Scripting Languages
Assignment 3- Software Based Solution

Contents

GENERAL REQUIREMENTS

- Your script will be marked on a standard Linux installation using the bash shell.
- Refrain from using *non-core* commands, tools and utilities in your bash shell scripts. Non-standard bash commands, tools and utilities will not be downloaded and installed by your tutor.
- Ensure each script you write is *fully self-contained* and is not configured to be dependent on external files, libraries or resources to run.
- Do **not** use the *trap* command in any of your scripts.
- Each script you submit must contain your full name and student number at the beginning as code comments.

IMPORTANT NOTES

- This is an **individual** assignment only and must **not** be completed in collaboration with other students
- You may **not** work with others, acquire code from others, or provide code to others
- Further, you may **not** post any of the assignment tasks below to a code-development community of any kind seeking solutions or advice
- Nor may you copy and paste or otherwise reproduce code provided by external sources and use it as part of your own solutions unless it is cited in-code according to APA 7 referencing guidelines. The more code that is externally sourced, the less original your own solution, the lower your potential mark for the given requirement(s)
- Where it is found that any of these instructions have been ignored, academic misconduct proceedings may be initiated
- Please read the checklist below and watch the associated video **BEFORE** submitting your assignment

ACADEMIC INTEGRITY TICK-BEFORE-SUBMIT CHECKLIST

PLAGIARISM

✓I have not copy and pasted from external sources without appropriate citation
✓My in-text and end-text citations follow APA 7 guidelines
✓I have not used my own or other student's previous assignment work

COLLUSION

✓I have not worked with any other students on this assignment unless permitted
✓My assignment is not based on or derived from the work of any other students
✓I have not shown or provided other student(s) with my assignment at any point

CONTRACT CHEATING

✓I have not asked or paid someone to do this assignment for me
✓I have not used any content from a "study notes" or "tutoring" service / website
✓I have not had a friend or family member assist me with this assignment

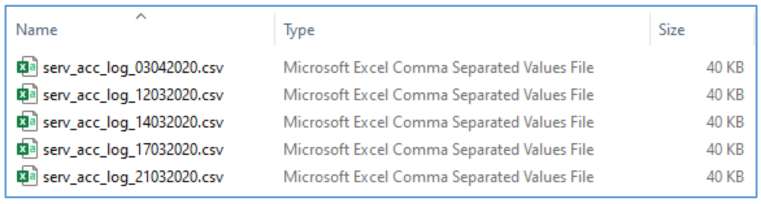
IF YOU ARE UNSURE ABOUT ANY OF THE ABOVE, DO NOT SUBMIT YOUR ASSIGNMENT BEFORE SPEAKING WITH YOUR UNIT COORDINATOR OR ECU LEARNING ADVISOR

Watch this video before submitting your assignment

Assignment Description

Having completed your two main portfolio activities, you are now required to further develop your shell scripting skills by developing a script that automates a task commonly performed by Linux administrators - the analysis of server access logs to identify and report upon suspicious activity.

To develop and test your script, you have been provided with a set of five (5) server access logs in a zip folder named *serv_acc.zip*.



Name	Type	Size
serv_acc_log_03042020.csv	Microsoft Excel Comma Separated Values File	40 KB
serv_acc_log_12032020.csv	Microsoft Excel Comma Separated Values File	40 KB
serv_acc_log_14032020.csv	Microsoft Excel Comma Separated Values File	40 KB
serv_acc_log_17032020.csv	Microsoft Excel Comma Separated Values File	40 KB
serv_acc_log_21032020.csv	Microsoft Excel Comma Separated Values File	40 KB

*Please note: Your tutor will run your script in the same folder as the server access logs located on his/her computer, so you do **not** have to prompt the user for their location.*

Each server access log contains 500 records organised into the following columns:

DATE	Not required for the assignment
DURATION	Not required for the assignment
PROTOCOL	TCP, UDP, ICMP, GRE
SRC IP	Various codes
SRC PORT	Port *from* which incoming packets have been sent
DEST IP	Various codes
DEST PORT	Port *to* which incoming packets have been sent
PACKETS	Number of packets sent in a transfer
BYTES	Size of packets sent in a transfer
FLOWS	Not required for the assignment
FLAGS	Not required for the assignment
TOS	Not required for the assignment
CLASS	*suspicious* or *normal*

Please note: Fields in grey will not be used in the assignment.

Part 1 – Write the Code (Shell Script, 30 marks)

Basic Functional Requirements (10 marks)

Your server access log must provide the user with **all** the following functionality:

1. Run a search on *one (1)* server access log of the user's choosing based on *one (1)* field criteria input, also of the user's choosing, e.g. PROTOCOL=`TCP`
2. The results of each search the user conducts are to be displayed to the terminal **and** also exported to a *.csv* file with a name of the user's choosing. Each results file created must be uniquely named so that the results files of previous searches are **not** overwritten
3. Any log file records in which the CLASS field is set to *normal* are to be automatically excluded from the search results printed to the screen/written to file

4. When the PACKETS **and/or** BYTES fields are selected by the user as search criteria, the user should be able to choose *greater than* **(-gt)**, *less than* **(-lt)**, *equal to* **(-eq)** or *not equal to* **!(-eq)** the specific value they provide, e.g. find all matches where `PACKETS > `10``
5. When the SRC IP or DEST IP fields are used as search criteria, the user should only need provide a *partial* search string rather than a complete value, e.g. search using the partial string EXT rather than the exact value EXT_SERVER

Advanced Functional Requirements (10 marks)

Implement ***two (2)*** of the following advanced functionalities:

1. Enable the log tool script to run searches on a *single* server access log of the user's choice using both two (2) and *three (3)* field criteria inputs, e.g. find all matches where PROTOCOL=`TCP` **and** SRC IP=`ext` **and** PACKETS > `10`
2. Enable the log tool script to run searches on *all* available server access logs based on *one (1)* field criteria input, e.g., find all matches where PROTOCOL=`TCP` in all available log files
3. When the PACKETS **and/or** BYTES fields are used as search criteria, **totals** for each of these should also be calculated and displayed as the final row of the search results printed to terminal/file

Note: Please ensure that the enhanced functionalities you choose to implement are clearly identified in your code (using comments) and clearly addressed in your video demonstration.

Usability, Reliability and Efficiency Requirements (10 marks)

1. All string-based searches should be *case insensitive*.
2. The results of any search are to be printed to terminal/file in a **columnar** format, uniformly aligned and spaced.
3. All user inputs are to be fully validated and sanitised as required to ensure the correct execution of subsequent code.
4. The script is to display a high level of *abstraction*,.i.e., the hard-coding of values is to be avoided.
5. The **efficiency** of your code will also be considered, hence the degree of thought you apply to the selection of and interaction between shell script elements such as *logical tests*, *control structures (if-elif-fi, loops, arrays)*, *functions*, *command substitution*, *regular expressions*, *piping*, *redirection* and *utilities*, e.g. awk, is important.
6. The user must be able to conduct as many search operations as they wish without the script terminating. Hence, the script must continue to run until the user *specifically* chooses to terminate it via a menu option.
7. All menus, options and prompts are to be easily understood and require minimal input from the user in response.
8. Sound code structure and full commenting will be examined by your tutor and factor into your grade.

Part 2 – Explain Your Work (Video, 10 Marks)

Required Video Elements

Record a video using Panopto that fulfills the following criteria:

- ☐ Begin with you appearing on-screen displaying your Student ID card and verbally stating your full name and student number
- ☐ A full run-through of your code demonstrating **Basic Functional Requirements 1** through **5** in action
- ☐ A full run-through of your code demonstrating the *two (2)* **Advanced Functional Requirements** in action you chose to implement
- ☐ Explain how you have addressed **Usability, Reliability and Efficiency Requirements 1** through **8**, pointing to specific example(s) in the code and code output in each case.
- ☐ Change one element of code, e.g., such as a message displayed when invalid input is provided and then re-run the code to show the change in action.
- ☐ If there were any requirements you were **not** able to implement, the please identify these and briefly explain the reasons why this was the case, e.g. ran out of time, couldn't figure out the code required, etc.
- ☐ Both the video and audio elements of your presentation should be of good quality.
- ☐ Your video must **not** be more than 8 minutes long, i.e. your tutor will stop viewing your presentation at the 8 minute mark and anything thereafter will **not** factor into your grade.

Marking Key

Criteria	NOT DEMONSTRATED → DEMONSTRATED					
BASIC FUNCTIONAL REQUIREMENTS (10 Marks)						
Run a search on *one (1)* server access log of the user's choosing based on *one (1)* field criteria input, also of the user's choosing, e.g. PROTOCOL=\`TCP\`	0	1	2	3		
The results of each search the user conducts are to be displayed to the terminal **and** also exported to a *.csv* file with a name of the user's choosing. Each results file created <u>must</u> be uniquely named so that the results files of previous searches are **not** overwritten	0	1	2	3		
Any log file records in which the CLASS field is set to *normal* are to be automatically excluded from the search results printed to the screen/written to file	0	1				
When the PACKETS **and/or** BYTES fields are selected by the user as search criteria, the user should be able to choose *greater than* **(-gt)**, *less than* **(-lt)**, *equal to* **(-eq)** or *not equal to* **!(-eq)** the specific value they provide, e.g. find all matches where PACKETS > \`10\`	0	1				
When the SRC IP or DEST IP fields are used as search criteria, the user should only need provide a *partial* search string rather than a complete value, e.g. search using the partial string EXT rather than the exact value EXT_SERVER	0	1	2			
ADVANCED FUNCTIONAL REQUIREMENTS (10 Marks)						
Selected advanced functional requirement 1	0	1	2	3	4	5
Selected advanced functional requirement 2	0	1	2	3	4	5
USABILITY, RELIABILITY AND EFFICIENCY REQUIREMENTS (10 Marks)						
All string-based searches should be case insensitive.	0	0.5	1			
The results of any search are to be printed to the screen in a columnar format, uniformly aligned and spaced.	0	0.5	1			
All user inputs are to be fully validated and sanitised as required to ensure the correct execution of subsequent code.	0	0.5	1	1.5		
The script is to display a high level of abstraction, .i.e. the hard-coding of values is to be avoided.	0	0.5	1	1.5		
The efficiency of your code will also be considered, hence the degree of thought you apply to the selection of and interaction between shell script elements such as logical tests, control structures (if-elif-fi, loops, arrays), functions, command substitution, regular expressions, piping, redirection and utilities, e.g. awk, is important.	0	0.5	1	1.5		
The user must be able to conduct as many search operations as they wish without the script terminating. Hence, the script must continue to run until the user specifically chooses to terminate it via a menu option.	0	0.5	1	1.5		
All menus, options and prompts are to be easily understood and require minimal input from the user in response.	0	0.5	1			
Sound code structure and full commenting will be examined by your tutor and factor into your grade.	0	0.5	1			
EXPLAINER VIDEO (10 Marks)						
Video begins with you appearing on-screen displaying your Student ID card and verbally stating your full name and student number.	0	0.5	1			

A full run-through of your code demonstrating **Basic Functional Requirements 1** through **5** in action.	0	1	2	
A full run-through of your code demonstrating the *two (2)* **Advanced Functional Requirements** in action you chose to implement.	0	1	2	
Explain how you have addressed **Usability, Reliability and Efficiency Requirements 1** through **8**, pointing to specific example(s) in the code and code output in each case.	0	1	2	
Change one element of code, e.g., such as a message displayed when invalid input is provided and then re-run the code to show the change in action.	0	0.5	1	
Explanation of difficulties and challenges experienced during the development of the log analysis tool.	0	0.5	1	
Both the video and audio elements of your presentation should be of good quality.	0	0.5	1	
TOTAL SCORE:		**/40**		

How to submit Assignment 3 to Canvas

- Submit the single shell script (bash) file you have created in a **zipped folder** with the following naming format - `[surname]_[student-ID]_CSP2101_ASS3.zip` into the file upload box provided
- Place the URL of your completed Panopto video into the text box provided

END OF ASSIGNMENT BRIEF