Secure Web-Based Systems Fall 2020
CS 4339/5339 Professor L. Longpré

Assignment 3

In this assignment you will implement the authentication part of a generic web-based system.

Motivation

Nowadays, almost every web-based system includes an authentication module where users need to sign-in to access some features of the system. Attackers attempt to bypass the authentication to get unauthorized access. There are many ways an attacker can succeed, and no matter how good our authentication is, the system should have other ways to detect and react to unauthorized access. Yet, it would be careless not to include some basic design features of a password based authentication that makes it harder for the attacker and not more burdensome for the legitimate user. In this assignment, you will implement such an authentication module that you should be able to import into future web-based systems.

Overview

Your web site will have three types of users: visitors, regular users, and administrators. Information about regular users and administrators will be stored in a MySQL database. Visitors have access to public information without needing to go through authentication. Access to web pages will depend on which type of user attempts to load the page.

Pages

Your web site should have:

1. a main page named mainpage.php,
2. a sign in page named signin.php
3. a page accessible by all signed-in users named user.php,
4. a page accessible only by signed-in administrators named admin.php.

Access

mainpage.php and signin.php can be accessed by all.
user.php can only be accessed by signed-in users.
admin.php can only be accessed by signed-in administrators.
Trying to access a page where access is denied should do one of the following (your choice):

- redirect to the sign-in page
- display an appropriate error message and a display similar to the sign-in page
- display an appropriate error message and a button or link that will bring the user to the sign-in page

Page contents

All pages should have a sign-in button if the visitor is not signed in, and a sign-out button otherwise.
All pages should have links to the other accessible pages, and no link to non-accessible pages. For example, if an administrator visits the main page, there should be a link to user.php and to admin.php.
user.php should display the user's information.
admin.php should have a form to add new users. The form should allow adding either regular users or other administrators. It should also have a link or button to display the list of registered users. (Optional: have a way to delete users, but don't allow deleting yourself.)

Users database

You will need a database with a table of registered users. You can choose to either have separate tables for regular and registered users, or have only one table with a field indicating if the user is regular or administrator. In addition to data needed for authentication, the database should include first name, last name, username, time of account creation and time of last login. You will use PHP's `password_hash()` function to generate the hash that will be included in the user's record. In addition to the salt that is automatically added by the `password_hash` function, your program should have a constant salt string of random characters which it appends to the password:

`password_hash($password.$salt, PASSWORD_DEFAULT);`. This way, one would need to have access to both the database and the php program to mount an offline password cracking attack. Usernames should be unique. Create at least one administrator account with admin as user name and nimda339 as password. This will allow convenient access for the grader. Create at least one regular user account. Please include this regular user's password in your report.

Authentication

Do not use the HTTP authentication headers described in our textbook. When signing in, check the username and password against the registered users table. Then use PHP sessions to keep the user logged in. Make sure you destroy the session when the user signs out.

Testing

First create and test all your pages on your localhost. Also put a copies of your pages on the UTEP server. Since I believe everyone can have access to your main page on the UTEP server, create a directory with a unguessable name that effectively acts as an access password. Put your assignment files in that directory. Make sure your main directory has an `index.html` so that visitors are not able to see the your directory's name. Test your UTEP server installation to make sure it works there as well.

Submission

Write a report that contains the following:

1. The username and password of one regular user account.
2. How long you spent to do this assignment?
3. The name of the directory where your files are.
4. What problems (if any) did you encounter in this assignment, and if yes, how did you solve the problems?
5. Comment on how useful this assignment was as a way to learn about authentication on web-based systems.

Using blackboard, turn in a zipped folder containing your files and your report.

Grading

Grades for this assignment are based on functionality and security. Although you're welcome to make web pages with nice designs, you will not get extra points for that. It is ok if your pages show only text, bare forms and buttons.

Due date

Tuesday, November 10th, 11:59pm.