

PROTECT. DETECT. RESPOND.

Threat intelligence, risk assessment & mitigation guidance

THREAT INTEL CVE ANALYSIS INCIDENT RESPONSE RISK MITIGATION

PHP2Shell Vulnerability Gets a Public PoC Exploit for Security Testing

ADVISORY ID
CSA-2026-1138

CLASSIFICATION
INFORMATIONAL

DATE ISSUED
21-10-2026

SEVERITY LEVEL
Critical

ISSUED BY
Managed Security Service

VERSION 1.0

PHP2Shell Vulnerability Gets a Public PoC Exploit for Security Testing

Advisory ID	CSA-2026-1138
Date Issued	21 October 2026
Severity	Critical
Classification	Informational
Issued By	Managed Security Service
Distribution	IT Security Team Senior Management System Owners

1. Executive Summary

Two critical PHP vulnerabilities, tracked as **CVE-2026-109618** and **CVE-2026-109619**, have attracted attention following the publication of a public proof-of-concept (PoC) repository known as PHP2Shell. According to the supplied report, the vulnerabilities may enable remote code execution under specific conditions involving PHP session upload-progress handling, memory management, and application-level weaknesses.

The reported attack paths include session-file injection combined with local file inclusion (LFI), as well as a use-after-free memory flaw involving destructor re-entry. Additional research describes potential PHP-FPM serialization-state and deserialization-handler issues. These are research-reported paths and should not be interpreted as universally exploitable across every PHP deployment.

The report identifies PHP versions 5.4.0 through 8.5.11 as potentially affected and lists PHP 8.2.35, 8.3.36, 8.4.27, and 8.5.12 as planned fixed versions. However, at the time of the report, the scheduled PHP security release had not yet occurred, and these version details had not been confirmed by the full official PHP advisory.

ACTION REQUIRED: Organizations shall inventory PHP deployments across web servers, containers, PHP-FPM environments, and managed hosting platforms; assess exposure to the reported PHP2Shell vulnerabilities; and verify affected and patched versions against the official PHP security advisory before upgrading. Security teams shall review session handling, local file inclusion and unsafe deserialization risks, apply temporary mitigations where appropriate, and prioritize migration from unsupported PHP branches. Public PoC code shall only be tested in isolated, authorized environments

2. Threat Overview

Threat Category: Remote Code Execution (RCE), Session Handling Vulnerability, Memory Corruption / Use-After-Free
CVE ID: CVE-2026-109618, CVE-2026-109619
CVSS Score: 9.2
Severity: Critical
Attack Vector: Network
Privileges Required: None
User Interaction: None

Primary Target: Organizations or users running potentially vulnerable PHP versions (5.4.0–8.5.11), particularly internet-facing PHP-based websites, web applications, and PHP-FPM servers with session upload-progress enabled, local file inclusion (LFI) vulnerabilities, or unsafe deserialization behavior.

The principal concern is potential remote code execution when vulnerable PHP behaviour can be combined with specific application weaknesses or runtime conditions. The session-based path reportedly requires upload-progress functionality and an application-level local file inclusion weakness. Other described paths involve a use-after-free condition or unsafe deserialization behaviour and may require additional prerequisites.

The publication of a PoC increases the urgency of validating exposure but does not by itself confirm active exploitation or universal exploitability. The supplied report does not confirm active exploitation in the wild.

3. Technical Details

Key Behaviors

- **Session upload-progress handling:** The research repository claims that PHP's session upload-progress handler stores a supplied filename in session data without adequate validation. Under the reported conditions, this may contribute to session-file injection.
- **Session identifier handling:** The repository reports that a session identifier supplied in a POST body may be accepted despite `session.use_only_cookies` being configured to restrict session identifiers to cookies. This behaviour should be verified against the official advisory and the specific PHP configuration.
- **Local file inclusion chain:** One reported attack path combines session-file injection with an application-level LFI vulnerability. If the application can be induced to include the relevant session file, the chain may lead to PHP code execution. This requires the necessary configuration and application weakness; it is not a universal attack against all PHP sites.
- **Use-after-free condition:** The researcher describes a memory-management issue involving re-entry into a `var_destroy` destructor. The repository reports using AddressSanitizer to identify the memory error. The precise exploitability and conditions should be confirmed through official technical details.
- **PHP-FPM serialization state:** The research also describes a possible state-persistence issue across PHP-FPM requests and a handler mismatch that may cause a destructor to run after deserialization fails. Some reported paths may require an application gadget or access to a local file.

CVE-2026-109618 and CVE-2026-109619 are reported as critical PHP vulnerabilities associated with session handling and memory-management behaviour. The research repository describes multiple potential paths to code execution rather than one exploit that works against every PHP installation.

The PHP security team had announced an emergency release scheduled for 21 October 2026 at 14:00 UTC. At the time represented by the supplied article, the scheduled release had not yet occurred. The reported technical details, affected ranges, and planned fixed builds therefore require confirmation against the official PHP security announcement.

Infection Chain

PHP deployment with vulnerable behaviour → Attacker reaches an exposed application path → Session handling or memory-management weakness is triggered → Additional application/configuration prerequisites are met → Potential remote code execution.

For the session-based path, the reported sequence is:

Session upload-progress enabled → Session data manipulated → Application LFI weakness permits inclusion of a session file → Potential PHP code execution.

4. Affected Systems & Scope

The supplied report identifies PHP versions **5.4.0 through 8.5.11** as potentially affected according to the research repository. The complete affected-version mapping and final fixed releases were not yet confirmed by the official PHP security advisory at the time described in the article.

The primary targets for assessment are internet-facing PHP websites, business-critical web applications, PHP-FPM services, containerized PHP workloads, and hosting environments running potentially affected versions. Systems using upload-progress functionality alongside LFI weaknesses or unsafe deserialization behaviour warrant heightened priority.

The report cites research estimating that 883,717 internet-facing hosts were running PHP 8.1 or earlier, based on Shodan data collected on 20 October 2026. This figure is an external research estimate and does not mean that all identified hosts are vulnerable to these specific CVEs.

Product	Affected Version	Reported Planned Fixed Version
PHP 8.1.x and older supported -range candidate	Research repository broadly reports versions as far back as 5.4.0; exact per-CVE ranges require official confirmation	No official PHP project fix expected for end-of-life branches; migrate to a supported release
PHP 8.2.x	Reported affected through 8.2.34; verify exact range officially	8.2.35
PHP 8.3.x	Reported affected through 8.3.35; verify exact range officially	8.3.36
PHP 8.4.x	Reported affected through 8.4.26; verify exact range officially	8.4.27
PHP 8.5.x	Reported affected through 8.5.11	8.5.12

5. Recommendations

The following actions are recommended to detect, prevent, and respond to the identified threat:

- Recommended checking the PHP project's official security announcement and release notes for CVE-2026-109618 and CVE-2026-109619. Confirm the exact affected ranges and fixed builds before declaring a deployment remediated.
- Upgrade affected PHP deployments: Once the official patched builds are confirmed, upgrade to the appropriate fixed release within a supported branch. Test compatibility and restart PHP-FPM or the relevant PHP service after updating so the patched binaries are loaded.
- Prioritize unsupported versions: Identify PHP 8.1 and older deployments and develop a migration plan to a currently supported PHP branch. Where immediate migration is not possible, apply temporary exposure-reduction measures and document the remaining risk.
- Review session settings: Assess whether session.upload_progress is required. Where it is unnecessary, consider disabling it. Review session.use_only_cookies and apply the recommended session configuration from the official advisory. These measures may reduce exposure to specific paths but do not replace patching.
- Inspect application code: Review PHP applications for local file inclusion, unsafe deserialization, and user-controlled file-path handling. Correct application-level weaknesses and validate that request data cannot cause unintended server-side file inclusion.

- Review logs and investigate suspicious activity: Examine web-server, PHP-FPM, application, and security-monitoring logs for unusual upload-progress requests, unexpected session behaviour, suspicious file inclusion attempts, and abnormal PHP execution patterns. The supplied report does not provide specific malicious domains, URLs, or hashes for IOC-based detection.
- Test safely: Conduct validation only in isolated laboratory environments or authorized staging systems. Avoid executing public PoC code against production systems or third-party infrastructure.
- Apply defence in depth: Restrict unnecessary access to administrative interfaces and internal services, keep web-server and application components patched, and ensure the organization has monitoring and incident-response procedures for suspected server-side code execution.

6. Reference Links

The following sources were referenced in the preparation of this advisory:

- ▶ PHP2Shell Vulnerability Gets a Public PoC Exploit for Security Testing
[PHP2Shell Vulnerability Gets a Public PoC Exploit for Security Testing](#)
- ▶ PHP RCE in 8.5 and Earlier: CVE-2026-109618 and CVE-2026-109619
[PHP RCE in 8.5 and Earlier: CVE-2026-109618 and CVE-2026-109619 | Mallory](#)

DISCLAIMER: This document is issued by Paramount and is intended solely for the named recipients. The information contained herein is classified and must not be copied, forwarded, or disclosed to any third party without prior written authorization from Paramount Security. All recommendations are provided in good faith based on information available at the time of issue and may be subject to revision.
