0-DAY NO CVE HIGH PUBLIC

phpMyAdmin CSRF token strip misses PSR-7 body on /sql

phpMyAdmin 5.2.3 builds the PSR-7 request before the CSRF check. On mismatch, $_POST is cleared. sql_query was already copied from the snapshot. Non-AJAX POST /sql still runs. Labbed. No CVE yet.

NAME phpMyAdmin CSRF token strip misses PSR-7 body on /sql	TYPE 0-day analysis
CVE n/a	CVE RISK high
DISCLOSURE STATUS public	VENDOR phpMyAdmin
AFFECTED phpMyAdmin 5.2.3 (962857e). Default CookieSameSite=Strict blocks many cross-site browsers. Same-site / Lax / None / HTTP-auth remain.	PUBLISHED 12 Oct 2026
UPDATED 12 Oct 2026	TAGS 0-day, phpmyadmin, csrf, cwe-352, sql

TL;DR

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

01 The token strip missed the request object

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

This is **phpMyAdmin 5.2.3** (`962857e`). No CVE yet. CWE-352. 8.1 High (`CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:N`). Remote CSRF. Victim phpMyAdmin session cookie (or HTTP auth). No valid CSRF token.

`Common.php` builds the PSR-7 request from globals **before** `checkTokenRequestParam`. On mismatch, `Sanitize::removeRequestVars` clears `$_POST` and `$_GET` but not that snapshot. `$sql_query` is then copied from `$request->getParsedBodyParam('sql_query')`. AJAX POSTs abort on token mismatch. Non-AJAX POSTs continue. `SqlController` prefers stripped `$_POST['sql_query']` and otherwise keeps the global.

Default `CookieSameSite=Strict` in 5.2.3 blocks many cross-site browsers. Same-site, Lax, None, or HTTP-auth still hit it. This is a broken CSRF control either way.

02 Why CSRF after Strict

Same wave as WebAuthn. Routes/CSRF was the hunt. A CSRF token that empties `$_POST` and then copies `sql_query` from a snapshot taken before the empty is not a CSRF token. Strict SameSite is a cookie attribute. It is not a substitute for failing closed.

I only wanted Critical or High. UI:R because someone has to cause the victim browser to POST. PR:N on the vector because the attacker does not have a phpMyAdmin account. They need the victim's session cookie to be sent. Strict

makes classic cross-site forms fail. Same-site, a Lax/None deploy, or HTTP-auth still send it. The lab is a same-site loopback POST. Strict still sends the cookie.

03 Pinning RELEASE_5_2_3

Latest 5.2 stable. Tag `RELEASE_5_2_3`. Commit `962857e`. Image `phpmyadmin:5.2.3`. Cookie login as `appuser`. Marker row `PMA-CSRF-SQL-WITNESS`. Table `keepme` for the drop-table negative.

04 The skip list is half the hunt

- PMASA-2025-1 / 2 / 3 already patched in 5.2.2.
- Logout already fails closed on mismatch. Copy that.
- AJAX POSTs already abort on token mismatch. That is the control that `/sql` non-AJAX missed.
- Handlers that only read stripped `$_POST` (drop-table) hold. That is the negative, not a reason to skip `/sql`.
- Default Strict SameSite is documented. It is not "CSRF is closed."

I am not filing "PMA can run SQL with a valid token." I am filing SQL without one.

05 Surfaces that returned nothing

Same empty surfaces as the rest of the wave. Export `SaveDir`, partition DROP/TRUNCATE, account lock, `/config/set`, pmadb create are inferred siblings:

any handler that reads `$request->getParsedBody()` after the strip. I did not exercise those. `/sql` is the highest-impact instance that the lab actually ran.

06 The snapshot

PHP Copy

```php
$sql_query = '';
if ($request->isPost()) {
    $sql_query = $request->getParsedBodyParam('sql_query', '');
```

`$request` was `createFromGlobals()` before the token check. On mismatch, `$_POST` is empty. The object is not. Non-AJAX continues. `SqlController` uses global `$sql_query` if `$_POST` is empty. Route `GET|POST /sql`.

AJAX plus mismatch exits. That is why the same POST with `ajax_request=1` is Token mismatch and does not INSERT.

07 The lab

Loopback `http://127.0.0.1:19240`. Compose project `phpmyadmin-csrf-sql`. Same-site POST without a token returned the marker row `PMA-CSRF-SQL-WITNESS` in the SQL results. General log showed the SELECT as `appuser`. The same POST with `ajax_request=1` returned Token mismatch and did not INSERT. POST `/database/structure/drop-table` without a token left `keepme` in place.

PLAIN TEXT Copy

```
cd lab && ./run.sh
```

Last line:

PLAIN TEXT Copy

```
SUCCESS phpmyadmin-csrf-sql sql-ran=yes ajax-abort=yes drop-held=yes PMA-CSRF-SQL
```

08 Wrong turns, in the order I made them

Treating AJAX `/sql` as the leftover. It aborts Token mismatch. The leftover is non-AJAX.

Treating drop-table as the leftover. That handler only reads stripped `$_POST`. `keepme` stayed. That is the negative.

Treating default Strict SameSite as "CSRF is closed." The control is still broken. Same-site / Lax / None remain. The lab is same-site on purpose.

Claiming unauthenticated PMA. Cookie login required first. The attacker needs the victim session, not a password.

A reverse shell. Theatre. The oracle is a SELECT marker on loopback.

09 What an attacker can do

Anyone who can cause the victim browser to POST to phpMyAdmin without `ajax_request` can run SQL as that MySQL user.

PRIVILEGE

Run attacker SQL as the victim MySQL account (integrity + recon of that SQL box)

```
PRIVILEGE
```

Abort the same handler on AJAX

Skip handlers that only read stripped `$_POST`

Same gap on export (`SaveDir`), partition DROP/TRUNCATE, account lock, `/config/set` , pma

Closed on this lab: unauthenticated PMA, AJAX routes, drop-table without a token, OS shell. Default Strict SameSite means a classic cross-site form often will not send the cookie; the leftover is still a failed CSRF check on same-site and on Lax/None deploys.

I am not going to print a form you can paste at someone else's PMA. If you own the instance, run the proof of concept against loopback.

10 The fix

On mismatch, replace the request with an empty body, or refuse **all** POSTs, not only AJAX. Do not copy `sql_query` from a stale request. Fail closed like logout.

Same pin leftovers: phpmyadmin-webauthn, phpmyadmin-pmadb-central-sqli, phpmyadmin-designer-json-sqli, phpmyadmin-setup-validate.

11 References

- Proof of concept: abraxas/phpmyadmin-csrf-sql · phpmyadmin-csrf-sql-Abraxas-Labs.py
- Lab: `lab/` · docker-compose.yml · run.sh · poc.py
- @abraxas_null · github.com/abraxas · abraxaslabs.tech · abraxas.null@proton.me
- CWE-352 · 8.1 High
- Tree: phpmyadmin/phpmyadmin RELEASE_5_2_3 (`962857e`) · `Common.php` · `Sanitize::removeRequestVars` · `SqlController`

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