0-DAY NO CVE MEDIUM PUBLIC

Moodle, core_grades_get_feedback group IDOR

Moodle 5.2.3 core_grades_get_feedback still only checks grader:view. Other grade web services gained separate-groups checks in MSA-26-0016. A non-editing teacher in Group A AJAX-reads Group B feedback and email. No CVE yet.

NAME Moodle, core_grades_get_feedback group IDOR	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). Non-editing teacher with gradereport/grader:view, without accessallgroups. Course in Separate groups. Live install lab.	PUBLISHED 12 Oct 2026
UPDATED 12 Oct 2026	TAGS 0-day, moodle, idor, grades, cwe-863, cwe-200, authenticated

TL;DR

Lab files and proof-of-concept **here**.

01 the grader UI learned groups. this AJAX service did not

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

This is **Moodle LMS 5.2.3** (`344232c15336c71b80f9aca8359ce0e0a9f3d116`). No CVE yet. CWE-863 / CWE-200. 4.3 Medium (`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N`). Remote IDOR. Non-editing teacher. Separate groups. Not a score write. Not RCE.

MSA-26-0016 / MDL-88667 put `groups_user_groups_visible` / `groups_group_visible` on other grade web services. `core_grades_get_feedback` still only calls `require_capability('gradereport/grader:view')` . The JSON comes back with another student's `feedbacktext` , `fullname` , `picture` , and the first identity field (email on a default profile).

Same pin: Blackboard img file read, enrol_database SQLi, hidden grade items.

02 Why grade web services

Moodle 5 still has two graders: the report UI, and a pile of AJAX methods behind `lib/ajax/service.php` . MSA-26-0016 was a group-leak pass. Incomplete siblings of a group-leak pass are where I look next.

I only wanted Critical or High, with Medium if a teacher without access-all-groups could read another group's work. This is that Medium. Userids are sequential ints. The service is AJAX-only.

03 Pinning v5.2.3

Same tag as the other three Moodle leftovers. MSA-26-0016 is in the tree. `get_gradable_users.php` already throws `cannotaccessgroup`. `get_feedback.php` does not call that helper. That is the leftover.

04 The skip list

Do not re-file the grade WS that MDL-88667 already patched. Do not re-file MSA-26-0041 XML grade CSRF. Do not re-file MSA-26-0031 question-bank SQLi. The MSA-26 list in the Blackboard post is the skip list for this hunt too.

05 Surfaces that returned nothing

`gradereport_user` external uses `groups_user_groups_visible($course, $user->id)` and throws `notingroup`. `core_grades_get_gradable_users` checks `groups_group_visible`. Those are the controls. I wanted a service in the same folder that skipped both.

`get_feedback.php` was that service.

Student-as-caller is `nopermissions`. That is not the bug. The bug is a teacher who is not supposed to see Group B.

06 The handler

`public/grade/classes/external/get_feedback.php`:

PHP Copy

```
$context = \context_course::instance($courseid);
parent::validate_context($context);

require_capability('gradereport/grader:view', $context);

$gtree = new \grade_tree($params['courseid'], false, false, null, !$CFG->enablec
$gradeitem = $gtree->get_item($params['itemid']);

if ($gradeitem === false) {
    throw new invalid_parameter_exception('Course ID and item ID mismatch');
}

$grade = $gradeitem->get_grade($params['userid'], false);
$user = \core_user::get_user($params['userid']);
$extrafields = \core_user\fields::get_identity_fields($context);

return [
    'feedbacktext' => $grade->feedback,
    'title' => $gradeitem->get_name(true),
    'fullname' => fullname($user),
    'picture' => $OUTPUT->user_picture($user, ['size' => 50, 'link' => false]),
    'additionalfield' => empty($extrafields) ? '' : $user->{$extrafields[0]},
];
```

No `groups_user_groups_visible`. No enrolment check on the target user. Capability is the only gate. A non-editing teacher has `gradereport/grader:view`. They do not have `moodle/site:accessallgroups` unless someone granted it.

Sibling `get_gradable_users.php`:

PHP Copy

```
if ($params['groupid'] && !groups_group_visible($params['groupid'], $course)) {
    throw new \moodle_exception('cannotaccessgroup', 'core_grades');
}
```

That is the check this service does not make.

07 The lab

Live Moodle **v5.2.3**. php:8.3-cli + MariaDB 11.4. Compose project `moodle-grades-feedback-idor`. Loopback `:19580`. `how=live`.

Seed builds a Separate-groups course (`groupmode=1`), teacher Alpha in Group A only, student Bravo in Group B only, CAP_PREVENT on `moodle/site:accessallgroups`, and a manual grade item whose feedback is the witness. Compose bind-mounts a Moodle v5.2.3 tree as `/opt/moodle-pin`. The Moodle tree is not in the pack.

Teacher call for Bravo: HTTP 200, `error=false`, feedback `MOODLE-GRADES-FEEDBACK-IDOR-WITNESS`, `additionalfield=studentb@example.com`, `fullname=Student Bravo`. Same-group call for Alpha: 200, allowed. Sibling `core_grades_get_gradable_users` for Group B: `cannotaccessgroup`. Student call: `nopermissions`.

PLAIN TEXT Copy

```
SUCCESS moodle-grades-feedback-idor how=live pin=v5.2.3 teacher-role=teacher acce
```

PLAIN TEXT Copy

```
python3 moodle-grades-feedback-idor-Abraxas-Labs.py
```

or `cd lab && ./run.sh`. Target only `http://127.0.0.1:19580`. Stack tears down after the run.

08 Wrong turns, in the order I made them

Treating the sibling miss as the whole story. `get_gradable_users` already throws `cannotaccessgroup`. That is the control. The leftover is the service that does not call

it.

Calling this a score write because the payload names `itemid`. The return is feedback text and profile fields. No grade update.

Student-as-attacker. `nopermissions`. The writer is a non-editing teacher.

Forgetting CAP_PREVENT on accessallgroups. Default teacher in some installs can already see every group. The leftover is the teacher who cannot, except through this AJAX method.

A reverse shell. Theatre. The oracle is Bravo's feedback string plus email in a teacher-A session.

09 What an attacker can do

Writer: a non-editing teacher enrolled in a Separate-groups course, in Group A only, with grader view and without access-all-groups. They pick another enrolled student's `userid` and call the AJAX service.

Class: IDOR. **Reach:** Remote.

PRIVILEGE	METHOD / SINK
Read another group's grade feedback	POST `/lib/ajax/service.php?sesskey=` `core_grades_get_feedback` `{courseid, userid, itemid}`
Read that student's identity field and display name	Same response `additionalfield` / `fullname` / `picture`
Walk sequential `userid` values in the same course	Same method, guessed ids

PRIVILEGE	METHOD / SINK
List Group B through the sibling selector	`core_grades_get_gradable_users {courseid, groupid=B}`
Student calls the same feedback service	student role, same args
Same-group feedback (Group A student)	teacher call for Student Alpha
PHP RCE / OS LPE / grade write	n/a

Default Moodle without Separate groups, or a teacher who already has `moodle/site:accessallgroups`, is not this leftover.

10 The fix

After `validate_context` / `require_capability`, load the course and the target user, then refuse unless `groups_user_groups_visible($course, $userid)` and the target is enrolled. Same shape as other grade WS after MDL-88667. Add a Behat/PHPUnit case: Separate groups, teacher without accessallgroups, `core_grades_get_feedback` for a userid in another group must fail.

I am not going to print an AJAX POST you can paste at someone else's `/lib/ajax/service.php`. The missing group check is the useful part. If you own the site, run the proof of concept against loopback.

11 References

- Proof of concept: abraxas/moodle-grades-feedback-idor · moodle-grades-feedback-idor-Abraxas-Labs.py
- Lab: `lab/` · Dockerfile · docker-compose.yml · run.sh · poc.py · seed.php
- @abraxas_null · github.com/abraxas · abraxaslabs.tech · abraxas.null@proton.me
- CWE-863 · CWE-200
- Tree: moodle/moodle v5.2.3 (`344232c`) · `get_feedback.php` · `get_gradable_users.php`
- Nearby patched (not this service): MSA-26-0016 / MDL-88667
- Same pin: moodle-blackboard-img · moodle-enrol-database-sqli · 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