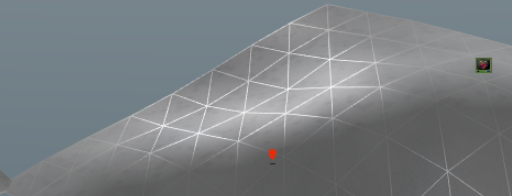
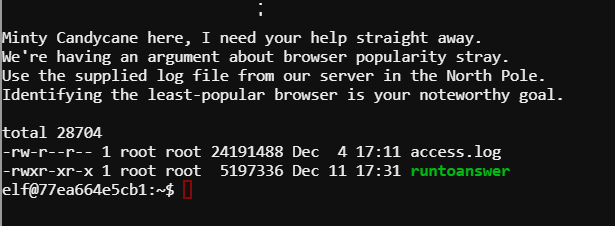
# Bumble’s Bounce Challenge, or Web Log Terminal Challenge

Both system administrators and security practitioners need to be able to parse log files quickly. Since log files are usually large (easily Gigabytes long), being able to use simple commands to weed through the files quickly is an important skill. Unix/Linux administrators have developed several useful tools for examining large text files.

The terminal is in the far right of the game.



Minty Candycane needs to determine the \*least\* popular browser that contacted her webserver. The access.log file is a standard Apache access log.



# Hints

This challenge is like a Linux lab from the first semester. It can be solved using cut, uniq, and sort.

The less command is not available, but more is. It shows the format is:

XX.YY.11.135 - - [19/Nov/2017:08:56:36 -0500] "GET /js/lib/log.js HTTP/1.1" 304 0 "https://www.northpolechristmastown.com/" "Mozilla/5.0 (X11; Linux x86\_64; rv:57.0) Gecko

/20100101 Firefox/57.0"

There are many ways to attack this problem, and you are free to use your own judgement in how to do so. If you are unsure, this method should lead you to a solution. Use cut, with quote (“) as a delimiter, to get to this:

Mozilla/5.0 (X11; Linux x86\_64; rv:57.0) Gecko /20100101 Firefox/57.0

Then cut with ) as a delimiter to get

Gecko /20100101 Firefox/57.0

Finally, cut with / as a delimiter to get

Gecko.

Use uniq to remove duplicates, and count the number of occurrences with the -c switch. But, before using uniq, remember that the list first has to be sorted with sort. Finally, you can sort by the numbers that the uniq -c command gives you.

Pipe all your cuts, sorts and uniqs together and you can get the answer in a one-liner!

Note: A couple of lines (Plus; Android 7.0 and ; Android 7.0) snuck through my filter. Also, curl and masscan are not browsers.

# Questions

1. Submit the command you used to find the answer.