Moodle Cross Site Scripting / Server-Side Request Forgery

Authored by rekter0 | Site r0.haxors.org

Posted Nov 9, 2021

Moodle versions 3.10 to 3.10.1, 3.9 to 3.9.4, 3.8 to 3.8.7, and 3.5 to 3.5.16 suffer from cross site scripting and server-side request forgery vulnerabilities.

tags | advisory, vulnerability, xss
advisories | CVE-2021-20280
SHA-256 | 5ebbb3e3b937891a7993ff7cfa746f4eb1c07b7273456d6b43b919d3917226a0 Download | Favorite | View

Related Files

Share This

Like Twee LinkedIn Reddit Digg StumbleUpon

Change Mirror Download

```
22-10-2021 - rekter0

Moodle is an opensource learning management system, popular in universities and workplaces largely used to
manage courses, activities and learning content, with about 200 million users


Versions affected   3.10 to 3.10.1, 3.9 to 3.9.4, 3.8 to 3.8.7, 3.5 to 3.5.16
CVE identifier   CVE-2021-20280

# Summary

When managing a course in Moodle, it's possible to add a 'Feedback' activity. This activity-type allows
enrolled students to provide feedback to different questions created by the teacher. Some of these question
types allow the students to provide text-input as feedback (eg. 'Short text answer'). The input provided has
HTML striped before being inserted into the database and is supposedly sanitized in a safe way before being
rendered, during this process, and for unkown reasons to me, moodle did html entities decoding leading to a
stored XSS vulnerability and Blind SSRF.

# Vulnerability analysis

When a student submits their feedback text answer it is processed with s() function before being stored in the
database

/mod/feedback/classes/completion.php

>    $itemobj = feedback_get_item_class($item->typ);
>    $newvalue['value'] = $itemobj->create_value($data->$keyname);

    // Update or insert the value in the 'feedback_valuetmp' table.
    if (array_key_exists($item->id, $existingvalues)) {
        $newvalue['id'] = $existingvalues[$item->id];
        $DB->update_record('feedback_valuetmp', $newvalue);
    } else {
        $DB->insert_record('feedback_valuetmp', $newvalue);
    }

feedback_get_item_class loads class processor for that feedback input type

/mod/feedback/item/textfield/lib.php

    public function create_value($value) {
>       return s($value);
    }

create_value() process input with s() function

/lib/weblib.php

/**
 * Add quotes to HTML characters.
 *
 * Returns $var with HTML characters (like "<", ">", etc.) properly quoted.
 * Related function {@link p()} simply prints the output of this function.
 *
 * @param string $var the string potentially containing HTML characters
 * @return string
 */
function s($var) {

    if ($var === false) {
        return '0';
    }

>   return preg_replace('/&#(\d+|x[0-9a-f]+);/i', '&#$1;',
>           htmlspecialchars($var, ENT_QUOTES | ENT_HTML401 | ENT_SUBSTITUTE));
}

As in function description, it removes tags and process the input with htmlspecialchars

Stored XSS
When rendering the answer entry, mid of the process, moodle used to do html_entity_decode

/mod/feedback/classes/response_table.php

    public function other_cols($column, $row) {
        if (preg_match('/^val(\d+)$/', $column, $matches)) {
            $items = $this->feedbackstructure->get_items();
            $itemobj = feedback_get_item_class($items[$matches[1]]->typ);
            $printval = $itemobj->get_printval($items[$matches[1]], (object) ['value' => $row->$column]);
            if ($this->is_downloading()) {
>                $printval = html_entity_decode($printval, ENT_QUOTES);
            }
            return trim($printval);
        }
        return $row->$column;
    }

So, if a user supplied a payload with hex-encoded values, e.g. '&#x3c ;' instead of '<' it would have remained
the same after s() have had processed it. this would have gone under the radar of the sanitizer, and moodle
would have decoded it during rendering process. The stored XSS could have been leveraged to trigger a blind
SSRF.
# Impact

An authenticated attacker with the least privilege (student), could inject html/js with a crafted response to
feedback activity leading to a stored XSS and blind SSRF. Successful exploitation of the XSS vulnerability
allows the attacker to takeover moodle users including teachers and administrators or perform actions on their
behalf. Exploiting the Blind SSRF would have given the attacker the ability to interact with internal server
services and possible RCE in some environement setups.

# Timeline

12-01-2021 - Reported
01-02-2021 - Vendor confirmed
15-03-2021 - Fixed in new release
```


File Archive: December 2022 <

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Top Authors In Last 30 Days

- Red Hat 180 files
- Ubuntu 78 files
- Debian 24 files
- LiquidWorm 23 files
- malvuln 12 files
- nu11secur1ty 10 files
- Gentoo 9 files
- Google Security Research 8 files
- T. Weber 4 files
- Julien Ahrens 4 files

File Tags

ActiveX (932)
Advisory (79,733)
Arbitrary (15,694)
BBS (2,859)
Bypass (1,619)
CGI (1,018)
Code Execution (6,924)
Conference (673)
Cracker (840)
CSRF (3,290)
DoS (22,601)
Encryption (2,349)
Exploit (50,358)
File Inclusion (4,165)
File Upload (946)
Firewall (821)
Info Disclosure (2,660)
Intrusion Detection (867)
Java (2,899)
JavaScript (820)
Kernel (6,290)
Local (14,201)
Magazine (586)
Overflow (12,418)
Perl (1,418)
PHP (5,093)
Proof of Concept (2,291)
Protocol (3,435)
Python (1,467)
Remote (30,043)
Root (3,504)
Ruby (594)
Scanner (1,631)
Security Tool (7,776)
Shell (3,103)
Shellcode (1,204)
Sniffer (886)

File Archives

December 2022
November 2022
October 2022
September 2022
August 2022
July 2022
June 2022
May 2022
April 2022
March 2022
February 2022
January 2022
Older

Systems

AIX (426)
Apple (1,926)
BSD (370)
CentOS (55)
Cisco (1,917)
Debian (6,634)
Fedora (1,690)
FreeBSD (1,242)
Gentoo (4,272)
HPUX (878)
iOS (330)
iPhone (108)
IRIX (220)
Juniper (67)
Linux (44,294)
Mac OS X (684)
Mandriva (3,105)
NetBSD (255)
OpenBSD (479)
RedHat (12,448)
Slackware (941)
Solaris (1,607)

Spoof (2,166)
SQL Injection (16,101)
TCP (2,379)
Trojan (686)
UDP (876)
Virus (662)
Vulnerability (31,132)
Web (9,357)
Whitepaper (3,729)
x86 (946)
XSS (17,494)
Other

SUSE (1,444)
Ubuntu (8,199)
UNIX (9,158)
UnixWare (185)
Windows (6,511)
Other

packet storm

Site Links

News by Month
News Tags
Files by Month
File Tags
File Directory

About Us

History & Purpose
Contact Information
Terms of Service
Privacy Statement
Copyright Information

Hosting By

Rokasec

Follow us on Twitter

Subscribe to an RSS Feed