

MERCHANTS, PROVIDERS EN ISO'S

PHP2Shell: wat de nieuwe PHP-kwetsbaarheden betekenen voor iedereen die kaartbetalingen accepteert

Gepubliceerd op 22 oktober 2026

Twee kritieke kwetsbaarheden in PHP kunnen een niet-geauthenticeerde aanvaller code laten uitvoeren op een webserver. Voor een merchant is dat een risico voor de integriteit van de checkout, voor betaalgegevens en voor de beschikbaarheid. Voor een provider is het een portefeuille risico. Wat er is gebeurd, wat u moet controleren en wat u uw hostingpartij kunt vragen.

Op 21 oktober 2026 bracht het PHP-project noodpatches uit voor twee kritieke kwetsbaarheden in de taal zelf, geregistreerd als CVE-2026-109618 en CVE-2026-109619 en bijgenaamd **PHP2Shell**. Beide krijgen 9,2 op CVSS 4.0. Elk van de twee kan een niet-geauthenticeerde aanvaller code laten uitvoeren op een getroffen webserver, en de kwetsbaarheden zitten in PHP, niet in één specifieke applicatie die erop is gebouwd.

Voor merchants ontstaat daardoor een risico voor de integriteit van de checkout, voor betaalgegevens en voor de beschikbaarheid. Voor betaalproviders en ISO's ontstaat mogelijke blootstelling over hun hele merchantportefeuille. Hier staat wat er is gebeurd, wat u moet controleren en wat u uw hostingpartij kunt vragen.

PHP draait een zeer groot deel van het web, inclusief de meeste webshopsoftware die kleine en middelgrote merchants gebruiken. Dat maakt dit net zozeer een betaalverhaal als een beveiligingsverhaal.

Wat is gevonden

- **CVE-2026-109618** is een geheugenbeschadigingsfout. Als PHP halverwege het opzetten van een verzoek zonder geheugen komt te zitten, kan de opruiming daarna half opgebouwde gegevens als geldig behandelen.
- **CVE-2026-109619** is een fout met verouderde status, een use-after-free. PHP kan een interne status bewaren die verwijst naar verzoekgegevens dat al is vrijgegeven, en latere verwerking kan

die gebruiken. PHP noemt uitvoering van code op afstand en het vrijkomen van gevoelig geheugen als gevolgen.

- **Bereik.** De oudste getroffen code gaat terug tot PHP 5.4. Het PHP-project stelt dat elke installatie die webverzoeken afhandelt risico loopt, of die nu PHP-FPM, Apache met mod_php, CGI, FrankenPHP of LiteSpeed draait. PHP beoordeelt het risico als het hoogst bij mod_php, waar de gedemonstreerde aanval kan toeslaan voordat enige code van de site zelf draait. Dat beschrijft de aanval die is getoond en is geen bewering dat andere opstellingen veilig zijn.
- **Gerepareerde versies.** PHP 8.5.12, 8.4.27, 8.3.36 en 8.2.35. Voor de versies 7.4 tot en met 8.1 is er geen officiële release, alleen minimale backports van de broncode. De versies 5.4 tot en met 7.3 zijn end of life en worden door het PHP-project helemaal niet meer gerepareerd. Sommige Linux-distributies en leveranciers van langetermijnondersteuning backporten wel fixes naar oudere takken. Raadpleeg dus het beveiligingsadvies van uw eigen leverancier voordat u aanneemt dat een bepaalde installatie niet is gepatcht.

Waar we nu staan

De details lekten op 20 oktober uit, een dag voor de geplande release, die zelf al was vervroegd van 22 oktober. Het threat-intelligenceteam van Amazon meldde op 17 oktober, vier dagen voor de openbare release, een poging tegen zijn honeypots en sindsdien meer, kennelijk van meerdere partijen. Cloudflare en watchTowr melden aanvalsverkeer, en watchTowr zegt beide kwetsbaarheden te hebben nagebootst. Openbare exploitcode bestaat, al is de betrouwbaarheid ervan niet geverifieerd, en het PHP-project zei bij de release dat het geen werkende exploit had geïdentificeerd. Of de tot nu toe geziene pogingen zouden slagen tegen echte productiesystemen, is nog niet vastgesteld.

Dit is een geloofwaardige en urgente dreiging. Aanvalspogingen en proof-of-conceptcode tonen op zichzelf niet aan dat betaalgegevens van merchants zijn gestolen, en per 22 oktober is ons geen bevestigde compromittering van betaalgegevens bekend die aan PHP2Shell is toe te schrijven. Patch getroffen systemen nu, onderzoek alles wat verdacht is en bewaar de relevante logbestanden. Het ontbreken van een bevestigde inbreuk is geen reden om te wachten.

Waarom dit ertoe doet voor kaartbetalingen

Een merchant hoeft geen eigen PHP-code te draaien om getroffen te zijn. De software onder de webshop is genoeg, en de gevolgen bereiken de betaalkant op vier manieren.

1. **De server die de checkoutpagina opbouwt.** Wie op die server code kan uitvoeren, kan veranderen wat de checkoutpagina doet. Een script injecteren dat kaartgegevens kopieert terwijl klanten ze intypen (vaak skimming of een aanval in Magecart-stijl genoemd) is de al lang bekende route waarlangs gestolen kaartgegevens PHP-webshops verlaten.
2. **Inloggegevens en sleutels.** API-sleutels van de betaalgateway, webhookgeheimen en beheerdersaccounts staan meestal in de configuratie van de webshop. Wie op de server code

kan uitvoeren, kan ze lezen.

3. **Fraude en misbruik.** Gestolen sleutels kunnen worden gebruikt om terugbetalingen te doen of testtransacties uit te voeren. Een gecompromitteerde site kan ook een tweede, verborgen checkout hosten, die voor de provider erg lijkt op [transaction laundering](#).
4. **Downtime.** Zelfs een mislukte aanval kan workers laten crashen. Een webshop die offline is, neemt geen bestellingen aan, en een herstel dat dagen duurt kan cycli van [abonnementsfacturering](#) verstoren en tot terugbetalingen en geschillen leiden.

Merchants die doorverwijzen naar de gehoste betaalpagina van de provider, of de velden van de provider zelf insluiten, staan er beter voor: het kaartnummer komt nooit op hun server. Ze blijven wel blootgesteld via het hierboven beschreven beheer, de terugbetalingen en de API-sleutels.

Wat u deze week moet controleren

- **Zoek uit welke PHP u draait.** Voer op uw eigen server `php -v` uit. Bij beheerde of gedeelde hosting kijkt u in het controlepaneel of vraagt u het de hostingpartij. Plug-ins en webshopsoftware melden het niet, en een webshop kan jaren na de update van de software nog op een oude PHP-versie draaien.
- **Pas de fix van de leverancier toe.** Voor de meeste Linux-distributies en hostingpartijen betekent dat de pakketupdate van de leverancier, gevolgd door een herstart van PHP zodat de wijziging ingaat. Vertrouw niet alleen op het PHP-versienummer, want distributies backporten de fix vaak zonder het te wijzigen.
- **Draait u PHP 7.3 of ouder, plan dan de overstap.** Het PHP-project repareert deze versies niet. Vraag uw hostingpartij of ontwikkelaar of hun leverancier een fix backport, en vraag om de datum, niet om de intentie.
- **Mitigaties zijn tijdelijk.** Het PHP-project documenteert instellingen die elke kwetsbaarheid beperken, en een webapplicatiefirewall of netwerkbepkeringen kunnen de blootstelling verkleinen terwijl u een update voorbereidt. Geen ervan is een volledige oplossing. Stap zo snel mogelijk over op een gerepareerde versie of pas de geverifieerde beveiligingspatch van uw distributie toe, en als een systeem dat vanaf internet bereikbaar is niet snel kan worden gepatcht, overweeg dan de bereikbaarheid ervan te beperken.
- **Vervang geheimen als u een compromittering niet kunt uitsluiten.** Sleutels van de betaalgateway, webhookgeheimen, beheerderswachtwoorden en databasegegevens. Controleer op nieuwe beheerders, gewijzigde checkoutbestanden en onverwachte scripts op de betaalpagina.
- **Bekijk de checkout.** Vergelijk de scripts die de betaalpagina laadt met wat u verwacht. Onbekende domeinen van derden zijn het waarschuwingssignaal.

PCI DSS en uw provider

Valt u binnen de reikwijdte van PCI DSS, dan valt dit onder uw bestaande verantwoordelijkheden voor kwetsbaarheidsbeheer. Eis 6.3.1 vereist dat u nieuwe kwetsbaarheden identificeert en er risicoclassificaties aan toekent. Eis 6.3.3 vereist dat kritieke of zwaarwegende beveiligingspatches, volgens die classificatie, binnen een maand na release worden geïnstalleerd.

Dat is een maximum, geen aanbeveling om te wachten. Gezien de ernst van PHP2Shell en de meldingen van uitbuitingsactiviteit moeten getroffen, vanaf internet bereikbare systemen onmiddellijk worden hersteld. Valt de checkoutpagina binnen de reikwijdte, dan kunnen ook de eisen voor scripts op betaalpagina's relevant zijn.

Een merchant die niet kan aantonen dat hij snel heeft gepatcht en niet kan vastleggen waarom, staat zwak als er iets misgaat. Een korte schriftelijke notitie met wanneer de kwetsbaarheid werd geïdentificeerd, wat er is gedaan en wanneer, is voor een begin voldoende.

Waarschuw uw provider vroeg als u reden hebt te denken dat de webshop is gecompromitteerd. Betaalproviders kunnen meld- en escalatieverplichtingen hebben op grond van de toepasselijke regels van de kaartnetwerken, en een merchant die het op de eerste dag aankaart, wordt heel anders behandeld dan een merchant die de provider ontdekt via een waarschuwing van het kaartnetwerk of een reeks fraudemeldingen.

Voor providers en ISO's

De blootstelling beslaat uw hele portefeuille en is niet gelijk verdeeld. Merchants op end-of-life PHP, op goedkope gedeelde hosting en op zelfbeheerde webshopsoftware zijn de eersten om naar te kijken. Redelijke stappen zijn:

- Merchants op zelf gehoste webshopsoftware benaderen met een korte, duidelijke melding en vragen naar de datum waarop ze hebben gepatcht.
- Letten op de tekenen van een gecompromitteerde checkout: een plotselinge stijging van card testing, onverwachte weigeringen, terugbetalingen die niet bij bestellingen passen, of geschillen die fraude melden op kaarten die alleen bij die merchant zijn gebruikt.
- De websitemonitoring opnieuw draaien op de merchants met het hoogste volume en zoeken naar nieuwe scripts of onbekende betaalbestemmingen.
- Bedenken dat dezelfde kwetsbaarheid uw eigen applicaties raakt en de sites van uw partners en affiliates.

Bij onboarding is een nuttige vraag welk platform en welke hosting de merchant gebruikt, en wie verantwoordelijk is om ze actueel te houden. Het antwoord zegt veel over hoe een account zich in een week als deze zal gedragen.

Bronnen

- [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 lezen: <https://payments-marketplace.com/nl-NL/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