

• SECURITY ADVISORY | SX-ADV-2026-1021

Critical PHP Flaws, Patch Lands Today

Two remote code execution vulnerabilities in PHP, tracked as [CVE-2026-109618](#) and [CVE-2026-109619](#) and nicknamed "php2shell", leaked before coordinated disclosure. The PHP security team rates both 9.2 under CVSS v4.0 and ships fixes on 21 October 2026 at 14:00 UTC.

SEVERITY

Critical

CVSS v4.0 9.2, both CVEs, rated by the PHP security team

FIX RELEASE

21-10-2026

14:00 UTC, 19:30 IST. PHP may release earlier

EXPLOITATION

Not observed

No in-the-wild activity reported as of 20-10-2026

VULNERABILITY

Remote code execution

Reported in PHP itself, not in one application

AFFECTED VERSIONS

Unconfirmed

Reported as PHP 5.x to 8.x. Official list ships with the fix

ADVISORY STATUS

Pre-patch, v1.0

ScruteX will reissue once fixed versions are published

SCRUTEX VERDICT

**Prepare now.
Patch within
24 hours.**

Treat both CVEs as critical for every internet-facing PHP system. Build the inventory today so the patch can go out within hours of the **14:00 UTC** release. Expect public exploit code soon after the fix ships, because patch diffs reveal the flaw. Plan for **end-of-life PHP (8.1 and older) to get no fix at all**: upgrade it or take it off the internet.

EXECUTIVE SUMMARY

A Runtime Flaw, Not One App

On 20 October 2026, details of two critical PHP vulnerabilities appeared on a public developer forum ahead of their planned disclosure. The post came down within hours, but screenshots spread. Independent researchers confirmed the flaws are real and said they reproduced CVE-2026-109618. The PHP security team then published an advance notice and moved the security release to **Wednesday 21 October 2026 at 14:00 UTC**, a day ahead of the original embargo.

Public reporting describes both as remote code execution in PHP's core. If that holds, an attacker who reaches a vulnerable server can make it run commands of their choosing. The bug sits in the language runtime, so every WordPress, Drupal, Joomla or Magento site, every hosting control panel and every appliance that bundles PHP is in scope until PHP narrows the affected range.

What is confirmed, reported and unknown

ITEM	STATUS	DETAIL
CVE IDs	CONFIRMED	CVE-2026-109618, CVE-2026-109619
Severity	CONFIRMED	Critical. CVSS v4.0 9.2 for each, per the PHP security team
Fix date	CONFIRMED	21-10-2026, 14:00 UTC. PHP will release earlier if details go public or exploitation appears
Flaw type	REPORTED	Remote code execution in PHP core. One CVE reproduced by independent researchers
Affected versions	UNKNOWN	Reports point to PHP 5.x through 8.x. PHP publishes affected and fixed versions with the release
Fix coverage	EXPECTED	Researchers expect fixes only for supported branches (8.2 to 8.5), leaving older branches unpatched
Attack path	UNKNOWN	No root cause published. Not yet known whether a plain web request reaches it, or which configuration it needs
Exploitation	NOT OBSERVED	None reported in the wild as of 20-10-2026. A short exploitation video was posted and removed
Public exploit	NONE KNOWN	No public proof of concept. Expected to follow quickly once the fix and its details are out
Workaround	UNCONFIRMED	Disable <code>session.upload_progress.enabled</code> as interim hardening (page 4)
Indicators	NONE YET	No confirmed indicators of compromise. Detection relies on behaviour (see page 7)

THE VULNERABILITIES

CVE-2026-109618 and CVE-2026-109619

Both flaws sit in PHP, the scripting language behind a large share of the web and the default on most hosting platforms. Here is what is known, and what is not, before the release.

What is known	The PHP security team rates both critical , CVSS v4.0 9.2 . Public reporting describes them as remote code execution in PHP's core: an attacker who reaches a vulnerable server could make it run commands of their choosing. Independent researchers say they reproduced CVE-2026-109618.
What it affects	Not yet published. Reports name PHP 5 through 8, and researchers expect fixes only for supported 8.x branches. One research team places the exposure at PHP 5.4 and later. PHP will publish affected and fixed versions with the release.
How it works	No root cause is public. It is not yet known which part of PHP is involved, whether a plain web request reaches it, or whether it needs a particular configuration.
Why "php2shell"	The informal name used by researchers. A name in that pattern usually means one request to a website returns a shell on the server behind it.

Official notice from the PHP security team

Upcoming Critical PHP Security Release »

20 Oct 2026

The PHP security team would like to give advance notice of an **upcoming security release**.

It is scheduled for Wednesday, 21 October 2026 at 14:00 UTC.

It addresses two critical vulnerabilities, both rated CVSS v4.0 9.2.

All PHP users are encouraged to prepare now so they can upgrade as soon as the release is available.

We reserve the right to release earlier should details of either vulnerability become public, or should evidence of active exploitation appear in the wild.

Full details, affected versions, and the fixed versions will be published with the release.

Thank you for helping us keep PHP secure.

Source: php.net news, "Upcoming Critical PHP Security Release", posted 20-10-2026. Captured 21-10-2026.

BEFORE THE PATCH

Buy Time Until 14:00 UTC

No setting is confirmed to block these flaws. The steps below shrink exposure while you wait for the fix, and they are safe to apply today.

Temporary workaround until the patch

NOT CONFIRMED BY PHP

```
; php.ini, or the PHP-FPM pool / vhost config for each PHP version
session.upload_progress.enabled = Off
session.use_strict_mode = 1           ; optional: reject session IDs the server never issued

# apply, then verify on every PHP version in use
systemctl restart php8.3-fpm        # or restart Apache when PHP runs as mod_php
php -i | grep -E 'upload_progress.enabled|use_strict_mode'
```

What it does

Upload progress tracking (on by default since PHP 5.4) writes client-supplied form data into the session store during uploads. PHP's own manual advises turning it off where no upload progress bar uses it. Turning it off removes that attacker-controlled write path.

How to apply and the trade-off

It is a per-directory setting: use `php.ini`, `.user.ini`, an FPM `php_admin_flag`, or the cPanel / Plesk PHP editor. `ini_set()` will not change it. The only side effect is that upload progress bars built on PHP sessions stop updating.

Status: one research team links this setting to the flaws. Another says no workaround exists before the patch. Apply it as low-risk hardening that buys time, not as protection. Patch at release regardless.

If the patch cannot go out tonight

Disable upload progress	Set <code>session.upload_progress.enabled = Off</code> (above). Low-risk, unconfirmed as a fix.
Restrict reach	Limit the site to known IP ranges or move admin and internal tools behind a VPN.
Virtual patching	Web application firewall vendors usually ship rules once technical details are public. Enable them, but do not treat them as the fix.
Close CGI exposure	Never expose <code>php-cgi</code> binaries directly through a <code>cgi-bin</code> path. Earlier PHP flaws such as CVE-2012-1823 and CVE-2024-4577 used that route.
Watch harder	Raise alert priority on PHP-spawned shells for these hosts until they are patched.

TIMELINE

Leak to Patch, Thirty Hours

- 20-10-2026, MORNING**
A post on a public developer forum claims two critical PHP flaws are heading for disclosure. It comes down, but screenshots circulate.
- 20-10-2026, WITHIN HOURS**
Independent researchers confirm the flaws are real and reproduce CVE-2026-109618. Coordinated disclosure had been set for Thursday 22-10-2026, 11:00 ET.
- 20-10-2026**
The PHP security team posts an advance notice: **two critical flaws, CVSS v4.0 9.2, fix on 21-10-2026 at 14:00 UTC**, with the right to release earlier.
- 21-10-2026, 08:30 IST**
ScruteX issues this advisory to clients and partners.
- 21-10-2026, 14:00 UTC**
Scheduled PHP security release with full details, affected versions and fixed versions.
- FROM 22-10-2026**
Technical write-ups and proof-of-concept code are likely. Unpatched internet-facing servers become the target.

INVENTORY

Where PHP Hides

Places teams forget

- **Control panels.** cPanel and Plesk run their own PHP builds beside the system PHP, often several versions at once.
- **Vendor appliances and Windows software** that bundle php.exe or php-cgi.exe: monitoring, helpdesk, VoIP and backup consoles.
- **Container images** pinned to old php: tags. These need a rebuild, not a package update.
- **Agency and marketing sites** on your domains that your own team does not patch.
- **Suppliers** who host your data on PHP stacks. Ask them for their patch plan today.

Quick checks

```
# Linux: version and installed packages
php -v ; php-fpm -v 2>/dev/null
dpkg -l | grep -i php    # Debian, Ubuntu
rpm -qa | grep -i php    # RHEL family
# Control panel builds
ls /opt/cpanel/ea-php*/root/usr/bin/php
ls /opt/plesk/php/*/bin/php
# Windows: bundled copies
where /r C:\ php.exe php-cgi.exe
# Containers
docker ps --format '{{.Image}}' | grep -i php
```

AFFECTED SCOPE

Which PHP Branches Get a Fix

Support status below is from the PHP project's published schedule. Whether each branch is vulnerable stays unconfirmed until the release. PHP 6 never shipped as a stable version, so reports naming "versions 5 through 8" cover the 5.x, 7.x and 8.x lines.

BRANCH	SUPPORT STATUS	LATEST BUILD	FIX EXPECTED	WHAT TO DO
8.5	Active support	8.5.11	YES	Patch on release
8.4	Active support	8.4.26	YES	Patch on release
8.3	Security fixes only	8.3.35	YES	Patch on release
8.2	Security fixes only, ends 31-12-2026	8.2.34	YES	Patch now, plan a move to 8.4 or 8.5
8.6	Release candidate, not for production	RC3	UNKNOWN	Keep out of production
8.1	End of life since 31-12-2025	No new releases	UNLIKELY	Upgrade to a supported branch. Until then, remove from the internet or put behind a VPN or web application firewall
8.0	End of life since 26-11-2023	No new releases	UNLIKELY	
7.x	End of life, 7.4 ended 28-11-2022	No new releases	UNLIKELY	
5.x	End of life, 5.6 ended 31-12-2018	No new releases	UNLIKELY	

Distribution packages. Red Hat, Debian, Ubuntu, SUSE, cPanel and Plesk ship their own PHP builds and backport fixes under their own version strings. Check your vendor's advisory for the patched package. Do not rely on the upstream version number alone. Some extended-support vendors may also patch end-of-life branches.

Fix release in your time zone, Wednesday 21 October 2026

UTC 14:00 Wed 21 Oct	US EASTERN 10:00 Wed 21 Oct	INDIA IST 19:30 Wed 21 Oct	THAILAND 21:00 Wed 21 Oct	MY / SG 22:00 Wed 21 Oct	SYDNEY AEDT 01:00 Thu 22 Oct
-----------------------------------	--	---	--	---------------------------------------	---

DETECTION

Watch What PHP Spawns

With no root cause public, no exploit signature exists yet. Detection rests on behaviour: after code execution, a PHP worker process starts programs it never runs in normal operation.

WATCH FOR	EXAMPLES	WHY IT MATTERS
Parent process	php-fpm, php-cgi, php-cgi.exe, httpd, apache2, w3wp.exe	The web or PHP worker that an exploit would control
Shell as child	sh, bash, dash, cmd.exe, powershell.exe	Normal PHP rarely needs a shell. Highest-value alert
Interpreter as child	python, perl	Common second stage after the first command
Download tools	curl, wget, nc, ncat, certutil.exe, bitsadmin.exe	Pulls a payload or webshell onto the host
Reverse shell strings	/dev/tcp/, fsockopen, mkfifo, bash -i, nc -e	Interactive access back to the attacker
Recon commands	whoami, id, uname -a, hostname, ipconfig	The first thing an operator types on a new shell
Write, run, delete	Short-lived .php file with random marker strings dropped in a temp folder, run, then removed. Prints !notwrtbl! if no folder is writable	Probe for writable paths, seen in recent PHP intrusions
Persistence	New .php files in web roots or upload folders, new cron jobs, scheduled tasks, local accounts	Survives the patch. Patching alone does not evict it

```
# Web-server process tree: look for shells or tools under PHP workers
ps -eo pid,ppid,user,etime,cmd --forest | grep -E -A3 'php-fpm|php-cgi|apache2|httpd'
# PHP files changed in the web root in the last 3 days
find /var/www -name '*.php' -mtime -3 -ls
# auditd: log every program the web account runs (www-data is 33 on Debian/Ubuntu, apache is 48 on RHEL)
auditctl -a always,exit -F arch=b64 -S execve -F euid=33 -k php-exec
# Windows (Sysmon Event ID 1): ParentImage ends php-cgi.exe or w3wp.exe AND Image is cmd.exe or powershell.exe
```

Indicators of compromise. None confirmed as of this issue. No exploit signature exists until the root cause is public. ScruteX will add indicators and detection logic to version 1.1 of this advisory once the PHP release and researcher analysis are out. Until then, alert on behaviour, not on hashes or IP addresses.

Baseline today, compare tomorrow

Record which child processes each PHP server starts in a normal day. After exploit code goes public, anything new under a PHP worker stands out against that record within minutes.

Linux web servers without endpoint tools

Most internet-facing PHP runs on Linux web and shared hosting servers with little monitoring. Turn on `auditd` process logging there first, since these hosts are where exploitation can go unnoticed.

RECOMMENDED ACTIONS

Thirteen Steps, Three Windows

PHASE 1

Today, before release

Now to 14:00 UTC

- 1 **Build the inventory.** List every PHP install, internet-facing first, including copies bundled in panels, appliances and containers.
- 2 **Flag end-of-life branches** (8.1 and older). For each, decide now: upgrade, take it off the internet, or put it behind a VPN or web application firewall.
- 3 **Apply the interim setting** `session.upload_progress.enabled = 0` on every PHP version, restart PHP, and confirm with `php -i` (page 4).
- 4 **Book the change window** for the release time in your zone and pre-approve it as an emergency change.
- 5 **Capture a baseline** of normal child processes on PHP servers and turn on process execution logging where it is missing. New behaviour stands out against it after exploits appear.
- 6 **Contact hosting providers, agencies and suppliers** running PHP for you. Ask for their patch time and get it in writing.

PHASE 2

At release

14:00 UTC onwards

- 7 **Read the official advisory** for affected and fixed versions, and the attack conditions. Re-rank your inventory against it.
- 8 **Patch internet-facing systems first**, then internal ones. Use your distribution or panel vendor's backported package where you do not build PHP yourself.
- 9 **Restart PHP-FPM and the web server.** An updated package does nothing until the running processes reload the new binary.
- 10 **Verify on every host** with `php -v` and the PHP-FPM version, then rescan from outside.

PHASE 3

After release

Next 24 to 72 hours

- 11 **Hunt back to 20-10-2026** with the detection table on page 7. The leak started the clock, not the patch.
- 12 **If you find something, isolate the server** and preserve web server and PHP logs before you change anything. Where you confirm a webshell, rebuild from a known-good image rather than cleaning in place.
- 13 **Track every end-of-life system to retirement** with an owner and a date. A compensating control is temporary, not a fix.

SELF-CHECK

Am I Exposed?

You likely have exposure if any of these is true. Each one takes minutes to check.

- ☐ You run any internet-facing PHP application, including WordPress, Drupal, Joomla or Magento.
- ☐ Any host, internal or external, still runs PHP 8.1 or older.
- ☐ You use cPanel, Plesk or a vendor appliance that ships its own PHP build.
- ☐ Your container images pin a PHP base tag that nobody has rebuilt in the last three months.
- ☐ An agency, reseller or supplier runs a PHP site on one of your domains and your team does not patch it.

WHAT SCRUTEX FINDS HERE

Your PHP, Seen From Outside

ScruteX fingerprints the PHP version your internet-facing hosts reveal through response headers and page technology signals, and flags end-of-life branches. It maps subdomains you may not track, including agency-run and legacy sites, so your inventory starts from what an attacker can reach. Once PHP publishes fixed versions, ScruteX matches observed versions against them and raises affected hosts as findings.

What ScruteX does not see: PHP on internal servers, hosts that hide their version, and copies bundled inside appliances. Those need your own inventory from page 5.

What a finding looks like

Web technology: end-of-life PHP

HIGH

SUBDOMAIN	OBSERVED VERSION	CURRENT SUPPORTED
portal.[redacted].com	PHP 7.4.33	PHP 8.5.11
BRANCH STATUS	EVIDENCE	STATUS
End of life since 28-11-2022	X-Powered-By: PHP/7.4.33	Open

Recommendation: Migrate to a supported branch. Until then, restrict the host to known networks. This branch is unlikely to receive a fix for CVE-2026-109618 or CVE-2026-109619.

Illustrative example in the ScruteX finding format. Host name and values are redacted and do not describe any client.

Which of your hosts run PHP, and which branch?

Clients: ask your ScruteX analyst for a PHP exposure list across your monitored assets today. Partners: send the domain, nothing else, and ScruteX returns the externally visible PHP hosts and versions.

scrutext.ai →

• ABOUT THIS ADVISORY

Exposure Found First, Fixed Before Exploited

ScruteX is a continuous threat exposure management platform. It watches your external attack surface, brand impersonation, vendor risk and dark web leaks, and tells you what to fix first. ScruteX Threat Intelligence issues advisories like this one when a flaw affects a large share of client estates.

Advisory details

Advisory ID: SX-ADV-2026-1021

CVEs: CVE-2026-109618, CVE-2026-109619

Version 1.0, issued 21-10-2026, 08:30 IST, before the fix release

Next update: once PHP publishes affected and fixed versions

Distribution: TLP: CLEAR, share freely

Primary sources

PHP security team, "Upcoming Critical PHP Security Release", 20-10-2026

php.net/archive/2026.php

PHP supported versions schedule
php.net/supported-versions.php

PHP downloads and changelogs
php.net/downloads.php

This advisory reflects information available at the time of issue. Details marked reported or unknown may change when the PHP security team publishes its release. Test patches in line with your change process before wide deployment. ScruteX provides this information for defensive use and accepts no liability for actions taken on it.