Exercise 5

coursys.sfu.ca/2021fa-cmpt-470-d1/pages/Exercise5

For this exercise, you will be attempting to apply your knowledge of security holes in web applications.

Have a look at the extremely simple personal information manager (PIM) application that I have created. Accounts have been created with usernames “user1” through “user4”. In each case, the password is the same as the username.

The system's functionality is essentially:

- A user can create “contacts”. Each contact has a first name, last name, email address, phone number, and notes about them.
- A user can select other users as “friends” and share his/her contact list with that person.
- When that other person logs in, they can see all of that user's contacts.

It's not exactly advanced, but it will do.

I originally created a system that I thought was reasonably secure, but then went back and introduced several security holes. Your task for this exercise is to **find any security holes in the system**.

Create a plain text file and document each security hole you find. That is, indicate what the problem is (where it can be found, how to trigger it, etc), the type of security hole, and the impact of the flaw. It shouldn't take more than a sentence or two to describe each one: don't write an essay.

Hints

Here is the general list of security holes I know about:

- 1 × XSS possibility
- 2 × insufficient authentication/permissions not checked
- 1 × SQL injection (which I know theoretically is there, but haven't been able to exploit myself)
- 2 × CSRF possible
- 1 × other inappropriately-trusted input

You certainly don't have to find all of these to get full marks for the exercise: do your best to explore the system, and revisit after I post the solutions.

Notes

The database for the system is reset to its initial state every hour, on the hour. So, you can add and delete information without worrying about causing permanent damage. But, please remember that others are using the system, and be nice with the information you add/delete. Also, try not to give things away to somebody else exploring the system.

The hourly database refresh will log everybody out of the system.

If you stumble across a security hole that can cause serious damage, be polite and just document it without leaving the system unusable for others. Any damage you can cause should be repaired along with the hourly database reset.

Please, no denial-of-service attacks, brute-forcing, or anything else that will make the tech staff yell at me.

Submit the text file documenting security holes to CourSys activity Exercise 5.

Updated Mon Aug. 30 2021, 07:36 by tienv.