsock' context.

The following sub-contexts are available:
 advfirewall bridge dhcp dhcpclient firewall http interface ipsec lan n
ras rpc winhttp winsock

To view help for a command, type the command, followed by a space, and
 type ?.

C:\Users\Administrator>_
```

Syntax and Examples

Syntax:

netsh[**-a** *AliasFile*] [**-c** *Context*] [**-r** *RemoteComputer*] [**-u** [*DomainName*\ ] *UserName*] [**-p** *Password* | *]
[{*NetshCommand* | **-f** *ScriptFile*}]

Full syntax and parameters

Examples:

Display command help – **netsh /?**

Display LAN context help **- netsh lan /?**

Display ip interface config on remote host - **netsh -r mypc -u mypc\administrator -p My!Pass1 interface ip show config**

Run script - **netsh –f <scriptfile>**

Open a port on firewall - **netsh firewall set portopening tcp 445 smb enable**

Export interface configuration to file - **netsh –c interface dump > test.txt**

Reset all IP protocol configurations on interface and send the output to a log file - **netsh int ipv4 reset resetlog.txt**

IP to MAC (ARP)

ARP command displays and modifies entries in the Address Resolution Protocol (ARP) cache.
The ARP cache contains one or more tables that are used to store IP addresses and their resolved Ethernet physical addresses. There is a separate table for each Ethernet network adapter installed on your computer.

Command Prompt

```
C:\Users>arp -a

Interface: 192.168.100.6 --- 0x3
  Internet Address      Physical Address      Type
  192.168.100.1         44-55-b1-a9-29-77     dynamic
  192.168.100.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 192.168.56.1 --- 0x11
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

C:\Users>
```

Syntax and Examples

Syntax:

arp [/a [<inetaddr>] [/n <ifaceaddr>]] [/g [<inetaddr>] [-n <ifaceaddr>]] [/d <inetaddr> [<ifaceaddr>]]

Full syntax and parameters

Examples:

Display command help – **arp /?** or **arp**

Display the arp cache tables for all interface: **arp /a**

Display the arp cache table for the interface that is assigned the IP address *192.168.100.6*: **arp /a /n 192.168.100.6**

Add a static arp cache entry that resolves the IP address *192.168.100.6* to the physical address *00-AA-00-4F-2A-9C:*
arp /s 192.168.100.6 00-AA-00-4F-2A-9C

Netstat

Netstat displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics (for the IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for the IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols). Used without parameters, this command displays active TCP connections.

Command Prompt

```
C:\Users>
C:\Users>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    100.64.6.95:50190      52.97.189.98:https     ESTABLISHED
  TCP    100.64.6.95:54182      a104-81-60-24:https    ESTABLISHED
  TCP    100.64.6.95:54190      ec2-18-196-241-73:https  TIME_WAIT
  TCP    100.64.6.95:54192      10.6.236.94:https      ESTABLISHED
  TCP    100.64.6.95:55348      resolver1:https        TIME_WAIT
  TCP    100.64.6.95:55349      13.107.136.9:https     ESTABLISHED
  TCP    100.64.6.95:55350      13.107.136.9:https     ESTABLISHED
  TCP    100.64.6.95:55352      13.88.28.53:https      ESTABLISHED
  TCP    100.64.6.95:55353      ec2-18-196-241-73:https  ESTABLISHED
  TCP    100.64.6.95:55356      ec2-3-125-77-172:https  ESTABLISHED
  TCP    100.64.6.95:57372      40.101.12.130:https    ESTABLISHED
  TCP    100.64.6.95:57636      ec2-3-125-77-172:https  ESTABLISHED
  TCP    100.64.6.95:59155      wl-in-f188:5228        ESTABLISHED
  TCP    100.64.6.95:59612      40.101.83.194:https    TIME_WAIT
^C
C:\Users>
```

Syntax and Examples

Syntax:

netstat [-a] [-e] [-n] [-o] [-p <Protocol>] [-r] [-s] [<interval>]

Full syntax and parameters

Examples:

Display command help – **netstat /?**

Display both the Ethernet statistics and the statistics for all protocols: **netstat -e –s**

Display the statistics for only the TCP and UDP protocols: **netstat -s -p tcp udp**

Display active TCP connections and the process IDs every 5 seconds: **netstat -o 5**

display active TCP connections and the process IDs using numerical form: **netstat -n -o**

Ipconfig

Ipconfig displays all current TCP/IP network configuration values and refreshes Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) settings.

Used without parameters, **ipconfig** displays Internet Protocol version 4 (IPv4) and IPv6 addresses, subnet mask, and default gateway for all adapters.

Command Prompt

```
Ethernet adapter VirtualBox Host-Only Network:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::129:1109:782a:e7e0%17
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Local Area Connection* 1:

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

Wireless LAN adapter Local Area Connection* 2:

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

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::5dd1:b8f4:54e8:4d38%3
   IPv4 Address. . . . . . . . . . . : 192.168.100.6
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.100.1
```

Syntax and Examples

Syntax:

ipconfig [/all] [/renew [<adapter>]] [/release [<adapter>]] [/renew6[<adapter>]] [/flushdns] [/displaydns]

Full syntax and parameters

Examples:

Display command help – **ipconfig /?**

Display the basic TCP/IP configuration for all adapters- **ipconfig**

Display the full TCP/IP configuration for all adapters – **ipconfig /all**

Renew a DHCP-assigned IP address configuration for only the LAN adapter **- ipconfig /renew LAN**

Flush the DNS resolver cache when troubleshooting DNS name resolution problems – **ipconfig /flushdns**

route

route is a utility used to view and manipulate the IP routing table in operating systems.

Each packet that's processed by the computer is evaluated against the rules in the routing table. If the packet's destination address matches the destination subnet for the rule, the packet is sent to the specified gateway via the specified network interface. If not, the next rule is applied.

Command Prompt

```
C:\Users>route print
===========================================================================
Interface List
 10...00 ff 63 fd c3 ee ......Sophos SSL VPN Adapter
 18...98 e7 f4 33 53 e9 ......Intel(R) Ethernet Connection I219-LM
 17...0a 00 27 00 00 11 ......VirtualBox Host-Only Ethernet Adapter
 13...ce b0 da 13 27 b4 ......Microsoft Wi-Fi Direct Virtual Adapter
 19...ce b0 da 13 2f b4 ......Microsoft Wi-Fi Direct Virtual Adapter #2
  3...cc b0 da 13 27 b4 ......Broadcom BCM943228Z 802.11abgn 2x2 M.2 Wi-Fi Adapter
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0    192.168.100.1    192.168.100.6     50
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
     192.168.56.0    255.255.255.0         On-link      192.168.56.1    281
     192.168.56.1  255.255.255.255         On-link      192.168.56.1    281
   192.168.56.255  255.255.255.255         On-link      192.168.56.1    281
    192.168.100.0    255.255.255.0         On-link     192.168.100.6    306
    192.168.100.6  255.255.255.255         On-link     192.168.100.6    306
```

Syntax and Examples

Syntax:

route [-f] [-p] [-4|-6] [Command [Destination] [mask Netmask] [Gateway] metric Metric if Interface

Examples:

Print the local route table – **route print**

Add static route to 192.168.0.0/25 network :

route ADD 192.168.0.0 MASK 255.255.255.0 192.168.100.1 METRIC 3 IF 2

Change existing route: **route CHANGE 157.0.0.0 MASK 255.0.0.0 157.55.80.5 METRIC 2 IF 2**

Delete route: **route delete 192.168.0.0**

NSLookup

Nslookup displays information that you can use to diagnose Domain Name System (DNS) infrastructure. The nslookup command-line tool is available only if the TCP/IP protocol Is installed

The nslookup command-line tool has two modes: interactive and noninteractive.

If you need to look up only a single piece of data, we recommend using the non-interactive mode.

If you need to look up more than one piece of data, you can use interactive mode.

Command Prompt

```
C:\Users>
C:\Users>nslookup /?
Usage:
   nslookup [-opt ...]             # interactive mode using default server
   nslookup [-opt ...] - server    # interactive mode using 'server'
   nslookup [-opt ...] host        # just look up 'host' using default server
   nslookup [-opt ...] host server # just look up 'host' using 'server'

C:\Users>nslookup tut.by 8.8.8.8
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    tut.by
Addresses:  2a0a:7d80::c
          2a0a:7d80::d
          2a0a:7d80::b
          2a0a:7d80::a
          178.172.160.4
          178.172.160.3
          178.172.160.5
          178.172.160.2


C:\Users>
```

Syntax and Examples

Syntax:

nslookup [exit | finger | help | ls | lserver | root | server | set | view] [options]

Full syntax and parameters

Examples:

Display command help – **nslookup /?**

Display MX record for redhat.com site: **nslookup -query=mx redhat.com**

Reverse DNS Lookup: **nslookup 209.132.183.181**

DNS lookup site redhat.com by using non-default ns1.redhat.com DNS server: **nslookup redhat.com ns1.redhat.com**

Nslookup by using specific port 56: **nslookup -port 56 redhat.com**

Nslookup with debug options: **nslookup -debug redhat.com**

Ping

The **ping** command verifies IP-level connectivity to another TCP/IP computer by sending Internet Control Message Protocol (ICMP) echo Request messages. The receipt of corresponding echo Reply messages are displayed, along with round-trip times. ping is the primary TCP/IP command used to troubleshoot connectivity, reachability, and name resolution.

Command Prompt

```
C:\Users>ping -n 10 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=15ms TTL=117
Reply from 8.8.8.8: bytes=32 time=61ms TTL=117
Reply from 8.8.8.8: bytes=32 time=76ms TTL=117
Reply from 8.8.8.8: bytes=32 time=80ms TTL=117
Reply from 8.8.8.8: bytes=32 time=15ms TTL=117
Reply from 8.8.8.8: bytes=32 time=19ms TTL=117
Reply from 8.8.8.8: bytes=32 time=19ms TTL=117
Reply from 8.8.8.8: bytes=32 time=18ms TTL=117
Reply from 8.8.8.8: bytes=32 time=19ms TTL=117
Reply from 8.8.8.8: bytes=32 time=16ms TTL=117

Ping statistics for 8.8.8.8:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 15ms, Maximum = 80ms, Average = 33ms
```

Syntax and Examples

Syntax:

ping [-t] [-a] [-n <Count>] [-l <Size>] [-f] [-I <TTL>] [-v <TOS>] [-r <Count>] [-s <Count>] [{-j <Hostlist> |

Full syntax and parameters

Examples:

Display command help – **ping /?**

Ping by DNS name - **ping google.com**

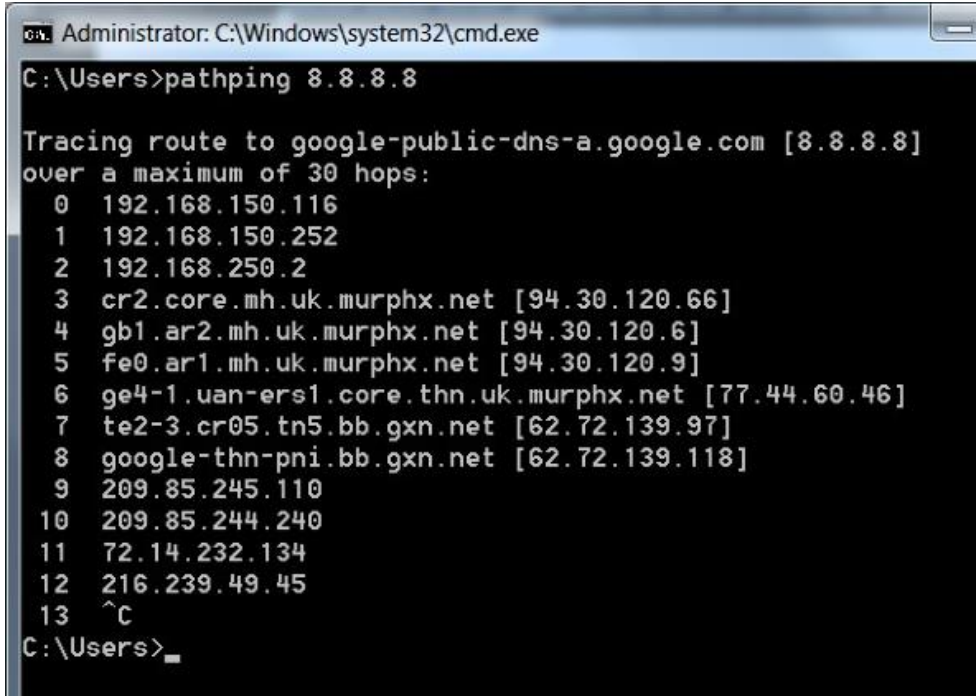
Ping the destination 37.17.10.55 and resolve to its host name - **ping -a 37.17.10.55**

Ping the destination 37.17.10.55 and record the route for 4 hops **- ping -r 4 37.17.10.55**

Ping the destination 37.17.10.55 with 10 echo Request messages, each of which has a Data field of 1000 bytes – **ping -n 10 -l 1000 37.17.10.55**

PathPing

Provides information about network latency and network loss at intermediate hops between a source and destination. **pathping** sends multiple echo Request messages to each router between a source and destination over a period of time and then computes results based on the packets returned from each router.
Because **pathping** displays the degree of packet loss at any given router or link, you can determine which routers or subnets might be having network problems.



Syntax and Examples

Syntax:

pathping [/n] [/h] [/g <Hostlist>] [/p <Period>] [/q <NumQueries> [/w <timeout>] [/i <IPaddress>]

Full syntax and parameters

Examples:

Display command help – **pathping /?**

Pathping google.com- **pathping google.com**

Pathping google.com with wait period and max hops – **pathping –p 10 –h 20 google.com**

Tracert

Tracert determines the path taken to a destination by sending ICMP echo Request or ICMPv6 messages to the destination with incrementally increasing time to Live (TTL) field values.

The path displayed is the list of near/side router interfaces of the routers in the path between a source host and a destination. The near/side interface is the interface of the router that is closest to the sending host in the path.

C:\WINDOWS\system32\cmd.exe

```
icrosoft Windows [Version 10.0.18363.720]
c) 2019 Microsoft Corporation. All rights reserved.

:\Users\anton_pavlovsky>tracert 8.8.8.8

racing route to dns.google [8.8.8.8]
ver a maximum of 30 hops:

  1     7 ms     2 ms     1 ms  192.168.100.1
  2     7 ms     5 ms     5 ms  mm-1-72-122-178.mgts.dynamic.pppoe.byfly.by [178.122.72.1]
  3     9 ms     9 ms    10 ms  mm-49-80-84-93.dynamic.pppoe.mgts.by [93.84.80.49]
  4     8 ms    10 ms    10 ms  core1.net.belpak.by [93.85.80.45]
  5    10 ms     7 ms    10 ms  ie1.net.belpak.by [93.85.80.38]
  6    27 ms     4 ms     4 ms  asbr10.net.belpak.by [93.85.80.229]
  7    33 ms    27 ms    31 ms  194.158.197.209
  8    27 ms    27 ms    28 ms  108.170.231.235
  9    29 ms    27 ms    27 ms  72.14.234.115
 10    53 ms    33 ms    32 ms  dns.google [8.8.8.8]

race complete.

:\Users\anton_pavlovsky>
```

Syntax and Examples

Syntax:

tracert [/d] [/h <MaximumHops>] [/j <Hostlist>] [/w <timeout>] [/R] [/S <Srcaddr>] [/4][/6] <TargetName>

Full syntax and parameters

Examples:

Display command help – **tracert /?**

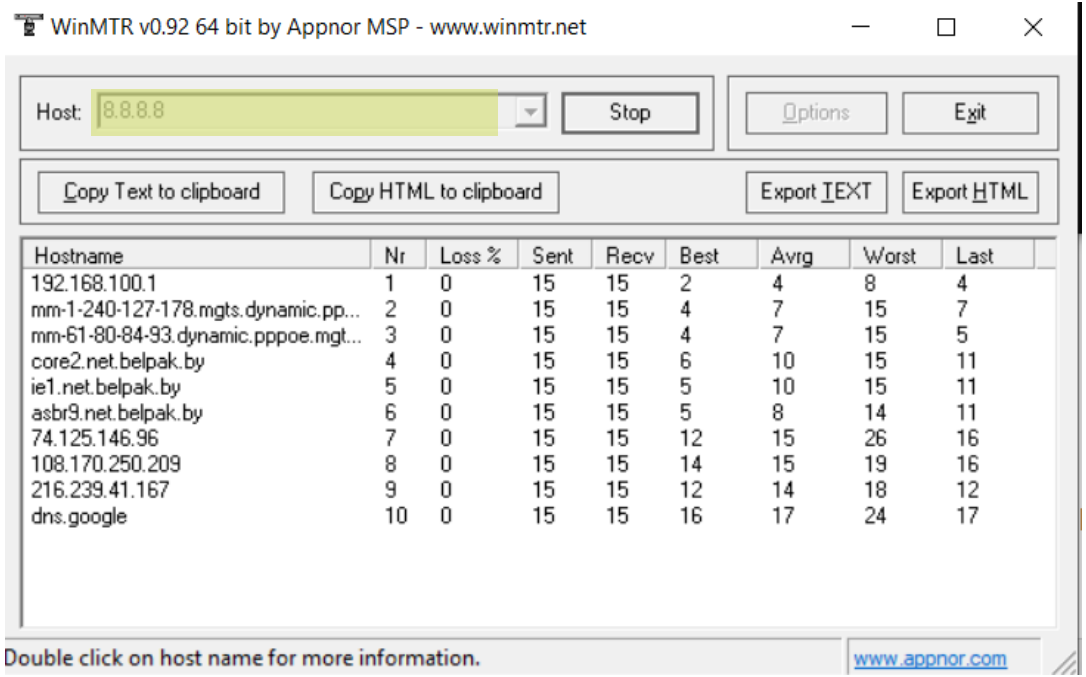
To trace the path to google.com and prevent the resolution of each IP address to its name:
tracert –d google.com

Trace the path to google.com and use the loose source route 10.12.0.1/10.29.3.1/10.1.44.1:
tracert /j 10.12.0.1 10.29.3.1 10.1.44.1 google.com

WinMTR

WinMTR (My traceroute) allows you to constantly poll a remote server. This diagnostic tool combines the ‘traceroute’ and ‘ping’ function.

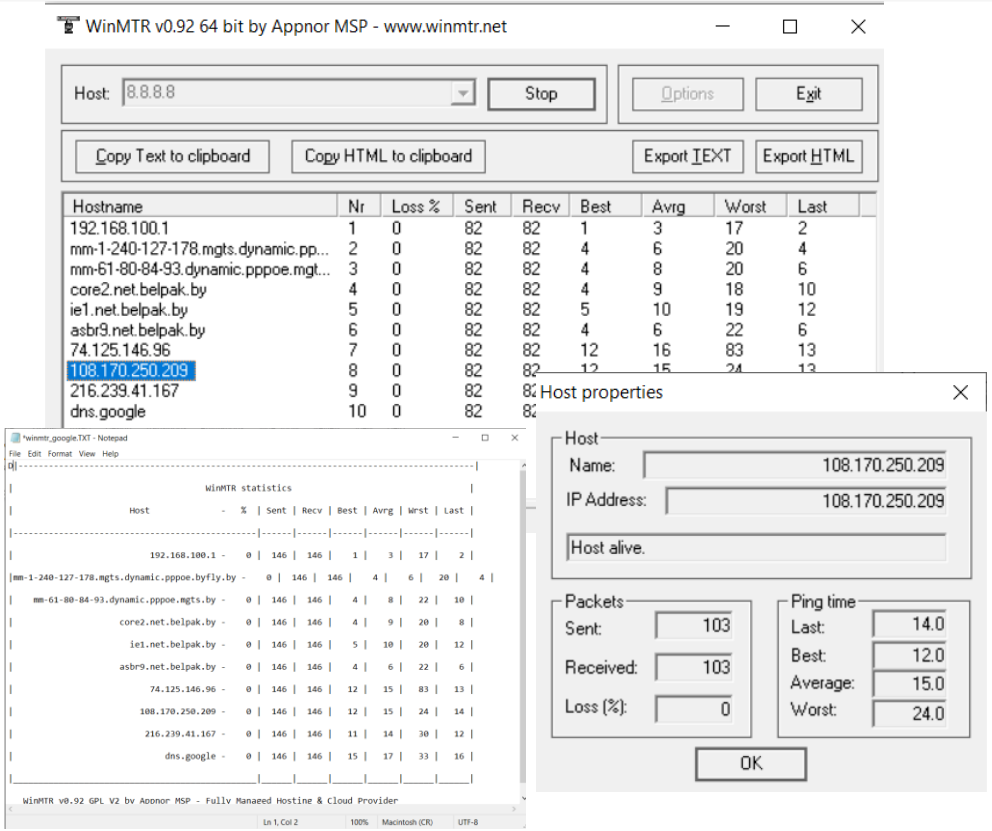
It represents an evolution of the traceroute command by providing a greater data sample as if augmenting traceroute with ping output and is useful for seeing how a server’s latency and performance changes over time.



Example

How to install and use:

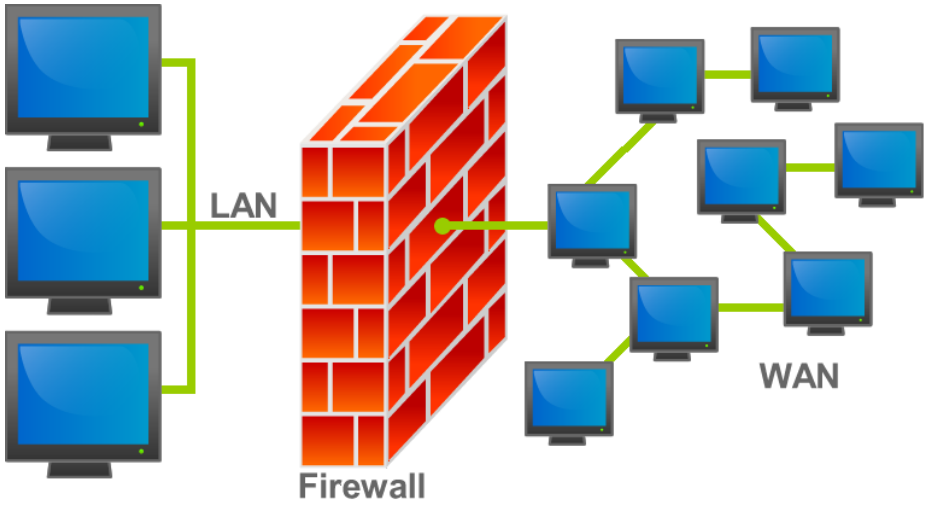
1. Download the latest version of WinMTR
2. Extract the **WinMTR.exe** from downloaded ZIP archive to Windows\System32 or other folder
3. Open CLI and run winmtr.exe



Firewalls

A **firewall** is a system that provides network security by filtering incoming and outgoing network traffic based on a set of user-defined rules. In general, the purpose of a firewall is to reduce or eliminate the occurrence of unwanted network communications while allowing all legitimate communication to flow freely.

In most server infrastructures, firewalls provide an essential layer of security that, combined with other measures, prevent attackers from accessing your servers in malicious ways.



THANK YOU!