Report 2

The first thing that we did in class was to practice the Basic Commands in the Windows Command Line. To be honest, there are only some commands that I remember from using the Windows Terminal. The first command we went over is on the **cd** command. This is simple; it changes the directory. The next command was **ver.** This one was not too complicated either. It displays your windows version. The **clip** command was an interesting one. This copies the result of the preceding command. We used the **time /t | clip to** copy the result and then pasted it in Notepad. This command might not seem like much; however, it seems like a very handy command to utilize If you are executing a command with several outputs, and you are interested in the results. I haven't considered this but maybe its command is used in automation. Now, the next command is the **mode**. You can change the number of presses each key does. The lines and columns on a given page. Also, the delay when you press a key. I can see how this command can cause problems for a user if it is used for malicious purposes. Also, seems like a command that you can show off to your friend while they are on the receiving end as a joke. The date & time command was self-explanatory. It's neat that removing the **/t** allows you to set the time and date. Despite that, it doesn't seem like a useful command other than in operating systems that do not have a GUI like a custom Debian server. The next 3 were key commands we went over some of the ones I was not aware. **F7** displays the command history. If I had known this command previously, I know it would have saved me time locating a command I had typed in the past. Also helps delete commands from the history. **F8 is** kind of like pressing the Up Arrow Key but the difference is that you can get suggestions in commands that are related to any of the commands you typed based on what is saved in your command history. To coincide with that, **F9** allows users to select a command that was saved in history and allow them to display that command based on the number that command was entered. This is not only great for automation but also for making a custom list of commands already saved in the command line history. Personally, I can already see myself using F9 to make network tests much easier. With that in mind, maybe there is some way to have a custom list saved and not be overrated by any new commands that are typed. This was the end of the Basic Commands.

The next set of commands involves File and Directory commands. The first one was the **echo** command**.** I recognize this one from using the Linux Terminal. There were 2 states that you can utilize for this command. The first one to go over is the **echo off** command. Basically, it allows you to enter commands with no output. On that note, the second state this command has is the **echo on** which is used for turning on the output. Turning it off seems to be useful when you are debugging. Also, its command attackers will likely use to write many commands instantaneity without any delay. I am not sure if those commands are saved in your history. The next command dealt with copying files. Yet another simple command that also is utilized in various ways. One command that is as simple is the **ren** command. As the name implies this just renames a file. I wonder if you can use this to change the extension of the file as well. Another command that is simple is the move command. This allows you to move files to a different directory. The next command we went over is **comp**. This command compares the contents of the 2 files. I wonder what is used to calculate the comparison between 2 different files. On that matter, the **fc** command compares two files using binary. I was not aware that Windows had a command like this. I could see a command like this being used to gather forensic evidence. Deleting files is easy as it is which is the **del** command. This got me thinking, will files be deleted the same way as deleting a file in the recycle bin? We know that files can be recovered after deletion under most circumstances. The next command was **md/mkdir.** Just as the one on Linux, this makes creates an empty directory. **rd** is much the same as it removes a directory. The next commands were the **pucd** and **popd.** You can change to a working directory. With the other command, you can make previous changes as well. This is the end of the File and Directory Commands.

The next set of commands involved networking. The first command is an easy one. **Hostname** displays the host's name. My name for the desktop is Eduardo-PC. Since you can change the name of your computer in the settings, there must be a command to allow you to change this in the command line as well. The next command is the **getmac.** This is one of the commands I remember using in my Cybersecurity class. This command displays the MAC address of all the interfaces that are available. I also knew that you could change your MAC address as well. The next series of commands is one everyone should know. The first one is the **ping** command. This is used to test network connections using the ICMP protocol. Very useful to test connections to websites. You can also change the number of replies you receive and the packets you send. The second is **pathping**. As the name implies, the path an address takes are outlined by a number of hops, shows the network latency, and shows a detailed view of how many packets were lost. An interesting thing that I noticed it that that one of the entries in my path was omitted. I think this can either mean one of two things. One, there could be congestion in the network therefore it does not retrieve that information. Two, the information on that route is blocked for security purposes. The next command is the **tracert.** This displays the IP address of routers between your computers and your networks. The **route** commands display the routing table. The address resolution command **arp** displays and can modify data in the ARP cache on the local machine. I remember that an attacker can poison the contents of the ARP catch and make appear as if you are a genuine user. The next command was the **netstat**. This one shows the current connections of TCP/IP connections. This is the end of the network commands.

The next set of commands was the Partitions and Drives command. The first thing we had to do was download a utility called SysInternalSuite. This reminds me of installing packages on Linux. Where you get can utilize extra commands that do more specialized things than you want. The first command was the **coreinfo**. It essentially displays information about the CPU and the cache memory. I did not know Windows had this package. Gunna experiment with it more on my PC since my Laptop is less powerful than my PC. The next command was **cleanmgr.** This proves a disk cleanup for your selected drive. The **chkdsk** commands check the status on the disk. I used this when I had to check a hard drive was not operating properly. Turns out if was full of bad sectors. I ended up replacing the drive. The **driverquery** command shows all installed device drivers on the machine. This command is too useful not to have in the Windows Command Line by default. This is the end of the Drives and Partition Commands

The next set of commands involves displaying, setting, or removing environment variables. These are the Environment Commands. There is a lot that you can do with this command so what was shown to us was just a glimpse. I noticed that you could assign the number of processors. So, I was thinking if you can use this command to set a certain amount of processor to a program that is running. The next command was the **doskey.** It is like an alias like in Linux; however, in my case the commands that had alias only worked for that session in the Windows Commands Line. When I opened another window, it did not recognize my alias. Maybe the command only affects that part of the session only? This is the end of the Environment Commands. This is the end of the Environment Commands.

The set of commands involves the Services Commands and the SysInternals Networking Suite. The first one is **tasklist.** Very similar to the task manager application that you can open on Windows. You can see a list of tasks that are currently running in process. To end one, you can you the **taskkill** followed by a PID number. The s**c** command which can be used in conjunction with **queryex eventlog** gives a detailed description of the state, type, PID, and any relevant information. The **whois** command retrieves the registration record for the domain name or IP address that you want to specify. I tried this in my home network and there was at least one server that was omitted from the results. The **psgetsid** command retrieves the security identifier. I am not sure what this identifier does. Could be a number that is generated when you activate you the Windows Operating system. The **pslist** is like the **tasklist.** The difference is that **in**stead of including the memory, it is replaced by the CPU time. Essentially its more tailored to getting process information regarding your CPU. On that note, **psinfo** shows the system information. You can see the Kernel version, type of product, version, and many other details. The next command **psping** utilizes TCP to measure the latency and bandwidth. You can change the amount replies you want to receive and the number of bytes each ping can contain. For example, if you want to ping yahoo.com with 2 replies and using TCP you can **type psping–n 2 -4 yahoo.com.** The next command is **tcpview.** A new window will open that shows all the TCP and UDP endpoints on your system. This is like Wireshark, but this command lists the TCP and UDP connections of both IPv4 and IPv6. The next command **procexp** opens a news window detailing information on each process. You can also dump a process to look for any potential errors in the process. You are also able to see if they contain any viruses. The next command **bginfo** shows the selected information of the computer in the background. This can include the IP address, the MAC, CPU specs, default gate way etc. This is the end of the Service and Networking suite commands.

The second part of lab was the Python portion. I had issues installing python on my main operating system, but I was able to do it in the virtual machine. The first thing we went over in Python was some of the basic aspects. The first fact that comes to mind is that I know about python is that one, it’s an interpreted object-oriented language. Unlike Java, each metadata in python is an object. In python an object resembles and entity that contains data along the associated functionality. One of the aspects that caught my attention in python is when you utilize the equals to sign to compare 2 numbers. In Java, the compare method **==** is used to compare if the objects are the same. This is regardless of what is contained in that data. In python, the == operator overrides the contents of the objects by overriding the equals method. In java == operator used to compare the reference of that object. If you want to compare numbers in python, you can use the **is** method to do that. The next part of the lab was discussing the data type differences between Java and Python. There was a difference that I noticed while working on with Python. In Java, there are primitive data types and non-primitive data types. In python these data types are dynamic. Java you need to define the data type and the data is tied to that data type. An int can only contain integers. With Python you don’t need to necessarily declare the data types in Python. Since Python automictically locates the type of variable it needs, it only performs operation based on that data itself. This is the end of the Python Portion.