

sxc2044-pixel / hajimi Public

<> Code **Issues** 1 Pull requests Actions Projects Security Insights

New issue



code-projects Exam Form Submission Project V1.0 /admin/update_fst.php cross site scripting #1

Open

zzb1388 opened 2 weeks ago



code-projects Exam Form Submission Project V1.0 /admin/update_fst.php cross site scripting

Email OF AFFECTED PRODUCT(S)

. Exam Form Submission

Vendor Homepage

- code-projects

submitter

- Vulnerable File
- hajimi

VERSION(S)

- V1.0

Software Link

<https://code-projects.org/exam-form-submission-in-php-with-source-code/>

Vulnerability Type

- XSS

Root Cause

- An XSS vulnerability was found in the '/admin/update_fst.php' file of the 'Exam Form Submission' project. The reason for this issue is that attackers inject malicious script code from the parameter 'sname' and the system outputs the user input directly to the web page without appropriate encoding or filtering. This allows attackers to execute arbitrary script code in the victim's browser, thereby performing unauthorized operations.

Impact

- Attackers can exploit this XSS vulnerability to steal cookies, session tokens, or other sensitive information of the victim, perform actions on behalf of the victim, deface web pages, redirect users to malicious websites, and even gain control of the victim's browser, posing a serious threat to user privacy and system security.

EmailRIPITION

- During the security review of "Exam Form Submission", I discovered a critical XSS vulnerability in the "/admin/update_fst.php" file. This vulnerability stems from insufficient user input validation and output encoding of the 'sname' parameter, allowing attackers to inject malicious script code. Therefore, attackers can execute arbitrary scripts in the victim's browser, steal sensitive information, and perform operations on behalf of the victim. Immediate remedial measures are needed to ensure system security and protect user data.

No login or authorization is required to exploit this vulnerability

Vulnerability details and POC

Vulnerability location:

- 'sname' parameter

Payload:

```
<script>prompt(/xss/);</script>
```

The following are screenshots of some specific information obtained from testing and running with the relevant tool:

"http://127.0.0.1/EXAM_FORM_SUBMISSION/admin/update_fst.php"



127.0.0.1

/xss/

确定 取消

Suggested repair

1. Output encoding:

Encode user input when outputting it to the web page. Different contexts (such as HTML, JavaScript, CSS, URL) require different encoding methods to ensure that the input is treated as pure text and not executed as code.

2. Input validation and filtering:

Strictly validate and filter user input data. Only allow input that conforms to the expected format and reject or escape any potentially malicious content, such as script tags, event handlers, etc.

3. Use Content Security Policy (CSP):

Implement a strict CSP to restrict the sources of scripts that can be executed on the web page, preventing the execution of unauthorized inline scripts and external scripts.

4. Set secure and HttpOnly flags for cookies:

For sensitive cookies (such as session cookies), set the HttpOnly flag to prevent access via JavaScript, and set the Secure flag to ensure they are only transmitted over HTTPS, reducing the risk of cookie theft.

5. Regular security audits:

Regularly conduct code and system security audits to promptly identify and fix potential XSS vulnerabilities and other security issues.

[Sign up for free](#) to join this conversation on GitHub. Already have an account? [Sign in to comment](#)

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects


Milestone

No milestone

Relationships

None yet

Development

 Code with agent mode

No branches or pull requests

Participants



