



Live feeds

About

Mastodon is the best way to keep up with what's happening.

Follow anyone across the fediverse and see it all in chronological order. No algorithms, ads, or clickbait in sight.

Create account

Login



◆◆Catherine◆◆

@whitequark@treehouse.systems

11h*

btw you *might* be able to patch the PHP vuln with `session.upload_progress.enabled = off`

source: x.com/J4ck3LSyN/status/2112703... (i don't give a fuck about the embargo, i need to defend things)

edit: see boost below, this is not actually enough according to a php core dev

```
zval *post_bytes_processed; /* -> data['bytes_processed'] bucket */
zval *current_file_bytes_processed; /* -> current_file['bytes_processed'] */

Bug 1 (CVE-2026-109619): 'post_bytes_processed' cached ONCE at first FILE_START
(session.c:3091 / 8.4.4:3316); never refreshed on later FILE_START.
Bug 2 (CVE-2026-109618): 'current_file_bytes_processed' refreshed at EVERY
FILE_START (session.c:3117 / 8.4.4:3342) but persists across chunks of ONE file.

## 2b. Why they dangle
FILE_DATA stores through the pointers (session.c:3091-area :3355/:3356) and
FILE_END stores the outer one (:3375).
On the SECOND update() (post min_freq/stall), 'php_session_initialize()'
re-reads the session file; 'php_session_track_init()' (session.c:246) destroys
the PRIOR decoded tree; then 'zend_hash_update' (session.c:3214) replaces the
freshly-decoded tree, calling its HashTable destructor 'zval_ptr_dtor'.
'add_next_index_zval' (zend_hash.c:1192) and 'add_assoc_zval_ex' (zend_API.c)
are ZVAL_COPY_VALUE (NO address): the inner 'current_file[]' and 'files[]'
arrays end up double-referenced at refcount 1. When the decoded tree is torn
down, the arrays can be freed while a cached bucket pointer still targets them.
```

X (formerly Twitter) · 12h

Jłckłı (@J4ck3LSyN) on X

I have successfully recreated & lab/field tested CVE-2026-109618...

2



Gina Peter Banyard

@Girgias@phpc.social

10h

@whitequark The only thing I can say is that the only mitigation is patching / upgrading PHP. I sadly know too much.

Oct 21, 2026, 04:13 AM · 🌐 · Island One

74 boosts · 13 quotes · 88 favorites



◆◆Catherine◆◆

@whitequark@treehouse.systems

10h

@Girgias grim. well, this wasn't my only line of defense

1



Gina Peter Banyard

@Girgias

10h

@whitequark if you can afford turning of a server for a day, that might be the most sensible thing to do. But that is far from a reasonable thing to demand for most people.

2



◆◆Catherine◆◆

@whitequark@treehouse.systems

10h

@Girgias I can afford to turn off most things except my only PHP instance. but I put it behind basic auth on nginx. should be good enough!

0



Alper Çuğun-Gscheidel

@alper@rls.social

2h

@Girgias @whitequark If this is your suggested mitigation, then there may be a large chain of choices that you need to reconsider.

Given the codebase in question there's nothing preventing another dozen of these popping up in the next years.

0



Eesger

@eesger@mastodon.frl

3h

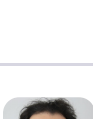
@Girgias @whitequark

but.. but does setting:
`session.upload_progress.enabled = 0`

Do nothing, or not enough?

And, you state that "patching / upgrading PHP." is the only real solution.. that usually is.. but could you share tips on how to prevent abuse (besides shutting down PHP.)

0



Xela

@xela@troet.cafe

3h*

@Girgias how about turning file uploads off? Would that help until official patches get rolled out?

Something like `file_uploads = Off` if the application can work properly without uploads...

Edit: never mind, no it wouldn't, but there are instructions to catch multipart headers before they reach PHP.

1



David Gerard

@davidgerard@circumstances.run

1h

@xela @Girgias it's anything that takes POST

0



xmripper

@xmripper

4h

@Girgias @whitequark Hi Gina, will affected EOL PHP versions receive security patches for these CVEs too?

1



Latte macchiato

@privateger@plasmatrap.com

3h

@xmripper@phpc.social @Girgias@phpc.social @whitequark@social.treehouse.systems no

1



Oliver Maksimović

@oliver

3h

@privateger @xmripper @Girgias @whitequark well that sucks. if the scale of the issue is so big, then older officially unsupported versions should still get the patch instead of shrugging it off. not everyone can just jump several versions up in a day

2



Latte macchiato

@privateger@plasmatrap.com

3h

@oliver@phpc.social @xmripper@phpc.social @Girgias@phpc.social @whitequark@social.treehouse.systems PHP itself never patches end-of-life branches; fixes only land on the four supported branches. Versions get two years of feature support, then 2 more years of security patches.

Some distros do backport patching for versions not supported by upstream anymore, like RHEL/ESM or Debian LTS. If you are on such an old version, you really should have some means of getting patches anyway; otherwise, you already have holes.

1



Latte macchiato

@privateger@plasmatrap.com

3h

@oliver@phpc.social @xmripper@phpc.social @Girgias@phpc.social @whitequark@social.treehouse.systems see <https://www.php.net/supported-versions.php>

0



wall-e

@wall_e@ioc.exchange

2h*

@oliver @privateger @xmripper @Girgias @whitequark not to be mean, but what is so hard to understand about EOL?

Anyone still on ≤8.1 has basically explicitly been saying:"I accept the risk of *exactly this scenario*", for at least the past 10 months.

0



Androcat

@androcat@toot.cat

4h

@Girgias

I hear Apparmor or Selinux can stop the code execution part.

@whitequark

0



ComputerNutm43

@computernutm43@nexto.my

7h

@Girgias God Speed my friend (Is that the saying?) lol

0