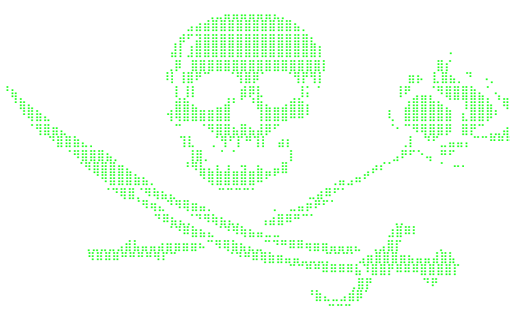




☆ Steal Everything ☆
☆ Leave Nothing ☆

EST. 2014-2026

PHP-UP-2026

PHP session.upload_progress UAF **CVE-2026-109618** / **CVE-2026-109619**

> [HOME] [OVERVIEW] [SOURCE] [MITIGATION] [USAGE] [DOWNLOAD]

phpup@C4b1Nkr3W:~	[active]
[*] Stage: CVE-2026-109618 / -109619 (PHP session.upload_progress heap UAF) [*] Author: J4ck3LSyN (ATLAS) [>] research: Andrew Dorman (@acd421) · Arch&Yuki (@ArchKnight23) [>] language: python · status: weaponized (lab)	

OVERVIEW

Two cached bucket pointers, one freed tree. A heap use-after-free in PHP session.upload_progress reaching zend_call_function.

PHP's session.upload_progress handler (php_session_rfc1867_callback in ext/session/session.c) caches two raw zval * bucket pointers into live HashTables: post_bytes_processed and current_file_bytes_processed. Both point into arrays that the request lifecycle frees under the attacker's feet. Press [SOURCE] above to walk the full bundle: the trigger request, both PoC chains, the extraction harness, and the validated per-build symbol deltas.

Stock configuration is enough. session.upload_progress.enabled=1 and the default name, prefix, freq=1%, min_freq=1. A single fast multipart POST arms both caches; a second update() re-reads the session file and zend_hash_update destroys the previous decoded tree through its HashTable destructor zval_ptr_dtor. The freed bucket is then reclaimed and the stale pointer lands an attacker-shaped zend_fcall_info / zend_internal_function that calls zend_call_function, i.e. system().

All of the following is verified against real php-src (8.4.4 and 8.2.34): every line reference in the original writeup, the ZVAL_COPY_VALUE no-addrf moves, the re-decode/track_init free, and the struct offsets (zend_fcall_info 64B with param_count at +48, zend_internal_function.handler at +88, IS_STRING_EX 0x106, IS_OBJECT_EX 0x308).

The framework in this post bakes in per-build delta extraction (--syms) because the offset between zval_ptr_dtor and zend_call_function changes with every build. Official LiteSpeed lsphp82 8.2.34 debug symbols give the exact values for the three distro sub-builds (bookworm 0xc930, noble 0xcef0, jammy 0xd060) and are included below.

PHP-UP-2026 (BUNDLE)/

Click directories to expand · click files to view content

php-up-2026/	tree
▼ 📁 php-up-2026 📄 php_up_framework.py 📄 poc0_109618.py 📄 poc1_109619.py 📄 poc_trigger.py 📄 DELTAS_LSpHP.md 📄 README_FRAMEWORK.md 📄 REPORT_VALIDATION.md — view select a file	
📄 README.md	
select a file from the tree to inspect source	

USAGE

How to Validate, Extract & Deliver the Chain

shell	[cmd]
\$ python3 php_up_framework.py --selftest # forge + body layout checks \$ python3 php_up_framework.py --syms --binary /path/to/lsphp # extract dtor / call / delta \$ python3 php_up_framework.py --build --bug 619 --dtor <leak> -o body.bin \$ python3 php_up_framework.py --lab --bug 618 --dtor <leak> # php-fpm lab delivery \$ python3 php_up_framework.py --live <target-url> --bug 619 --dtor <leak>	

MITIGATION (PRE-PATCH)

What to do before the official advisory ships. You can do all of this tonight.

1. Inventory your PHP

Know every host and version (php -v). Anything on PHP 7.x or older won't get a fix, so plan to upgrade.

2. Apply the stopgap (best-intel, low-risk, reversible)

In php.ini / .user.ini / .htaccess:

php.ini	[stopgap]
session.upload_progress.enabled = Off	

Reload PHP-FPM or your web server. My analysis says this closes one of the two likely paths.

3. Clamp multipart uploads at the edge

The second likely path lives in the multipart parser itself, which runs for any multipart/form-data POST regardless of the setting above. If you've got a WAF or reverse proxy, tighten inspection or rate-limit multipart POSTs to endpoints that don't need uploads. This is the same network-layer mitigation the enterprise vendors are quietly deploying for their paying customers.

4. Patch the second it drops

Current stable: 8.5.11 / 8.4.26 / 8.3.35 / 8.2.34. If your hosting is managed, confirm your provider's timeline today.

Why I'm posting this for free

A pre-patch mitigation shouldn't be a subscription perk. I'm a one man shop, and the orgs most likely to get burned in the gap between leak and patch are the small teams without an enterprise threat intel contract. The steps above cost nothing. Spend tonight reducing risk, not tomorrow doing incident response.

I'll follow up once the official advisory publishes and confirm whether my root cause read was right, together with [Andrew Dorman \(@acd421\)](#) and [Arch&Yuki \(@ArchKnight23\)](#).

DOWNLOAD

Where to get it.

Signed release from the usual place:

wget http://jackalsyn.com/posts/phpupAssets/php-up-2026.tar.gz

tar -xzf php-up-2026.tar.gz

hash(256) a29d640e0481ff64a39541faeeec0f998ecd18fb692816e75a6bcfa0f683017

Credits:

author @J4ck3LSyN

research Andrew Dorman (@acd421)

research Arch&Yuki (@ArchKnight23)

ver weaponized (dev)

