**Unit One Lab – Setting up and Testing Environment**

Description: In this lab, you will set up your environment for this class in Netlabs on an Ubuntu installation. It is highly recommended that you set up an environment at home on a Windows machine or Linux. Running Ubuntu on a VM in VMWare Workstation is tested and may give you flexibility. It can also be set up on a Mac, but I do not recommend it for this class. The document, *Set up Environment*, should be used with this lab to perform any procedures. We will not be doing everything in the above-named document in this lab, so follow the instructions below carefully **20 points**:

1. On Netlabs, the following **are already set up for you**, and **you should not change them**:
   1. Installed Python
   2. Installed Requests
   3. Installed PIP (separately on Linux)
   4. Installed IDLE3 (separately on Linux)
   5. Installed GIT
   6. Installed PyCharm
   7. Installed POSTMAN
   8. Installed Cisco AnyConnect VPN Client
2. It is recommended that you do the above on a home computer if you have the drive space. It will be easier to develop on your own computer outside of class, but you can complete the class without installing anything on your home computer. Use the, *Set up Environment*, document to set up and configure the above items. **You should postpone installing AnyConnect until we start using CML in the Cisco Developer Sandbox.**
3. In Netlab, set your display settings:
   1. Log into Ubuntu with student/student as your credentials
   2. Undock your PC in Netlab for best resolution
      1. On the client tab in Netlabs, click on the **down arrow** and select **undock**
   3. Right Click on your desktop and choose **display settings**
   4. It is recommended that you set your display to 1360 x 768, but you should choose a resolution that works best for you.
4. We will be using Linux as our platform. By now, you have had a Linux class in the program. You may be rusty, and that is alright. The exercises below are designed to help you refresh your memory on key Linux functions that we will be doing in the class. Fortunately, there is also plenty of help on the Internet, though Linux users tend to assume a certain level of understanding when posting responses on the Internet. <https://ubuntu.com/tutorials/command-line-for-beginners#1-overview> is an Ubuntu beginner tutorial that may serve you, though not all questions below are answered in this tutorial. You should review your notes and lab from your previous Linux class(es). Begin reviewing Linux by completing the following. When answering the questions, submit this sheet, with your answers and a screenshot showing the command or activity. Note that **you will submit your answers/screenshots below:**
   1. **What command tells you what your working directory is (the directory that you are currently in)?** 
      1. **pwd**
   2. **The “cd” command will move you between directories in your file system. What shortcut for the “cd” command will move your current directory to your home directory?**
      1. **Cd ~**
   3. **What shortcut for the “cd” command will move your current directory to the previous directory?**
      1. **Cd .**
   4. **What shortcut for the “cd” command will move your current directory to the root directory?**
      1. **Cd /**
   5. **What command lists the files in a directory?**
      1. **ls**
   6. **Do the following, being sure to submit a screenshot for each step:**
      1. **Create a text file called, MyTextFile, in your home directory, and place “Hello World” in the contents of the file.**

**Graphical user interface

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

* + 1. **Make a directory in your home directory called “HelloWorld”**

**A screenshot of a computer

Description automatically generated with medium confidence**

* + 1. **Move the MyTextFile to the HelloWorld directory**

**Text

Description automatically generated**

* + 1. **Change the file permissions on the file to rwx for owner, rw for group, and r for everyone.**

**Text

Description automatically generated**

* + 1. **Prove that the permissions are set with an ls command**

**Text

Description automatically generated**

* + 1. **Delete MyTextFile**

**A screenshot of a computer

Description automatically generated with medium confidence**

* + 1. **Delete the directory, HelloWorld**

**Graphical user interface, application

Description automatically generated**

* 1. **What are two ways, one command line and one graphical, that you can unzip a file in Ubuntu Linux?**
     1. **Unzip <filename>**
     2. **Right click file and click extract here**
  2. **What command is used to run privileged processes on Linux without compromising the root user?**
     1. **sudo**

1. We will be using Python 3.9.1 for this class. Again, you may feel rusty. We will review some Python basics over the next couple of weeks to prepare you for this class. If you are feeling uncertain about Python, you should spend some time reviewing over the next two weeks. It is a critical component of this class. You can use, <https://www.w3schools.com/python/>, as a reference for reviewing Python over the next couple of weeks. To begin the reviewing process, complete the following by **getting a sceenshot, answering the question, or both**:
   1. **Launch the Python 3.9.1 interpreter by typing *python3* from your CLI.**

**Text

Description automatically generated**

* 1. We can execute one line at a time in the interpreter. As you do the following, grab a screenshot of either each step or a collection of steps:
     1. **Assign your name to the variable, *name* (e.g., name = “YourName”).**

**A screenshot of a computer

Description automatically generated**

* + 1. **Use the print command to display your name, using the variable.**

**Text

Description automatically generated**

* + 1. **Assign your age to a variable called, *age*. Do not save it as a string. You can force it as an integer, for example as in: age = 25, not age = “25”**
       1. **Age = 23**
    2. **What Python type is your variable, *name*? Use the type() method to prove it.**
       1. **string**
    3. **What Python type is your variable, *age*? Use the type() method to prove it.**
       1. **Int**

Text

Description automatically generated

* + 1. **What are some things you can do to age if it is saved as a native number type like an integer that you cannot do to name, which should be a string, in a program or script?**
       1. **Multiply, divide, do arithmetic’s that you wouldn’t be able to do to a word.**
    2. **What are some things you can do to name that you cannot do to age in a script or program?**
       1. **Create output that is easier to read than a series of numbers, good for troubleshooting or just letting the user know what something is going to do**
    3. **Use the print command to print something like, “My name is Denny and my age is 55.”, where your name and age are retrieved using the variables and their values above. If you get an error with *age* when you concatenate the string, use the str(age) method to see if you can use *age* as a string in the concatenated string. Note that strings can be concatenated using +. See the website reference above for examples of concatenating strings.**

**Graphical user interface, text

Description automatically generated**

* + 1. Exit the interpreter, by typing, *quit()*.
    2. Make a directory in your home directory called, myPythonScripts
    3. Launch your IDLE3, a python Integrated Development Environment (IDE) by typing *idle-python3.9* from your CLI.
    4. An IDE helps a developer to create code easier. For example, type *in[tab]* (i.e., in + the tab key to tab complete), and see your options. Now try *inp[tab]*. The input command allows us to prompt the user and get information from the stdio (keyboard/monitor). Example:
       1. *variableName = input(“Enter a value: “)*
       2. The above line will display a prompt, Enter a value: , and wait for the user to type something and hit return. Whatever is typed will be stored in variableName.
       3. **Try it out by writing a short program that asks the user for their name and responds with a personal welcome that uses the user’s name. You just need a line for the input and a print line.**

**Graphical user interface, text, application, Word

Description automatically generated**

* + - 1. **Once you type the two line script, save it in your directory, myPythonScripts. Once you do this, you can run the module by choosing: Run🡪Run Module or hit F5. A Python Interpreter screen will open automatically to run your code. If you made a mistake, you can close the interpreter and go back to the IDLE3 screen to edit your code.**

**Graphical user interface, text, application, email

Description automatically generated**

1. **Organize all of your answers and screenshots, placing them with the questions above and submit to Blackboard by the due date. To receive full credit, you must complete the entire lab.**