CS 3030 Scripting Languages
Lab 3: Failed Login Report

Introduction

Lab 3 is an opportunity to use UNIX/Linux commandline tools `sed, sort, uniq and printf` to search a logfile using regular expressions, count the number of occurrences and print an simple report in HTML format.

Requirements

Your script shall be named `~/cs3030/lab3/flog` and be marked executable. Here is how you execute your script:

```
./flog LOGFILE
```

Your script shall search LOGFILE for all lines containing this string:

```
“Failed password for USERID”
```

where USERID is either the name of a user or the word `“invalid”`. You are to perform the following operations:

- Change the word “invalid” to <UNKNOWN>
- Count the number of every failed password messages for each user (including <UNKNOWN>)
- Sort the output first by number of occurrences in descending order, then by userid in ascending order
- Output a header with the current date as shown below
- Punctuate all large numbers with commas for readability (hint: `printf`)
- Produce an HTML document on stdout with <html>, <body>, <h1> and
 tags as shown below
- “<UNKNOWN>” should be output using the HTML characters “<” and “>”
- Print `Usage: flog LOGFILE` and exit with return code 1 if the user does not specify a LOGFILE

When your script is working, test it with the provided sample logfile:

```
./flog /var/classes/cs3030/lab3/secure
```

Your output should look like this:

CS 3030 Scripting Languages
Lab 3: Failed Login Report

```
<html>
<body><h1>Failed Login Attempts Report as of Thu Dec 25 07:00:00 MST 2014</h1>
<br />   1,325 &lt;UNKNOWN&gt;
<br />    505 root
<br />     17 adm
<br />     16 ftp
<br />     10 apache
<br />     10 mail
<br />      2 backuppc
<br />      2 games
<br />      2 news
<br />      2 squid
<br />      1 bin
<br />      1 nobody
<br />      1 operator
<br />      1 smmsp
<br />      1 sshd</body> </html>
```

Now, to actually see your HTML rendered as a web page in a browser, redirect your HTML output to your public_html folder. Then `chmod` your html file so that it is readable by the Apache web server on icarus:

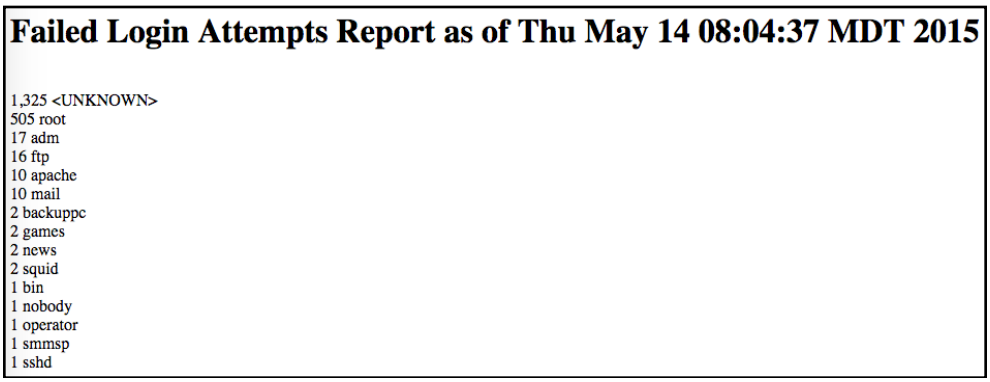
```
./flog /var/classes/cs3030/lab3/secure >~/public_html/flog.html

chmod 644 ~/public_html/flog.html
```

Open your favorite web browser and go to this URL (replacing USERID with your icarus userid):

```
icarus.cs.weber.edu/~USERID/flog.html
```

Here is an example of what you should see:



CS 3030 Scripting Languages
Lab 3: Failed Login Report

Hints

- You are free to use whatever UNIX/Linux commandline tools that are available in order to produce the requested output.
- You can count and summarize each occurrence using `uniq`
- `sort` can sort numbers with commas imbedded in them and can sort in descending sequence by multiple fields
- The header for the report should be output with an <h1> HTML tag.
- Your HTML should look like the example above, without any extra HTML tags. A single
 should appear before each line of output after the header line.
- Output the current date in the report header as produced by the `date` utility.
- Use whitespace and indenting to make your script readable
- Add comments to your script to document your logic

Run cucumber 2-3 times to determine your grade

```
tar xvf /var/classes/cs3030/lab3/cuke.tar

./cucumber -s
```

The cucumber script creates a new random logfile for each scenario. This is helpful for finding obscure bugs in your code. It also allows for partial credit (students tend to like partial credit).

Files

For this lab you will have created folder `lab3` and the following executable files:

`flog`

CS 3030 Scripting Languages
Lab 3: Failed Login Report

Grading

Here is how you earn points for this assignment:

FEATURES	**POINTS**
Must-Have Features	
Script is named correctly and found in its proper place on icarus	10
Script is executable	10
Required Features	
Exit code is zero for normal execution	10
Exit cod is 1 for abnormal execution	10
A “Usage” statement is printed if the PATH is not specified on the commandine	10
Report header is in the correct format	20
Output is in HTML format	30
<UNKNOWN> user should appear just below the header	40
Output is correct: all users are counted, counts are correct and is sorted properly	30
Large numbers should contain commas	30
Grand Total	200