

ESERCENTI, FORNITORI E ISO

PHP2Shell: cosa significano le nuove vulnerabilità di PHP per chi accetta pagamenti con carta

Pubblicato il 22 ottobre 2026

Due vulnerabilità critiche di PHP possono permettere a un aggressore non autenticato di eseguire codice su un server web. Per un esercente è un rischio per l'integrità del checkout, per le credenziali di pagamento e per la disponibilità. Per un fornitore è un rischio di portafoglio. Che cosa è successo, cosa verificare e cosa chiedere al proprio hosting.

Il 21 ottobre 2026 il progetto PHP ha rilasciato correzioni di emergenza per due vulnerabilità critiche del linguaggio stesso, identificate come CVE-2026-109618 e CVE-2026-109619 e soprannominate **PHP2Shell**. Entrambe hanno un punteggio di 9,2 su CVSS 4.0. Ciascuna può permettere a un aggressore non autenticato di eseguire codice su un server web interessato, e le vulnerabilità sono in PHP, non in una singola applicazione costruita sopra di esso.

Per gli esercenti ciò crea un rischio per l'integrità del checkout, per le credenziali di pagamento e per la disponibilità. Per i fornitori di pagamenti e gli ISO crea una possibile esposizione sull'intero portafoglio di esercenti. Ecco che cosa è successo, che cosa verificare e che cosa chiedere al proprio fornitore di hosting.

PHP regge una parte molto ampia del web, compresa la maggior parte del software per negozi online usato dagli esercenti piccoli e medi. Questo rende la vicenda tanto una storia di pagamenti quanto una di sicurezza.

Che cosa è stato trovato

- **CVE-2026-109618** è un errore di corruzione della memoria. Se a PHP si esaurisce la memoria a metà della preparazione di una richiesta, la pulizia successiva può trattare come validi dati costruiti solo in parte.
- **CVE-2026-109619** è un errore di stato obsoleto, di tipo use-after-free. PHP può conservare uno stato interno che punta a memoria della richiesta già liberata, e un'elaborazione successiva può

usarla. PHP indica come effetti l'esecuzione di codice da remoto e la divulgazione di memoria sensibile.

- **Portata.** Il codice interessato più vecchio risale a PHP 5.4. Il progetto PHP afferma che ogni installazione che serve richieste web è a rischio, che usi PHP-FPM, Apache con mod_php, CGI, FrankenPHP o LiteSpeed. PHP valuta il rischio come più alto con mod_php, dove l'attacco dimostrato può arrivare prima che venga eseguito qualsiasi codice del sito. Questo descrive l'attacco che è stato mostrato, non afferma che le altre configurazioni siano sicure.
- **Versioni corrette.** PHP 8.5.12, 8.4.27, 8.3.36 e 8.2.35. Le versioni da 7.4 a 8.1 non hanno un rilascio ufficiale, solo backport minimi del codice sorgente. Le versioni da 5.4 a 7.3 sono a fine vita e non saranno corrette affatto dal progetto PHP. Alcune distribuzioni Linux e alcuni fornitori di supporto a lungo termine riportano effettivamente le correzioni sui rami più vecchi, quindi consulti l'avviso di sicurezza del proprio fornitore prima di dare per scontato che una specifica installazione non sia stata corretta.

A che punto siamo

I dettagli sono trapelati il 20 ottobre, un giorno prima del rilascio previsto, che a sua volta era già stato anticipato dal 22 ottobre. Il team di threat intelligence di Amazon ha segnalato un tentativo contro i propri honeypot il 17 ottobre, quattro giorni prima del rilascio pubblico, e altri da allora, apparentemente da più soggetti. Cloudflare e watchTowr segnalano traffico di attacco, e watchTowr dice di aver riprodotto entrambe le vulnerabilità. Esiste codice di exploit pubblico, anche se la sua affidabilità non è verificata, e il progetto PHP ha dichiarato al rilascio di non aver individuato un exploit funzionante. Non è ancora stabilito se i tentativi osservati finora riuscirebbero contro sistemi di produzione reali.

È una minaccia credibile e urgente. Tentativi di sfruttamento e codice proof-of-concept non dimostrano, da soli, che dati di pagamento siano stati rubati a esercenti, e al 22 ottobre non siamo a conoscenza di alcuna compromissione confermata di dati di pagamento attribuibile a PHP2Shell. Applichi subito le correzioni ai sistemi interessati, indagli su qualsiasi elemento sospetto e conservi i log pertinenti. L'assenza di una violazione confermata non è un motivo per aspettare.

Perché conta per i pagamenti con carta

Un esercente non ha bisogno di eseguire codice PHP proprio per essere colpito. Basta il software sotto il negozio, e le conseguenze raggiungono il lato pagamento in quattro modi.

1. **Il server che costruisce la pagina di checkout.** L'esecuzione di codice su quel server permette a un aggressore di cambiare ciò che fa la pagina di checkout. Iniettare uno script che copia i dati della carta mentre i clienti li digitano (spesso chiamato skimming o attacco di tipo Magecart) è la via storica con cui i dati di carta rubati lasciano i negozi in PHP.
2. **Credenziali e chiavi.** Le chiavi API del gateway di pagamento, i segreti dei webhook e gli accessi amministrativi si trovano di solito nella configurazione del negozio. Chiunque possa eseguire

codice sul server può leggerli.

3. **Frode e abuso.** Le chiavi rubate possono essere usate per emettere rimborsi o eseguire transazioni di prova. Un sito compromesso può anche ospitare un secondo checkout nascosto, che per il fornitore somiglia molto al [transaction laundering](#).
4. **Interruzione del servizio.** Anche un attacco non riuscito può mandare in crash i processi di lavoro. Un negozio fermo non riceve ordini, e un ripristino che dura giorni può compromettere i cicli di [fatturazione degli abbonamenti](#) e causare rimborsi e contestazioni.

Gli esercenti che rimandano alla pagina di pagamento ospitata del fornitore, o che incorporano i campi del fornitore stesso, sono in una posizione migliore: il numero di carta non passa mai dal loro server. Restano comunque esposti sul fronte di amministrazione, rimborsi e chiavi API descritto sopra.

Cosa verificare questa settimana

- **Scopra quale PHP utilizza.** Sul proprio server, esegua `php -v`. Con hosting gestito o condiviso, guardi nel pannello di controllo o lo chieda al fornitore. I plugin e il software del negozio non lo segnalano, e un negozio può restare su una vecchia versione di PHP per anni dopo che il suo software è stato aggiornato.
- **Applichi la correzione del fornitore.** Per la maggior parte delle distribuzioni Linux e degli hosting, significa l'aggiornamento del pacchetto del fornitore, seguito da un riavvio di PHP perché la modifica abbia effetto. Non si affidi al solo numero di versione di PHP, perché le distribuzioni spesso riportano la correzione senza cambiarlo.
- **Se usa PHP 7.3 o precedente, pianifichi il passaggio.** Il progetto PHP non correggerà queste versioni. Chieda al suo fornitore di hosting o allo sviluppatore se il loro fornitore riporta una correzione, e chieda la data, non l'intenzione.
- **Le mitigazioni sono temporanee.** Il progetto PHP documenta impostazioni che restringono ciascuna vulnerabilità, e un web application firewall o restrizioni di rete possono ridurre l'esposizione mentre prepara un aggiornamento. Nessuna è una correzione completa. Passi a una versione corretta o applichi la patch di sicurezza verificata della sua distribuzione il prima possibile e, se un sistema esposto a internet non può essere corretto rapidamente, valuti di limitarne l'esposizione.
- **Cambi i segreti se non può escludere una compromissione.** Chiavi del gateway di pagamento, segreti dei webhook, password amministrative e credenziali del database. Cerchi nuovi utenti amministratori, file di checkout modificati e script inattesi nella pagina di pagamento.
- **Guardi il checkout.** Confronti gli script caricati dalla pagina di pagamento con quelli che si aspetta. I domini di terze parti sconosciuti sono il campanello d'allarme.

PCI DSS e il suo fornitore

Se rientra nell'ambito PCI DSS, questo ricade nelle sue responsabilità già esistenti di gestione delle vulnerabilità. Il requisito 6.3.1 impone di identificare le nuove vulnerabilità e di assegnare classificazioni di rischio. Il requisito 6.3.3 impone che le patch critiche o di alta sicurezza, secondo tale classificazione, siano installate entro un mese dal rilascio.

Quello è un massimo, non un invito ad aspettare. Data la gravità di PHP2Shell e le segnalazioni di attività di sfruttamento, i sistemi interessati esposti a internet vanno corretti subito. Quando la pagina di checkout rientra nell'ambito, possono essere rilevanti anche i requisiti sugli script delle pagine di pagamento.

Un esercente che non è in grado di dimostrare di aver applicato le correzioni con tempestività, e non ha traccia del motivo, si trova in una posizione debole se qualcosa va storto. Per cominciare basta una breve nota scritta con la data in cui la vulnerabilità è stata individuata, che cosa è stato fatto e quando.

Avvisi presto il suo fornitore se ha motivo di ritenere che il negozio sia stato compromesso. I fornitori di pagamenti possono avere obblighi di segnalazione e di escalation ai sensi delle regole applicabili dei circuiti di carte, e un esercente che solleva la questione il primo giorno viene trattato in modo molto diverso da uno che il fornitore scopre tramite un allarme del circuito o una serie di segnalazioni di frode.

Per fornitori e ISO

L'esposizione riguarda l'intero portafoglio e non è distribuita in modo uniforme. Gli esercenti con PHP a fine vita, con hosting condiviso economico e con software di negozio gestito in autonomia sono i primi da esaminare. Passi ragionevoli sono:

- Contattare gli esercenti che usano software di negozio ospitato in proprio con un avviso breve e chiaro e la richiesta della data in cui hanno applicato la correzione.
- Osservare i segnali di un checkout compromesso: un improvviso aumento di card testing, rifiuti inattesi, rimborsi che non corrispondono agli ordini o contestazioni che segnalano frodi su carte usate solo presso quell'esercente.
- Rieseguire il monitoraggio dei siti web sugli esercenti con i volumi più alti e cercare nuovi script o destinazioni di pagamento sconosciute.
- Ricordare che la stessa vulnerabilità riguarda le proprie applicazioni e i siti dei propri partner e affiliati.

All'attivazione, una domanda utile è quale piattaforma e quale hosting usa l'esercente, e chi è responsabile di mantenerli aggiornati. La risposta dice molto su come si comporterà un conto in una settimana come questa.

Fonti

- [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 \(2026-018\)](#)
- [watchTower: PHP remote code execution exploited](#)

Leggi online: <https://payments-marketplace.com/it-IT/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