0-DAY NO CVE MEDIUM PUBLIC

CakePHP, attachment Content-Type / Content-ID CRLF leftover

CakePHP 5.4.2 strips CR/LF from multipart filename=. Content-Type and Content-ID are still concatenated raw. UploadedFile client MIME is copied through. Labbed on getBodyString(). No CVE yet.

NAME CakePHP, attachment Content-Type / Content-ID CRLF leftover	TYPE 0-day analysis
CVE n/a	CVE RISK medium
DISCLOSURE STATUS public	VENDOR CakePHP
AFFECTED CakePHP 5.4.2 (887b641). App that attaches a user upload (UploadedFile client MIME) or a user contentId. filename= strip on this pin still holds.	PUBLISHED 12 Oct 2026
UPDATED 12 Oct 2026	TAGS 0-day, cakephp, crlf, mime, cwe-93, authenticated

TL;DR

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

01 filename= got the strip. Content-Type did not

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

This is **CakePHP 5.4.2** (`887b641de9f7aba06071f98f17edcbc573cbcf63`). No CVE yet. CWE-93. 5.4 Medium (`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:N`). Remote MIME-part header injection. Needs an application that attaches a user upload or a user `contentId` .

5.4.2 already strips `"` , `\r` , and `\n` from multipart `name=` / `filename=` (`FormDataPart::_headerParameterToString`). That is the bakery note. That is the control. `Content-Type` and `Content-ID` were left as raw concatenation. `Message::setAttachments` copies `UploadedFile::getClientMediaType()` onto the type field with no extra filter.

The extra line lives in the SMTP body, not the envelope. It does not add `RCPT` by itself. It can still smuggle part headers or break the multipart structure. `filename=` with CRLF on this pin is stripped. That control still holds.

02 Why CakePHP

Same 5.4.2 leftover pass as window and collate. Framework. Documented mailer API. Incomplete sibling of a fix that already shipped. I only wanted Critical or High, and this one is Medium. It still chains with the same pin's mail story: GHSA-2qh5 closed pair `\r\n` in custom headers, 5.4.2 closed `filename=` , and these two header fields were left.

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

03 Pinning 5.4.2

Latest calendar tag. The bakery notes for that tag are the reason I opened `FormDataPart`. If you just fixed `filename=`, the next question is every other concatenation in `__toString()`.

04 The skip list is half the hunt

FunctionsBuilder SQLi closed. `jsonValue` closed. View `element()` path closed. Paginator XSS closed. RateLimit default `remote_addr`. `Query::limit` / `offset` ints. SmtpTransport pair-only strip is a sibling leftover, not this sink. Envelope `Bcc` via this MIME-part path is closed for this leftover.

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

05 The sanitizer stops at parameters

`_headerParameterToString`:

```php
$transliterated = Text::transliterate(str_replace(['"', "\r", "\n"], '', $value))
```

`__toString()` still does:

```php
```

```
if ($this->type) {
    $out .= 'Content-Type: ' . $this->type . "\r\n";
}
if ($this->contentId) {
    $out .= 'Content-ID: <' . $this->contentId . ">\r\n";
}
```

`Message::setAttachments`:

PHP Copy

```
} elseif ($fileInfo['file'] instanceof UploadedFileInterface) {
    $fileInfo['mimetype'] = $fileInfo['file']->getClientMediaType();
```

`attachFiles` / `attachInlineFiles` then call `$part->type($fileInfo['mimetype'])` and `$part->contentId($fileInfo['contentId'])`. Client MIME is attacker-controlled. So is an explicit `contentId` the app copied from a form.

06 What I almost filed instead

Treating envelope `RCPT` as this leftover. The extra line is under the MIME part. SmtpTransport pair-only strip is the envelope sibling. Different method. Different post.

Claiming `filename=` still injects. It does not. That is the control. If the control fails, the pin is wrong.

Sending SMTP. The oracle is `Message::getBodyString()`. I did not send mail.

A reverse shell. Theatre.

07 The lab

PHP 8.3. CakePHP tag 5.4.2 cloned at build. `composer install --no-dev`. `lab_server.php` exposes `Message::setAttachments` on loopback `http://127.0.0.1:19460`. Compose project `cakephp-attachment-crlf`. Witness `CAKEPHP-ATTACHMENT-CRLF-WITNESS`. Four cases.

Filename with CRLF: no injected header line. Explicit mimetype, explicit contentId, and Laminas `UploadedFile::getClientMediaType()` each emit `X-Injected: CAKEPHP-ATTACHMENT-CRLF-WITNESS`.

Last line:

PLAIN TEXT Copy

```
SUCCESS cakephp-attachment-crlf how=live pin=5.4.2 sha=887b641de9f7aba06071f98f17
```

Replay:

PLAIN TEXT Copy

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

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

08 Wrong turns, in the order I made them

Stopping at `filename=`. The bakery note is the skip. The leftover is the fields next to it.

Calling it envelope injection. Body part headers are not `RCPT`. CVSS stays Medium because of that.

Using `mail()` as the oracle. Transport is out of scope for this sink. `getBodyString()` is the MIME.

Inflating to RCE. Extra MIME headers are not PHP. Not a shell.

A reverse shell. Theatre. The oracle is a second header line under the part.

09 What an attacker can do

Someone who can influence attachment metadata the app passes into `Message::setAttachments` / `addAttachment` can inject MIME-part headers.

Integrity of MIME part headers (lab-proven). Extra `Content-Type` line via `'mimetype' ⇒ "text/plain\\r\\nX-Injected: …"`. Extra `Content-ID` line via `'contentId' ⇒ "cid@…>\\r\\nX-Injected: …"`. Same injection via `UploadedFile::getClientMediaType()`. Witness line present in all three.

Closed (lab-proven). `filename=` CRLF is stripped. Envelope `RCPT` / `Bcc` via this sink is closed. SMTP send is closed. Lab calls `getBodyString()` only.

Inferred, not demonstrated. A crafted type or contentId that includes a multipart boundary could split the message into extra parts in a receiving MUA. I did not send mail.

10 The fix

Run `type`, `contentId`, and `transferEncoding` through the same `"` / `\r` / `\n` strip as `_headerParameterToString`. Reject the value if it still contains a line break after strip. Cover UploadedFile client MIME and explicit `mimetype` / `contentId` in the mailer tests, not only `filename=`.

I am not going to print an upload you can paste at someone else's contact form. The missing strip is the useful part. If you own the app, run the proof of concept against

loopback.

11 References

- Proof of concept: abraxas/cakephp-attachment-crlf · cakephp-attachment-crlf-Abraxas-Labs.py
- Lab: `lab/` · Dockerfile · docker-compose.yml · run.sh · poc.py · lab_server.php
- @abraxas_null · github.com/abraxas · abraxaslabs.tech · abraxas.null@proton.me
- CWE-93
- Tree: cakephp/cakephp 5.4.2 (`887b641de9f7aba06071f98f17edcbc573cbcf63`) · `FormDataPart.php` · `Message.php`
- Same pin: cakephp-window-sqli · cakephp-collate-sqli · cakephp-cache-unserialize

A
X
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