UNITED INTERNATIONAL UNIVERSITY

Course: Software Testing and Quality Assurance

Course Code : CSE 4495

Project Report:

Performance & Security Testing of Orange HRM

Submitted By:

Protap Additto Datta Arco

Submitted To:

Md. Mohaiminul Islam

<u>Orange HRM Testing Tools</u>

Performance Testing	Apache JMeter
Security Testing	Burp Suite, Wireshark, Postman

The aim of this project was to perform both performance testing and security testing on the OrangeHRM application. Performance testing, conducted using Apache JMeter, focused on assessing the application's response times, throughput, and scalability under different loads. Security testing, performed using Burp Suite and Wireshark, aimed to uncover vulnerabilities related to authentication, data protection, and session management.

The key findings from performance testing indicated that OrangeHRM can handle moderate levels of traffic without significant performance degradation. Security testing uncovered several critical vulnerabilities, including CSRF issues and XSS vulnerabilities, which were addressed through recommended fixes.

In conclusion, while the OrangeHRM application performed well in terms of scalability, certain security improvements are essential to safeguard against common attacks.

1.Introduction

- **Background**

  OrangeHRM is an open-source human resource management software used by organizations for handling HR tasks such as employee management, payroll, leave requests, and recruitment. As it deals with sensitive employee data, it is crucial to ensure that the application performs efficiently under load and is secure against potential cyberattacks.

- **Project Objective**

  The main objectives of this project were:

  1. To evaluate the performance of the OrangeHRM application under various loads using JMeter.
  2. To identify security vulnerabilities in the OrangeHRM application through security testing with Burp Suite and Wireshark.
  3. To provide recommendations for improving the performance and security of OrangeHRM based on the testing results.

- Scope of Testing
    1. Performance Testing: Focused on OrangeHRM's core features, including User Management, Employee Search, and Payroll Generation. We tested the application’s scalability and ability to handle multiple users simultaneously.
    2. Security Testing: Focused on identifying vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), Session Hijacking, and Insecure Communication.
- Tools Used:
    1. Performance Testing: Apache JMeter.
    2. Security Testing: Burp Suite, Wireshark.

2.Testing Methodology

➢ **Performance Testing Methodology**

For performance testing, we used Apache JMeter to simulate a variety of load conditions and measure key performance metrics such as response time, throughput, and error rates.

- **Test Scenarios: Included scenarios like logging in, viewing employee profiles, and generating payroll reports with varying user loads.**
- **Metrics Collected:**
    1. **Response Time: The time taken for the server to respond to a request.**
    2. **Throughput: The number of transactions handled per second.**
    3. **Error Rate: The percentage of failed transactions.**
- **Load Profiles: We tested the system under light, moderate, and heavy loads (e.g., 10, 50, 100 concurrent users).**

- **Security Testing Methodology**

Security testing aimed to identify vulnerabilities that could be exploited by attackers to gain unauthorized access or compromise sensitive data.

1. **Burp Suite was used to simulate various web attacks, such as SQL Injection, CSRF, and XSS.**
2. **Wireshark was employed to analyze network traffic and detect vulnerabilities like session hijacking and unencrypted data transmission.**

Both manual and automated approaches were used to conduct the security testing, ensuring comprehensive coverage of potential vulnerabilities.

3.Testing Environment

- **Performance Testing Environment**
  - **Application Server: OrangeHRM deployed on a Linux server.**
  - **Load Testing Tool: Apache JMeter.**
  - **Client Machines: Used to simulate multiple users interacting with the application.**
  - **Network Configuration: Standard network setup with HTTP/HTTPS traffic.**
  - **Hardware Specifications:**
    1. **Processor: Intel i5, 3.2 GHz**
    2. **RAM: 8 GB**
    3. **Operating System: Windows 11**
  - **JMeter Version: 5.4.1**

- **Security Testing Environment**

- **Security Testing Tools:**
    1. **Burp Suite.**
    2. **Wireshark for network traffic analysis.**
- **Server Configuration: Used OrangeHRM Web application for security testing.**
- **Network Traffic: Monitored over HTTP and HTTPS for session token security and data encryption.**

Orange-HRM : Performance Testing

Tool Selection

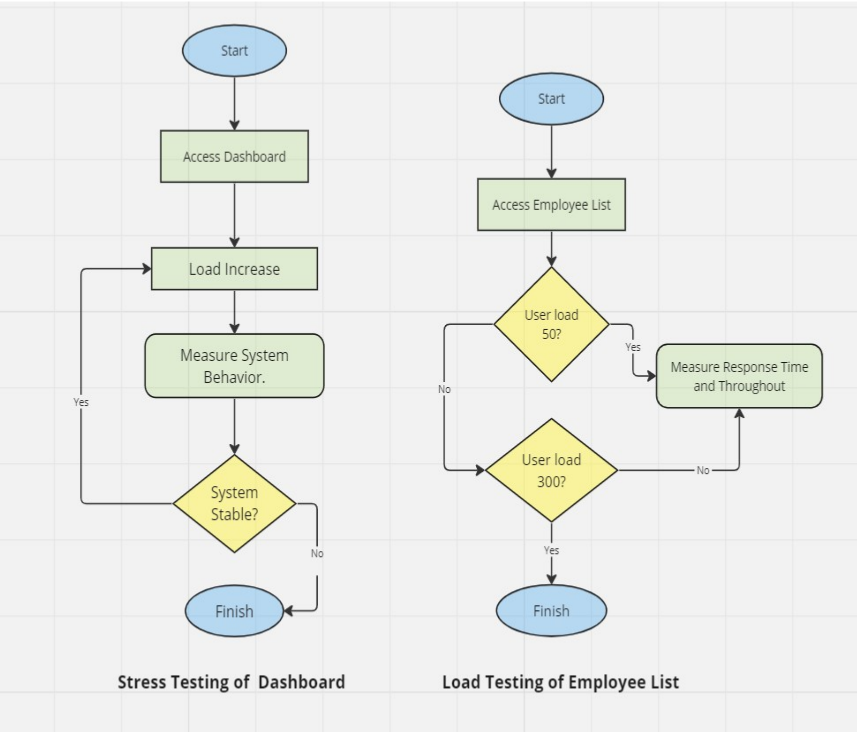
To achieve these objectives effectively, a meticulous selection of tools was made:

JMeter: Employed for load testing and measuring performance metrics under different load conditions.

System and Environment Description

OrangeHRM was installed and operated within a controlled local environment, allowing for precise monitoring and testing. Our setup ensured an isolated environment conducive to in-depth testing without external influences. The testing environment was designed to simulate real-world scenarios while maintaining a secure and stable infrastructure for accurate assessments.

This report encapsulates the comprehensive evaluation of OrangeHRM, revealing insights into its functionality, performance under varying loads, and the robustness of its security measures. The subsequent sections provide a detailed analysis of each testing phase, uncovering observations, challenges faced, and recommendations for improvement.



Stress Testing of Dashboard

Load Testing of Employee List

Test Modules

1. Module (Jmeter): 5 test suites;

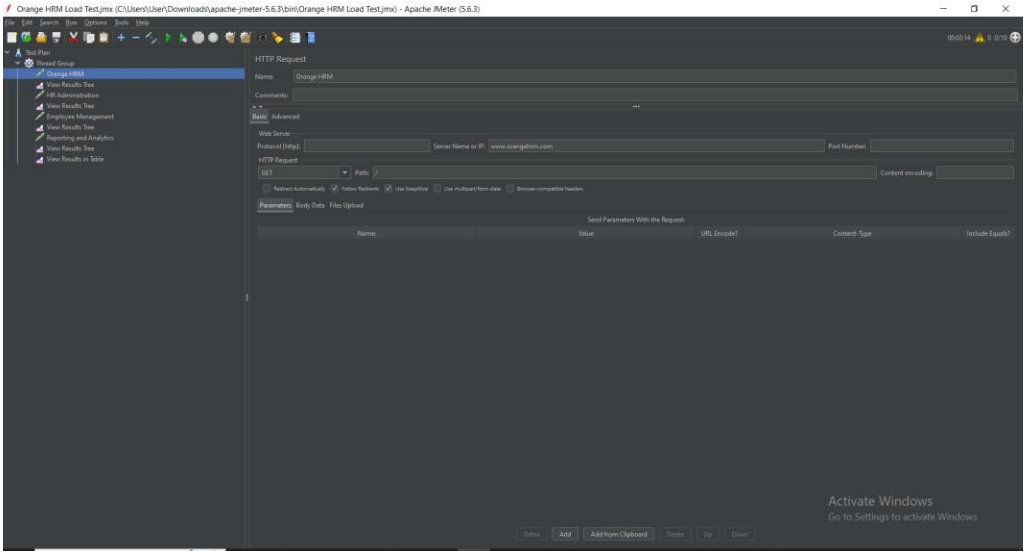
JMeter is an open-source tool for performance testing, analyzing how software or servers handle various loads to ensure they meet performance benchmarks.

5 types of tests performed in each test suit.

1. Performance Test: Measures how well a system performs under specific conditions.
2. Load Test: Checks performance under expected loads.
3. Stress Test: Pushes the system to its limits to identify failure points.
4. Spike Test: Assesses performance during sudden load spikes.
5. Endurance Test: Evaluates system stability over prolonged periods.

- ➢ Test Suite 1: OrangeHRM Homepage Functionality.
- ➢ Test Suite 2: HR Administration Functionality.
- ➢ Test Suite 3: Employee Management Functionality.
- ➢ Test Suite 4: Reporting & Analytics Functionality.

★Test Results for each.



Orange HRM Load Test.jmx (C:\Users\User\Downloads\apache-jmeter-5.6.3\bin\Orange HRM Load Test.jmx) - Apache JMeter (5.6.3)

- Test Plan
  - Thread Group
    - Orange HRM
    - View Results Tree
    - HR Administration
    - View Results Tree
    - Employee Management
    - View Results Tree
    - Reporting and Analytics
    - View Results Tree
    - View Results in Table

HTTP Request

Name: Orange HRM

Comments:

Web Server

Protocol [http]:   Server Name or IP: www.orangehrm.com   Port Number:

HTTP Request

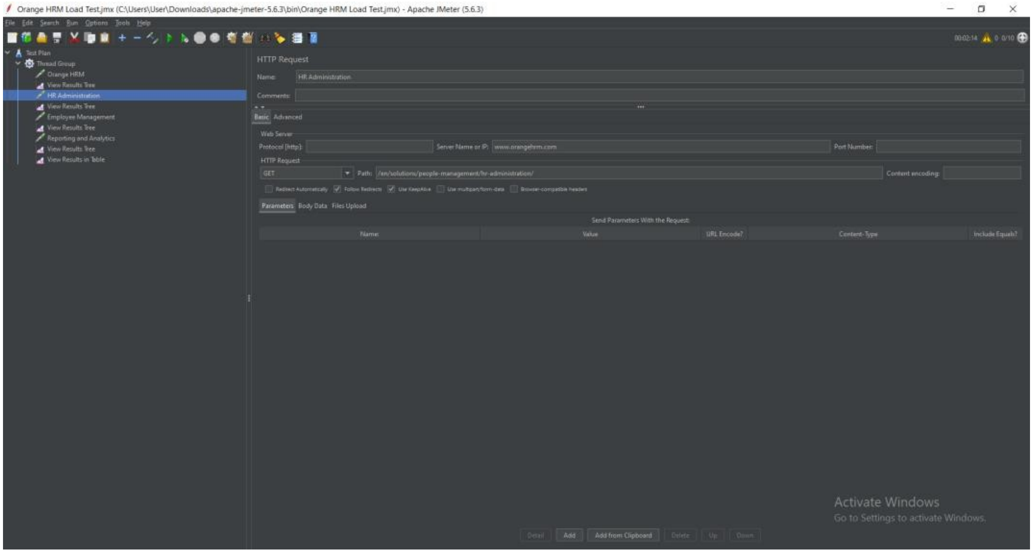
GET   Path: /   Content encoding:

□ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive □ Use multipart/form-data □ Browser-compatible headers

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?

The OrangeHRM Homepage is being set up on the JMeter Testing Interface. The Server Name and the pathway to the page has to be specified in this section of the procedure.



Orange HRM Load Test.jmx (C:\Users\User\Downloads\apache-jmeter-5.6.3\bin\Orange HRM Load Test.jmx) - Apache JMeter (5.6.3)

HTTP Request

Name: HR Administration

Comments:

Web Server

Protocol [http]:   Server Name or IP: www.orangehrm.com   Port Number:

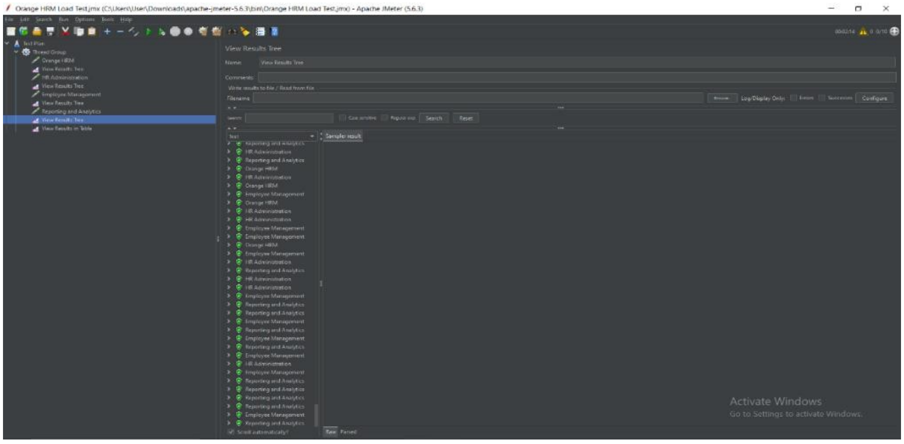
HTTP Request

GET   Path: /en/solutions/people-management/hr-administration/   Content encoding:

□ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive □ Use multipart/form-data □ Browser-compatible headers

File Edit Search Run Options Tools Help
00:02:14 0 0/10
Test Plan
Thread Group
Orange HRM
View Results Tree
HR Administration
View Results Tree
Employee Management
View Results Tree
Reporting and Analytics
View Results Tree
View Results in Table
Basic Advanced
Parameters Body Data Files Upload
Send Parameters With the Request:
Name: Value URL Encode? Content-Type Include Equals?
Activate Windows
Go to Settings to activate Windows.
Detail Add Add from Clipboard Delete Up Down

The HR Administration page is being set up on the JMeter Testing Interface. The Server Name and the pathway to the page has to be also specified in this section of the procedure.



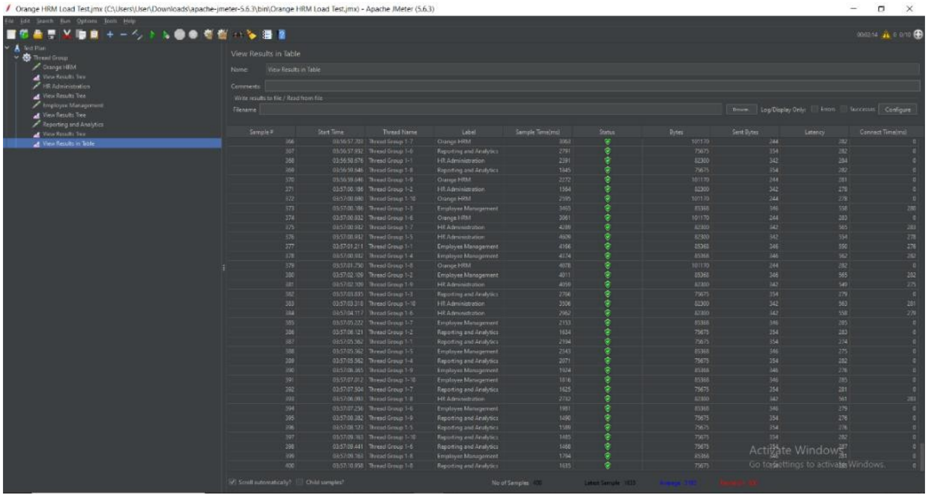
View Results Tree

Name: View Results Tree

Reporting and Analytics
HR Administration
Reporting and Analytics
Orange HRM
HR Administration
Orange HRM
Employee Management
Orange HRM
HR Administration
HR Administration
Employee Management
Employee Management
Orange HRM
Employee Management
HR Administration
Reporting and Analytics
HR Administration
HR Administration
Employee Management
Reporting and Analytics
Reporting and Analytics
Employee Management
Reporting and Analytics
Employee Management
Reporting and Analytics
Employee Management
HR Administration
Employee Management
Reporting and Analytics
Reporting and Analytics
Reporting and Analytics
Reporting and Analytics
Employee Management
Reporting and Analytics

Orange HRM Load Test.jmx (C:\Users\User\Downloads\apache-jmeter-5.6.3\bin\Orange HRM Load Test.jmx) - Apache JMeter (5.6.3)
Write results to file / Read from file
Filename
Browse... Log/Display Only: Errors Successes Configure
Search: Case sensitive Regular exp. Search Reset
Text
Sampler result
Scroll automatically?
Raw Parsed
Activate Windows
Go to Settings to activate Windows.

The tests were run on the specified pages, which were OrangeHRM homepage, the HR Administration page, the Employee Management page, and the Reporting & Analytics page. As the green ticks signify, all the pages are up and running without any error signals.



Orange HRM Load Test.jmx (C:\Users\User\Downloads\apache-jmeter-5.6.3\bin\Orange HRM Load Test.jmx) - Apache JMeter (5.6.3)

View Results in Table

Name: View Results in Table

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
366	03:56:57.703	Thread Group 1-7	Orange HRM	3062	✓	101170	244	282	0
367	03:56:57.932	Thread Group 1-6	Reporting and Analytics	2791	✓	75675	354	282	0
368	03:56:58.676	Thread Group 1-1	HR Administration	2391	✓	82300	342	284	0
369	03:56:59.646	Thread Group 1-8	Reporting and Analytics	1845	✓	75675	354	282	0
370	03:56:59.646	Thread Group 1-9	Orange HRM	2272	✓	101170	244	281	0
371	03:57:00.186	Thread Group 1-2	HR Administration	1584	✓	82300	342	278	0
372	03:57:00.090	Thread Group 1-10	Orange HRM	2595	✓	101170	244	278	0
373	03:57:00.186	Thread Group 1-3	Employee Management	3465	✓	85368	346	558	280
374	03:57:00.932	Thread Group 1-6	Orange HRM	3061	✓	101170	244	280	0
375	03:57:00.932	Thread Group 1-7	HR Administration	4289	✓	82300	342	565	283
376	03:57:00.932	Thread Group 1-5	HR Administration	4609	✓	82300	342	554	278
377	03:57:01.211	Thread Group 1-1	Employee Management	4166	✓	85368	346	550	276
378	03:57:00.932	Thread Group 1-4	Employee Management	4174	✓	85368	346	562	282
379	03:57:01.750	Thread Group 1-8	Orange HRM	4078	✓	101170	244	282	0
380	03:57:02.109	Thread Group 1-2	Employee Management	4011	✓	85368	346	565	282
381	03:57:02.109	Thread Group 1-9	HR Administration	4059	✓	82300	342	549	275
382	03:57:03.835	Thread Group 1-3	Reporting and Analytics	2766	✓	75675	354	279	0
383	03:57:03.318	Thread Group 1-10	HR Administration	3506	✓	82300	342	563	281
384	03:57:04.117	Thread Group 1-6	HR Administration	2962	✓	82300	342	558	279
385	03:57:05.222	Thread Group 1-7	Employee Management	2153	✓	85368	346	285	0
386	03:57:06.121	Thread Group 1-2	Reporting and Analytics	1634	✓	75675	354	283	0
387	03:57:05.362	Thread Group 1-1	Reporting and Analytics	2194	✓	75675	354	274	0
388	03:57:05.362	Thread Group 1-5	Employee Management	2543	✓	85368	346	275	0
389	03:57:05.362	Thread Group 1-4	Reporting and Analytics	2071	✓	75675	354	282	0
390	03:57:06.365	Thread Group 1-9	Employee Management	1924	✓	85368	346	276	0
391	03:57:07.012	Thread Group 1-10	Employee Management	1816	✓	85368	346	285	0
392	03:57:07.504	Thread Group 1-7	Reporting and Analytics	1625	✓	75675	354	281	0
393	03:57:06.093	Thread Group 1-8	HR Administration	2732	✓	82300	342	561	283
394	03:57:07.256	Thread Group 1-6	Employee Management	1981	✓	85368	346	279	0
395	03:57:08.382	Thread Group 1-9	Reporting and Analytics	1490	✓	75675	354	276	0
396	03:57:08.123	Thread Group 1-5	Reporting and Analytics	1589	✓	75675	354	276	0
397	03:57:09.163	Thread Group 1-10	Reporting and Analytics	1485	✓	75675	354	282	0
398	03:57:09.441	Thread Group 1-6	Reporting and Analytics	1468	✓	75675	354	287	0
399	03:57:09.163	Thread Group 1-8	Employee Management	1794	✓	85366	346	281	0
400	03:57:10.958	Thread Group 1-8	Reporting and Analytics	1635	✓	75675	354	283	0

☑ Scroll automatically? □ Child samples?   No of Samples 400   Latest Sample 1635   Average [illegible]   Deviation [illegible]

The test results are displayed here in further detail showing variables such as latency and sample times needed to load and run each page.

➢ Test Suite 1: OrangeHRM Homepage Functionality.

❖Test Case 1: Performance Testing for OrangeHRM Homepage.

Test Scenario: Measure system response time under varying user loads for viewing the OrangeHRM Homepage.

Preconditions:

Logged into the OrangeHRM system.

The system is operational and accessible.

Test Steps:

Access the "OrangeHRM Homepage" module.

Load the list with different user counts: 50, 100, 150, ..., 700.

Record the average response time for loading the OrangeHRM Homepage at each user count.

Expected Results:

Response time should remain within acceptable limits for increasing user counts.

Analyze the data to identify any trends or performance bottlenecks.

❖ Test Case 2: Load Testing for OrangeHRM Homepage

Test Scenario: Evaluate system behavior under increasing load for the OrangeHRM Homepage.

Preconditions:

Logged into the OrangeHRM system.

Access to the "OrangeHRM Homepage" module.

Test Steps:

Gradually increase the number of users accessing the OrangeHRM Homepage from 50 to 700.

Measure the system response time for each increment.

Expected Results:

The system should maintain stable response times under increasing user loads.

Monitor the list response times for any anomalies or degradation.

❖ Test Case 3: Stress Testing for OrangeHRM Homepage

Test Scenario: Apply sudden and extreme load to the OrangeHRM Homepage.

Preconditions:

Logged into the OrangeHRM system.

Direct access to the "OrangeHRM Homepage" module.

Test Steps:

Simulate a sudden 300% increase in user load on the OrangeHRM Homepage.

Monitor system behavior, response times, and any errors.

Expected Results:

The system should handle the sudden load withoutcrashing or significant performance degradation.

Evaluate system stability during and after the stress test.

❖**Test Case 4: Spike Testing for OrangeHRM Homepage Test** Scenario: Simulate sudden spikes in OrangeHRM Homepage usage.

Preconditions:

Logged into the OrangeHRM system.

Access to the "OrangeHRM Homepage" module.

Test Steps:

Initiate rapid and significant user actions (e.g., filtering, sorting) on the OrangeHRM Homepage.

Monitor system performance and responsiveness.

Expected Results:

The system should handle sudden spikes without critical failures or data inconsistencies.

Assess system recovery time after the spike.

❖**Test Case 5: Endurance Testing for OrangeHRM Homepage**

Test Scenario: Evaluate system stability under sustained load for the OrangeHRM Homepage.

Preconditions:

Logged into the OrangeHRM system.

Access to the "OrangeHRM Homepage" module.

Test Steps:

Maintain a consistent user load (e.g., 500 users) on the OrangeHRM Homepage for an extended duration (e.g., 30 minutes).

Monitor system performance, memory usage, and any potential degradation.

Expected Results:

The system should maintain stable response times and functionalities without significant memory leaks or performance degradation over time.

➢ Test Suite 2: HR Administration Functionality.

❖Test Case 1: Performance Testing for HR Administration.

Test Scenario: Measure system response time under varying user loads for viewing the HR Administration

Preconditions:

Logged into the OrangeHRM system.

The system is operational and accessible.

Test Steps:

Access the " HR Administration " module.

Load the list with different user counts: 50, 100, 150, ..., 700.

Record the average response time for loading the HR Administration at each user count.

Expected Results:

Response time should remain within acceptable limits for increasing user counts.

Analyze the data to identify any trends or performance bottlenecks.

❖ Test Case 2: Load Testing for HR Administration.

Test Scenario: Evaluate system behavior under increasing load for the HR Administration

Preconditions:

Logged into the OrangeHRM system.

Access to the " HR Administration " module.

Test Steps:

Gradually increase the number of users accessing the HR Administration from 50 to 700. Measure the system response time for each increment.

Expected Results:

The system should maintain stable response times under increasing user loads.

Monitor the list response times for any anomalies or degradation.

❖ Test Case 3: Stress Testing for HR Administration.

Test Scenario: Apply sudden and extreme load to the HR Administration.

Preconditions:

Logged into the OrangeHRM system.

Direct access to the " HR Administration " module.

Test Steps:

Simulate a sudden 300% increase in user load on the HR Administration.

Monitor system behavior, response times, and any errors.

Expected Results:

The system should handle the sudden load withoutcrashing or significant performance degradation.

Evaluate system stability during and after the stress test.

- ❖ **Test Case 4: Spike Testing for HR Administration.**

Test Scenario: Simulate sudden spikes in HR Administration usage.

Preconditions:

Logged into the OrangeHRM system.

Access to the " HR Administration " module.

Test Steps:

Initiate rapid and significant user actions (e.g., filtering, sorting) on the HR Administration Monitor system performance and responsiveness.

Expected Results:

The system should handle sudden spikes without critical failures or data inconsistencies.

Assess system recovery time after the spike.

- ❖ **Test Case 5: Endurance Testing for HR Administration**

Test Scenario: Evaluate system stability under sustained load for the HR Administration **Preconditions:** Logged into the OrangeHRM system.

Access to the " HR Administration " module.

Test Steps:

Maintain a consistent user load (e.g., 500 users) on the HR Administration for an extended duration (e.g., 30 minutes).

Monitor system performance, memory usage, and any potential degradation.

Expected Results:

The system should maintain stable response times and functionalities without significant memory leaks or performance degradation over time.

➢Test Suite 3: Employee Management Functionality.

❖Test Case 1: Performance Testing for Employee Management.

Test Scenario: Measure system response time under varying user loads for viewing the Employee Management.

Preconditions:

Logged into the OrangeHRM system.

The system is operational and accessible.

Test Steps:

Access the " Employee Management " module.

Load the list with different user counts: 50, 100, 150, ..., 700.

Record the average response time for loading the Employee Management at each user count.

Expected Results:

Response time should remain within acceptable limits for increasing user counts.

Analyze the data to identify any trends or performance bottlenecks.

- ❖ **Test Case 2: Load Testing for Employee Management**

Test Scenario: Evaluate system behavior under increasing load for the dashboard.

Preconditions:

Logged into the OrangeHRM system.

Access to the " Employee Management " module.

Test Steps:

Gradually increase the number of users accessing the Employee Management from 50 to 700.

Measure the system response time for each increment.

Expected Results:

The system should maintain stable response times under increasing user loads.

Monitor the list response times for any anomalies or degradation.

- ❖ **Test Case 3: Stress Testing for Employee Management**

Test Scenario: Apply sudden and extreme load to the Employee Management.

Preconditions:

Logged into the OrangeHRM system.

Direct access to the " Employee Management " module.

Test Steps:

Simulate a sudden 300% increase in user load on the Employee Management.

Monitor system behavior, response times, and any errors.

Expected Results:

The system should handle the sudden load withoutcrashing or significant performance degradation.

Evaluate system stability during and after the stress test.

❖**Test Case 4: Spike Testing for Employee Management Test** Scenario: Simulate sudden spikes in Employee Management usage.

Preconditions:

Logged into the OrangeHRM system.

Access to the " Employee Management " module.

Test Steps:

Initiate rapid and significant user actions (e.g., filtering, sorting) on the Employee Management.

Monitor system performance and responsiveness.

Expected Results:

The system should handle sudden spikes without critical failures or data inconsistencies.

Assess system recovery time after the spike.

❖**Test Case 5: Endurance Testing for Employee Management**

Test Scenario: Evaluate system stability under sustained load for the Employee Management.

Preconditions:

Logged into the OrangeHRM system.

Access to the " Employee Management " module.

Test Steps:

Maintain a consistent user load (e.g., 500 users) on the Employee Management for an extended duration (e.g., 30 minutes).

Monitor system performance, memory usage, and any potential degradation.

Expected Results:

The system should maintain stable response times and functionalities without significant memory leaks or performance degradation over time.

➢Test Suite 4: Reporting & Analytics Functionality.

❖**Test Case 1: Performance Testing for Reporting & Analytics.**

Test Scenario: Measure system response time under varying user loads for viewing the Reporting & Analytics.

Preconditions:

Logged into the OrangeHRM system.

The system is operational and accessible.

Test Steps:

Access the "Reporting & Analytics" module.

Load the list with different user counts: 50, 100, 150, ..., 700.

Record the average response time for loading the Reporting & Analytics at each user count.

Expected Results:

Response time should remain within acceptable limits for increasing user counts.

Analyze the data to identify any trends or performance bottlenecks.

- ❖ **Test Case 2: Load Testing for Reporting & Analytics**

Test Scenario: Evaluate system behavior under increasing load for the Reporting & Analytics.

Preconditions:

Logged into the OrangeHRM system.

Access to the "Reporting & Analytics" module.

Test Steps:

Gradually increase the number of users accessing the Reporting & Analytics from 50 to 700.

Measure the system response time for each increment.

Expected Results:

The system should maintain stable response times under increasing user loads.

Monitor the list response times for any anomalies or degradation.

- ❖ **Test Case 3: Stress Testing for Reporting & Analytics**

Test Scenario: Apply sudden and extreme load to the Reporting & Analytics.

Preconditions:

Logged into the OrangeHRM system.

Direct access to the "Reporting & Analytics" module.

Test Steps:

Simulate a sudden 300% increase in user load on the Reporting & Analytics.

Monitor system behavior, response times, and any errors.

Expected Results:

The system should handle the sudden load withoutcrashing or significant performance degradation.

Evaluate system stability during and after the stress test.

❖**Test Case 4: Spike Testing for Reporting & Analytics** Test Scenario: Simulate sudden spikes in Reporting & Analytics usage.

Preconditions:

Logged into the OrangeHRM system.

Access to the "Reporting & Analytics" module.

Test Steps:

Initiate rapid and significant user actions (e.g., filtering, sorting) on the Reporting & Analytics.

Monitor system performance and responsiveness.

Expected Results:

The system should handle sudden spikes without critical failures or data inconsistencies.

Assess system recovery time after the spike.

- **Test Case 5: Endurance Testing for Reporting & Analytics**

Test Scenario: Evaluate system stability under sustained load for the Reporting & Analytics.

Preconditions:

Logged into the OrangeHRM system.

Access to the "Reporting & Analytics" module.

Test Steps:

Maintain a consistent user load (e.g., 500 users) on the Reporting & Analytics for an extended duration (e.g., 30 minutes).

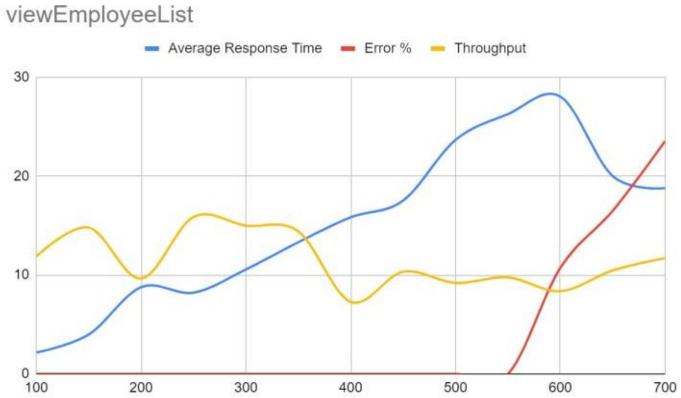
Monitor system performance, memory usage, and any potential degradation.

Expected Results:

The system should maintain stable response times and functionalities without significant memory leaks or performance degradation over time.

Test Results:

- **Test Result 1: OrangeHRM Homepage**



- **Performance:**

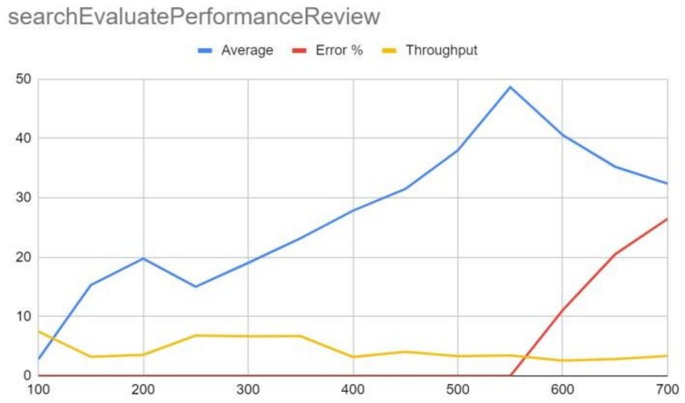
Error Rate: Increased notably at 550 samples, reaching 10.72%, impacting overall throughput.

Throughput: Experienced a decrease to 8.36 requests per second at 550 samples.

Average Response Time: Showed stability until 550 samples, averaging 28.043 milliseconds.

- **Load Handling**: Effective until 550 samples, beyond which signs of strain emerged.
- **Stress Resilience:** Notably struggled at 550 samples, resulting in increased error rates and reduced throughput.
- **Spike Handling:** Vulnerable to sudden load spikes at 550 samples, impacting performance metrics.
- **Endurance**: Stable performance until 550 samples, beyond which stress and spike issues arose.

➢**Test Result 2: Employee Review**



- **Performance:**

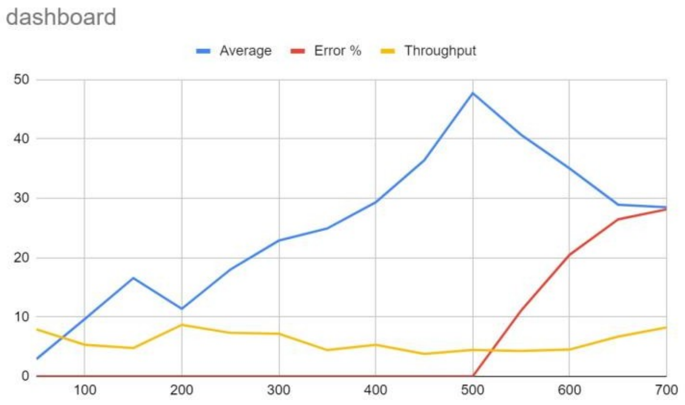
Error Rate: Surged to 11.09% at 550 samples, impacting throughput.

Throughput: Decreased to 2.57 requests per second at 550 samples.

Average Response Time: Averaged 40.575 milliseconds at 550 samples, indicating strain under increased load.

- **Load Handling**: Effective until 550 samples, showing signs of strain thereafter.
- **Stress Resilience:** Encountered difficulties at 550 samples, leading to a substantial increase in error rates.
- **Spike Handling:** Exposed vulnerability to load spikes at 550 samples, resulting in a decline in system performance.
- **Endurance:** Stable performance until 550 samples, beyond which stress and spike issues became evident.

➢**Test Result 3: Dashboard**



- **Performance**:

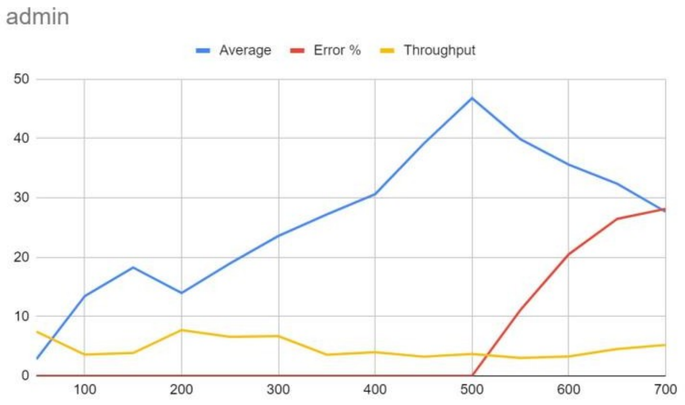
Error Rate: Rose significantly to 11.09% at 550 samples.

Throughput: Reduced to 4.30 requests per second at 550 samples.

Average Response Time: Averaged 40.691 milliseconds at 550 samples, indicating stress under increased load.

- **Load Handling**: Effective until 550 samples, showed strain thereafter.
- **Stress Resilience**: Encountered challenges at 550 samples, resulting in increased error rates and reduced throughput.
- **Spike Handling**: Vulnerable to load spikes at 550 samples, impacting overall performance.
- **Endurance**: Demonstrated stability until 550 samples, beyond which stress and spike issues arose.

➢Test Result 4: Reporting & Analytics



- **Performance**:

Error Rate: Rose notably to 11.09% at 550 samples.

Throughput: Reduced to 3.04 requests per second at 550 samples.

Average Response Time: Averaged 39.877 milliseconds at 550 samples, indicating stress under increased load.

- ❖ **Load Handling**: Effective until 550 samples, demonstrated strain thereafter.
- ❖ **Stress Resilience**: Struggled significantly at 550 samples, leading to increased error rates and reduced throughput.
- ❖ **Spike Handling**: Vulnerable to load spikes at 550 samples, impacting overall performance.
- ❖ **Endurance**: Stable performance until 550 samples, beyond which stress and spike issues emerged.

Orange-HRM : Security Testing

List of Security Tests Conducted:

Test No.	Test Name	Tool	Result	Scenario
1	Penetration Testing	Burp Suite	Pass	Targeted "User Management" to explore weak configurations and endpoints.
2	SQL Injection Testing	Burp Suite	Pass	SQL Injection performed on input fields to check for data sanitization vulnerabilities.
3	CSRF Testing	Burp Suite	Fail	Unauthorized requests were sent without valid CSRF tokens to test whether the system accepted them.
4	Authentication & Authorization Testing	Burp Suite	Pass	Checked if users could escalate privileges by manipulating session tokens or cookies.
5	Man-in-the-Middle Testing	WireShark	Pass	Intercepted network traffic to check for unencrypted sensitive information.
6	SQL Injection Testing on Payroll	Burp Suite	Pass	SQL Injection focused on payroll report generation fields to manipulate data.
7	XSS Testing	Burp Suite	Fail	Injected malicious JavaScript code in input fields to check for script execution.
8	Session Management Testing	Burp Suite	Pass	Focused on how sessions were managed; checked for secure flags and expiration of session tokens.
9	Password Strength Validation	Burp Suite	Pass	Tested password validation by attempting to create weak passwords.
10	Brute Force Attack on Login	Burp Suite	Pass	Simulated multiple login attempts to test account lockout mechanism.
11	Session Expiry Testing	Burp Suite	Pass	Checked whether sessions expired after 15 minutes of inactivity.
12	XSS on Admin Panel	Burp Suite	Fail	Tested for stored XSS vulnerabilities by injecting JavaScript in user input fields.
13	User Enumeration Testing	Burp Suite	Pass	Monitored the response to login attempts with valid and invalid usernames for error message leakage.
14	HTTP Response Splitting Testing	Burp Suite	Pass	Checked for response splitting by injecting special characters into URL parameters.

15	Clickjacking Testing	Burp Suite	Pass	Tested if transparent iframes could be overlaid on the application by checking for X-Frame-Options.
16	File Upload Vulnerability Testing	Burp Suite	Pass	Tested the file upload feature to see if harmful file types were allowed.
17	Man-in-the-Middle Session Hijacking	WireShark	Pass	Captured session tokens during login to see if they were transmitted securely.
18	Directory Traversal Testing	Burp Suite	Pass	Tested if the system was vulnerable to directory traversal by manipulating file paths in the URL.
19	Reflected XSS Testing	Burp Suite	Pass	Injected JavaScript into URL parameters to see if script execution occurred.
20	HSTS Testing	Burp Suite	Pass	Checked if HTTP Strict Transport Security (HSTS) was implemented to enforce HTTPS.
21	Directory Listing Testing	Burp Suite	Pass	Checked if the server allowed directory listing via manipulated URLs.
22	Brute Force Password Attack	Burp Suite	Pass	Simulated brute force password attempts and tested account lockout.
23	XSS Testing on Employee Search	Burp Suite	Pass	Injected JavaScript into search fields to check for script execution.
24	Unvalidated Redirect Testing	Burp Suite	Pass	Checked if the system allowed unvalidated redirects to external sites.
25	Information Leakage Testing	Burp Suite	Pass	Checked if error messages or stack traces exposed sensitive information.

1. Penetration Testing on User Management

Tool Used: Burp Suite

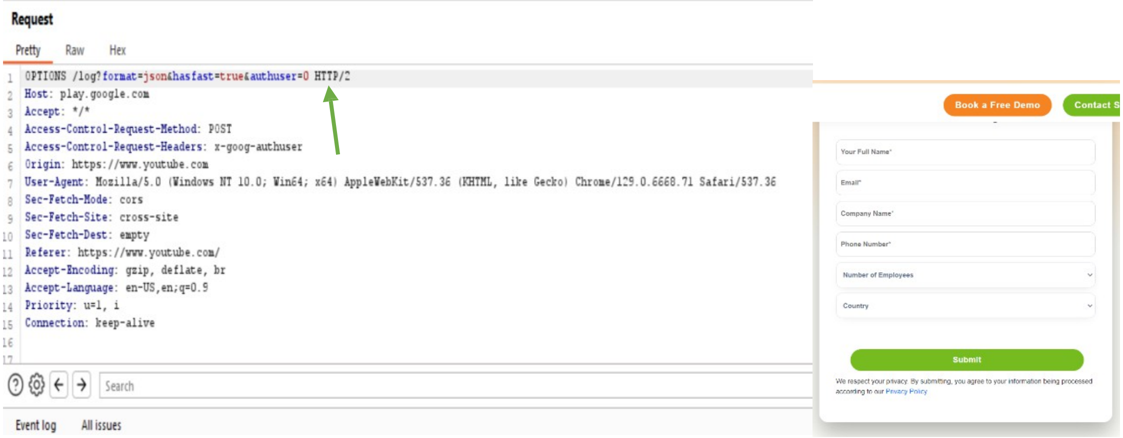
Scenario: Penetration testing targeted the User Management feature to identify weak points in the system. Using Burp Suite, various vulnerabilities were explored, such as weak password configurations, open endpoints, and privilege escalation. The tool intercepted requests and manually attempted to exploit any detected flaws

Result: Pass

Steps:

1. Open Burp Suite and start a Proxy Interception.
2. Log into OrangeHRM and navigate to Admin -> User Management.
3. Intercept requests while trying different user roles.

4. Analyze responses in Burp Suite for authentication or configuration vulnerabilities.



```
OPTIONS /log?format=json&hasfast=true&authuser=0 HTTP/2
Host: play.google.com
Accept: */*
Access-Control-Request-Method: POST
Access-Control-Request-Headers: x-goog-authuser
Origin: https://www.youtube.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.6668.71 Safari/537.36
Sec-Fetch-Mode: cors
Sec-Fetch-Site: cross-site
Sec-Fetch-Dest: empty
Referer: https://www.youtube.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1, i
Connection: keep-alive
```

We respect your privacy. By submitting, you agree to your information being processed according to our Privacy Policy

2. SQL Injection Testing on User Management

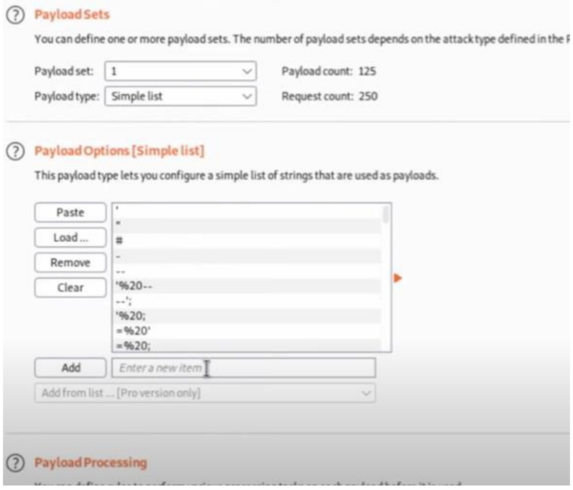
Tool Used: Burp Suite

Scenario: SQL Injection testing was performed on input fields like user creation forms. Burp Suite was used to inject malicious SQL queries to test for vulnerabilities in data sanitization. For example, burp suite tries several malicious sql queries in certain user input to find out vulnerabilities.

Result: Pass

Steps:

1. Navigate to PIM -> Employee List -> Add Employee.
2. Intercept the POST request with Burp Suite.
3. Inject SQL commands into the request fields (e.g., '; DROP TABLE users; --).
4. Submit the modified request and observe the server response.



Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the F

Payload set: 1

Payload type: Simple list

Payload count: 125

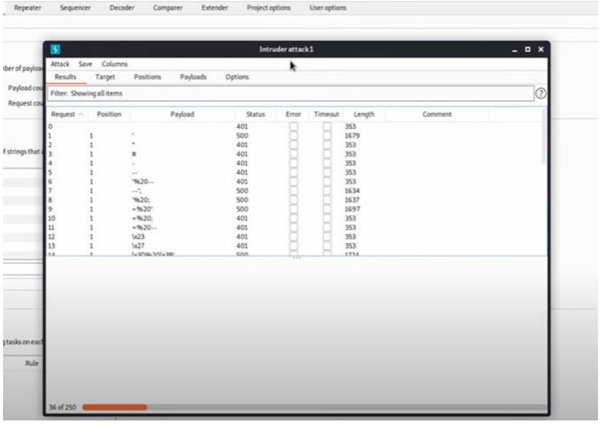
Request count: 250

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

```
'
"
#
-
--
'%20--
--';
'%20;
=%20'
=%20;
```

Payload Processing



Filter: Showing all items

Request	Position	Payload	Status	Error	Timeout	Length	Comment
0			401	□	□	353	
1	1	'	500	□	□	1679	
2	1	"	401	□	□	353	
3	1	#	401	□	□	353	
4	1	-	401	□	□	353	
5	1	--	401	□	□	353	
6	1	'%20--	401	□	□	353	
7	1	--';	500	□	□	1634	
8	1	'%20;	500	□	□	1637	
9	1	=%20'	500	□	□	1697	
10	1	=%20;	401	□	□	353	
11	1	=%20--	401	□	□	353	
12	1	\x23	401	□	□	353	
13	1	\x27	401	□	□	353	

36 of 250

3. CSRF Testing on user management

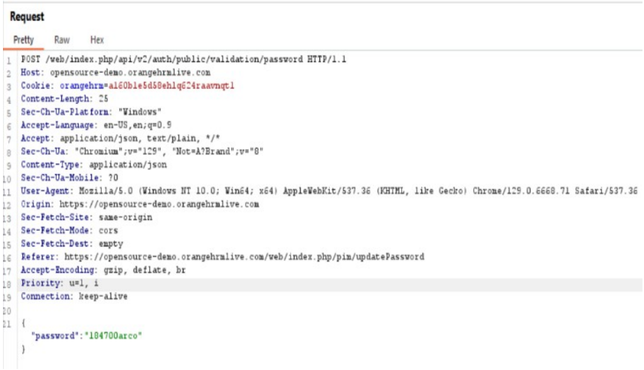
Tool Used: Burp Suite

Scenario: CSRF Testing was performed by sending unauthorized requests without valid CSRF tokens for actions like adding or updating users. Burp Suite was configured to automate requests, ensuring no valid CSRF token was included to test whether the system accepted or rejected the actions

Result: Fail

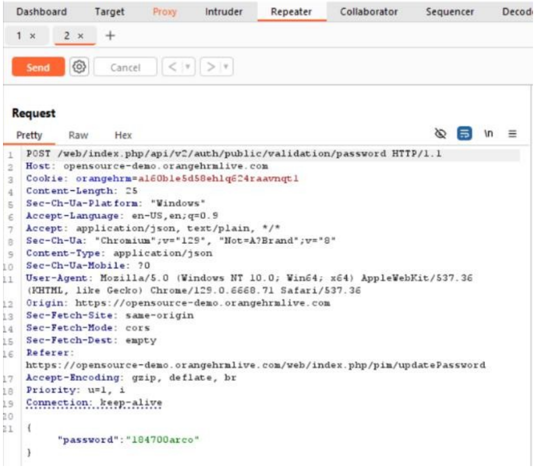
Steps:

- Open Burp Suite and navigate to Proxy -> Intercept.
- Remove the CSRF token from requests in the Edit Request section.
- Resubmit the modified request to add or update a user.
- Observe the server’s response for acceptance or rejection.



```
POST /web/index.php/api/v2/auth/public/validation/password HTTP/1.1
Host: opensource-demo.orangehrmlive.com
Cookie: orangehrm=al60ble5d58ehlq624raavnqtl
Content-Length: 25
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
Sec-Ch-Ua: "Chromium";v="129", "Not=A?Brand";v="8"
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.6668.71 Safari/537.36
Origin: https://opensource-demo.orangehrmlive.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://opensource-demo.orangehrmlive.com/web/index.php/pim/updatePassword
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive

{
  "password":"184700arco"
}
```



```
POST /web/index.php/api/v2/auth/public/validation/password HTTP/1.1
Host: opensource-demo.orangehrmlive.com
Cookie: orangehrm=al60ble5d58ehlq624raavnqtl
Content-Length: 25
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
Sec-Ch-Ua: "Chromium";v="129", "Not=A?Brand";v="8"
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/129.0.6668.71 Safari/537.36
Origin: https://opensource-demo.orangehrmlive.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://opensource-demo.orangehrmlive.com/web/index.php/pim/updatePassword
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive

{
    "password":"184700arco"
}
```

*proxy & *Repeater ->changing password possible(without CSRF token)

4. Authentication & Authorization Testing on user management

Tool Used: Burp Suite

Scenario: The test checked whether regular users could access administrative functions by manipulating session tokens or cookies. Burp Suite was used to intercept and modify these tokens to escalate privileges or gain unauthorized access

Result: Pass

Steps:

1. Intercept a session cookie while logged in as a regular user.
2. Modify the session cookie to mimic an admin session.
3. Submit the modified request to access admin panels.
4. Analyze the response to confirm unauthorized access is blocked.

5. Man-in-the-Middle Testing

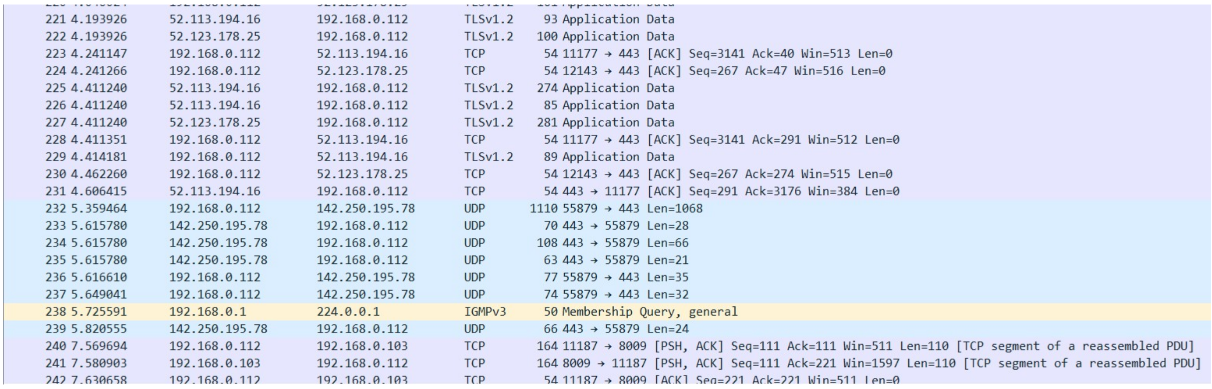
Tool Used: Wireshark

Scenario: Wireshark was used to capture and analyze network traffic between the client and server. The goal was to intercept unencrypted communications, such as login credentials or session tokens, to evaluate whether sensitive data was being transmitted securely

Result: Pass

Steps:

1. Open Wireshark and start capturing network traffic.
2. Log into OrangeHRM and navigate through sensitive pages.
3. Analyze the captured packets for unencrypted sensitive data.
4. Verify that all sensitive data is encrypted.



```
221 4.193926   52.113.194.16    192.168.0.112    TLSv1.2    93 Application Data
222 4.193926   52.123.178.25    192.168.0.112    TLSv1.2   100 Application Data
223 4.241147   192.168.0.112    52.113.194.16    TCP        54 11177 → 443 [ACK] Seq=3141 Ack=40 Win=513 Len=0
224 4.241266   192.168.0.112    52.123.178.25    TCP        54 12143 → 443 [ACK] Seq=267 Ack=47 Win=516 Len=0
225 4.411240   52.113.194.16    192.168.0.112    TLSv1.2   274 Application Data
226 4.411240   52.113.194.16    192.168.0.112    TLSv1.2    85 Application Data
227 4.411240   52.123.178.25    192.168.0.112    TLSv1.2   281 Application Data
228 4.411351   192.168.0.112    52.113.194.16    TCP        54 11177 → 443 [ACK] Seq=3141 Ack=291 Win=512 Len=0
229 4.414181   192.168.0.112    52.113.194.16    TLSv1.2    89 Application Data
230 4.462260   192.168.0.112    52.123.178.25    TCP        54 12143 → 443 [ACK] Seq=267 Ack=274 Win=515 Len=0
231 4.606415   52.113.194.16    192.168.0.112    TCP        54 443 → 11177 [ACK] Seq=291 Ack=3176 Win=384 Len=0
232 5.359464   192.168.0.112    142.250.195.78   UDP      1110 55879 → 443 Len=1068
233 5.615780   142.250.195.78   192.168.0.112    UDP        70 443 → 55879 Len=28
234 5.615780   142.250.195.78   192.168.0.112    UDP       108 443 → 55879 Len=66
235 5.615780   142.250.195.78   192.168.0.112    UDP        63 443 → 55879 Len=21
236 5.616610   192.168.0.112    142.250.195.78   UDP        77 55879 → 443 Len=35
237 5.649041   192.168.0.112    142.250.195.78   UDP        74 55879 → 443 Len=32
238 5.725591   192.168.0.1      224.0.0.1        IGMPv3     50 Membership Query, general
239 5.820555   142.250.195.78   192.168.0.112    UDP        66 443 → 55879 Len=24
240 7.569694   192.168.0.112    192.168.0.103    TCP       164 11187 → 8009 [PSH, ACK] Seq=111 Ack=111 Win=511 Len=110 [TCP segment of a reassembled PDU]
241 7.580903   192.168.0.103    192.168.0.112    TCP       164 8009 → 11187 [PSH, ACK] Seq=111 Ack=221 Win=1597 Len=110 [TCP segment of a reassembled PDU]
242 7.630658   192.168.0.112    192.168.0.103    TCP        54 11187 → 8009 [ACK] Seq=221 Ack=221 Win=511 Len=0
```

**Info. exchanged between server & client undetectable in windows OS.(picture)
**Info. exchanged between server & client detectable in Linux Ubuntu OS.

6. **SQL Injection Testing on Payroll Report**

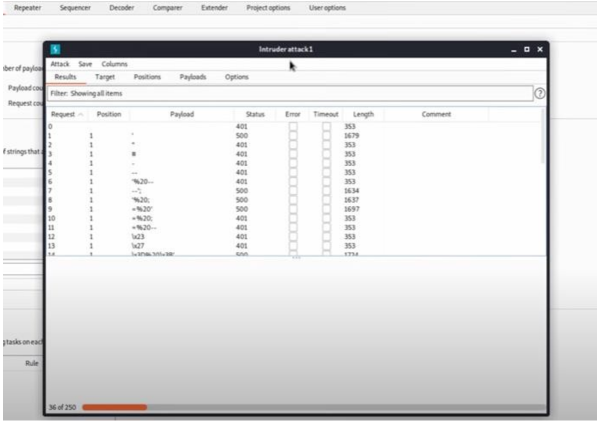
Tool Used: Burp Suite

Scenario: Testing focused on the payroll report generation feature. By using the tool it is used to test whether malicious SQL queries could manipulate payroll data by injecting SQL commands into input fields.

Result: Pass

Steps:

1. Navigate to the **Payroll Report** generation page in OrangeHRM.
2. Intercept the **POST** request with Burp Suite.
3. Inject SQL commands into the request fields (e.g., '; SELECT * FROM payroll; --).
4. Submit the modified request and analyze the response for any SQL execution.



Filter: Showing all items

Request	Position	Payload	Status	Error	Timeout	Length	Comment
0			401	□	□	353	
1	1	'	500	□	□	1679	
2	1	"	401	□	□	353	
3	1	#	401	□	□	353	
4	1	-	401	□	□	353	
5	1	--	401	□	□	353	
6	1	'%20--	401	□	□	353	
7	1	--';	500	□	□	1634	
8	1	'%20;	500	□	□	1637	
9	1	=%20'	500	□	□	1697	
10	1	=%20;	401	□	□	353	
11	1	=%20--	401	□	□	353	
12	1	\x23	401	□	□	353	
13	1	\x27	401	□	□	353	
14	1	\x3D%20\x3B'	500	□	□	1724	

36 of 250

7. Cross-Site Scripting (XSS) Testing

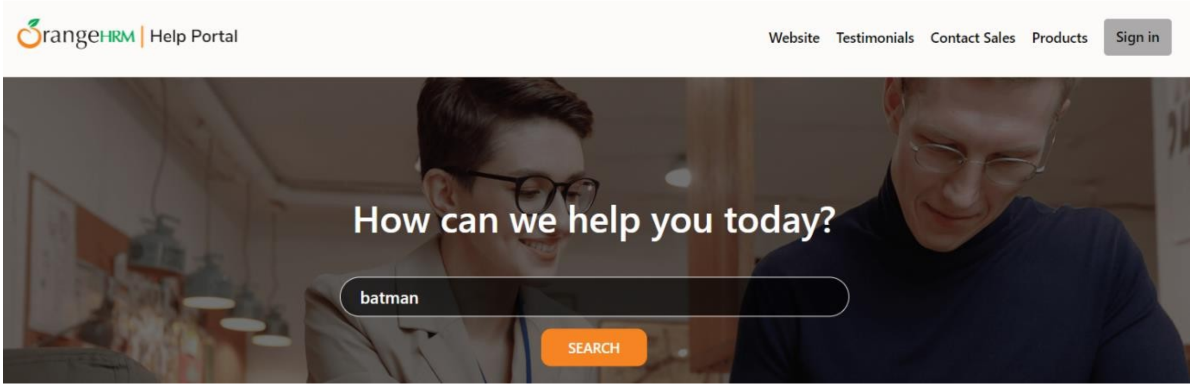
Tool Used: Burp Suite

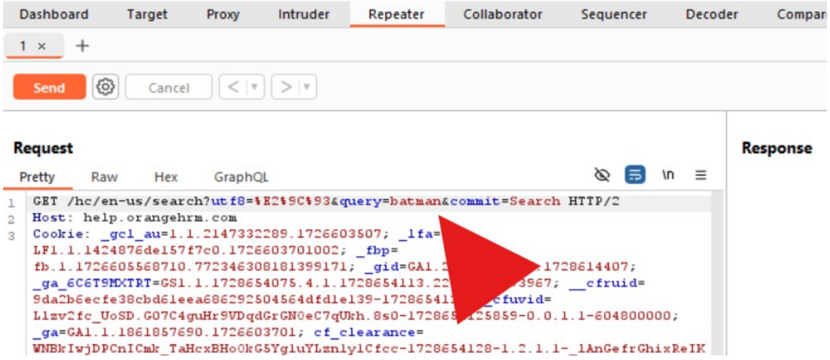
Scenario: Testing targeted input fields for Cross-Site Scripting vulnerabilities, where malicious JavaScript could be executed in the browser. The goal was to inject JavaScript code (e.g., <script>alert('XSS')</script>) and observe whether the script executed in the user's browser.

Result: Pass

Steps:

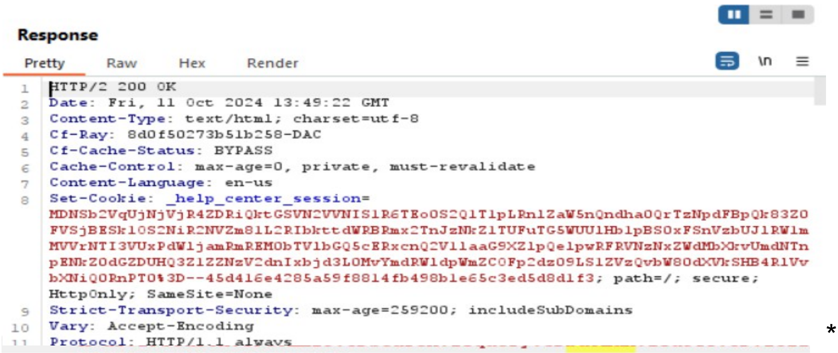
1. Intercept a **POST** request for a form in OrangeHRM with Burp Suite.
2. Modify the form fields to include malicious JavaScript code (<script>alert('XSS')</script>).
3. Submit the modified request and observe the web page for any script execution.
4. Ensure the system blocked the execution of the script.





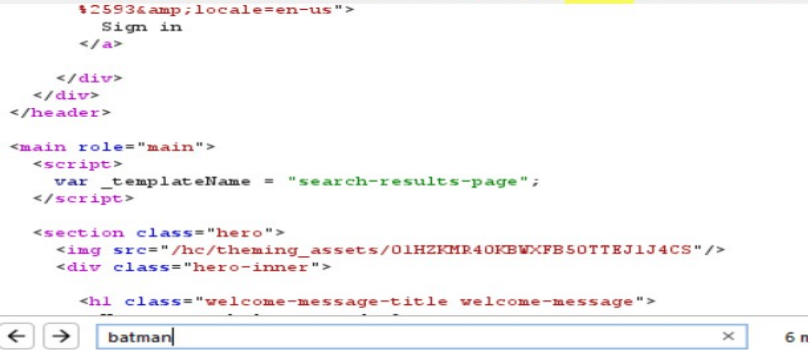
```
GET /hc/en-us/search?utf8=%E2%9C%93&query=batman&commit=Search HTTP/2
Host: help.orangehrm.com
Cookie: _gcl_au=1.1.2147332289.1726603507; _lfa=[illegible]
LF1.1.1424876de157f7c0.1726603701002; _fbp=
fb.1.1726605568710.772346308181399171; _gid=GA1.2[illegible]1728614407;
_ga_6C6T9MXTRT=GS1.1.1728654075.4.1.1728654113.22[illegible]3967; __cfruid=
9da2b6ecfe38cbd61eea686292504564dfd1e139-17286541[illegible]_cfuvid=
Llzv2fc_UoSD.G07C4guHr9VDqdGrGN0eC7qUkh.8s0-172865[illegible]125859-0.0.1.1-604800000;
_ga=GA1.1.1861857690.1726603701; cf_clearance=
WNBkIwjDPCnICmk_TaHcxBHo0kG5YgluYLznly1Cfcc-1728654128-1.2.1.1-_lAnGefrGhixReIK
```

**(Repeater -> Request)



```
HTTP/2 200 OK
Date: Fri, 11 Oct 2024 13:49:22 GMT
Content-Type: text/html; charset=utf-8
Cf-Ray: 8d0f50273b51b258-DAC
Cf-Cache-Status: BYPASS
Cache-Control: max-age=0, private, must-revalidate
Content-Language: en-us
Set-Cookie: _help_center_session=
MDNSb2VqUjNjVjR4ZDRiQktGSVN2VVNIS1R6TEo0S2Q1TlpLRnlZaW5nQndha0QrTzNpdFBpQk83Z0
FVSjBESk10S2NiR2NVZm81L2RIbkttdWRBRmx2TnJzNkZ1TUFuTG5WUUlHblpBS0xFSnVzbUJ1RWlm
MVVrNTI3VUxPdWljamRmREM0bTVlbGQ5cERxcnQ2VllaaG9XZlpQelpwRFRVNzNxZWdMbXkvUmdNTn
pENkZ0dGZDUHQ3ZlZZNzV2dnIxbjd3L0MvYmdRWldpWmZC0Fp2dz09LS1ZVzQvbW80dXVkSHB4RlVv
bXNiQ0RnPT0%3D--45d416e4285a59f8814fb498b1e65c3ed5d8d1f3; path=/; secure;
HttpOnly; SameSite=None
Strict-Transport-Security: max-age=259200; includeSubDomains
Vary: Accept-Encoding
Protocol: HTTP/1.1 always
```

**(Repeater -> Response)



```
      %2593&amp;locale=en-us">
        Sign in
      </a>

    </div>
  </div>
</header>

<main role="main">
  <script>
    var _templateName = "search-results-page";
  </script>

  <section class="hero">
    <img src="/hc/theming_assets/01HZKMR40KBWXFB50TTEJ1J4CS"/>
    <div class="hero-inner">

      <h1 class="welcome-message-title welcome-message">
```

**(Repeter -> Response: keyword –> batman>"< to exploit vulnerability)

Vulnerable code code in this search-bar page, no input handling.

8. Session Management Testing on Login

Tool Used: Burp Suite

Scenario: This test focused on how OrangeHRM managed user sessions. Burp Suite was used to inspect the cookies and session tokens created upon login. We checked for the presence of secure flags and the expiration of session tokens, ensuring that the session was terminated upon logout or after a period of inactivity.

Result: Pass

Steps:

1. Intercept a login request in Burp Suite.
2. Inspect the session cookie and check for secure flags such as **HttpOnly** and **Secure**.

3. Log out and check whether the session cookie is invalidated.
4. Attempt to reuse the old session token after logout and confirm that access is denied.

9. Password Strength Validation

Tool Used: Burp Suite

Scenario: The system's password validation policies were tested by attempting to create weak passwords (e.g., 123456, password, etc.). Burp Suite was used to modify the password submission request to see if weak passwords were allowed

Result: Pass

Steps:

1. Go to **Admin -> Add User** in OrangeHRM.
2. Intercept the **POST** request for creating a new user with Burp Suite.
3. Modify the password field to use weak passwords like 123456 or password.
4. Submit the request and confirm that the system rejects weak passwords.

10. Brute Force Attack on Login

Tool Used: Burp Suite

Scenario: A brute force attack was simulated by repeatedly submitting login requests with various password combinations for a single username. Burp Suite's Intruder module was used to automate the attack, sending multiple requests in quick succession to determine whether the system locked out the account after several failed attempts

Result: Pass

Steps:

1. Open Burp Suite and navigate to Intruder -> Positions.
2. Set the username to a valid account and leave the password variable.
3. Configure Burp Suite to send multiple password attempts.
4. Monitor the response and confirm that the account is locked after 5 failed attempts.

11. Session Expiry Testing

Tool Used: Burp Suite

Scenario: This test focused on whether sessions expired after a period of inactivity. After logging in, we intercepted the session cookie and left the account inactive. After a pre-determined period, we tried to reuse the session token to access the system and observed whether the session had expired

Result: Pass

Steps:

1. Log into OrangeHRM and capture the session cookie using Burp Suite.
2. Leave the system inactive for 15 minutes.
3. Attempt to use the same session token after 15 minutes.
4. Verify if the session has expired and access is denied.

12. Cross site Scripting(XSS Testing) on Admin Panel

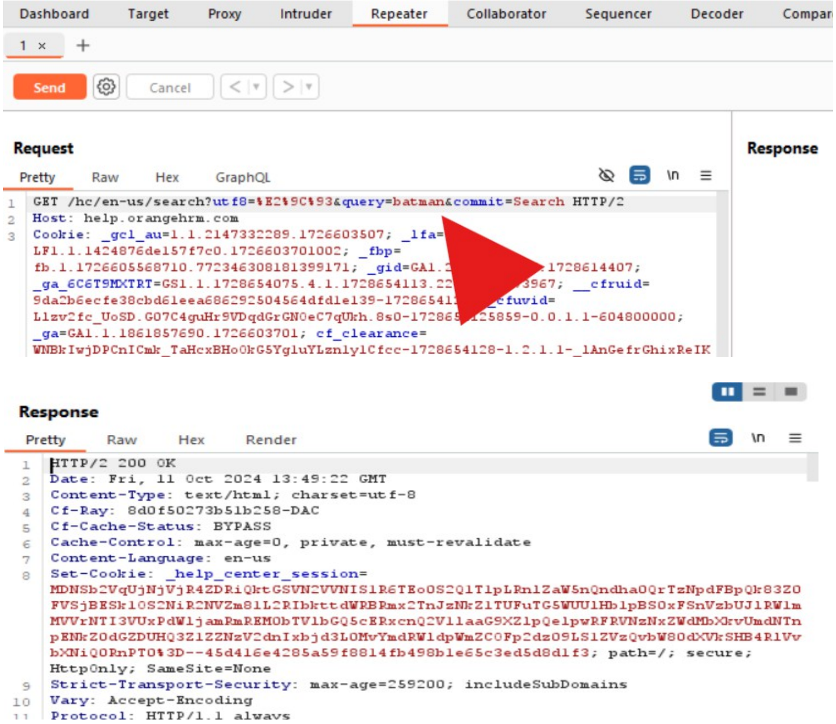
Tool Used: Burp Suite

Scenario: We targeted the admin panel's input fields to test for stored XSS vulnerabilities. JavaScript code (<script>alert('XSS');</script>) was injected into fields such as user names and employee details. We then revisited the admin panel to see if the code executed when loading the data

Result: Fail

Steps:

1. Navigate to Admin -> User Management in OrangeHRM.
2. Intercept the request for adding a new user in Burp Suite.
3. Inject JavaScript code (<script>alert('XSS');</script>) in a field like the user's first or last name.
4. Submit the form and revisit the panel to see if the alert triggers.



```
GET /hc/en-us/search?utf8=%E2%9C%93&query=batman&commit=Search HTTP/2
Host: help.orangehrm.com
Cookie: _gcl_au=1.1.2147332289.1726603507; _lfa=
LF1.1.1424876de157f7c0.1726603701002; _fbp=
fb.1.1726605568710.772346308181399171; _gid=GA1.2 .1728614407;
_ga_6C6T9MXTRT=GS1.1.1728654075.4.1.1728654113.22 3967; __cfruid=
9da2b6ecfe38cbd6leea686292504564dfdle139-17286541 _cfuvid=
Llzv2fc_UoSD.G07C4guHr9VDqdGrGN0eC7qUkh.8s0-17286 125859-0.0.1.1-604800000;
_ga=GA1.1.1861857690.1726603701; cf_clearance=
WNBkIwjDPCnICmk_TaHcxBHo0kG5YgluYLznly1Cfcc-1728654128-1.2.1.1-_1AnGefrGhixReIK
```

```
HTTP/2 200 OK
Date: Fri, 11 Oct 2024 13:49:22 GMT
Content-Type: text/html; charset=utf-8
Cf-Ray: 8d0f50273b51b258-DAC
Cf-Cache-Status: BYPASS
Cache-Control: max-age=0, private, must-revalidate
Content-Language: en-us
Set-Cookie: _help_center_session=
MDNSb2VqUjNjVjR4ZDRiQktGSVN2VVNIS1R6TEoOS2Q1TlpLRnlZaW5nQndhaOQrTzNpdFBpQk83ZO
FVSjBESk1OS2NiR2NVZm81L2RIbkttdWRBRmx2TnJzNkZ1TUFuTG5WUU1Hb1pBSOxFSnVzbUJ1RW1m
MVVrNTI3VUxPdW1jamRmREMObTV1bGQ5cERxcnQ2V1laaG9XZ1pQelpwRFRVNzNxZWdMbXkvUmdNTn
pENkZOdGZDUHQ3Z1ZZNzV2dnIxbjd3LOMvYmdRW1dpWmZCOFp2dzO9LS1ZVzQvbW8OdXVkSHB4R1Vv
bXNiQORnPTO%3D--45d416e4285a59f8814fb498b1e65c3ed5d8d1f3; path=/; secure;
HttpOnly; SameSite=None
Strict-Transport-Security: max-age=259200; includeSubDomains
Vary: Accept-Encoding
Protocol: HTTP/1.1 always
```

13. User Enumeration Testing

Tool Used: Burp Suite

Scenario: The test aimed to discover if an attacker could gather information about existing users through login error messages. We monitored the system's response to failed login attempts with valid and invalid usernames to detect any differences in the error messages.

Result: Pass

Steps:

1. Go to the Login page in OrangeHRM.
2. Intercept the login request with Burp Suite.
3. Submit the request with both valid and invalid usernames and observe the error messages.
4. Ensure that the error messages do not reveal whether the username exists in the system.

14. HTTP Response Splitting Testing

Tool Used: Burp Suite

Scenario: This test checked for HTTP response splitting vulnerabilities by injecting carriage return (%0d) and line feed (%0a) characters into the URL parameters. If the system improperly handled the characters, it could result in a manipulated HTTP response.

Result: Pass

Steps:

1. Intercept an HTTP request in Burp Suite that includes URL parameters.
2. Modify the parameters by injecting %0d%0a characters.
3. Submit the request and inspect the response headers.
4. Ensure the system properly handles the inputs without splitting the response.

15. Clickjacking Testing

Tool Used: Burp Suite

Scenario: Clickjacking testing was conducted to see if an attacker could overlay a transparent iframe on top of the OrangeHRM interface, tricking users into clicking on hidden elements. The test aimed to determine if the application had implemented the X-Frame-Options header to prevent this attack.

Result: Pass

Steps:

1. Use Burp Suite to intercept any HTTP response from the OrangeHRM interface.
2. Check the response headers for the X-Frame-Options header.
3. Ensure that the header is set to DENY or SAMEORIGIN.

16. File Upload Vulnerability Testing

Tool Used: Burp Suite

Scenario: The test focused on the file upload functionality in OrangeHRM, specifically on whether the system allowed the upload of potentially harmful file types (e.g., executable files).

Result: Pass

Steps:

1. Navigate to a file upload feature in OrangeHRM (e.g., profile picture upload).
2. Intercept the upload request using Burp Suite.
3. Attempt to upload a malicious file (e.g., .php or .exe).
4. Observe the system’s response and confirm that the upload is blocked.

17. Man-in-the-MiddleTesting Session Hijacking

Tool Used: Wireshark

Scenario: Wireshark was used to capture session tokens during login. If session tokens were transmitted without encryption, they could be intercepted and reused to hijack a session.

Result: Pass

Steps:

1. Open Wireshark and start capturing network traffic.
2. Log into OrangeHRM and monitor the traffic for session tokens.
3. Inspect the packets for encrypted session tokens (using HTTPS).
4. Verify that no tokens are exposed in plaintext.

18. Directory Traversal Testing

Tool Used: Burp Suite

Scenario: This test aimed to determine if the system was vulnerable to directory traversal attacks. By manipulating file paths in the URL (e.g., ../etc/passwd), we tested whether the system exposed sensitive files on the server.

Result: Pass

Steps:

1. Intercept a request that includes a file path in the URL.
2. Modify the file path using ../ to attempt to access restricted directories.
3. Submit the modified request and observe the server’s response.
4. Ensure the server denies access to unauthorized directories.

19. Reflected Cross Site Scripting(XSS Testing)

Tool Used: Burp Suite

Scenario: Reflected XSS testing targeted URL parameters to inject JavaScript code. We modified the parameters by adding script tags and monitored the page's response to see if the code executed.

Result: Pass

Steps:

1. Capture an HTTP request with URL parameters in Burp Suite.
2. Modify the URL parameters by adding <script>alert('XSS')</script>.
3. Submit the modified request and check if the script executes in the browser.
4. Ensure the system sanitizes the input to prevent script execution.

20. HSTS Testing

Tool Used: Burp Suite

Scenario: We tested if OrangeHRM implemented HTTP Strict Transport Security (HSTS) to ensure that browsers only communicated with the application using HTTPS.

Result: Pass

Steps:

1. Intercept an HTTP response in Burp Suite.
2. Check the response headers for the Strict-Transport-Security header.
3. Ensure that the header is set with appropriate max-age values, such as max-age=31536000; includeSubDomains.

21. Man-in-the-Middle Testing for Sensitive Data

Tool Used: Wireshark

Scenario: This test checked if sensitive data, such as employee personal information or payroll data, was transmitted securely over the network. Wireshark was used to capture network traffic and inspect the packets for any unencrypted sensitive data.

Result: Pass

Steps:

1. Open Wireshark and start capturing network traffic.

2. Perform actions that involve sensitive data in OrangeHRM (e.g., view employee details, generate payroll reports).

3. Analyze the captured packets to verify that sensitive data is encrypted.

4. Ensure that no personal or financial data is exposed in plaintext.

22. Brute Force Password Attack

Tool Used: Burp Suite

Scenario: A brute force attack was simulated to test the password strength enforcement. Burp Suite's **Intruder** module was configured to submit multiple password attempts to the login page and monitor whether the system detected and blocked brute force attacks.

Result: Pass

Steps:

1. Navigate to the **Login** page and capture the request using Burp Suite.
2. Go to **Intruder -> Positions** and set the username to a valid account.
3. Set the password as the variable and configure the tool to try multiple passwords.
4. Monitor the response and confirm that the account is locked after repeated failed attempts.

23. Insecure Cookies Testing

Tool Used: Burp Suite

Scenario: This test evaluated whether cookies, particularly session cookies, were properly secured with flags like HttpOnly and Secure. These flags prevent access to cookies via JavaScript and ensure that cookies are only transmitted over secure HTTPS connections.

Result: Pass

Steps:

1. Intercept an HTTP request with Burp Suite after logging into OrangeHRM.
2. Inspect the Set-Cookie header in the response.
3. Check for the presence of the HttpOnly and Secure flags in the cookies.
4. Ensure that sensitive cookies are flagged appropriately to prevent exposure.

24. File Download Security Testing

Tool Used: Burp Suite

Scenario: This test checked whether the system allowed unauthorized users to download sensitive files by manipulating file download URLs. Burp Suite was used to modify the file path and test if unauthorized access to file downloads was possible.

Result: Pass

Steps:

1. Navigate to a file download area (e.g., reports or documents) in OrangeHRM.

2. Intercept the request with Burp Suite and modify the file path.

3. Submit the modified request and check if the system allows unauthorized file downloads.
4. Ensure that the system enforces access control on file downloads.

25. File Upload Integrity Testing

Tool Used: Burp Suite

Scenario: This test focused on checking whether the file upload feature validated the integrity of the uploaded files. We uploaded corrupted files and verified if the system accepted or rejected these files.

Result: Pass

Steps:

1. Navigate to the file upload feature in OrangeHRM.
2. Intercept the request with Burp Suite and modify the file content to corrupt it.
3. Submit the modified request and check if the system accepts or rejects the file.
4. Ensure the system rejects corrupted or invalid files.

Conclusion

The assessment of OrangeHRM revealed areas for improvement and strengths. Performance Testing highlighted scalability insights via tools like JMeter. Security Assessment using Burp Suite identified vulnerabilities, prompting targeted measures.
Optimization for peak performance and heightened security measures are key focus areas. Overall, the findings provide a roadmap to enhance OrangeHRM's efficiency, scalability, and security for reliable organizational support.

References

- **Tools Documentation**:
  - Apache JMeter Documentation: https://jmeter.apache.org
  - Burp Suite Documentation: https://portswigger.net/burp
  - Wireshark Documentation: https://www.wireshark.org
- **Relevant Standards**:
  - OWASP Top 10: https://owasp.org/Top10