0-DAY NO CVE HIGH PUBLIC

# phpMyAdmin central columns pmadb SQLi

phpMyAdmin 5.2.3 concatenates POST extra and attribute into a single-quoted pmadb INSERT/UPDATE as the control user. Neighboring fields are escaped. These two are not. Labbed. No CVE yet.

| | |
|---|---|
| NAME<br>phpMyAdmin central columns pmadb SQLi | TYPE<br>0-day analysis |
| CVE<br>n/a | CVE RISK<br>high |
| DISCLOSURE STATUS<br>public | VENDOR<br>phpMyAdmin |
| AFFECTED<br>phpMyAdmin 5.2.3 (962857e) with pmadb central columns enabled. CVE-2016-5703 was the database name, not these fields. | PUBLISHED<br>12 Oct 2026 |
| UPDATED<br>12 Oct 2026 | TAGS<br>0-day, phpmyadmin, sqli, pmadb, cwe-89, authenticated |

TL;DR

Lab files and proof-of-concept here.

## 01 Extra and attribute never got `escapeString`

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

This is **phpMyAdmin 5.2.3** ( `962857e` ). No CVE yet. CWE-89. 8.1 High ( `CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N` ). Remote SQLi. Authenticated cookie login. The sink is the **control user** on `pma__central_columns` , not the signed-in SQL box.

phpMyAdmin configuration storage is a second MySQL account. Bookmarks, history, designer, central columns, userconfig. The UI is the logged-in user. The writes are `queryAsControlUser` / `CONNECT_CONTROL` . Neighboring fields on the same INSERT go through `escapeString` . `col_extra` and `field_attribute` are `implode` into a single-quoted literal. A quote in the column name stores as `neg'col` . A quote in the attribute returns MySQL 1064 as the control user, and the error body contains the witness.

CVE-2016-5703 was the **database name** in central columns. These extra and attribute fields were not that bug.

## 02 Why phpMyAdmin, why this column

Same wave as WebAuthn. Eight parallel source hunts of `RELEASE_5_2_3` . SQLi/pmadb was the hunt that was supposed to come back empty because "the user already has a SQL box." That sentence is true for `appdb` . It is false for `phpmyadmin.*` as the control user. The logged-in account in this lab has no GRANT on `phpmyadmin` . The control user does.

I only wanted Critical or High. A leftover here is unescaped bytes on the control link, not "PMA can run SELECT." phpMyAdmin 6.x `quoteString` s the designer JSON blob. 5.2.3 still raw-concats this column. That is the shape.

## 03 Pinning RELEASE_5_2_3

Latest 5.2 stable. Tag `RELEASE_5_2_3`. Commit `962857e`. `Version::VERSION = 5.2.3`. Image `phpmyadmin:5.2.3`. Official image, not a source build. pmadb on, central columns table created, control user granted DML on `phpmyadmin.*`.

## 04 The skip list is half the hunt

- PMASA-2025-1 / 2 / 3 already patched in 5.2.2. Not this.
- CVE-2016-5703 central columns **database name**. Different field, same feature. Do not re-file that.
- Designer username / db-name CVEs (2016-6616, 2019-11768, 2019-18622). Different leftover. The JSON blob is phpmyadmin-designer-json-sqli.
- Unquoted `authentication_plugin` in `ALTER USER`. Different sink. Own-password path. Not pmadb.
- ImportLdi `TERMINATED BY`. FILE-gated. Medium.

I am not filing pmadb-off. Feature no-ops. That is a FAIL, not this leftover.

## 05 Surfaces that returned nothing

XSS, file/LFI, deser, transformations, changelog leftovers at this bar came back empty or Medium for this hunt. Auth password held. CSRF is a different leftover. Setup SSRF is a different leftover. The pmadb hunt had two Highs: this column, and the designer JSON blob.

`db`, `col_name`, `col_type`, `col_length`, `collation`, and `col_default` are `escapeString`'d on the same statements. That is the negative control. Use it.

## 06 The concat

`libraries/classes/Database/CentralColumns.php` `getInsertQuery`:

```php
. '\'' . implode(',', [$extra, $attribute])
. '\',\'' . $this->dbi->escapeString($default) . '\');';
```

`updateOneColumn` UPDATE path:

```php
. ', col_extra = \''
. implode(',', [$col_extra, $col_attribute]) . '\''
```

`Controllers/Database/CentralColumnsController.php` POST `multi_edit_central_column_save` and `edit_save` / `add_new_column`. Route `GET|POST /database/central-columns`.

A signed-in user who can open Central columns POSTs a quote in `field_attribute`. `tryQuery(..., CONNECT_CONTROL)` runs that string as the control user.

## 07 The lab

Loopback `http://127.0.0.1:19210`. Compose project `phpmyadmin-pmadb-central-sqli`. Images `phpmyadmin:5.2.3` and `mysql:8.0`. Cookie login as `appuser`. Central columns page loads.

Quote in `field_name` stored as `neg'col`. `escapeString` held. `poscol` is not inserted. Quote in `field_attribute` returned `#1064` as the control user. Error body contained `PMA-PMADB-CENTRAL-SQLI-WITNESS`. General log did not show a successful unescaped write. mysqli does not enable multi-query by default. This lab stops at the broken string.

PLAIN TEXT Copy

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

Last line:

PLAIN TEXT Copy

```
SUCCESS phpmyadmin-pmadb-central-sqli how=live insert-syntax=yes field-name-escap
```

## 08 Wrong turns, in the order I made them

**Treating a quote in `field_name` as the leftover.** It stores as `neg'col`. That is the negative. The leftover is extra / attribute.

**Expecting the general log to show the unescaped witness.** The INSERT dies with `#1064` before a successful write. The oracle is the control-user error body.

**Treating pmadb-off as this leftover.** Feature no-ops. FAIL, not SUCCESS.

**Claiming stacked queries.** Default `mysqli::query` is single-statement. Extra statements are inferred if an install enables multi-query. This lab did not.

**Calling it the signed-in SQL box.** `appuser` has appdb. The control user has `phpmyadmin.*`. Different account, different privilege.

**A reverse shell.** Theatre. The oracle is a `#1064` JSON error on loopback.

## 09 What an attacker can do

A signed-in phpMyAdmin user who can open Central columns can inject SQL as the **control user** on `pma__central_columns`.

| PRIVILEGE | METHOD / PATH |
| --- | --- |
| **Break the control-user INSERT/UPDATE string (integrity of pmadb SQL)** | `POST index.php?route=/database/c` `multi_edit_centra` with a quote in `fie` `col_extra` |
| **Keep neighboring fields in-band** | Same POST, quote |
| **Read or overwrite other users' central-column rows** | Same sink, once th |
| **Extra statements as the control user** | Same sink if the co allows multi-query |
| **Reach other pmadb tables (bookmarks, history, `pma__userconfig`)** | Only if stacked que style write is possib |

Closed on this lab: unauthenticated access, the signed-in appuser SQL box (appdb only), OS shell, and stacked queries. Feature no-ops if pmadb / central columns is off.

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

## 10 The fix

`escapeString` / `quoteString(..., CONNECT_CONTROL)` on extra and attribute. Allowlist `auto_increment`, `UNSIGNED`, `BINARY`, `UNSIGNED ZEROFILL`, and `on update CURRENT_TIMESTAMP`. phpMyAdmin 6.x `quoteString` on the designer JSON blob is the patched shape for this class of pmadb concat.

Same pin leftovers: phpmyadmin-webauthn, phpmyadmin-designer-json-sqli, phpmyadmin-setup-validate, phpmyadmin-csrf-sql.

## 11 References

- Proof of concept: abraxas/phpmyadmin-pmadb-central-sqli · phpmyadmin-pmadb-central-sqli-Abraxas-Labs.py
- Lab: `lab/` · Dockerfile · docker-compose.yml · run.sh · poc.py
- @abraxas_null · github.com/abraxas · abraxaslabs.tech · abraxas.null@proton.me
- CWE-89 · 8.1 High
- Tree: phpmyadmin/phpmyadmin RELEASE_5_2_3 ( `962857e` ) · `CentralColumns.php` `getInsertQuery` / `updateOneColumn`

**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