0-DAY NO CVE HIGH PUBLIC

CakePHP, COLLATE collation SQL injection

CakePHP 5.4.2 StringExpression concatenates collation. Value is bound. Postgres quoteIdentifier is trim-and-wrap, so an embedded quote breaks CAST. Lab executed a SELECT oracle. No CVE yet.

NAME CakePHP, COLLATE collation SQL injection	TYPE 0-day analysis
CVE n/a	CVE RISK high
DISCLOSURE STATUS public	VENDOR CakePHP
AFFECTED CakePHP 5.4.2 (887b641) cakephp/database. App that forwards collation/locale into StringExpression or IdentifierExpression. Default cakephp/app does not expose collation on a public form by itself.	PUBLISHED 12 Oct 2026
UPDATED 12 Oct 2026	TAGS 0-day, cakephp, sql-injection, cwe-89, postgres, authenticated

TL;DR

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

01 The value is bound. The collation is not

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

This is **CakePHP 5.4.2** (`887b641de9f7aba06071f98f17edcbc573cbcf63`). `cakephp/database` 5.4.2. 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. Needs an application that forwards collation or locale into `StringExpression` / `IdentifierExpression`.

CVE-2026-79752 put `ensureSimpleString()` on `cast` / `extract` / `dateAdd`. Collation is the same kind of fragment: it is not a bound value, it is a SQL identifier-shaped token, and Postgres wrapping is `'"' . trim($collation, '"') . '"'`. An embedded quote closes `CAST` early. SQLite and MySQL still concatenate. Postgres is the wrap that turns one quote into a breakout.

That is the leftover. Same class as the GHSA. Different argument. Same tag.

02 Why CakePHP

Same pass as the window leftover. Framework, not a CMS. The bar is a documented builder API that concatenates a structural fragment after the project already announced that class of bug closed. I only wanted Critical or High. Local is in-scope. Medium stays on the floor unless it chains.

Cake's own intake is security@cakephp.org.

03 Pinning 5.4.2

Latest calendar tag. `VERSION.txt` `5.4.2`. Commit `887b641de9f7aba06071f98f17edcbc573cbcf63`. After FunctionsBuilder and `jsonValue`. After `filename=`. After `Query::limit` became an int. Collation was still a string glued onto `COLLATE`.

04 The skip list is half the hunt

Same list as the window post. FunctionsBuilder is closed. `jsonValue` is closed. `limit` / `offset` are ints. View `element()` path is closed. Paginator XSS is closed. RateLimit default is `remote_addr`. Multipart `filename=` is stripped. Smtp CRLF pair-only strip is a sibling, not this leftover.

Hunts that returned nothing in this bar: CSRF / FormProtection / `_method`; View LFI siblings; BodyParser XXE; mass assignment; helper XSS leftovers; `clientIp()` without `trustProxy`.

05 Ten surfaces, this one executes

The remaining-SQLi pass had two leftovers. Window `range()` is compile-only on SQLite. Collation is the one I could execute.

`StringExpression::sql`:

PHP Copy

```php
$placeholder = $binder->placeholder('c');
$binder->bind($placeholder, $this->string, 'string');
return $placeholder . ' COLLATE ' . $this->collation;
```

Value bound. Collation concatenated. `IdentifierExpression` does the same for `note COLLATE <payload>`. There is no `ensureSimpleString`.

Postgres `quoteIdentifier` is not identifier quoting. It is:

```php
return '"' . trim($collation, '"') . '"';
```

An embedded `"` closes the identifier. `--` eats the leftover closer Cake still appends. Compiled SQL becomes `COLLATE "C" AS text)) || (SELECT note FROM lab_secret LIMIT 1) --`.

Chain: `new StringExpression($term, $request->getQuery('collation'))` → `:c COLLATE <user>`. Stock `cakephp/app` does not expose collation on a public form. An app that forwards locale or a collation picker does. Same AT:P model as 79752.

06 What I almost filed instead

SQLite as the only driver. Concat is real on SQLite. The wrap-breakout is Postgres. A compile-only SQLite lab would have been a weaker sibling of the window leftover. I started Postgres 16.

Treating the bound value as the injection. `:c0` stays the literal `x`. The hole is the second argument.

Claiming `ensureSimpleString` already ran here. It did not. A collation that would pass `^[a-zA-Z0-9]+$` is not this payload. That is the fix, not the current code.

A reverse shell. Theatre.

07 The lab

`cakephp/database` **5.4.2** plus Postgres 16. Tiny app forwards `?collation=` into `StringExpression` / `IdentifierExpression` and executes `CAST(... AS text)`. Secret table holds `CAKEPHP-COLLATE-SQLI-WITNESS`. Loopback `http://127.0.0.1:19450`. Compose project `cakephp-collate-sqli`.

01 Health pin 5.4.2.

02 Control `collation=C` → raw `:c0 COLLATE C`, compiled `COLLATE "C"`, result `x`.

03 Inject `C" AS text)) || (SELECT note FROM lab_secret LIMIT 1) --`.

04 Compiled SQL contains the Postgres wrap breakout and `lab_secret`. Execute returns `xCAKEPHP-COLLATE-SQLI-WITNESS`. `ensure_simple_ok=false`.

Last line:

PLAIN TEXT Copy

```
SUCCESS cakephp-collate-sqli pin=5.4.2 raw=concat pg-wrap=breakout exec=yes CAKEP
```

Replay:

PLAIN TEXT Copy

```
git clone https://github.com/abraxas/cakephp-collate-sqli
cd cakephp-collate-sqli
python3 cakephp-collate-sqli-Abraxas-Labs.py
```

or `cd lab && ./run.sh`. Bind it to loopback.

08 Wrong turns, in the order I made them

Stopping at SQLite concat. It proves the fragment is raw. It does not prove a table read. Postgres wrap is the breakout.

Looking at the bound placeholder. The first argument is fine. The second is not.

Treating `quoteIdentifier` as quoting. `trim` plus wrap is the opposite of quoting. It is a closer for anyone who brought their own quote.

Inflating to unauthenticated RCE on a skeleton app. Default `cakephp/app` does not forward collation. The class is SQLi in the expression API.

A reverse shell. Theatre. The oracle is a SELECT of a witness row.

09 What an attacker can do

Needs an application that passes user/locale collation into those constructors.

Integrity of compiled SQL (lab-proven). `StringExpression::sql` concatenates `$this->collation`. Raw SQL contains the payload unbound. `IdentifierExpression` is the same splice on `note COLLATE <payload>`.

Integrity of Postgres wrap (lab-proven). `quoteIdentifier` trim-and-wrap. Compiled SQL is `COLLATE "C" AS text)) || (SELECT ...`.

Recon of the database (lab-proven). Execute returns `xCAKEPHP-COLLATE-SQLI-WITNESS`. No exec error.

Integrity of rows (inferred). Same injection with the connection role can modify. Lab is a SELECT oracle.

Closed. Bound string value (`:c0` stays `x`). A collation that would pass `ensureSimpleString`. Unauthenticated RCE on a skeleton app. A shell. Control collation `C` returns `x` and compiles `COLLATE "C"`.

10 The fix

`ensureSimpleString()` (or a collation allowlist) on the collation argument. Postgres must quote identifiers properly, not `trim` plus wrap.

I am not going to print a collation string you can paste at someone else's locale picker. The missing allowlist is the useful part. If you own the app, run the proof of concept against loopback.

11 References

- Proof of concept: abraxas/cakephp-collate-sqli · cakephp-collate-sqli-Abraxas-Labs.py
- Lab: `lab/` · Dockerfile · docker-compose.yml · run.sh · poc.py · index.php
- @abraxas_null · github.com/abraxas · abraxaslabs.tech · abraxas.null@proton.me
- CWE-89
- Tree: cakephp/cakephp 5.4.2 (`887b641de9f7aba06071f98f17edcbc573cbcf63`) · `StringExpression.php` · `IdentifierExpression.php` · `Postgres.php`
- Nearby patched: CVE-2026-79752
- Same pin: cakephp-window-sqli · cakephp-attachment-crlf · cakephp-cache-unserialize

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

Home Research Open Source Reading Room About GitHub @abraxas_null Console

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

© 2026 Abraxas Labs