

Comunicado: proteção contra uma falha de segurança no PHP

O que aconteceu?

O ILLI funciona graças a várias peças de software. Uma delas se chama **PHP**: ela recebe os pedidos que você faz no sistema (abrir uma tela, salvar uma venda, enviar uma foto) e entrega as respostas.

Em outubro de 2026, especialistas do mundo todo descobriram **duas falhas no PHP**. Ele é usado por milhões de sites, então o problema não é só do ILLI: atinge a internet inteira.

Como seria um ataque?

Pense no PHP como um **carteiro** que recebe pacotes na porta do prédio.

1. Um espertinho manda um pacote **gigante e muito embrulhado**: uma caixa dentro de outra caixa, dentro de outra caixa...
2. O carteiro tenta abrir tudo, fica sem espaço nos braços e **se atrapalha**.
3. Nessa confusão, ele pode **esquecer um bilhete do pacote anterior no bolso** e entregá-lo para a pessoa errada.

Um atacante esperto conseguiria usar essa confusão para dar ordens ao carteiro, como se fosse o dono do prédio.

O que a Illimitar fez?

Ainda não existe uma versão consertada para a peça que usamos. Por isso colocamos **várias proteções em volta dela**, como um condomínio que reforça a segurança:

Proteção	Comparação
☐ Porteiro na entrada	Antes de o pacote chegar ao carteiro, um porteiro confere o tamanho. Pacote grande demais nem entra.
☐ Fila organizada	Se alguém tenta mandar pacotes demais, rápido demais, o porteiro pede para esperar.
☐ Caixas dentro de caixas, só até certo ponto	Pacotes com embrulhos demais são recusados.
☐ Bolsos sempre vazios	Depois de cada entrega, o carteiro troca de uniforme. Não sobra bilhete de ninguém para o próximo.
☐ Sem pânico por falta de espaço	O carteiro não se atrapalha mais quando o pacote é grande. Se algo der muito errado, ele é trocado por outro, descansado e com os bolsos vazios.
☐ O resto do prédio continua seguro	O sistema dele tem um limite próprio e não afeta o restante do servidor.

Também guardamos uma **cópia de tudo como estava antes**, para desfazer as mudanças rapidamente se for preciso.

Meus dados estão seguros?

As proteções já estão sendo aplicadas nos servidores, e cada um passa por uma verificação depois da mudança. Também procuramos sinais de tentativas de ataque nos registros dos servidores e continuamos de olho.

Quando a versão consertada do PHP estiver disponível e testada com o ILLI, vamos instalá-la também. Será como trocar a fechadura velha por uma nova, além de manter o porteiro.

Vou perceber alguma diferença?

Quase nenhuma. O ILLI, as vendas e o sincronismo das lojas continuam funcionando normalmente.

A única mudança que você pode notar:

- **Arquivos muito grandes** (mais de 32 MB) não podem mais ser enviados de uma vez. Fotos, logos e planilhas do dia a dia são bem menores que isso.

Preciso fazer alguma coisa?

Não. Tudo foi feito do nosso lado.

Só entre em contato com o suporte se:

- aparecer um erro ao **enviar um arquivo** que antes funcionava;
- a **sincronização** de alguma loja parar.

Assim conseguimos ajustar rapidinho.

Informações técnicas

Esta seção é para a equipe de TI da sua empresa. Quem não é da área pode pular.

Vulnerabilidades

Identificação	Descrição	Gravidade
CVE-2026-109618	Use-after-free quando o PHP atinge o <code>memory_limit</code> durante o parsing de requisições <code>multipart/form-data</code> . O erro fatal deixa o estado da requisição corrompido.	Crítica
CVE-2026-109619	Estado global do <code>session.upload_progress</code> reaproveitado entre requisições atendidas pelo mesmo processo do PHP-FPM.	Crítica

Combinadas, as duas permitem execução remota de código no servidor web sem autenticação. A correção oficial existe a partir do PHP 8.2.35, 8.3.36, 8.4.27 e 8.5.12. Versões anteriores dependem de correções publicadas pelas distribuições e mantenedores de pacotes.

Mitigações aplicadas pela Illimitar

Camada	Medida	Efeito
PHP	<code>session.upload_progress.enabled = Off</code>	Remove o estado global explorado pela CVE-2026-109619.
PHP	<code>memory_limit = -1</code> , com teto de memória no cgroup do serviço (systemd)	O PHP não aborta no meio do parsing. Se o limite for atingido, o sistema operacional encerra o processo inteiro, que é recriado limpo.
PHP	<code>max_input_nesting_level</code> reduzido; <code>post_max_size</code> e <code>upload_max_filesize</code> em 32 MB	Reduz a amplificação de memória por campos aninhados e o tamanho máximo do corpo.

Camada	Medida	Efeito
PHP-FPM	<code>pm.max_requests = 1</code>	Cada processo atende uma única requisição e é recriado, então nenhum estado passa para a próxima.
NGINX	<code>client_max_body_size</code> de 32 MB e <code>fastcgi_request_buffering on</code>	Corpos grandes são recusados antes do PHP (HTTP 413), e o PHP só recebe requisições completas, com tamanho declarado.
NGINX	Limite de requisições <code>multipart</code> por IP	Reduz tentativas em massa (HTTP 503 para o excesso). Outros tipos de requisição não são afetados.

Todas as alterações têm backup e podem ser desfeitas, e cada servidor recebe um laudo automático antes e depois da aplicação. O laudo verifica a configuração efetiva de cada pool do PHP-FPM e procura indícios de exploração nos logs.

Próximas etapas

- Substituição do PHP por pacotes com a correção oficial (*backport*), servidor a servidor, após homologação com o ILLI e em janela de manutenção.
- Depois disso, os ajustes temporários de memória e de reciclagem de processos voltam aos valores normais. As demais proteções continuam.

Referências

- PHP Security Advisories: <https://github.com/php/php-src/security/advisories>
- Google Cloud: <https://cloud.google.com/blog/products/identity-security/responding-to-core-php-vulnerabilities-cve-2026-109618-cve-2026-109619>
- Cloudflare: <https://blog.cloudflare.com/php-vulnerabilities-waf-protection/>

Equipe Illimitar

Revision #1

Created 2026-10-22 22:27:01 UTC by Heron Santos

Updated 2026-10-22 22:30:38 UTC by Heron Santos