Osama Abou Hajar – C00220135

Security Application Project Documentation

Introduction

**Create an authentication mechanism for a web application using XAMPP, PHP & MySQL. Your application should create the underlying database on requesting the login page as a root user with no password. NB: Please test this functionality before mailing me your code, as I cannot test your application without your underlying database**

Requirements

**Register with the system.** **(10% Total)**

* + The system should allow users to register with the system using a username and password.
  + Complexity rules regarding the password should be enforced.
  + Password storage should be salted and hashed.

**On an unsuccessful authentication attempt** **(20% Total)**

A generic error message is presented back to the end user outlining that the username & password combination cannot be authenticated at the moment. ie… “The username Richard and password could not be authenticated at the moment”. Note that the username supplied during the authentication attempt is reflected back to the user interface in the event of an unsuccessful login attempt.

* + Reflect the supplied username provided in the above message. Ensure that this reflected parameter in not susceptible to XSS. You are to write your own sanitisation code for characters that can be utilised for XSS.
  + Lockout after 5 attempts for 3 minutes.

**On successful authentication** **(15% Total)**

* + The system should greet the user by their username.
  + Create an active authenticated session.
  + Allow for the authenticated user to view some pages (at least two) that an unauthenticated user will not have access to.
  + Allow for the user to logout securely.
  + Lockout after 10 minutes of inactivity. (I must block the session for 5 min)
  + Max session duration of one hour irrespective of in session activity.

**Password Change** **(15% Total)**

* + Authenticated users should be capable of changing their password.
  + Complexity rules regarding the password should be enforced.
  + On password change the active session should be expired.
  + The user will have to re-authenticate using new credentials to gain access to the system.
  + No out of band communication, mechanism is required to inform the user that their credentials has been updated.
  + You are to implement Cross Site Request Forgery (CSRF) protection on this page.
* Note: “In the real world” the values contained in this request would be passed as a POST request. **However, to expedite the correction & testing of this assignment you are to pass the values for this functionality in a HTTP GET request.**

**Event Log & ADMIN user (10% Total)**

* + Your application should store unsuccessful and successful login attempts to an event log. This event log should accessible and viewable to the authenticated user “ADMIN” only. (I should display the data)
  + This users authentication details are as follows

Username = “ADMIN”

Password = “SAD\_2019!”

* + This account is to be created, when your database is being created.

Testing Documentation

**Register with the system test:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No** | **Test Objective/Scenario** | **Test Cases (input data)** | **Expected Result** | **Actual Result** |
| 1 | allow users to register with the system using a username and password. | Username= “**osama**”  Password=”**1qaz2wsX@**” | alert('YOUR REGISTRATION HAS BEEN SUCSSEFUL !!'); | PASS |
| 2 | Complexity rules regarding the password should be enforced | Password=”**1qaz2wsX@**” | Password must contain at least one :  uppercase  lowercase  number  charecter  and more than 4 char long | PASS |
| 3 | Password storage should be salted and hashed | Password=”**1qaz2wsX@**” | Password should store in the DB salted and hashed  Results = “**636c640f1693925945b8e3978ea91f8b**” | PASS |

**On an unsuccessful authentication attempt test:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No** | **Test Objective/Scenario** | **Test Cases (input data)** | **Expected Result** | **Actual Result** |
| 1 | Reflect the supplied username provided in the above messag. | Username = “**osama**”  Password=”**1qaz2wsXX**” | The username saf and password could not be authenticated at the moment  Top of Form | PASS |
| 2 | Ensure that this reflected parameter in not susceptible to XSS | Username = **“<script>alert('osama')</script>”** | The username get sanitized and display to the user without any XSS reflect | PASS |
| 3 | Lockout after 5 attempts for 3 minutes | Enter user name and password wrong 5 time | Locked out for 5 minutes for the attempted user agent and ip address | PASS |
| 4 | Lockout after 5 attempts for 3 minutes | Enter user name and password wrong 5 time (trying from different browser than the previous one) | Working and display the login page | PASS |

**On successful authentication attempt test:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No** | **Test Objective/Scenario** | **Test Cases (input data)** | **Expected Result** | **Actual Result** |
| 1 | The system should greet the user by their username. | Username = “**osama**”  Password=”**1qaz2wsX@**” | YOU ARE LOGED IN AS : osama | PASS |
| 2 | Create an active authenticated session | Username = “**osama**”  Password=”**1qaz2wsXX**” | Active new session created and assigned to the user | PASS |
| 3 | Allow for the authenticated user to view some pages (at least two) that an unauthenticated user will not have access to | Page-1.php and page-2.php created and accessed by the user | The two pages were successfully reachable and displayed | PASS |
| 4 | Allow for the user to logout securely | User osama logout from the system | All created session get destroyed and redirect to the login page | PASS |
| 5 | Lockout after 10 minutes of inactivity | Non active user ‘testuser’ for 10 minutes | The user logged out after 10 minutes and asked to login again | PASS |
| 6 | Max session duration of one hour irrespective of in session activity | Creating a session and spend more than one hour working with the same session | The system should logout the user and destroy the session | PASS the user asked to login again |

**Password Change** **attempt test:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No** | **Test Objective/Scenario** | **Test Cases (input data)** | **Expected Result** | **Actual Result** |
| 1 | Authenticated users should be capable of changing their password | Attempting with the following data:  Username=”**osama**”  oldPass=”**1qaz2wsX@”**  newPass=”**123qwEE@”**  confirmPass=”**123qwEE@**” | Alert(YOUR PASS WORD HAS BEEN CHANGED)  Alert (YOU HAVE TO LOG IN AGAIN PLEASE !!) | PASS and the session destroyed, and the user asked to login again |
| 2 | Authenticated users should be capable of changing their password | Attempting with the following data:  Username=”**osama**”  oldPass=”**1qaz2wsX”**  newPass=”**123qwEE@**”  confirmPass=”**123qwEE@**” | Alert tells the user that the (OLD PASSWORD WRONG) | FAIL old password was wrong |
| 3 | Authenticated users should be capable of changing their password | Attempting with the following data:  Username=”**osama**”  oldPass=”**1qaz2wsX@”**  newPass=”**123qwEE@”**  confirmPass=”**123qEXX@”** | Alert tells the user that the (new Password not matched) | FAIL new password and confirm are not matched |
| 4 | Complexity rules regarding the password should be enforced | Password =”123A” | Password must contain at least one :  uppercase  lowercase  number  charecter  and more than 4 char long | FAIL the password should follow the criteria |
| 5 | On password change the active session should be expired | Attempting with the following data:  Username=”**osama**”  oldPass=”**1qaz2wsX@”**  newPass=”**123qwEE@”**  confirmPass=”**123qwEE@**” | Session expired and new session assign with user redirect the login page | PASS |
| 6 | The user will have to re-authenticate using new credentials to gain access to the system | Attempting with the following data:  Username=”**osama**”  oldPass=”**1qaz2wsX@”**  newPass=”**123qwEE@”**  confirmPass=”**123qwEE@**” | Session expired and new session assign with user redirect the login page | PASS |
| 7 | No out of band communication, mechanism is required to inform the user that their credentials has been updated | Enter correct data | Alert(YOUR PASS WORD HAS BEEN CHANGED) | PASS |
| 8 | Cross Site Request Forgery (CSRF) | http://localhost/C00220135/  changePass.php?  oldPass=123qwEEE&  newPass=1qaz2wsX%40&c  onfirmPass=1qaz2wsX%40  &submit=Submit | YOUR PASS WORD HAS BEEN CHANGED | PASS |

**Event Log & ADMIN user test:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No** | **Test Objective/Scenario** | **Test Cases (input data)** | **Expected Result** | **Actual Result** |
| 1 | store unsuccessful and successful login attempts to an event log | Make successful and unsuccessful login to the system | Logs stored in the database and being able to be displayed to the admin user | PASS |
| 2 | event log should accessible and viewable to the authenticated user “ADMIN” only | Login with the ADMIN user | The admin is able to see the logs option in the menu | PASS |
| 3 | Create  Username = “ADMIN”  Password = “SAD\_2019!” when index.php page request if not exist | Request the index.php page | The Username = “ADMIN”  Password = “SAD\_2019!”  Were created | PASS |