

CSE 331 - Summer 2015
PROGRAMMING ASSIGNMENT 3
Due Tuesday, July 21, 2015 @ 2:10pm

IMPORTANT: The stated deadline on assignments will be strictly enforced. I will go over solutions at the deadline and will not accept submissions after the solutions have been presented.

Relevant code can be found at <https://bitbucket.org/hartloff/cse331-summer2015/> in the `assignments.assignment3` package. It is recommended that you clone the entire repository and pull regularly as all the code for this course will be placed there.

1 Website Security (100 points)

You are about to go live with a new website that will be home to a database with sensitive information that poses a critical security risk. With this, it is likely that some talented hackers will attempt to attack the site and compromise the system. The launch of the site seems somewhat hopeless until a well meaning, yet vague, white hat hacker informs you that k of the n users of the site are attackers each of whom know a different vulnerability in your site. You would like to use this information to ensure that the launch of the site goes well.

The soon to be users and attackers are constantly pinged your site waiting for it to go live. You don't want to go live with any security holes, but you can activate a fake site to any subset of the users by IP address to check if any of them are attackers. These fake launches will not expose any sensitive data, but will appear to be a legitimate launch to the attackers. Each time you launch a fake site, you can tell if there are any attackers in the chosen subset of users by observing the site after launch. If exactly one hacker attacks the site, you will be able to analyze the behavior and patch the security hole used to compromise the fake site. However, if more than one hacker attacks the site on a fake launch they will alter enough of the site that you will be unable to learn about the vulnerabilities they exploited. Launching fake sites to different subsets of the user base can be used to identify attackers and identify security holes, but each launch is expensive and can irritate the legitimate users of the site.

Your goal is to identify each of the attackers by launching fake sites that only go live to one attacker at a time while minimizing the total number of fake launches. This will be done by implementing Method 1 in the `Assignment3` class.

Method 1. `public static void securityScreen(Website website, int k)`

For specific implementation details, see the `Website` class in the repository.

Grading rubric: Though we are usually concerned with asymptotic runtimes in this course, this assignment will be graded based on absolute values of the number fake launches used to neutralize all the attackers for various values of n and k . All the values for the number of fake launches is for the worst case input.

- (20 points) Method 1 launches the live website with $k = 1$ and $n \leq 100$.
 - 10 points for a successful live launch
 - 15 points for a successful live launch with ≤ 5 fake launches.
 - 20 points for a successful live launch with 1 fake launch.
- (30 points) Method 1 launches the live website with $k = 2$ and $n \leq 100$.
 - 10 points for a successful live launch
 - 20 points for a successful live launch with ≤ 20 fake launches
 - 30 points for a successful live launch with ≤ 14 fake launches
- (25 points) Method 1 launches the live website with $k = 3$ and $n \leq 100$.
 - 10 points for a successful live launch
 - 15 points for a successful live launch with ≤ 20 fake launches
 - 25 points for a successful live launch with ≤ 14 fake launches
- (25 points) Method 1 launches the live website with $k = 4$ and $n \leq 100$.
 - 10 points for a successful live launch
 - 20 points for a successful live launch with ≤ 30 fake launches
 - 25 points for a successful live launch with ≤ 24 fake launches
- (Bonus: 10 points) Output a successful live launch for all n and k with ≤ 1 fake launch by any means available except by altering the `Website.java` file. This solution will not count for any of the regular credit points which must be obtained without manipulating the given classes. If you complete this part, submit it as a separate file named `Assignment3Bonus.java`.

2 Submission

Submit the file `Assignment3.java` which contains your implementation of Method 1. You may add additional methods or classes as long as they are all contained in the `Assignment3.java` file. All other files and classes in the package will be used exactly as they are given for grading and should not be modified.

The preferred method for submission is the `submit.cse331` command. Emailed submissions will also be accepted.