Abraxas Labs
ABRAXAS

DARK LIGHT

Research / cakephp-window-sqli

0-DAY NO CVE HIGH PUBLIC

CakePHP, WindowExpression range/frame SQL injection

CakePHP 5.4.2 WindowExpression::range() and frame() still splice string offsets into OVER. rows()/groups() are int. Sibling of CVE-2026-79752. Labbed compile-only. No CVE yet.

NAME CakePHP, WindowExpression range/frame 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 request data into →over()→range($start, $end) or frame(). Default cakephp/app does not expose this on a public form by itself.	PUBLISHED 12 Oct 2026
UPDATED 12 Oct 2026	TAGS 0-day, cakephp, sql-injection, cwe-89, authenticated

TL;DR

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

01 rows() got the int. range() did not

I am @abraxas_null. The proof of concept is on GitHub: **abraxas/cakephp-window-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 request data into `->over()->range($start, $end)` or `frame()`.

CVE-2026-79752 taught CakePHP that structural SQL fragments are not values. `cast`, `extract`, `datePart`, `dateAdd` grew `ensureSimpleString()`. Window frames were not on that list. `rows()` and `groups()` already take `?int`. `range()` still takes a string, because interval literals look like `"'1 day'"` and the framework tests pass them raw. `buildOffsetSql` concatenates the offset, the frame type, and `PRECEDING` / `FOLLOWING`. The binder sits in the signature and does nothing unless you already wrapped the offset in an `ExpressionInterface`.

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

02 Why CakePHP

I had just finished unpublished Crit/High hunts of products that ship a server. CakePHP is a framework. The bar changes. You do not get a finding because `query($userSql)` exists. You get a finding when a documented builder API concatenates a structural fragment the way `cast()` used to, 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. `cakephp/authentication` is a different repo. I wrote that on the skip list before I opened `FunctionsBuilder`.

Cake's own intake is security@cakephp.org. SECURITY.md on 5.4.2 still points at that mailbox.

03 Pinning 5.4.2

Latest calendar tag on the day I cloned. `VERSION.txt` says `5.4.2`. HEAD commit `887b641de9f7aba06071f98f17edcbc573cbcf63`. The bakery notes for that tag mention multipart `filename=` injection. The GHSA wave for FunctionsBuilder and `jsonValue` had already landed. That is the interesting pin: after the SQL-fragment lesson, before anyone asked whether window frames got the same lesson.

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" a CVE with a press release. On 5.4.2 the interesting ones were already closed:

- CVE-2026-79752 FunctionsBuilder `cast` / `extract` / `datePart` / `dateAdd`. `ensureSimpleString()`.
- CVE-2026-77635 `jsonValue()` Postgres SQLi. Path bound, not interpolated.
- CVE-2026-77634 SmtpTransport CRLF in custom headers. Incomplete sibling, pair-only strip. Not this post.
- CVE-2026-48820 `View::element()` path. `_checkFilePath()` on element/template/layout.

- CVE-2026-23643 `PaginatorHelper::limitControl()` XSS. `h($fieldName)`.
- GHSA-6g8q-qfpv-57wp `Query::limit` / `offset` SQLi. Typed `int`.
- 5.4.1 bakery: `RateLimitMiddleware` default `ipHeader` is now `remote_addr`.
- 5.4.2 notes: multipart `Content-Disposition` `filename=` strip.

`cakephp/authentication` GHSA-hhpq-7wg4-36jm is a different repo. Core Routing has no matching backslash bypass.

05 Ten surfaces, four leftovers

I split the tree: unserialize/cache/cookies, CSRF/FormProtection, View/LFI/files, remaining SQLi, mail CRLF siblings, IP/Host/proxy/SSRF, XXE/BodyParser, mass-assignment/crypto, XSS/helpers/routing, Http Client/cookie jar.

Most of that is Cake doing what Cake said it would do.

CSRF `_method` override is patched. View `element()` path is patched. BodyParser XML is off by default and `loadEntities` is false. Mass assignment is documented `$_accessible = ['*' ⇒ true]`. Helper XSS leftovers did not clear this bar. `clientIp()` `HTTP_CLIENT_IP` without `trustProxy` is already gated (LegalHackers). `Query::limit` / `offset` are ints.

The remaining-SQLi pass is where the temperature changed. FunctionsBuilder was clean. Window frames were not.

06 The test still wants a raw interval

`WindowExpressionTest::testRange` still asserts:



PHP Copy

```php
$w = (new WindowExpression())->range("'1 day'", "'10 days'");
```

That is an honest test for a SQL dialect that writes `RANGE BETWEEN '1 day' PRECEDING`. It is also a contract that the offset is a fragment, not a value. `rows()` and `groups()` already refused that contract. They take `?int`. `range()` and `frame()` take `ExpressionInterface|string|int|null`.

`buildOffsetSql`:

PHP Copy

```php
protected function buildOffsetSql(
    ValueBinder $binder,
    ExpressionInterface|string|int|null $offset,
    string $direction,
): string {
    if ($offset === 0) {
        return 'CURRENT ROW';
    }
    if ($offset instanceof ExpressionInterface) {
        $offset = $offset->sql($binder);
    }
    return sprintf('%s %s', $offset ?? 'UNBOUNDED', $direction);
}
```

`$binder` is unused unless you already wrapped the offset. A string from `$request->getQuery('start')` lands in `RANGE BETWEEN <user> PRECEDING`. Frame type and direction are concatenated the same way in `frame()`.

Chain: `$query->func()->count('*')->over()->range($request->getQuery('start'), $request->getQuery('end'))` → `RANGE BETWEEN <user> PRECEDING AND CURRENT ROW`.

Default `cakephp/app` does not call `range()` with query-string data. The leftover is the documented window API, the same AT:P model as CVE-2026-79752.

07 What I almost filed instead

Treating `rows()` as the hole. It is typed. GET `/?mode=rows&n=3` compiles to `ROWS BETWEEN 3 PRECEDING`. Numeric. Not quoted. Not bound. That is the control, not the bug.

Claiming stock `cakephp/app` exposes this on a public form. It does not. An analytics report builder that forwards "last N days" into `range()` does. That is the victim model. Same as `cast($col, $_GET['type'])` was the victim model for 79752.

Executing the injected SQL in SQLite as the oracle. POC-READY for this leftover is compile-only: the fragment must appear raw in `RANGE BETWEEN`, not bound, not quoted. SQLite will not make a comment-splice prove row read. I left it as compile. The collate leftover on the same pin is the one that executes.

A reverse shell. Theatre.

08 The lab

Tiny PHP app. `cakephp/database` **5.4.2**. SQLite `:memory:`. `php:8.2-apache`. Loopback `http://127.0.0.1:19440`. Compose project `cakephp-window-sqli`. Witness `CAKEPHP-WINDOW-SQLI-WITNESS`.

GET `/?mode=ping` returns package version 5.4.2. GET `/?mode=range&start=` is passed as a string into `range()`. GET `/?mode=rows&n=3` is the int control.

Injected start `1) /* CAKEPHP-WINDOW-SQLI-WITNESS */ --` compiles to:

PLAIN TEXT Copy

```
1  SELECT (COUNT(*) OVER (RANGE BETWEEN 1) /* CAKEPHP-WINDOW-SQLI-WITNESS */ -- PREC
```

Binder empty. Not quoted. Control:

```
PLAIN TEXT
1 SELECT (COUNT(*) OVER (ROWS BETWEEN 3 PRECEDING AND CURRENT ROW)) AS n FROM sqlit
```

Last line:

```
PLAIN TEXT
1 SUCCESS cakephp-window-sqli pin=5.4.2 range-raw=yes rows-numeric=yes bound=no quo
```

Replay:

```
PLAIN TEXT
1 git clone https://github.com/abraxas/cakephp-window-sqli
2 cd cakephp-window-sqli
3 python3 cakephp-window-sqli-Abraxas-Labs.py
```

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

09 Wrong turns, in the order I made them

Pinning 5.2.13 again. That is the CVE-2026-79752 lab. This leftover is the tag after that fix.

Looking at `FunctionsBuilder` first. It is clean. `ensureSimpleString()` holds. The leftover is the next structural API.

Treating a framework test that passes `"'1 day'"` as proof the API is safe. It is proof the API is a concatenator.

Using SQLite execute as SUCCESS. A comment-splice can compile and still not read a table. Compile-only is the honest oracle here.

Inflating to unauthenticated RCE on a skeleton app. Default `cakephp/app` does not forward query-string data into `range()`. The class is SQLi in the window API. Not a CMS route.

A reverse shell. Theatre. The oracle is a witness comment in compiled SQL.

10 What an attacker can do

A user who can set the window bound the app forwards (analytics "last N days", a report builder) changes the `OVER` clause into extra SQL.

Integrity of compiled SQL (lab-proven). `GET /?mode=range&start=` into `AggregateExpression::range($start, $end)`. Compiled SQL is `RANGE BETWEEN 1) /* CAKEPHP-WINDOW-SQLI-WITNESS */ -- PRECEDING` with an empty binder.

Integrity / recon of the database (inferred). Same query, executed, reads or changes rows as the app's DB user. This lab does not execute the splice.

Integrity of frame type / direction (inferred). `frame($type, $offset, $direction, ...)` concatenates those strings too. Lab used `range()`.

Closed. `rows()` / `groups()` typed `?int`. FunctionsBuilder `cast` / `extract` / `dateAdd` / `jsonValue`. `Query::limit` / `offset`. Unauthenticated RCE on a skeleton app. A shell.

11 The fix

Bind numeric offsets. Allowlist frame type (`RANGE` / `ROWS` / `GROUPS`) and direction (`PRECEDING` / `FOLLOWING`). For interval strings, quote them or require `ExpressionInterface`. Add a test that a non-int `range()` offset becomes a placeholder or a quoted literal, not raw SQL.

I am not going to print a query-string you can paste at someone else's report builder. The missing bind is the useful part. If you own the app, run the proof of concept against loopback.

12 References

- Proof of concept: abraxas/cakephp-window-sqli · cakephp-window-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`) · `WindowExpression.php` · `AggregateExpression.php`
- Nearby patched: CVE-2026-79752 · CVE-2026-77635
- Same pin: cakephp-collate-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