Exercise 06 - Parsing Log Files - Script 5

Goal:

The goal of this exercise is to create a shell script that displays the number of failed login attempts by IP address and location.

Scenario:

One day you received a call about a user being locked out of their account. Being the awesome sysadmin that you are, you decided to look at the log files to see why this person's account was locked. While doing so, you happened to notice ~~hundreds~~ thousands of failed login attempts!

You decide you need a way to quickly summarize the failed login attempts. That way you can quickly decide if an IP address needs to blocked.

Shell Script Requirements:

You think about what the shell script must do and how you would like it operate. You come up with the following list.

The script:

- Is named "`show-attackers.sh`".
- Requires that a file is provided as an argument. If a file is not provided or it cannot be read, then the script will display an error message and exit with a status of 1.
- Counts the number of failed login attempts by IP address. If there are any IP addresses with more than 10 failed login attempts, the number of attempts made, the IP address from which those attempts were made, and the location of the IP address will be displayed.
    - Hint: use the `geoiplookup` command to find the location of the IP address.
- Produces output in CSV (comma-separated values) format with a header of "Count,IP,Location".

Start the Virtual Machine and Log into It:

In a previous exercise you created a vagrant project called localusers. Use the VM created in that project for this exercise.

First, start a command line session on your local machine. Next, move into the working folder you created for this course.

```
cd shellclass
```

Change into the localusers directory, start the virtual machine with "vagrant up", and then connect to it with "vagrant ssh".

```
cd localusers
vagrant up
vagrant ssh
```

Navigate to the `/vagrant` Directory

```
cd /vagrant
```

Write the Shell Script

At this point, you can either create the script inside the virtual machine using the `vim`, `nano`, or `emacs` text editors or you can create the file using your favorite text editor on your local operating system. (Atom from https://atom.io/ is a good choice.)

When creating your script, refer back to the shell script requirements. If you want or need more detailed steps to help you write your script, refer to the pseudocode at the end of this document. It was intentionally placed at the end of the document because I want to encourage you to write the script on your own. It's fine if you need the pseudocode. As you get more scripting practice, you'll be able to script without any additional aids.

Test Your Script

Once you've finished writing the script, use it to parse the "`syslog-sample`" file provided.

Remember that the first time you execute the script you'll need to make sure it has executable permissions.

```
chmod 755 show-attackers.sh
```

Here is an example run of the script. (Portions typed are in bold.)

```
./show-attackers.sh syslog-sample
Count,IP,Location
6749,182.100.67.59,China
3379,183.3.202.111,China
3085,218.25.208.92,China
142,41.223.57.47,Kenya
87,195.154.49.74,France
57,180.128.252.1,Thailand
27,208.109.54.40,United States
20,159.122.220.20,United States
```

Make sure that if a file that doesn't exist is provided, that the script exits with an exit status of 1.

```
./show-attackers.sh /path/to/nowhere
Cannot open log file: /path/to/nowhere
echo ${?}
1
```

Make sure that if a file isn't provided as an argument the script exits with an exit status of 1.

```
./show-attackers.sh
Cannot open log file:
echo ${?}
1
```

Reference Material:

Vagrantfile for localusers

Here are the contents of the `shellclass/localusers/Vagrantfile` file with all the comments removed.

```
Vagrant.configure(2) do |config|
  config.vm.box = "jasonc/centos7"
  config.vm.hostname = "localusers"
end
```

Pseudocode

You can use the following pseudocode to help you with the logic and flow of your script.

```
# Make sure a file was supplied as an argument.

# Display the CSV header.

# Loop through the list of failed attempts and corresponding IP
addresses.

  # If the number of failed attempts is greater than the limit, display
count, IP, and location.
```