13 March 2017

I) Notes:
1) Results from the most recent competition:
   a) 18th Place out of 32
   b) Qualified to go to PCDC
      - April 9th @ Trident Tech
      - Six-man team
2) Working with Linux Command Line:
   a) Why use the command line?
      - It can be much faster and easier than using a GUI once you know what you are doing.
   b) Bash (Born Again Shell) – Interprets your input, runs the commands as processes, and returns the output to you
   c) Terminal vs. Shell – Terminal is used to access a shell but the shell is what actually runs the commands
   d) *pwd* (Present Working Directory) – A command that prints where the user currently is in the file system; This location is known as the current working directory
   e) Linux File System – Everything starts at / (at \ in Windows); Important directories from here are */etc*, */var*, */home*, */bin*, and */root*
      - */etc* – Where a lot of configuration files for the system are
      - */var* – Variable files; Log files from system processes
      - */home* – Where all of the user directories live
      - */bin* – Means binary; Where binary executable live
      - */root* – Home folder for the root user
   f) You will see something like this in your terminal – *user@kali*: ~$:
      - *user* – Who you are currently logged in as
      - *@kali* – The OS of the machine in which the terminal is being run
      - $ (or # for *root*) – Known as the prompt; Where you type your commands
   g) *ls* (List) – A command that prints the contents of the present working directory
   h) *cd* (Change Directory) – A command that lets the user change their current directory; Follow the command with the directory you want to move to
      - *cd ..* – Moves you up one folder; For example, from */home/user* to */home*
      - *cd* – Moves you to the current user's home directory (on most systems)
      - *cd ~* – Moves you to the current user's home directory
   i) When typing in commands, you can use the *'Tab'* key to finish directories; For example, typing in *cd Des* and then hitting *'Tab'* might finish the directory to *cd Desktop* if you are in a user's home directory
   j) Paths:
      - Relative Path – A path that is relative to the present working directory
      - Absolute Path – Anything that begins with a /, specifying the entire path
   k) *ls - l* – The *ls* option prints information on the contents of the current working directory (like normal for *ls*) in columns and with some extra information
   l) *ls – l - a* – Same as above, but also shows hidden files and directories; Anything that begins with a . is hidden in Linux

m) Some Directories You Will See With ls - l - a:
   - . – The current working directory (just like what you would see with pwd)
   - .. – The directory that contains the current working directory
n) Extra Information Printed With ls - l - a:
   - Far Left Column (Looks something like $drwxr$- xr- x
     - First letter will most likely either be d (A Directory) or – (A File)
     - Next Three Letters are User Permissions
     - Next Three Letters are Group Permissions
     - Next Three Letters are World Permissions
     - Each three-letter grouping has three bits
       (i) First Bit – Read Permission
       (ii) Second Bit – Write Permission
       (iii)Third Bit – Execute Permission
o) $touch$ – Searches for the given file and then updates the timestamp to the current time; If the provided file is not found, the file is created
p) In order to edit a text file, you can use a text editor
q) $nano$ – Opens up a terminal-based editor for the provided file
r) cp – Copies the first provided file into a file with the second provided filename; If the second file does not exist, it is created
s) mv – Moves the provided file to the provided directory OR the provided filename, which essentially renames the file
t) rm – Deletes the provided file
u) Something You DO NOT Want to Do:
   - rm - rf / – Recursively removes all the files and folders that beginning in the root directory and working its way into all subsequent folders; DON'T DO THIS – IT WILL DELETE EVERYTHING!
v) You can use the up and down arrow keys to access previously typed commands
w) ps – A command that prints the current processes being run by the operating system
x) $ps\ aux$ – A command that prints all processes running on the system
y) | (Known as a Pipe) – Used to send the output of the command on the left as input for the command on the right
z) $grep$ – Goes line by line in the input and looks for the provided string; Something like $grep\ root$ will look through whatever is inputted for any instance of the string $root$
aa) wc (Word Count) – Simply put, counts and prints the number of string separated by a space in the given input
bb) wc - l – Counts the lines of the provided input
cc) Something like $ps\ aux$ | $grep\ root$ |
dd) Output Streams – Don't Worry Much About These:
   - $stdin$ – 0
   - $stdout$ – 1
   - $stderr$ – 2
ee) man (Manual) – Prints out the documentation (basically a manual) for the provided command

ff) $less$ – Puts the output into a text viewer; Allows you nicely scroll through the output; Typing / something will search through the output for the string that you typed in; Type q to exit