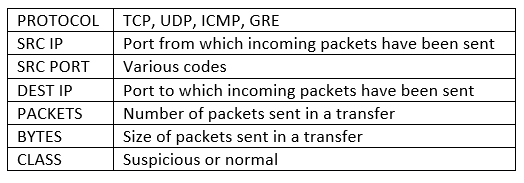
This script is about automating a task commonly performed by Linux administrators that analysis server access logs to identify and report upon suspicious activity. To develop this script, you will be provided 5 server access logs.

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



|  |  |
| --- | --- |
| No. | Requirements |
| 1. | Run a search on available server access logs based on either 1, 2 and 3 field criteria inputs.  Prompt user for how many inputs they want to run on a search and show a list of given field criteria to the user so that they can choose which field criteria they want to run a search on. Basically, the user can choose to run a search on the server access logs based on 1 input, 2 inputs or 3 inputs on the field criteria that they have chosen.  Also validate the user inputs and field criteria inputs. |
| 2. | Give the user the option to search all server access logs available in a directory or search just one specific log of the user’s choice |
| 3. | Export the results of any search to a text file and destination directory of the user’s choosing. If the file and destination directory nominated by the user are non-existent, the script will create them. |
| 4. | Any records in which the CLASS field is normal are to be automatically excluded from the search results printed to the screen but display search results for CLASS field that are suspicious on the screen. |
| 5. | When the PACKETS and/or BYTES field 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 the screen. |
| 6. | When the PACKETS and/or BYTES fields are used as search criteria, the user should be able to choose greater than, less than, equals to or not equal to the specific value they provide. |
| 7. | When 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. -> search using the partial string EXT rather than the exact value EXT\_SERVER. |
| 8. | All string-based searches should be case insensitive |
| 9. | The results of any search are to be printed to the screen in a columnar format, uniformly aligned and spaced. |
| 10. | All user inputs are to be fully validated and sanitised as required to ensure the correct execution of subsequent code. |
| 11. | 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. |
| 12. | All menus, options and prompts are to be easily understood and require minimal input from the user in response. |
| 13. | The efficiency of the code must be applied to the selection of and interaction between shell script elements such as **logical test, control structures, functions, command substitution, regular expressions, piping, redirection, AWK, SED and utilities.** |
| 14. | The script must be fully commented |
| 15. | Hard-coding of values is to be avoided. |
| 16. | Do not use non-standard bash commands, tools and utilities. |
| 17. | Ensure the script is fully self-contained and is not configured to be dependent on external files, libraries, or resources to run. |
| 18 | Do not use trap commands |
| 19. | Prompting the user for the server access logs file is not needed |

**List of commands/functions to use:**

Grep and Regex  
Anchors and Wildcards  
Extended RegEx Engine  
ERE Repetition and Optionality  
OR and Expression Grouping  
Piping and Redirection

For Loops

IFS  
C-style Loops

arrays  
While Loops  
Until Loops  
Break and Continue  
Nested Loops  
Infinite Loops

Wget and curl

Awk conditional statement

Awk Functions:

Normal functions

Sed commands