

Part 1: Security test planning and execution

A. Time and True Positives

Total time: 7 hours 21 minutes

Vulnerabilities found: 5

True Positives Found Per Hour: 1 every 88 minutes (1 hour and 28 minutes)

B. Test planning

1) ASVS V4.2 Operation Level Access Control

Unique ID: 4.2.2

CWE 352: Cross-Site Request Forgery (CSRF)

Repeatable steps:

1. Log in to the application and navigate to a state-changing operation, in this case changing the password.
2. Submit an empty password change form.
3. Examine the request to obtain the CSRF token. You can find this by inspecting the "Network" tab under "Developer Options". In the listed requests under "File" find the one named "user_info_ajax.php" and click it. Then, in the menu for that request, change to the "Request" tab to see the "Form data" which will contain the field "csrf_token_form". (Make sure to copy it down as it is needed in step).
4. Log out and then log back into the application and navigate back to the password change form and submit an empty password form again.
5. Examine the request, following the steps we took in step 3, to ensure the CSRF token is different.
6. Test the previous CSRF token to ensure it is not accepted by going to the console tab and entering the following (make sure to enter the current password, the password you are wanting to change it to, and the CSRF token from your previous login attempt):

```
fetch('http://localhost/interface/usergroup/user_info_ajax.php', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/x-www-form-urlencoded',
  },
  body: 'curPass=[current_password]&newPass=[new_password]
&newPass2=[new_password_repeat]&csrf_token_form=[old_csrf_token]',
  credentials: 'include'
}).then(response => response.text())
```

```
.then(data => console.log(data))  
.catch((error) => console.error('Error:', error));
```

Expected results:

1. The console output in the browser should show the application rejecting the request for state-changing operations that do not include a valid CSRF token.

2) **ASVS V3.4 Cookie-based Session Management**

Unique ID: 3.4.2

CWE 1004: Sensitive Cookie
Without 'HttpOnly' Flag

Repeatable steps:

1. Log into the application.
2. Access developer tools and navigate to either the 'Application' option if using Chrome or the 'Storage' option if using Firefox.
3. Locate the 'Cookies' section and select 'http://localhost'.
4. The JWT-SESSION should have a checkmark for HTTPOnly.
5. Access the console tab and enter: `console.log(document.cookie)`
6. Verify that you are unable to see 'JWT-SESSION' which indicates that the client is unable to access its value.

Expected results:

1. Verify that cookies containing sensitive data have the 'HttpOnly' flag set to prevent client-side scripts from accessing the data.

3) **ASVS V14.4 HTTP Security Headers**

Unique ID: 14.4.7

CWE 1021: Improper Restriction of
Rendered UI Layers or Frames.

Repeatable steps:

1. Open your Network tab in the developer options.
2. Change the request to the "Doc" tab, or depending on your browser, to the "Headers" tab.
3. Manually access the page by entering into your URL bar: `localhost:80`.

4. You should see under the "File" tab in the "Network" menu of developer options a file named "login.php?site=default".

5. Click on the file named "login.php?site=default" and look in the Headers tab of that request for "Content-Security-Policy" and "X-Frame-Options". You should see "frame-ancestors 'none'" in Content-Security-Policy and "DENY" in X-Frame-Options.

6. Create a simple text file on your computer and enter the following:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Clickjacking Test</title>
  </head>
  <body>
    <iframe src="http://localhost" width="800" height="600"></iframe>
  </body>
</html>
```

7. Save the file as something like "test.html". The extension must be .html

8. Open the file so that it opens in your web browser. The iframe should not load.

Expected results:

1. That the iframe fails to load on the webpage.

4) ASVS V5.3 Output Encoding and Injection Prevention

Unique ID: 5.3.3

CWE 79: Improper Neutralization of Input During Page Generation ('XSS')

Repeatable steps:

1. Log into the application.

2. At the top in the search bar, enter and submit: <script>alert('XSS')</script>

3. In the same search bar, enter and submit: .

4. In the same search bar, enter and submit:
%3Cscript%3Ealert('XSS')%3C/script%3E

Expected results:

1. All of the different attempts all have the same results and do not run any of the scripts submitted.

5) ASVS V6.2 Algorithms

Unique ID: 6.2.1

CWE 310: Weaknesses in this category are related to the use of cryptography.

Repeatable steps:

1. Logout of the application if you are logged in and access the sign in page.
2. Enter the username and an incorrect password and submit the form.
3. Inspect the network traffic and the request that shows "login.php?site=default"

Expected results:

1. You should not see any identifying information from the server such as: encrypted user information, error logs or messages providing insight from the server, etc. In the response, we should see the html page being returned.

6) ASVS V9.1 Client Communication Security

Unique ID: 9.1.1

CWE 319: The software transmits sensitive or security-critical data in cleartext in a communication channel that can be sniffed by unauthorized actors.

Repeatable steps:

1. Specifically connect to "http://localhost" and access the sign in page.
2. Access the developer options. If using Google Chrome, in the "Network" tab of developer options, make sure to check the box for "Preserve Log" and "Disable cache". If using Firefox, in the "Network" tab of developer options, make sure to check the box for "Disable cache" and then click the settings button, located to the right of "No Throttling", and make sure that "Persist Logs" is checked. (Note, you may need to use Chrome or Firefox to find the request).
3. Submit the username and password incorrectly and inspect the request in the Network tab of the developer tools. You should see a "POST" request and the "File" name should be called "main_screen.php?auth=login&site=default". Click that file to inspect it.

4. If using Firefox, you will need to click the tab called "Request" and in the "Form data" you should see the username and password entered. If using Chrome, you will need to select the "Payload" tab to find the "Form Data".

Expected results:

1. You should not see any identifying information from the server.
2. The request should use TLS or redirect to HTTPS.

7) ASVS V3.1 Fundamental Session Management Security

Unique ID: 3.1.1

CWE 598: Use of GET Request
Method With Sensitive Query Strings

Repeatable steps:

1. Log into the application with a username and password
2. Check the address bar for any visible session tokens

Expected results:

1. After a login, no session tokens should be in the url

8) ASVS V3.2 Session Binding

Unique ID: 3.2.2

CWE 331: Insufficient Entropy

Repeatable steps:

1. Log into the application with a username and password
2. Check the address bar for any visible session tokens
3. Count the number characters in any visible token and verify that it is at least 64 bits

Expected results:

1. Token should be at least 64 bits

9) ASVS V2.1 Password Security

Unique ID: 2.1.1

CWE 521: Weak Password

Requirements

Repeatable steps:

1. Log into the application with a username and password
2. From the Account Icon menu, select "Change Password"
3. In the "Current Password" field, enter your password
4. In the new password and repeat new password fields, type the following:
aB!123456
5. Click "Save Changes"
6. Using developer tools, view the network request to
"http://localhost:3000/interface/usergroup/user_info_ajax.php"
7. In the response tab, verify that you see the message "Password change successful"

Expected results:

1. Password with less than 12 characters should not be allowed

10) ASVS V3.7 Defenses Against Session Management Exploits

Unique ID: 3.7.1

CWE 306: Missing Authentication for Critical Function

Repeatable steps:

1. Log into the application with a username and password
2. In the top menu, click on Messages
3. Click "Add New"
4. Logout of the application
5. Click the back arrow in the browser to verify that you were redirected to the login screen instead of the Add Message Screen.

Expected results:

1. Should not be able to add a message after logging out

11) ASVS V5.1 Input Validation

Unique ID: 5.1.3

CWE 20: Improper Input Validation

Repeatable steps:

1. Log into the application with a username and password
2. On the calendar, click on the 8:00 time
3. Highlight the date in the Date field
4. Type letters and press enter
5. Verify that you are not allowed to add letters in the date field

Expected results:

1. Letters are not allowed in the Date field

12) ASVS V3.3 Session Termination

Unique ID: 3.3.1

CWE 613: Insufficient Session Expiration

Repeatable steps:

1. Log into the application with a username and password
2. Log out of the application
3. Click the back button in the browser
4. Verify that you are returned to the the login screen and not the authenticated user home screen

Expected results:

1. Should be returned to login screen

13) ASVS V5.2 Sanitizing and Sandboxing

Unique ID: 5.2.4

CWE 95: Improper Neutralization of special elements

Repeatable steps:

1. Log into the application using a valid username and password.
2. Choose the "Patient" option and click on "NEW/Search."
3. In the "NAME" and "LastName" registration fields, type the characters "(abdf)."
4. Click on "Create a new patient."
5. Verify that the patient is displayed with the inputted unusual characters and ensure the system recognizes and displays the patient as valid

Expected results:

1. Valid inputs should be processed successfully. However, inputs containing potentially harmful features, such as 'eval()' or other dynamic code execution features, should be rejected to prevent security risks

14) ASVS V12.1 File Upload

Unique ID: 12.1.1

CWE 400: Uncontrolled Resource Consumption

Repeatable steps:

1. Log into the application using a valid username and password.
2. Choose the "Patient" option and click on "Dashboard."
3. Select "Document" and choose the categories from the left corner blue toolbar.
4. Choose "Files" and press the blue upload button.
5. Verify that the uploaded image is visible in the left corner bar under the patient information category.

Expected results:

1. Should not be allowed to upload an image with a size larger than 20MB while maintaining the quality of the uploaded image.

15) ASVS V8.3 Sensitive Private Data

Unique ID: 8.3.2

CWE 212: Improper Removal of Sensitive Information Before Storage or Transfer

Repeatable steps:

1. Log into the application using a valid username and password.
2. Choose the "Patient" option and click on NEW/SEARCH.
3. Click on the Search button.
4. Select the intended patient.
5. Choose "DOCUMENTS" from the header toolbar. The patient can delete or remove previously uploaded documents by pressing the DELETE button.

Expected results:

1. Should indicate a method for patients to remove information or uploaded documents.

Part 2: Static application security testing (SAST)

True positives per hour

305 minutes for 6 true positives, that is about 51 minutes per true positive.

Alerts

Alert 1.

Add password protection to this database

Cross Reference

Responsibility issue | Not trustworthy

Add password protection to this database. [🔗](#)

A secure password should be used when connecting to a database [php:S2115](#)

Software qualities impacted: **Security** 🔴

☐ Open ☒ Not assigned ☐ Vulnerability ☒ Blocker

Where is the issue?

Why is this an issue?

How can I fix it?

Activity

More Info

Tags

cwe +

Effort

45min

Introduced

6 years ago

test > contrib/util/deidentification/deidentification.php [🔗](#)

Open in IDE

[See all issues in this file](#) ⚙️

61 daniel_

//*****

62

63

64

65

66

Add password protection to this database.

67

68

69

70

71

72

73

74

75

4

Screenshot

```

//Contact the database
//$con = mysqli_connect("HOST","USER","PASS","DATABASE") or die("Some error
occurred during connection " . mysqli_error($con));

//Instructions: Change these 4 values. The run as a normal php script
//Remember - there is no turning back

//To run script, comment out the return line, fill out the databae credentials,
and run. Uncomment out the return statement when complete.
return 0;
$host = 'localhost';
$user = 'root';
$pass = '';
$database = '';
$DEBUG = false;
//
//
//*****
//*****

$con = mysqli_connect($host, $user, $pass, $database) or die("Some error
occurred during connection. must enter Host, Username, password, and database
in mysqli_connect() " . mysqli_error($con));
echo("\n Successfully connected to database..... Waiting..... \n ");

```

Why is this a true positive

This is a true positive in that this code connects to a database that has no password protection. The script is intended to be modified by the user before running in order to add the database password, but the script, as is, allows the user to connect to a database without a password.

How to fix the vulnerability

The vulnerability can be fixed by using an environment variable to store the password for the database. Then use the environment variable in the script.

CWE Number

The CWE is CWE-521: Weak Password Requirements.

ASVS Control

One ASVS control is 2.1.1.

Alert 2.

Enable server certificate on this SSL/TLS connection.

Cross Reference

Responsibility issue | Not trustworthy

Enable server certificate validation on this SSL/TLS connection. [🔗](#)

Server certificates should be verified during SSL/TLS connections [php:S4830](#)

Software qualities impacted: **Security** 🔴

☐ Open ☒ Not assigned ☐ Vulnerability ☐ Critical

Where is the issue?

Why is this an issue?

How can I fix it?

Activity

More Info

Tags

cwe ... +

Effort

5min

Introduced

8 years ago

test > interface/eRxXMLBuilder.php [🔗](#)

Open in IDE

[See all issues in this file](#) [🔗](#)

↑

99 sam.li_

\$data = array('RxInput' => \$xml);

100

101

curl_setopt(\$curlHandler, CURLOPT_URL, \$this->getGlobals()->getPath());

102

curl_setopt(\$curlHandler, CURLOPT_POST, 1);

103

curl_setopt(\$curlHandler, CURLOPT_POSTFIELDS, 'RxInput=' . \$xml);

104

curl_setopt(\$curlHandler, CURLOPT_SSL_VERIFYPEER, 0);

Enable server certificate validation on this SSL/TLS connection.

105

curl_setopt(\$curlHandler, CURLOPT_FOLLOWLOCATION, 1);

106 robert_

curl_setopt(\$curlHandler, CURLOPT_COOKIESESSION, true);

107 sam.li_

curl_setopt(\$curlHandler, CURLOPT_COOKIEFILE, \$sitePath . '/newcrop-cookiefile');

108

curl_setopt(\$curlHandler, CURLOPT_COOKIEJAR, \$sitePath . '/newcrop-cookiefile');

109

curl_setopt(\$curlHandler, CURLOPT_COOKIE, session_name() . '=' . session_id());

110

curl_setopt(\$curlHandler, CURLOPT_USERAGENT, 'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)');

111

curl_setopt(\$curlHandler, CURLOPT_RETURNTRANSFER, true);

112

113 robert_

\$result = curl_exec(\$curlHandler) or die(curl_error(\$curlHandler));

↓

Screenshot

```

Codeium: Refactor | Explain | Generate Function Comment | ✕
public function checkError($xml)
{
    $curlHandler = curl_init($xml);
    $sitePath = $this->getGlobals()->getOpenEMRSiteDirectory();
    $data = array('RxInput' => $xml);

    curl_setopt($curlHandler, CURLOPT_URL, $this->getGlobals()->getPath());
    curl_setopt($curlHandler, CURLOPT_POST, 1);
    curl_setopt($curlHandler, CURLOPT_POSTFIELDS, 'RxInput=' . $xml);
    curl_setopt($curlHandler, CURLOPT_SSL_VERIFYPEER, 0);
    curl_setopt($curlHandler, CURLOPT_FOLLOWLOCATION, 1);
    curl_setopt($curlHandler, CURLOPT_COOKIESESSION, true);
    curl_setopt($curlHandler, CURLOPT_COOKIEFILE, $sitePath . '/' .
newcrop-cookiefile');
    curl_setopt($curlHandler, CURLOPT_COOKIEJAR, $sitePath . '/' .
newcrop-cookiefile');
    curl_setopt($curlHandler, CURLOPT_COOKIE, session_name() . '=' .
session_id());
    curl_setopt($curlHandler, CURLOPT_USERAGENT, 'Mozilla/4.0 (compatible;
MSIE 6.0; Windows NT 5.1)');
    curl_setopt($curlHandler, CURLOPT_RETURNTRANSFER, true);

    $result = curl_exec($curlHandler) or die(curl_error($curlHandler));

    curl_close($curlHandler);

    return $result;
}

```

Why is this a true positive

This is a true positive because the curl command does not verify the certificate of the URL when executing the POST call. This means that the user is potentially providing sensitive information to a malicious server.

How to fix vulnerability

The vulnerability can be fixed by forcing the curl command to verify the server's certificate. This would also require creating a certificate for the URL's that are used in this script.

CWE Number

The CWE is CWE-295 - Improper Certificate Validation.

ASVS Control

The ASVS control is 1.9.2 .

Alert 3.

Enable server hostname verification on this SSL/TLS connection

Cross Reference

Intentionality issue | Not complete

Enable server hostname verification on this SSL/TLS connection. [↗](#)

Server hostnames should be verified during SSL/TLS connections [php:S5527](#)

Software qualities impacted: **Security**

Open

Not assigned

Vulnerability

Critical

Tags

cwe ... +

Effort

5min

Introduced

1 year ago

Where is the issue?	Why is this an issue?	How can I fix it?	Activity	More Info
215 216 217	<div>Enable server certificate validation on this SSL/TLS connection.</div>	<pre>curl_setopt(\$ch, CURLOPT_URL, \$post_url); curl_setopt(\$ch, CURLOPT_POST, true); curl_setopt(\$ch, CURLOPT_SSL_VERIFYPEER, false);</pre>		
218	<div>Enable server hostname verification on this SSL/TLS connection.</div>	<pre>curl_setopt(\$ch, CURLOPT_SSL_VERIFYHOST, false);</pre>		
219 220 221 222		<pre>curl_setopt(\$ch, CURLOPT_RETURNTRANSFER, true); curl_setopt(\$ch, CURLOPT_HEADER, false); curl_setopt(\$ch, CURLOPT_HTTPHEADER, \$headers); curl_setopt(\$ch, CURLOPT_CONNECTTIMEOUT, 10);</pre>		

Screenshot

```

$post_this = [
    'validationObjective' => 'C-CDA_IG_Plus_Vocab',
    'referenceFileName' => 'noscenariofile',
    'vocabularyConfig' => 'ccdaReferenceValidatorConfig',
    'severityLevel' => 'ERROR',
    'curesUpdate' => true,
    'ccdaFile' => $file
];
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $post_url);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HEADER, false);
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 10);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $post_this);

$response = curl_exec($ch);
$status = curl_getinfo($ch, CURLINFO_RESPONSE_CODE);
if (empty($response) || $status !== '200') {
    $reply['resultsMetaData']['resultMetaData'][0]['count'] = 1;
    $reply['ccdaValidationResults'][] = array(
        'description' => xlt('Validation Request failed') .
            ': Error ' . (curl_error($ch) ?: xlt('Unknown')) . ' ' .
            xlt('Request Status') . ':' . $status
    );
}
curl_close($ch);

```

Why is this a true positive

This is a true positive because the curl command does not verify the hostname when it sends the POST command, so there is no guarantee that the host is the one you want to send the curl command to.

How to fix vulnerability

To fix this vulnerability, the user should verify the hostname by setting the verify host parameter to true. In addition, the host may need to be configured with certificates in order to support this feature.

CWE Number

The CWE is CWE-297: Improper Validation of Certificate with Host Mismatch.

ASVS Control

The ASVS control is 1.9.2.

Alert 4.

Enable server certificate validation on this SSL/TLS connection.

Cross Reference

Responsibility issue | Not trustworthy

Enable server certificate validation on this SSL/TLS connection. [🔗](#)

Server certificates should be verified during SSL/TLS connections [php:S4830](#)

Software qualities impacted: **Security**

☐ Open ☐ Not assigned ☒ Vulnerability ☒ Critical

Tags: cwe ... +

Effort: 5min

Introduced: 13 years ago

Where is the issue? Why is this an issue? How can I fix it? Activity More Info

test > library/maviq_phone_api.php [Open in IDE](#) [See all issues in this file](#)

```
48 robert_ $url .= (false === strpos($path, '?') ? "?" : "&") . $encoded;
49 jason_ }
50 bradyn_
51 // initialize a new curl object
52 $curl = curl_init($url);
53 curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);

54 curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);
```

Enable server certificate validation on this SSL/TLS connection.

Screenshot

```
50
51 // initialize a new curl object
52 $curl = curl_init($url);
53 curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
54 curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);
55 curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
56 switch (strtoupper($method)) {
57     case "GET":
58         curl_setopt($curl, CURLOPT_HTTPGET, true);
59         break;
60     case "POST":
61         curl_setopt($curl, CURLOPT_POST, true);
62         curl_setopt($curl, CURLOPT_POSTFIELDS, $encoded);
63         break;
```

Why is this a true positive

This is a true positive vulnerability. The curl command bypasses server certificate validation by setting the verify peer option to false in the curl command.

How to fix vulnerability

To fix this vulnerability, the verify peer option should be set to true. This would also require configuring the URL to use certificates, if it does not already.

CWE Number

The CWE is CWE-295 - Improper Certificate Validation.

ASVS Control

The ASVS control is 1.9.2 .

Alert 5.

Enable server hostname verification on this SSL/TLS certificate.

Cross Reference

Intentionality issue | Not complete

Enable server hostname verification on this SSL/TLS connection. [🔗](#)

Server hostnames should be verified during SSL/TLS connections [php:S5527](#)

Software qualities impacted: Security

Open

Not assigned

Vulnerability

Critical

Tags

cwe ... +

Effort

5min

Introduced

13 years ago

Where is the issue?	Why is this an issue?	How can I fix it?	Activity	More Info
49 jason... 50 bradyn...	<pre>} // initialize a new curl object \$curl = curl_init(\$url); curl_setopt(\$curl, CURLOPT_SSL_VERIFYPEER, false);</pre>	<div>Enable server certificate validation on this SSL/TLS connection.</div> <pre>curl_setopt(\$curl, CURLOPT_SSL_VERIFYHOST, false);</pre>	<div>Enable server hostname verification on this SSL/TLS connection.</div>	
55 robert... 56 bradyn...	<pre>curl_setopt(\$curl, CURLOPT_RETURNTRANSFER, true); switch (strtoupper(\$method)) {</pre>			

Screenshot

```

50
51 // initialize a new curl object
52 $curl = curl_init($url);
53 curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
54 curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);
55 curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
56 switch (strtoupper($method)) {
57     case "GET":
58         curl_setopt($curl, CURLOPT_HTTPGET, true);
59         break;
60     case "POST":
61         curl_setopt($curl, CURLOPT_POST, true);
62         curl_setopt($curl, CURLOPT_POSTFIELDS, $encoded);
63         break;

```

Why is this a true positive

This is a true positive. The curl command is not configured to verify the hostname when the command is sent.

How to fix vulnerability

This problem can be fixed if curl's verify host option is set to true. In addition to this, the host has to be configured with certificates to make this feature available.

CWE Number

The CWE is CWE-297: Improper Validation of Certificate with Host Mismatch.

ASVS Control

The ASVS control is 1.9.2.

Alert 6.

Make sure that using this pseudorandom number generator is safe here.

Cross Reference

Make sure that using this pseudorandom number generator is safe here. [🔗](#)

Using pseudorandom number generators (PRNGs) is security-sensitive [php:S2245](#)

Review priority:
⬆ Medium

Category:
Weak Cryptography

Assignee:
Not assigned ▾

Status: To review

This security hotspot needs to be reviewed to assess whether the code poses a risk.

Review

Where is the risk?

What's the risk?

Assess the risk

How can I fix it?

Activity

contrib/util/deidentification/deidentification.php [🔗](#)

Open in IDE

```

168     while ($result = mysqli_fetch_array($query)) {
169         if ($DEBUG === true) {
170             if ($i === 10) {
171                 break;
172             }
173         }
174
175         $i++;
176         $string = '';
177         //Give the user a new last name in patient_data.lname
178         $last_name = $lnames[rand(0, 800)];

```

Make sure that using this pseudorandom number generator is safe here.

Screenshot

```

174
175     $i++;
176     $string = '';
177     //Give the user a new last name in patient_data.lname
178     $last_name = $lnames[rand(0, 800)];
179
180     //Give the user a new first name
181     $first_name_male = $male[rand(0, 32)];
182     $first_name_female = $female[rand(0, 74)];
183

```

Why is this a false positive and how we verified

This is a false positive since the use of the pseudo random number generator is not for security sensitive information. Instead the PRNG is used to pick a random last name from a list of last names. I verified this through code inspection and the function header stated “This function replaces the first and last name of the patient with a auto generated name”.

Alert 7.

Detected 'password' in this variable name, review this potentially hardcoded credential.

Cross Reference


Detected 'password' in this variable name, review this potentially hardcoded credential. [🔗](#)

Hard-coded credentials are security-sensitive [php:S2068](#)


Status: To review

This security hotspot needs to be reviewed to assess whether the code poses a risk.

Review

Review priority:  High

Category: Authentication

Assignee: Not assigned 

Where is the risk?

What's the risk?

Assess the risk

How can I fix it?

Activity

interface/.../tests/Tests/Unit/TeleHealthUserRepositoryTest.php [🔗](#)

Open in IDE

↑...

```
13 namespace Comlink\OpenEMR\Modules\TeleHealthModule;
14
15 use Comlink\OpenEMR\Modules\TeleHealthModule\Models\TeleHealthUser;
16 use Comlink\OpenEMR\Modules\TeleHealthModule\Repository\TeleHealthUserRepository;
17 use OpenEMR\Common\Database\QueryUtils;
18 use PHPUnit\Framework\TestCase;
19
20 class TeleHealthUserRepositoryTest extends TestCase
21 {
22     const TEST_USERNAME = "phpunit-test-username";
23     const TEST_PASSWORD = "randomToken";
```

Detected 'password' in this variable name, review this potentially hardcoded credential.

Screenshot

```

19
20 Codeium: Explain
   class TeleHealthUserRepositoryTest extends TestCase
21 {
22     const TEST_USERNAME = "phpunit-test-username";
23     const TEST_PASSWORD = "randomToken";
24
25     Codeium: Refactor | Explain | Generate Function Comment | X
   protected function tearDown(): void
26     {
27         parent::tearDown(); // TODO: Change the autogenerated stub
28         QueryUtils::sqlStatementThrowException("DELETE FROM " .
           TeleHealthUserRepository::TABLE_NAME
29           . " WHERE username LIKE ?", ["%" . self::TEST_USERNAME . "%"]);
30     }
31

```

Why is this a false positive and how we verified it

This is a false positive. This password value is only used for testing purposes and does not provide an attacker with information to attack the system. We verified this through code inspection. This file only contains unit tests and therefore does not impact how the code is run operationally.

Alert 8.

Using http protocol is insecure. Use https instead.

Cross Reference

Using http protocol is insecure. Use https instead [🔗](#)

Using clear-text protocols is security-sensitive [php:S5332](#)

Review priority:

👉 Low

Category:

Encryption of Sensitive Data

Assignee:

Not assigned



Status: To review

This security hotspot needs to be reviewed to assess whether the code poses a risk.

Review

Where is the risk?

What's the risk?

Assess the risk

How can I fix it?

Activity

src/RestControllers/FHIR/FhirMetaDataRestController.php [🔗](#)

Open in IDE



125

}

126

127

/**

128

* Adds all of the FHIR REST Extensions needed for things such as SMART on FHIR

129

* @param FHIRCapabilityStatementSecurity \$statement

130

*/

131

private function addOAuthSecurityExtensions(FHIRCapabilityStatementSecurity \$statement): void

132

{

133

\$authServer = new AuthorizationController();

134

\$oauthExtension = new FHIRExtension();

135

\$oauthExtension->setUrl(new FHIRUrl(

"http://fhir-registry.smarthealthit.org/StructureDefinition/oauth-uris");

Using http protocol is insecure. Use https instead

Screenshot

```

127  /**
128  * Adds all of the FHIR REST Extensions needed for things such as SMART on
    FHIR
129  * @param FHIRCapabilityStatementSecurity $statement
130  */
Codeium: Refactor | Explain | X
131  private function addOAuthSecurityExtensions(FHIRCapabilityStatementSecurity
    $statement): void
132  {
133      $authServer = new AuthorizationController();
134      $oauthExtension = new FHIRExtension();
135      $oauthExtension->setUrl(new FHIRUrl("http://fhir-registry.smarthealthit.
    org/StructureDefinition/oauth-uris"));
136      $oauthUrIs = [
137          // @see http://www.hl7.org/fhir/smart-app-launch/
    StructureDefinition-oauth-uris.html
138          // and @see http://www.hl7.org/fhir/smart-app-launch/conformance/
    index.html#declaring-support-for-oauth2-endpoints
139          // token and authorize are required because we don't use implicit
    grant flow.
140          'token' => $authServer->getTokenUrl()
141          , 'authorize' => $authServer->getAuthorizeUrl()
142          , 'register' => $authServer->getRegistrationUrl()
143          , 'introspect' => $authServer->getIntrospectionUrl()
144          // TODO: if we have these URIs we can provide them
145          // , 'manage' => $authServer->getManageUrl()
146          // , 'revoke' => ''
147      ];

```

Why is this a true positive

This is a true positive. The application is using http and any data sent using this protocol will result in sensitive data being sent in cleartext..

How to fix vulnerability

To fix this vulnerability, the application should use https when transmitting data. In this case change http:// to https://

CWE Number

The CWE is CWE-319 Cleartext Transmission of Sensitive Information.

ASVS Control

The ASVS control is 9.2.2.

Alert 9.

Make sure automatically installing recommended packages is safe here.

Cross Reference

Make sure automatically installing recommended packages is safe here. [🔗](#)

Automatically installing recommended packages is security-sensitive [docker:S6500](#)

Status: To review

This security hotspot needs to be reviewed to assess whether the code poses a risk.

Review

Where is the risk?

What's the risk?

Assess the risk

How can I fix it?

Activity

docker/library/docker/dev-php-fpm-5-6-redis/Dockerfile [🔗](#)

Open in IDE

11

#

12

FROM php:5.6-fpm

13

14

Update

15

RUN apt-get update

16

17

Add mysql-client package that is needed in the OpenEMR Backup gui, which does direct command mysql commands

18

Add imagemagick that is needed for some image processing in OpenEMR

19

Note this basically add 160MB of space to the docker, so would be nice for OpenEMR to not require this stuff

20

and instead rely on php scripts, if possible.

21

RUN apt-get install -y mysql-client \

Make sure automatically installing recommended packages is safe here.

Review priority:
👍 Low

Category:
Others

Assignee:
Not assigned [▼](#)

Screenshot


```

#
FROM php:5.6-fpm

# Update
RUN apt-get update

# Add mysql-client package that is needed in the OpenEMR Backup gui, which does
direct command mysql commands
# Add imagemagick that is needed for some image processing in OpenEMR
# Note this basically add 160MB of space to the docker, so would be nice for
OpenEMR to not require this stuff
# and instead rely on php scripts, if possible.
RUN apt-get install -y mysql-client \
    imagemagick

# Add the php extensions (note using a very cool script by mlocati to do this)
ADD https://raw.githubusercontent.com/mlocati/docker-php-extension-installer/
master/install-php-extensions /usr/local/bin/
RUN chmod uga+x /usr/local/bin/install-php-extensions && sync && \
    install-php-extensions pdo_mysql \
    ldap \
    xsl \
    gd \
    zip \
    soap \
    gettext \
    mysqli \
    sockets \
    tokenizer \
    xmlreader \
    redis

```

Why is this a False Positive and how we verified it


This is a false positive. The hotspot is to remind the user that unused packages should be removed, but there are no issues with installing packages to run the software. We verified this through code inspection. The apt-get update ensures that the latest packages are installed which should force the docker container to have the latest patched packages.

Alert 10.

Make sure this weak hash algorithm is not used in a sensitive context here.

Cross Reference

Make sure this weak hash algorithm is not used in a sensitive context here.



Using weak hashing algorithms is security-sensitive [php:S4790](#)

Status: To review

This security hotspot needs to be reviewed to assess whether the code poses a risk.

Review


Review priority: Low

Category: Others

Assignee: Not assigned

WhereWhatAssessHowActivity

Review

portal/.../fwk/libs/verysimple/Phreeze/DataSet.php 

Open in IDE

...

510

}

511

512

/**

513

*

514

* @param

515

* \$cachekey

516

*/

517

private function LockCache(\$cachekey)

518

{

519

if (\$this->_phreezer->LockFilePath) {

520

touch(\$this->_phreezer->LockFilePath . md5(\$cachekey) . ".lock");

Make sure this weak hash algorithm is not used in a sensitive context here.

Screenshot

```

512  /**
513  *
514  * @param
515  *      $cachekey
516  */
Codeium: Refactor | Explain | ✕
517 private function LockCache($cachekey)
518 {
519     if ($this->_phreezer->LockFilePath) {
520         touch($this->_phreezer->LockFilePath . md5($cachekey) . ".lock");
521     }
522 }
523

```

Why is this a False Positive and how we verified it

This is a false positive. The md5sum is used to create a unique value based on the cache key. The value maps the cachekey to the lock. Since the md5sum is not used for any sensitive data, there is no concern. We verified this through code inspection. The function is used to lock a cache which is mainly used for multithreading purposes and not to hide sensitive information.