EECS 293
Software Craftsmanship
2016 Fall Semester

Programming Assignment 9

Due at the beginning of your discussion session on

October 31-November 4, 2016

Reading

Chapter 23 in Code Complete

Grading Guidelines

Points will be deducted if code and branch coverage is incomplete. An automatic C (or less) is triggered by:

- Any routine with complexity greater than 4,
- Any substantially repeated piece of code, or by
- Improperly named routines.

Programming

In this assignment, you will debug a security application (posted on blackboard). Here are the original specifications, which were then incorrectly implemented in the provided code.

Input: Security Log Summary

Servers keep logs of system activity and send daily a summary to the system operator. The reports cover both regular and suspicious activity. The operator can examine the summary and take preventive or remedial actions.

Failed Logins

An important section reports *failed logins*, which are login attempts with a right user name and three consecutive attempts at entering a wrong password. Here is a real example (a veritable Hall of Shame):

```
Failed logins from:
    2.115.68.148 (host148-68-static.115-2-b.business.telecomitalia.it): 7 times
    27.254.44.45: 6 times
    31.186.13.221 (server-1.gencfb.org): 7 times
    43.255.188.161: 2968 times
    50.62.42.229 (ip-50-62-42-229.ip.secureserver.net): 7 times
    50.63.56.230 (ip-50-63-56-230.ip.secureserver.net): 7 times
    58.137.72.110: 7 times
    59.175.153.96 (96.153.175.59.broad.wh.hb.dynamic.163data.com.cn): 7 times
    63.251.162.66 (insynq-1.border2.sef003.pnap.net): 396 times
```

The section starts with the line "Failed logins from:" and gives on each subsequent line the IP address of the client, followed by its fully qualified domain name in parenthesis, if available, then a colon and the number of times that a login attempt originated from the host. Failed login can be malicious, but can also naturally arise from user errors.

Illegal Users

The next section follows the failed logins after a single blank line and gives the login attempts that used a non-existing user name:

```
Illegal users from:
    2.115.68.148 (host148-68-static.115-2-b.business.telecomitalia.it): 1 time
    27.254.44.45: 1 time
    31.186.13.221 (server-1.gencfb.org): 1 time
    50.62.42.229 (ip-50-62-42-229.ip.secureserver.net): 1 time
    50.63.56.230 (ip-50-63-56-230.ip.secureserver.net): 1 time
    58.137.72.110: 1 time
    59.175.153.96 (96.153.175.59.broad.wh.hb.dynamic.163data.com.cn): 1 time
    63.251.162.66 (insynq-1.border2.sef003.pnap.net): 521 times
```

The format is the same as the failed logins. Illegal users can arise from user errors, or from a malicious attack that tries to guess the user names on the system under attack. After the illegal user section, many other sections of the security logs are summarized but do not affect your assignment.

Input Summary

Your program should blacklists the hosts that generated an excessive number of failed logins or illegal user attempts. Its command line argument is a security threshold, by default equal to three. It takes from standard input the full log summary (including but not limited to the failed login and illegal user sections).

Output: Deny List

Your program will generate on standard output a list of hosts to be blacklisted. The output should only list the hosts who made a

combined total of failed logins or illegal user attempts more than the threshold. It should use the fully qualified domain names when available and list the IP addresses otherwise. It should not list the same host more than once. The output consists of a sequence of lines with five leading blanks, all but the last one terminated by a backslash. Each line is a comma-separated list of offending hosts. No line should contain more than eighty characters. The only exception would be a long host name that would not fit in an eighty-character line. Here is an example (leading blanks not shown):

```
host148-68-static.115-2-b.business.telecomitalia.it, 27.254.44.45,\
server-1.gencfb.org, 43.255.188.161,\
ip-50-62-42-229.ip.secureserver.net,\
ip-50-63-56-230.ip.secureserver.net, 58.137.72.110,\
96.153.175.59.broad.wh.hb.dynamic.163data.com.cn,\
insynq-1.border2.sef003.pnap.net
```

Run

Create a new task called run (invoked with 'ant run' or 'make run') that executes your code taking the input from standard input and putting its output on standard output.

If the provided code fails to implement this specification, you should consider it a defect to fix.

Debug the code provided in blackboard to match the specification above. You are not allowed to execute brute-force debugging involving regular expressions. Submit a diff file to document all of the fixes that you made. Create test cases that capture the defects that you find.

General Considerations

Your implementation may contain as many auxiliary private methods as you see fit, and additional helper classes may be defined. After Programming Assignment 8, your code should have an extensive unit test suite. Your code should have a reasonable number of comments, but documentation is going to be the topic of a future assignment. As a general guideline at this stage of the course, comments should be similar to those accepted in EECS 132.

Discussion Guidelines

The class discussion will focus on refactoring (changes to the source code) to fix bugs, and on the debug test cases.

Submission

Create a repository called debug.git where you will post your submission. Make small regular commits and push your revised code and test cases on the git repository.