

Stored Cross-Site Scripting (XSS) vulnerability was found on the manage-employee.php page of the Kashipara Online Attendance Management System project V1.0. This vulnerability allows remote attackers to execute arbitrary scripts attack in /attendance/admin/manage-employee.php URL.

## Vulnerability Overview

### ➤ Official Website URL

<https://www.kashipara.com/project/php/13168/online-attendance-management-system-php-project-source-code>

### ➤ Affected Product Name: - Online Attendance Management System

<b>Affected Vendor</b>	Kashipara
<b>Affected Code File</b>	/attendance/admin/manage-employee.php
<b>Affected Parameter</b>	department
<b>Method</b>	POST
<b>Type</b>	Stored Cross-Site Scripting (XSS)
<b>Version</b>	V1.0

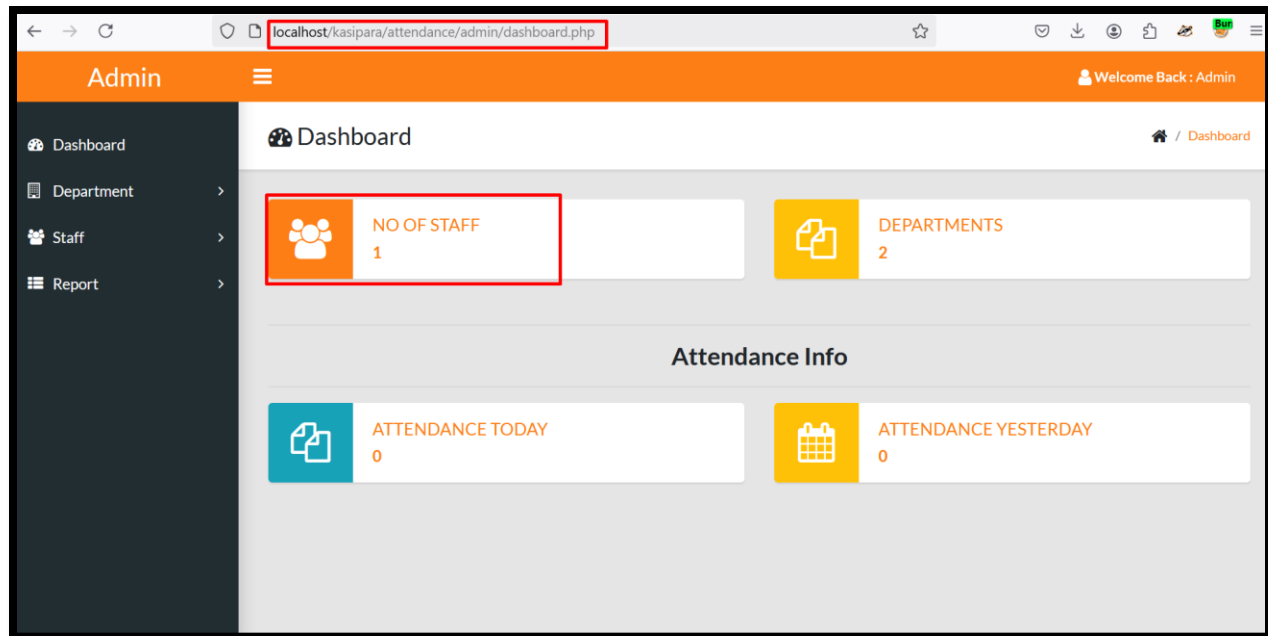
## Vulnerability Description

A stored XSS vulnerability exists in Online Attendance Management System version V1.0, allowing an attacker to inject malicious scripts via the “**department**” field in the manage employee section of a Admin Panel. The malicious payload executes when the admin reviews the comments, potentially compromising the admin account and the underlying application.

## Steps to Reproduce

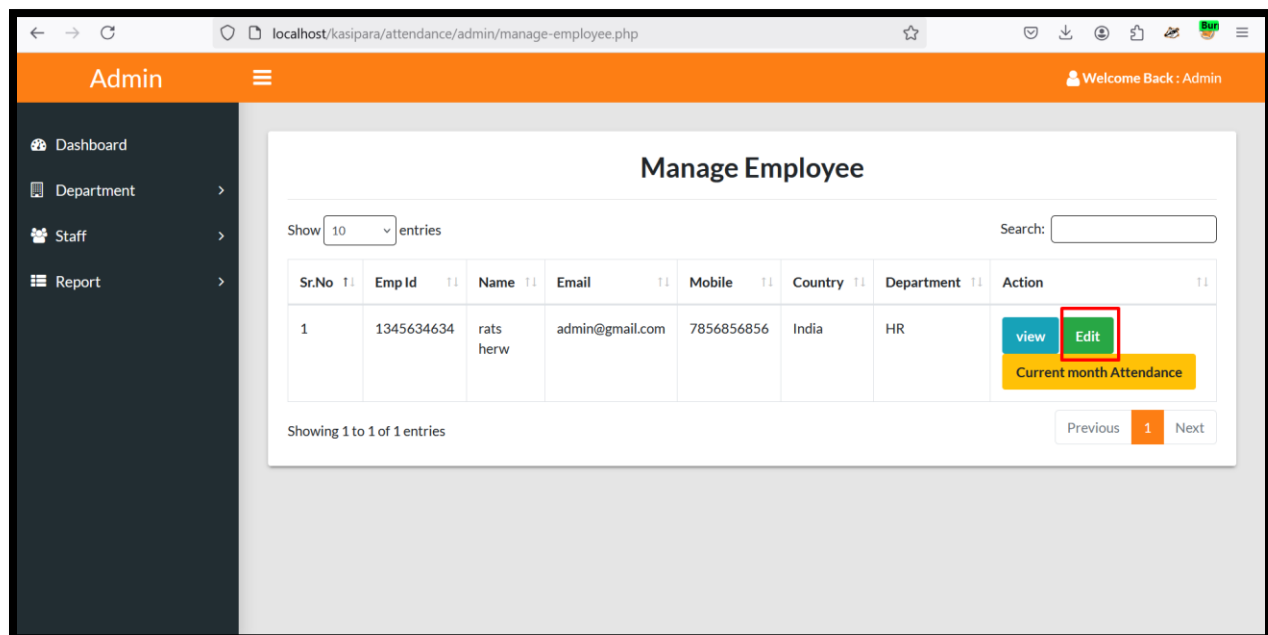
### 1. Access the Target application: -

- Navigate to the admin account and their Dashboard  
<http://localhost/kasipara/attendance/admin/dashboard.php>
- Log in with **valid credentials**
- After accessing the dashboard, click on the **NO OF STAFF** area.



## 2. Switch in the Manage Employee Section: -

- When you click on the number of staff area, you will see the Manage Employees section as shown in your images.
- After this you have to click on the edit button as shown in the images.



### 3. Navigate the Edit Page: -

- Then you will see an edit employee page open.
- All the required fields must be filled in.

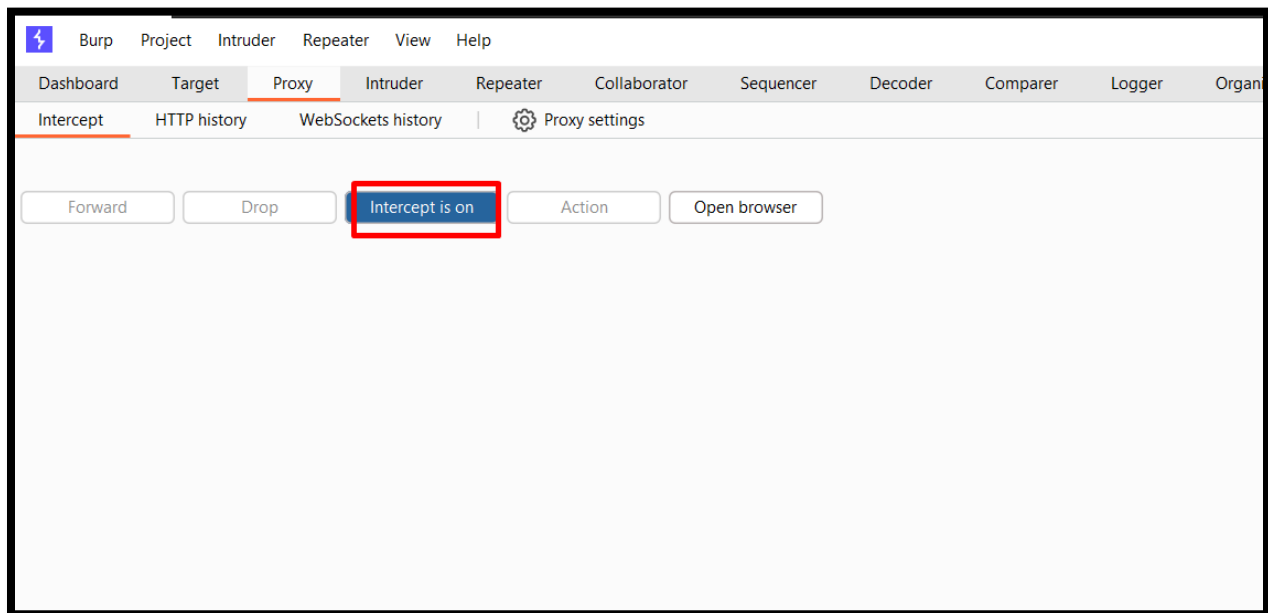
The screenshot shows a web application interface for an administrator. The top navigation bar is orange and labeled 'Admin'. A sidebar on the left contains links to 'Dashboard', 'Department', 'Staff', and 'Report'. The main content area is titled 'Edit Employee' and contains a form with the following fields:

- Emp ID: 1345634634
- First Name: rats
- Last Name: herw
- Department: HR (dropdown menu)
- Email ID: admin@gmail.com
- Mobile No: 7856856856
- Country: India
- State: RJ
- City: New Delhi
- DOB: 01 / 26 / 1999
- Date of Joining: 01 / 24 / 2022
- Photo: A small image of a man with a beard.
- Address: test

At the bottom of the form, there is a red 'Update' button. The browser's address bar shows the URL: localhost/kasipara/attendance/admin/edit-employee.php?empid=0.

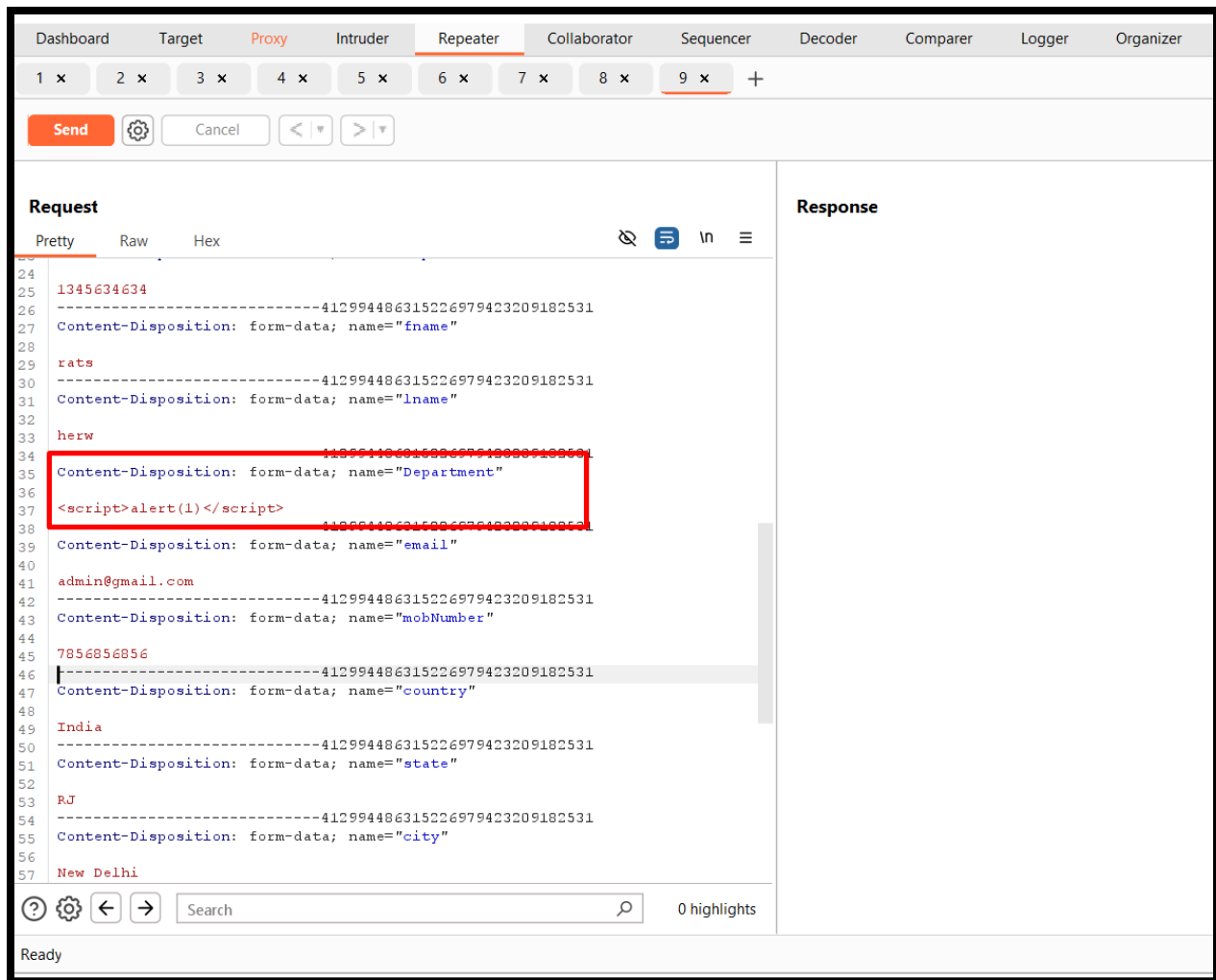
### 4. Intercept the Request:

- Enable Burp Suite and set up the browser to route traffic through it



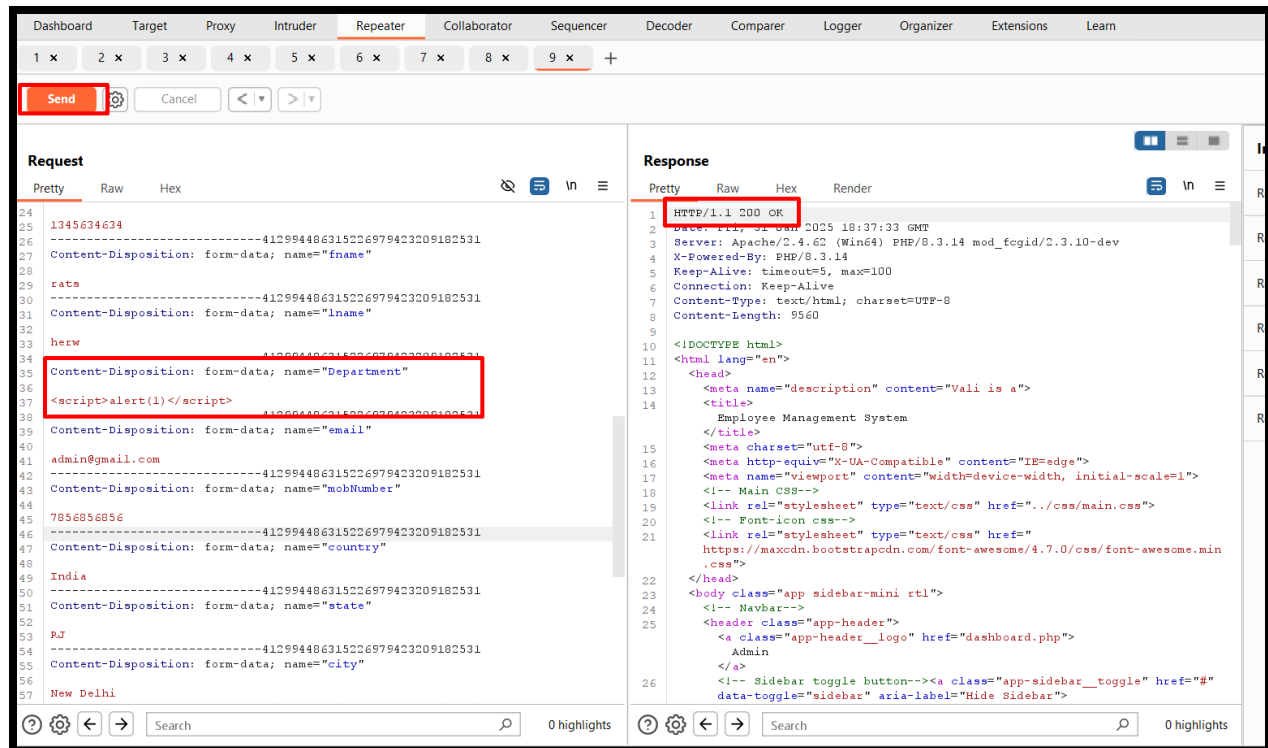
## 5. Modify the Parameter:

- Send the request to the Burp Suite Repeater and modify the “**department**” parameter with the following payload: `<script>alert(1)</script>`



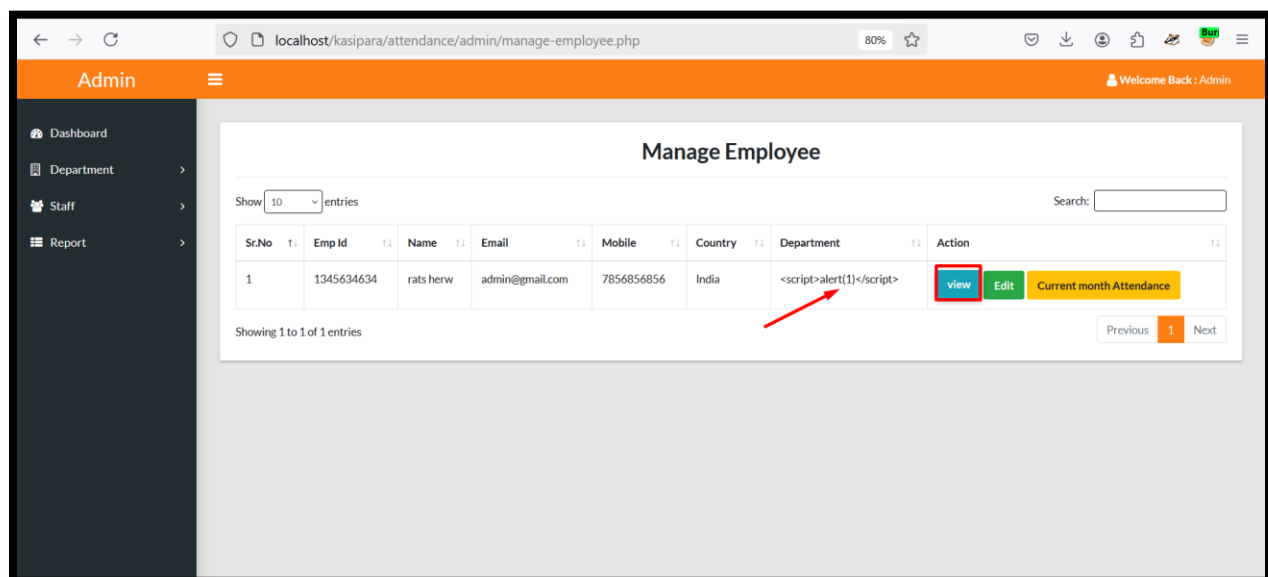
## 6. Send the Modified Request:

- Forward the modified request in the Burp Suite Repeater.
- Observe the delay in the response code.

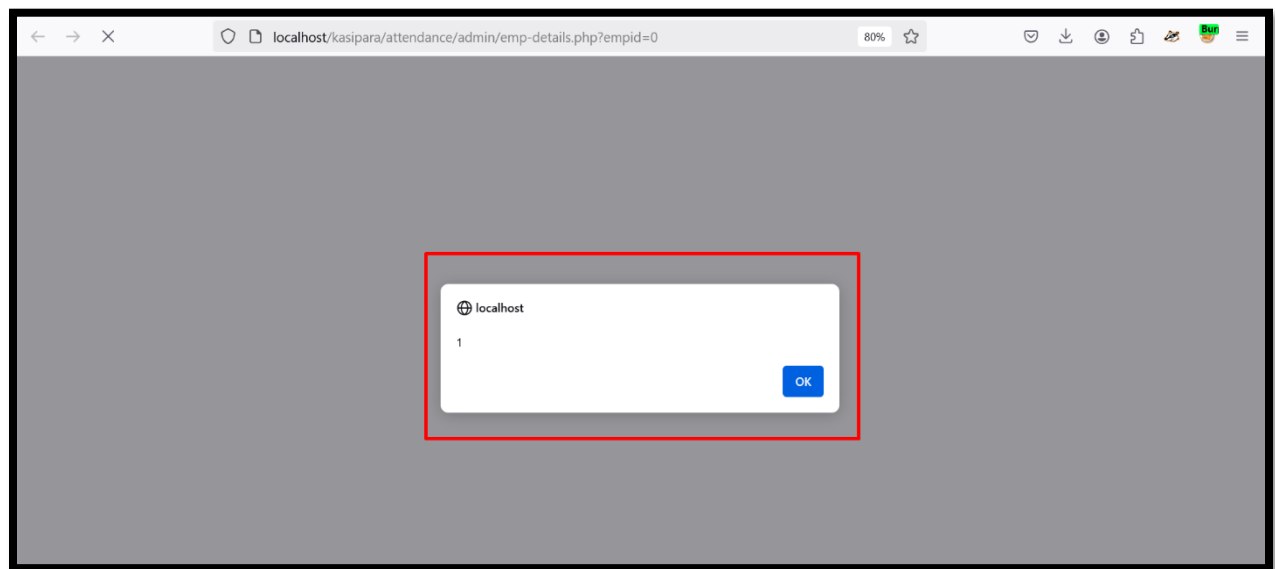


## 7. Trigger the Payload: -

- Navigate to the “ACTION” section in the Manage Employee.
- Here you can see the payload has been stored in the “department” parameter.



- The payload will be executed as soon as the click view button is clicked.
- Observe the execution of the injected JavaScript payload.



## Impact

- Execution of arbitrary JavaScript in the context of the admin's browser.
- Potential to steal admin cookies, hijack sessions, or perform actions on behalf of the admin.

## Recommendations

- Input Validation: Enforce strict validation to block harmful scripts or tags.
- Output Encoding: Encode user inputs before rendering.
- Content Security Policy (CSP): Use a strong CSP to block unauthorized scripts.

## References

OWASP XSS Prevention Cheat Sheet: <https://owasp.org>