Name: Zac Fawcett
Date: November 20th, 2021
Submission: Py-INI Justification Document

CPRG 260 Project #4

Script with variable configuration from Ini File

Our proposal is the following:

Too much of our time and resources are being spent writing individual scripts to view information on Linux workstations. What management is asking for is a script and accompanying configuration file. The script will be able to easily call the variables within the config file to initially be able to view system accounts and the groups these accounts belong to, along with log file entries corresponding to CRON processes. Then when the user requests, these reports will be printed to a file (also specified in the config file). Keep in mind that this is just an initial script and the entire idea is to have the whole thing be totally customizable. A simple change in the config file should elicit the corresponding change in the output of the script.

Function #1

- Using the confObj reader we can parse the INI file into separate variables and then add those variables to a dictionary for calling later. As long as we use section headers in the INI file we can easily add to and change the entries in the INI file. With this function we can call the dictionary at any time and using dictionary.get('variable') we can call the needed piece for comparison or adding to a command.

Function #2

- After calling the dictionary from the first function we can name variables that we need. In this case we used:
    - **sortAccountsReverse** = a boolean value that will allow us to print the groups and accounts in alphabetical order or in reverse. Can be switched in INI.
    - **sortAccountsCriteria** = Allows us to call only entries related to a specific criteria like the name of an account or group ID etc.

- **linesOfAccountsData** = This variable will allow us to decide how much data is printed from the output. We can choose a number or write All for everything allowing for complete control and customization of amount of data. This means we can fin-tune resources spent.

- After naming our variables we can add an attractive title and format the output to list neatly and attractively. Then we'll print this to the screen.

Function #3

- After calling the dictionary from the first function we can name variables that we need. In this case we used:
  - **sortLogsReverse** = a boolean value that will allow us to print the logs in forwards or in reverse. Can be switched in INI.
  - **logTimeFrom** = Allows us to set a date so that only logs from after that date will be printed. Allows for pinpointing a spot in time for troubleshooting and administration
  - **logTimeTo** = Allows us to set a date so that only logs from before that date will be printed. When used in conjunction with logTimeFrom, we can create a date range and only get entries from within that range.
  - **logCriteria** = Allows us to view only log entries that hold the criteria we specify. This allows for total customization of the output. We could specify and filename, process, user etc, and only get log entries pertaining to those criterion.
- After naming our variables we can add an attractive title and format the output, adding the criteria we used along with a date range used to search.

Function #4

- This function will take the print statements that functions 2 and 3 create and redirect them to the output file specified in the INI file.
- Also added will be a header containing the machine name of the system using the program, and the date and time the report is created.

Name: Zac Fawcett
Date: November 20th, 2021
Submission: Py-INI Justification Document

The options (print accounts/groups, print logs and write report will all be housed in a while loop menu. Options will be entered by the user and they can press enter to quit the program at any time. Overall this script is useful because of its ease of adaptation. Without changing the script we can easily fine tune the output by editing the INI file. Furthermore with the code included in the script we could easily refine it further by duplication certain comparison and search lines that would allow for the entering of additional INI options to laser focus searched by more criteria.

With this program we can greatly reduce the use of administrators' time and energy by allowing for even new hires to efficiently manipulate this scripts output through the use of changing the INI entries.