SCHOOL OF DIGITAL MEDIA AND INFOCOMM TECHNOLOGY
DIPLOMA IN INFOCOMM SECURITY MANAGEMENT

Year 2

ST2512 PROGRAMMING IN SECURITY

~ ASSIGNMENT 2 ~

Plagiarism Warning Statement

Warning: Plagiarism means passing off as one's own the ideas, works, writings, etc., which belong to another person. In accordance with this definition, you are committing plagiarism if you copy the work of another person and turning it in as your own, even if you would have the permission of that person. Plagiarism is a serious offence and disciplinary action will be taken against you. If you are guilty of plagiarism, you may fail all modules in the semester, or even be liable for expulsion.

Overview

This is a group assignment. Each group is formed by TWO students.
For this assignment you will be implementing a few python programs to fulfill **THREE** programming tasks.

1. This is a group assignment; each group can be made up with not more than two students. Each group has to work independently from other groups. (See the Plagiarism Warning Statement)

2. Your application must be submitted via blackboard by **18th of August 2016 05:00PM**.

3. Your python files should be named <**filename**>**XXXX_YYYY.py** (XXXX and YYYY are the last 4 digits of your student numbers). In case you choose to work on your own without a partner, your C file should be named **<filename>XXXX.py** (XXXX is the last 4 digits of your own student number).

   Important: Your source files of task 1 and task 2 must be able to be executed in the python 2.7/Ubuntu environment. Your source files of task 3 must be able to be executed in both of the python 2.7/Ubuntu and python 2.7/Windows environment.

4. You only need to submit **python source files**.

   Some data file, sample and supplementary source codes will be provided to you at the module page of Blackboard.

5. At the beginning of your source codes, add in comments, to state clearly of the names, student IDs and class of the two students whom are forming the group.

6. At the beginning of each function, add in comments, to state clearly of the purpose of the function, the input parameter(s) (if any), and the return value (if any).

7. To ensure the higher degree of readability, you may need to add in relevant inline comments to explain your program (usage / logic / assumptions).

8. Late Submission Penalties

   - Standard DMIT late submission penalty is applied.
     - 50% marks reduction per day. 0 mark after 1 working day.

9. References (See the Plagiarism Warning Statement)
   In case you are applying some codes / techniques that you have learned from external reference, you need to state it clearly in your source codes. E.g. A sample codes in python from a web page, the usage of a particular python library module/function from a text book, etc.

Detail Program Tasks Requirement

Task 1: Log Scanning

Write a python program to scan and analyse the given sample log file, access_log*.

You should be able to find this log file in your Ubuntu system under the ~/Misc directory. It is stored in the access_log.gz.

This sample log file is in 'Common Log Format':

Each line in the file contains a web access log entry in the following format:

```
%h %l %u %t \"%r\" %>s %b
```

Where:

- %h is the IP address of the client
- %l is identity of the client, or "-" if it's unavailable
- %u is username of the client, or "-" if it's unavailable
- %t is the time that the server finished processing the request. The format is [day/month/year:hour:minute:second zone]

- %r is the request line from the client is given (in double quotes). It contains the method, path, query-string, and protocol or the request.
- %>s is the status code that the server sends back to the client. You will see see mostly status codes 200 (OK - The request has succeeded), 304 (Not Modified) and 404 (Not Found). See more information on status codes in W3C.org
- %b is the size of the object returned to the client, in bytes. It will be "-" in case of status code 304.

*This sample log file is obtained from www.udacity.com.

The objective of this task is to scan the access_log to find out:

1) How many log entries were stored in the access_log file?
2) How many hits were made to the page: /assets/js/the-associates.js ?
3) How many hits were made by the IP address 10.99.99.186 ?
4) Identify the largest page or object that was returned to the requester and what was its size in bytes?
5) Which was the most visited page and what was the corresponding number of hits?

   Hints:

   [The correct answer for the most visited page hits should not be 117348]

   [The sample log file contains the log entries for the site: "www.the-associates.co.uk"]

   [You may consider to use regular expression to simplify your implements]

Below is a sample output of the program after scanning the entries from a smaller log file.

Take note that, the program start and end time are also required to be displayed.

```
>>>
Program started time:  2016-07-12 15:37:32
Total entries were  50
Total hits of '/assets/js/the-associates.js' were  3
Total hits made by 10.99.99.186 were  0
The largest page/object was  /assets/swf/home-media-block.swf  with the size of
 123884  bytes
The highest number of hits was  4  for the page  /
program completed time:  2016-07-12 15:37:32
>>>
```

Task 2: Password Recovery

Write a python program to recover the passwords of five user accounts that listed in the given shadow.txt*.

*This shadow.txt is extracted from a '/etc/shadow' file of an Ubuntu system. You can download it from the BB ST2512 assignment page.

Requirement/Restrictions/Hints:

- You are not allowed to use 'rainbow table' from the third parties to recover the passwords.
- You may use crypt python library (only available in linux platform) to solve this programming problem.
- The passwords you need to recovered only made up with 'unique' decimal digits (0..9), and none of their length are longer than 6 digits (i.e. at most, up to 6 digits).

Below is a sample output of the program which has successfully recovered all 5 user passwords. (In this example: The recovered passwords were made up with lowercase alphabets)

Take note that, the program start and end time have also been displayed.

```
Program started time:  2016-07-12 18:30:27
password of user  pseUser5  is  boiz
password of user  pseUser2  is  ckrx
password of user  pseUser4  is  msir
password of user  pseUser1  is  qyva
password of user  pseUser3  is  wifn
Program completed time:  2016-07-12 19:39:32
```

Task 3: Multiuser Chat server

Write a python program to facilitate an online chat service that supports 'WhatsApp' like text base chatting.

Sample Client

A sample chat client program – simChatClient.py* , is given as reference. Your chat server program is required to work with this simChatClient.py and allow multiple instances of them to chat with each other.

*simChatClient.py - You can download it from the BB ST2512 assignment page.

Detail Requirement

- Your server program is required to be functioning in Python 2.7/Windows platform.
- You need to write a special admin client and it can be used to manage the chat service:
    - List the current connected user
        - Show client ip address, nick name (at the admin client interface).
    - Broadcasting administrative message to all clients.
    - Send special private administrative message to particular client.
    - Kill/Force disconnection of particular client.
    - Shutdown the chat server.

Suggestions:

Referring to sample codes from Lecture 7 to implement your own solution.

Avoid to adopt sample codes in your solution that you do not have clear understanding.

Marking Scheme

Here is the marking scheme of this assignment

Category	Description	Type (Group or Individual)	Weightage
Proper Submission	.Fulfilling the submission requirement stated in this document	Group	5 marks
Readability and General Programming techniques	.inline comments /documentation .choice of the variable names .proper choice of programming constructs/data types .proper breakdown of the program with sub functions	Group	15 marks
Task 1 : log scanning	.Reasonable Program Efficiency .proper choice of python libraries .proper choice of algorithms .Able to find the correct answers	Group	15 marks
Task 2 : password recovery	.Reasonable Program Efficiency .proper choice of algorithms .Able to find the correct answers	Group	15 marks
Task 3a: Multi-chat server	.Fulfilment of the given specifications to support simChatClient . Fulfilment of the given specifications to support Admin Client .Runtime errors free	Group	25 marks
Task 3b: Admin Client	.Fulfilment of the given specifications .Innovative user interface .Runtime errors free	Group	15 marks
Written Report	outline .Your contributions .the problems you have encountered .the knowledge you have gained from this assignment	Individual	10 marks

Total: 100 marks

~ The End ~