0-DAY NO CVE MEDIUM PUBLIC

Moodle, Blackboard img src file read

Moodle 5.2.3 Blackboard question import still joins img src onto filebase with no PARAM_PATH. A teacher zip copies config.php into the draft area. MSA-26-0012 jailed database-activity import. This join was not in that set. No CVE yet.

NAME Moodle, Blackboard img src file read	TYPE 0-day analysis
CVE n/a	CVE RISK medium
DISCLOSURE STATUS public	VENDOR Moodle
AFFECTED Moodle LMS 5.2.3 tag v5.2.3 (344232c). Teacher question-bank / lesson Blackboard import. Students cannot import. Gadget lab, not a full PHP app.	PUBLISHED 12 Oct 2026
UPDATED 12 Oct 2026	TAGS 0-day, moodle, blackboard, file-read, cwe-22, cwe-73, authenticated

TL;DR

Lab files and proof-of-concept here.

01 the image was config.php

I am @abraxas_null. The proof of concept is on GitHub: **abraxas/moodle-blackboard-img** (loopback client). The lab stack is `lab/`: Dockerfile, docker-compose.yml, run.sh, poc.py, gadget.php. Authorized lab only. It talks to loopback.

This is **Moodle LMS 5.2.3** (`344232c15336c71b80f9aca8359ce0e0a9f3d116`). No CVE yet. CWE-22 / CWE-73. 6.5 Medium (`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N`). Remote file read. Teacher. Not RCE. Not LPE.

MSA-26-0012 taught database-activity import to jail paths with `PARAM_FILE`. Blackboard question import still treats `img src` as a path, concatenates it onto `$this→filebase`, and copies whatever the PHP user can open into the draft filearea. The "image" is the file. Typical prize is `config.php`.

Same pin: enrol_database SQLi, grades feedback IDOR, hidden grade items.

02 Why Moodle

I had just finished leftover passes on n8n, Gitea, and Signal-Server. Moodle is a large PHP tree that still ships question-format importers from a decade of LMS mergers. Teachers import zips. Those zips have HTML. HTML has `img src`. That is a file API wearing a quiz costume.

I only wanted Critical or High, with Medium if it leaked something a student or teacher should not see. Unauthenticated RCE would have been the headline. 5.2.3 did not give me one. It gave me four leftovers of MSA-26 patches that stopped one file too soon.

Moodle's own intake is the security form. There is no bounty on this path. This post is the public pack.

03 Pinning v5.2.3

Latest stable. Tag `v5.2.3`, commit `344232c`, `$release = '5.2.3 (Build: 20260914)'`. Code under `public/`. MSA-26-0042/0043 are in this tag, which is why I sat on it instead of an older minor that still had those as findings.

GitHub releases and the Moodle download page agreed. I did not take an RC.

04 The skip list is half the hunt

Before you look for leftovers, you write down what is already public so you do not "discover" an MSA with a press release. On 5.2.3 the interesting ones were already closed:

- MSA-26-0001 restore-files RCE
- MSA-26-0002 TeX admin RCE
- MSA-26-0005 `auth_db` SQLi (parameterized; the enrol sibling is a different post)
- MSA-26-0006 Google Drive RCE
- MSA-26-0012 database-activity import file read (`PARAM_FILE`)
- MSA-26-0013 email MFA
- MSA-26-0014 backup restore file read (contenthash allowlist)
- MSA-26-0016 grade WS group checks (other services; feedback is a different post)

- MSA-26-0030 URL downloader SSRF (IPv4-mapped unwrap)
- MSA-26-0031 question bank WS SQLi
- MSA-26-0033 report builder audience class
- MSA-26-0034 forum post template XSS
- MSA-26-0041 XML grade CSRF
- MSA-26-0042 profile availability SQLi
- MSA-26-0043 login notification bypass

MSA-22-0016 cleaned Blackboard `baseurl`. It did not jail `img src`.

05 Surfaces that returned nothing

I split the tree: SQLi in core `$DB`, backup/unserialize, XSS/CSRF, web services/LTI, SSRF/pluginfile, auth/MFA. Most of that is Moodle doing what the MSA list said it would do.

Restore still wants a contenthash. URL downloader unwraps mapped IPv4. Question WS is parameterized. Forum templates are escaped. Profile availability is a whitelist. Email MFA does not skip on this tag.

XSS leftovers of MSA-26-0034/0039 and the SSRF leftover of MSA-26-0030 did not clear the bar. I left them.

Then I opened Blackboard because `get_filecontent()` already knew about `PARAM_PATH` and `text_field()` still did a raw join. Incomplete sibling **inside** the same format.

06 The join

`formatbase.php` `text_field`:

PHP Copy

```php
preg_match_all('|<img[^>]+src="([^"]*)"|i', $text, $out);
foreach ($out[1] as $path) {
    $fullpath = $this->filebase . '/' . $path;

    if (is_readable($fullpath) && !in_array($path, $filepaths)) {
        $dirpath = dirname($path);
        $filename = basename($path);
        $newfilename = $this->store_file_for_text_field($data, $this->filebase, $
```

`store_file_for_text_field` then:

PHP Copy

```php
$fs->create_file_from_pathname($filerecord, $tempdir . '/' . $filepathinsidetempd
```

`$filepathinsidetempdir` is `dirname($path)`. If `$path` is `../../../../html/config.php`, that dirname is outside the extract tree. `is_readable` is the only gate. The PHP user can read `config.php`. The draft "image" is those bytes.

Same-family `format.php` `get_filecontent()` already jails:

PHP Copy

```php
$fullpath = $this->tempdir . '/' . clean_param($path, PARAM_PATH);
```

They had the helper. They used it on one path. Not this one.

07 The lab

I did not boot a full Moodle PHP app for this leftover. The pin functions are the argument. The gadget copies `text_field` and `store_file_for_text_field` from v5.2.3, plants a zip with `<img src="../../../../html/config.php">`, and serves drafts on loopback `:19560`. `how=gadget` on the SUCCESS line.

Teacher import matched `config.php` byte-for-byte. Draft contained `MOODLE-BLACKBOARD-IMG-WITNESS`. GET `/draft` returned 298 bytes, sha256 equal to the planted config. Student import was 403 `teacher-only`. The same zip with `PARAM_PATH` jail stayed under filebase and did not copy config.

PLAIN TEXT Copy

```
SUCCESS moodle-blackboard-img how=gadget pin=v5.2.3 teacher-import=yes draft-matc
```

PLAIN TEXT Copy

```
python3 moodle-blackboard-img-Abraxas-Labs.py
```

or `cd lab && ./run.sh`. Target only `http://127.0.0.1:19560`. Compose project `moodle-blackboard-img`. Stack tears down after the run.

08 Wrong turns, in the order I made them

Treating student import as the leak. Students cannot import Blackboard. 403 is the control, not the finding.

Calling this RCE because the copied file is PHP. The draft stores bytes. It does not `include` them. Not RCE unless someone chains a second leftover.

Assuming MSA-22-0016 had finished the job. That advisory cleaned `baseurl`. `img src` is a different string.

Assuming MSA-26-0012 had finished the job. Database-activity import got `PARAM_FILE`. Blackboard `text_field` was not in that patch.

Booting a full Moodle just to copy a file. The join is the leftover. A gadget of the pin functions is the honest lab. `how=gadget` so nobody can pretend I stood up question-bank UI.

A reverse shell. Theatre. The oracle is draft bytes matching `config.php` plus the witness.

09 What an attacker can do

Writer: a teacher who can import a Blackboard zip into the question bank (or lesson). Typical prize is `config.php` (DB password) as the PHP user.

Class: File read. **Reach:** Remote.

PRIVILEGE	METHOD / SINK	RI IN AV
Read a file the PHP user can open, via `../` in `img src`	`text_field` joins `$this->filebase . '/' . $path`; `store_file_for_text_field` copies into the draft area	R b
Download that draft as the "image"	draft itemid + rewritten `@@PLUGINFILE@@` filename	R
Anyone who can later open the imported question	Same draft / pluginfile	R
Student import	same zip, role=student	n

PRIVILEGE	METHOD / SINK	RI II AV
Same path through `clean_param($path, PARAM_PATH)`	gadget `jail=1`	n
PHP RCE / OS LPE	n/a	n

Default Moodle without Blackboard import is not this bug.

10 The fix

`clean_param($path, PARAM_PATH)` on `img src`, then `realpath` must stay under `realpath($this→filebase)`. Same shape as `get_filecontent()`. Add a Behat case: teacher zip with `../` in `img src` must not copy a file from outside the extract tree.

I am not going to print a zip you can paste at someone else's question bank. The missing `PARAM_PATH` is the useful part. If you own the site, run the proof of concept against loopback.

11 References

- Proof of concept: abraxas/moodle-blackboard-img · moodle-blackboard-img-Abraxas-Labs.py
- Lab: `lab/` · Dockerfile · docker-compose.yml · run.sh · poc.py · gadget.php

- @abraxas_null · github.com/abraxas · abraxaslabs.tech · abraxas.null@proton.me
- CWE-22 · CWE-73
- Tree: moodle/moodle v5.2.3 (`344232c`) · `formatbase.php` · `format.php`
- Same pin: moodle-enrol-database-sqli · moodle-grades-feedback-idor · moodle-grades-gradeitems

Abraxas Labs

Independent security research by abraxas.

Analysis of 0-days, n-days, malware, and the leftovers that still matter.

GitHub **@abraxas_null** **abraxas.null@proton.me** **abraxaslabs.tech**

Independent research. Private analysis by arrangement. No ads. abraxas.null@proton.me · For defensive awareness and authorized testing only.

© 2026 Abraxas Labs