

PHP-Sicherheitslücke: Was jetzt zu tun ist

Für Website-Betreiber, Agenturen und Hoster
Allgemeine Ausgabe | Stand: 22. Oktober 2026

Autor: **Dominik Jedro** · justdigital.world

Priorität: PHP-Patchstatus jetzt bestätigen lassen. Die am 21. Oktober veröffentlichten Korrekturen schließen zwei kritische Lücken, die Codeausführung ohne Anmeldung ermöglichen können. [1]

1. Eine Person übernimmt die Koordination

Eine verantwortliche Person übernimmt die Koordination. Diese eröffnet beim Hoster ein dringendes Ticket, ruft bei ausbleibender Rückmeldung an und dokumentiert Ansprechpartner, Status und nächsten Rückmeldetermin.

2. Zuständigkeit und betroffene Systeme klären

Eigene Website oder eigener Server: Hoster bzw. Administration prüft alle PHP-Anwendungen, Domains, Subdomains, Testseiten und eingebetteten PHP-Komponenten. Auch Joomla, Drupal, Shops und eigene Anwendungen einbeziehen.

Website-Baukasten oder betreuter Dienst: Den Anbieter den PHP-Einsatz und die Behebung bestätigen lassen. **Nur Website-Besucher?** PHP wird auf dem Server aktualisiert; im Browser gibt es dafür keinen Patch zu installieren.

Einordnung: Diese allgemeine Anleitung enthält keine Prüfung einer konkreten Website. Ob eine Installation betroffen oder bereits gepatcht ist, muss ihr Betreiber feststellen.

WordPress- und Plugin-Updates ersetzen den PHP-Patch nicht: Angriffe können vor dem Anwendungscode greifen. Betroffen sind CVE-2026-109618 und CVE-2026-109619. [1]

3. Den passenden Sicherheitspatch installieren [1]

PHP-Zweig	Korrigierte Version*
8.2	8.2.35
8.3	8.3.36
8.4	8.4.27
8.5	8.5.12

* Oder eine spätere korrigierte Version desselben Zweigs. Ein vom Anbieter dokumentierter Backport für beide CVEs ist ebenfalls möglich. PHP 8.1 und älter: Anbieter-Nachweis oder Upgrade auf einen unterstützten Zweig verlangen. [1]

So behebt ihr die Lücke

Wählt den Weg, der zu eurem Hosting passt.

A. WordPress mit Webhosting oder Managed Hosting

1. PHP-Version ablesen: Im WordPress-Admin unter Werkzeuge > Website-Zustand > Bericht > Server nachsehen. Auf Englisch: Tools > Site Health > Info > Server. Notiert die vollständige PHP-Version. Diese Ansicht zeigt Informationen; sie installiert keinen PHP-Patch. [3]

2. Hoster prüfen lassen: Im Hosting-Konto die PHP-Einstellungen bzw. Versionsverwaltung öffnen. Der Anbieter muss den Patch für die betroffene Website bereitstellen. Die Auswahl „PHP 8.3“ allein bestätigt keinen Patchstand; genaue Version oder Anbieter-Backport nachweisen lassen.

3. Patch aktivieren: Nach Backup und Prüfung der Wiederherstellung die korrigierte Version einsetzen lassen. Wenn im Konto keine Patchauswahl möglich ist, Supportticket von Seite 4 senden. Nicht vorschnell auf einen anderen PHP-Zweig wechseln.

4. Ergebnis prüfen: PHP-Version erneut ablesen, den aktiven Patch vom Hoster bestätigen lassen und Funktionen gemäß Seite 3 testen. WordPress, Themes und Plugins ebenfalls aktuell halten; das ersetzt die PHP-Korrektur nicht. [1]

B. Eigener PHP-Server, VPS oder Container

Linux-Pakete: Distribution, Paketquelle und den tatsächlich verwendeten Web-PHP-Dienst feststellen. Die korrigierten PHP-Pakete aus einer gepflegten Quelle installieren; Erweiterungen passend halten. Anbieterhinweise auf beide CVEs prüfen. Konfiguration prüfen und die betroffenen FPM-/Webserver-Prozesse neu starten. Nur das CLI-Paket zu aktualisieren reicht nicht.

Container: Ein nachweislich korrigiertes Basis-Image beziehen, Anwendungs-Image bei Bedarf neu bauen und alle laufenden Instanzen ersetzen. Ein geändertes Image-Tag allein aktualisiert keine laufenden Container.

Windows oder Quellcode-Build: Passenden korrigierten Hersteller-Build bzw. gepatchten Quellcode mit kompatiblen Erweiterungen bereitstellen und die betroffenen Dienste neu starten. Die genauen Befehle müssen zu Betriebssystem, Installation und Dienst passen.

Wenn ein Update scheitert: Zugriff auf das verwundbare System blockieren und die Wiederherstellung mit dem Betreiber abstimmen. Keine ungepatchte Version wieder öffentlich betreiben.

Checkliste für den Hoster

Ziel: Der Patch ist in jeder betroffenen Web-Umgebung aktiv.

Vor und während des Updates

- ☐ Je Domain bzw. Umgebung PHP-Version, Anbieterpaket und verantwortlichen Betreiber erfassen. Wenn kein PHP eingesetzt wird, dies ausdrücklich dokumentieren.
- ☐ Aktuelles Backup von Dateien und Datenbank sowie Wiederherstellbarkeit prüfen. Bei Verdacht auf einen Einbruch vorhandene Logs und Beweise sichern.
- ☐ Sicherheitspatch möglichst innerhalb des bestehenden unterstützten PHP-Zweigs einspielen. Einen unnötigen großen Versionswechsel im Notfall vermeiden; bei veralteten Zweigen Kompatibilität gezielt prüfen.
- ☐ Alle betroffenen Instanzen aktualisieren, einschließlich mehrerer Server, Container und separater PHP-Pools. PHP-Dienste kontrolliert neu starten bzw. neu bereitstellen.
- ☐ Version im tatsächlich verwendeten Web-PHP prüfen. Ein `php -v` auf der Kommandozeile allein belegt den Patchstatus der Website nicht. Keine öffentliche phpinfo-Seite zurücklassen.

Wenn ein sofortiges Update nicht möglich ist

Mit dem Betreiber eine vorübergehende Isolation der betroffenen Anwendung vom öffentlichen Internet veranlassen, bis der Patch installiert und geprüft ist. CERT-EU empfiehlt, diese Maßnahme zu erwägen. [2]

Bei einer statischen Wartungsseite muss der Zugriff auf das ungepatchte PHP-System technisch blockiert bleiben. Ein WordPress-Wartungsplugin oder eine Firewall-Regel allein ist kein Beleg für vollständigen Schutz. [1,2]

Nach dem Update: Betreiber und Technik gemeinsam

- ☐ Startseite, wichtige Unterseiten, Formulare inklusive E-Mail-Zustellung, Anmeldung und gegebenenfalls Kaufabschluss testen. Jede Anwendung separat prüfen.
- ☐ Logs, unbekannte Administratoren, geplante Aufgaben und veränderte Dateien auf Auffälligkeiten prüfen lassen. Ein Patch entfernt keinen bereits eingeschleusten Schadcode.
- ☐ Bei konkreten Einbruchshinweisen: Anwendung isolieren, Beweise erhalten und Incident-Response-Fachleute hinzuziehen.

Nachricht an den Hoster

Kopiervorlage für ein dringendes Supportticket.

Betreff: Dringend - PHP-Patchstatus für unsere Websites bestätigen

Hallo zusammen,

bitte prüft umgehend unsere Hosting-Umgebungen für [Domains / Anwendungen eintragen] einschließlich Subdomains und Testseiten auf CVE-2026-109618 und CVE-2026-109619.

Bitte bestätigt je Umgebung:

1. Wird PHP eingesetzt, und welche genaue Web-PHP-Version bzw. welches Anbieterpaket ist aktiv?
2. Sind beide Lücken durch die Sicherheitsupdates vom 21. Oktober 2026 oder einen dokumentierten Anbieter-Backport behoben?
3. Wann wurde der Patch aktiv, und wurden alle PHP-Prozesse bzw. Instanzen neu gestartet oder neu bereitgestellt?
4. Gibt es Auffälligkeiten in Logs oder Dateien? Welcher Zeitraum wurde geprüft?

Bitte nennt bei einem noch offenen Patch einen verbindlichen Umsetzungstermin und schlägt eine vorübergehende Isolation der betroffenen Anwendung vor. Nennt außerdem alle Umgebungen, die ihr nicht betreut, damit wir diese separat klären können.

Bitte antwortet mit einem technischen Nachweis des aktiven Patchstands je Umgebung und einer zuständigen Kontaktperson. Vielen Dank.

Erledigt, wenn ...

alle Umgebungen erfasst sind, der aktive Patchstand bzw. die Nichtbetroffenheit belegt ist, die Funktionstests bestanden sind und Auffälligkeiten bewertet wurden. Eine Rückmeldung wie „WordPress ist aktuell“ genügt dafür nicht.

Verantwortlich: _____ Ticket: _____
Abgeschlossen am / durch: _____

Quellen und Einordnung

[1] PHP Project: Security advisory, 21 October 2026.

<https://www.php.net/>

[2] CERT-EU: Security Advisory 2026-018.

<https://cert.europa.eu/publications/security-advisories/2026-018/>

[3] WordPress.org: Site Health screen.

<https://wordpress.org/documentation/article/site-health-screen/>

Quellenstand: 22.10.2026. Allgemeine Handlungsempfehlung, keine Sicherheitsprüfung. Die tatsächliche Behebung muss der Betreiber nachweisen. Bei späterer Verwendung die aktuellen Herstellerhinweise prüfen.

Prompt für Claude / GPT

Kopieren, Platzhalter ausfüllen und in Claude oder ChatGPT einfügen.

Ohne verbundene Hosting- oder Serverwerkzeuge kann die KI anleiten, aber den Server nicht selbst aktualisieren. Ein WordPress-MCP-Zugang allein gibt normalerweise keinen Zugriff auf PHP-Pakete. Zugangsdaten nur über sichere Verbindungen hinterlegen, niemals im Chat posten.

Hilf mir, meine eigene WordPress-/PHP-Installation gegen CVE-2026-109618 und CVE-2026-109619 abzusichern. Erkläre die Schritte verständlich und führe mich bis zur überprüften Behebung.

Website(s): [Domains]

Anwendung: [WordPress / andere / unbekannt]

Hosting und Betriebssystem: [Anbieter / System / unbekannt]

Zugriff: [WordPress-Admin / Hosting-Panel / SSH / Container / MCP / keiner]

Backupstatus: [Datum / unbekannt]

1. Prüfe zuerst aktuelle Primärquellen: <https://www.php.net/> und <https://cert.europa.eu/publications/security-advisories/2026-018/> sowie Hinweise meines Anbieters. Stand 22.10.2026: korrigiert sind PHP 8.2.35, 8.3.36, 8.4.27 und 8.5.12 oder spätere korrigierte Versionen desselben Zweigs. Akzeptiere dokumentierte Anbieter-Backports für beide CVEs. Falls du nicht browsen kannst, sage das und bitte um den aktuellen Anbieterhinweis.

2. Beginne nur lesend. Frage nach fehlenden Angaben und nutze nur tatsächlich verfügbare, autorisierte Werkzeuge. Ermittle Web-PHP-Version, Installationsweg und alle betroffenen Umgebungen. Unterscheide CLI-PHP von Web-PHP. WordPress: Werkzeuge > Website-Zustand > Bericht > Server. Ohne Zugriff formuliere ein konkretes Hoster-Ticket. Behaupte keine durchgeführten Prüfungen.

3. Lege einen kurzen Patchplan mit Backup, Wiederherstellung und Funktionsprüfung vor. Zeige genaue, zur Installation passende Befehle oder Panel-Schritte und ihre Wirkung. Bitte vor produktiven Änderungen oder Ausfallzeiten um meine Freigabe. Bevorzuge den Patch im bestehenden unterstützten Zweig; plane bei alten Zweigen einen kompatiblen Wechsel oder belegten Anbieterfix.

4. Nach Freigabe: Aktualisiere nur im vereinbarten Umfang, starte betroffene Dienste neu bzw. ersetze Container und prüfe alle Instanzen. Behaupte nicht, dass Plugin-Updates, WAF-Regeln oder ein Wartungsplugin die PHP-Lücken beheben. Nutze keine Exploit-Tests und keine pauschalen Speicherlimit-Tricks. Frage nie nach Passwörtern, API-Schlüsseln oder vollständigen Konfigurationsdateien im Chat.

5. Weise den aktiven Patch im Web-PHP nach. Teste Seiten, Formulare, E-Mail-Zustellung, Anmeldung und gegebenenfalls Checkout; vereinbare vor echten Sendungen oder Bestellungen einen Test. Prüfe Logs und Dateien auf Hinweise eines Einbruchs. Bei Verdacht: Beweise erhalten und Fachleute hinzuziehen.

6. Wenn Patchen nicht sofort geht, schlage kontrollierte Isolation vom öffentlichen Internet vor. Melde abschließend je Umgebung: vorher/nachher, Nachweis, Tests, offene Punkte und Zuständigkeit. Status nur „bestätigt gepatcht“, „belegt nicht betroffen“ oder „ungeklärt“. Eine öffentliche Website oder grüne WordPress-Anzeige ist kein Patchnachweis.