

COMERCIANTES, PROVEDORES E ISOS

PHP2Shell: o que as novas falhas do PHP significam para quem aceita pagamentos com cartão

Publicado em 22 de outubro de 2026

Duas falhas críticas no PHP podem permitir que um invasor não autenticado execute código em um servidor web. Para o comerciante, isso é um risco para a integridade do checkout, as credenciais de pagamento e a disponibilidade. Para o provedor, é um risco de carteira. O que aconteceu, o que verificar e o que perguntar à sua hospedagem.

Em 21 de outubro de 2026, o projeto PHP lançou correções de emergência para duas falhas críticas na própria linguagem, registradas como CVE-2026-109618 e CVE-2026-109619 e apelidadas de **PHP2Shell**. As duas têm nota 9,2 no CVSS 4.0. Qualquer uma delas pode permitir que um invasor não autenticado execute código em um servidor web afetado, e as falhas estão no PHP, não em um aplicativo específico construído sobre ele.

Para os comerciantes, isso cria um risco para a integridade do checkout, as credenciais de pagamento e a disponibilidade. Para provedores de pagamentos e ISOs, cria uma possível exposição em toda a carteira de comerciantes. Veja o que aconteceu, o que verificar e o que perguntar ao seu provedor de hospedagem.

O PHP sustenta uma parcela muito grande da web, incluindo a maior parte do software de loja virtual usado por comerciantes de pequeno e médio porte. Isso faz desta uma história de pagamentos tanto quanto de segurança.

O que foi encontrado

- **CVE-2026-109618** é um erro de corrupção de memória. Se o PHP ficar sem memória no meio da preparação de uma requisição, a limpeza posterior pode tratar dados montados pela metade como se fossem válidos.
- **CVE-2026-109619** é um erro de estado obsoleto, do tipo use-after-free. O PHP pode manter um estado interno apontando para memória da requisição que já foi liberada, e o processamento

posterior pode usá-la. O PHP lista como efeitos a execução remota de código e a divulgação de memória sensível.

- **Alcance.** O código afetado mais antigo remonta ao PHP 5.4. O projeto PHP afirma que toda instalação que atende requisições web está em risco, seja com PHP-FPM, Apache com mod_php, CGI, FrankenPHP ou LiteSpeed. O PHP avalia o risco como mais alto no mod_php, onde o ataque demonstrado pode chegar antes de qualquer código do próprio site ser executado. Isso descreve o ataque que foi mostrado, não uma afirmação de que as outras configurações são seguras.
- **Versões corrigidas.** PHP 8.5.12, 8.4.27, 8.3.36 e 8.2.35. As versões 7.4 a 8.1 não têm lançamento oficial, apenas backports mínimos do código-fonte. As versões 5.4 a 7.3 chegaram ao fim da vida útil e não serão corrigidas pelo projeto PHP. Algumas distribuições Linux e fornecedores de suporte de longo prazo fazem backport das correções para ramos mais antigos, portanto consulte o aviso de segurança do seu fornecedor antes de presumir que uma instalação específica está sem correção.

Em que pé estão as coisas

Os detalhes vazaram em 20 de outubro, um dia antes do lançamento planejado, que por sua vez já havia sido antecipado de 22 de outubro. A equipe de inteligência de ameaças da Amazon relatou uma tentativa contra seus honeypots em 17 de outubro, quatro dias antes do lançamento público, e outras desde então, aparentemente de vários agentes. A Cloudflare e a watchTowr relatam tráfego de ataque, e a watchTowr diz ter reproduzido as duas falhas. Existe código de exploração público, embora sua confiabilidade não esteja verificada, e o projeto PHP disse, no lançamento, que não havia identificado um exploit funcional. Ainda não está estabelecido se as tentativas vistas até agora teriam sucesso contra sistemas reais de produção.

Esta é uma ameaça crível e urgente. Tentativas de exploração e código de prova de conceito não demonstram, por si sós, que dados de pagamento foram roubados de comerciantes, e, em 22 de outubro, não temos conhecimento de nenhum comprometimento confirmado de dados de pagamento atribuível ao PHP2Shell. Corrija os sistemas afetados agora, investigue qualquer coisa suspeita e guarde os logs relevantes. A ausência de uma violação confirmada não é motivo para adiar.

Por que isso importa para pagamentos com cartão

Um comerciante não precisa executar código PHP próprio para ser afetado. O software por baixo da loja já basta, e as consequências chegam ao lado do pagamento de quatro formas.

1. **O servidor que monta a página de checkout.** A execução de código nesse servidor permite que um invasor altere o que a página de checkout faz. Injetar um script que copia os dados do cartão enquanto o cliente digita (em geral chamado de skimming ou ataque do tipo Magecart) é a forma antiga pela qual dados de cartão roubados saem de lojas em PHP.

2. **Credenciais e chaves.** As chaves de API do gateway de pagamento, os segredos de webhook e os logins de administrador costumam ficar na configuração da loja. Quem consegue executar código no servidor consegue lê-los.
3. **Fraude e abuso.** Chaves roubadas podem ser usadas para emitir reembolsos ou fazer transações de teste. Um site comprometido também pode hospedar um segundo checkout, oculto, que para o provedor se parece muito com [lavagem de transações](#).
4. **Indisponibilidade.** Mesmo um ataque malsucedido pode derrubar processos de trabalho. Uma loja fora do ar não recebe pedidos, e uma recuperação que leve dias pode quebrar ciclos de [cobrança de assinaturas](#) e gerar reembolsos e disputas.

Comerciantes que redirecionam para a página de pagamento hospedada do provedor, ou que incorporam os campos do próprio provedor, estão em melhor situação: o número do cartão nunca passa pelo servidor deles. Ainda assim, continuam expostos no administrador, nos reembolsos e nas chaves de API descritos acima.

O que verificar esta semana

- **Descubra qual PHP você executa.** No seu próprio servidor, execute `php -v`. Em hospedagem gerenciada ou compartilhada, consulte o painel de controle ou pergunte ao provedor de hospedagem. Plugins e softwares de loja não informam isso, e uma loja pode ficar em uma versão antiga do PHP por anos depois de o software ter sido atualizado.
- **Aplique a correção do fornecedor.** Para a maioria das distribuições Linux e hospedagens, isso significa a atualização de pacote do fornecedor, seguida de uma reinicialização do PHP para que a mudança entre em vigor. Não confie apenas no número da versão do PHP, porque as distribuições costumam fazer backport da correção sem alterá-lo.
- **Se você usa PHP 7.3 ou anterior, planeje a migração.** O projeto PHP não vai corrigir essas versões. Pergunte ao seu provedor de hospedagem ou desenvolvedor se o fornecedor deles faz backport de uma correção, e peça a data, não a intenção.
- **Mitigações são temporárias.** O projeto PHP documenta configurações que reduzem cada falha, e um firewall de aplicação web ou restrições de rede podem diminuir a exposição enquanto você prepara uma atualização. Nenhuma é uma correção completa. Atualize para uma versão corrigida ou aplique o patch de segurança verificado da sua distribuição o mais rápido possível e, se um sistema exposto à internet não puder ser corrigido logo, considere restringir sua exposição até que possa.
- **Troque os segredos se não puder descartar um comprometimento.** Chaves do gateway de pagamento, segredos de webhook, senhas de administrador e credenciais do banco de dados. Procure novos usuários administradores, arquivos de checkout alterados e scripts inesperados na página de pagamento.
- **Olhe o checkout.** Compare os scripts que a página de pagamento carrega com o que você espera. Domínios de terceiros desconhecidos são o sinal de alerta.

PCI DSS e o seu provedor

Se você está no escopo do PCI DSS, isso se enquadra nas suas responsabilidades já existentes de gestão de vulnerabilidades. O requisito 6.3.1 exige que você identifique novas vulnerabilidades e atribua classificações de risco. O requisito 6.3.3 exige que patches críticos ou de alta segurança, conforme essa classificação, sejam instalados em até um mês após o lançamento.

Esse é um máximo, não uma recomendação de esperar. Dada a gravidade do PHP2Shell e os relatos de atividade de exploração, os sistemas afetados expostos à internet devem ser corrigidos imediatamente. Quando a página de checkout está no escopo, os requisitos sobre scripts em páginas de pagamento também podem ser relevantes.

Um comerciante que não consegue demonstrar que aplicou a correção com rapidez, e não tem registro do motivo, fica em posição frágil se algo der errado. Uma nota escrita e curta, com a data em que a falha foi identificada, o que foi feito e quando, já é um bom começo.

Avise seu provedor cedo se tiver motivo para achar que a loja foi comprometida. Os provedores de pagamentos podem ter obrigações de comunicação e escalonamento nos termos das regras aplicáveis das bandeiras, e um comerciante que levanta o assunto no primeiro dia é tratado de forma muito diferente de um que o provedor descobre por um alerta da bandeira ou por uma sequência de relatos de fraude.

Para provedores e ISOs

A exposição abrange toda a sua carteira e não está distribuída de forma uniforme. Comerciantes com PHP em fim de vida útil, em hospedagem compartilhada barata e com software de loja autogerenciado são os primeiros a examinar. Passos razoáveis são:

- Entrar em contato com comerciantes que usam software de loja hospedado por conta própria, com um aviso curto e claro e o pedido da data em que aplicaram a correção.
- Observar os sinais de um checkout comprometido: aumento repentino de teste de cartões, recusas inesperadas, reembolsos que não correspondem a pedidos ou disputas que relatam fraude em cartões usados apenas naquele comerciante.
- Executar de novo o monitoramento de sites nos comerciantes de maior volume e procurar scripts novos ou destinos de pagamento desconhecidos.
- Lembrar que a mesma falha afeta os seus próprios aplicativos e os sites dos seus parceiros e afiliados.

Na admissão, uma pergunta útil é qual plataforma e qual hospedagem o comerciante usa, e quem é responsável por mantê-las atualizadas. A resposta diz muito sobre como uma conta vai se comportar em uma semana como esta.

Fontes

- [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](#)

Leia online: <https://payments-marketplace.com/pt-BR/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