Eagle

THREAT BRIEF · ANÁLISE TÉCNICA E EXECUTIVA · OUTUBRO 2026

PHP2Shell: quando a exploração chega antes do patch.

CVE-2026-109618 e CVE-2026-109619 — o que está confirmado, como as falhas funcionam, o que elas expõem e por que a resposta depende menos da velocidade do patch e mais da capacidade de saber onde, quanto e para quem o risco existe.

PUBLICAÇÃO

Eagle · Futture.ai

Gestão integrada de riscos cibernéticos

VERSÃO

1.1 · 23 out 2026

Revisão com fontes primárias consultadas em 23/10/2026 · Classificação: pública

ANTES DE LER

Um evento em curso. Separe o que se sabe do que se diz.

O PHP2Shell vazou antes da divulgação coordenada, e circularam versões conflitantes sobre escopo, pontuação e versões afetadas. Esta versão foi revisada contra os comunicados do PHP, os advisories no GitHub, os registros CVE e NVD, o catálogo KEV e as publicações originais dos pesquisadores, consultados em 23/10/2026.

Figura 1. Classificação das evidências usada neste documento

CONFIRMADO
Sustentado por fonte primária verificável: comunicado do PHP, advisory GHSA, registro CVE/NVD, catálogo KEV ou alerta de autoridade.

REPORTADO
Divulgado por fonte identificada — inclusive a própria organização que observou o fato — sem confirmação independente.

INFERIDO
Interpretação técnica sustentada pelas evidências, mas não demonstrada diretamente.

DESCONHECIDO
Informação não estabelecida, contraditória ou indisponível na data de consulta.

Números entre colchetes [n] remetem à lista de referências (pp. 22–23). Uma afirmação repetida por várias publicações que usam a mesma origem continua "reportada".

Sumário

—	Resumo executivo	03
01	O que aconteceu	04
02	Anatomia técnica e condições de exposição	05
02b	Acontecimento relacionado: CVE-2026-115560	07
03	Cadeia de exploração (nível conceitual)	08
04	Severidade, probabilidade e risco	09
05	Cronologia	10
06	A janela de exposição	12
07	Limites da gestão tradicional de vulnerabilidades	13
08	Threat Intelligence e exploração ativa	14
09	Do achado ao risco: gestão e governança	15
10	BIA e quantificação	16
11	Capacidades necessárias e o papel do Eagle	17
12	Recomendações técnicas	18
13	Recomendações executivas e estruturais	20
—	Conclusão · Referências · Sobre o Eagle	21

Registro de versões

1.0 · 23/10	Primeira publicação.
1.1 · 23/10	Fontes primárias lidas diretamente; CVE-2026-115560, EPSS e KEV incluídos; cronologia e estatísticas com fonte e método; recomendações por tipo de medida; referências numeradas.

Escopo e limites

Material educativo e defensivo. Não contém payloads nem instruções de exploração. As medidas de configuração reproduzem as orientações do PHP e não substituem a atualização.

RESUMO EXECUTIVO

Duas falhas no motor do PHP e uma janela que abriu antes do patch.

Em 21 de outubro de 2026, o projeto PHP publicou correções para CVE-2026-109618 e CVE-2026-109619, conhecidas como PHP2Shell. As duas permitem execução remota de código sem autenticação em instalações que atendem requisições web.[1] Os detalhes vazaram antes da divulgação coordenada, e a Amazon relata uma tentativa de exploração quatro dias antes da correção.[15]

SEVERIDADE TÉCNICA

9,2

CVSS v4.0 do PHP para cada CVE; complexidade alta e pré-requisitos presentes.[3][4] CONFIRMADO

CORREÇÃO

8.5.12 · 8.4.27

8.3.36 · 8.2.35

Sem release oficial para 8.1 e anteriores; backports mínimos em código-fonte para 7.4, 8.0 e 8.1.[1] CONFIRMADO

EXPLORAÇÃO

Tentativas relatadas

Amazon e Cloudflare relatam tentativas; sucesso não estabelecido. Fora do KEV (versão 2026.10.23). EPSS 11,6% e 6,0%.[10][11][15][16] REPORTADO

Principais riscos

- **Execução de código com os privilégios da conta do worker web**, sem login e sem interação. Com mod_php, o ataque documentado ocorre antes de qualquer código da aplicação.[1][3]
- **Divulgação de memória do processo**, listada pelo PHP como impacto confirmado de CVE-2026-109619.[4]
- **Base instalada sem correção oficial**: ramos 8.1 e anteriores dependem de backport, migração ou isolamento.[1]
- **PHP embarcado em produtos de terceiros**: o NVD já associa configurações de appliances de outros fabricantes às duas CVEs.[7][8]
- **Correção aparente**: pacote atualizado com workers antigos em memória ou runtime diferente do binário verificado.[17]

Decisões que cabem à liderança

- Autorizar manutenção emergencial dos ativos expostos à internet, começando por mod_php.
- Decidir, ativo a ativo, entre corrigir, mitigar temporariamente, isolar ou retirar do ar.
- Formalizar qualquer aceite de risco residual com dono, prazo e controle compensatório.
- Cobrar fornecedores, hospedagens e fabricantes de appliances que embarcam PHP.
- Exigir evidência de remediação validada, não apenas de pacote instalado.

Atualização de 23/10: CVE-2026-115560

O PHP publicou um aviso informativo sobre outra falha no progresso de upload de sessão, que afeta inclusive as versões já corrigidas, mas só sob uma combinação incomum de configuração e código. A mitigação é a mesma já recomendada para CVE-2026-109619: `session.upload_progress.enabled=0`. O PHP não a classifica como parte do PHP2Shell; ver p. 7.[2][9]

Capacidade organizacional que o caso testa

Responder em horas exige saber, antes do incidente, onde o PHP roda, em qual versão e modo de execução, quem responde por ele, o que cada ativo sustenta no negócio e quais controles compensatórios existem.

01 O QUE ACONTECEU

A falha não está na aplicação. Está no runtime que a executa.

PHP2Shell é o nome informal dado a duas vulnerabilidades no núcleo do PHP; o nome não vem do projeto. Como o defeito está no interpretador, uma aplicação sem nenhum erro próprio pode estar exposta, e atualizar WordPress, Drupal ou plugins não resolve.

Segundo a Wiz Research, o primeiro defeito foi identificado em 19 de agosto de 2026 e reportado ao time de segurança do PHP em 24 de agosto.[14] Os advisories creditam como reportadores pesquisadores da Wiz, da OpenAI e outros, e a Cloudflare cita Calif.io e Quarkslab entre os responsáveis pela divulgação.[3][16] Em 20 de outubro, a existência das falhas vazou antes da data combinada; no mesmo dia, o PHP antecipou a release e a CERT-EU publicou o advisory 2026-018.[12][25]

A Wiz relata o uso de modelos de IA na descoberta, no desenvolvimento de exploits e na validação de patches.[14] Para defensores, o ponto relevante é o efeito: o custo de transformar defeitos antigos de memória em exploração prática caiu, e a distância entre divulgação e ataque encolheu.

Figura 2. O que está confirmado e o que ainda é relato (consulta em 23/10/2026)

Afirmação	Estado	Base e observações
Duas CVEs no núcleo do PHP	CONFIRMADO	CVE-2026-109618 ↔ GHSA-w8hw-w9fx-j79w e CVE-2026-109619 ↔ GHSA-6wmf-4rv5-p3rp, publicadas pelo PHP como CNA.[3][4][7][8]
RCE sem autenticação	CONFIRMADO	Validada pelo PHP contra mod_php; em FPM, validada para o ciclo de sessão de CVE-2026-109619, dependente da aplicação.[3][4]
CVSS 9,2 (v4.0)	CONFIRMADO	Pontuação do CNA; o NVD ainda não publicou avaliação própria. O vazamento falava em 10,0.[7][25]
Divulgação de memória	CONFIRMADO	Listada como impacto confirmado de CVE-2026-109619.[4] Na v1.0 constava como reportada.
Versões afetadas abaixo de 8.2	REPORTADO	Registros oficiais listam 8.2–8.5; o NVD mapeia CVE-2026-109618 a partir da 7.3.0; pesquisadores citam a série 5.4 em diante para alguns caminhos. Escopo exato por ramo não publicado pelo PHP.[6][7][14]
Tentativa antes do patch	REPORTADO	Amazon (17/10) e Cloudflare relatam tráfego de ataque em suas próprias redes.[15][16]
Comprometimentos confirmados	DESCONHECIDO	Nenhuma fonte primária estabelece explorações bem-sucedidas; o CCCS não identificou exploração ativa até 23/10.[13]
Mapeamento defeito ↔ CVE	DESCONHECIDO	A Wiz agrupa quatro defeitos sob as duas CVEs de forma diferente dos títulos do PHP. Este documento segue o CNA.[14]

02 — ANATOMIA TÉCNICA DAS FALHAS

Quando o PHP para no meio do caminho, o que fica para trás vira superfície de ataque.

As duas falhas exploram o tratamento de condições excepcionais durante uma requisição. Quando a memória acaba ou ocorre um erro fatal, o PHP abandona a operação por um salto direto (`longjmp`), sem a limpeza normal. Estruturas internas podem ficar parcialmente inicializadas ou continuar apontando para memória já liberada.[3][4]

CVE-2026-109618 · GHSA-W8HW-W9FX-J79W

RCE via corrupção por falta de memória

Uma falha de alocação interrompe o PHP enquanto memória da requisição já está acessível, mas ainda não inicializada. O encerramento da requisição lê ou toma decisões sobre esse estado inválido. A falha de alocação não precisa estar relacionada à memória inválida.[3]

O advisory descreve formas de chegar ao limite de memória dentro dos limites aparentes do PHP via requisições *multipart*. Com mod_php, o encerramento roda mesmo quando a inicialização falha, antes de qualquer código da aplicação. No FPM, o processo filho é encerrado nessa situação; a análise de código indica suscetibilidade em outros pontos, mas sem exploit completo demonstrado.[3]

CONFIRMADO

CVE-2026-109619 · GHSA-6WMF-4RV5-P3RP

RCE via estado global obsoleto

Estado global pertencente a uma requisição sobrevive depois que sua memória é liberada. Uma operação da requisição seguinte, no **mesmo worker**, consome esse estado antes da reinicialização normal: um *use-after-free*. A exploração exige um "produtor" e um "consumidor" do estado.[4]

Casos confirmados: os backtraces de erro fatal do PHP 8.5 (base da RCE confirmada em mod_php) e o ciclo de progresso de upload de sessão (`session.upload_progress`, ativo por padrão), validado contra mod_php e FPM. Diagnósticos de configuração aplicados por requisição também podem atuar como consumidores.[4]

CONFIRMADO

Figura 3. O que o vetor CVSS v4.0 diz e o que ele não diz

Métrica	Valor	Leitura para defesa
Vetor de ataque (AV)	Rede	Alcançável por requisição HTTP a endpoint tratado pelo PHP.
Privilégios / interação	Nenhum / nenhuma	Não exige conta, sessão autenticada nem ação de usuário.
Complexidade (AC) e pré-requisitos (AT)	Alta / presentes	Portabilidade e confiabilidade dependem do build, das extensões, do frontend e do layout do alocador; mais workers aumentam as tentativas necessárias, sem eliminar a condição. [3][4]
Impacto no sistema vulnerável	Alto (C, I, A)	Execução com os privilégios do worker web. Não equivale a acesso administrativo ao sistema operacional.
Impacto em sistemas subsequentes	Nenhum	O vetor não pontua movimento lateral; ele depende do que o servidor alcança e das credenciais que guarda.

Vetor: CVSS:4.0/AV:N/AC:H/AT:P/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N.[3][4] CWEs nos advisories: 416, 755, 824, 1325 e 1419; o NVD exibe, para CVE-2026-109618, apenas as quatro primeiras.[7]

02 ANATOMIA TÉCNICA · CONDIÇÕES DE EXPOSIÇÃO

Toda instalação que serve web está no escopo. A forma de execução muda a urgência.

O PHP considera em risco qualquer instalação que atende requisições web.[1] Os modos de execução diferem no que já foi demonstrado, não na existência da falha: a ordem abaixo é de urgência, não uma lista de quem pode esperar.

Figura 4. Exposição por modo de execução, segundo os advisories do PHP

Modo	CVE-2026-109618	CVE-2026-109619	Prioridade
Apache mod_php	RCE validada via HTTP, com limites padrão, antes de código da aplicação CONF.	RCE confirmada (backtrace no 8.5 e ciclo de sessão) CONF.	Primeira; maior risco segundo o PHP
PHP-FPM	Suscetível por análise de código; sem exploit completo INF.	RCE validada no ciclo de sessão, dependente de operação da aplicação; exposição condicional via PHP_VALUE do frontend CONF.	Alta
FrankenPHP	Provavelmente suscetível a ataques independentes da aplicação, por analogia ao mod_php INF.	Consumidor de configuração não estabelecido no padrão; não implica imunidade DESC.	Tratar como exposto
LiteSpeed (LSAPI)	Modelado como FPM; exploits dependentes da aplicação, prováveis e não validados INF.	Candidato condicional INF.	Tratar como exposto
php-cgi	Listado como afetado no comunicado; sem detalhamento nos advisories DESC.		Tratar como exposto

Fontes: comunicado de 21/10 e advisories GHSA-w8hw-w9fx-j79w e GHSA-6wmf-4rv5-p3rp.[1][3][4]

Figura 5. Versões e caminho de correção

Ramo	Primeira versão corrigida	Caminho
8.5 · 8.4 · 8.3 · 8.2	8.5.12 · 8.4.27 · 8.3.36 · 8.2.35[1]	Atualizar e reiniciar todos os processos que carregam PHP. Distribuições e fornecedores LTS receberam os patches antecipadamente; pacotes podem manter o número de versão antigo com o backport aplicado.[1][17]
8.1 · 8.0 · 7.4	Sem release oficial	Backports mínimos em código-fonte do PHP, para quem compila; pacote de distribuição ou fornecedor LTS; ou migração.[1]
7.3 e anteriores	Sem correção	Migrar, isolar ou retirar do ar. O escopo exato de afetação nesses ramos não foi publicado pelo PHP.[6][7]

O php -v engana

O binário de linha de comando pode ter versão diferente do que roda no FPM ou no Apache. Verificar no runtime que atende a web.[17]

Workers em memória

Processos em execução mantêm o código vulnerável até o reinício, o que pesa em CVE-2026-109619.[4][17]

PHP de terceiros

O NVD associa às CVEs configurações de appliances Siemens e Palo Alto Networks. Consultar o advisory de cada fabricante.[7][8]

02B — ACONTECIMENTO RELACIONADO · NOVO EM 23/10

CVE-2026-115560: mesmo recurso, outra falha, outra urgência.

Em 23 de outubro, o projeto PHP publicou um aviso informativo recomendando desativar o progresso de upload de sessão até a próxima release regular, para proteger contra um vetor de execução remota que descreve como improvável.[2] A falha não faz parte das correções de 21 de outubro e afeta também as versões já corrigidas, como a 8.5.12.[2][16]

O QUE O PHP PUBLICOU

Condição de exploração

A aplicação precisa definir `session.serialize_handler` como `php_binary` ou `php_serialize` na configuração e trocá-lo para `php` em tempo de execução, com `ini_set()`, antes de chamar `session_start()` — com o progresso de upload ainda ativo. Aplicações que nunca mudam o serializador em tempo de execução podem manter o recurso.[2]

CONFIRMADO

ESTADO EM 23/10

Sem correção ainda

O PHP tratará a falha no ciclo regular e publicará o advisory completo com a próxima release; não há release extraordinária.[2][12] O registro CVE está como **RESERVED**, sem descrição, CVSS nem faixa de versões.[9]

CONFIRMADO CVSS DESCONHECIDO

Figura 6. Relação com o PHP2Shell

Aspecto	O que as fontes estabelecem
É parte do PHP2Shell?	O PHP não a classifica assim: trata-a como problema distinto, ligado ao mesmo recurso. A CERT-EU a descreve como "terceira vulnerabilidade" em seu histórico de versões; a Cloudflare, como "vulnerabilidade relacionada".[2][12][16] Este documento a apresenta como acontecimento relacionado.
Mitigação	`session.upload_progress.enabled=0`, a mesma medida já listada para CVE-2026-109619. Quem a aplicou em 21/10 não precisa de ação adicional.[2][13]
Efeito prático	A medida que era temporária para CVE-2026-109619 passa a ser recomendada mesmo após a atualização, até a próxima release. O advisory de 21/10 já sugeria mantê-la desativada quando o recurso não é usado.[2][4]
WAF	A Cloudflare informa cobertura contra tentativas conhecidas apenas quando a aplicação usa o nome padrão do campo de progresso de upload; quem alterou `session.upload_progress.name` não está coberto pela regra.[16] REPORTADO
Exploração	Nenhuma fonte consultada relata exploração. DESCONHECIDO

O que muda na decisão

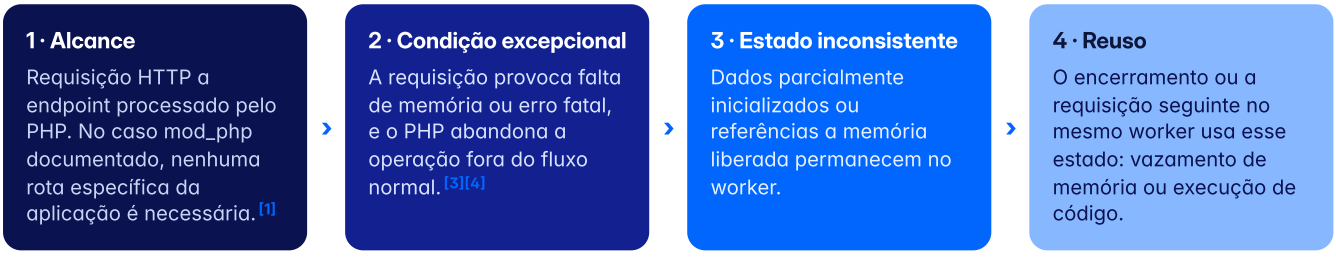
Atualizar para 8.5.12, 8.4.27, 8.3.36 ou 8.2.35 continua necessário, mas deixou de ser suficiente para fechar todos os vetores do progresso de upload. Onde o recurso não é usado, desativá-lo é uma decisão de baixo custo que cobre os dois casos.

03 — CADEIA DE EXPLORAÇÃO · NÍVEL CONCEITUAL

Do pedido HTTP ao processo comprometido, em quatro passos e nenhuma credencial.

A descrição é deliberadamente conceitual. Ela serve para entender onde cada controle pode atuar, não para reproduzir o ataque.

Figura 7. Encadeamento conceitual e pontos de intervenção



Barreiras: WAF com regras específicas (paliativo; a Cloudflare observou mutações de payload para contorná-lo) [16]; limites de corpo e de partes multipart no proxy. [3]

Barreiras: medidas temporárias do PHP sobre limite de memória e backtraces. [1]

Barreiras: layout do alocador e número de workers reduzem a confiabilidade, não eliminam a falha. [3][4]

Barreiras: reciclagem de worker por requisição no FPM (custo de desempenho) e, sobretudo, a correção. [4]

O que o atacante obtém

Execução de código com os privilégios da conta do worker web, conforme os advisories. [3][4] Isso não é, por si só, acesso administrativo ao sistema operacional. Na prática, porém, essa conta costuma ler arquivos de configuração, credenciais de banco de dados e chaves de integração, e pode alcançar serviços internos a partir do servidor. INFERIDO

A divulgação de memória é um impacto independente da RCE: mesmo sem executar código, dados que circularam pelo processo podem ser expostos. [4] CONFIRMADO

Relação com MITRE ATT&CK

Técnica	Base
T1190 Exploit Public-Facing Application	APLICÁVEL decorre da natureza da falha
T1505.003 Web Shell · **T1552** Unsecured Credentials · **T1496** Resource Hijacking	INFERIDO pós-exploração possível; não observado em campanhas

Nenhuma campanha, ator ou conjunto de IoCs foi associado ao PHP2Shell nas fontes primárias consultadas. [26]

04 — SEVERIDADE, PROBABILIDADE E RISCO

Um 9,2 não é um risco de negócio. É um dos insumos para calculá-lo.

O CVSS mede a gravidade intrínseca da falha em condições genéricas. Ele não sabe se o servidor está exposto, se há exploração em curso, nem o que o servidor sustenta.

Figura 8. Três perguntas diferentes, três fontes diferentes

SEVERIDADE

Quão grave é a falha?

CVSS 9,2: rede, sem privilégio, sem interação, impacto alto, complexidade alta. [3][4]

PROBABILIDADE

Quão provável aqui?

Exposição, modo de execução, PoCs, tentativas observadas, controles na frente do ativo. O EPSS de 23/10 estima 11,6% (109618) e 6,0% (109619) em 30 dias.

RISCO DE NEGÓCIO

O que se perde?

Dados processados, credenciais guardadas, serviços alcançáveis, processos que param. Fonte: inventário, BIA e gestão de riscos.

Figura 9. Ordem de grandeza da exposição — cada número conta uma coisa diferente

Número	O que conta	Fonte e método	O que não prova
69,3%	Sites cuja linguagem de servidor é identificável e que usam PHP	W3Techs, levantamento de 23/10/2026, por amostragem de sites. [19] CONF.	Não mede versão nem exposição; não significa que esses sites estejam vulneráveis.
167 mi	Instalações de pacotes PHP em 12 mil clientes Qualys; 73% em ativos classificados como voltados à internet	Qualys TRU, telemetria anonimizada própria, 20–22/10. [17] REP.	Instalação ≠ host ≠ serviço exposto. O mesmo texto cita 166 mi e 74% em outro trecho.
71%	Das 155 mi instalações com versão conhecida na mesma base, em ramos 5.4–8.1	Qualys TRU. [17] REP.	Ramos sem release oficial; parte pode ter backport de distribuição.
2,24 mi	Hosts na internet que anunciam PHP em banners	Contagem Shodan publicada pela Mallory em 20/10, reproduzida pela CybelAngel. [20] REP.	Hosts, não sites nem organizações; ~740 mil ocultam versão; hospedagem compartilhada agrega muitos sites em um IP.
59%	Dos ~1,5 mi hosts que exibem versão (883.717), em ramos 8.1 ou anteriores	Mesma fonte. [20] REP.	Versão de banner pode não refletir backport; não mede exploração.

Telemetrias de fornecedores descrevem as próprias bases e não são comparáveis entre si nem extrapoláveis para o mercado. Nenhuma delas mede instalações efetivamente exploráveis.

A gravidade se sustenta sem adjetivos: alcance amplo, nenhum pré-requisito de acesso, cauda longa de versões sem suporte e informação técnica circulando antes do patch. Ao mesmo tempo, a própria Amazon afirma não estar claro se as tentativas observadas teriam sucesso em sistemas reais, e o CCCS não havia identificado exploração ativa até 23/10. [13][15] Esses dados envelhecem rápido e não autorizam adiar a correção, mas impedem que se fale em comprometimento em massa sem evidência.

05 — CRONOLOGIA

Dois meses de sigilo, quatro dias de exposição invisível.

Cada evento traz a fonte original e o estágio de exploração a que corresponde. Horários aparecem só quando a fonte os informa; quando fontes divergem, as duas versões são indicadas.

INFORMATIVO EXPLOIT DESENVOLVIDO EXPLOIT PUBLICADO TENTATIVA OBSERVADA EXPLORAÇÃO BEM-SUCEDIDA (nenhum registro público)

Data	Acontecimento	Fonte	Classificação
19–21 ago	Wiz Research identifica o primeiro defeito e confirma RCE em laboratório contra Drupal (20/08) e WordPress (21/08). EXPLOIT DESENVOLVIDO	Wiz Research [14]	REPORTADO
24 ago	Falhas reportadas ao time de segurança do PHP.	Wiz Research [14]	REPORTADO
21 set	PHP reconhece os relatos e inicia correção coordenada; provedores de nuvem e CDN participam.	Wiz Research [14]	REPORTADO
Início de out	Cloudflare desenvolve proteções de WAF.	Cloudflare [16]	REPORTADO
17 out	**Tentativa de exploração contra a rede de honeypots MadPot**, a partir de rede de anonimização comercial, quatro dias antes da divulgação. A Amazon não sabe como o ator obteve a informação. TENTATIVA OBSERVADA	Amazon Threat Intelligence [15]	REPORTADO fonte primária do observador
20 out	**Vazamento** antecipa as falhas e os identificadores; a divulgação coordenada estava prevista para 22/10.	Cobertura jornalística [25]	REPORTADO
20 out	PHP antecipa a release para 21/10. CERT-EU publica o advisory 2026-018 (v1.0). INFORMATIVO	PHP; CERT-EU [1][12]	CONFIRMADO
21 out, ~02h UTC	Amazon identifica as primeiras provas de conceito públicas e, até ~08h UTC, novas tentativas de infraestruturas distintas, sugerindo múltiplos atores. EXPLOIT PUBLICADO TENTATIVA	Amazon Threat Intelligence [15]	REPORTADO
21 out, 03h02 UTC	Cloudflare ativa proteção padrão no WAF, cerca de nove horas antes da divulgação, e relata tráfego de ataque contra clientes e mutações de payload para contornar as regras. TENTATIVA OBSERVADA	Cloudflare [16]	REPORTADO

05 CRONOLOGIA · CONTINUAÇÃO

Data	Acontecimento	Fonte	Classificação
21 out	**PHP publica 8.5.12, 8.4.27, 8.3.36 e 8.2.35** e o advisory com medidas temporárias. O horário anunciado era 14h UTC; a CERT-EU registra a publicação às 13h30 UTC. INFORMATIVO	PHP; CERT-EU [1][12]	CONFIRMADO
21 out	Registros CVE publicados pelo PHP como CNA; anúncio na lista oss-security.	NVD; oss-security [5][7][8]	CONFIRMADO
21 out	CCCS (Canadá) publica o alerta AL26-026. CISA recomenda correção imediata; NCSC (Reino Unido) sugere retirar do ar o que não puder ser corrigido.	CCCS [13]; CISA e NCSC via imprensa [22][23]	CONFIRMADO CCCS · REP. CISA/NCSC
21 out	Amazon relata PoCs adicionais horas após o patch, usadas quase imediatamente contra seus sensores; afirma não estar claro se alguma tentativa teria sucesso em sistemas reais. TENTATIVA OBSERVADA	Amazon Threat Intelligence [15]	REPORTADO
21–22 out	watchTowr reproduz as duas falhas e observa tentativas em sua rede de honeypots; em entrevista, afirmou que nenhuma havia tido sucesso até então. EXPLOIT DESENVOLVIDO TENTATIVA	watchTowr [18]; entrevista [24]	REPORTADO
23 out	**PHP publica aviso informativo sobre CVE-2026-115560** no progresso de upload de sessão. CERT-EU atualiza o advisory (v1.2); CCCS publica atualização; Cloudflare atualiza seu comunicado. INFORMATIVO	PHP; CERT-EU; CCCS; Cloudflare [2][12][13][16]	CONFIRMADO
23 out	Catálogo KEV versão 2026.10.23: nenhuma das CVEs listada. EPSS publicado: 11,6% e 6,0%.	CISA; FIRST [10][11]	CONFIRMADO
23 out	Relatos de que parte das PoCs públicas usa técnicas antigas de envenenamento de sessão contra falhas de inclusão de arquivo, e não o PHP2Shell; NCSC-NL mantém inventário de produtos afetados.	Field Effect [21]	REPORTADO

LEITURA DEFENSIVA

O patch não abriu a janela

Se o relato da Amazon estiver correto, alguém testava a falha antes de existir correção. A própria Amazon recomenda iniciar investigações a partir de 17/10. [15]

DIVULGAÇÃO COORDENADA

O que funcionou

Distribuições e fornecedores LTS receberam os patches antecipadamente; WAFs de borda tinham regras antes do anúncio. [1] [16]

DIVULGAÇÃO COORDENADA

O que falhou

O vazamento antecipou a informação a atacantes e deixou defensores com cerca de um dia entre o aviso e o patch.

Estágios não equivalentes

Exploit desenvolvido em laboratório, exploit publicado, tentativa observada, exploração bem-sucedida e comprometimento confirmado são estados distintos. Até 23/10, as fontes primárias registram os três primeiros; não há registro público dos dois últimos.

06 — A JANELA DE EXPOSIÇÃO

O tempo que importa não é o do patch. É o do risco residual em cada ativo.

A janela de exposição começa quando a falha passa a ser explorável por alguém que não o defensor, e termina quando a organização comprova que o ativo deixou de ser explorável. Entre esses pontos, há etapas que o fornecedor controla e etapas que só a organização controla.

Figura 10. As oito etapas da janela, de quem depende cada uma e o que é visível

	Antes da divulgação	Divulgação	Resposta interna	Fechamento
Etapas	1 · Falha existe no código 2 · Primeiras tentativas 3 · Pesquisadores identificam	4 · Mantenedor notificado 5 · Correção disponível	6 · Ativos identificados 7 · Compensação aplicada	8 · Remediação aplicada e validada
Quem controla	Ninguém do lado defensor	Pesquisadores e mantenedor	A organização e seus fornecedores	A organização
Visibilidade	Só via Threat Intel e telemetria própria	Advisories, CVE, CERTs	Inventário e scanners atualizados	Revalidação no runtime
Risco se falhar	Exploração silenciosa sem alerta de ninguém	Atacante lê o mesmo advisory que o defensor	Ativo esquecido fica fora da conta	"Corrigido" no relatório, vulnerável em memória

No PHP2Shell, a etapa 2 (17/10, segundo a Amazon) precede a 5 (21/10). [1][15] A CVE-2026-115560 mostra que a etapa 8 pode ser reaberta por uma falha relacionada. [2]

O limite do MTTR isolado

O tempo médio de remediação mede a velocidade de quem fecha chamados. Uma média baixa pode esconder exatamente o caso que importa: o servidor legado em PHP 7.4, sem dono claro, que fica meses aberto enquanto centenas de ativos simples puxam a média para baixo. Ele também começa a contar tarde, a partir da detecção, ignorando a parte da janela que antecede o conhecimento da organização.

Em eventos como o PHP2Shell, interessa a distribuição por criticidade e exposição, não a média, e medir cada trecho da janela separadamente.

Indicadores que descrevem a janela

Identificar ativos	Do advisory à lista de onde o PHP roda, com versão e modo.
Determinar exposição	Até saber o que é alcançável da internet e o que protege.
Compensar	Até a medida temporária aplicada e verificada.
Detectar sinais	Até revisar logs do período anterior.
Remediação validada	Até comprovar no runtime em execução.
Exposição residual	Ativos e dias ainda abertos, aceitos ou não.

Não há referência de mercado consolidada para esses tempos; as metas decorrem do apetite a risco de cada organização.

07 — LIMITES DA GESTÃO TRADICIONAL DE VULNERABILIDADES

Scanner, lista de CVEs e prazo de patch continuam necessários. Não são suficientes.

Programas de gestão de vulnerabilidades foram desenhados em ciclos: varrer, listar, priorizar por pontuação, corrigir no prazo, reportar. O PHP2Shell comprime o ciclo em horas. A resposta não é abandonar essas práticas, e sim complementá-las com contexto de exposição, ameaça, ativo e negócio.

Figura 11. O que cada prática entrega, onde falha neste caso e o que a complementa

Prática	Continua necessária porque	Onde falha no PHP2Shell	Complemento
Inventário de ativos	Sem ele não há escopo.	Raramente registra o PHP embarcado em contêineres, appliances e painéis — e o NVD já lista appliances afetados. [7]	Inventário de software com dono e função de negócio.
Varreduras periódicas	Detectam versões e configurações vulneráveis.	Ciclo mais lento que a janela; cerca de um terço dos hosts que anunciam PHP oculta a versão. [20]	Detecção por advisory sobre o inventário; verificação autenticada.
Listas de CVEs	Linguagem comum entre ferramentas.	Mostram a falha, não o caminho até ela nem o que há atrás do servidor.	Relacionamento CVE ↔ ativo ↔ exposição ↔ processo.
Priorização por CVSS	Mede gravidade intrínseca de forma padronizada.	O 9,2 não diferencia o servidor público do laboratório isolado.	Sinais de exploração, exposição real e criticidade do ativo.
Relatório de vulnerabilidades abertas	Dá visibilidade de volume e tendência.	Contagem não mostra onde está o risco nem se a correção chegou ao processo.	Exposição residual por ativo crítico, com evidência.
SLA de patch	Cria disciplina e cobrança.	Prazos usuais para "crítico" são longos quando há tentativa observada; sem correção, o SLA não se aplica.	Exceção acionada por Threat Intel e decisão formal de tratamento.
Indicadores de remediação	Mostram desempenho operacional.	Medem chamados fechados, não exposição removida.	Indicadores por etapa da janela e criticidade.

Coletar uma vulnerabilidade não é entender o risco que ela representa

Um scanner informa que um host roda PHP 8.1. A pergunta de gestão é outra: ele está exposto, a que dados e credenciais dá acesso, que processo para se ele cair, quem decide e até quando. Exposure Management responde a essas perguntas usando, entre outros insumos, o resultado da gestão de vulnerabilidades.

08 — THREAT INTELLIGENCE E EXPLORAÇÃO ATIVA

"Explorada ativamente" descreve quatro situações muito diferentes.

Inteligência de ameaças muda a prioridade quando informa o estado real da exploração. No PHP2Shell, manchetes trataram como equivalentes coisas que não são.

Figura 12. Escada de evidências de exploração e o estado do PHP2Shell em 23/10/2026

	Exploit disponível	Tentativa de exploração	Exploração bem-sucedida	Comprometimento confirmado
Significa	Código capaz de explorar existe e está acessível	Tráfego direcionado à falha foi observado	A falha foi acionada e produziu efeito	Há evidência forense de acesso, persistência ou dano
PHP2Shell	PoCs públicas desde ~02h UTC de 21/10; parte delas não explora o PHP2Shell REP. [15][21]	Amazon e Cloudflare relatam tentativas em suas redes REP. [15][16]	Não estabelecida; Amazon não sabe se teriam sucesso DESC. [15]	Nenhum caso público; CCCS sem exploração ativa identificada DESC. [13]
Efeito na decisão	Eleva a urgência acima do CVSS	Justifica tratamento emergencial dos expostos	Exige caça a ameaças nos próprios ativos	Aciona resposta a incidentes e notificações

Inteligência externa não é evidência interna

Um relatório de tentativas em honeypots prova que a falha está sendo atacada no mundo. Não prova que um ativo específico foi atacado, nem que não foi. Essa conclusão depende de evidência do próprio ambiente.

A ausência de alertas também não prova ausência de comprometimento. O próprio PHP adverte que a ausência de erros nos logs não é prova de segurança, porque a falha pode ocorrer antes de a mensagem ser escrita e logs antigos podem ter sido coletados com severidades excluídas.[4]

KEV, EPSS e sinais de fornecedores

CISA KEV: as duas CVEs não constam no catálogo versão 2026.10.23, verificado na fonte. A ausência indica falta de confirmação suficiente, não baixo risco.[10] CONF.

EPSS (23/10): 0,116 para CVE-2026-109618 (percentil 96) e 0,060 para CVE-2026-109619 (percentil 93).[11] O EPSS compara populações de vulnerabilidades; não decide sobre um ativo isolado e muda diariamente. CONF.

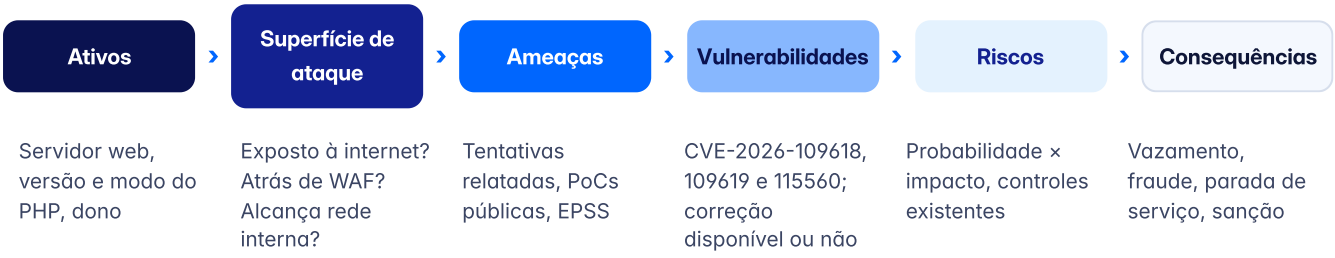
Fornecedores: sinais rápidos e úteis, geralmente sem contagens nem metodologia. Use-os para antecipar decisões e registre a origem de cada sinal.

09 — DO ACHADO AO RISCO · GESTÃO E GOVERNANÇA

Uma lista de vulnerabilidades não decide nada. Um risco contextualizado tem dono e prazo.

Transformar o PHP2Shell em decisão exige ligar a falha técnica ao que ela ameaça. O modelo abaixo é conceitual: nenhuma ferramenta descobre e valida sozinha todos esses relacionamentos. Parte vem de integrações, parte de cadastro e parte do julgamento de quem conhece o negócio.

Figura 13. Cadeia de contextualização do risco



Quatro estratégias de tratamento, nenhuma automática

MITIGAR
Atualizar e reiniciar; aplicar medidas temporárias enquanto isso. Para ramos sem release, backport ou migração.

EVITAR
Isolar da internet ou desligar o serviço quando não pode ser corrigido nem compensado, como recomendam CERT-EU e CCCS.

TRANSFERIR
Exigir correção contratual de hospedagem e terceiros; avaliar cobertura de seguro. Transfere parte da consequência, não a exposição.

ACEITAR
Só com justificativa, autoridade competente, controle compensatório verificado, prazo e revisão. Nunca como padrão para o que "não tem patch".

O que a governança precisa registrar

Para cada ativo afetado: a decisão, quem a tomou e com qual alçada, a justificativa, os controles compensatórios e como foram verificados, o prazo e a data de revisão. Esse registro permite responder a auditores, reguladores e clientes, e impede que um aceite temporário vire permanente por esquecimento.

Plataformas que integram vulnerabilidades, ativos, ameaças e contexto de negócio num mesmo modelo de risco, categoria em que o **Eagle** se posiciona, existem para sustentar esse encadeamento: tirar a decisão da planilha e do e-mail, atribuir dono e prazo e acompanhar o tratamento até o fechamento. O valor está no registro rastreável da decisão, não em substituir o julgamento de quem decide.

As capacidades específicas e seu estado de validação estão na seção 11.

10 — BUSINESS IMPACT ANALYSIS E QUANTIFICAÇÃO

Um servidor "comum" pode carregar um processo crítico inteiro.

A criticidade técnica de um servidor descreve sua configuração. A criticidade de negócio descreve o que depende dele. O portal de clientes em PHP, a API que alimenta o aplicativo, o sistema interno de cotação: em cada caso, a mesma falha produz consequências de natureza e magnitude diferentes.

Figura 14. Da dependência técnica à consequência operacional



Exemplo ilustrativo. A BIA define o tempo máximo tolerável de parada (MTPD), RTO e RPO de cada processo, que passam a ser critérios de priorização da correção.

Integrar BIA e gestão de exposição permite ordenar a correção pelo que se perde, e não só pelo que é vulnerável. Também revela dependências pouco óbvias: um servidor interno, não exposto, mas que guarda as credenciais usadas por um servidor exposto, entra na mesma decisão. Onde a plataforma de gestão de riscos já relaciona ativos a processos de negócio, essa leitura deixa de ser um exercício manual em cada evento.

Quantificação: útil quando há dados, enganosa quando é inventada

AVALIAÇÃO QUALITATIVA
Matriz de probabilidade e impacto em escalas. Rápida e suficiente para a decisão emergencial.

PRIORIZAÇÃO TÉCNICA
Ordena correções por gravidade, exposição e exploração. Não estima perda.

ESTIMATIVA QUANTITATIVA
Faixas de frequência e magnitude de perda, com incerteza explícita. Apoia aceite, investimento e seguro.

PERDA OBSERVADA
O que de fato custou um incidente. É dado histórico, não previsão.

Este documento não apresenta estimativa de perda para o PHP2Shell

Não há dados públicos suficientes, e qualquer número seria ilustrativo. Para quantificar o cenário em uma organização específica, seriam necessários: o inventário de ativos afetados e sua exposição; os registros e volumes de dados processados; a frequência de tentativas na própria telemetria; a eficácia verificada dos controles compensatórios; os custos de parada derivados da BIA; e premissas sobre resposta, notificação e sanções. O EPSS pode informar a frequência de forma genérica, mas não substitui dados do ambiente.[11] Resultados de simulação são distribuições de possibilidade, não previsões.

11 — CAPACIDADES NECESSÁRIAS E O PAPEL DO EAGLE

O que o caso exige de uma organização e onde uma plataforma integrada contribui.

O Eagle é uma plataforma de governança, gestão de riscos cibernéticos e compliance. Segundo a descrição institucional, ele integra e contextualiza dados de ferramentas técnicas e não substitui scanners de vulnerabilidade.[27] O quadro relaciona cada problema evidenciado pelo PHP2Shell à capacidade correspondente.

Figura 15. Quadro de capacidades

Problema	Capacidade	Módulo Eagle	Contribuição	Estado	Evidência
Não saber onde o PHP roda nem quem responde	Inventário de ativos com dono	Cyber Exposure Management	Consolidar ativos, versões e responsáveis num registro usado pelas demais análises.	NÃO VERIFICADO	Descrição institucional[27]
Achados de scanners dispersos	Integração com gestão de vulnerabilidades	Cyber Exposure Management	Associar achados de scanners aos ativos. Conectores específicos dependem de integração configurada.	NÃO VERIFICADO	Descrição institucional[27]
Priorizar sem saber o estado da exploração	Threat Intelligence aplicada à exposição	Cyber Exposure Management	Associar sinais de ameaça e criticidade do ativo à priorização.	NÃO VERIFICADO	Descrição institucional[27]
Criticidade avaliada só pela técnica	Relação ativo ↔ processo de negócio	Business Continuity Management (BIA)	Ligar ativos a processos, MTPD, RTO e RPO para ordenar a correção pela consequência.	NÃO VERIFICADO	Descrição institucional[27]
Decisão de tratamento sem registro	Gestão e governança de riscos	Cyber Risk Management	Modelar cenários do risco inerente ao residual, com controles e tratamento.	NÃO VERIFICADO	Descrição institucional[27]
Compensações sem verificação	Controles e evidências	Governance, Risk & Compliance	Acompanhar controles, evidências e planos de ação.	NÃO VERIFICADO	Descrição institucional[27]
Terceiros que operam PHP	Risco de terceiros	Third-Party Risk Management	Registrar fornecedores, criticidade e planos de ação.	NÃO VERIFICADO	Descrição institucional[27]
Decisões sem tradução econômica	Quantificação de risco	Cyber Risk Management	Avaliação qualitativa e quantitativa quando houver dados.	NÃO VERIFICADO	Descrição institucional[27]

"Não verificado" significa que a validação técnica (código, testes, evidências de funcionamento) não foi realizada nesta revisão — não que a funcionalidade inexista. As capacidades estão descritas no material institucional da Futture.ai.

Fora do escopo da plataforma neste cenário

Bloquear tráfego de exploração, aplicar patches, reiniciar serviços e detectar exploração em tempo real são funções de WAF, gestão de patches, EDR e SIEM. Não há evidência, nesta publicação, de que o Eagle identifique automaticamente o PHP2Shell nos ativos de um cliente.

12 — RECOMENDAÇÕES TÉCNICAS

Corrigir é a única solução completa. O resto compra tempo.

As ações estão organizadas pelo tipo de efeito que produzem. Correção definitiva remove a falha; as demais reduzem exposição ou apoiam a investigação, com limites que o próprio PHP declara.[1][3][4] Em caso de dúvida sobre um parâmetro específico, prevalece o advisory oficial.

1 · Correção definitiva

REMOVE A FALHA

- **Inventariar todo PHP**: ramos paralelos, contêineres, painéis de hospedagem, appliances e software de terceiros, com versão, modo de execução, exposição e dono. O CCCS recomenda identificar produtos e appliances que embarcam PHP.[13]
- **Atualizar** para 8.5.12, 8.4.27, 8.3.36 ou 8.2.35, ou para o pacote corrigido do fornecedor, conferindo pela versão do pacote e pelo advisory da distribuição.[1][17] Ramos 7.4–8.1: backport ou migração. Reconstruir imagens de contêiner.
- **Reiniciar todos os processos** que carregam PHP e confirmar que os workers em execução usam o build corrigido — não o `php -v` da linha de comando.[17]
- **Priorizar** ativos voltados à internet com mod_php; depois os demais expostos; por fim os internos, ordenados pela criticidade de negócio.[1][13]

2 · Mitigação temporária

BLOQUEIA CAMINHOS ESPECÍFICOS · NÃO TORNA A INSTALAÇÃO SEGURA

CVE-2026-109619 e CVE-2026-115560. Definir `session.upload_progress.enabled=0` em php.ini, no pool do FPM ou na configuração do servidor — não via `ini_set()` ou `.user.ini` — e reiniciar todos os workers. Bloqueia os caminhos de sessão confirmados, não outras variantes. Efeito colateral: o acompanhamento de upload por sessão deixa de funcionar. Manter mesmo após atualizar, até a próxima release.[1][2][4]

PHP 8.5. Definir `fatal_error_backtraces=0` administrativamente e reiniciar os workers. Remove o produtor do caso confirmado de RCE; não cobre outros estados.[4]

CVE-2026-109618. `memory_limit=-1` imposto administrativamente (por exemplo, `php_admin_value`) remove o limite controlável explorado no caso confirmado, mas não é mitigação completa: falhas de alocação subjacentes ainda alcançam o caminho vulnerável.[3] **Risco operacional:** retira o teto de memória por requisição e aumenta a chance de indisponibilidade; só faz sentido como medida curta, combinada com limite de memória por worker (cgroup) e limites de requisição no frontend. Não usar `RLIMIT_AS` ou `RLIMIT_DATA`, que provocam justamente a falha de alocação vulnerável.[3]

3 · Redução de exposição

DIMINUI PROBABILIDADE · ÚTIL MESMO APÓS O PATCH

- **Limites de requisição.** `post_max_size` conservador — o PHP sugere 8M sem upload e 32M com upload, com `upload_max_filesize` abaixo dele — e limites no frontend para tamanho decodificado, número de campos, partes multipart, cabeçalhos por parte e aninhamento, inclusive em requisições sem `Content-Length` utilizável. Bloquear multipart onde a aplicação não o usa. Validar compatibilidade com fluxos de upload legítimos antes de aplicar.[1][3]
- **Configuração por requisição.** Remover diretivas inválidas ou obsoletas aplicadas antes da inicialização (`php_value`, `.htaccess`, PHP_VALUE enviado pelo frontend). Ocultar diagnósticos com `error_reporting`, `display_errors` ou `@` não protege.[4]
- **Erros fatais alcançáveis.** Corrigir erros fatais e timeouts que um atacante consegue provocar, inclusive em callbacks, uploads, importações e rotinas de limpeza.[1][4]
- **FPM.** Se o desempenho permitir, `pm.max_requests=1` limita cada worker a uma requisição. Opcional e com custo relevante; não substitui a correção.[4]
- **WAF.** Útil como paliativo, nunca como correção: a Cloudflare observou tentativas de contornar suas regras e não cobre aplicações que alteraram o nome do campo de progresso de upload. Tráfego direto à origem contorna o WAF.[16]

12 RECOMENDAÇÕES TÉCNICAS · CONTINUAÇÃO

4 · Monitoramento e investigação

DETECTA, NÃO PREVINE

- **Período:** desde pelo menos 17/10, como recomenda a Amazon.[15]
- **O que procurar:** avisos e erros repetidos antes da execução de scripts; diagnósticos durante a limpeza de extensões ou do servidor; erros fatais e "Maximum execution time exceeded"; falhas concentradas em rotas, tipos de conteúdo ou uploads controlados por terceiros.[4] Processos, comandos, arquivos, contas ou conexões de saída inesperados ligados ao PHP ou ao servidor web.[13]
- **Limites:** ausência de erro em log não prova segurança; a auditoria cobre só as rotas exercitadas.[4] Havendo indício, tratar como incidente e preservar evidências antes de corrigir.
- **Segredos:** dada a divulgação de memória confirmada, avaliar a rotação de credenciais, tokens e chaves usados por servidores expostos que ficaram vulneráveis.[4] INFERIDO

5 · Isolamento ou retirada de serviço

QUANDO NADA ACIMA É VIÁVEL

Para ativos que não podem ser corrigidos nem compensados — incluindo ramos 7.3 e anteriores —, CERT-EU e CCCS recomendam isolar aplicações expostas de redes públicas até que a correção seja aplicada e verificada.[12][13] A decisão tem impacto de negócio e deve seguir a alçada definida na governança.

Validação e retorno

Confirmar que os processos em execução usam o build corrigido; repetir a verificação; só então reverter medidas emergenciais como `memory_limit=-1` e `pm.max_requests=1`. Manter `session.upload_progress.enabled=0` até a release que corrigir CVE-2026-115560, se o recurso não for necessário.[2][17]

13 — RECOMENDAÇÕES EXECUTIVAS E ESTRUTURAIS

O próximo PHP2Shell vai acontecer. A diferença será o que já estiver pronto.

Para a liderança, o caso pede duas respostas: decisões rápidas sobre o evento atual e investimentos que encurtem a próxima janela. O roteiro organiza as segundas em quatro horizontes.

Figura 16. Roteiro de fortalecimento após o PHP2Shell

	Agora · semana 1	30 dias	90 dias	Contínuo
Governança	Dono executivo do evento e alçada para retirar serviços do ar Registro formal de cada decisão de tratamento	Política de exceção ao SLA guiada por Threat Intel	Critérios de aceite de risco alinhados ao apetite	Revisão periódica de riscos aceitos
Pessoas e terceiros	Cobrança formal de hospedagens e fabricantes de appliances	Cláusulas de prazo de correção em contratos	Exercício simulado de vulnerabilidade crítica	Avaliação de terceiros por exposição
Processos	Revisão retroativa de logs desde 17/10	Ligação entre gestão de vulnerabilidades e BIA	Plano de migração de ramos sem suporte	Lições aprendidas incorporadas aos playbooks
Tecnologia e dados	Inventário do runtime PHP com dono	Integração de scanners ao inventário e ao modelo de risco	Gestão contínua da superfície de ataque	Indicadores por etapa da janela no painel executivo

MARCO SEMANA 1
Todos os expostos corrigidos, compensados ou isolados

MARCO 30 DIAS
Nenhum PHP sem dono no inventário

MARCO 90 DIAS
Prioridade calculada com exposição e negócio

CONTÍNUO
Exposição residual medida e reportada

Perguntas que o conselho e a diretoria devem fazer

- Quantos ativos expostos à internet rodam PHP e quantos já têm remediação validada?
- Quais estão em versões sem correção oficial, e qual é o plano para cada um?
- Que riscos foram aceitos, por quem, com qual controle compensatório e até quando?
- Quais terceiros e appliances operam PHP em nosso nome e o que confirmaram?
- Revisamos sinais de exploração desde 17/10? Com que cobertura de logs?
- Quanto tempo levamos para responder à pergunta "onde estamos expostos"?

CONCLUSÃO

A correção chegou em um dia. A exposição dura o que durar o desconhecimento.

O PHP2Shell combina características que raramente aparecem juntas: falhas no runtime e não na aplicação, ausência de pré-requisito de acesso, uma base instalada ampla e envelhecida, e um vazamento que entregou a atacantes e defensores a mesma informação ao mesmo tempo. Dois dias depois, uma falha relacionada no mesmo recurso mostrou que "corrigido" pode ser um estado provisório.

O caso também mostra o valor da divulgação coordenada que funcionou em parte: distribuições, provedores de nuvem e WAFs receberam informação antes do anúncio. Quem opera PHP próprio, legado ou embarcado em produtos de terceiros dependeu da própria capacidade de encontrá-lo.

Não há, até aqui, estudos que meçam quais organizações responderam mais rápido e por quê. O que a mecânica do evento permite afirmar é mais modesto e mais útil: cada etapa da resposta — localizar o runtime, entender a exposição, estimar a consequência, decidir e registrar — depende de informação que precisa existir antes do advisory. Onde ela não existe, a resposta começa por produzi-la, e esse tempo soma à janela.

Gestão de vulnerabilidades continua indispensável. O PHP2Shell evidencia que ela precisa operar dentro de uma gestão de exposição e de risco que conecte a falha técnica ao ativo, à ameaça, ao processo e à decisão.

REFERÊNCIAS TÉCNICAS

Fontes numeradas, com data de publicação e de consulta

Tipos: 1 Oficial/mantenedor 2 Autoridade 3 Pesquisador original 4 Relatório técnico 5 Cobertura jornalística. Todas consultadas em 23/10/2026; "direta" indica leitura da página original.

#	Organização · título	Publicação	Tipo	URL
1	The PHP Group · Security advisory: Critical remote code execution vulnerabilities, update now	21/10/2026	1 direta	https://www.php.net/archive/2026.php#2026-10-21-5
2	The PHP Group · Informational advisory: Session upload progress (CVE-2026-115560)	23/10/2026	1 direta	https://www.php.net/archive/2026.php#2026-10-23-1
3	The PHP Group (GitHub) · GHSA-w8hw-w9fx-j79w — RCE via OOM corruption (CVE-2026-109618)	21/10/2026	1 direta	https://github.com/php/php-src/security/advisories/GHSA-w8hw-w9fx-j79w
4	The PHP Group (GitHub) · GHSA-6wmf-4rv5-p3rp — RCE via stale global state (CVE-2026-109619)	21/10/2026	1 direta	https://github.com/php/php-src/security/advisories/GHSA-6wmf-4rv5-p3rp
5	oss-security (Jakub Zelenka, PHP) · CVE-2026-109618: PHP 8.2–8.5 RCE via OOM corruption	21/10/2026	1 direta	https://www.openwall.com/lists/oss-security/2026/10/21/6
6	oss-security (Greenbone) · Re: CVE-2026-109618 — faixa de versões citada pela CrowdStrike (a partir da 7.3.0)	23/10/2026	4 direta	https://www.openwall.com/lists/oss-security/2026/10/23/5
7	NIST · NVD — CVE-2026-109618 Detail (modificado em 23/10)	21/10/2026	2 direta	https://nvd.nist.gov/vuln/detail/CVE-2026-109618
8	NIST · NVD — CVE-2026-109619 Detail (modificado em 23/10)	21/10/2026	2 direta	https://nvd.nist.gov/vuln/detail/CVE-2026-109619
9	CVE Program · CVE-2026-115560 (estado: RESERVED)	—	1 direta	https://www.cve.org/CVERecord?id=CVE-2026-115560
10	CISA · Known Exploited Vulnerabilities Catalog (feed JSON, versão 2026.10.23)	23/10/2026	2 direta	https://www.cisa.gov/sites/default/files/feeds/known_exploited_vulnerabilities.json
11	FIRST · EPSS API — CVE-2026-109618 e CVE-2026-109619	23/10/2026	4 direta	https://api.first.org/data/v1/epss?cve=CVE-2026-109618,CVE-2026-109619
12	CERT-EU · Security Advisory 2026-018 — Critical Vulnerabilities in PHP (v1.2)	20–23/10/2026	2 direta	https://cert.europa.eu/publications/security-advisories/2026-018/
13	Canadian Centre for Cyber Security · AL26-026 — Critical vulnerabilities impacting PHP (Update 1)	21–23/10/2026	2 direta	https://cyber.gc.ca/en/alerts-advisories/al26-026-critical-vulnerabilities-impacting-php-cve-2026-109618-cve-2026-109619
14	Wiz Research · PHP2Shell: AI uncovers critical RCE vulnerabilities affecting 70% of the Internet	21/10/2026	3 direta	https://wiz.io/blog/php2shell-php-rce

REFERÊNCIAS TÉCNICAS · CONTINUAÇÃO

#	Organização · título	Publicação	Tipo	URL
15	Amazon Threat Intelligence (AWS Security Blog) · Amazon Threat Intelligence identifies PHP2Shell exploitation prior to public disclosure in MadPot	21/10/2026	3 direta	https://aws.amazon.com/blogs/security/amazon-threat-intelligence-identifies-php2shell-exploitation-prior-to-public-disclosure-in-madpot/
16	Cloudflare · Cloudflare deploys protections against php2shell vulnerability (atualizado em 23/10)	21/10/2026	3 direta	https://blog.cloudflare.com/php-vulnerabilities-waf-protection
17	Qualys Threat Research Unit · Two Critical PHP Flaws Could Hand Attackers Your Web Servers (atualizado em 22/10)	20/10/2026	4 direta	https://blog.qualys.com/vulnerabilities-threat-research/2026/10/20/php-announces-critical-security-release-for-two-vulnerabilities
18	watchTowr · Frequently Asked Questions About PHP Remote Code Execution (PHP2Shell)	21–22/10/2026	4 direta	https://watchtowr.com/intelligence/php-cve-2026-109618-faq/
19	W3Techs · Usage statistics and market share of PHP for websites	23/10/2026	4 direta	https://w3techs.com/technologies/details/pl-PHP
20	CybelAngel · 2.24M Internet-Facing PHP Hosts, 59% on Versions With No Fix Coming (dados Shodan publicados pela Mallory em 20/10)	21/10/2026	4 direta	https://cybelangel.com/blog/critical-php-rce-vulnerabilities-cve-2026-109618-and-cve-2026-109619/
21	Field Effect · Another PHP issue identified while vendors report affected products	23/10/2026	4 direta	https://fieldeffect.com/blog/another-php-issue-identified-report-affected-products
22	SC World · CISA advises orgs to patch PHP2Shell bugs immediately (comunicado da CISA não localizado em cisa.gov nesta revisão)	21/10/2026	5 direta	https://www.scworld.com/news/cisa-advises-orgs-to-patch-php2shell-bugs-immediately
23	ITPro · Update issued for critical PHP vulnerabilities: security agencies warn patch now or take systems offline (orientação do NCSC)	21/10/2026	5 direta	https://itpro.com/security/fixes-issued-for-critical-php-vulnerabilities-security-agencies-warn-patch-now-or-take-systems-offline
24	Dark Reading · 'PHP2Shell' Bugs Signal Red Alert for Security Industry (entrevista com watchTowr)	21/10/2026	5 direta	https://www.darkreading.com/application-security/php2shell-bugs-red-alert-security-industry
25	BleepingComputer · PHP warns admins of two critical flaws, emergency patches coming tomorrow	20/10/2026	5 via busca	https://bleepingcomputer.com/news/security/php-warns-admins-of-two-critical-flaws-emergency-patches-coming-tomorrow
26	MITRE · ATT&CK T1190 Exploit Public-Facing Application (também T1505.003, T1552, T1496)	—	Framework	https://attack.mitre.org/techniques/T1190/
27	Futture.ai · Eagle: Plataforma GRC e Gestão de Riscos com IA (descrição institucional dos módulos)	—	Institucional	https://www.futture.ai/pt/eagle.html

Limitações das fontes

O comunicado da CISA e a orientação do NCSC foram obtidos por cobertura jornalística; as páginas originais não foram localizadas nesta revisão. A leitura de "via busca" indica que o conteúdo foi verificado por resultados de busca, não pela página aberta. Relatos de fornecedores sobre o que observaram em suas próprias redes são fontes primárias para o fato observado, mas não confirmação independente. Status no KEV, EPSS e indicadores de comprometimento mudam com frequência e devem ser rechecados a cada republicação.

SOBRE O EAGLE



FUTTURE.AI

futture.ai

Conteúdo educativo sobre risco, exposição e governança: futture.ai/blog

Eagle

Plataforma de gestão integrada de segurança, riscos cibernéticos e conformidade, desenvolvida pela Futture.ai.

Organizações acumulam ferramentas de segurança que não conversam entre si, dados espalhados e relatórios que já nascem desatualizados. O Eagle foi concebido para conectar essas ferramentas em vez de substituí-las, relacionar achados técnicos a ativos, ameaças e processos de negócio, e apoiar decisões de risco documentadas.

Seus módulos descritos publicamente são Governance, Risk & Compliance; Cyber Risk Management; Cyber Exposure Management; Third-Party Risk Management; e Business Continuity Management.[27]

Este documento integra a linha editorial da Futture.ai sobre eventos de segurança relevantes para o mercado. A prioridade é a precisão: o que está confirmado, o que ainda é relato e o que cada organização precisa decidir.

Conectar
ferramentas existentes, sem trocá-las

Contextualizar
achados técnicos com ativos e negócio

Decidir
com dono, prazo e justificativa

Acompanhar
o tratamento até o fechamento

Descrição institucional baseada no guia de marca e no site da Futture.ai. Estado de validação das capacidades: ver seção 11.