Lab Report1

Group Member:

Name	E-Mail	P-Num
Wei Shuang	shuang.wei83@gmail.com	8306193627
Ji Yuan	stevedjyz@gmail.com	9312059059

Q1: Flow control and structure design

We first analyze if the command format which user types in is correct. If it's correct, we will check which type of command it is, 4 arguments or 2 arguments. If it's 4, we handle the commands with line limitation. In print_log function, we used "case in" to get the result accordingly. If it's 2, we still handle it by print_log, just the line number of the object file will be as the line parameter. Besides, using pattern in "if" and parameter transmission among functions exhausted us much.

Q2: Lines limitation

For simplifying the process, we always give the number of line to print_log function. When the user input limitation number, we transfer the number to print_log, otherwise, we provide whole analyzed object's length by default. Since all result will be less than the whole object's lines number.

Q3: Which IP addresses makes the most number of connection attempts?

This is our first actual difficulty. How to check how many IPs appear in the log file and sum them up then compare the total amount of them? Follow this idea, we found awk can be used and implement this very easy. Fortunately, this idea can be also continue applied by the following functions.

Q4: Which IP addresses makes the most number of successful connection attempts?

According to CLF, the successful connection code is 2xx, so we just find all code which are 2xx, then sum up all IPs whose code is 2xx by using awk. When printing out the result, we wanted the columns can be aligned. This spend us some time.

Q5: What are the most common result codes and where do they come from (IP number)?

Firstly, list and sequence all result code without duplication. Then match result code in file with outputting its same line IP and then added to temp file. Here we need a null temperate file to list the required amount of lines by "/dev/null". Use a "sort" before "uniq" to firstly put same line together because "uniq" only counts successive same lines.

Q6: What are the most common result codes that indicate failure (no auth, not found etc) and where do they come from?

This is implemented in same way as feature -r. But we set a "if [$loop -lt 200 -o $loop -gt 399]" to ensure result code in 3XX or 2XX.

Q7: Which IP number get the most bytes sent to them?

Same way as "-c" but self increase by the byte num instead of one and finally it works magically. But to sort the 2nd column exhausted us.

Q8: (-e)Compare DNS in file to dns blacklist and if same DNS available, output its IP.

Here, as we need to sort string instead of number, it cannot be implemented in same way above. Thus, we use awk to list not duplicated addresses and then match IP with blacklist file. But so far, we can only match whether the DNS name we have in log file with the ones in blacklist file then print it out.