CS 161

Computer Security

Dawn Song

Fall 2012

Due Date: Monday November 19, 2012 by 11:59pm

Document Revision History

Nov 5, 2012	Lab released.
Nov 15, 2012	Several corrections about IDs in Q3 and submisson issues.
Nov 16, 2012	Corrected note about problems that use the VM.

Administrative notes

- There are 3 questions, and each is worth 25 points, out of a total of 75.
- You will need a VM to answer the timing attack question. You can find the VM image at `http://samuel.cs.berkeley.edu/lab5/lab5.ova` The username and password are `ubuntu` and `ubuntu`, respectively.
- Note that for the `http://samuel.cs.berkeley.edu` URLs in the lab, there is no "`www`" preceding the URL.
- Hint: Unlike some previous labs, this lab requires more thinking and less coding. You will only need to code up very small programs.
- Each group member MUST submit their own answers, even if they are identical to their partner's answers.

1 Setup

Before you begin, you need to go to samuel.cs.berkeley.edu/lab5/id.php. This will allow you to access your lab 5 ID. Just enter in your student ID and it will give you your lab 5 ID. This is necessary because one of the problems in the lab involves SQL Injection, and we don't want you to directly access the student IDs of other students (as per Cal policy). For the rest of the lab, if we say just "ID," we are referring to your lab 5 ID. We'll say "student ID" if we mean your student ID.

2 SQL Injection

For this problem, you will perform an SQL injection attack to infiltrate an online database of informants and retrieve the password of the particular informant that corresponds to your ID. The database only stores passwords hashed with a salt but the original passwords are limited to only lowercase alphanumeric characters. After you retrieve the hash value from the database, you must crack it and retrieve the actual password. You may write your own program to crack the password or use an existing program. Start by visiting the website that interacts with the database `http://samuel.cs.berkeley.edu/lab5/informantlist/search.php`

Create a file call q1.txt whose first and only line is the password for the informant with your ID. Do not write anything else in the file. You will lose ALL points for not following this format. Note that the question is asking for the password, and not for the hash value.

You should also submit a q1 directory containing the code that you used to solve the question.

3 XSS Attack

In this problem, you will be attacking a website that lets students create a profile online. All students must create a profile that administrators can view to get more information about them. The profile is simple; it consists of your name, an “About You” paragraph, and a homepage URL. All the other details, available only to administrators, are filled in by the dreaded secret police. You can view your profile, as can administrators, but other regular users cannot. Your goal for this problem is to execute code as an administrator by successfully completing an XSS attack via your user profile.

To start this assignment, you will need your user id and password. Your user id is the same as in part 1. Your password is the password you have hacked in part 1.

Begin by visiting `http://samuel.cs.berkeley.edu/lab5/xss/` and logging in with your user id and password.

From there, there are two pages of interest: update and view. You will probably want to start by just playing around with the update page, entering in relevant information, and then viewing the results on the view page. In order to get credit for this portion of the assignment, you must create a profile that, when viewed and activated by the administrator calls the JavaScript function `winning()`. We have put that function on the page so you can test your injection.

You may assume that the administrator will view, click, and generally explore your profile. That is, your attack may execute `winning()` on load or on some particular activation; it is your choice. We will test your solution by actually loading your profile from the site, so make sure you leave it in an injected state at the deadline. Additionally, this also implies that there is no file submission for this question.

4 Timing Attack

For this problem, we have created a login with your ID and a random password at `http://final-test.evil/` which is only accessible from your VM. The test is for you to figure out the password. Note that this question does NOT require a naive brute-force; we will be looking at your source files to make sure you are not naively brute-forcing the solution.

Also of note, there is an error in the distrubted VM. It asks for your “Student ID”. That is an error. It should have asked just for your “ID”, that is, your Lab 5 ID.

You should create a file named q3.txt with your password as the only line. You should also submit a q3 directory containing all the code you wrote for this question.

5 Submission

Create a text file named extra.txt listing: your student ID, your name, your partner's name, and, a description of your approach for each question. Thus, extra.txt will have 3 lines followed by text. The first line MUST only contain numbers, specifically your student ID (NOT your lab 5 ID). Also note that each member of your group MUST submit their own answers with their respective solutions for their ID. You must submit your own or you will not receive credit.

In order to create the proper submission, you'll need access to a Linux machine, thus you can do this all within the lab 5 VM, if needed. Create a directory called “answers,” placing in it all the files you need to submit. Thus, it should contain:

answers/
answers/q1.txt
answers/q1
answers/q3.txt
answers/q3
answers/extra.txt

Then, from the directory that contains the “answers” directory, run the following:

```
tar -czf answers.tar.gz answers
```

This should result in a file “answers.tar.gz”. This is the file you should submit on bspace.

The whole assignment will be graded by a computer. You WILL get zero points if you do not follow this format, and we will not consider requests for exceptions. An example submission is available at `http://samuel.cs.berkeley.edu/lab5/answers.tar.gz` Please review it to see if you have any questions about the format.

6 Resources

Here are several sources you may find helpful:

- `http://www.securitytube.net/video/2172`
- `http://www.securitytube.net/video/2182`
- `https://owasp.org/index.php/SQL_Injection`
- `https://owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet`
- `http://www.unixwiz.net/techtips/sql-injection.html`
- `http://www.thegeekstuff.com/2012/02/xss-attack-examples/`