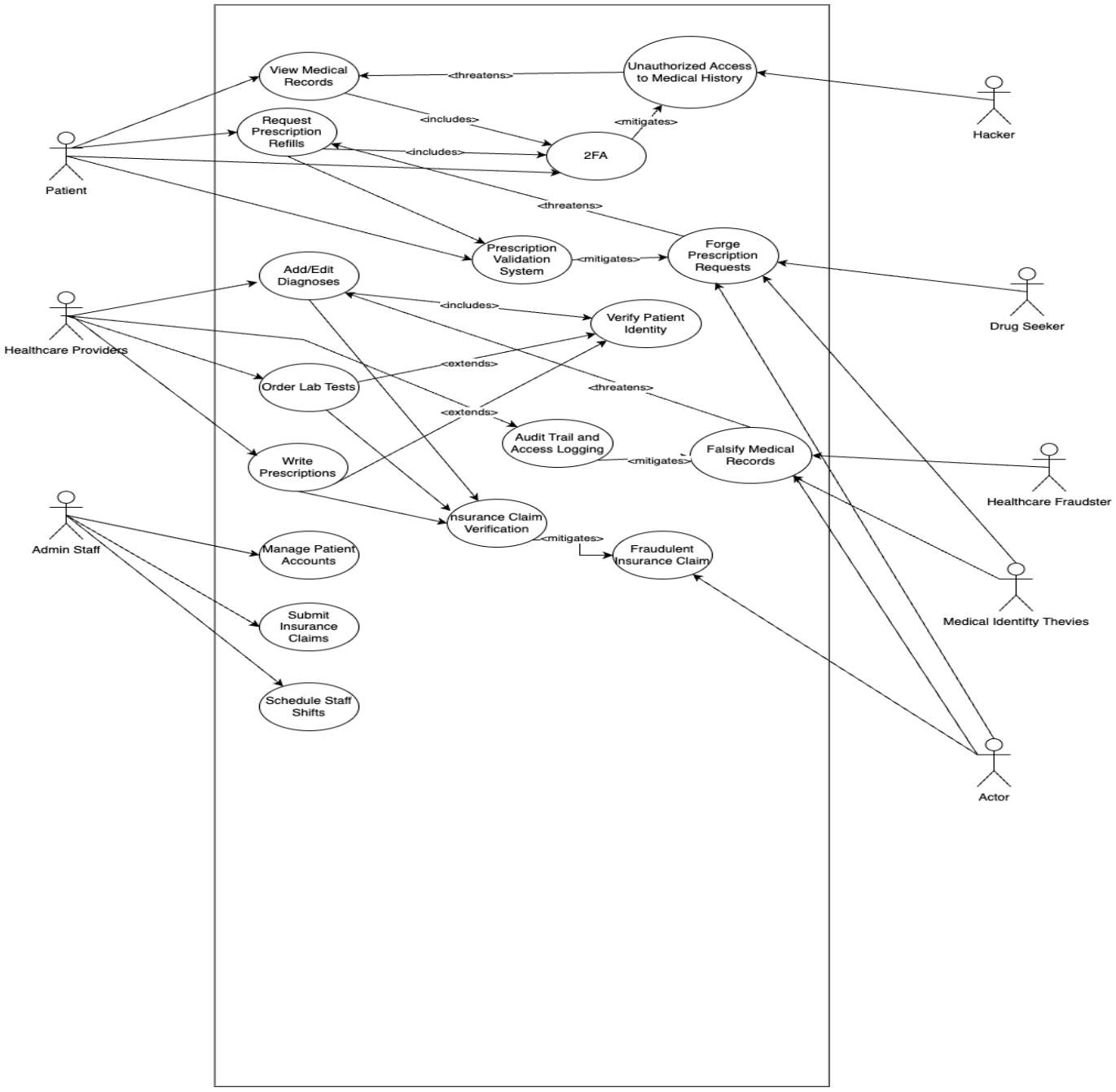
Project 4 :

Part 1 :

Abuse and Misuse Case Diagram:



Part 2:

Team member 1

Testcase fixed from Project 1:

ASVS V2.1 Password Security
Unique ID: 2.1.1-1
CWE: 521
Description: Verify that user set passwords are at least 12 characters in length (after multiple spaces are combined).

Repeatable step:

1. Launch OpenEMR using URL: http://localhost:80/
2. Login using credentials
3. Click on the user profile icon on the top right corner and then in the drop down menu, select change password to redirect to that page.
4. Enter current password.
5. Enter a new password with length less than 12 characters.
6. Re-enter new password.
7. Click on save changes.

Expected results:

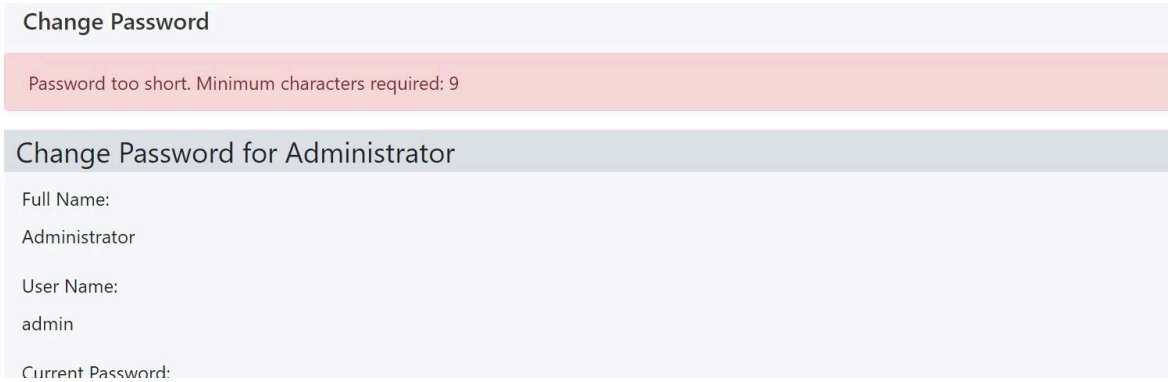
1. Unable to change password.
2. The error should be shown as “Password too short. Minimum characters required: 12”

Fix:

```
'gbl_minimum_password_length' => array(
    xl('Minimum Password Length'),
    array(
        '0' => xl('No Minimum'),
        '4' => '4',
        '5' => '5',
        '6' => '6',
        '7' => '7',
        '8' => '8',
        '9' => '9',
        '10' => '10',
        '11' => '11',
        '12' => '12',
        '13' => '13',
        '14' => '14',
        '15' => '15',
        '16' => '16',
        '17' => '17',
        '18' => '18',
        '19' => '19',
        '20' => '20',
    ),
    '12',                               // default
    xl('Minimum length of password.')
),
```

Explanation:
The above test case fails as the by default minimum length in password setting is 9 for any new user. There are 2 easy we can define password, reset user password, or admin can create new users, where they can make a password. For both of these, minimum password length is 9. As per ASVS document, minimum user-set password length should be no less than 12 characters. To resolve this issue, I cloned the penEMR version 7.0.2 code in my machine, ran it using docker configurations in docker/development_easy and then used frontend to trace back where the minimum password length check is defined. It lead me to \src\Common\Auth\AuthUtils.php file. This file is used for setting passwords and updating them. It has a function for updating passwords, so I inspected it and found out that a superglobal associative array is used to define password length and is set to default value of 9. I changed it to 12.

Before Fix:



After Fix:

Team member 2

Testcase fixed :

ASVS V5.1 : Input Validation
Identifier: 5.1.4-1
CWE Code: 20
Description : Verify that structured data is strongly typed and validated against a defined schema including allowed characters, length and pattern (e.g. credit card numbers, e-mail addresses, telephone numbers, or validating that two related fields are reasonable, such as checking that suburb and zip/postcode match).

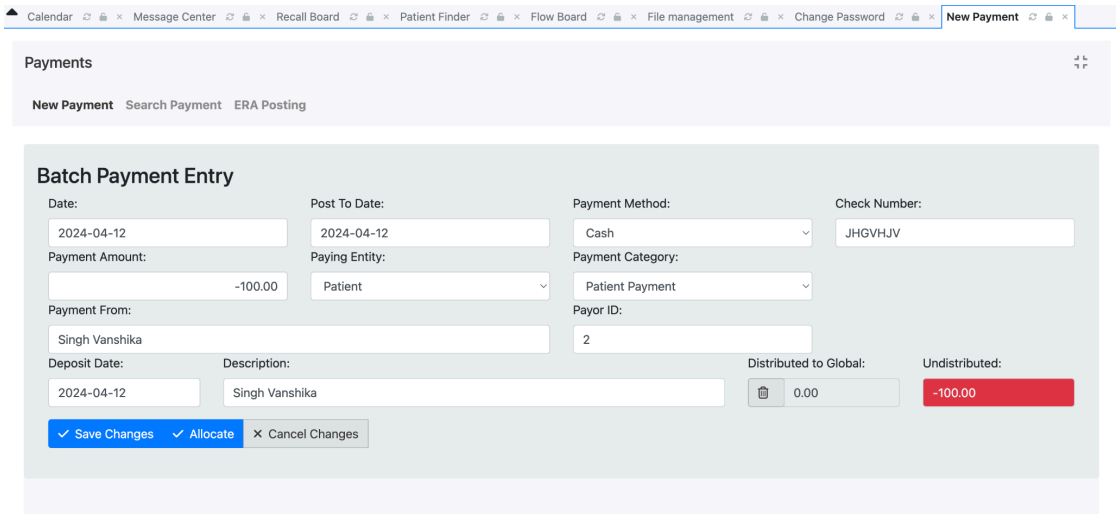
Steps to Replicate:

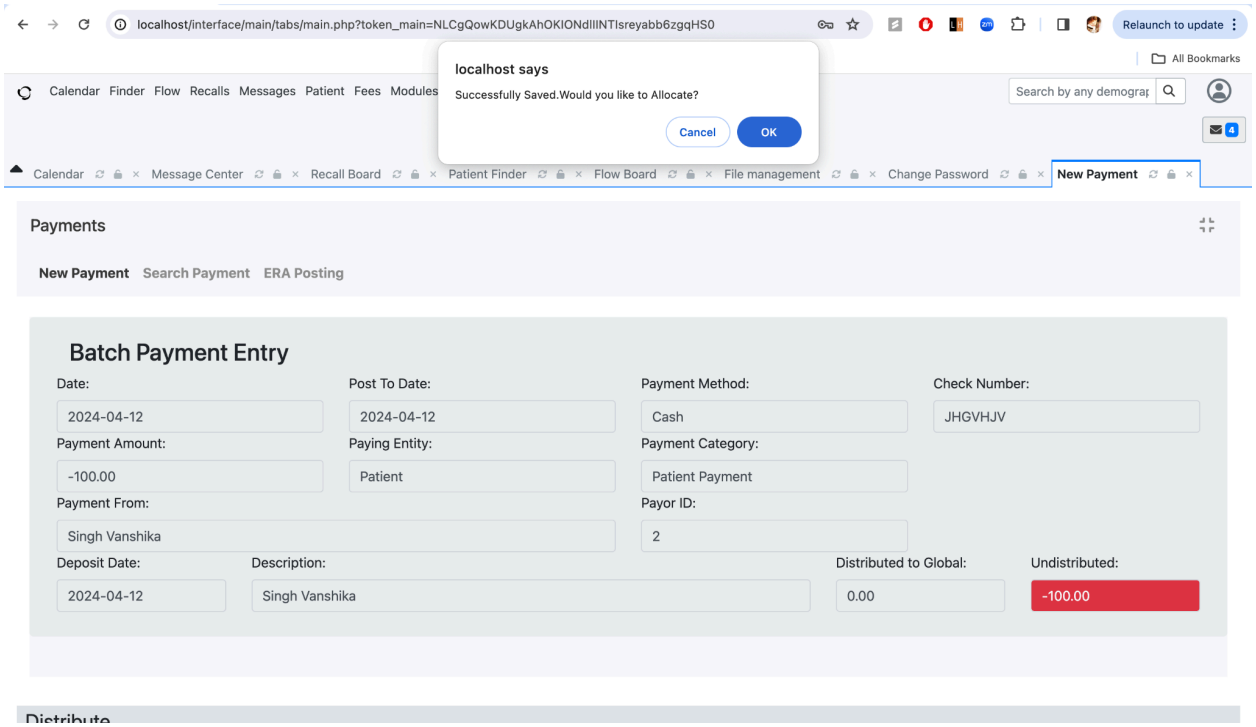
- Initiate OpenEMR at: http://localhost:80
- Sign in with given credentials.
- Go through the following menu path: Fee -> Batch Payments -> Batch Payment Entry
- Select Date as today's date, Payment Method as "Cash", Check Number as "12345" , Payment amount as "-100" **i.e Negative 100,** Payment From - Client and Deposit Date as today's date and click on Save Changes.

Expected Outcomes:

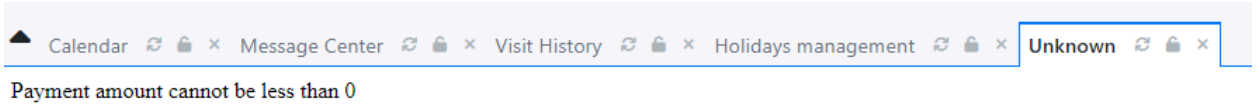
- After clicking on Save Changes, an Error should be displayed if a negative value is put in the payment Amount text Field.
- The error should be compliant with ASVS standards.

Screenshot Before Fix :




localhost/interface/main/tabs/main.php?token_main=NLCgQowKDUgkAhOKIONdlIINTIsreyabb6zgqHS0
Relaunch to update
All Bookmarks
localhost says
Successfully Saved.Would you like to Allocate?
Cancel
OK
Calendar Finder Flow Recalls Messages Patient Fees Modules
Search by any demograp
Calendar
Message Center
Recall Board
Patient Finder
Flow Board
File management
Change Password
New Payment
Payments
New Payment
Search Payment
ERA Posting
Batch Payment Entry
Date:
2024-04-12
Post To Date:
2024-04-12
Payment Method:
Cash
Check Number:
JHGVHJV
Payment Amount:
-100.00
Paying Entity:
Patient
Payment Category:
Patient Payment
Payment From:
Singh Vanshika
Payor ID:
2
Deposit Date:
2024-04-12
Description:
Singh Vanshika
Distributed to Global:
0.00
Undistributed:
-100.00
Distribute

After Fix:


Calendar
Message Center
Visit History
Holidays management
Unknown
Payment amount cannot be less than 0

Fixes in source code:

new_payment.php

```
if(((float)trim(formData('payment_amount')))<0){
        die("Payment amount cannot be less than 0");
    }
```

Explanation :

The vulnerability that was fixed is associated with the ASVS V5.1 input validation, identified as 5.1.4-1 and corresponding to CWE Code: 20, which pertains to improper input validation. This particular security weakness was manifested in the OpenEMR application's handling of payment amounts. Specifically, the application failed to properly validate that the payment amount field only contained positive numerical values. As a result, it erroneously accepted negative values, which could potentially disrupt financial transaction integrity and lead to accounting inconsistencies or exploitation for fraudulent purposes.

The demonstrated fix, as visible in the provided screenshot, has successfully mitigated this vulnerability by implementing strict validation rules that disallow the entry of negative numbers. Now, when an attempt is made to enter a negative payment amount, the application responds with a clear and compliant error message: "Payment amount cannot be less than 0." This not only prevents the entry of invalid data but also upholds the standards set forth by the ASVS for secure input handling, ensuring that all structured data, like payment amounts, adhere to the defined schema in terms of allowed characters, length, and pattern.

Team Member 3

ASVS Section 7.4.1 - Error Handling
Unique Identifier: 7.4.1-1
CWE: 210
Description : Verify that a generic message is shown when an unexpected or security sensitive error occurs, potentially with a unique ID which support personnel can use to investigate..

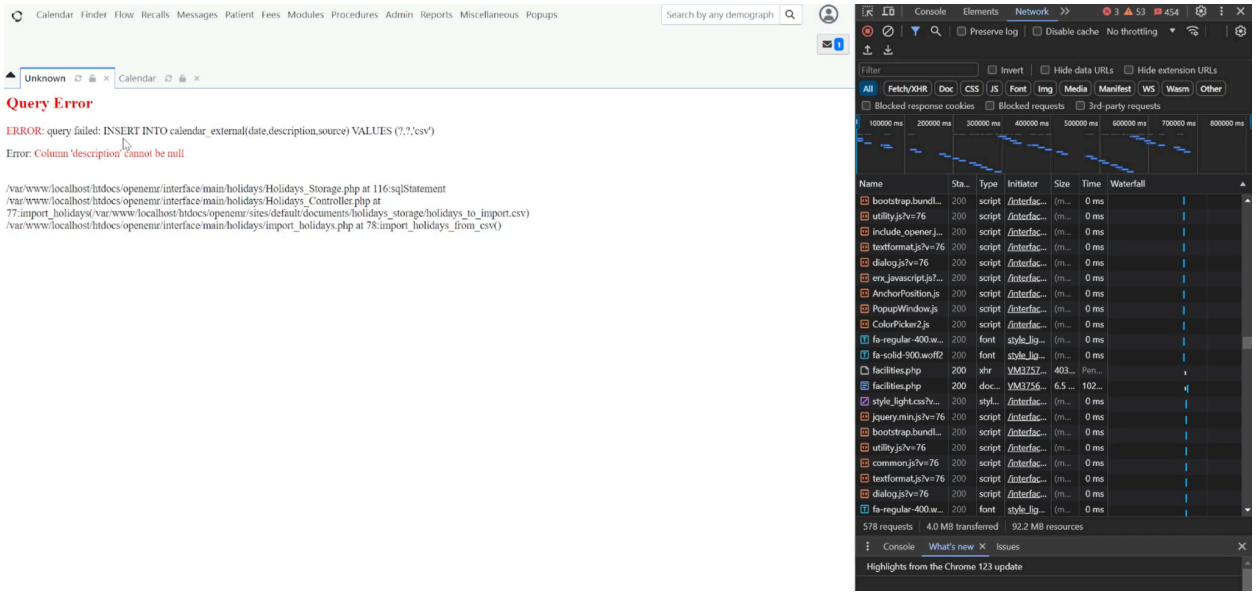
Steps to Replicate:

- Initiate OpenEMR via the URL: http://localhost:80/
- Authenticate using specified user credentials.
- Navigate through admin -> clinic -> Import Holidays via the menu.
- Attempt to upload a PDF document where a CSV file is expected and select Upload/Save.
- After attempting upload, proceed to click on Import holiday events.

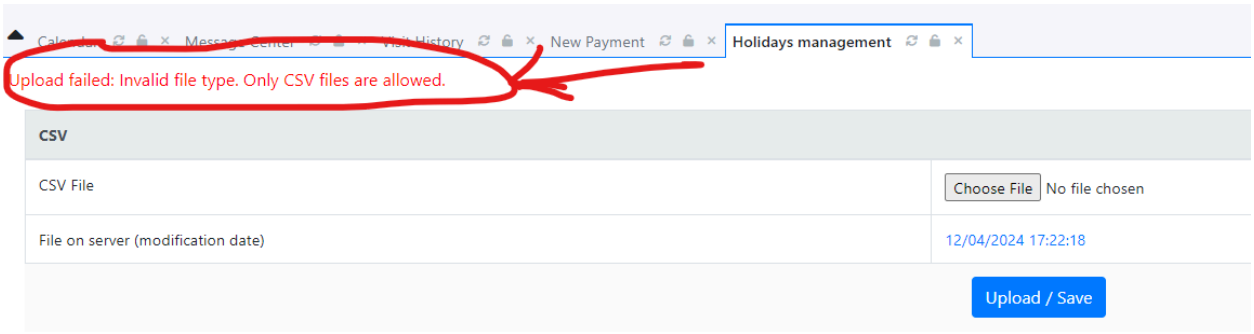
Expected Outcomes:

The system should block the upload of files that are not in CSV format.
Should a non-CSV file be uploaded by any chance, the error handling should be robust, obscuring any sensitive details.

Before Fix



After Fix



Fixes in sourcecode

import_holidays.php

```
if (!empty($_POST['bn_upload'])) {
    if (!CsrfUtils::verifyCsrfToken($_POST["csrf_token_form"])) {
        CsrfUtils::csrfNotVerified();
    }

    $uploadResponse = $holidays_controller->upload_csv($_FILES);
    if ($uploadResponse === "success") {
        $csv_file_data = $holidays_controller->get_file_csv_data();
        $uploadSuccess = true;
    } else {
        $uploadSuccess = false;
        $uploadError = $uploadResponse;
    }
}
-----------------------------------------------------------------
<?php
if (!empty($uploadSuccess)) {
    echo "<p style='color:green'>" . xlt('Successfully Completed') .
"</p>\n";
} elseif (isset($uploadSuccess) && !$uploadSuccess) {
    echo "<p style='color:red'>" . xlt('Upload failed: ') .
xlt($uploadError) . "</p>\n";
}
?>
```

Holidays_Controller.php

```php
public function upload_csv($files)
{
    if (!file_exists($GLOBALS['OE_SITE_DIR'] . "/" . self::UPLOAD_DIR)) {
        if (!mkdir($GLOBALS['OE_SITE_DIR'] . "/" . self::UPLOAD_DIR . "/",
0700, true)) {
            return "Failed to create directory.";
        }
    }

    $uploaded_file_type = pathinfo($files["form_file"]["name"],
PATHINFO_EXTENSION);
    if (strtolower($uploaded_file_type) != "csv") {
        return "Invalid file type. Only CSV files are allowed.";
    }

    if (move_uploaded_file($files["form_file"]["tmp_name"],
$this->target_file)) {
        return "success";
    }

    return "Failed to upload the file.";
}
```

Explanation :

The correction made in OpenEMR's import functionality remedied a critical vulnerability, ensuring compliance with ASVS Section 7.4.1 guidelines related to CWE-210, which demand secure error handling. Previously, uploading an incorrect file type like a PDF instead of a CSV led to revealing SQL errors, risking sensitive data exposure. The patch in `import_holidays.php` ensures only CSV files are accepted, and upon violation, a non-revealing error message is displayed, greatly enhancing the system's resilience against such information leakage and aligning with best practices for error messaging.

Team Member 4

ASVS Section V3.4 Cookie-based Session Management
Unique Identifier: 3.4.1-1
CWE: 1004
Description : Verify that cookie-based session tokens have the 'HttpOnly' attribute set.

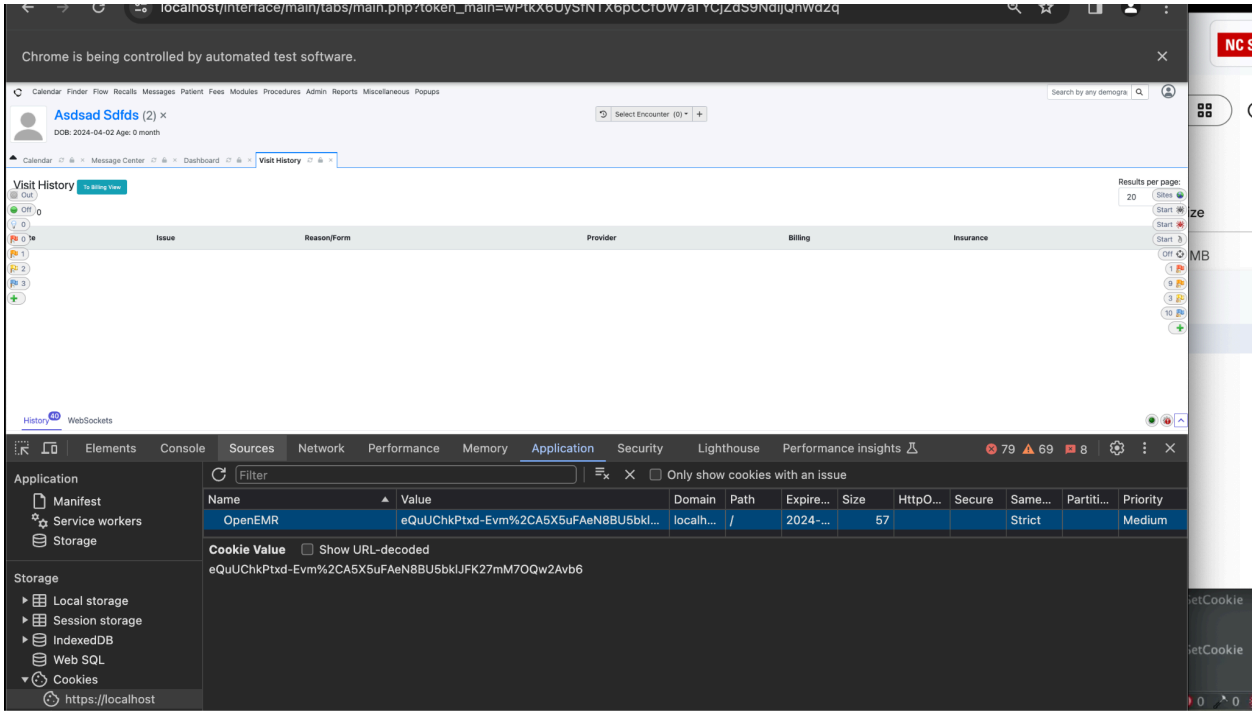
Steps to Replicate:

- Launch OpenEMR application using the URL: http://localhost:80 and open Developer
- tools.
- Login using admin credentials(username: admin,password:pass)
- Now open the POST request and go to the cookies tab..

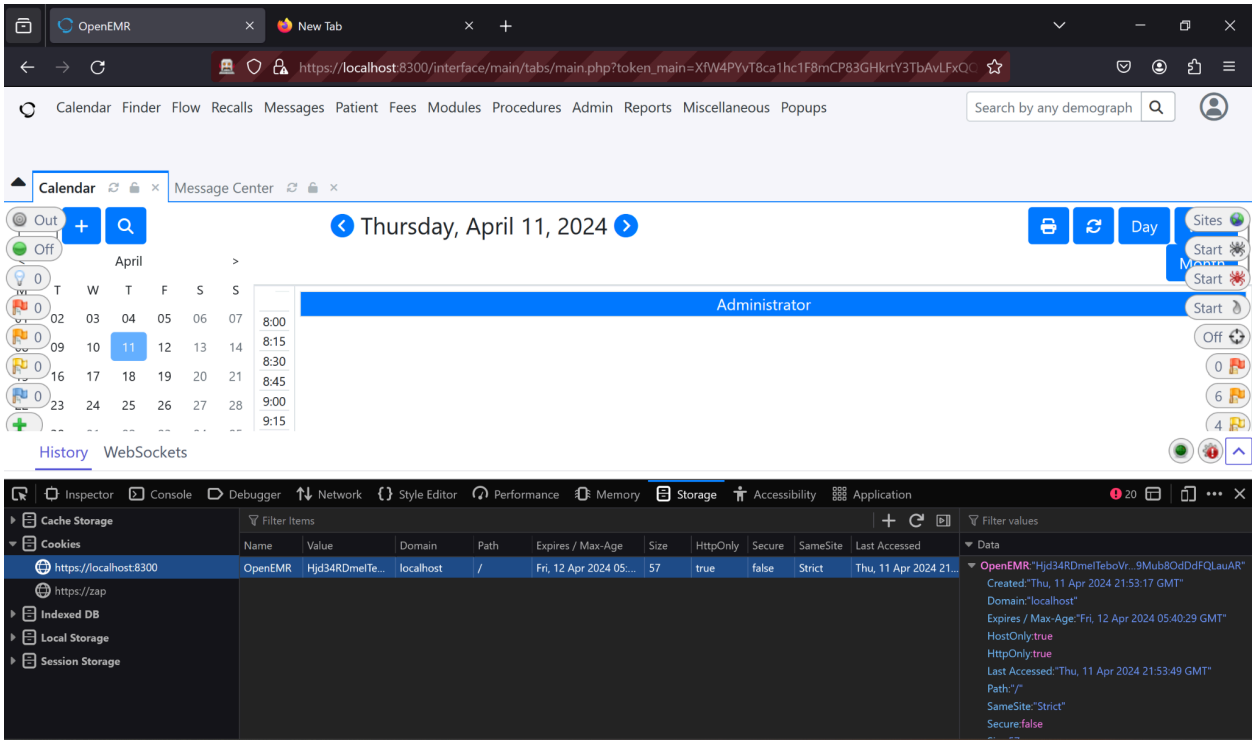
Expected Outcomes:

In the Cookies tab, a ‘HttpOnly’ attribute has to be set.

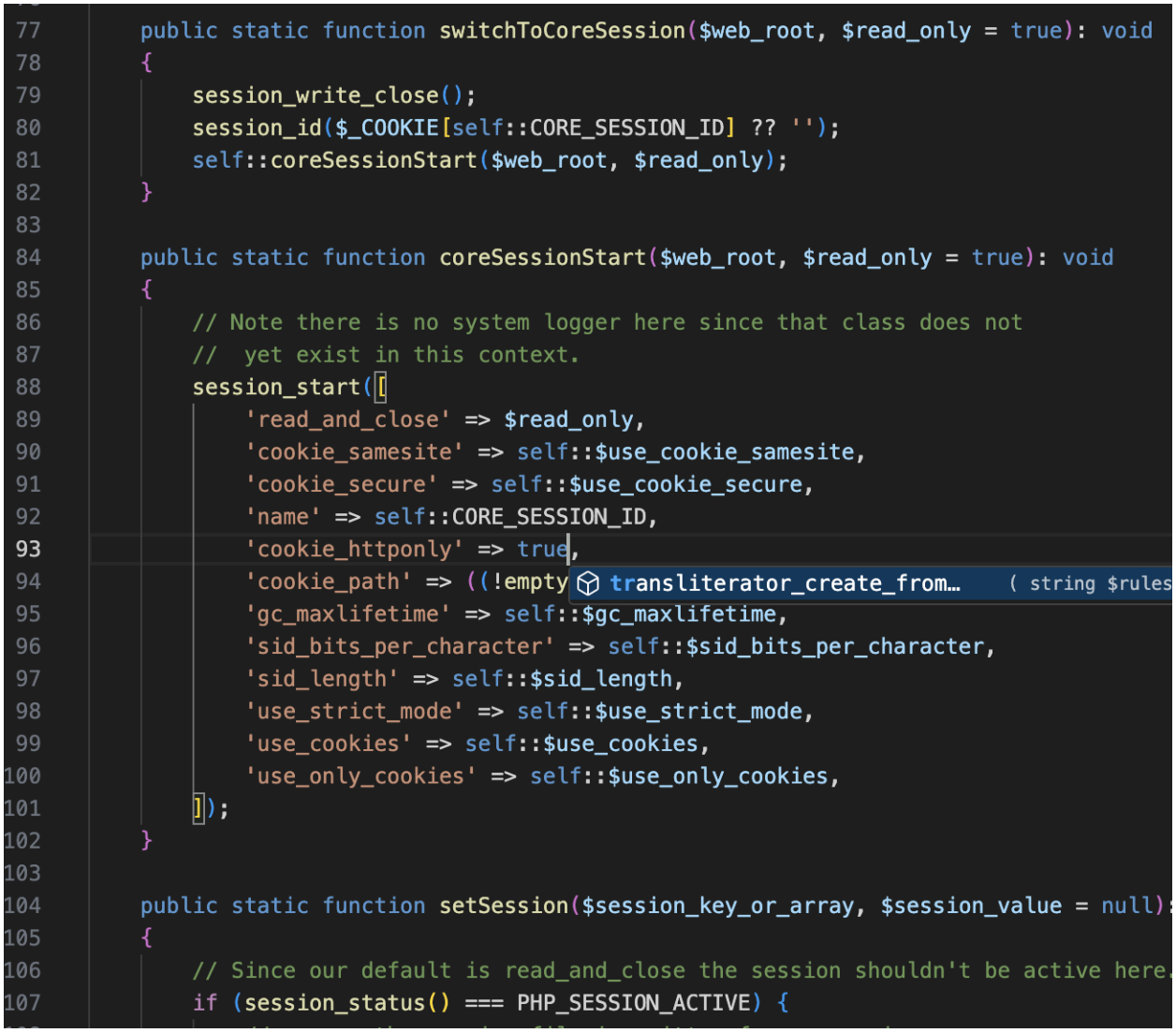
Before Fix:



After fix:


OpenEMR
New Tab
https://localhost:8300/interface/main/tabs/main.php?token_main=XfW4PYvT8ca1hc1F8mCP83GHkrtY3TbAvLFxQC
Calendar Finder Flow Recalls Messages Patient Fees Modules Procedures Admin Reports Miscellaneous Popups
Search by any demograph
Calendar
Message Center
Out
Off
Thursday, April 11, 2024
Day
Sites
Start
Start
Start
Off
April
T W T F S S
02 03 04 05 06 07
09 10 11 12 13 14
16 17 18 19 20 21
23 24 25 26 27 28
8:00
8:15
8:30
8:45
9:00
9:15
Administrator
History
WebSockets
Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessibility
Application
Cache Storage
Cookies
https://localhost:8300
https://zap
Indexed DB
Local Storage
Session Storage
Filter Items
Name Value Domain Path Expires / Max-Age Size HttpOnly Secure SameSite Last Accessed
OpenEMR Hjd34RDmelTe... localhost / Fri, 12 Apr 2024 05:... 57 true false Strict Thu, 11 Apr 2024 21...
Filter values
Data
OpenEMR:"Hjd34RDmelTeboVr...9Mub8OdDdFQLauAR"
Created:"Thu, 11 Apr 2024 21:53:17 GMT"
Domain:"localhost"
Expires / Max-Age:"Fri, 12 Apr 2024 05:40:29 GMT"
HostOnly:true
HttpOnly:true
Last Accessed:"Thu, 11 Apr 2024 21:53:49 GMT"
Path:"/"
SameSite:"Strict"
Secure:false

Code:



```
77      public static function switchToCoreSession($web_root, $read_only = true): void
78      {
79          session_write_close();
80          session_id($_COOKIE[self::CORE_SESSION_ID] ?? '');
81          self::coreSessionStart($web_root, $read_only);
82      }
83
84      public static function coreSessionStart($web_root, $read_only = true): void
85      {
86          // Note there is no system logger here since that class does not
87          //  yet exist in this context.
88          session_start([
89              'read_and_close' => $read_only,
90              'cookie_samesite' => self::$use_cookie_samesite,
91              'cookie_secure' => self::$use_cookie_secure,
92              'name' => self::CORE_SESSION_ID,
93              'cookie_httponly' => true,
94              'cookie_path' => ((!empty transliterator_create_from…    ( string $rules
95              'gc_maxlifetime' => self::$gc_maxlifetime,
96              'sid_bits_per_character' => self::$sid_bits_per_character,
97              'sid_length' => self::$sid_length,
98              'use_strict_mode' => self::$use_strict_mode,
99              'use_cookies' => self::$use_cookies,
100             'use_only_cookies' => self::$use_only_cookies,
101         ]);
102     }
103
104     public static function setSession($session_key_or_array, $session_value = null)
105     {
106         // Since our default is read_and_close the session shouldn't be active here.
107         if (session_status() === PHP_SESSION_ACTIVE) {
```

Explanation:
The httponly cookie value was set to false at start of the session storage. The same has been modified to be true. As such when a new session is created after login, the HTTPOnly parameter will be set to true. This ensures compliance with ASVS requirement 3.4 which requires secure cookie based session management.

Part-3

Team member 1

Vulnerability #	Elapsed Time	Ref info for video traceability	CWE	Commentary
1	2 hours, 37 minutes and 28 seconds	02:27:28	CWE-116	XSS attack invalidates effect of a button i.e. makes button unusable
2	2 hours, 47 minutes and 48 seconds	02:47:48	CWE-209	Generation of Error Message Containing Sensitive Information about database

Black box testcases:

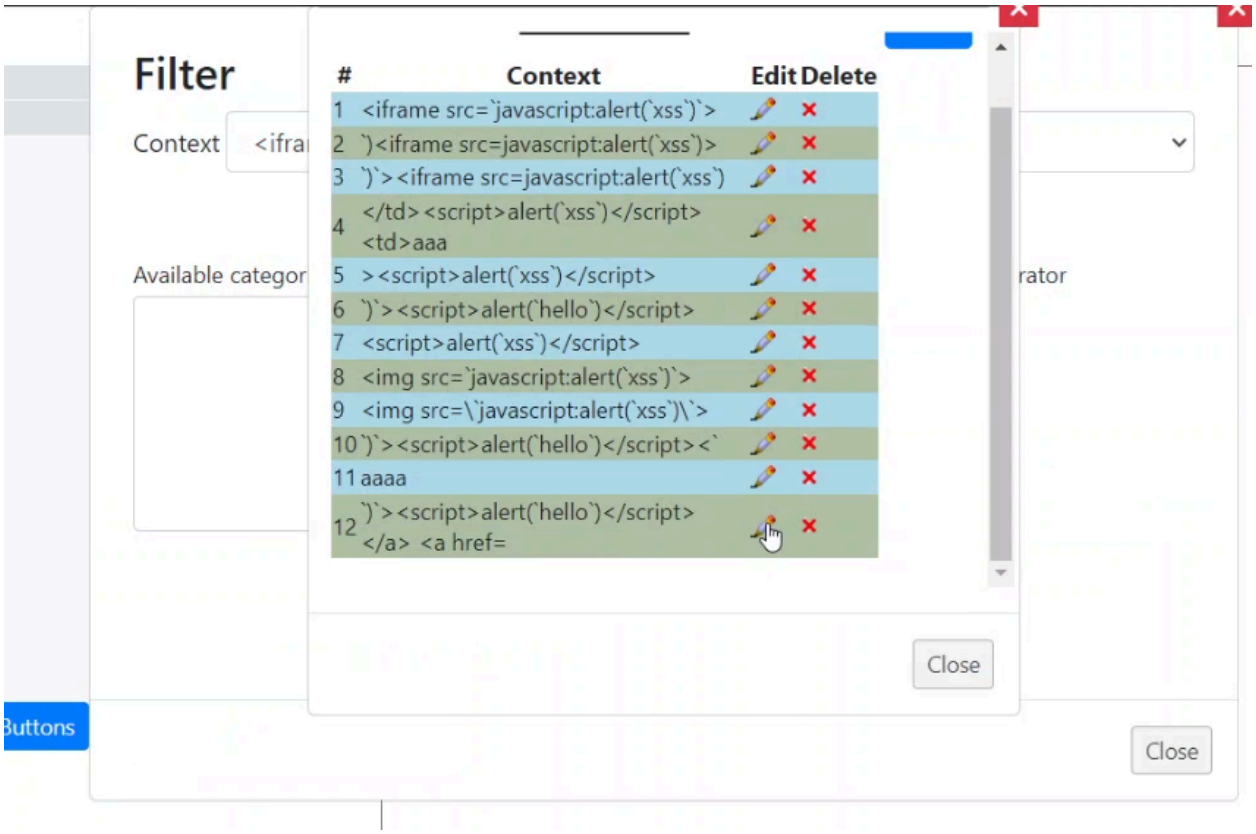
Test 1:
ASVS V5.2.1 Sanitization
Unique ID: 5.2.1-1
CWE: 116
Description: Verify that all untrusted HTML input from WYSIWYG editors or similar is properly sanitized with an HTML sanitizer library or framework feature.

Repeatable step:

1. Launch OpenEMR using URL: http://localhost:80/
2. Login using credentials
3. Go to Patient->new/search patient and search for a patient.
4. Open a patient dashboard using search patient
5. Click on + sign on the top-middle section of page, just below the menubar. It will open encounter form
6. Fill all fields, and then write something in the Reason for visit field.
7. Double click on Reason for visit, it will open a pop up for add missing template.
8. Click on personalize button
9. Click on add context
10. Rename an existing context using edit button as follows:
    a. <img src=”javvascript:alert(‘XSS’)”>
11. Now click on edit button again

Expected results:

1. The edit button should work and let the user edit the context name.



Test 2:
ASVS V7.4.1 Error Handling
Unique ID: 7.4.1-1
CWE: 210
Description: Verify that a generic message is shown when an unexpected or security sensitive error occurs, potentially with a unique ID which support personnel can use to investigate.

Repeatable step:

1. Launch OpenEMR using URL: http://localhost:80/
2. Login using credentials
3. Go to admin -> clinic -> Import Holidays using the menubar
4. Upload a pdf file in CSV file upload and click on Upload/Save
5. Once uploaded, click on Import holiday events

Expected results:

1. The file upload should prevent you from uploading non-CSV file.
2. When clicked on import holiday event, if a non-csv file was somehow uploaded, it should handle error gracefully without revealing sensitive data

DOB: 2024-03-01 Age: 1 month

Open Encounter: 2024-04-12 (3)

Calendar × Message Center × Dashboard × 2024-04-12 Encounter × Unknown ×

Query Error

ERROR: query failed: INSERT INTO calendar_external(date,description,source) VALUES (?,?,'csv')

Error: Column 'description' cannot be null

/var/www/localhost/htdocs/openemr/interface/main/holidays/Holidays_Storage.php at 116:sqlStatement
/var/www/localhost/htdocs/openemr/interface/main/holidays/Holidays_Controller.php at 77:import_holidays(/var/www/localhost/htdocs/openemr/sites/default/documents/holidays_storage/holidays_to_import.csv)
/var/www/localhost/htdocs/openemr/interface/main/holidays/import_holidays.php at 78:import_holidays_from_csv()

Team Member 2

Vulnerability #	Elapsed Time	Ref info for video traceability	CWE	Commentary
1	98 mins i.e I hour 38 mins	Screen Recording 2024-04-11 at 9.15.26 PM - 26mins-44secs	CWE-209	Generation of Error Message Containing Sensitive Information about database
2	102 mins i.e 1 hour 42 mins	Screen Recording 2024-04-11 at 9.15.26 PM - 30mins-10secs	CWE - 73	External Control of File Name or Path
3	110 mins i.e 1 hour 52 mins	Screen Recording 2024-04-11 at 9.15.26 PM - 38mins-47secs	CWE-200	Exposure of Sensitive Information to an Unauthorized Actor

Testcase 1 : (Screenshot attached)

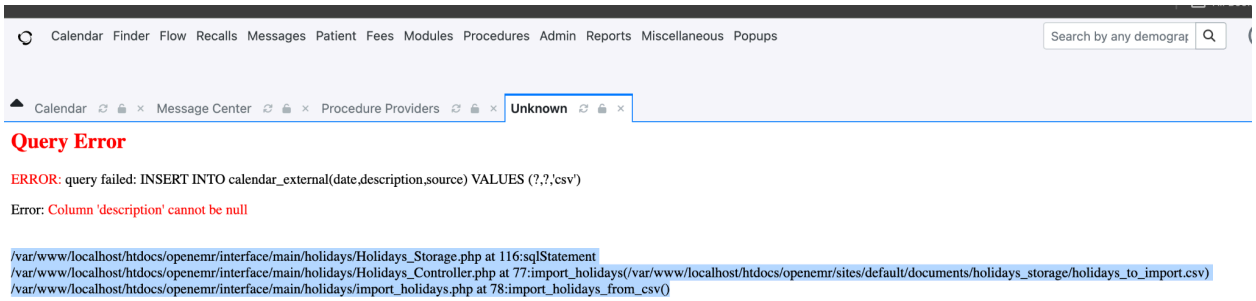
ASVS Section 7.4.1: Error Handling
Identifier: 7.4.1-1
CWE Code: 209
Description : Verify that a generic message is shown when an unexpected or security sensitive error occurs, potentially with a unique ID which support personnel can use to investigate..

Steps to Replicate:

- Initiate OpenEMR at: http://localhost:80
- Sign in with given credentials.
- Go through the following menu path: Admin -> Clinic -> Import Holidays
- Attempt to upload a PDF document in a field meant for CSV files, then press the Upload/Save button.
- After uploading, click on the option to Import holiday events.

Expected Outcomes:

The system should block the upload of any files not in CSV format.
If a non-CSV file is uploaded by mistake, the error should be managed in a controlled manner, ensuring no sensitive information is exposed.



Testcase 2 : (Screenshot attached)

ASVS Section 5.2.6: Sanitization and Sandboxing
Identifier: 5.2.6-1
CWE Code: 73

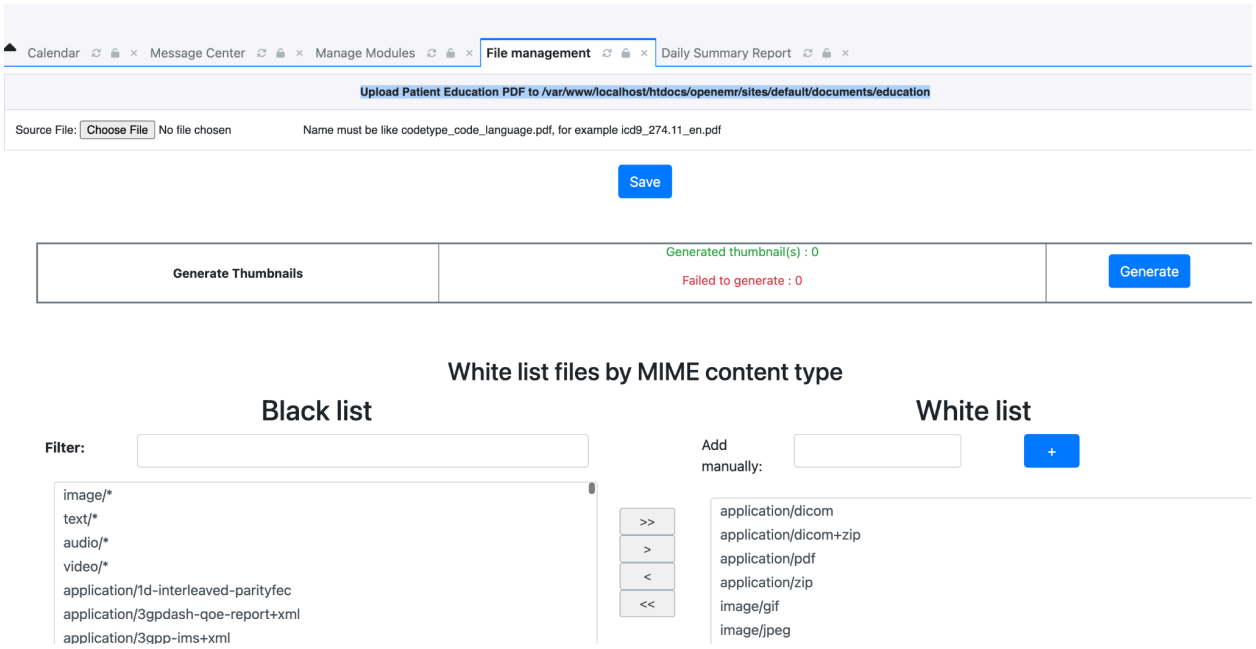
Description : Verify that the application protects against SSRF attacks, by validating or sanitizing untrusted data or HTTP file metadata, such as filenames and URL input fields, and uses allow lists of protocols, domains, paths and ports.

Steps to Replicate:

- Initiate OpenEMR at: http://localhost:80
- Sign in with given credentials.
- Go through the following menu path: Admin -> System ->Files
- We should be able to upload the files in the Text Box available on the screen

Expected Outcomes:

- The system should allow the upload of any files in PDF format
- Other file upload directory file path should not be visible



Testcase 3 : (Screenshot attached)

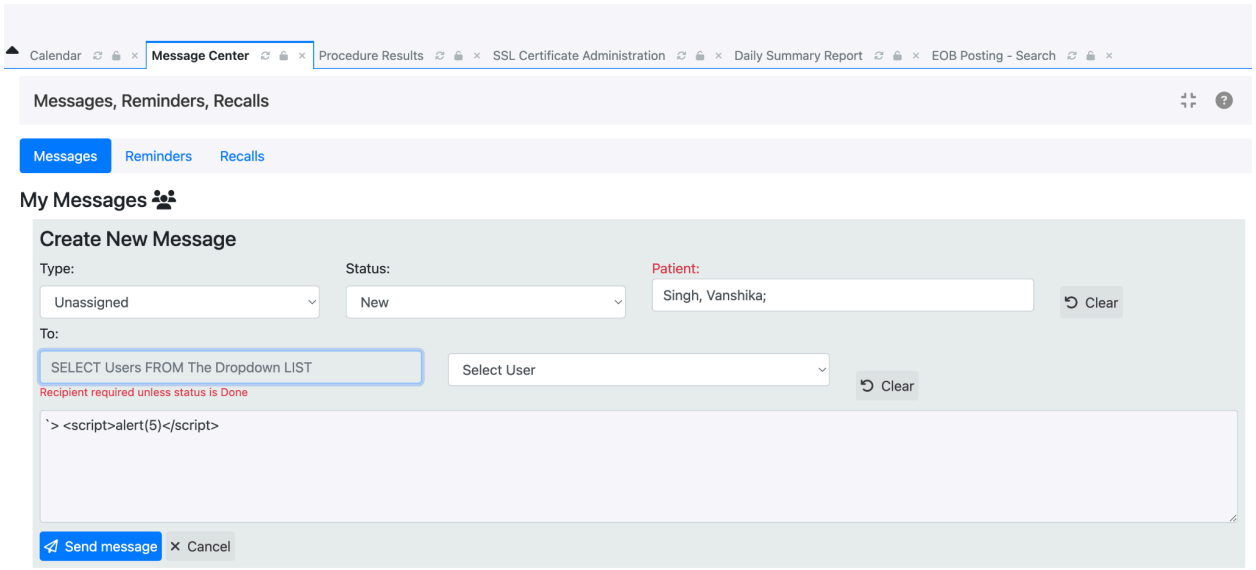
ASVS Section 7.4.1: Error Handling
Identifier: 7.4.1-1
CWE Code: 200
Description : Verify that a generic message is shown when an unexpected or security sensitive error occurs, potentially with a unique ID which support personnel can use to investigate..

Steps to Replicate:

- Initiate OpenEMR at: http://localhost:80
- Sign in with given credentials.
- Go through the following menu path: Messages -> Add New
- In the To text Field, it should be empty and we can be able to add the user

Expected Outcomes:

- The system should have a text field where we can add the name of the user to which we need to send the message.


Calendar
Message Center
Procedure Results
SSL Certificate Administration
Daily Summary Report
EOB Posting - Search
Messages, Reminders, Recalls
Messages
Reminders
Recalls
My Messages
Create New Message
Type:
Unassigned
Status:
New
Patient:
Singh, Vanshika;
Clear
To:
SELECT Users FROM The Dropdown LIST
Recipient required unless status is Done
Select User
Clear
`> <script>alert(5)</script>
Send message
Cancel

Team Member 3

Vulnerability #	Elapsed Time	Ref info for video traceability	CWE	Commentary
1	50:53	OpenEMR Penetration Testing -Nisarg.mp4	CWE-210	Generation of Error Message Containing Sensitive Information about database

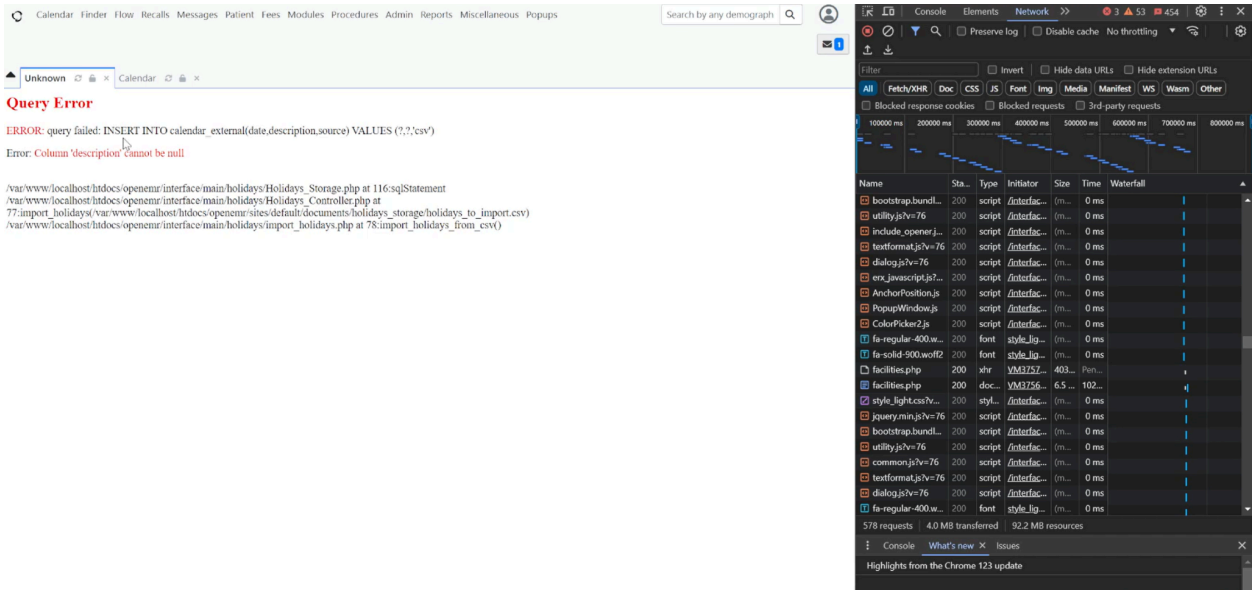
Testcase 1
ASVS Section 7.4.1 - Error Handling
Unique Identifier: 7.4.1-1
CWE: 210
Description : Verify that a generic message is shown when an unexpected or security sensitive error occurs, potentially with a unique ID which support personnel can use to investigate..

Steps to Replicate:

- Initiate OpenEMR via the URL: http://localhost:80/
- Authenticate using specified user credentials.
- Navigate through admin -> clinic -> Import Holidays via the menu.
- Attempt to upload a PDF document where a CSV file is expected and select Upload/Save.
- After attempting upload, proceed to click on Import holiday events.

Expected Outcomes:

The system should block the upload of files that are not in CSV format.
Should a non-CSV file be uploaded by any chance, the error handling should be robust, obscuring any sensitive details.


Calendar Finder Flow Recalls Messages Patient Fees Modules Procedures Admin Reports Miscellaneous Popups
Search by any demograph
Unknown
Calendar
Query Error
ERROR: query failed: INSERT INTO calendar_external(date,description,source) VALUES (?,?,'csv')
Error: Column 'description' cannot be null
/var/www/localhost/htdocs/openemr/interface/main/holidays/Holidays_Storage.php at 116:sqlStatement
/var/www/localhost/htdocs/openemr/interface/main/holidays/Holidays_Controller.php at
77:import_holidays(/var/www/localhost/htdocs/openemr/sites/default/documents/holidays_storage/holidays_to_import.csv)
/var/www/localhost/htdocs/openemr/interface/main/holidays/import_holidays.php at 78:import_holidays_from_csv()
Console Elements Network
Preserve log Disable cache No throttling
Filter Invert Hide data URLs Hide extension URLs
All Fetch/XHR Doc CSS JS Font Img Media Manifest WS Wasm Other
Blocked response cookies Blocked requests 3rd-party requests
100000 ms 200000 ms 300000 ms 400000 ms 500000 ms 600000 ms 700000 ms 800000 ms
Name Sta... Type Initiator Size Time Waterfall
bootstrap.bundl... 200 script /interfac... (m... 0 ms
utility.js?v=76 200 script /interfac... (m... 0 ms
include_opener.j... 200 script /interfac... (m... 0 ms
textformat.js?v=76 200 script /interfac... (m... 0 ms
dialog.js?v=76 200 script /interfac... (m... 0 ms
erx_javascript.js?... 200 script /interfac... (m... 0 ms
AnchorPosition.js 200 script /interfac... (m... 0 ms
PopupWindow.js 200 script /interfac... (m... 0 ms
ColorPicker2.js 200 script /interfac... (m... 0 ms
fa-regular-400.w... 200 font style_lig... (m... 0 ms
fa-solid-900.woff2 200 font style_lig... (m... 0 ms
facilities.php 200 xhr VM3757... 403... Pen...
facilities.php 200 doc... VM3756... 6.5 ... 102...
style_light.css?v... 200 styl... /interfac... (m... 0 ms
jquery.min.js?v=76 200 script /interfac... (m... 0 ms
bootstrap.bundl... 200 script /interfac... (m... 0 ms
utility.js?v=76 200 script /interfac... (m... 0 ms
common.js?v=76 200 script /interfac... (m... 0 ms
textformat.js?v=76 200 script /interfac... (m... 0 ms
dialog.js?v=76 200 script /interfac... (m... 0 ms
fa-regular-400.w... 200 font style_lig... (m... 0 ms
578 requests 4.0 MB transferred 92.2 MB resources
Console What's new Issues
Highlights from the Chrome 123 update

Team Member 4

Vulnerability #	Elapsed Time	Ref info for video traceability	CWE	Commentary
1	Video 10, 13 minutes and 20 seconds	00:13:20	CWE-1004	cookie-based session tokens have the 'HttpOnly' attribute is not set
2	Video 10, 20 minutes and 10 seconds	00:20:10	CWE-287	Verify that time-based OTP can be used more than once within the validity period.

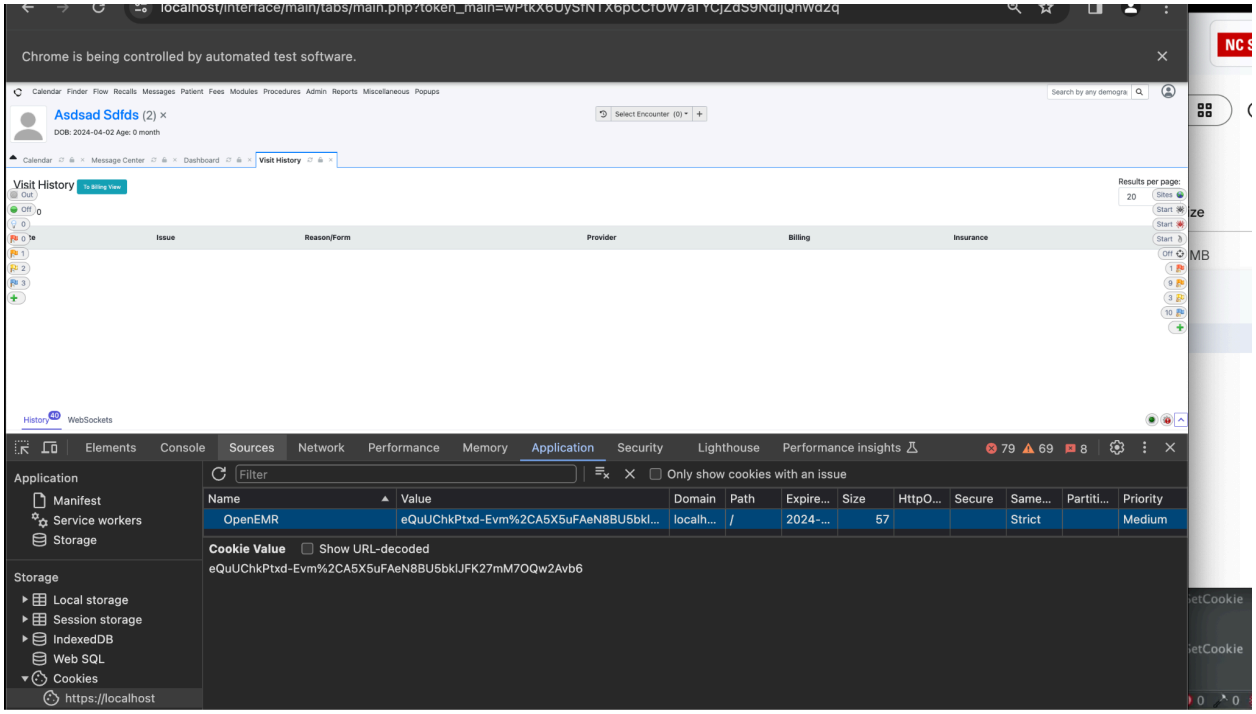
Testcase 1
ASVS Section V3.4 Cookie-based Session Management
Unique Identifier: 3.4.1-1
CWE: 1004
Description : Verify that cookie-based session tokens have the 'HttpOnly' attribute set.

Steps to Replicate:

- Launch OpenEMR application using the URL: http://localhost:80 and open Developer
- tools.
- Login using admin credentials(username: admin,password:pass)
- Now open the POST request and go to the cookies tab..

Expected Outcomes:

In the Cookies tab, a ‘HttpOnly’ attribute has to be set.



Testcase 2
ASVS SectionV2.8 One Time Verifier
Unique Identifier: 2.8.1-1
CWE: 287
Description : Verify that time-based OTP can be used only once within the validity period.

Steps to Replicate:

- Launch OpenEMR application using the URL: http://localhost:80
- Login using admin credentials(username: admin,password: Software@123)
- If you enable MFA authentication using TOTP, you'll be asked to enter a code.
- Refer to the code in the authentication app and enter it in OpenEMR.
- Now, logout and relogin with the same code before the code expires..

Expected Outcomes:

When you re-enter the same code in the last step, it should throw an exception that the code is being re-used..

.

Part 4:

Throughout the testing efforts in Project Parts 1, 2, and 3, our team dedicated a total of 84.75 hours to do entire testing. During this time, we identified 49 true positives. This results in a metric of approximately 0.52 true positive vulnerabilities found per hour of total testing effort. This metric is vital for understanding the effectiveness and efficiency of our testing processes across the different project phases.

Technique	# true positive vulnerabilities (A)	Total time (hours) (B)	Efficiency (A/B)	Detecting exploitable vulnerabilities? (high/med/low)	Unique CWE numbers
Manual black box	7	15 hours	7/15	high	CWE - 521, 598, 613, 400
SAST	6	14	6/14	high	CWE - 798, 259, 250, 311, 295, 338
DAST	3	6	3/6	high	CWE - 532, 210
Penetration testing	6	12	6/12	high	CWE - 209,116,73,210,200, 1004, 287

Write a Reflection:

When assessing the maturity of vulnerability detection techniques, both efficiency and the range of detected Common Weakness Enumeration (CWE) types must be considered. The manual black box method, despite being the most time-consuming with 15 hours of effort, managed to identify 7 vulnerabilities, displaying a mature understanding of diverse and complex exploitable vulnerabilities as indicated by the range of CWEs identified. This suggests that while time-intensive, it provides a depth of insight valuable for a comprehensive security audit.

The use of SonarQube for SAST, uncovering 6 vulnerabilities in 14 hours, shows a high level of efficiency, while also displaying a broad range of CWEs. This points to a mature capability in detecting both implementation bugs and design flaws within a reasonable timeframe, making it a well-rounded approach for static code analysis.

DAST, conducted through ZAP, while efficient in time at only 6 hours, identified fewer vulnerabilities (3 true positives). However, the high exploitable nature of the vulnerabilities it detects, as reflected by the CWEs, underlines its effectiveness in a more focused domain. The

rapid feedback loop it offers is instrumental for iterative security testing in agile development environments.

Penetration testing's performance, with 6 vulnerabilities detected in 12 hours, demonstrates a mature balance between the in-depth analysis of manual techniques and the expediency of automated tools. This method's ability to identify a wide range of CWE types suggests a comprehensive understanding of various attack vectors, aligning with a robust security testing regimen.

In summary, while SAST with SonarQube and DAST with ZAP offer more efficient means of detecting vulnerabilities, manual black box testing and penetration testing exhibit a broader range of CWE types, which is indicative of a mature understanding of the application's security landscape. Optimal vulnerability management might, therefore, entail a combination of these methods, leveraging the strengths of each to cover both the depth and breadth of potential security threats.