Lab 8

CSRF Protection and Flash Messages

Do not use any third-party packages such as ‘flash’, ‘csrf’, or ‘sessions’ in this lab exercise. The purpose of the lab is to understand how these packages work, not how to use some 3rd party package. Any submission using these additional packages will receive a score of zero, even if the submission works.

# Overview

In this lab, you will take an existing application that is susceptible to cross site request forgery attacks, and you will implement the ‘tokenizer’ pattern to prevent such attacks. You will also replace the message passing via query parameters with flash messages to make the interface look a bit better.

# Get Familiar with the Application

This application planned to be a system for participants to register for an event. The idea is that many people would register to attend an event and their applications would be stored. Some administrative type person would log in and review the applications and approve some of them. Please try the following sequence of actions to ensure that you understand how this program works:

1. Register 3 people.
2. Log in as the administrator.
3. Look at the list of people.
4. Approve a couple of applications.

You do not need any type of screenshots or answer any questions for this activity, this is just so that you understand what the system does.

# Session Architecture Review

Please review the session implementation in the supplied program of how sessions are implemented. This application uses a very different approach to managing sessions than we have seen in the lectures and labs.

There is a file called session\_details.txt in the project. Please answer the following questions there.

* How are expired sessions removed from the persistence?
* Could you do something to improve the session clean up? The answer of there is nothing to be done is okay but you will need to explain. Telling us to switch to mongoDB is not an option so do not mention it!

# Demonstrate Vulnerability

* Sign up as a new user and try putting in html tags including <script> tags to do a simple alert. You can also “hack” the admin’s view by changing the color of the text or background.
* Log in as the admin to see what chaos the user has generated.

# Create the Attack

On page 4 of the slides, there is an ajax call that caused some unintentional side effect in the background. Use that piece of code to cause a CSRF attack. When executed, the code should cause the application to be approved.

The demonstration in the lecture used the jQuery library. This time you must use the “fetch” operation so that no additional libraries are required. You can use the following to help you send the data:



# Prevent CSRF Attacks

Using the technique shown in class, only allow for the “post” operation to be performed from the actual application pages. Recall what you need to do:

1. When you load the page of applicants, generate as CSRF token.
2. Put the CSRF token in the form. In this page, you will have multiple forms but you can use the same token for each since we will just post one time.
3. When the POST happens, check that the CSRF token is as expected:
   1. If the CSRF token was valid, update the data and clear the token.
   2. If the token is not valid then show some error. Just a simple page, no template even required for this.
4. Verify that the attack from before now fails.

Take a look at the applications.handlebars file and look where the this.description has been placed. Do you notice the triple {{{}}}? Change it to {{}} and see what the page looks like when you try the attack. This is not preventing CSRF attacks in general, but it does help in this case. Go back and look at the layouts that you created for the {{{}}}.

# Flash Messages

You will notice several places where messages are being shown to the user. Convert each of these to use the one-time flash style messages as shown in the lecture.

This turns out to be quite challenging. You will need to create a session for each person (give the username “”) so that they can hold the flash message. When you log out, you will need to create a new session with the empty username.