

eness

php2shell-CVE-2026-109618-CVE-2026-109619-PoC

Public

Notifications

Fork 86

Star 461

main

Go to file

Code

eness Merge pull request #3 from Ant...

e8db5ee · 2 hours ago

apache

feat: add affected vers...

9 hours ago

cloudflare-rule-fix

feat: add affected vers...

9 hours ago

frankenphp

feat: add affected vers...

9 hours ago

lighttpd

feat: add affected vers...

9 hours ago

patch

feat: add affected vers...

9 hours ago

quick-fix

fix: restore missing he...

4 hours ago

README.md

docs: add reference to...

2 hours ago

banner.jpg

docs: add banner ima...

9 hours ago

poc_109618.py

feat: add PoC exploit f...

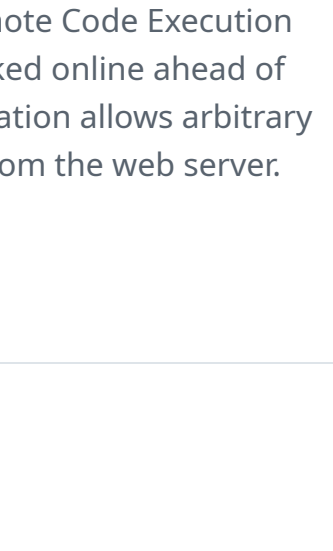
10 hours ago

README

THREAT UPDATE

Critical RCE vulnerabilities in PHP

CVE-2026-109618
CVE-2026-109619



php2shell — CVE-2026-109618 / CVE-2026-109619

Critical Advisory (CVSS v4.0: 9.2 - Critical): On October 20, details of two pre-auth Remote Code Execution (RCE) flaws in the PHP core leaked online ahead of coordinated disclosure. Exploitation allows arbitrary command execution straight from the web server.

Table of Contents

- Overview
 - Root Cause
 - Verification & Targets
- Affected PHP Versions
- Threat Landscape & Detection Guidance
 - Behavioral Indicators & Suspicious Child Processes
 - Monitoring Recommendation
- Action Items for Organizations
- Proof of Concept (PoC)
- Mitigation & Source Patching
 - 1. Configuration Mitigation (Immediate Stopgap)
 - 2. Automated Quick-Fix Script
 - 3. Source Code Patch (session-upload-progress-counters.patch)
 - 4. Cloudflare WAF Custom Rules
 - 5. FrankenPHP Caddyfile Configuration
 - 6. Lighttpd Configuration
 - 7. Apache Mitigation (ModSecurity & .htaccess)
- Disclaimer

Overview

php2shell refers to two critical vulnerabilities (CVE-2026-109618 & CVE-2026-109619) affecting core PHP functionality. The primary demonstrated vector is a heap use-after-free (UAF) in the PHP core session upload progress mechanism leading to arbitrary code execution (zend_call_function / system()).

- Severity:** Critical (CVSS v4.0: 9.2)
- Impacted Versions:** Reports indicate PHP versions 5, 6, 7, and 8 may be affected. Fixes are primarily expected for currently supported PHP 8 branches, leaving end-of-life (EOL) versions (PHP 5.x, 7.x, 8.0, 8.1) exposed without official vendor patches.
- Industry Tracking:** Confirmed by security researchers (watchTowr reproduced CVE-2026-109618; monitoring by Huntress and Florian Roth / process-watcher).

Root Cause

PHP's upload progress handler (php_session_rfc1867_callback in ext/session/session.c) stores two raw zval * bucket pointers (post_bytes_processed and current_file_bytes_processed) directly in live HashTable structures.

During request processing:

- Trigger:** A standard multipart POST request activates progress tracking with default configuration values (session.upload_progress.enabled = 1).
- Free:** A progress update cycle re-reads session data. The function zend_hash_update() destroys the previous data tree using the destructor zval_ptr_dtor , freeing the memory block.
- Reclaim:** The freed bucket memory is reclaimed. The dangling pointer is redirected to crafted zend_fcall_info and zend_internal_function structures.
- Execution:** PHP calls zend_call_function() , resulting in arbitrary system command execution.

Verification & Targets

- Confirmed on standard php-src builds (including versions 8.4.4 and 8.2.34).
- Uses symbol offsets (--syms) between zval_ptr_dtor and zend_call_function to adapt to distribution-specific binary layouts (e.g., Debian Bookworm, Ubuntu Noble, Ubuntu Jammy).

Affected PHP Versions

The following table summarizes the PHP versions affected by CVE-2026-109618 and CVE-2026-109619, along with their support and patch statuses:

PHP Branch	Status	Affected Versions	Fixed / Secure Version	Notes
PHP 8.5	Supported (Active)	< 8.5.12	8.5.12	Emergency published
PHP 8.4	Supported (Active)	< 8.4.27	8.4.27	Emergency published
PHP 8.3	Supported (Active)	< 8.3.36	8.3.36	Emergency published
PHP 8.2	Supported (Security Only)	< 8.2.35	8.2.35	Emergency published
PHP 8.1	End of Life (EOL)	All 8.1.x versions	No Official Patch (EOL)	Set session.upload_progress.enabled = off or rebuild PHP from source using the provided C source patch.
PHP 8.0	End of Life (EOL)	All 8.0.x versions	No Official Patch (EOL)	Official suconfigurator mandatory
PHP 7.x	End of Life (EOL)	All 7.0 - 7.4 versions	No Official Patch (EOL)	Core ext, vulnerability upgrade r
PHP 5.x	End of Life (EOL)	5.4.0+ (with RFC1867 progress tracking)	No Official Patch (EOL)	Official su or disable

Warning

PHP 8.1 and older End-of-Life (EOL) versions **do not** receive official security patches from the PHP Foundation. For these environments, immediately apply session.upload_progress.enabled = Off or rebuild PHP from source using the provided C source patch.

Threat Landscape & Detection Guidance

Exploitation of php2shell typically manifests as anomalous child processes spawned directly by PHP worker processes (php-fpm , php-cgi , php-cgi.exe , httpd , apache2).

Behavioral Indicators & Suspicious Child Processes

Monitor and alert on PHP worker processes executing:

- Interactive Shells:** /bin/sh , /bin/bash , cmd.exe , powershell.exe
- Script Interpreters:** python , perl
- Network / Transfer Utilities:** curl , wget , nc , certutil.exe
- Common Reverse Shell Patterns:** /dev/tcp / , fssockopen , mkfifo
- Immediate Reconnaissance Commands:** whoami , id , uname -a , ipconfig

Monitoring Recommendation

For Linux-based web environments without an endpoint detection engine, tools like Florian Roth's free process-watcher can be run in observe mode to flag unusual child process spawning in real time.

Action Items for Organizations

- Inventory All PHP Environments:**
 - Public-facing web servers running applications like WordPress, Drupal, Joomla, or Magento.
 - Internal appliances, hosting panels (cPanel, Plesk), and bundled standalone binaries (php.exe , php-cgi.exe).
- Prepare for Patch Windows:**
 - Official emergency patches from PHP and downstream Linux distributions are scheduled for release around **Wednesday, October 21 at 14:00 UTC (10:00 AM ET)**.
 - Apply vendor updates immediately once released.
- Isolate End-of-Life (EOL) Installations:**
 - Environments running unsupported versions (PHP 5.x, 7.x, 8.0, 8.1) may not receive official upstream backports.
 - Restrict access via VPN, apply strict WAF rules, or disable public internet exposure.
- Incident Response & Investigation:**
 - If suspicious child processes or unexpected files are detected, isolate the host and preserve web server/PHP logs.
 - Inspect web roots and temporary directories (/tmp , /var/tmp) for dropped webshells or malware execution artifacts.

Proof of Concept (PoC)

This repository is dedicated to security research and Proof-of-Concept (PoC) development for php2shell (CVE-2026-109618 / CVE-2026-109619).

The PoC exploit is currently being developed and tested. The code will be published in this repository very soon.

★ **Star this repository** to stay updated on the upcoming PoC release.

Mitigation & Source Patching

Until official vendor packages are updated in your repositories, you can protect your environments using configuration changes, automated scripts, or by compiling PHP with the included source patch.

1. Configuration Mitigation (Immediate Stopgap)

The vulnerability relies on PHP's session upload progress mechanism (session.upload_progress.enabled), which is enabled by default. Disabling this directive neutralizes the attack vector while preserving regular file upload functionality:

In your php.ini :

```
session.upload_progress.enabled = off
```

Or for PHP-FPM pool configurations (e.g., /etc/php/*/fpm/pool.d/www.conf):

```
php_admin_flag[session.upload_progress.enabled] = off
```

After modifying the configuration, reload/restart your web services (e.g. systemctl reload php-fpm or systemctl reload apache2).

2. Automated Quick-Fix Script

A mitigation helper script is available under the quick-fix/ directory:

- Check Current Status:**

```
sudo ./quick-fix/run.sh --check
```

- Apply Mitigation (Dry Run):**

```
sudo ./quick-fix/run.sh --dry-run
```

- Apply Mitigation:**

```
sudo ./quick-fix/run.sh
```

- Revert Changes:**

```
sudo ./quick-fix/run.sh --revert
```

3. Source Code Patch (session-upload-progress-counters.patch)

If you build PHP from source or maintain custom distribution packages, a C-level source patch is provided under patch/session-upload-progress-counters.patch .

What the patch fixes: The patch resolves the root cause in ext/session/session.c and ext/session/php_session.h . Instead of maintaining vulnerable, raw zval * bucket pointers (post_bytes_processed and current_file_bytes_processed) into dynamic HashTable structures (which becomes dangling pointers when the hash table is cleared or re-allocated during upload progress updates), the patch:

- Stores counters as direct zend_long primitives.
- Introduces php_session_rfc1867_sync_counters() to safely resolve and update array entries without dangling references.

How to Apply the Patch:

- Navigate to the root of your php-src source directory:

```
cd /path/to/php-src
```

- Test applying the patch (dry-run):

```
# Using git (if building from a git clone)
git apply --check /path/to/patch/session-upl

# OR using GNU patch
patch -p1 --dry-run < /path/to/patch/session-
```

- Apply the patch:

```
# Using git
git apply /path/to/patch/session-upload-progi

# OR using GNU patch
patch -p1 < /path/to/patch/session-upload-pri
```

- Recompile and install PHP:

```
./buildconf --force # if needed
./configure <your-options>
make -j$(nproc)
sudo make install
```

- Restart your PHP runtime / web services:

```
sudo systemctl restart php-fpm
# or
sudo systemctl restart apache2
```

Community Backports for EOL Versions (PHP 5.6, 7.4, 8.0, 8.1)

For legacy systems that cannot immediately upgrade to supported PHP 8.2+ releases (e.g. legacy OpenCart, WordPress, or control panels running on PHP 5.6, 7.4, 8.0, or 8.1), dedicated backports combining the official PHP Group OOM fix with the root-cause counter sync are available:

- 🔥 **Repository:** [Anton-Barinov/php2shell-patches](#)
- Includes automated build scripts to compile isolated, side-by-side runtimes into /opt/php-<ver>-patched without overwriting system packages, plus multi-PHP FPM pool hardening tools.

4. Cloudflare WAF Custom Rules

Recommended Cloudflare WAF rule templates to filter malicious traffic are located in the cloudflare-rule-fix/ directory:

About

Proof-of-Concept (PoC) exploit for php2shell (CVE-2026-109618 / CVE-2026-109619), demonstrating remote code execution in vulnerable PHP environments. For security research and educational purposes only.

- Readme
 - Activity
 - 28 stars
 - 28 watching
 - 86 forks
- Report repository

Releases

No releases published

Contributors 2

- eness enes sönmez
- test555

Languages



