NEO LMS & MATRIX LMS quiz-type assignment vulnerabilities

Mauro M. <hello@maurom.dev> – 7th of February 2021

Table of contents

Introduction

Both NEO LMS and MATRIX LMS's test-type assignments can be used as a vector for a type II cross-site scripting attack on an instructor's client (CWE-79)[1], furthermore, there are no limitations to disabling or altering the timer on quiz-type assignments, the user can run client-side code to disable the timer on their session with no objection from the server.

The following white paper intends to explain and provide a proof-of-concept for both possible attacks, one of them being a security vulnerability and another being a disabling of a major function of the quiz-type assignment, allowing for unfair evaluation and diminishing instructor's trust in the platform.

Testing Environment

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

BURP Proxy Suite, Community Edition, Build 5278

Methodology

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

1- Verifying the existence of XSS vulnerabilities in content submitted by the user in the quiz assignment type, verifying every type of question
2- Verifying if the timer could be disabled or if the time kept by it could be altered

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>>
  ```

- A BURP proxy suite, set up with default settings was utilized to intercept the requests and responses

Quiz Assignment Used

- A quiz assignment was created and assigned to all students in the group, with the following parameters:
    - Name: “Testing”
    - Proposition: “Testing”
- A singular question bank was created, named “Testing”, it contains the following (format: question type – question value: possible answers):
    - A true or false question, with the name “True or False Testing”
    - A multiple-choice question (with one correct answer) where the responses are the numbers 1 to 12, the first answer (1) is correct
    - A multiple-choice question (with multiple correct answers) where the possible answers are the numbers 1 to 12 and 4,5 and 7 are the correct answers”
    - A “fill in the blanks” question where the question is “Fill in the blanks testing BLANK BLANK” and the correct answers are “Hello” and “World” respectively
    - A freeform question where the question is “Freeform testing”
    - A matching question where the question is “Matching question testing” and the correct answer key is:
        - 1 => 1
        - 2 => 1
        - 3 => 1
        - 4 => 1
        - 5 => 1
        - 6 => 1
        - 7 => 1
        - 8 => 1
        - 9 => 1
        - 10 => 1

- An arithmetic question where the operators are addition, the operands are 2 numbers ranging from 0 to 9 and answers can be zero or positive

Proof-of-concept

The Firefox add-on intercepts the response sent by the web server to any URL matching `https://*/student_take_quiz_assignment/display/*` hence affecting any quiz-type assignments set in MATRIX LMS or NEO LMS.

Having intercepted the response, it proceeds to iterate through all script tags in the response, I avoided hard coding the index in case there are custom HTML headers set which may, or may not, include script tags. Once it finds the script tag that contains the call to `init_quiz()` it converts the arguments into a map, this allows for the addon to be customised to change other parameters and not just the `total_seconds` one.

In this case, it sets the `total_seconds` parameter to null, this effectively disables the timer, allowing the student to remain as much time as he or she may wish on the quiz.

It then proceeds to replace the generated function call with the one created by the addon and sends the modified response to the user.

It is noteworthy that we can alter any other parameter passed into the function, including `show_score` and `instant_feedback`. This add-on can also be very quickly adapted to run on chromium-based browsers, as the API is very similar to that of Firefox, allowing the add-on to be run on approx. 90% of browsers

Results

Component Tested	Vulnerabilities Found	CVSS Attributed
Quiz Assignment	XSS in arithmetic question type	5.4 (Medium)
	Possibility to disable timer or alter time	N/A

Regarding all the question types, except arithmetic, there was proper HTML sanitation, the characters <> are automatically replaced and the <script> tag is automatically deleted, even when injecting the payload into the request.

Regarding the arithmetic question type, there is no input sanitation, as such, it is possible to simply input a payload into the text box.

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 the timer, it is trivially simple to disable it, by intercepting the server's response, the values passed to the `init_quiz()` function can be changed. By passing a null value as the `total_seconds` parameter it is possible to disable the timer completely. Since there is no check server-side the quiz responses are accepted.

Recommendations

Regarding the XSS vulnerability in the arithmetic question type, a regular expression should be applied to validate the input, for example `^-?\d+$`, this could be used, both on the client side, using the input tag's pattern atrribute[3] and on the server side, using ruby's build in regex[4].

Regarding the timer server-side checking should be implemented, as in, upon receiving a submission, the server compares the time elapsed with the time allowed to complete the assignment, rejecting the submission if the time taken surpasses the time allotted. This should be a relatively simple implementation given that the server already stores an amount of time allowed, which is originally passed as the `total_seconds` parameter.

Author's Notes

I would like to emphasise the need to not overlook the possibility to disable the timer, even though it does not constitute a security vulnerability within itself (even though type 1 cross-site scripting is possible by altering the `do_you_want_to_leave_text` and the `cannot_connect_to_server_text`) it poses a major risk to the viability of using the NEO LMS platform as an online evaluation tool during these times where so many students are learning from home. The proof-of-concept attempted to demonstrate how widely this vulnerability could exploited used by a user with no knowledge of how to run a man-in-the-middle attack[5], it is as simple as installing a browser add-on.

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 - Global desktop browser market share for 2021. (2020, November 03). Retrieved February 07, 2021, from https://kinsta.com/browser-market-share/

3 - HTML attribute: Pattern. (2020, September 15). Retrieved February 07, 2021, from https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/pattern

4 - Class regexp. (n.d.). Retrieved February 07, 2021, from https://docs.ruby-lang.org/en/3.0.0/Regexp.html

5 - CWE-345: Insufficient Verification of Data Authenticity. (2006, July 19). Retrieved February 07, 2021, from https://cwe.mitre.org/data/definitions/345.html

Additional information and declarations

Competing interests

There are no competing interests

Acknowledgments

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