

HÄNDLER, ANBIETER UND ISOS

PHP2Shell: was die neuen PHP-Schwachstellen für alle bedeuten, die Kartenzahlungen annehmen

Veröffentlicht am 22. Oktober 2026

Zwei kritische Schwachstellen in PHP können es einem nicht authentifizierten Angreifer ermöglichen, Code auf einem Webserver auszuführen. Für Händler ist das ein Risiko für die Integrität des Checkouts, für Zahlungsdaten und für die Verfügbarkeit. Für Anbieter ist es ein Portfoliorisiko. Was passiert ist, was zu prüfen ist und was Sie Ihren Hoster fragen sollten.

Am 21. Oktober 2026 hat das PHP-Projekt Notfall-Korrekturen für zwei kritische Schwachstellen in der Sprache selbst veröffentlicht. Sie sind als CVE-2026-109618 und CVE-2026-109619 erfasst und tragen den Spitznamen **PHP2Shell**. Beide werden mit 9,2 nach CVSS 4.0 bewertet. Jede von ihnen kann es einem nicht authentifizierten Angreifer ermöglichen, Code auf einem betroffenen Webserver auszuführen. Die Schwachstellen liegen in PHP, nicht in einer bestimmten Anwendung, die darauf aufbaut.

Für Händler entsteht damit ein Risiko für die Integrität des Checkouts, für Zahlungsdaten und für die Verfügbarkeit. Für Zahlungsanbieter und ISOs entsteht eine mögliche Gefährdung über ganze Händlerportfolios hinweg. Hier steht, was passiert ist, was Sie prüfen sollten und was Sie Ihren Hosting-Anbieter fragen sollten.

PHP läuft auf einem sehr großen Teil des Webs, auch auf den meisten Shop-Systemen, die kleine und mittlere Händler einsetzen. Das macht dies ebenso zu einer Zahlungsgeschichte wie zu einer Sicherheitsgeschichte.

Was entdeckt wurde

- **CVE-2026-109618** ist ein Speicherbeschädigungsfehler. Geht PHP mitten in der Vorbereitung einer Anfrage der Arbeitsspeicher aus, kann die anschließende Bereinigung halb aufgebaute Daten als gültig behandeln.

- **CVE-2026-109619** ist ein Fehler mit veraltetem Zustand, ein Use-after-free. PHP kann einen internen Zustand behalten, der auf bereits freigegebenen Anfragespeicher zeigt, und spätere Verarbeitung kann ihn verwenden. PHP nennt als Folgen die Ausführung von Code aus der Ferne und die Offenlegung sensibler Speicherinhalte.
- **Reichweite.** Der früheste betroffene Code reicht bis PHP 5.4 zurück. Das PHP-Projekt erklärt, dass jede Installation, die Webanfragen bedient, gefährdet ist, egal ob mit PHP-FPM, Apache mit mod_php, CGI, FrankenPHP oder LiteSpeed. PHP schätzt das Risiko bei mod_php am höchsten ein, weil der gezeigte Angriff dort greifen kann, bevor irgendein Code der Website selbst läuft. Das beschreibt den vorgeführten Angriff und ist keine Aussage, dass andere Konfigurationen sicher wären.
- **Korrigierte Versionen.** PHP 8.5.12, 8.4.27, 8.3.36 und 8.2.35. Für die Versionen 7.4 bis 8.1 gibt es keine offizielle Veröffentlichung, nur minimale Backports im Quellcode. Die Versionen 5.4 bis 7.3 haben das Support-Ende erreicht und werden vom PHP-Projekt gar nicht mehr korrigiert. Einige Linux-Distributionen und Anbieter mit Langzeit-Support portieren Korrekturen auf ältere Zweige zurück. Prüfen Sie daher die Sicherheitshinweise Ihres eigenen Anbieters, bevor Sie annehmen, eine bestimmte Installation sei ungepatcht.

Wo die Dinge stehen

Die Einzelheiten wurden am 20. Oktober bekannt, einen Tag vor der geplanten Veröffentlichung, die ihrerseits schon vom 22. Oktober vorgezogen worden war. Das Bedrohungsanalyse-Team von Amazon meldete am 17. Oktober, vier Tage vor der öffentlichen Veröffentlichung, einen Versuch gegen seine Honeypots und seither weitere, offenbar von mehreren Akteuren. Cloudflare und watchTowr berichten von Angriffsverkehr, und watchTowr gibt an, beide Schwachstellen nachgestellt zu haben. Öffentlicher Exploit-Code existiert, seine Zuverlässigkeit ist allerdings ungeprüft, und das PHP-Projekt erklärte bei der Veröffentlichung, es habe keinen funktionierenden Exploit identifiziert. Ob die bisher beobachteten Versuche gegen echte Produktsysteme Erfolg hätten, ist noch nicht geklärt.

Das ist eine glaubwürdige und dringende Bedrohung. Angriffsversuche und Proof-of-Concept-Code zeigen für sich genommen nicht, dass Zahlungsdaten von Händlern gestohlen wurden, und uns ist zum Stand 22. Oktober keine bestätigte Kompromittierung von Zahlungsdaten bekannt, die sich PHP2Shell zuordnen ließe. Patchen Sie betroffene Systeme jetzt, untersuchen Sie alles Verdächtige und bewahren Sie die relevanten Protokolle auf. Dass es keinen bestätigten Vorfall gibt, ist kein Grund zu warten.

Warum das für Kartenzahlungen wichtig ist

Ein Händler muss keinen eigenen PHP-Code betreiben, um betroffen zu sein. Die Software unter dem Shop genügt, und die Folgen erreichen die Zahlungsseite auf vier Wegen.

1. **Der Server, der die Checkout-Seite aufbaut.** Wer dort Code ausführen kann, kann verändern, was die Checkout-Seite tut. Ein Skript einzuschleusen, das Kartendaten beim Eintippen abgreift (oft Skimming oder Magecart-Angriff genannt), ist der altbekannte Weg, auf dem gestohlene Kartendaten PHP-Shops verlassen.
2. **Zugangsdaten und Schlüssel.** API-Schlüssel des Zahlungs-Gateways, Webhook-Geheimnisse und Admin-Zugänge liegen meist in der Konfiguration des Shops. Wer auf dem Server Code ausführen kann, kann sie lesen.
3. **Betrug und Missbrauch.** Gestohlene Schlüssel lassen sich nutzen, um Erstattungen auszulösen oder Testtransaktionen durchzuführen. Eine kompromittierte Website kann außerdem einen zweiten, versteckten Checkout hosten, der für den Anbieter dem **Transaction Laundering** sehr ähnlich aussieht.
4. **Ausfallzeit.** Selbst ein erfolgloser Angriff kann Worker-Prozesse abstürzen lassen. Ein Shop, der nicht erreichbar ist, nimmt keine Bestellungen an, und eine Wiederherstellung, die Tage dauert, kann Abrechnungszyklen von **Abonnements** stören und Erstattungen und Streitfälle nach sich ziehen.

Händler, die auf die gehostete Zahlungsseite des Anbieters weiterleiten oder die Felder des Anbieters einbetten, stehen besser da: Die Kartennummer liegt nie auf ihrem Server. Das oben beschriebene Risiko bei Admin-Zugang, Erstattungen und API-Schlüsseln bleibt aber bestehen.

Was diese Woche zu prüfen ist

- **Finden Sie heraus, welches PHP Sie einsetzen.** Auf Ihrem eigenen Server führen Sie `php -v` aus. Bei verwaltetem oder geteiltem Hosting schauen Sie im Kundenbereich nach oder fragen den Hoster. Plugins und Shop-Software melden das nicht, und ein Shop kann noch Jahre nach dem Update der Software auf einer alten PHP-Version laufen.
- **Spiele Sie die Korrektur des Herstellers ein.** Für die meisten Linux-Distributionen und Hoster bedeutet das das Paket-Update des Anbieters, gefolgt von einem Neustart von PHP, damit die Änderung wirkt. Verlassen Sie sich nicht allein auf die PHP-Versionsnummer, denn Distributionen portieren die Korrektur oft zurück, ohne sie zu ändern.
- **Wenn Sie PHP 7.3 oder älter einsetzen, planen Sie den Umstieg.** Das PHP-Projekt wird diese Versionen nicht korrigieren. Fragen Sie Ihren Hoster oder Entwickler, ob deren Anbieter eine Korrektur zurückportiert, und lassen Sie sich das Datum nennen, nicht die Absicht.
- **Gegenmaßnahmen sind vorübergehend.** Das PHP-Projekt dokumentiert Einstellungen, die jede Schwachstelle eingrenzen, und eine Web Application Firewall oder Netzwerkbeschränkungen können die Angriffsfläche verringern, während Sie ein Update vorbereiten. Keine davon ist eine vollständige Lösung. Wechseln Sie so bald wie möglich auf eine korrigierte Version oder spielen Sie den geprüften Sicherheits-Patch Ihrer Distribution ein, und wenn ein aus dem Internet erreichbares System nicht zügig gepatcht werden kann, erwägen Sie, seine Erreichbarkeit einzuschränken.

- **Tauschen Sie Geheimnisse aus, wenn Sie eine Kompromittierung nicht ausschließen können.** Schlüssel des Zahlungs-Gateways, Webhook-Geheimnisse, Admin-Passwörter und Datenbank-Zugangsdaten. Suchen Sie nach neuen Admin-Benutzern, veränderten Checkout-Dateien und unerwarteten Skripten auf der Zahlungsseite.
- **Sehen Sie sich den Checkout an.** Vergleichen Sie die Skripte, die die Zahlungsseite lädt, mit dem, was Sie erwarten. Unbekannte Drittanbieter-Domains sind das Warnzeichen.

PCI DSS und Ihr Anbieter

Wenn Sie in den Geltungsbereich von PCI DSS fallen, gehört das zu Ihren bestehenden Pflichten im Schwachstellenmanagement. Anforderung 6.3.1 verlangt, dass Sie neue Schwachstellen identifizieren und mit Risikostufen versehen. Anforderung 6.3.3 verlangt, dass kritische oder hochgradig sicherheitsrelevante Patches entsprechend dieser Einstufung innerhalb eines Monats nach Veröffentlichung eingespielt werden.

Das ist eine Höchstfrist, keine Empfehlung zu warten. Angesichts der Schwere von PHP2Shell und der Berichte über Ausnutzungsaktivität sollten betroffene, aus dem Internet erreichbare Systeme sofort behoben werden. Liegt die Checkout-Seite im Geltungsbereich, können auch die Anforderungen an Skripte auf Zahlungsseiten relevant sein.

Ein Händler, der nicht belegen kann, dass er zügig gepatcht hat, und keine Aufzeichnung der Gründe besitzt, steht schlecht da, wenn etwas schiefgeht. Eine kurze schriftliche Notiz, wann die Schwachstelle erkannt wurde, was getan wurde und wann, ist für den Anfang ausreichend.

Informieren Sie Ihren Anbieter früh, wenn Sie Grund zur Annahme haben, dass der Shop kompromittiert wurde. Zahlungsanbieter können nach den geltenden Regeln der Kartenorganisationen Melde- und Eskalationspflichten haben, und ein Händler, der das am ersten Tag anspricht, wird ganz anders behandelt als einer, den der Anbieter über einen Alarm der Kartenorganisation oder eine Serie von Betrugsmeldungen findet.

Für Anbieter und ISOs

Die Gefährdung betrifft Ihr gesamtes Portfolio und ist nicht gleichmäßig verteilt. Händler mit PHP am Lebensende, mit günstigem geteiltem Hosting und mit selbst verwalteter Shop-Software sind die ersten, die man sich ansehen sollte. Angemessene Schritte sind:

- Händler mit selbst gehosteter Shop-Software mit einem kurzen, klaren Hinweis kontaktieren und um das Datum bitten, an dem sie gepatcht haben.
- Auf die Anzeichen eines kompromittierten Checkouts achten: ein plötzlicher Anstieg von Card-Testing, unerwartete Ablehnungen, Erstattungen, die nicht zu Bestellungen passen, oder Streitfälle, die Betrug mit Karten melden, die nur bei diesem Händler verwendet wurden.
- Das Website-Monitoring für die umsatzstärksten Händler erneut laufen lassen und nach neuen Skripten oder unbekannten Zahlungszielen suchen.

- Bedenken, dass dieselbe Schwachstelle Ihre eigenen Anwendungen und die Websites Ihrer Partner und Affiliates betrifft.

Beim Onboarding ist eine nützliche Frage, welche Plattform und welches Hosting der Händler einsetzt und wer dafür verantwortlich ist, sie aktuell zu halten. Die Antwort sagt viel darüber aus, wie sich ein Konto in einer Woche wie dieser verhalten wird.

Quellen

- [Tenable: PHP2Shell \(CVE-2026-109618, CVE-2026-109619\) frequently asked questions](#)
- [Qualys: PHP announces critical security release for two vulnerabilities](#)
- [Cisco Talos: Two critical RCEs in PHP core](#)
- [CERT-EU: Critical vulnerabilities in PHP \(2026-018\)](#)
- [watchTowr: PHP remote code execution exploited](#)

Online lesen: <https://payments-marketplace.com/de-DE/insights/php2shell-payments/>

Payments Marketplace Ltd

Registered in England & Wales No. 17424288

Garden Ctg, Dereham Road, Dereham, Norfolk, NR19 2PW, UK

connect@payments-marketplace.com · +44 1366 727140