DV1457 Lab 1 by Anton Lindstrand and Anton Myrberg

Problems encountered and how we solved them:

To solve the different tasks of the assignment, we quickly realized that we could not “program” it like we usually do in C or C++, and instead we had to run command chains to reach the desired result. An example of this is where we count the number of connection attempts and sort them from most to least connection attempts:

```
output=`awk '{ print $1 }' $filename\
    | sort -V\
    | uniq -c -d\
    | sort -nr\
    | awk '{print $2, "\t", $1}'`
```

Here we don’t directly use the language capabilities to solve the problem, but instead we use several commands outside of the shell language after one another to achieve the desired result.

The biggest problem we encountered was when we started implementing the “-e” flag. The premise was that we had the correct output from all the other flags so we wanted to add the “*Blacklisted!*” column to those lines that matched any of the blacklisted domains. To do this, we tried to read the multiline variable line by line, except in posix shell you cannot read a multiline variable line by line. The workaround was that we read the correctly formatted output into a temporary file, then used `while read line; do` to read the file line by line, which had the same result in the end.

Commands used and why:

The commands we used in the assignment were mainly grep, awk, sort, uniq and echo.

- grep - For extracting phrases from a line
- awk - For extracting certain columns of a line
- sort - For sorting the lines
- uniq - For counting and making addresses unique
- echo - For outputting results and to declare and redeclare variables