**CYB6004 Assessment 4: Software Based Solution**

Ahmed Moussa

15/8/2024

Mentor / Tutor: Rajesh SONI

Student Number: 10682275

**Selection of Data**

This report is based on Bash Scripting using Regex (Regular Expressions) to Process, Download Contents, Filter, Search, Word Match and Edit a Website. We needed to search and choose a Website that contains useful information related to Cyber-Security. The Website chosen for this report was:

<https://www.digitalguardian.com/blog/what-cyber-security#Managing>

I found this Website having a huge amount of information about Cyber-Security, informative, high level of complexity and was easy for me to access Data from within.

I focused on three common keywords for analyzing / Processing Website. These words were “Skill”, “Protect” and “Career”. This included any Off-shoots of the word and regardless of capital or smaller letters .i.e. Skill, skill, skills, skilled, protection, protected..etc…

**Script Usability**

The Script was based o everything I learnt during week 1-5 within my Graduate Certificate in Cyber-Security at Edward Cowan University. May of the Command-line arguments and Script Codes were taken from week1-5. This included but not limited to IF, passwords, menu, file / folder making, calculating, BC, url-download, wget, number guessing game, AwkFilter, SED and GREP.

The Script Starts by asking user for an Access Password, which is hidden within a text file called password.txt. In the final version I also added Hash encryption for better protection.

Password is Figlet1!

(for Assessors Reference)

This secures the Script File from anyone to access it. If wrong then a message is printed with “Acess Denied”. If correct, then a “Cowsay” fancy message pops up and then it starts Hacking into / Downloading all the contents of Website URL (using wget command) while explaining to user Step by Step the Process.

url="<https://www.digitalguardian.com/blog/what-cyber-security#Managing>"

file\_path="$location/$(basename "$url").html"

wget -O "$file\_path" "$url"

> “url” was the name (variable) given to the website chosen.

> “location” was the directory or folder name chosen or created by user.

> “file\_path” is the name (variable) that defines the name and location of the saved the website.

> “basename” tells the script to save the file using the last part of the website url. In our website the last part is “what-cyber-security#Managing" and it is saving it as a “.html” file.

> “-o” in wget tells the script to over-write any existing file with the same name.

Once done downloading and saved to a folder of User’s choice, The Script then brings up a fancy Menu, making use of the commands Figlet. I have named this Script “WEBSITE HACKER”.

The User friendly Menu gives options to search the downloaded content using either three keywords and extracts all sentences containing these keywords. It then replaces these keywords with the word “Experience” and saves the website to your\_version.html. Menu also gives an option to play a Mathematical Calculator Game or to set a new Secure Password for future Access, otherwise to exit at any time.

**Text Processing**

The use of Regular Expressions to process downloaded data:

grep -in -E "${key}(s)?" "$file\_path" | nl

> “grep” searches the keywords in the downloaded document and prints out all the lines which contain those keywords.

> “key” is the user chosen keyword.

> “file\_path” is the name (variable) that defines the name and location of the saved the website.

> “-i” in grep tells to search for the keyword regardless of whether it is Capital letters or small letters (case-insensitive)

> “-n” references the line numbers that these sentences (with keywords) exist on the website. i.e. 1518<h3> What Skills do you need for Cyber Security? Refers to header 3 at line 1518 scrolling down the website.

> “-E” enables Extended Regular Expressions.

> “(s)?” enables pluralisation. Meaning any words that maybe in the plural form with an s at the end.

> “| nl” pipes the outputs of the grep command to Numbered Lines (nl). For every found result (if found) it will number it sequentially 1, 2, 3….

Then we use the Command “sed” to edit the newly saved html and replace all keywords withe the word “Experience”:

sed -i "s/$key/Experience/g" "$location/your\_version.html"

> “-i” edits the file without backup

> “s” means substitute / replace the keyword variable $key with the word Experience

> “g” is for searching globally over the document for all the occurrences of the keyword.

**Code Structure**

Made use of “functions” to structure the code. Having 4 main functions:

> Menu() was the main menu options page. Which was constantly called upon for user choices. It also allowed for Exiting program

> Regex() which searched and edited the keywords

> Calculate() which operated the Mathematic Calculator Game

> Passwordcheck() which allows user to change password to a “strong” password. It ensures the user chooses a secure password.

**Evaluation / Limitations**

I really made use of Kali-Linux for this project. My PC is running Windows, so I downloaded Kali-Linux to help write and Evaluate Script. Using Kali-Linux was a great experience and helped save time. Rather than the slower logging into Azure Labs.

Would have been better if used AwkFilter to have the data organized into tables with headers. Also could have used a more powerful scripting language like Python for more complex websites and better results. Great data processing and for better searching keywords,I could have used Perl as well.

Running out of time I kept testing and editing my script. I could have allowed for Users to type in their choice of keywords to search on the website. Also could have allowed Users to also input their choice of Word to substitute words with.

Maybe I could have also included the password within a function as well. Rather than a txt file, which can be easily found and hacked into. In final version of the Script I also added a Hash password encrypting system. I used the sha256sum to encrypt passwords.

Overall, I believe it was a greatly presented script with a lovely User Friendly Menu and detailed explanations for Users. Step by Step running through the processes with User.