

CISA urges immediate patching of critical PHP flaws CVE-2026-109618 and CVE-2026-109619



CISA urges immediate patching of two critical PHP flaws, known as "php2shell", that allow unauthenticated remote code execution. No exploitation had been observed at publication, but unpatched systems remain at risk.

- CVE-2026-109618 and CVE-2026-109619, named "php2shell", allow unauthenticated remote code execution.
- All PHP installations that serve web requests are affected.
- Fixed versions are PHP 8.5.12, 8.4.27, 8.3.36 and 8.2.35.
- CISA had not observed exploitation at the time of publication.
- PHP 8.1 and earlier are unsupported; systems that cannot be updated should be taken offline.

The US Cybersecurity and Infrastructure Security Agency (CISA) has called on organizations to patch two critical PHP vulnerabilities, CVE-2026-109618 and CVE-2026-109619, without delay. Together they are known as "php2shell" and allow an unauthenticated attacker to execute code remotely. They affect every PHP installation that serves web requests.

What is affected and what fixes it

The fixes are in PHP 8.5.12, 8.4.27, 8.3.36 and 8.2.35. CISA states that it had not seen these flaws exploited when it published the alert, but unpatched systems remain exposed. PHP 8.1 and earlier versions are unsupported.

CISA also notes that PHP is frequently embedded in other software, so its presence may not be obvious from a normal asset list.

Recommended response

CISA's guidance centres on finding PHP, patching it, and watching for misuse:

- Inventory all systems running any PHP version, including those where PHP is a bundled component.
- Patch immediately, starting with internet-facing systems.
- Take PHP-based systems offline if they cannot be updated promptly or run PHP 8.1 or earlier.
- Use a web application firewall as interim protection only, not as a substitute for patching.
- Watch for PHP worker processes launching shells or downloading tools.

Teams should also read the official PHP security advisory for patching details. For regulated institutions, the practical priority is a complete inventory, since hidden PHP components are the most likely gap.

What to do now

1. Inventory all systems running PHP, including products that bundle it.
2. Upgrade to PHP 8.5.12, 8.4.27, 8.3.36 or 8.2.35, starting with internet-facing systems.
3. Take PHP-based systems offline if they cannot be patched promptly or run PHP 8.1 or earlier.
4. Enable or tighten web application firewall rules as a temporary measure.
5. Alert security monitoring to PHP worker processes launching shells or downloading tools.
6. Record the exposure and remediation through your vulnerability and change management processes.