

**Secure App Development, Test Cases,**

**Secure Web App System.**

Date: 01/12/2019

Author: Kevin Quinn, C00216607.

Table of Contents

[Brute Forcing Testing 3](#_Toc26459015)

[Cross Site Scripting 4](#_Toc26459016)

[Password Complexity Testing 5](#_Toc26459017)

[Session Management Testing 6](#_Toc26459018)

[CSRF Testing 7](#_Toc26459019)

# Brute Forcing Testing

My application stops brute forcing by recording a user’s attempt at logging in and creating accounts in a database along with their user agent and ip to identify them again should they delete their session. If they reach 5 attempts for either login or create account, the user will be locked out for 3 minutes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Location:** | **Username:** | **Password:** | **Description:** | **Response:** |
| Login Page (UI) | Testuser123 | 123 | Attempt to login in multiple times with wrong credentials. | User is locked out after 5 attempts for 3 minutes. |
| Login Page (ZAP) | Testuser123 | 123 | Attempt to login in multiple times using ZAP manual request to bypass front end and delete session on each request | User is still locked out after 5 attempts for 3 minutes. |
| Create Account Page (UI) | TestUser123 ++ | aA123456@ | Attempt to make multiple accounts | After 5th attempt of multiple accounts or inputting bad details user is locked out for 3 mins. |
| Create Account Page (ZAP) | TestUser123 ++ | aA123456@ | Attempt to make multiple accounts using ZAP to delete session each attempt | User is locked out for 3 mins. |

# Cross Site Scripting

To stop XSS in my app, I have a method which sanitizes all user input and replaces any dangerous characters with their html code equivalent allowing them to be safely displayed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Location:** | **Username:** | **Password:** | **Description:** | **Response:** |
| Login Page(UI) | <script>alert()</script> | 123 | Fill in the login credentials. | Page displays message “the username <script>alert()</script> or password are incorrect”. |
| Login Page(UI) | <scr<script>ipt> alert() </scr</script>ipt> | 123 | Fill in the login credentials. | Page displays message “the username <scr<script>ipt> alert() </scr</script>ipt> or password are incorrect”. |
| Create Account Page (UI) | <script>alert()</script> | 123 | Attempt to create a user with the following username and try store the script in the database. | The user is not allowed to create an account with that username. |
| Create Account Page (ZAP) | <script>alert()</script> | 123 | Attempt the same process again using ZAP to bypass front end security. | The user is not allowed to create the account. |
| Admin View Page (UI) | <script>alert()</script> | 123 | When a user attempts to login it is stored in a log viewable on the admin page, if not handled correctly persistent xss is possible. After an attempt login with this username check admin page. | The log contains the username but no script is ran as it is sanitized before being stored in the database. |

# Password Complexity Testing

Password standards are implemented on the create account and change password page, this is a php method which takes the users desired password and checks that it meets the standards of at least 8 chars in length, one lower- and upper-case char, one number and one special character. If the user’s password doesn’t meet these standards, they are informed of the necessary changes to make.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Location:** | **Username:** | **Password:** | **Description:** | **Response:** |
| Create Account Page (UI) | Testuser123 | 123 | Input the following data | The users account is not created and is informed of the password complexity rules they need to follow. |
| Create Account Page (ZAP) | Testuser123 | 123 | Attempt to bypass front end patterns and directly send request using ZAP. | User’s account still not made and is informed of the password complexity rules that they need to follow. |
| Change Password Page (UI) | N/A | 123 | Input the following data. | Users password is not changed and they are informed that of what complexity standards their new password must meet |
| Change Password Page (ZAP) | N/A | 123 | Attempt to bypass front end and send request directly using ZAP. | Users password is not changed and they are informed that of what complexity standards their new password must meet |

# Session Management Testing

Using php sessions and using the users user agent and ip as the session identifier I can restrict access to pages to certain groups of users i.e. un-authorised, authorised or admin.

|  |  |  |  |
| --- | --- | --- | --- |
| **User Type:** | **Location:** | **Description:** | **Response:** |
| Un-authorised | Welcome Page | An un-authorised user attempts to enter the welcome page without authorising by trying to jump direct to URL. | Users request is rejected, and they are redirected to the Login Page where they can authenticate. |
| Un-authorised | Change Password Page | An un-authorised user attempts to enter the Change Password page without authorising by trying to jump direct to URL. | Users request is rejected, and they are redirected to the Login Page where they can authenticate. |
| Un-authorised | Admin Page | An un-authorised user attempts to enter the Admin page without authorising by trying to jump direct to URL. | Users request is rejected, and they are redirected to the Login Page where they can authenticate. |
| Authorised | Login Page | Whilst logged in an authorised user attempts to jump straight to the login page without logging out using the URL. | Users request is rejected, and they are redirected to the Welcome Page where they can logout. |
| Authorised | Create Account Page | Whilst logged in an authorised user attempts to jump straight to the Create Account page without logging out using the URL. | Users request is rejected, and they are redirected to the Welcome Page where they can logout. |
| Authorised | Admin Page | Whilst logged in an authorised user attempts to jump straight to the Admin page without being an admin using the URL. | Users request is rejected, and they are redirected to the Welcome Page as they are not an admin. |

Also, with session management I can log users out after a max session timeout and inactivity timeout. I do this by logging when they login and when they change page that value is checked against the current time to see if they have gone over their timeout.

|  |  |  |
| --- | --- | --- |
| **Timeout Test:** | **Description:** | **Response:** |
| 10 min wait (UI) | An Authenticated user is inactive for 10 mins, i.e. hast changed pages. | The user is logged out. |
| 10 min wait (ZAP) | An Authenticated user is inactive for 10 mins, but they are using ZAP proxy to delete their session to try bypass the logout. | The user is logged out. |
| 1 hour wait (UI) | An Authenticated user is logged in for an hour even whilst being active and change pages. | The user is logged out for a session lasting over an hour. |
| 1 hour wait (ZAP) | An Authenticated user is logged in for an hour whilst being active but is using ZAP proxy to delete their session to try bypass the 1 hour lockout time limit. | The user is logged out for the session lasting over an hour. |

# CSRF Testing

On the change password page, the implementation of a CSRF Token on the form allows us to prevent CSRF Attack on the user by assigning them a CSRF Token and sending it with the form when trying to change the password. The token with the form is then checked against the users and if they don’t match the user is logged out as a safety precaution.

|  |  |  |
| --- | --- | --- |
| **Page:** | **Description:** | **Response:** |
| Change Password Page | To test this functionality, using ZAP proxy to send the change password request with the hidden form field changed to something else. | Users account is logged out to protect them as someone malicious is trying to interfere. |