SAD CA
Testing and Results

James Hall
C00007006

Secure Application Development

Institute of Technology Carlow

Tutor: Richard Butler
Date: 19th March 2021

Table of contents

Abstract

This document aims to detail the methods used to prevent a variety of malicious actions against a web application and database. The document will explain the methods used to prevent such actions, and the tests performed to ensure the methods are suitable for prevention. The main code base for the application was taken from an online tutorial [1], and the security methods required by the project specifications were added to the codebase.

Security Measures

Each page in the application has a variety of security measures added to prevent various types of cross site scripting and CSRF attacks. There are also sanitisation methods added to ensure that all queries to the database are sanitised.

SanitiseFunctions.php

This file contains the main sanitisation function for input fields in the other files. It takes possibly malicious characters and replaces them with encoded values.

```
<?php

function Sanitise($val)
{

    $data = $val;

    $arrSearch = array('/"/', '/</', '/>/', '/#/', '/%/', '/\{/',
'/\}/', '/\|/', '/\\\/', '/\^/', '/~/', '/\[/', '/\]/', '/`/', '/=/');
    $arrReplace = array('&quot', '&lt', '&gt', '&num', '&percnt',
'&lbrace', '&rbrace', '&vert', '&bsol', '&Hat', '&tilde', '&lbrack',
'&rbrack', '&grave', '&#61');
    $data = preg_replace($arrSearch, $arrReplace, $data);

    return $data;
}
?>
```

Register.php

Xss prevention

In this file all instances of `$_POST["username"]`, `$_POST["password"]` and `$_POST["confirm_password"]` are first passed through the Sanitise() function to ensure they no longer contain malicious characters.

```
$password = trim(Sanitise($_POST["password"]));
```

Password validation

The password is validated to ensure it follows complexity rules. It must contain an upper and lowercase letter, a number and a special character. It must also be at least 6 characters long. This is done with the following code.

```
$correctPassword = false;

      $uppercase = preg_match('@[A-Z]@', $_POST["password"]);
      $lowercase = preg_match('@[a-z]@', $_POST["password"]);
      $number     = preg_match('@[0-9]@', $_POST["password"]);
      $specialChars = preg_match('@[^\w]@', $_POST["password"]);

      if($uppercase && $lowercase && $number && $specialChars) {
          $correctPassword = true;
      }

      if(empty(trim($_POST["password"]))){
          $password_err = "Please enter a password.";
      } elseif(strlen(trim($_POST["password"])) < 6){
          $password_err = "Password must have at least 6 characters.";
      } elseif($correctPassword == false){
        $password_err = "Password must contain at least 1 upper and
lowercase letter, 1 number and 1 special character";
      } elseif(strpos(trim($_POST["password"]),
trim($_POST["username"])) !== false){
          $password_err = "Password must not contain username";
      } else{
$password = trim(Sanitise($_POST["password"]));
}
```

Salting and Hashing the password

When the password is created, it has a unique salt prepended to it and it is then hashed before being stored on the database. The unique salt is also stored separately on the database. This is done as follows.

```php
if(empty($username_err) && empty($password_err) &&
empty($confirm_password_err)){
      //Function to generate salt
      function generateRandomString($length = 64) {
            $characters =
'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ?*%';
            $charactersLength = strlen($characters);
            $randomString = '';
            for ($i = 0; $i < $length; $i++) {
                  $randomString .= $characters[rand(0, $charactersLength
- 1)];
                  }
            return $randomString;
      }
      //Create salt
      $salt = generateRandomString(64);

      // Prepare an insert statement
      $sql = "INSERT INTO users (username, password, salt) VALUES (?, ?,
?)";

      if($stmt = mysqli_prepare($link, $sql)){
            // Bind variables to the prepared statement as parameters
            mysqli_stmt_bind_param($stmt, "sss", $param_username,
$param_password, $salt);

            // Set parameters
            $param_username = $username;
            $saltAndPwd = $salt.$password;
            $param_password = password_hash($saltAndPwd,
PASSWORD_DEFAULT); // Creates a password hash

            // Attempt to execute the prepared statement
            if(mysqli_stmt_execute($stmt)){
                  // Redirect to login page
                  header("location: login.php");
            } else{
                  echo "Something went wrong. Please try again later.";
            }

            // Close statement
            mysqli_stmt_close($stmt);
      }
      }
```

Login.php

Xss prevention

In this file all instances of `$_POST["username"]`, `$_POST["password"]` are first passed through the Sanitise() function to ensure they no longer contain malicious characters.

```
$password = trim(Sanitise($_POST["password"]));
```

Lockout after 5 attempts

If the user attempts to login and fails 5 times in 3 minutes the system will block the user from logging in. This is done using a login table on the database and querying it on every login attempt.

```
//Get number of failed attempts at login
                        $num = mysqli_query($link, "SELECT
failedattemptscount FROM logins WHERE ipaddress LIKE '$ip' and useragent
LIKE '$useragent' and timestamp = (SELECT MAX(timestamp) FROM logins
WHERE ipaddress LIKE '$ip' and useragent LIKE '$useragent')");
                        //Convert to num array
                        $count = mysqli_fetch_array($num, MYSQLI_NUM);
                        //Get last login within previous 3 minutes
                        $lastlog = mysqli_query($link, "SELECT
MAX(timestamp) FROM logins WHERE ipaddress LIKE '$ip' and useragent LIKE
'$useragent' and timestamp > (now() - interval 3 minute) and
failedattemptscount >= 5");
                        //Convert to array
                        $logprint = mysqli_fetch_array($lastlog);

                        //If password is correct and user has not failed
login 5 times and 3 minutes has not elapsed
                        if(password_verify($saltAndPwd, $hashed_password)
&& ($count[0] < 5 || $logprint[0] == "")){
                              //Update logins table
                              mysqli_query($link, "INSERT INTO `logins`
(`ipaddress`,`timestamp`,`success`,`failedattemptscount`,`useragent`)VAL
UES ('$ip',CURRENT_TIMESTAMP,1,0,'$useragent')");
                              mysqli_query($link, "DELETE FROM logins
WHERE ipaddress LIKE '$ip' AND useragent LIKE '$useragent' AND timestamp
!= (SELECT MAX(timestamp) FROM logins WHERE ipaddress LIKE '$ip' AND
useragent LIKE '$useragent')");
                              mysqli_query($link, "INSERT INTO
`login_logs`
(`ip`,`useragent`,`success`,`logintime`,`usernameused`)VALUES
('$ip','$useragent','yes',CURRENT_TIMESTAMP,'$username')");
```

This also updates the login table if the user continues to attempt to login.

```
//If user has failed 5 times and 3 minutes has not elapsed
                        } elseif(($count[0] >= 5) && ($logprint[0] !=
"")){
                              mysqli_query($link, "INSERT INTO logins
(ipaddress,timestamp,success,failedattemptscount,useragent)VALUES
('$ip',CURRENT_TIMESTAMP,0,((SELECT l1.failedattemptscount FROM logins
l1 WHERE l1.ipaddress LIKE '$ip' AND l1.timestamp = (SELECT
MAX(l2.timestamp)FROM logins l2 WHERE l2.ipaddress LIKE
'$ip'))+1),'$useragent')");
                              mysqli_query($link, "DELETE FROM logins
WHERE ipaddress LIKE '$ip' AND useragent LIKE '$useragent' AND timestamp
!= (SELECT MAX(timestamp) FROM logins WHERE ipaddress LIKE '$ip' AND
useragent LIKE '$useragent')");
                              mysqli_query($link, "INSERT INTO
`login_logs`
(`ip`,`useragent`,`success`,`logintime`,`usernameused`)VALUES
('$ip','$useragent','no',CURRENT_TIMESTAMP,'$username')");
                              $password_err = "Only 5 attempts allowed in
3 minutes. $count[0] $logprint[0]";
                        } else{
                              // Display an error message if password is
not valid
                              mysqli_query($link, "INSERT INTO logins
(ipaddress,timestamp,success,failedattemptscount,useragent)VALUES
('$ip',CURRENT_TIMESTAMP,0,((select l1.failedattemptscount from logins
l1 where l1.ipaddress LIKE '$ip' and l1.timestamp = (SELECT
MAX(l2.timestamp)FROM logins l2 WHERE l2.ipaddress LIKE
'$ip'))+1),'$useragent')");
                              mysqli_query($link, "INSERT INTO
`login_logs`
(`ip`,`useragent`,`success`,`logintime`,`usernameused`)VALUES
('$ip','$useragent','no',CURRENT_TIMESTAMP,'$username')");
                              $password_err = "The username $username and
password you entered could not be authenticated. $count[0]";
                        }
```

Creating a new session on login

When the user successfully logs in to the system, the login session is destroyed and a new session is created. A new session id is created at this point.

```
function generateRandomString($length = 26) {
                              $characters =
'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
                              $charactersLength = strlen($characters);
                              $randomString = '';
                              for ($i = 0; $i < $length; $i++) {
                                    $randomString .= $characters[rand(0,
$charactersLength - 1)];
                                    }
                              return $randomString;
                              }
                              // Password is correct, so start a new
session
                              session_unset();
                              session_destroy();
                              session_id(generateRandomString(26));
                              session_start();

                              // Store data in session variables
                              $_SESSION["login_time"] = time();
                              $_SESSION["loggedin"] = true;
                              $_SESSION["id"] = $id;
                              $_SESSION["username"] = $username;
                              $_SESSION["isadmin"] = $isadmin;
```

Welcome.php and Page2.php

Xss prevention

The sanitise function is used again here as the users username is displayed on the page.

```
<?php echo Sanitise($_SESSION["username"]); ?>
```

Logging user out after inactivity

This page and all other pages requiring the user to be logged in contain code to log the user out after 10 minutes of inactivity, as well as after one hour regardless of activity. This code provides this functionality. The logout.php file contains the necessary functionality to properly logout the user and destroy the current session.

```
if (!isset($_SESSION['last_activity'])){
      $_SESSION['last_activity'] = time();
}

if(time() - $_SESSION['last_activity'] > 600 || time()
-$_SESSION["login_time"] > 3600) {
      header("location: logout.php");
      exit;
}
$_SESSION['last_activity'] = time();
```

Logout.php

Logout.php is used to safely log the user out of the system.

```
$_SESSION = array();

function generateRandomString($length = 26) {
      $characters =
'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
      $charactersLength = strlen($characters);
      $randomString = '';
      for ($i = 0; $i < $length; $i++) {
      $randomString .= $characters[rand(0, $charactersLength - 1)];
      }
      return $randomString;
}

// Destroy the session.
session_unset();
session_destroy();
session_id(generateRandomString(26));
session_start();

// Redirect to login page
header("location: login.php");
exit;
```

AdminPage.php

This page only allows a person with admin privileges to access the page. This is done with a simple check on page load with this code at the top of the file.

```
if($_SESSION['isadmin'] != 1){
      header("location: Welcome.php");
}
```

ResetPassword.php

Xss prevention

In this file all instances of `$_POST["username"]`, `$_POST["new_password"]` and `$_POST["confirm_password"]` are first passed through the Sanitise() function to ensure they no longer contain malicious characters.

```
$password = trim(Sanitise($_POST["new_password"]));
```

CSRF prevention

The project specification required CSRF prevention on this page. This was achieved using the following code added as php in the html section of the page.

```
<?php $_SESSION["CSRF_Token"] =
base64_encode(openssl_random_pseudo_bytes(64)); ?>


<input type="hidden" name="token" value="<?= $_SESSION['CSRF_Token']
?>">
```

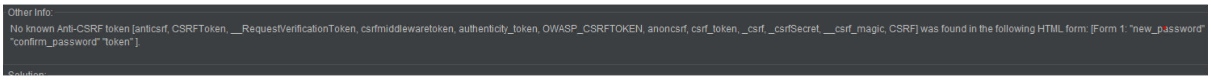
Further up the page where the new password is validated and submitted the CSRF token is checked.

```
// Processing form data when form is submitted
if(isset($_GET["new_password"])&&isset($_GET["confirm_password"])&&(isse
t($_GET["token"]))){
      if(($_GET["token"])==($_SESSION["CSRF_Token"])){

      // Validate new password //////////////
```

OWASP CSRF detection

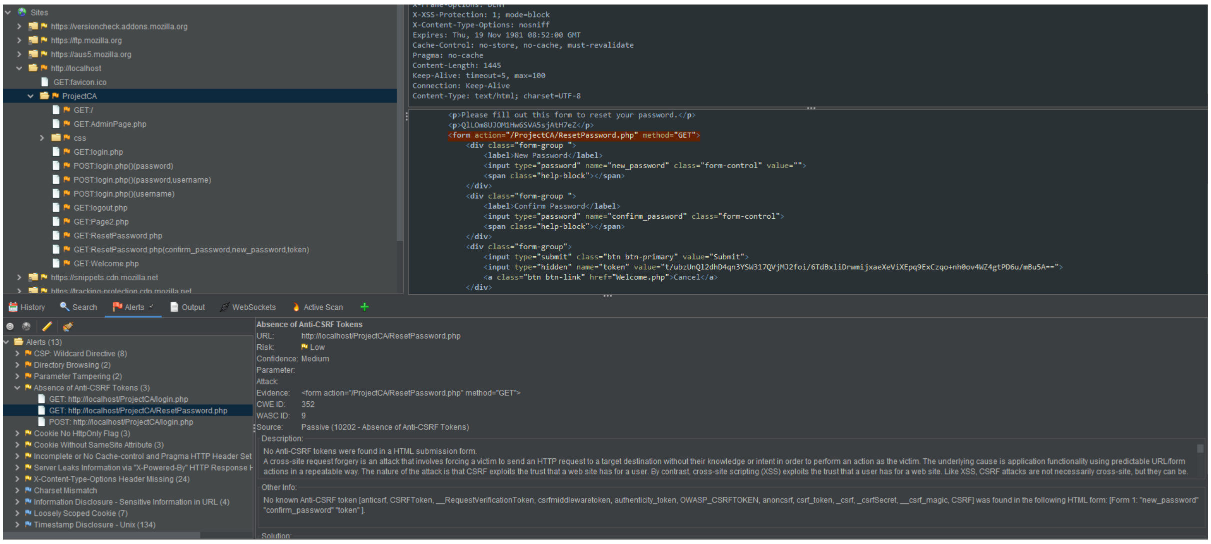
Despite the inclusion of a self made CSRF token, an OWASP scan highlighted the lack of CSRF token on the Reset Password page. This seems to be due to the use of a self made token and not any of the tokens listed by the program under the other info section.



Other Info:

No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF] was found in the following HTML form: [Form 1: "new_password" "confirm_password" "token"].

Solution:



Sites
https://versioncheck.addons.mozilla.org
https://ftp.mozilla.org
https://aus5.mozilla.org
http://localhost
GET:favicon.ico
ProjectCA
GET:/
GET:AdminPage.php
css
GET:login.php
POST:login.php()(password)
POST:login.php()(password,username)
POST:login.php()(username)
GET:logout.php
GET:Page2.php
GET:ResetPassword.php
GET:ResetPassword.php(confirm_password,new_password,token)
GET:Welcome.php
https://snippets.cdn.mozilla.net
https://tracking-protection.cdn.mozilla.net
History
Search
Alerts
Output
WebSockets
Active Scan
Alerts (13)
CSP: Wildcard Directive (8)
Directory Browsing (2)
Parameter Tampering (2)
Absence of Anti-CSRF Tokens (3)
GET: http://localhost/ProjectCA/login.php
GET: http://localhost/ProjectCA/ResetPassword.php
POST: http://localhost/ProjectCA/login.php
Cookie No HttpOnly Flag (3)
Cookie Without SameSite Attribute (3)
Incomplete or No Cache-control and Pragma HTTP Header Set
Server Leaks Information via "X-Powered-By" HTTP Response H
X-Content-Type-Options Header Missing (24)
Charset Mismatch
Information Disclosure - Sensitive Information in URL (4)
Loosely Scoped Cookie (7)
Timestamp Disclosure - Unix (134)

```
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 1445
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

```
        <p>Please fill out this form to reset your password.</p>
        <p>QlLOm8UJOM1Hw6SVA5sjAtH7eZ</p>
        <form action="/ProjectCA/ResetPassword.php" method="GET">
            <div class="form-group ">
                <label>New Password</label>
                <input type="password" name="new_password" class="form-control" value="">
                <span class="help-block"></span>
            </div>
            <div class="form-group ">
                <label>Confirm Password</label>
                <input type="password" name="confirm_password" class="form-control">
                <span class="help-block"></span>
            </div>
            <div class="form-group">
                <input type="submit" class="btn btn-primary" value="Submit">
                <input type="hidden" name="token" value="t/ubzUnQl2dhD4qn3YSW317QVjMJ2foi/6TdBxliDrwmijxaeXeViXEpq9ExCzqo+nh0ov4WZ4gtPD6u/mBu5A==">
                <a class="btn btn-link" href="Welcome.php">Cancel</a>
            </div>
```

Absence of Anti-CSRF Tokens

URL:	http://localhost/ProjectCA/ResetPassword.php
Risk:	Low
Confidence:	Medium
Parameter:	
Attack:	
Evidence:	<form action="/ProjectCA/ResetPassword.php" method="GET">
CWE ID:	352
WASC ID:	9
Source:	Passive (10202 - Absence of Anti-CSRF Tokens)

Description:

No Anti-CSRF tokens were found in a HTML submission form.
A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be.

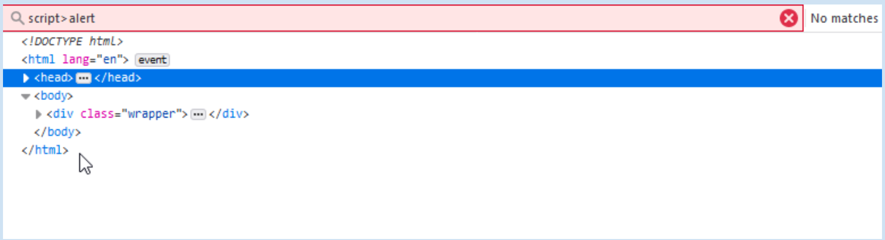
Other Info:

No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF] was found in the following HTML form: [Form 1: "new_password" "confirm_password" "token"].

Solution:

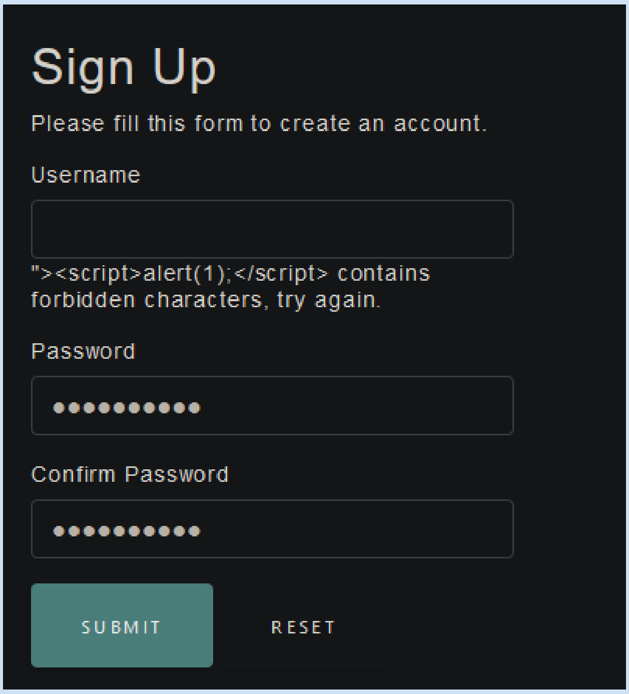
Registration Page

Xss using username field in address bar

Vulnerability tested	Xss using username field in address bar
Input	http://localhost/ProjectCA/Register.php?username=%3Cscript%3Ealert(1);%3C/script%3E
Output	 script>alert No matches <!DOCTYPE html> <html lang="en"> event <head>…</head> <body> <div class="wrapper">…</div> </body> </html>
Outcome	The script was not triggered
Test Result	Pass

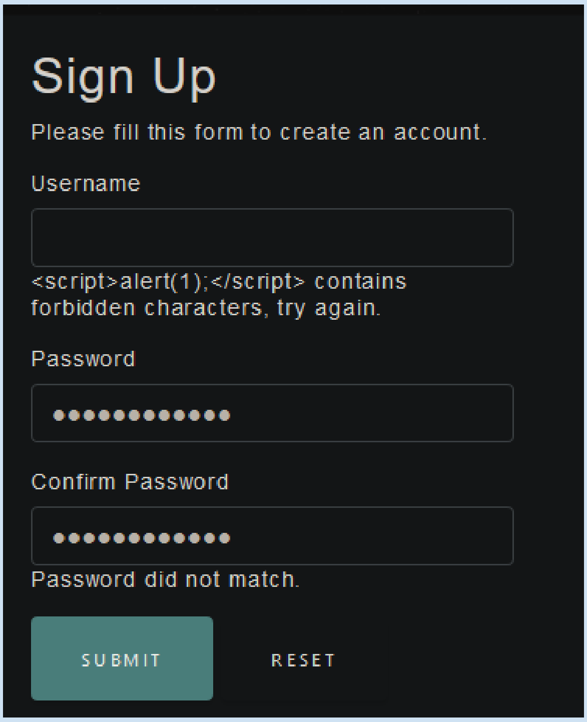
Xss using username text field

Vulnerability tested	Xss using username text field
Input	“><script>alert(1);</script>

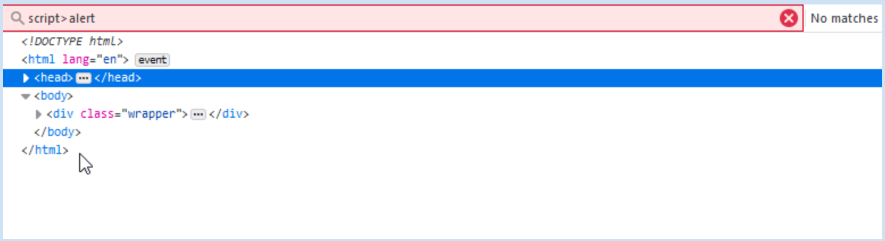
Output	 Sign Up Please fill this form to create an account. Username "><script>alert(1);</script> contains forbidden characters, try again. Password ●●●●●●●●●● Confirm Password ●●●●●●●●●●
Outcome	The script was not triggered
Test Result	Pass

Xss using username text field

Vulnerability tested	Xss using username text field
Input	<script>alert(1);</script>

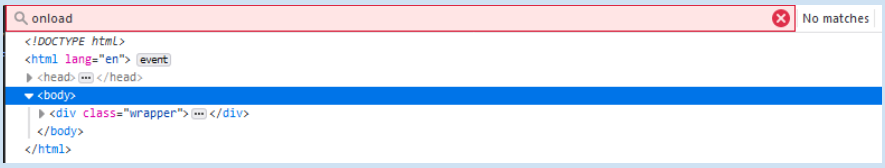
Output	 Sign Up Please fill this form to create an account. Username <script>alert(1);</script> contains forbidden characters, try again. Password ●●●●●●●●●●●● Confirm Password ●●●●●●●●●●●● Password did not match.
Outcome	The script was not triggered
Test Result	Pass

Xss using password and confirm_password field in address bar

Vulnerability tested	Xss using password and confirm_password field in address bar
Input	http://localhost/ProjectCA/Register.php?username=JamesTest&password=%3CScript%3Ealert(1);%3C/script%3E&confirm_password=%3CScript%3Ealert(1);%3C/script%3E
Output	 script>alert No matches <!DOCTYPE html> <html lang="en"> event ▸ <head> ⋯ </head> ▾ <body> ▸ <div class="wrapper"> ⋯ </div> </body> </html>
Outcome	The script was not triggered
Test Result	Pass

Xss using username field in address bar

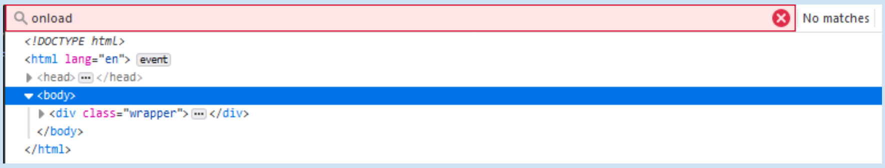
Vulnerability tested	Xss using username field in address bar
Input	http://localhost/ProjectCA/register.php?username=James%3Cbody%20onload=alert(1)%3E

Output	
	onload
	No matches
	`<!DOCTYPE html>`
	`<html lang="en">` event
	`<head>…</head>`
	`<body>`
	`<div class="wrapper">…</div>`
	`</body>`
	`</html>`

Outcome	The script was not triggered
Test Result	Pass

Xss using password field in address bar

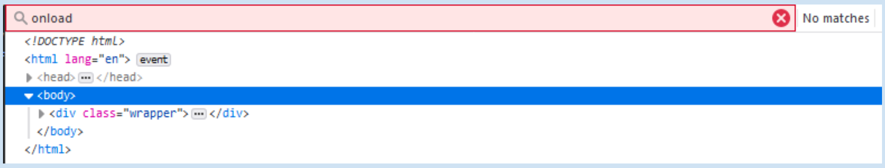
Vulnerability tested	Xss using password field in address bar
Input	http://localhost/ProjectCA/register.php?password=James%3Cbody%20onload=alert(1)%3E

Output	
	onload
	No matches
	`<!DOCTYPE html>`
	`<html lang="en">` event
	`<head>…</head>`
	`<body>`
	`<div class="wrapper">…</div>`
	`</body>`
	`</html>`

Outcome	The script was not triggered
Test Result	Pass

Xss using confirm_password field in address bar

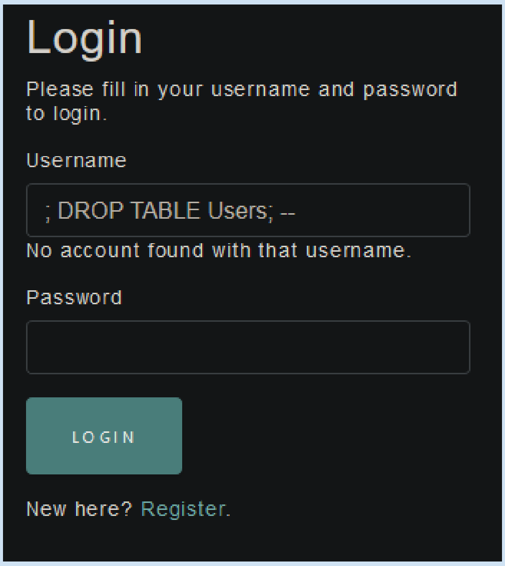
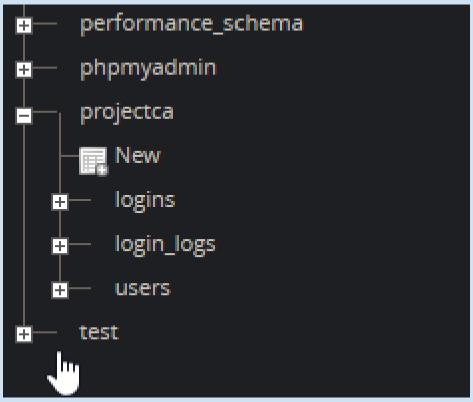
Vulnerability tested	Xss using confirm_password field in address bar
Input	http://localhost/ProjectCA/register.php?password=James%3Cbody%20onload=alert(1)%3E

Output	
	onload
	No matches
	`<!DOCTYPE html>`
	`<html lang="en">` event
	`<head>…</head>`
	`<body>`
	`<div class="wrapper">…</div>`
	`</body>`
	`</html>`

Outcome	The script was not triggered

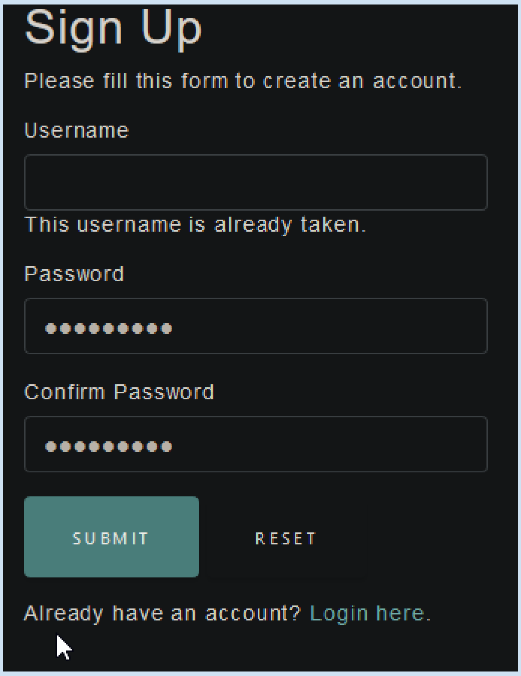
Test Result	Pass

SQL injection

Vulnerability tested	SQL injection
Input	username = ; DROP TABLE users; --
Output	 Login Please fill in your username and password to login. Username ; DROP TABLE Users; -- No account found with that username. Password New here? Register.  performance_schema phpmyadmin projectca New logins login_logs users test
Outcome	Table still exists. MySQL does not support multi queries by default. A separate multi query function is required in the php. The use of AND could be used to return data, though the page layout does not support this.
Test Result	Pass

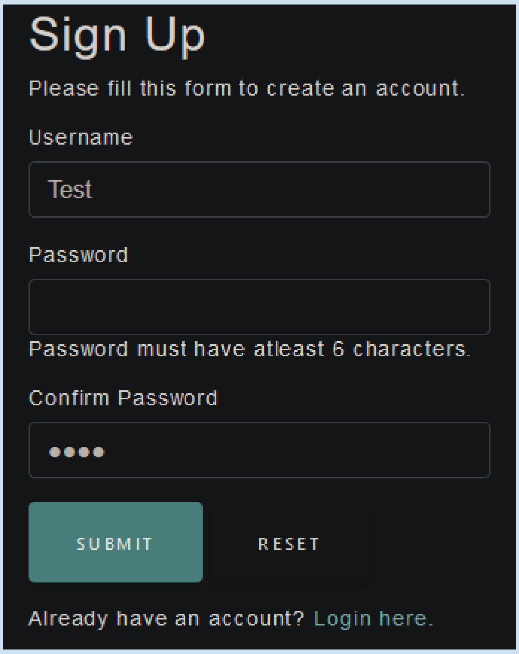
Check if username exists

Vulnerability tested	Check if username exists

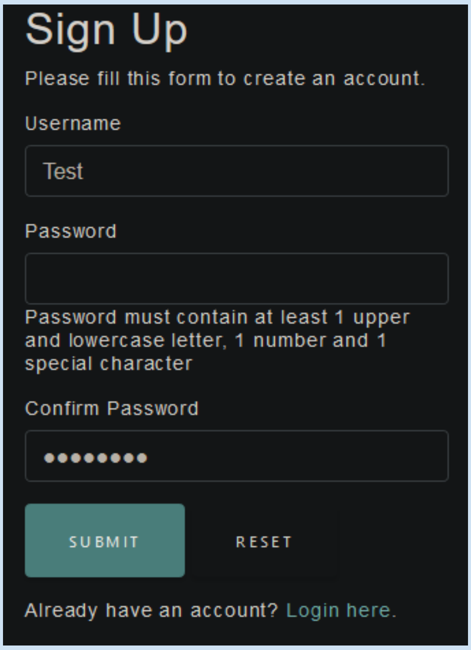
Input	ADMIN
Output	 Sign Up Please fill this form to create an account. Username This username is already taken. Password ●●●●●●●●● Confirm Password ●●●●●●●●● SUBMIT RESET Already have an account? Login here.
Outcome	Reuse of username not allowed
Test Result	Pass

Check if password complies with standards 1

Vulnerability tested	Check if password complies with standards
Input	username=Test password=pass

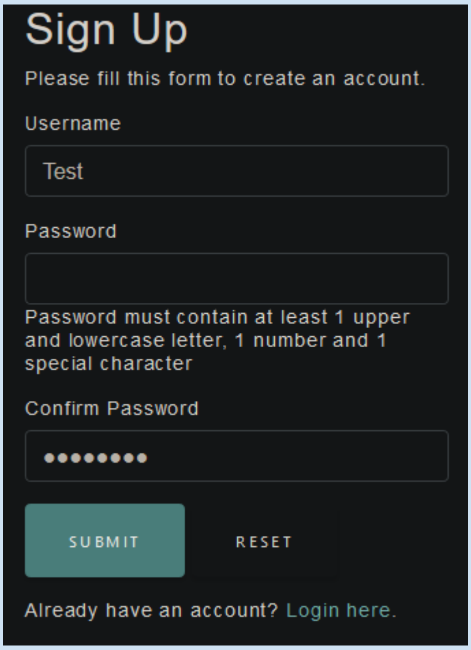
Output	 Sign Up Please fill this form to create an account. Username Test Password Password must have atleast 6 characters. Confirm Password •••• Already have an account? Login here.
Outcome	Password must be at least 6 characters long
Test Result	Pass

Check if password complies with standards 2

Vulnerability tested	Check if password complies with standards
Input	username=Test password=password
Output	 Sign Up Please fill this form to create an account. Username Test Password Password must contain at least 1 upper and lowercase letter, 1 number and 1 special character Confirm Password •••••••• Already have an account? Login here.
Outcome	Password must contain at least 1 upper and lowercase letter, 1

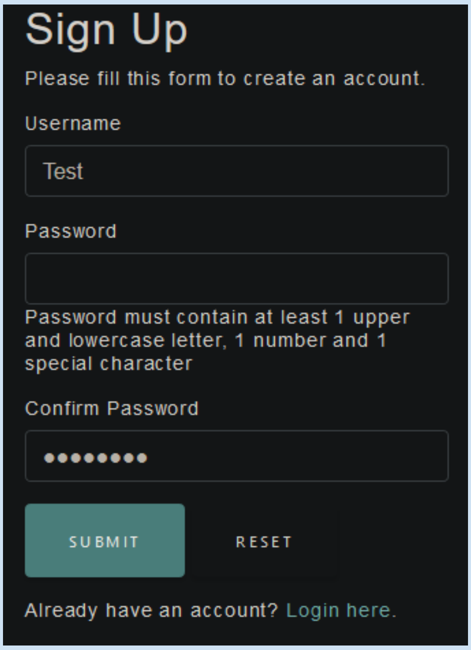
	number and 1 special character
Test Result	Pass

Check if password complies with standards 3

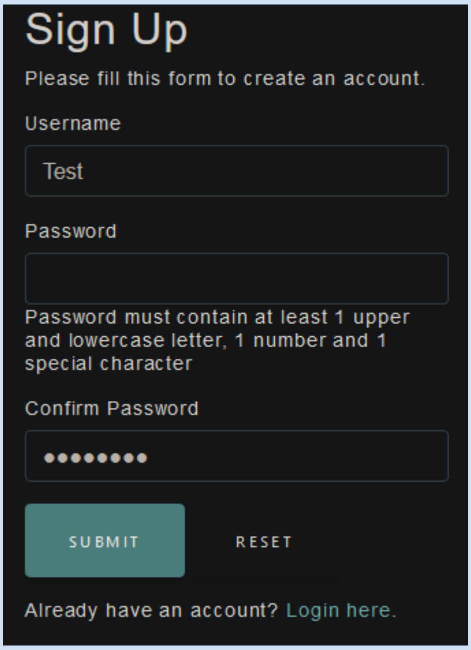
Vulnerability tested	Check if password complies with standards
Input	username=Test password=Password
Output	 Sign Up Please fill this form to create an account. Username Test Password Password must contain at least 1 upper and lowercase letter, 1 number and 1 special character Confirm Password •••••••• Already have an account? Login here.
Outcome	Password must contain at least 1 upper and lowercase letter, 1 number and 1 special character
Test Result	Pass

Check if password complies with standards 4

Vulnerability tested	Check if password complies with standards
Input	username=Test password=Password1

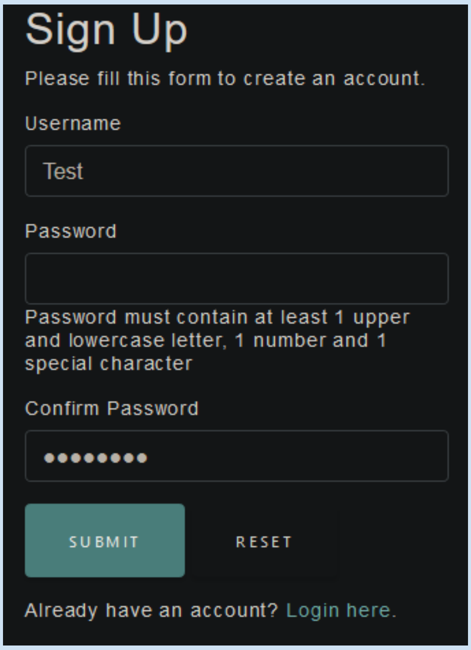
Output	 Sign Up Please fill this form to create an account. Username Test Password Password must contain at least 1 upper and lowercase letter, 1 number and 1 special character Confirm Password ••••••••
Outcome	Password must contain at least 1 upper and lowercase letter, 1 number and 1 special character
Test Result	Pass

Check if password complies with standards 5

Vulnerability tested	Check if password complies with standards
Input	username=Test password=Password!
Output	 Sign Up Please fill this form to create an account. Username Test Password Password must contain at least 1 upper and lowercase letter, 1 number and 1 special character Confirm Password ••••••••

Outcome	Password must contain at least 1 upper and lowercase letter, 1 number and 1 special character
Test Result	Pass

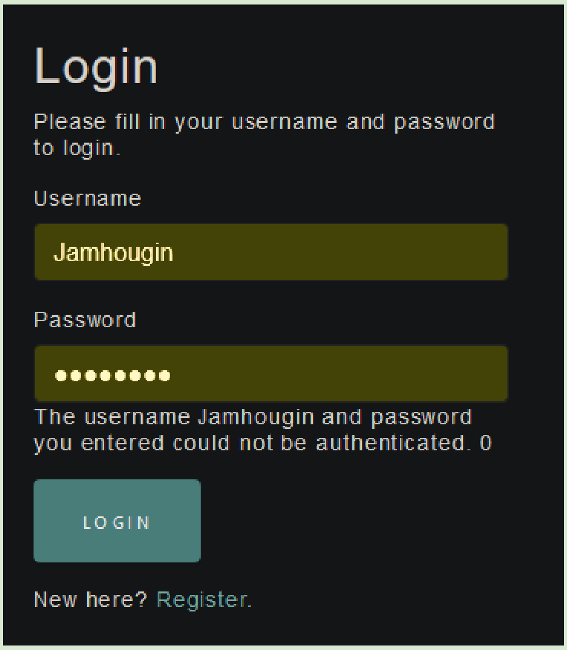
Check if password complies with standards 6

Vulnerability tested	Check if password complies with standards
Input	username=Test password=password1!
Output	 Sign Up Please fill this form to create an account. Username Test Password Password must contain at least 1 upper and lowercase letter, 1 number and 1 special character Confirm Password •••••••• Already have an account? Login here.
Outcome	Password must contain at least 1 upper and lowercase letter, 1 number and 1 special character
Test Result	Pass

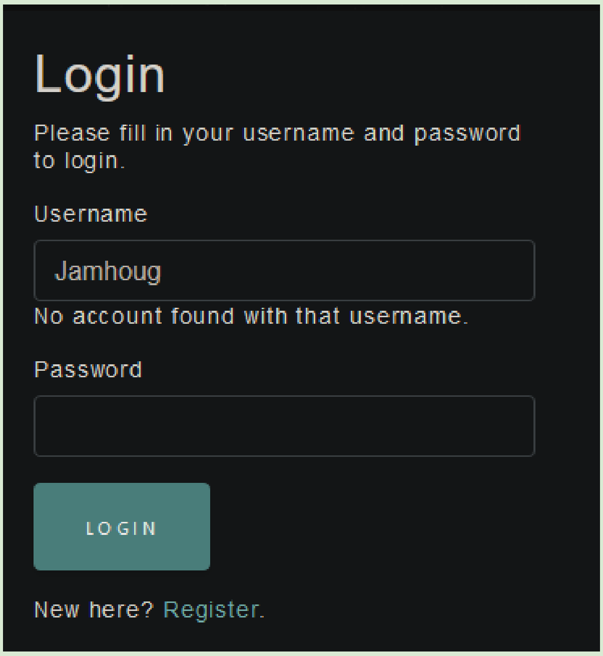
Login Page

Login with wrong password

Vulnerability tested	Login with wrong password
Input	wrongpassword

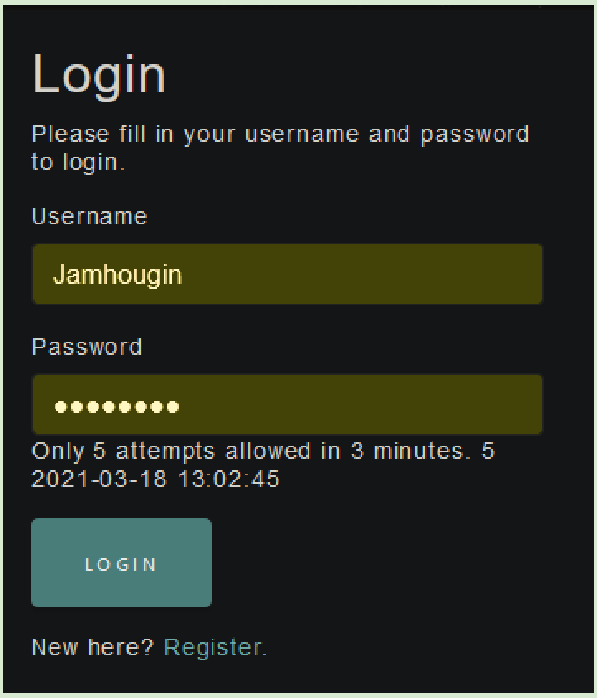
Output	 Login Please fill in your username and password to login. Username Jamhougin Password •••••••• The username Jamhougin and password you entered could not be authenticated. 0 LOGIN New here? Register.
Outcome	The username Jamhougin and password you entered could not be authenticated.
Test Result	Pass

Login with invalid username

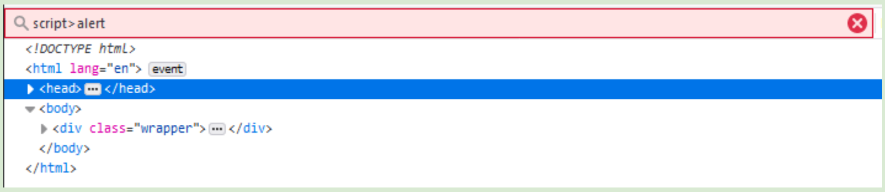
Vulnerability tested	Login with invalid username
Input	Jamhoug
Output	 Login Please fill in your username and password to login. Username Jamhoug No account found with that username. Password LOGIN New here? Register.

Outcome	No account found with that username
Test Result	Pass

Lockout after more than 5 attempts for 3 minutes

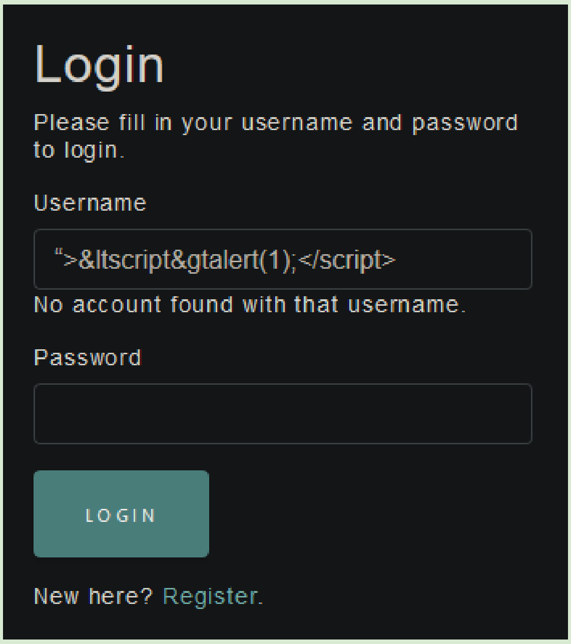
Vulnerability tested	Lockout after more than 5 attempts for 3 minutes
Input	Wrongpassword five times then correct password once.
Output	 Login Please fill in your username and password to login. Username Jamhougin Password •••••••• Only 5 attempts allowed in 3 minutes. 5 2021-03-18 13:02:45 LOGIN New here? Register.
Outcome	Only 5 attempts allowed.
Test Result	Pass

Xss using username field in address bar

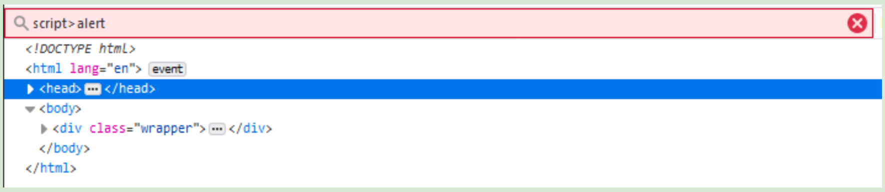
Vulnerability tested	Xss using username field in address bar
Input	http://localhost/ProjectCA/Login.php?username=%3Cscript%3Ealert(1);%3C/script%3E
Output	 script>alert <!DOCTYPE html> <html lang="en"> event <head> … </head> <body> <div class="wrapper"> … </div> </body> </html>

Outcome	The script was not triggered
Test Result	Pass

Xss using username text field

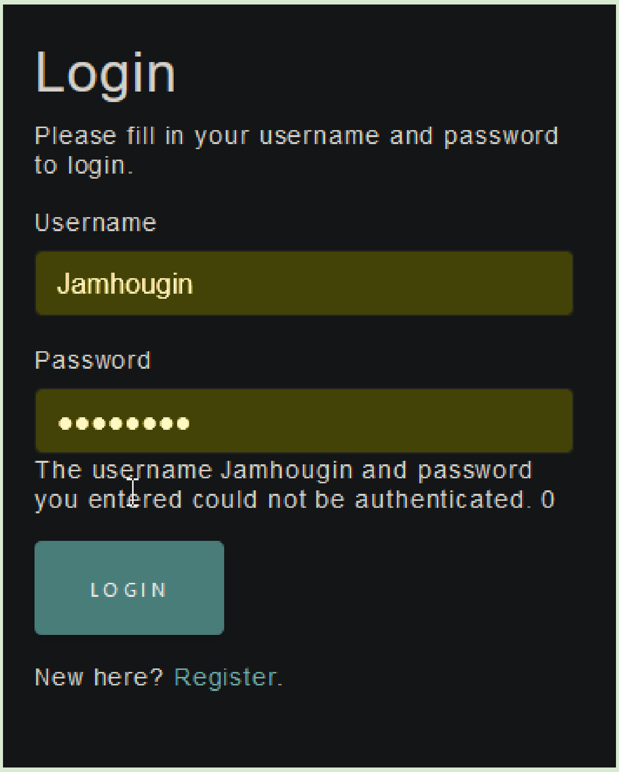
Vulnerability tested	Xss using username text field
Input	“><script>alert(1);</script>
Output	 Login Please fill in your username and password to login. Username “>&ltscript&gtalert(1);</script> No account found with that username. Password LOGIN New here? Register.
Outcome	The script was not triggered
Test Result	Pass

Xss using password field in address bar

Vulnerability tested	Xss using password field in address bar
Input	http://localhost/ProjectCA/Login.php?password=%3Cscript%3Ealert(1);%3C/script%3E
Output	 script>alert <!DOCTYPE html> <html lang="en"> event ▸ <head> ⋯ </head> ▾ <body> ▸ <div class="wrapper"> ⋯ </div> </body> </html>
Outcome	The script was not triggered

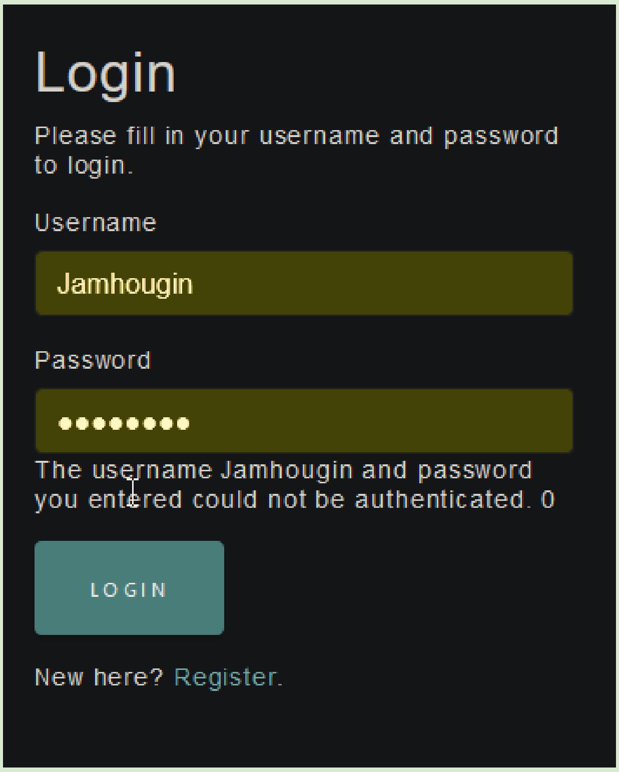
Test Result	Pass

Xss using password text field

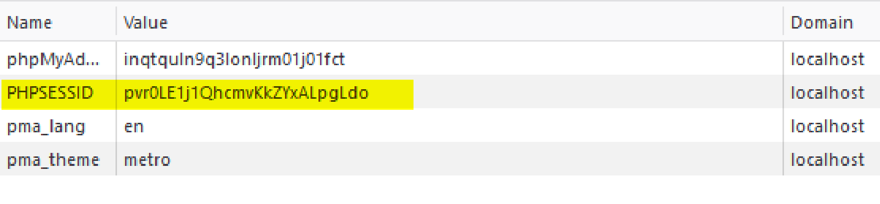
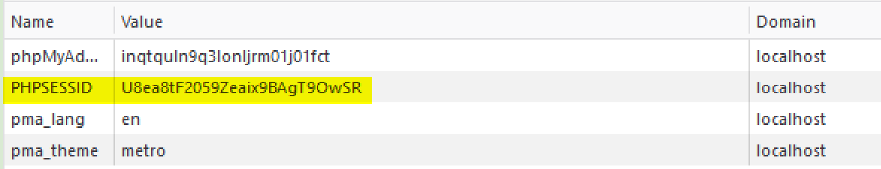
Vulnerability tested	Xss using password text field
Input	“><script>alert(1);</script>
Output	 Login Please fill in your username and password to login. Username Jamhougin Password ●●●●●●●●● The username Jamhougin and password you entered could not be authenticated. 0 LOGIN New here? Register.
Outcome	The script was not triggered
Test Result	Pass

Xss using password text field

Vulnerability tested	Xss using password text field
Input	%3Cbody%20onload=alert(1)%3E

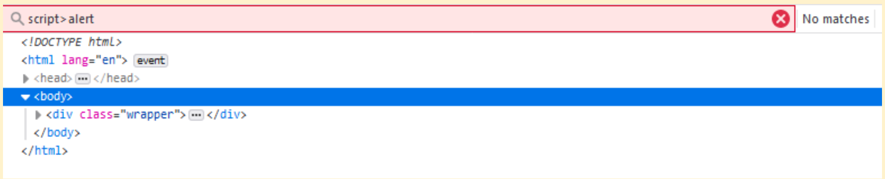
<table>
<tr><td>Output</td><td>
Login
Please fill in your username and password to login.
Username
Jamhougin
Password
●●●●●●●●
The username Jamhougin and password you entered could not be authenticated. 0
LOGIN
New here? Register.</td></tr>
<tr><td>Outcome</td><td>The script was not triggered</td></tr>
<tr><td>Test Result</td><td>Pass</td></tr>
</table>

Anonymous session destroyed and new session created

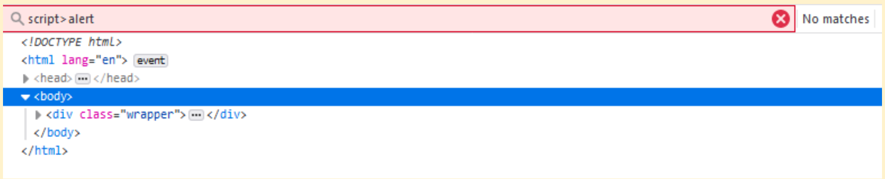
<table>
<tr><td>Vulnerability tested</td><td>Anonymous session destroyed and new session created</td></tr>
<tr><td>Input</td><td>Login as valid user</td></tr>
<tr><td>Output</td><td>Before login:

<table>
<tr><th>Name</th><th>Value</th><th>Domain</th></tr>
<tr><td>phpMyAd...</td><td>inqtquln9q3lonljrm01j01fct</td><td>localhost</td></tr>
<tr><td>PHPSESSID</td><td>pvr0LE1j1QhcmvKkZYxALpgLdo</td><td>localhost</td></tr>
<tr><td>pma_lang</td><td>en</td><td>localhost</td></tr>
<tr><td>pma_theme</td><td>metro</td><td>localhost</td></tr>
</table>
After login:

<table>
<tr><th>Name</th><th>Value</th><th>Domain</th></tr>
<tr><td>phpMyAd...</td><td>inqtquln9q3lonljrm01j01fct</td><td>localhost</td></tr>
<tr><td>PHPSESSID</td><td>U8ea8tF2059Zeaix9BAgT9OwSR</td><td>localhost</td></tr>
<tr><td>pma_lang</td><td>en</td><td>localhost</td></tr>
<tr><td>pma_theme</td><td>metro</td><td>localhost</td></tr>
</table></td></tr>
<tr><td>Outcome</td><td>Session id changed</td></tr>
<tr><td>Test Result</td><td>Pass</td></tr>
</table>

Reset Password Page

Xss using new_password field in address bar

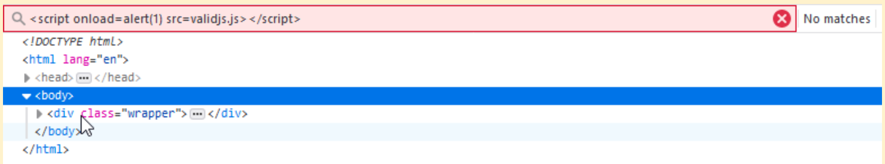
Vulnerability tested	Xss using new_password field in address bar
Input	http://localhost/ProjectCA/ResetPassword.php?new_password=%3Cscript%3Ealert(1);%3C/script%3E
Output	 script>alert No matches <!DOCTYPE html> <html lang="en"> event ▶ <head>…</head> ▼ <body> ▶ <div class="wrapper">…</div> </body> </html>
Outcome	The script was not triggered
Test Result	Pass

Xss using confirm_password field in address bar

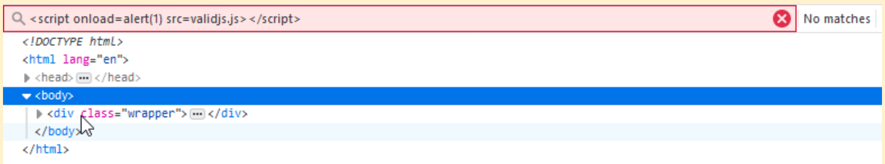
Vulnerability tested	Xss using confirm_password field in address bar
Input	http://localhost/ProjectCA/ResetPassword.php?confirm_password=%3Cscript%3Ealert(1);%3C/script%3E
Output	 script>alert No matches <!DOCTYPE html> <html lang="en"> event ▶ <head>…</head> ▼ <body> ▶ <div class="wrapper">…</div> </body> </html>
Outcome	The script was not triggered
Test Result	Pass

Xss using password field

Vulnerability tested	Xss using password field
Input	<script onload=alert(1) src=validjs.js></script>

Output	 <script onload=alert(1) src=validjs.js> </script> No matches <!DOCTYPE html> <html lang="en"> <head> ··· </head> <body> <div class="wrapper"> ··· </div> </body> </html>
Outcome	The script was not triggered
Test Result	Pass

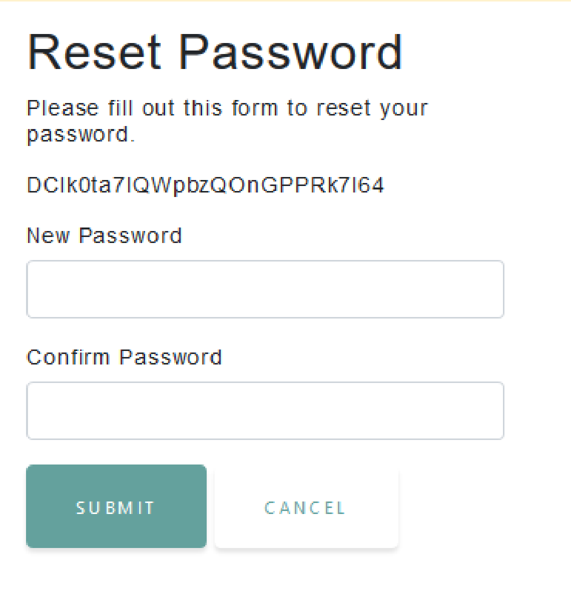
Xss using confirm_password field

Vulnerability tested	Xss using confirm_password field
Input	<script onload=alert(1) src=validjs.js></script>
Output	 <script onload=alert(1) src=validjs.js> </script> No matches <!DOCTYPE html> <html lang="en"> <head> ··· </head> <body> <div class="wrapper"> ··· </div> </body> </html>
Outcome	The script was not triggered
Test Result	Pass

CSRF exploit

Vulnerability tested	CSRF exploit
Input	

<table>
<tr>
<td>Output</td>
<td>



Reset Password

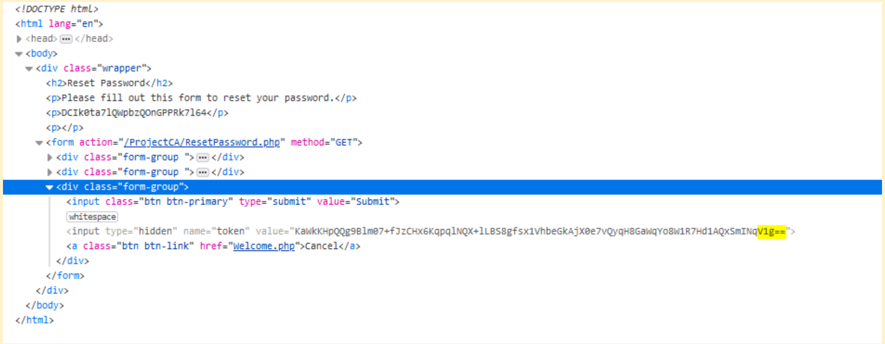
Please fill out this form to reset your password.

DCIk0ta7lQWpbzQOnGPPRk7l64

New Password

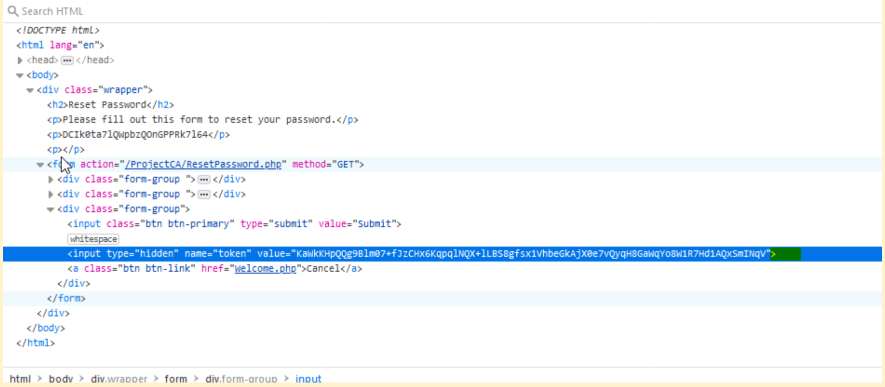
Confirm Password

The CSRF token is displayed on page only for the purpose of demonstration



```
<!DOCTYPE html>
<html lang="en">
▶ <head> ⋯ </head>
▼ <body>
  ▼ <div class="wrapper">
      <h2>Reset Password</h2>
      <p>Please fill out this form to reset your password.</p>
      <p>DCIk0ta7lQWpbzQOnGPPRk7l64</p>
      <p></p>
    ▼ <form action="/ProjectCA/ResetPassword.php" method="GET">
      ▶ <div class="form-group "> ⋯ </div>
      ▶ <div class="form-group "> ⋯ </div>
      ▼ <div class="form-group">
          <input class="btn btn-primary" type="submit" value="Submit">
          whitespace
          <input type="hidden" name="token" value="KaWkKHpQQg9Blm07+fJzCHx6KqpqlNQX+lLBS8gfsx1VhbeGkAjX0e7vQyqH8GaWqYo8W1R7Hd1AQxSmINqV1g==">
          <a class="btn btn-link" href="Welcome.php">Cancel</a>
        </div>
      </form>
    </div>
  </body>
</html>
```

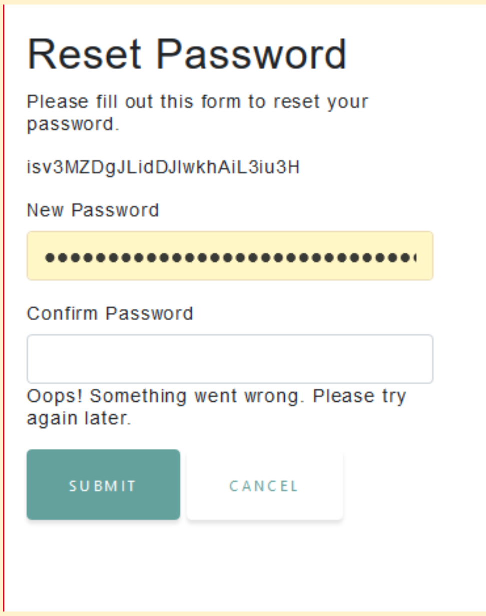
The token is changed here



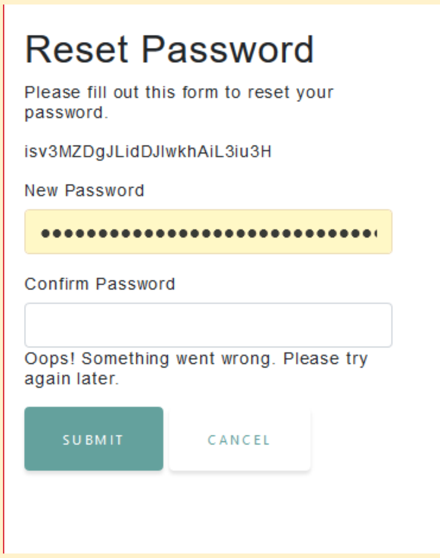
```
<!DOCTYPE html>
<html lang="en">
▶ <head> ⋯ </head>
▼ <body>
  ▼ <div class="wrapper">
      <h2>Reset Password</h2>
      <p>Please fill out this form to reset your password.</p>
      <p>DCIk0ta7lQWpbzQOnGPPRk7l64</p>
      <p></p>
    ▼ <form action="/ProjectCA/ResetPassword.php" method="GET">
      ▶ <div class="form-group "> ⋯ </div>
      ▶ <div class="form-group "> ⋯ </div>
      ▼ <div class="form-group">
          <input class="btn btn-primary" type="submit" value="Submit">
          whitespace
          <input type="hidden" name="token" value="KaWkKHpQQg9Blm07+fJzCHx6KqpqlNQX+lLBS8gfsx1VhbeGkAjX0e7vQyqH8GaWqYo8W1R7Hd1AQxSmINqV">
          <a class="btn btn-link" href="Welcome.php">Cancel</a>
        </div>
      </form>
    </div>
  </body>
</html>
```

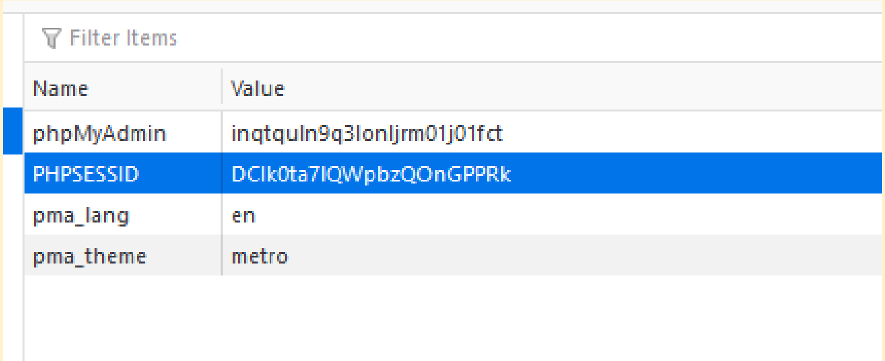
And an error message is displayed

</td>
</tr>
</table>

	 Reset Password Please fill out this form to reset your password. isv3MZDgJLidDJlwkhAiL3iu3H New Password Confirm Password Oops! Something went wrong. Please try again later.
Outcome	The password was not changed
Test Result	Pass

Successful password change and logout, changes session id

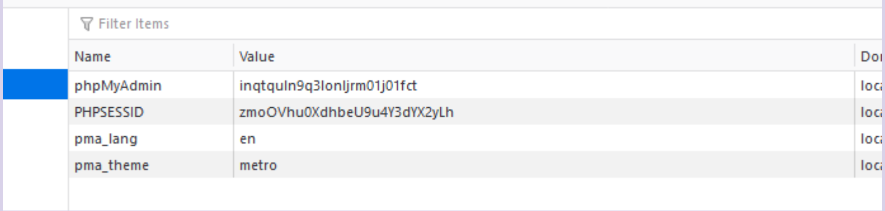
Vulnerability tested	Successful password change and logout, changes session id
Input	Successful password change
Output	 Reset Password Please fill out this form to reset your password. isv3MZDgJLidDJlwkhAiL3iu3H New Password Confirm Password Oops! Something went wrong. Please try again later. Session id before password change.

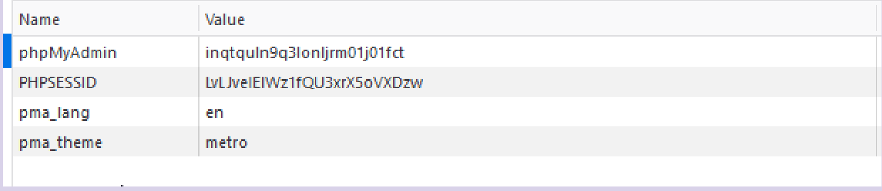
<table>
<tr><td></td><td>
<table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>phpMyAdmin</td><td>inqtquln9q3lonljrm01j01fct</td></tr>
<tr><td>PHPSESSID</td><td>DClk0ta7lQWpbzQOnGPPRk</td></tr>
<tr><td>pma_lang</td><td>en</td></tr>
<tr><td>pma_theme</td><td>metro</td></tr>
</table>

Season id on login page after password is changed.</td></tr>
<tr><td>Outcome</td><td>New session is created</td></tr>
<tr><td>Test Result</td><td>Pass</td></tr>
</table>

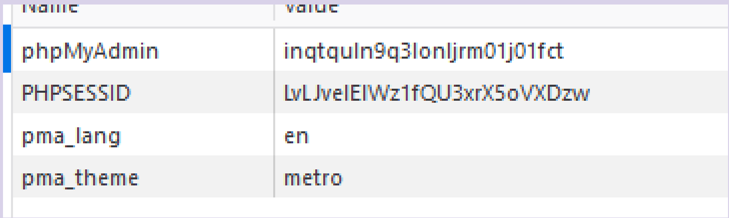
Welcome Page

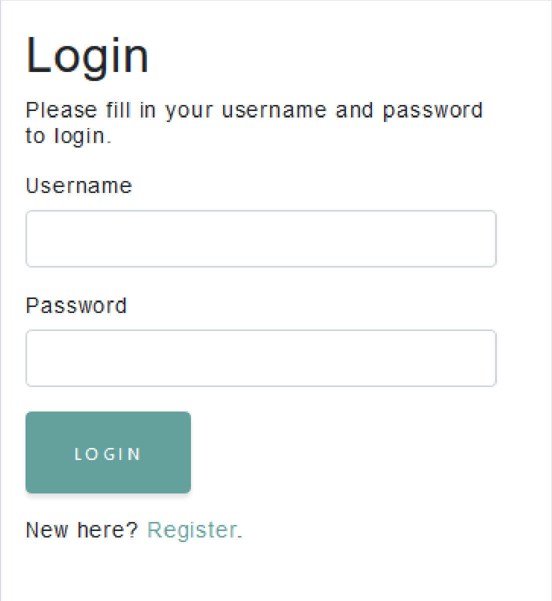
Logout, changes session id

<table>
<tr><td>Vulnerability tested</td><td>Logout, changes session id</td></tr>
<tr><td>Input</td><td>Click logout button</td></tr>
<tr><td>Output</td><td>
<table>
<tr><th>Name</th><th>Value</th><th>Do</th></tr>
<tr><td>phpMyAdmin</td><td>inqtquln9q3lonljrm01j01fct</td><td>loc</td></tr>
<tr><td>PHPSESSID</td><td>zmoOVhu0XdhbeU9u4Y3dYX2yLh</td><td>loc</td></tr>
<tr><td>pma_lang</td><td>en</td><td>loc</td></tr>
<tr><td>pma_theme</td><td>metro</td><td>loc</td></tr>
</table>

Session id before logout.

<table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>phpMyAdmin</td><td>inqtquln9q3lonljrm01j01fct</td></tr>
<tr><td>PHPSESSID</td><td>LvLJvelEIWz1fQU3xrX5oVXDzw</td></tr>
<tr><td>pma_lang</td><td>en</td></tr>
<tr><td>pma_theme</td><td>metro</td></tr>
</table>
Season id after logout.</td></tr>
<tr><td>Outcome</td><td>New session is created</td></tr>
<tr><td>Test Result</td><td>Pass</td></tr>
</table>

User is logged out after 10 minutes of inactivity

<table>
<tr><td>Vulnerability tested</td><td>User is logged out after 10 minutes of inactivity</td></tr>
<tr><td>Input</td><td>Do nothing for 10 minutes then try to navigate to page 2</td></tr>
<tr><td>Output</td><td>

<table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>phpMyAdmin</td><td>inqtquln9q3lonljrm01j01fct</td></tr>
<tr><td>PHPSESSID</td><td>LvLJvelElWz1fQU3xrX5oVXDzw</td></tr>
<tr><td>pma_lang</td><td>en</td></tr>
<tr><td>pma_theme</td><td>metro</td></tr>
</table>
Session id before redirect.


Login

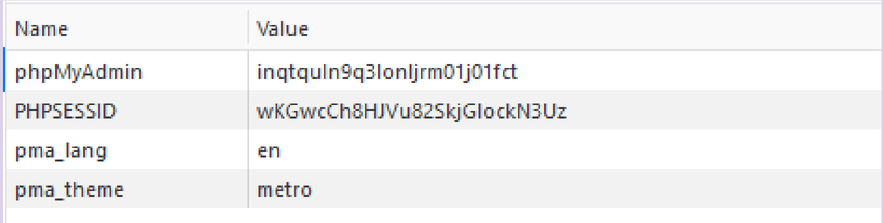
Please fill in your username and password to login.

Username

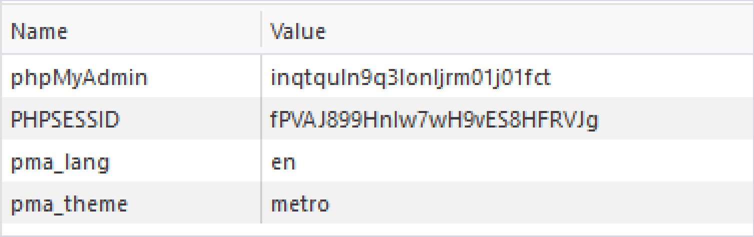
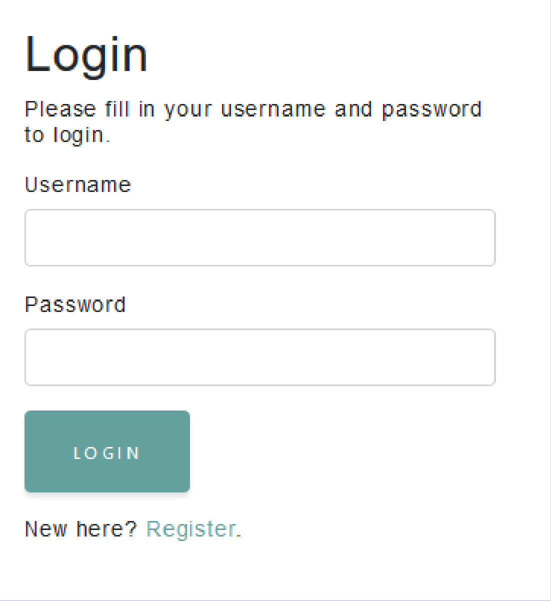
Password

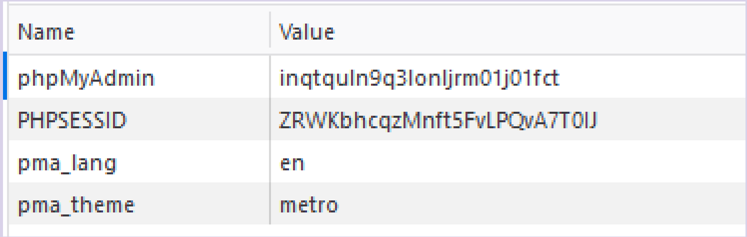
New here? Register.

User is redirected


<table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>phpMyAdmin</td><td>inqtquln9q3lonljrm01j01fct</td></tr>
<tr><td>PHPSESSID</td><td>wKGwcCh8HJVu82SkjGlockN3Uz</td></tr>
<tr><td>pma_lang</td><td>en</td></tr>
<tr><td>pma_theme</td><td>metro</td></tr>
</table>
Session id after redirect
</td></tr>
<tr><td>Outcome</td><td>User is directed to the login page.</td></tr>
<tr><td>Test Result</td><td>Pass</td></tr>
</table>

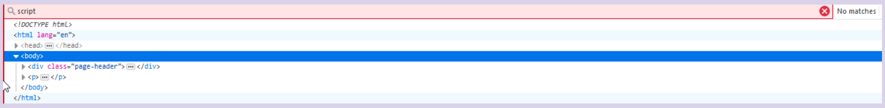
User is logged out after 1 hour

<table>
<tr><td>Vulnerability tested</td><td>User is logged out after 1 hour</td></tr>
<tr><td>Input</td><td>Navigate between pages for an hour and then navigate to page 2. (To test this the time required was changed to 30 seconds)</td></tr>
<tr><td>Output</td><td>

<table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>phpMyAdmin</td><td>inqtquln9q3lonljrm01j01fct</td></tr>
<tr><td>PHPSESSID</td><td>fPVAJ899Hnlw7wH9vES8HFRVJg</td></tr>
<tr><td>pma_lang</td><td>en</td></tr>
<tr><td>pma_theme</td><td>metro</td></tr>
</table>
Session id before redirect.

Login
Please fill in your username and password to login.
Username
Password

New here? Register.
User is redirected

<table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>phpMyAdmin</td><td>inqtquln9q3lonljrm01j01fct</td></tr>
<tr><td>PHPSESSID</td><td>ZRWKbhcqzMnft5FvLPQvA7T0IJ</td></tr>
<tr><td>pma_lang</td><td>en</td></tr>
<tr><td>pma_theme</td><td>metro</td></tr>
</table>
Session id after redirect</td></tr>
<tr><td>Outcome</td><td>User is directed to the login page.</td></tr>
<tr><td>Test Result</td><td>Pass</td></tr>
</table>

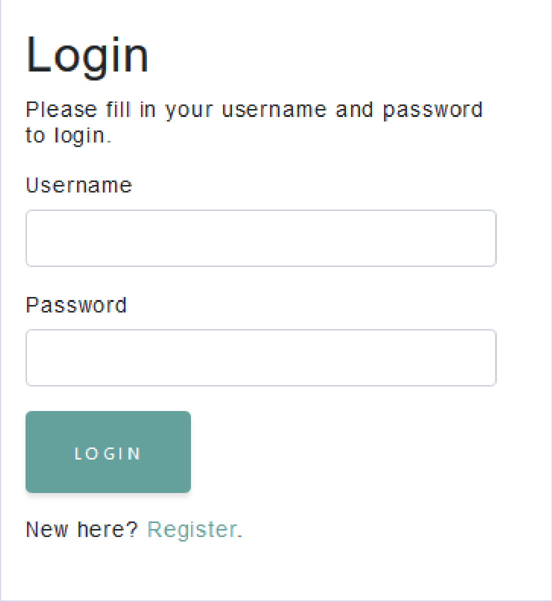
Xss using username field in address bar

Vulnerability tested	Xss using username field in address bar
Input	http://localhost/ProjectCA/Welcome.php?username=%3Cscript%3Ealert(1);%3C/script%3E

Output	
	script
	No matches
	`<!DOCTYPE html>`
	`<html lang="en">`
	`<head>…</head>`
	`<body>`
	`<div class="page-header">…</div>`
	`<p>…</p>`
	`</body>`
	`</html>`

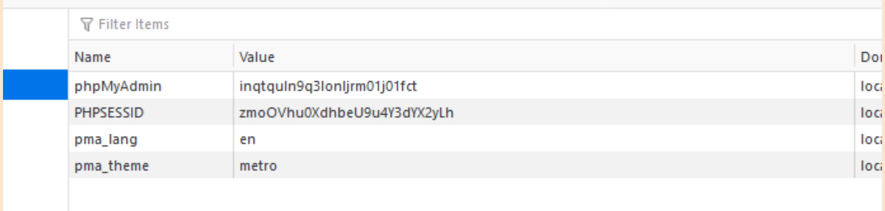
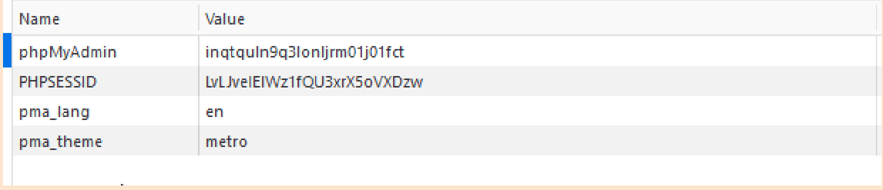
Outcome	Script was not injected
Test Result	Pass

Attempt to access the page using url, while not logged in

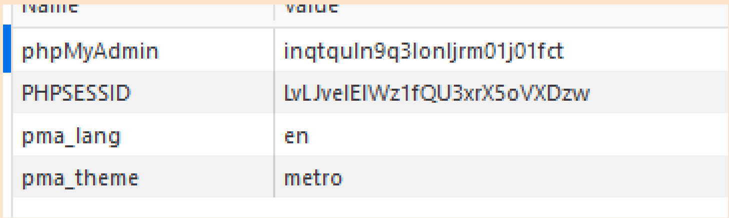
Vulnerability tested	Attempt to access the page using url, while not logged in with a valid session.
Input	http://localhost/ProjectCA/Welcome.php
Output	 Login Please fill in your username and password to login. Username Password LOGIN New here? Register.
Outcome	Page was not accessed
Test Result	Pass

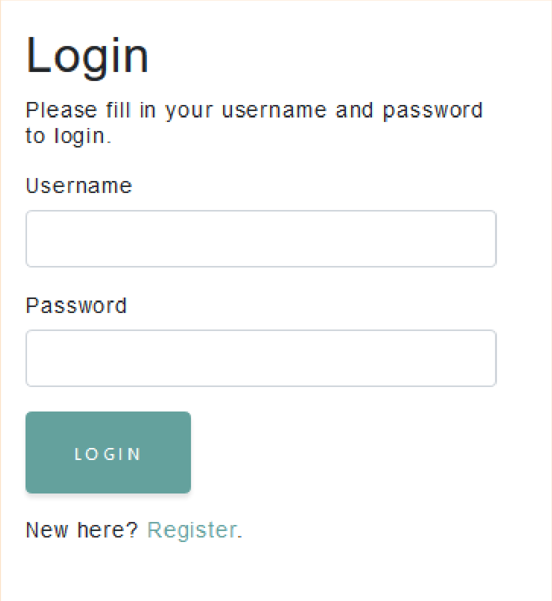
Page 2

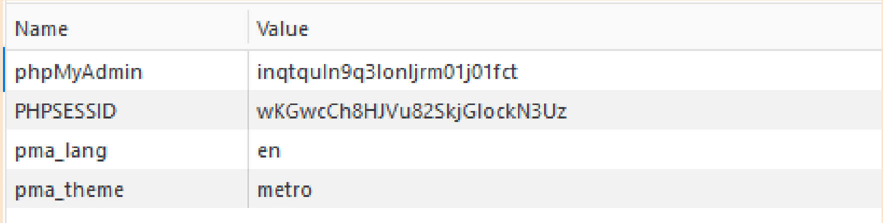
Logout, changes session id

Vulnerability tested	Logout, changes session id
Input	Click logout button
Output	 Filter Items Name · Value · Do phpMyAdmin · inqtquln9q3lonljrm01j01fct · loc PHPSESSID · zmoOVhu0XdhbeU9u4Y3dYX2yLh · loc pma_lang · en · loc pma_theme · metro · loc Session id before logout.  Name · Value phpMyAdmin · inqtquln9q3lonljrm01j01fct PHPSESSID · LvLJvelEIWz1fQU3xrX5oVXDzw pma_lang · en pma_theme · metro Season id after logout.
Outcome	New session is created
Test Result	Pass

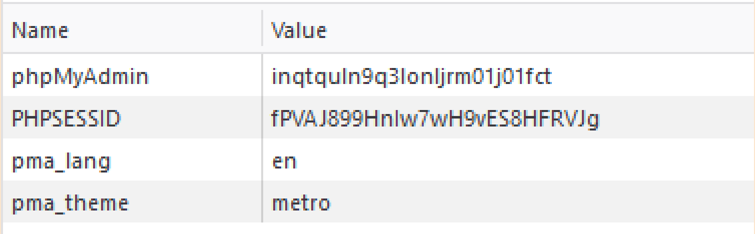
User is logged out after 10 minutes of inactivity

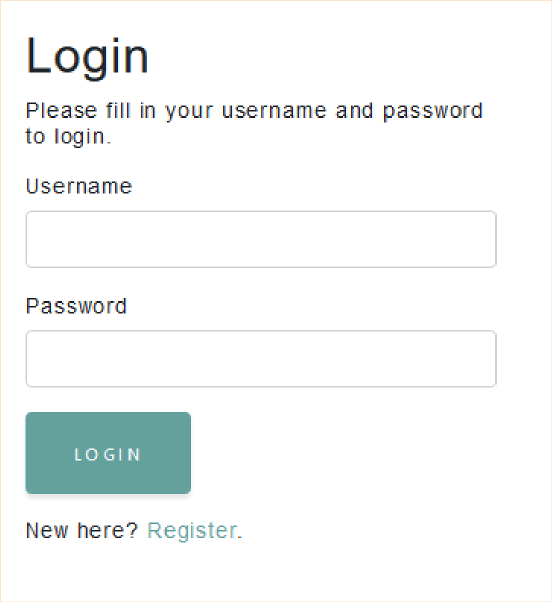
Vulnerability tested	User is logged out after 10 minutes of inactivity
Input	Do nothing for 10 minutes then try to navigate to page 2
Output	 Name · Value phpMyAdmin · inqtquln9q3lonljrm01j01fct PHPSESSID · LvLJvelEIWz1fQU3xrX5oVXDzw pma_lang · en pma_theme · metro Session id before redirect.

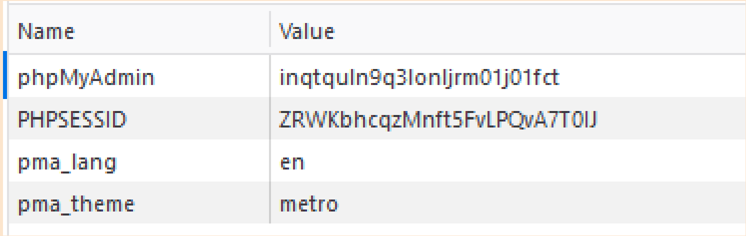
<table>
<tr><td></td><td>

Login
Please fill in your username and password to login.
Username
Password

New here? Register.
User is redirected

<table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>phpMyAdmin</td><td>inqtquln9q3lonljrm01j01fct</td></tr>
<tr><td>PHPSESSID</td><td>wKGwcCh8HJVu82SkjGlockN3Uz</td></tr>
<tr><td>pma_lang</td><td>en</td></tr>
<tr><td>pma_theme</td><td>metro</td></tr>
</table>
Session id after redirect
</td></tr>
<tr><td>Outcome</td><td>User is directed to the login page.</td></tr>
<tr><td>Test Result</td><td>Pass</td></tr>
</table>

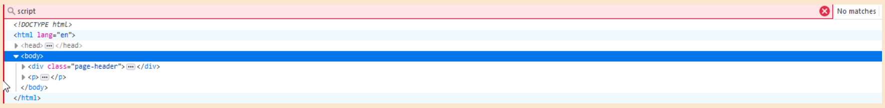
User is logged out after 1 hour

<table>
<tr><td>Vulnerability tested</td><td>User is logged out after 1 hour</td></tr>
<tr><td>Input</td><td>Navigate between pages for an hour and then navigate to page 2. (To test this the time required was changed to 30 seconds)</td></tr>
<tr><td>Output</td><td>

<table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>phpMyAdmin</td><td>inqtquln9q3lonljrm01j01fct</td></tr>
<tr><td>PHPSESSID</td><td>fPVAJ899Hnlw7wH9vES8HFRVJg</td></tr>
<tr><td>pma_lang</td><td>en</td></tr>
<tr><td>pma_theme</td><td>metro</td></tr>
</table>
Session id before redirect.
</td></tr>
</table>

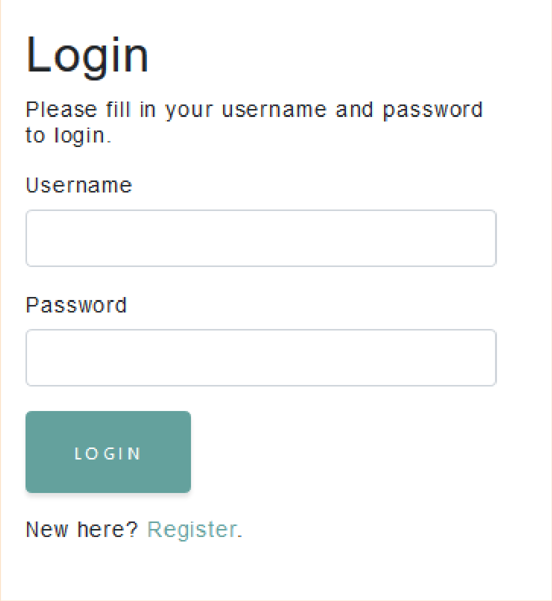
<table>
<tr><td></td><td>
Login
Please fill in your username and password to login.
Username
Password

New here? Register.
User is redirected

<table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>phpMyAdmin</td><td>inqtquln9q3lonljrm01j01fct</td></tr>
<tr><td>PHPSESSID</td><td>ZRWKbhcqzMnft5FvLPQvA7T0IJ</td></tr>
<tr><td>pma_lang</td><td>en</td></tr>
<tr><td>pma_theme</td><td>metro</td></tr>
</table>
Session id after redirect</td></tr>
<tr><td>Outcome</td><td>User is directed to the login page.</td></tr>
<tr><td>Test Result</td><td>Pass</td></tr>
</table>

Xss using username field in address bar

Vulnerability tested	Xss using username field in address bar
Input	http://localhost/ProjectCA/Welcome.php?username=%3Cscript%3Ealert(1);%3C/script%3E
Output	 script No matches <!DOCTYPE html> <html lang="en"> <head>…</head> <body> <div class="page-header">…</div> <p>…</p> </body> </html>
Outcome	Script was not injected
Test Result	Pass

Attempt to access the page using url, while not logged in

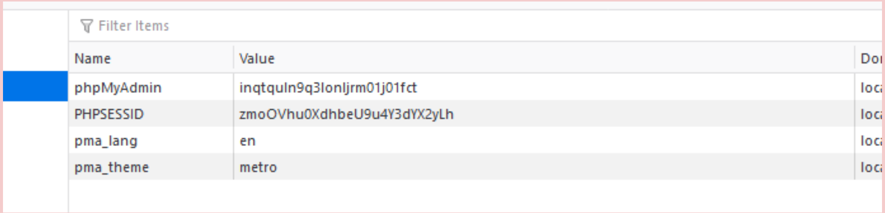
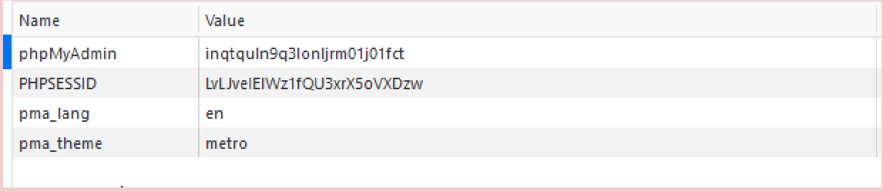
Vulnerability tested	Attempt to access the page using url, while not logged in with a valid session.
Input	http://localhost/ProjectCA/Welcome.php
Output	 Login Please fill in your username and password to login. Username Password New here? Register.
Outcome	Page was not accessed
Test Result	Pass

Admin Page

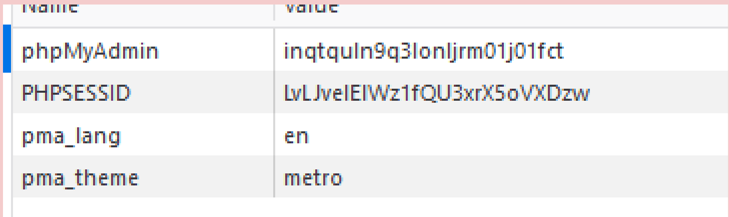
Non admin attempts to access page

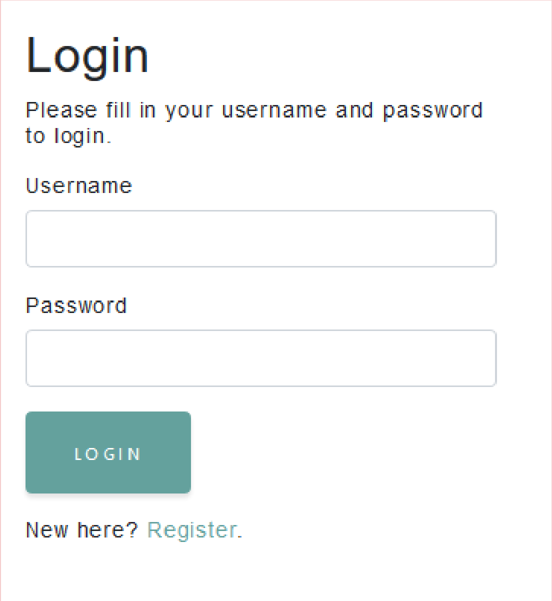
Vulnerability tested	Non admin attempts to access page
Input	Click Admin button
Output	Nothing happens
Outcome	Nothing happens
Test Result	Pass

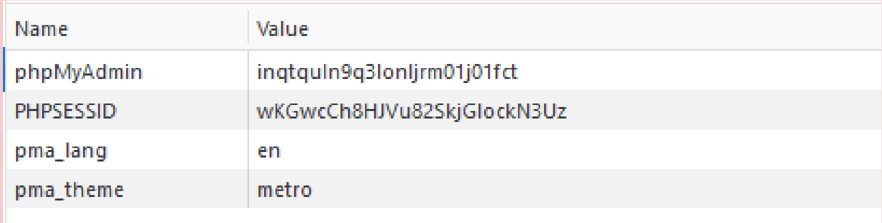
Logout, changes session id

<table>
<tr><td>Vulnerability tested</td><td>Logout, changes session id</td></tr>
<tr><td>Input</td><td>Click logout button</td></tr>
<tr><td>Output</td><td>
<table>
<tr><th>Name</th><th>Value</th><th>Do</th></tr>
<tr><td>phpMyAdmin</td><td>inqtquln9q3lonljrm01j01fct</td><td>loc</td></tr>
<tr><td>PHPSESSID</td><td>zmoOVhu0XdhbeU9u4Y3dYX2yLh</td><td>loc</td></tr>
<tr><td>pma_lang</td><td>en</td><td>loc</td></tr>
<tr><td>pma_theme</td><td>metro</td><td>loc</td></tr>
</table>
Session id before logout.

<table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>phpMyAdmin</td><td>inqtquln9q3lonljrm01j01fct</td></tr>
<tr><td>PHPSESSID</td><td>LvLJvelEIWz1fQU3xrX5oVXDzw</td></tr>
<tr><td>pma_lang</td><td>en</td></tr>
<tr><td>pma_theme</td><td>metro</td></tr>
</table>
Season id after logout.</td></tr>
<tr><td>Outcome</td><td>New session is created</td></tr>
<tr><td>Test Result</td><td>Pass</td></tr>
</table>

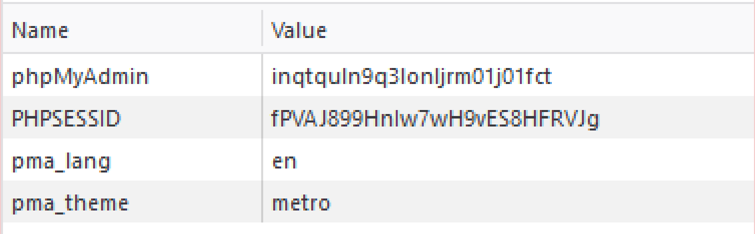
User is logged out after 10 minutes of inactivity

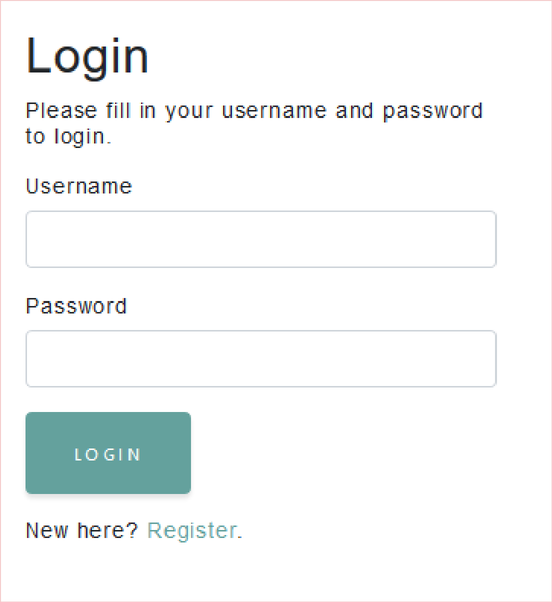
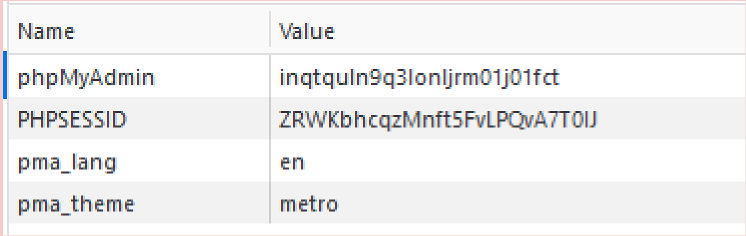
<table>
<tr><td>Vulnerability tested</td><td>User is logged out after 10 minutes of inactivity</td></tr>
<tr><td>Input</td><td>Do nothing for 10 minutes then try to navigate to page 2</td></tr>
<tr><td>Output</td><td>
<table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>phpMyAdmin</td><td>inqtquln9q3lonljrm01j01fct</td></tr>
<tr><td>PHPSESSID</td><td>LvLJvelEIWz1fQU3xrX5oVXDzw</td></tr>
<tr><td>pma_lang</td><td>en</td></tr>
<tr><td>pma_theme</td><td>metro</td></tr>
</table>
Session id before redirect.</td></tr>
</table>

<table>
<tr><td></td><td>
Login
Please fill in your username and password to login.
Username
Password

New here? Register.
User is redirected

<table><tr><th>Name</th><th>Value</th></tr><tr><td>phpMyAdmin</td><td>inqtquln9q3lonljrm01j01fct</td></tr><tr><td>PHPSESSID</td><td>wKGwcCh8HJVu82SkjGlockN3Uz</td></tr><tr><td>pma_lang</td><td>en</td></tr><tr><td>pma_theme</td><td>metro</td></tr></table>
Session id after redirect</td></tr>
<tr><td>Outcome</td><td>User is directed to the login page.</td></tr>
<tr><td>Test Result</td><td>Pass</td></tr>
</table>

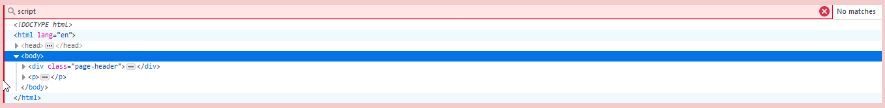
User is logged out after 1 hour

<table>
<tr><td>Vulnerability tested</td><td>User is logged out after 1 hour</td></tr>
<tr><td>Input</td><td>Navigate between pages for an hour and then navigate to page 2. (To test this the time required was changed to 30 seconds)</td></tr>
<tr><td>Output</td><td>
<table><tr><th>Name</th><th>Value</th></tr><tr><td>phpMyAdmin</td><td>inqtquln9q3lonljrm01j01fct</td></tr><tr><td>PHPSESSID</td><td>fPVAJ899Hnlw7wH9vES8HFRVJg</td></tr><tr><td>pma_lang</td><td>en</td></tr><tr><td>pma_theme</td><td>metro</td></tr></table>
Session id before redirect.</td></tr>
</table>

	 Login Please fill in your username and password to login. Username Password LOGIN New here? Register. User is redirected  Name Value phpMyAdmin inqtquln9q3lonljrm01j01fct PHPSESSID ZRWKbhcqzMnft5FvLPQvA7T0IJ pma_lang en pma_theme metro Session id after redirect
Outcome	User is directed to the login page.
Test Result	Pass

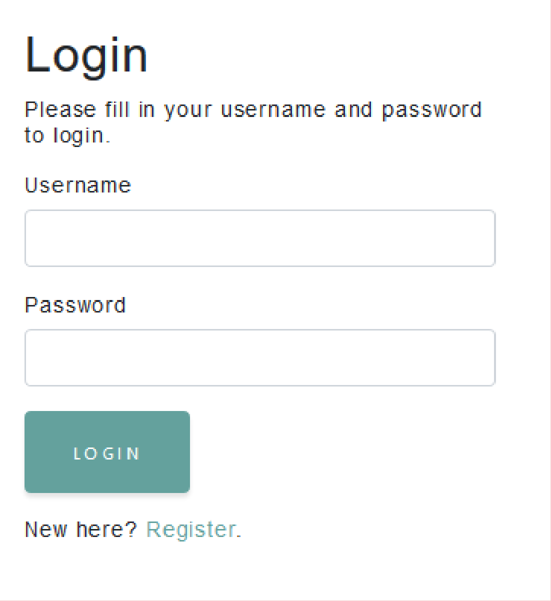
Xss using username field in address bar

Vulnerability tested	Xss using username field in address bar
Input	http://localhost/ProjectCA/Welcome.php?username=%3Cscript%3E alert(1);%3C/script%3E

Output	
	script No matches
	`<!DOCTYPE html>`
	`<html lang="en">`
	`<head>…</head>`
	`<body>`
	`<div class="page-header">…</div>`
	`<p>…</p>`
	`</body>`
	`</html>`

Outcome	Script was not injected
Test Result	Pass

Attempt to access the page using url, while not logged in

Vulnerability tested	Attempt to access the page using url, while not logged in with a valid session.
Input	http://localhost/ProjectCA/Welcome.php
Output	 Login Please fill in your username and password to login. Username Password LOGIN New here? Register.
Outcome	Page was not accessed
Test Result	Pass

Conclusion

While it is the opinion of the author that the project specifications have been met, it is clear that a fully secure application is perhaps not achievable. Cross site scripting alone is complex and there are many ways to exploit a system. As MySQL does not allow for mutli queries and all queries to the database involve “SELECT” and “UPDATE” statements it would not be possible to delete tables from the database, but it would be possible through “AND” statements to perhaps get information from the database and this could have been explored in more detail.

There were also a host of exploits of moderate severity highlighted by OWASP, not covered by the project spec that would be of interest to explore moving forward. The testing, while useful to achieve an understanding of the security issues being explored also begs the question of “How much testing is enough?”.

References

[1] TutorialRepublic.com (2021) online available at: https://www.tutorialrepublic.com/php-tutorial/php-mysql-login-system.php accessed on: 20th January 2021