Windows Command Line

Learned:

- Commands like `ping`, `tracert`, and `nslookup` help verify network connectivity and troubleshoot routing issues, which can be useful in detecting potential network attacks or failures.

- `netstat` allows you to know which connection's are active and view the listening ports, this would help to spot unusual or suspicious activity on the system.

- File and directory commands `cd`, `dir`, `mkdir`, `rmdir`, `copy`, `move`, `del` are useful for managing files and directories, including identifying malicious files.

- Commands such as `chkdsk`, `sfc /scannow`, `driverquery` can verify the system integrity and ensure that system files and drivers haven't been tampered with or corrupted.

Windows Command

We can run the `systeminfo` command to list various information about the system such as OS information, system details, processor and memory.

Terminal

```
C:\>systeminfo

Host Name:                 WIN-SRV-2019
OS Name:                   Microsoft Windows Server 2019 Datacenter
OS Version:                10.0.17763 N/A Build 17763
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             Multiprocessor Free
[...]
```

you can pipe it through `more` if the output is too long.

`driverquery | more.`

You can also use `ipconfig /all` for more information about your network configuration. As shown in the terminal below, we can view our DNS servers and confirm that DHCP is enabled.

Terminal

```
C:\>ipconfig /all

Ethernet adapter Ethernet 3:

   Connection-specific DNS Suffix  . : eu-west-1.compute.internal
   Description . . . . . . . . . . . : Amazon Elastic Network Adapter
   Physical Address. . . . . . . . . : 02-B7-DF-1D-0D-99
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::90df:4861:ba40:f2a8%4(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.10.230.237(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Lease Obtained. . . . . . . . . . : Wednesday, May 1, 2024 2:38:05 PM
   Lease Expires . . . . . . . . . . : Wednesday, May 1, 2024 4:08:07 PM
   Default Gateway . . . . . . . . . : 10.10.0.1
   DHCP Server . . . . . . . . . . . : 10.10.0.1
   DHCPv6 IAID . . . . . . . . . . . : 134353458
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-27-E3-D1-2B-0E-F8-30-D0-72-3F
   DNS Servers . . . . . . . . . . . : 10.0.0.2
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Network Troubleshooting

One common troubleshooting task is checking if the server can access a particular server on the Internet. The command syntax is `ping target_name`. We send a ICMP packet and listen for a response. If a response is received, we know that we can reach the target and that the target can reach us.

Terminal

```
C:\>ping example.com

Pinging example.com [93.184.215.14] with 32 bytes of data:
Reply from 93.184.215.14: bytes=32 time=78ms TTL=52
Reply from 93.184.215.14: bytes=32 time=78ms TTL=52
Reply from 93.184.215.14: bytes=32 time=78ms TTL=52
Reply from 93.184.215.14: bytes=32 time=78ms TTL=52

Ping statistics for 93.184.215.14:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 78ms, Maximum = 78ms, Average = 78ms
```

Another valuable tool for troubleshooting is `tracert`, which stands for *trace route*. The command `tracert target_name` traces the network route traversed to reach the target.

Terminal

```
C:\>tracert example.com

Tracing route to example.com [93.184.215.14]
over a maximum of 30 hops:

  1    59 ms    32 ms    42 ms  ec2-3-248-240-3.eu-west-1.compute.amazonaws.com
[3.248.240.3]
  2     *        *        *     Request timed out.
  3     *        *        *     Request timed out.
  4     *        *        *     Request timed out.
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8     *        *        *     Request timed out.
  9    <1 ms    13 ms    <1 ms  100.100.2.56
 10    15 ms    11 ms    11 ms  ae-42.a03.londen12.uk.bb.gin.ntt.net
[131.103.117.104]
 11    17 ms    11 ms    12 ms  ae-14.r20.londen12.uk.bb.gin.ntt.net
[129.250.3.248]
 12    81 ms    80 ms    80 ms  ae-7.r20.nwrknj03.us.bb.gin.ntt.net [129.250.6.147]
 13    83 ms    83 ms    86 ms  ae-0.a02.nycmny17.us.bb.gin.ntt.net [129.250.3.9]
 14    79 ms    79 ms    96 ms  ce-0-3-0.a02.nycmny17.us.ce.gin.ntt.net
[128.241.1.14]
 15    81 ms    86 ms    79 ms  ae-67.core1.nyd.edgecastcdn.net [152.195.68.135]
 16    78 ms    78 ms    78 ms  93.184.215.14

Trace complete.
```

`nslookup`. It looks up a host or domain and returns its IP address. The syntax `nslookup example.com` will look up `example.com` using the default name server; however, `nslookup example.com 1.1.1.1` will use the name server `one.one.one.one`.

Terminal

```
C:\>nslookup example.com
Server:  ip-10-0-0-2.eu-west-1.compute.internal
Address:  10.0.0.2

Non-authoritative answer:
```

```
Name:    example.com
Addresses:  2606:2800:21f:cb07:6820:80da:af6b:8b2c
          93.184.215.14

C:>nslookup example.com 1.1.1.1
Server:  one.one.one.one
Address:  1.1.1.1

Non-authoritative answer:
Name:    example.com
Addresses:  2606:2800:21f:cb07:6820:80da:af6b:8b2c
          93.184.215.14
```

This command displays current network connections and listening ports. A basic `netstat` command with no arguments will show you established connections.

Terminal

```
C:\>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    10.10.230.237:22       ip-10-11-81-126:53486  ESTABLISHED
```

Netstat Commands with arguments:

- `-a` displays all established connections and listening ports
- `-b` shows the program associated with each listening port and established connection
- `-o` reveals the process ID (PID) associated with the connection
- `-n` uses a numerical form for addresses and port numbers

We combine these four options and execute the `netstat -abon` command. The result is quite long, but we display the first few lines in the terminal below. It is clear now that the executable `sshd.exe` is responsible for listening for incoming connections on port 22, as shown in the first line. We can also see the process ID (PID) associated with each connection.

Terminal

```
C:\>netstat -abon
```

```
Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    0.0.0.0:22             0.0.0.0:0              LISTENING       2116
 [sshd.exe]
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING       820
  RpcSs
 [svchost.exe]
[...]
  TCP    0.0.0.0:49669          0.0.0.0:0              LISTENING       2036
 [spoolsv.exe]
  TCP    0.0.0.0:49670          0.0.0.0:0              LISTENING       584
 Can not obtain ownership information
  TCP    0.0.0.0:49686          0.0.0.0:0              LISTENING       592
 [lsass.exe]
  TCP    10.10.230.237:22       10.11.81.126:53486     ESTABLISHED     2116
 [sshd.exe]
 [...]
```

Using `ipconfig/all` will allow you to see the MAC address.

File and Disk Management

You can use `cd` without parameters to display the current drive and directory. It is the equivalent of asking the system, *where am I?*

You can view the child directories using `dir`.

Terminal

```
C:\Users\strategos>cd
C:\Users\strategos

C:\Users\strategos>dir
 Volume in drive C has no label.
 Volume Serial Number is A8A4-C362

 Directory of C:\Users\strategos

05/01/2024  02:40 PM    <DIR>          .
05/01/2024  02:40 PM    <DIR>          ..
11/14/2018  06:56 AM    <DIR>          Desktop
05/01/2024  02:40 PM    <DIR>          Documents
09/15/2018  07:19 AM    <DIR>          Downloads
```

```
09/15/2018  07:19 AM    <DIR>          Favorites
09/15/2018  07:19 AM    <DIR>          Links
09/15/2018  07:19 AM    <DIR>          Music
09/15/2018  07:19 AM    <DIR>          Pictures
09/15/2018  07:19 AM    <DIR>          Saved Games
09/15/2018  07:19 AM    <DIR>          Videos
               0 File(s)              0 bytes
              11 Dir(s)  14,984,953,856 bytes free
```

- `dir /a` - Displays hidden and system files as well.
- `dir /s` - Displays files in the current directory and all subdirectories.

You can type `tree` to visually represent the child directories and subdirectories.

Terminal

```
C:\Users\strategos>tree
Folder PATH listing
Volume serial number is A8A4-C362
C:.
├───Desktop
├───Documents
├───Downloads
├───Favorites
├───Links
├───Music
├───Pictures
├───Saved Games
└───Videos
```

You can use `cd ..` to go up one level.

Terminal

```
C:\>cd
C:\

C:\>cd Users

C:\Users>cd
C:\Users

C:\Users>cd ..
```

```
C:\>cd
C:\
```

To create a directory, use `mkdir directory_name`; `mkdir` stands for *make directory*. To delete a directory, use `rmdir directory_name`; `rmdir` stands for *remove directory*.

Terminal

```
C:\example>mkdir backup_files

strategos@WIN-SRV-2019 C:\example>dir
 Directory of C:\example

05/02/2024  07:36 AM    <DIR>          .
05/02/2024  07:36 AM    <DIR>          ..
05/02/2024  07:36 AM    <DIR>          backup_files
               0 File(s)              0 bytes
               3 Dir(s)  14,984,724,480 bytes free

C:\example>rmdir backup_files

C:\example>dir
 Directory of C:\example

05/02/2024  07:36 AM    <DIR>          .
05/02/2024  07:36 AM    <DIR>          ..
               0 File(s)              0 bytes
               2 Dir(s)  14,984,724,480 bytes free
```

Working With Files

You can easily view text files with the command `type`. This command will dump the contents of the text file on the screen; this is convenient for files that fit within your terminal window.

For long text files, `more` will display a single page and wait for you to press `Spacebar` to move by one page or `Enter` to move by one line.

The `copy` command allows you to copy files from one location to another.

Terminal

```
C:\example>dir
 Directory of C:\example

05/02/2024  08:12 AM    <DIR>          .
```

```
05/02/2024  08:12 AM    <DIR>          ..
05/02/2024  07:57 AM                17 test.txt
               1 File(s)             17 bytes
               2 Dir(s)  14,983,409,664 bytes free

C:\example>copy test.txt test2.txt
        1 file(s) copied.

C:\example>dir
 Directory of C:\example

05/02/2024  08:12 AM    <DIR>          .
05/02/2024  08:12 AM    <DIR>          ..
05/02/2024  07:57 AM                17 test.txt
05/02/2024  07:57 AM                17 test2.txt
               2 File(s)             34 bytes
               2 Dir(s)  14,983,409,664 bytes free
```

You can move files using the `move` command.

Terminal

```
C:\example>dir
 Directory of C:\example

05/02/2024  08:12 AM    <DIR>          .
05/02/2024  08:12 AM    <DIR>          ..
05/02/2024  07:57 AM                17 test.txt
05/02/2024  07:57 AM                17 test2.txt
               2 File(s)             34 bytes
               2 Dir(s)  14,983,409,664 bytes free

C:\example>move test2.txt ..
        1 file(s) moved.

C:\example>dir
 Directory of C:\example

05/02/2024  08:13 AM    <DIR>          .
05/02/2024  08:13 AM    <DIR>          ..
05/02/2024  07:57 AM                17 test.txt
               1 File(s)             17 bytes
               2 Dir(s)  14,983,409,664 bytes free
```

We can delete a file using `del` or `erase`.

Terminal

```
C:\example>dir
 Directory of C:\example

05/02/2024  08:16 AM    <DIR>          .
05/02/2024  08:16 AM    <DIR>          ..
05/02/2024  07:57 AM                17 test.txt
05/02/2024  07:57 AM                17 test2.txt
               2 File(s)             34 bytes
               2 Dir(s)  14,983,409,664 bytes free

C:\example>erase test2.txt

C:\example>dir
 Directory of C:\example

05/02/2024  08:16 AM    <DIR>          .
05/02/2024  08:16 AM    <DIR>          ..
05/02/2024  07:57 AM                17 test.txt
               1 File(s)             17 bytes
               2 Dir(s)  14,983,409,664 bytes free
```

You can also use the wildcard character `*` to refer to multiple files. For example, `copy *.md C:\Markdown` will copy all files with the extension `md` to the directory `C:\Markdown`.

Task and Process Management

We can list the running processes using `tasklist`.

Terminal

```
C:\>tasklist

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0          8 K
System                           4 Services                   0         88 K
```

```
Registry                      84 Services                   0     50,700 K
smss.exe                     276 Services                   0      1,132 K
csrss.exe                    372 Services                   0      5,264 K
wininit.exe                  448 Services                   0      6,892 K
csrss.exe                    456 Console                    1      5,028 K
winlogon.exe                 516 Console                    1     11,144 K
services.exe                 584 Services                   0      7,492 K
lsass.exe                    592 Services                   0     16,108 K
svchost.exe                  704 Services                   0     23,432 K
fontdrvhost.exe              736 Console                    1      4,256 K
[...]
```

You can check all available filters by displaying the help page using `tasklist /?`. Let's say that we want to search for tasks related to `sshd.exe`, we can do that with the command `tasklist /FI "imagename eq sshd.exe"`.

Terminal

```
C:\>tasklist /FI "imagename eq sshd.exe"

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
sshd.exe                      2116 Services                   0      6,992 K
sshd.exe                      2712 Services                   0      7,668 K
sshd.exe                      4752 Services
```

With the process ID (PID) known, we can terminate any task using `taskkill /PID target_pid`. For example, if we want to kill the process with PID `4567`, we would issue the command `taskkill /PID 4567`.

- `chkdsk`: checks the file system and disk volumes for errors and bad sectors.
- `driverquery`: displays a list of installed device drivers.
- `sfc /scannow`: scans system files for corruption and repairs them if possible.
- `/?` can be used with most commands to display a help page.
- we used the command `more` in two ways:
  - Display text files: `more file.txt`
  - Pipe long output to view it page by page: `some_command | more`