LETTERKENNY INSTITUTE OF TECHNOLOGY

ASSIGNMENT COVER SHEET

Lecturer's Name: _____

Assessment Title: _____

Work to be submitted to: _____

Date for submission of work: _____

Place and time for submitting work: _____

To be completed by the Student

Student's Name: **Derek Troy**

Student Number: L00170167

Class: **BSC in Data Centre Management**

Subject Module: OOPR For Server Administration

Word Count (where applicable): _____

I confirm that the work submitted has been produced solely through my own efforts.

Student's signature:  Date: **23/11/2021**

Notes

Penalties: The total marks available for an assessment is reduced by 15% for work submitted up to one week late. The total marks available are reduced by 30% for work up to two weeks late. Assessment work received more than two weeks late will receive a mark of zero. [Incidents of alleged plagiarism and cheating are dealt with in accordance with the Institute's Assessment Regulations.]

Plagiarism: Presenting the ideas etc. of someone else without proper acknowledgement (see section L1 paragraph 8).

Cheating: The use of unauthorised material in a test, exam etc., unauthorised access to test matter, unauthorised collusion, dishonest behaviour in respect of assessments, and deliberate plagiarism (see section L1 paragraph 8).

Continuous Assessment: For students repeating an examination, marks awarded for continuous assessment, shall normally be carried forward from the original examination to the repeat examination.

Introduction

In this assignment we were tasked with 6 Individual Questions. In this conclusion section I will look at each question and try draw out what lessons and learnings I took from each component part. All code and screenshots are online @ https://github.com/derektroy/OOPR_Assignment_L00170167

Question 1

Question 1 was again a straightforward task, installing Apache2 on the VM we were using for this assignment. Logged into the VM under my own account, run the apt install under super user permissions and once that's complete allow the Apache through the Firewall, and Restart. I would conclude in this Question that running the command sudo ufw allow 'Apache Full' is not secure and best practices as this opens the firewall to all apache traffic across all ports. And best practise would suggest locking this down to number of known ports eg. 80, 443.

Question 2

Question 2 was a straight forward question. Once BeautifulSoup was installed, I also added the XML Parser to it as well, as I had concluded from the instruction "parse it minimally" that XML was the format was required.

I then created the code for pulling the headers from the HTML, in previous iterations of HTML Code you would have <h1> or <h2> tags in the code, but this time it was defined in CSS Code which was slightly harder to find.

I also added the code to search and count Apache2, I noticed that the code returned a case sensitive result.

Question 3

Question 3 was a clear-cut question, connect using Python to an SSH enabled Server. Using Paramkio to create an SSH Session, I just passed it the details of IP, Username and Password to connect. For Security purposes I would conclude that having all these details together in a single file is a risk. Passing those as inputs could be a better conclusion for security in this aspect.

Question 4

As this was an Ubuntu Desktop operating as a Webserver there wasn't a lot of ports open. I had opened RDP as I was connecting to it from a Windows desktop. I expected HTTP, SSH and RDP to be open. I added SMTP as a demonstration that all "pretty" ports weren't opened and listening. I can conclude that Ubuntu is secure operating system and doesn't have a lot of ports listening.

Question 5

This question follows on from Question 3, and again the same caveats apply, passing details in the Py File isn't best practise or secure. I wasn't 100% sure how to handle the additional command to display the last accessed command. But passing it back into a print statement to display the output I conclude was the best option to display it.

Question 6

This question on Terraform was quite interesting as it tied together everything we've been doing so far in this module and it demonstrated Infrastructure as Code. It concluded the assignment with a demonstrate of how code can be implemented to build, change, and remove servers.