DV1457 Programming in UNIX Environments

Assignment 1310
(2.5 ETCS, A-F)

Deadline 1	September 20th, 2024 18:00
Deadline 2	November 22nd, 2024 18:00
Deadline 3	January 10th, 2025 18:00
Submission	ZIP file uploaded to Canvas

---

1 Description

This assignment tasks students with writing a script that processes a web server log file, extracting and summarizing specific information related to the server activity. This task is a rather common one within the system administration of UNIX and GNU/Linux.

The log file that is provided to students, emulates the output produced by a thttpd web server. It follows a standard format that is common in servers running UNIX and GNU/Linux: a flat text file with one record per line, and fields delimited by a whitespace or other characters.

This assignment assesses students' understanding and domain over shell scripting using bash, as well as their capabilities of designing and implementing computational solutions based on that technology.

2 Instructions

Your task is to write a script, using bash, that extracts and summarizes relevant information about the operation of a thttpd web server, so any anomalies can be easily identified. This approach is, even if simple, a common form of intrusion detection.

The web server log file (named thttpd.log) consists of lines, and each line represents one access request. This formatting emulates the unified Apache combined log format (see the Apache documentation for more details). The fields in the log file describe the following:

1. The IP number the request originated from.
2. The ident answer from the originator (always '-').
3. The username of the requester, as determined by http authentication.
4. The date and time the request was processed.
5. The request line as it was received from the client, in double quotes.
6. The http status code that was sent to the client.
7. The number of bytes that were transferred to the client.
8. The referrer page (from the client).
9. The user agent string (from the client).

Note that the fields from the web server log file are not well delimited (i.e. there is no reserved character to separate the fields), which is, unfortunately, rather common when it comes to log files and a problem you have to contend with.

By accessing the information in the web server log file, your script (which must be named log_sum.sh) should be able to answer the following questions:

- Which IP addresses makes the most number of connection attempts
- Which IP addresses makes the most number of successful connection attempts?
- What are the most common result codes, and where do they come from? (i.e. from which IP number).
- What are the most common result codes that indicate failure (i.e. no authentication, not found, etc.) and where do they come from?
- Which IP number get the most bytes sent to them?

To address these questions, the results generated by your script should be in the form of a sorted list, with the largest number first (i.e. topmost). The user should be able to limit the amount of results provided by the script (i.e. by asking for only the 'x' topmost results). When no limit is provided, all available answers should be presented.

Your script must support the use of specific arguments, that should be coded as:

```
log_sum.sh [-L N] (-c|-2|-r|-F|-t) <filename>
```

Where:

- -L: Limit the number of results to N.
- -c: Provides the IP addresses that makes the most number of connection attempts.
- -2: Provides the IP addresses that makes the most number of successful attempts.
- -r: Provides the most common result codes, and where they come from.
- -F: Provides the most common result codes that indicate failure and where they come from.
- -t: Provides the IP numbers that get the most bytes sent to them.
- *filename* refers to the web server log file (i.e. thttpd.log).

From the aforementioned arguments, | denotes choice (i.e. only one of these options should be given at a time). Furthermore, all arguments within the square brackets are optional, while arguments inside the parenthesis are mandatory. For example,"[-L N]" means that the argument -L may or may not be given, while "(-c|-2|-r|-F|-t)" indicates that exactly one of the arguments -c, -2, -r, etc. must be given. If a mandatory argument is not given, this is an error condition.

The result output formating should be:
-c: xxx.xxx.xxx.xxx yyy where yyy is the number of connection attempts
-2: xxx.xxx.xxx.xxx yyy where yyy is the number of successful attempts
-r: yyy xxx.xxx.xxx.xxx where yyy is the result code, one ip per line
-F: yyy xxx.xxx.xxx.xxx where yyy is the result code indicating failure
-t: xxx.xxx.xxx.xxx yyy where yyy is the number of bytes sent from the server

xxx.xxx.xxx.xxx is the IP address in dotted quad format. yyy is an integer between 1 and the required number of digits (i.e. not necessarily just three digits). You are allowed to insert tabs and or whitespace between the elements

of the output according to your taste. However, do not add them within the elements (e.g. within an IP address). For -F and -r you are allowed to output multiple lines with the same result code or count, but the groups must still be sorted. There should still only be one IP address per line.

2.1 Examples

2.1.1 Example 1

```
$log_sum.sh –L 10 –c thttpd.log
213.64.237.230   2438
213.64.225.123   1202
213.64.141.89    731
213.64.64.53     591
213.64.214.124   480
213.64.55.182    429
213.64.246.37    400
213.64.153.92    336
213.64.100.52    336
213.64.19.224    272
```

In this example, we are asking for a sorted list of the top ten most connection-intensive IP addresses, most prolific first. This example can be used as a test case, as it is correct given the provided thttpd.log file.

2.1.2 Example 2

```
$log_sum.sh –L 3 –r thttpd.log
404 127.0.0.1
404 xxx.xxx.xxx.xxx
404 xxx.xxx.xxx.xxx

200 xxx.xxx.xxx.xxx
200 xxx.xxx.xxx.xxx
200 127.0.0.1

403 xxx.xxx.xxx.xxx
403 xxx.xxx.xxx.xxx
```

Example 2 is a fictitious case. Here, the status code groups have to be sorted by their frequency (i.e. which status code group appears more often), and the IP addresses have to be sorted by the number of occurrences within each of the groups. Furthermore, IP addresses may show up several times in the output for different status codes. However, within each status code group, an IP address should be presented only once. Note that these formatting rules also apply to the output of -F.

2.1.3 Example 3

```
$log_sum.sh -L 3 -t thttpd.log
xxx.xxx.xxx.xxx 19109572
xxx.xxx.xxx.xxx 18043610
xxx.xxx.xxx.xxx 1915720
```

Example 3 is also a fictitious case. The results for -t have to be sorted based on the sum of bytes.

Students are free to design any kind of architecture for the implementation of this assignment. Similarly, students are free to use any terminal command or terminal tool available by default in GNU/Linux (e.g. grep, awk, getops, etc.). Despite this, the script submitted by the students, and the processing of the web server log file, must be implemented using bash. Students are not allowed to modify the web server log file using any other external tools (e.g. Excel, Python scripts, etc.).

3 Requirements and Assessment

The development of this assignment is done in groups of 3 students and, despite the desired grade, every submission must:

- Be implemented using the bash shell.
- Present the solution in a bash script named log_sum.sh.
- Comply with all specifications described in Section 2.
- Include a .txt file with the students' names, the desired grade, which arguments are supported by the solution, and a brief description of the approach used to implement it.
- Include your script, the .txt file with students' information, and the web server log file, within in a single .zip file.

Each supported argument that outputs a correct result will award students with one point. The grade for this assignment is determined by the final number of points achieved by the students, based on the grading criteria shown in Table 1.

If the result produced by an argument is incorrect, incomplete (i.e. it lacks information), or does not comply with the formatting shown in the provided examples, no point will be awarded for it. Also, in accordance with BTH regulations, submissions graded with Fx can only be complemented up to E grade.

Table 1: Grading criteria for Assignment 1310.

Points	Grade
1	E
2	D
3	C
4	B
5	A

Submissions will only be accepted through Canvas and within deadline limits. Lastly, plagiarism is prohibited and any found case of plagiarism will have disciplinary actions taken against the students involved.

Developed by Stefan Axelsson.
Modified by Martin Boldt, Diego Navarro.
2024.