Assignment 3a

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

Motivation

Nowadays, almost every web-based system include an authentication module where users need sign in to access some features of the system. Attackers attempt to bypass the authentication to get unauthorized access. There are so many ways an attacker can succeed, and no matter how good our authentication is, we should be attempting 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. The regular users and administrators information will be stored in a MySQL database. Access to a page 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 for signed in users named `user.php`,
4. a page for administrators named `admin.php`. 
Access

mainpage.php and signin.php can be accessed by all visitors.
user.php can be only accessed by logged in regular users and administrators.
admin.php can only be accessed by logged in administrators.
Trying to access a page where access is denied should display an appropriate error message, for example, “you need to be logged in as an administrator to access this page” instead of displaying the regular contents of the page.

Page contents

mainpage.php should have a link to the sign in page if the user is not signed in. All pages should have “sign out” button if the user is signed in. All pages should have links to the other accessible pages, and no link to not accessible pages. For example, if an administrator visits the main page, there should be a link to user.php and to admin.php.
All pages should have some text in it indicating where we are.
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 administrators. It should also have a link or button to display the list of registered users.

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 salting and hashing to store the password in the table. For salting, you should use 2 strings: a constant string of random characters, and the username, so you will store in the table the hash function applied to the concatenation of 3 strings: a constant string of random characters, the username, and the password. In this way, one would need to have access to both the database and the php program to mount a password cracking attack, and the username used as an additional salt will slow down the brute force attack. Usernames should be unique.
Create an account administrator account with admin as user name and nimda18 as password. Create at least one regular user account.
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.

Submission
Put a copy of your web page files on the course web server. Please place the files under a directory with a name that is hard to guess, so that other students would not know how to access your pages.
Write a report that contains the following:
1. How long did you spend to do this assignment?
2. The name of the directory where your files are,
3. What problems (if any) did you encounter in this assignment, and if yes, how did you solve the problems?
4. Comment on how useful this assignment was as a way to learn about authentication on web-based systems.

Turn in by sending an e-mail with a .zip attachment containing your files and your report. Please use the subject line according to which course you are registered for: “CS4339 Assignment 3a submission’ or “CS5339 Assignment 3a submission”.

Grading
Grades for this assignment are based on functionality and security, so you won’t lose points if your pages show only text, bare forms and buttons, and you won’t get extra points for nice design.

Late submissions
The penalty for a late submission is the same as in previous assignments.

Due date
Tuesday, November 27, 11:00pm.