NEO LMS & MATRIX LMS Cross-Site Scripting Attack Vectors

Mauro M. <hello@maurom.dev> – 3rd of February 2021

Table of Contents

Introduction

Both NEO LMS and MATRIX LMS are vulnerable to a series of type II cross-site scripting attacks (CWE-79) [1]. These attacks allow an attacker, be it via their account or a compromised one, to execute malicious code on an instructor's or other user's device.

Noted that XSS escaping has been implemented in certain assignment types, namely those where many users are, to complete an assignment, required to load a possibly vulnerable page (for example a debate-type and forum-type assignments). There has been no thought given, however, to protecting an instructor from these types of attacks. An attacker can get an instructors IP address, change their grade or even impede the instructor from seeing their responses (as my proof-of-concept demonstrates).

Testing Environment

Firefox Version 83.0 (64-bit) on Parrot GNU/Linux 4.10 x86_64 (Kernel 5.7.0-2parrot2-amd64)

Methodology

Using a trail of MATRIX LMS and a NEO LMS instance, the following tests where performed:

1- Verifying the existence of XSS vulnerabilities in content submitted by a user in the following assignment types:
   a. Debate
   b. Essay
   c. Forum
   d. Survey
2- Verifying the existence of XSS vulnerabilities in task descriptions
3- Verifying the existence of XSS vulnerabilities in user-submitted content in the user's resources, and subsequently in their portfolio

Global Methodology

- The assignment was created by an administrator, in a course with all users enrolled, the assignment was then submitted, with the payload, by a user without administrative permissions.
- Unless a parameter is specified, it is set to its default
- The following snippets were used to verify for XSS
  `<script>alert(window.location.host)</script>`

  `<scr<script>ipt>alert(window.location.host)</<scr<script>ipt>>`

Debate Assignment Type

- A debate assignment was created and assigned to all students in the group, with the following parameters:
    - Name: “Debate Testing”
    - Proposition: “Testing”
- Using the built-in WYSIWYG text-editor’s code functionality, multiple variations of the snippets where inputted

Essay Assignment Type

- An essay assignment was created and assigned to all students in the group, with the following parameters:
    - Name: “Essay Testing”
    - Instructions: “Testing”
- Using the built-in WYSIWYG text-editor’s code functionality, multiple variations of the snippets where inputted

Forum Assignment Type

- An essay assignment was created and assigned to all students in the group, with the following parameters:
    - Name: “Essay Testing”
    - Instructions: “Testing”
- Using the built-in WYSIWYG text-editor’s code functionality, multiple variations of the snippets where inputted

Survey Assignment Type

- A survey assignment was created and assigned to all students in the group, with the following parameters:
    - Name: “Survey Testing”
    - Instructions: “Testing”
- A singular question bank was created, named “Survey Testing”, it contains a singular freeform question named “Freeform test” which is required
- Using the text box, multiple variations the snippets where inputted, additionally, the PoC was inputted, minified

Task Descriptions

- Any sort of assignment was created, using the built-in WYSIWYG text-editor's code functionality multiple variations of the snippets where inputted

User Resources & Portfolio

- A user created a page-type resource in their locker
- Using the built-in WYSIWYG text-editor's code functionality multiple variations of the following HTML was inputted

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>alert(window.location.host)</script>

</body>

</html>
```

- In NEO LMS, a user can add this resource to their portfolio

Proof-of-concept

The following proof-of-concept is intended to show a non-destructive but application-specific payload that can be executed via the attack vectors described herein

```
window.onload = function () {
    student = document.getElementById('selected_student')
    children = student.children
    for (var i = 0; i < children.length; i++) {
        let child = children[i];
        if (child.hasAttribute('selected')) {
            child.removeAttribute('selected')
            children[0].setAttribute('selected', 'selected')
        }
    }
    change_student_to_grade(student, 'student');

}
```

It intends to impede the instructor from correcting the attacker's response by redirecting the instructor’s correction page to the first student in the list of students to which the assignment was attributed.

This is only one of the many payloads, others may include, but are not limited to:

- Altering result displays
- Altering question responses after-the-fact

These are not persistent and would not show up in exports but valid concerns, nonetheless.

Results

Component Tested	Is XSS possible?	Affected User Group	CVSS Attributed
Debate Assignments	No	None	None
Essay Assignments	No	None	None
Forum Assignments	No	None	None
Survey Assignments	Yes	Instructors	5.4 (Medium)
Task Descriptions	Yes	Users, Instructors	4.8 (Medium)
User Portfolio	Yes	Users, Instructors	5.4 (Medium)

Table I: Summary of results obtained

Regarding debate, essay and forum assignments, there is HTML sanitation, the characters <> are automatically replaced and the <script> tag is automatically deleted, all my attempts to circumvent the filter were not successful.

Regarding survey assignments there appears to be no HTML sanitation on freeform questions. Payloads can simply be added between script tags and will be executed when the instructor goes to Assessments > {Survey Assessment Name} > Scores and clicks on the attacker's result (paper icon)

Vector String:
`CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:H/A:N/E:H/RL:U/RC:C`

Regarding task descriptions there, again, appears to be no XSS protection. This means code would be executed every time a student where to click on the assignment to complete it.

Vector String:
`CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:N/E:H/RL:U/RC:C`

Finally, regarding user resources there is, again, no HTML sanitation, the situation worsens on NEO LMS because of the user's ability to add the document to their portfolio. This makes those who visit the user's portfolio susceptible to the attack because the HTML is automatically loaded even without the user clicking on the resource per-say, this behaviour is also seen when the user navigates to their locker.

Recommendations

Most of the issues can be remedied by using sandboxed iframes[2] and enforcing correct CORS headers[3]. Always following the principle of least privilege[4] Additionally, regarding the user resources, I would recommend only rendering the resource when the user clicks on it, to either edit, or view it.

References

1 - CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting'). (2006, July 9). Retrieved February 03, 2021, from https://cwe.mitre.org/data/definitions/79.html

2 - HTML Specification - iframe. (2021, February 02). Retrieved February 03, 2021, from https://html.spec.whatwg.org/multipage/iframe-embed-object.html#attr-iframe-sandbox

3 - Cross-Origin Resource Sharing (CORS). (2021, January 25). Retrieved February 03, 2021, from https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

4 - Principle of least privilege. (2021, January 20). Retrieved February 03, 2021, from https://en.wikipedia.org/wiki/Principle_of_least_privilege

Additional information and declarations

Competing interests

There are no competing interests

Acknowledgments

- Noah van der Aa <ndvdaa@gmail.com>