

Windows and Linux Terminals & Command Lines

Tools and Tips for SEC301 and SEC401

Getting started:

- **c:\>** denotes a command to be run from Windows' cmd.exe
- **user\$** is for a Linux command
- **root#** means the Linux command needs to be run as a privileged, root user
- Linux commands are generally case-sensitive; Windows commands are generally not
- Mac terminals will, in most ways, act like Linux/Bash terminals

What directory am I in?

```
c:\> cd
user$ pwd
```

What files are in this directory?

```
c:\> dir
user$ ls -l
```

Copy a file

```
c:\> copy file.txt copy.txt
user$ cp file.txt copy.txt
```

Erase a file

```
c:\> erase file.txt
user$ rm file.txt
```

Print the contents of a file to the screen

```
c:\> type file.txt
user$ cat file.txt (Just dump the raw file)
user$ strings file.txt
(Dump only the readable characters)
```

See one screen at a time

```
c:\> type file.txt | more
user$ cat file.txt | more
user$ more file.txt (Same, just shorter)
user$ less file.txt (Same, but you can go up and down; q to quit)
```

Put text into a file

```
c:\> echo "Four score" > 1.txt
user$ echo "Four score" > 1.txt
```

Add text to a file

```
c:\> echo "and seven years" >> 1.txt
user$ echo "and seven years" >> 1.txt
```

Combine two files

```
c:\> type 1.txt 2.txt > 3.txt
user$ cat 1.txt 2.txt > 3.txt
```

Check who you're logged in as

```
c:\> whoami
user$ whoami
```

Hide command error messages

```
c:\> YOUR COMMAND 2>nul
user$ YOUR COMMAND 2>/dev/null
```

Find files in a filesystem

```
c:\> dir c:\ /b/s | find "password"
user$ find / -name *password*
user$ locate password (same, but faster)
root# updatedb (update the database for locate by indexing everything in the drive)
```

View all environment variables

```
c:\> set
user$ env
```

View one environment variable

```
c:\> set Path
c:\> echo %Path%
user$ env | grep PATH
user$ echo $PATH
```

What are environment variables?

They give your terminal context for running certain commands. For example, the PATH variable, in most operation systems, tells your terminal which directories to look in for programs when you type one in.

Note: the current directory, . (period), is in the Windows path by default - but not in Linux. So in Linux, we must be explicit when running something in our current working directory:

Run john when it's in your directory

```
c:\> john.exe
user$ ./john
```



The most trusted source for
cybersecurity training, certifications,
degrees, and research

See which ports the computer is listening for connections on

```
c:\> netstat -nao
c:\> netstat -naob (Same, but lists process name; requires Administrator)
user$ netstat -ant
root# netstat -pant (Same, but lists pid and name; requires root)
```

Look for lines containing specific text, e.g. 9999

```
c:\> netstat -naob | find 9999
root# user$ netstat -pant | grep 9999
```

See what tasks are running

```
c:\> tasklist
c:\> wmic process list full (Same, more info)
user$ ps -aux
```

Get more info about a specific process id, e.g. 45

```
c:\> wmic process where ProcessID=45
user$ ps -Flww -p 45
```

Check the system's hostname

```
c:\> hostname
user$ hostname
```

List processes that run at startup

```
c:\> wmic startup full list
user$ ls -l /etc/init.d
user$ crontab -l
user$ systemctl list-unit-files | grep enabled
user$ less /home/user/.bashrc
(There are other places where startup tasks can be stored in Linux, but these are the most common)
```

Scan a host to look for open ports, e.g. 192.168.1.100

```
c:\> nmap 192.168.1.100
user$ nmap 192.168.1.100
```

Scan a subnet of hosts and see what is really running on open ports

```
c:\> nmap 192.168.1.1-254 -sV
user$ nmap 192.168.1.1-254 -sV
```

Scan all 65,536 ports on a given host

```
c:\> nmap 192.168.1.100 -p0-65535
user$ nmap 192.168.1.100 -p0-65535
```

Ping another host four times

```
c:\> ping 192.168.1.200
user$ ping -c4 192.168.1.200
```

Connect to port 25 to see what banner it sends back, e.g. SMTP or 25/TCP

```
c:\> nc.exe 192.168.1.100 25
(Not installed by default)
user$ nc 192.168.1.100 25
(Usually available)
```

See your IP address(es)

```
c:\> ipconfig
user$ ip addr
```

Get help for a command (these work for most commands)

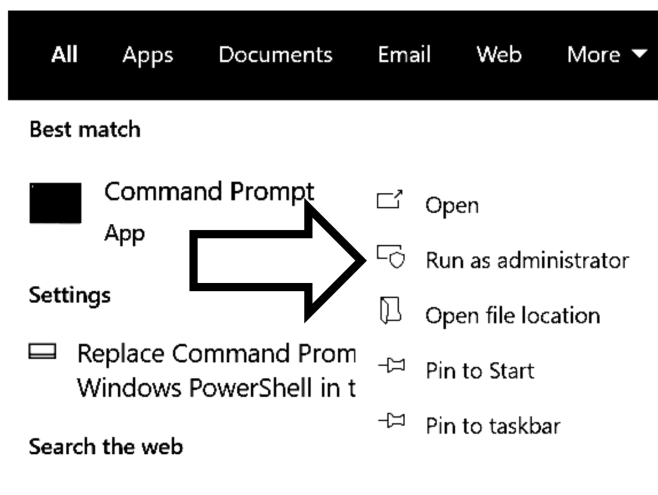
```
c:\> cd /?
user$ man cd (quit with q)
user$ cd --help (shorter output)
```

See what path executables will run from

```
c:\> echo %PATH%
user$ echo $PATH
```

Opening a Command Prompt/Terminal

In Windows, reach the cmd.exe command prompt by clicking the Windows button and typing **cmd**. If you need to run as a privileged user, right-click **Command Prompt** and choose **Run as administrator**.



In Slingshot Linux, open a Bash terminal by double-clicking **MATE Terminal** on the desktop. In some Linux systems, you can use **<Ctrl><Alt>t** as a keyboard shortcut. Use **su** to switch to a privileged/root user or **sudo YOUR COMMAND** to run a single command as root.

