

PHP2Shell: l'embargo si è rotto prima della patch, e chi ha aggiornato subito ha comunque una settimana cieca alle spalle

Il 21 ottobre 2026, alle 14:00 UTC, il progetto PHP ha pubblicato quattro versioni correttive per due vulnerabilità critiche nell'interprete. Sono la CVE-2026-109618 e la CVE-2026-109619, entrambe use after free, entrambe con punteggio CVSS v4.0 pari a 9,2, entrambe sfruttabili da remoto senza autenticazione. Il nome con cui circolano è PHP2Shell.

La parte tecnica è grave e si riassume in fretta. La parte che merita un articolo è un'altra: quando le patch sono uscite, qualcuno stava già provando a sfruttarle da giorni. Gli honeypot di Amazon registrano un tentativo il 17 ottobre. Cloudflare ha attivato le protezioni alle 03:02 UTC del 21, nove ore prima della divulgazione, e dichiara di avere osservato traffico di attacco contro i propri clienti. La divulgazione coordinata, in questo caso, non ha coordinato nulla.

La conseguenza operativa è contro intuitiva. Un'organizzazione che ha aggiornato il 21 ottobre entro poche ore ha fatto la cosa giusta e non ha chiuso il problema. Ha chiuso la finestra futura e si è lasciata alle spalle una finestra passata che nessun avviso italiano la aiuta a verificare.

Che cosa rompono le due

vulnerabilità

Entrambe nascono dallo stesso punto debole, ed è il modo in cui PHP gestisce gli errori fatali mentre sta elaborando una richiesta, in particolare l'esaurimento della memoria.

L'advisory GHSA-w8hw-w9fx-j79w, pubblicato il 21 ottobre, descrive la CVE-2026-109618. PHP usa la funzione `longjmp()` per uscire dal contesto corrente quando la memoria finisce. Se un'allocazione fallisce mentre la memoria della richiesta è già raggiungibile ma non ancora completamente inizializzata, l'uscita anticipata non viene contabilizzata. Lo spegnimento della richiesta legge allora strutture non valide. Il testo dell'advisory è asciutto: "If any allocation fails at that point, PHP does not properly account for the early exit".

Le strategie per provocare l'esaurimento passano dal parsing multipart. Una richiesta multipart senza content-length utilizzabile che supera il controllo su `post_max_size`, un numero illimitato di intestazioni per parte, controlli di file vuoti che non contano in `max_file_uploads` ma generano comunque array di metadati. Il vettore CVSS è 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.

L'advisory GHSA-6wmf-4rv5-p3rp, pari data, descrive la CVE-2026-109619. Qui lo stato globale del worker, errori, sessione, oggetti ed estensioni, resta puntato alla memoria della richiesta precedente dopo che quella memoria è stata liberata. Una richiesta successiva, servita dallo stesso worker, usa quei riferimenti prima che l'inizializzazione li azzeri. L'advisory lo scompone in produttore e consumatore: il produttore è l'operazione che lascia pubblicato lo stato, il consumatore è quella che lo rilegge dopo.

Il caso dimostrato più forte è PHP 8.5 con `fatal_error_backtraces` attivo, ed è la base dello sfruttamento confermato su `mod_php`. Il secondo caso confermato è il ciclo

di vita del session upload progress, validato sia su mod_php sia su PHP-FPM, con un innesco indipendente dall'applicazione.

Questa è la differenza che conta nel raccogliere le prove. Per la 109618 l'esecuzione di codice completa è stata validata via HTTP contro mod_php, con limiti di memoria di default. PHP-FPM è indicato come suscettibile sulla base dell'analisi del sorgente, senza un exploit completo dimostrato. Per la 109619 il percorso del session upload progress è validato su entrambe le configurazioni.

Gli advisory estendono il quadro oltre le due configurazioni più note. Per la 109618 FrankenPHP, come mod_php, richiama la procedura di spegnimento in modo incondizionato, quindi è indicato come probabilmente suscettibile ad attacchi indipendenti dall'applicazione. LSAPI segue il modello FPM, con exploit specifici dell'applicazione ritenuti probabili ma non validati.

Per la 109619 c'è un'esposizione di PHP-FPM che riguarda molte installazioni reali. È condizionata al frontend: nginx, Apache mod_proxy_fcgi o altro, quando inviano parametri PHP_VALUE o PHP_ADMIN_VALUE che generano una diagnostica prima dell'inizializzazione della richiesta.

Un dettaglio del percorso session upload progress merita attenzione perché smonta un'ipotesi comune. La funzione è attiva per impostazione predefinita e non richiede alcuna chiamata a session_start() da parte dell'applicazione. Non serve che il sito usi le sessioni.

La correzione, secondo l'advisory, rimuove longjmp() per le condizioni di esaurimento memoria. Al suo posto introduce un'uscita ritardata e ordinata al successivo punto sicuro della macchina virtuale, più un'uscita immediata con _exit() oltre una seconda soglia. Lo spegnimento non vede più stato non valido.

L'avviso di sicurezza PHP del 21 ottobre non lascia spazio:

non esistono mitigazioni complete, aggiornare è l'unica scelta sicura. Indica `mod_php` come configurazione a rischio più alto, con un exploit dimostrato in cui una richiesta non autenticata colpisce prima che venga eseguito qualunque codice applicativo.

C'è poi un avvertimento che vale la pena isolare, perché riguarda una misura che molti considerano irrigidimento. L'advisory della 109618 raccomanda di non usare `RLIMIT_AS` né `RLIMIT_DATA`. Quei limiti fanno fallire `mmap()` una volta raggiunta la soglia, e il fallimento attiva proprio il percorso `longjmp()` vulnerabile.

La mitigazione di configurazione indicata come principale è l'opposta, cioè `memory_limit=-1` imposto in modo che le applicazioni non possano sovrascriverlo. Rimuove il limite prevedibile che rende affidabile l'exploit confermato, e non è comunque completa. Sansec segnala il contrappeso: con `memory_limit=-1` una singola richiesta può consumare tutta la memoria del server.

Sul fronte delle dimensioni, l'avviso PHP dà numeri concreti. Dove non servono caricamenti di file, `post_max_size` a 8M. Dove servono, 32M, con `upload_max_filesize` sempre inferiore.

Il punto che gli amministratori sbagliano per primo

Il difetto sta nell'interprete, non nell'applicazione. Cloudflare lo scrive in una riga che vale un capitolo. Aggiornare il codice del sito o i plugin non corregge nulla, va aggiornato il software PHP che fa girare l'applicazione. Se è l'hosting a gestirlo, va chiesto all'hosting.

Wiz Research lo porta all'estremo: anche una semplice applicazione "Hello World" su una versione affetta può essere vulnerabile. Non serve una funzione esposta, non serve un

plugin scritto male, non serve un modulo di upload nel codice dell'applicazione.

Per una redazione che segue il parco italiano della pubblica amministrazione questo è il passaggio con più ricadute. A differenza del caso wp2shell, che ha coinvolto direttamente il core di WordPress, PHP2Shell interessa l'interprete sottostante, indipendentemente dal CMS utilizzato. Significa che l'inventario dei CMS, dei temi e dei plugin, che è il censimento che quasi tutti hanno, non risponde alla domanda. La domanda è quale versione dell'interprete serve le richieste, e la risposta può differire da quella restituita dalla riga di comando.

La cronologia, e il punto in cui si rompe

Le fonti disponibili permettono di ricostruire la sequenza, con due tracce di scoperta indipendenti.

Sansec, il 21 ottobre, colloca al 1 agosto 2026 la scoperta da parte di Calif di più percorsi di esecuzione di codice, e all'11 agosto la prima segnalazione al progetto PHP. Wiz, indica una propria cronologia distinta. Scoperta il 19 agosto, conferma dell'esecuzione di codice non autenticata su Drupal il 20 agosto e su WordPress il 21. Segnalazione al team di sicurezza PHP il 24 agosto, riconoscimento e avvio della correzione coordinata il 21 settembre.

I due advisory GitHub riportano come segnalanti lcfr-hax, sagitzadik-wiz, tomash-areion e amyb-oai. Cloudflare attribuisce la divulgazione a Calif.io, Wiz, OpenAI e Quarkslab. Sansec nomina Ryan Austin di Calif, Wiz Research e OpenAI. Wiz, dal canto suo, ringrazia il team PHP, Calif e OpenAI.

A sciogliere il nodo è il record CVE, perché il CNA è il

progetto PHP stesso. Il record della CVE-2026-109618 accredita Ryan at Calif.io e Wiz come reporter e OpenAI come finder. Quello della CVE-2026-109619 accredita tutti e tre come reporter.

La distinzione fra finder e reporter non è formale. Sulla prima vulnerabilità il progetto riconosce a OpenAI il ritrovamento, non la sola segnalazione, e questo dà sostanza alla tesi di Wiz sul peso degli strumenti automatici. Come sviluppatori della correzione i record nominano Arnaud Le Blanc, Ilija Tovilo, Jakub Zelenka e Nora Dossche.

Quarkslab non compare in nessuno dei due record, né fra i crediti degli advisory. Al 22 ottobre il blog di Quarkslab non contiene alcun articolo su PHP2Shell e si ferma all'audit di MbedTLS del 20 ottobre. Anche il repository delle pubblicazioni di Calif non contiene nulla sulle due CVE. L'attribuzione di Cloudflare, su questo punto, non trova riscontro.

Il 20 ottobre il progetto PHP annuncia una pubblicazione straordinaria per il giorno successivo alle 14:00 UTC, senza indicare gli identificativi CVE. L'annuncio si riserva di anticipare il rilascio se i dettagli diventassero pubblici o se emergessero prove di sfruttamento attivo in rete.

Una nota sulle date, perché il progetto diverge da se stesso. L'archivio delle notizie data le quattro versioni al 21 ottobre, mentre il ChangeLog di PHP 8 assegna a tutte e quattro il 22 ottobre. La data operativa resta il 21 alle 14:00 UTC, confermata dal preavviso, dalle note di rilascio e da tutti gli atti nazionali.

Lo stesso 20 ottobre l'embargo cede. Sansec lo registra nella propria cronologia come fuga dei dettagli, con watchTowr che conferma l'esecuzione di codice. Il Rapid Reaction di watchTowr, pubblicato il 21 ottobre alle 14:19 UTC, lo descrive in termini espliciti: a differenza di una

divulgazione coordinata tipica, questa è sfuggita a tutti, ed è l'embargo a cedere per primo. Dichiarò di avere riprodotto la CVE-2026-109618 e di non averne pubblicato i dettagli prima della correzione. La stessa pagina riferisce che la divulgazione, prevista per il 22 ottobre, è stata anticipata di un giorno dopo la fuga.

Nessuna delle due fonti dice da dove sia uscita l'informazione. L'advisory del CERT-EU, però, cita come segnalazioni pubbliche precedenti al rilascio un messaggio su Mastodon del ricercatore noto come GossiTheDog e un messaggio su X dell'account di watchTowr. Sono quelle le tracce pubbliche del 20 ottobre.

Una seconda divergenza di orario. Il CERT-EU scrive che il team PHP ha rilasciato le correzioni alle 13:30 UTC, mentre il preavviso del progetto, il CERT-FR e il BSI indicano le 14:00 UTC, cioè le 16:00 ora italiana. Mezz'ora, influente sulla sostanza, indicativa di quanto fosse concitata la giornata.

Un elemento di contesto va riportato senza trarne conclusioni. L'avviso PHP scrive che le distribuzioni Linux e i fornitori con supporto a lungo termine hanno ricevuto le patch in anticipo, e invita a verificare presso il proprio fornitore. Cloudflare dichiara di avere sviluppato le protezioni dall'inizio di ottobre dentro uno sforzo coordinato. Il cerchio degli informati, prima del 21 ottobre, era quindi largo. Nessuna fonte collega la fuga a uno di questi soggetti, e questo articolo non lo fa.

Chi ha visto cosa, e quando

Qui le fonti vanno tenute separate, perché dicono cose diverse e la differenza è sostanziale.

I due advisory GitHub, al momento della pubblicazione, non riportano sfruttamento in rete. Non lo riporta il CERT-EU nell'advisory 2026-018 del 21 ottobre, che si limita a

osservare che i primi proof of concept hanno iniziato a circolare e che questo aumenta la probabilità che gli attaccanti stiano già sfruttando. Non lo riportano il NCSC britannico né il BSI tedesco. Il vicepresidente del BSI Thomas Caspers formula una previsione, non una constatazione: uno sfruttamento su larga scala entro poche ore dalla pubblicazione degli aggiornamenti è ritenuto probabile.

Due aziende dicono invece di avere visto qualcosa, e nessuna delle due parla di compromissioni riuscite.

Amazon, il 21 ottobre, riferisce un tentativo di sfruttamento registrato dagli honeypot MadPot il 17 ottobre 2026, quando le vulnerabilità erano ancora sotto embargo, proveniente da una rete di anonimizzazione commerciale. Colloca altri tentativi fra le 02:00 e le 08:00 UTC del 21 ottobre, quindi prima del rilascio. Pubblica sei indirizzi IP osservati e raccomanda di indagare su quelle tracce invece di bloccarle, perché appartengono a infrastruttura condivisa. Precisa due limiti: non è chiaro se i tentativi sarebbero riusciti su sistemi reali, e non è noto come l'attore abbia ottenuto le informazioni prima della divulgazione. Aggiunge che payload e infrastrutture variavano, il che suggerisce più attori distinti.

Cloudflare, nello stesso giorno, dichiara di avere sviluppato le protezioni del firewall applicativo a partire dall'inizio di ottobre 2026, nell'ambito di uno sforzo coordinato con la PHP Foundation, OpenAI e Wiz. Le ha distribuite circa nove ore prima della divulgazione pubblica, dalle 03:02 UTC del 21 ottobre. Scrive di avere osservato traffico di attacco contro i propri clienti mentre la vulnerabilità era trapelata prima della data ufficiale, e di avere visto gli attaccanti mutare il payload per aggirare le protezioni. Non fornisce volumi, indirizzi o finestra temporale.

watchTowr dichiara di osservare tentativi attivi su entrambe le CVE, senza date, volumi o indicatori. Nella propria tabella

marca come sfruttata in rete la sola CVE-2026-109619.

La lettura della redazione è che il quadro probatorio sia asimmetrico e vada detto così. Esistono tentativi documentati con data e indirizzi su honeypot, esistono dichiarazioni di traffico osservato su una rete di distribuzione, e non esiste al momento nessuna compromissione dichiarata su sistemi reali. Trattare il primo insieme come prova di compromissioni diffuse sarebbe un errore. Trattarlo come irrilevante sarebbe l'errore opposto, perché quattro giorni di tentativi precedenti alla patch sono esattamente il periodo che nessuno sta guardando.

Fin dove arriva il fine supporto, e qui le fonti divergono

È la seconda discrepanza misurabile della vicenda, e tocca direttamente chi ha parco legacy.

I due advisory GitHub elencano come affetti i soli rami dalla 8.2 alla 8.5, con le correzioni nelle versioni 8.2.35, 8.3.36, 8.4.27 e 8.5.12.

Il CERT-FR, con l'alerte CERTFR-2026-ALE-014 del 21 ottobre, va molto oltre. Elenca fra i sistemi affetti tutte le versioni 5.x, 6.x, 7.x, 8.0.x e 8.1.x, oltre ai rami ancora supportati.

Va segnalato un difetto del documento francese, perché cade proprio sul punto in questione. Nella sezione riassuntiva la frase che dovrebbe spiegare il trattamento dei rami fuori supporto resta interrotta a metà. L'elenco dei sistemi affetti è esplicito, la motivazione no. Il BSI afferma che tutte le versioni note, risalendo almeno fino a PHP 5, risultano vulnerabili. Il CERT-EU indica come affetti anche i rami oltre il supporto. Wiz scrive che ogni versione PHP pubblicata nell'ultimo decennio è affetta da almeno una delle due, risalendo alla 5.4. Sansec colloca il codice vulnerabile nella versione 5.4.0, pubblicata nel 2012.

L'ACN, nell>alert nazionale, dichiara che per tutte le versioni di PHP 8.1 e precedenti il fornitore non rilascerà alcun workaround né patch, per fine supporto.

Il documento che pesa di più, però, dice un'altra cosa, ed è del progetto stesso. Nei record CVE, dove il CNA è PHP Group, lo stato predefinito dichiarato è "unaffected" e le uniche versioni elencate come affette sono i quattro rami dalla 8.2 alla 8.5. Formalmente, nella fonte macchina leggibile che alimenta scanner e inventari, i rami a fine supporto non risultano affetti.

Lo stesso vale per l'annuncio che Jakub Zelenka, del PHP security team, ha inviato il 21 ottobre alle 16:44 alla lista oss-security. Elenca le versioni affette e si ferma ai quattro rami supportati. Il titolo stesso del messaggio dice PHP 8.2 fino a 8.5.

La contraddizione è stata sollevata in pubblico, e non da questa redazione. La mattina del 22 ottobre, alle 8:45, Christian Fischer di Greenbone risponde in lista a Zelenka osservando che oggetto e testo possono essere letti come se solo le versioni dalla 8.2 alla 8.5 fossero affette. Obietta che l'esistenza dei backport fino alla 7.4 implica che siano affette almeno le versioni fino a quel ramo.

Al momento della chiusura di questo pezzo la domanda è senza risposta sulla lista. Non è una disputa accademica: decide se chi gestisce un parco su PHP 7.4 debba considerarsi esposto, e decide cosa segnalerà uno scanner che legge il record CVE invece dell>alert nazionale.

Qui c'è una precisazione che gli atti nazionali non riportano e che cambia le opzioni di chi deve decidere. L'avviso PHP del 21 ottobre scrive che il team di sicurezza ha fornito backport minimali del codice sorgente per PHP 7.4, 8.0 e 8.1. Non sono pacchetti pronti e non sono supporto, ma non sono nemmeno il nulla che gli avvisi italiano e francese lasciano intendere.

Chi ha un applicativo bloccato su 8.1 ha una strada tecnica, scomoda, che dagli atti pubblici non apprende.

Va aggiunto che PHP 8.2 esce dal supporto il 31 dicembre 2026. Chi oggi aggiorna alla 8.2.35 e considera chiuso il capitolo ha comprato poco più di due mesi.

Sul fronte dell'inventario, il NCSC britannico e il CERT-EU rimandano entrambi al repository NCSC-NL php2shell, che raccoglie informazioni operative e liste di prodotti con dipendenza da PHP. Fra questi WordPress, Drupal, WooCommerce, le immagini Docker ufficiali di PHP, cPanel, DokuWiki, MediaWiki e WHMCS.

È la fonte che si muove più in fretta, e va consultata di nuovo prima di qualunque decisione. Nella giornata del 22 ottobre il repository è passato da due a tre cartelle. La nuova è detection_mitigation, che secondo la tabella dei contenuti ospita materiale di rilevamento e mitigazione, fra cui uno script per cPanel. I messaggi dei commit registrano la conferma di quindici ulteriori correzioni di fornitori e l'avvio di una modellazione VEX. Il repository dichiara di non essere completo.

Quell'ultimo punto vale più di quanto sembri. VEX è il formato con cui si dichiara se una vulnerabilità presente in un componente sia effettivamente sfruttabile in un prodotto. Applicarlo con ambito sul runtime, e non sull'applicazione, è la risposta formale al problema aperto più sopra: la domanda non è quale CMS sia installato, è quale interprete serve le richieste. La voce 26 del piano, dedicata al VEX, trova qui il suo caso più concreto.

Rilevanza italiana: l'atto c'è, gli indicatori no

L'ACN ha pubblicato l>alert AL07/261021 il 21 ottobre alle

16:21, portandolo alla versione 1.2 alle 19:35. L'impatto sistemico assegnato è alto, 65,64.

Due raccomandazioni escono dall'ordinario e vanno lette insieme alla cronologia ricostruita sopra.

La prima è il threat hunting retrospettivo, chiesto a prescindere dall'applicazione degli aggiornamenti, attraverso l'analisi dei log e degli eventi di sicurezza disponibili. L'ACN indica cosa cercare. Richieste HTTP anomale, errori ricorrenti di esaurimento della memoria, interruzioni inattese dei processi PHP. Poi esecuzione di processi non previsti, modifiche non autorizzate ai file applicativi, connessioni di rete anomale.

La seconda è la raccomandazione di valutare di porre offline i siti e i servizi web su versioni vulnerabili, sospendendone l'accessibilità fino all'aggiornamento. È una misura che un'autorità nazionale scrive raramente, e il NCSC britannico la formula nello stesso senso. Aggiunge che chi non può aggiornare né spegnere dovrebbe verificare la presenza di compromissioni e backdoor, eventualmente con un fornitore di risposta agli incidenti certificato.

La prima raccomandazione è interessante perché presuppone qualcosa. Chiedere di cercare tracce a prescindere dalla patch significa ammettere che la compromissione possa precedere l'avviso. È la posizione corretta, ed è coerente con il tentativo del 17 ottobre registrato da Amazon.

Il problema è che l>alert non dà a chi cerca nulla con cui cercare. Nella tabella CVE del documento, alla versione 1.2, le colonne relative alla presenza di proof of concept e allo sfruttamento risultano vuote per entrambe le vulnerabilità. Nel frattempo i proof of concept sono pubblici, Amazon ha pubblicato sei indirizzi con le date di osservazione e Cloudflare dichiara traffico ostile sui propri clienti. Le due cose non sono in contraddizione formale, perché quelle colonne

registrano le verifiche del CSIRT, ma la distanza fra il documento nazionale e la telemetria disponibile è il dato.

Al mattino del 22 ottobre le due CVE non figurano nel catalogo KEV della CISA, fermo alle aggiunte del 20 ottobre. Chi usa il catalogo come criterio di priorità, e sono molte organizzazioni anche italiane, non vede ancora PHP2Shell.

C'è infine un dato di contesto nazionale che non va perso. Il 19 ottobre, tre giorni prima, un altro alert italiano aveva usato la stessa formula sul fine supporto per le vulnerabilità di Drupal. Due volte in tre giorni il CSIRT nazionale ha dovuto scrivere che per una parte del parco installato la correzione non arriverà. La seconda volta riguarda l'interprete su cui gira la maggioranza del web.

Che cosa cercare, se avete già aggiornato

La domanda operativa che questo caso pone non è come applicare la patch, è come verificare la settimana precedente. Le fonti consultate convergono su alcune tracce, e vale la pena metterle in fila perché nessun singolo documento le contiene tutte.

Dai log del server web, richieste multipart anomale, in particolare con intestazioni per parte in numero elevato o con dimensioni che si avvicinano al limite di `post_max_size`. Amazon suggerisce di cercare richieste POST contenenti la stringa `PHP_SESSION_UPLOAD_PROGRESS` nei moduli di upload.

Dai log applicativi e di sistema, errori ricorrenti di esaurimento memoria, risposte HTTP 500 ripetute, terminazioni e riavvii inattesi dei processi worker. Wiz aggiunge due segnali meno ovvi: cookie `PHPSESSID` di risposta anomali, troppo grandi o malformati, e un ricambio inspiegabile dei worker PHP.

Sul sistema, la lista del NCSC britannico e di watchTowr: nuove web shell, cron inattesi, account di nuova creazione collegati ai processi PHP, modifiche non autorizzate ai file applicativi.

L'elenco più autorevole, però, sta nell'advisory della 109619, in una sezione dedicata all'audit della configurazione e dei log che nessun avviso nazionale riprende. Chiede di abilitare `log_errors`, un `error_log` funzionante ed `error_reporting=E_ALL` in configurazione di avvio, durante un audit isolato, e poi di rileggere i log di Apache, PHP-FPM, proxy inverso e servizi cercando cinque cose.

Avvisi, deprecazioni ed errori del core ripetuti prima che gli script di destinazione vengano eseguiti. Avvisi e altre diagnostiche prodotti durante la pulizia delle estensioni o del server. Messaggi di errore fatale, di parsing e di eccezioni non catturate. La stringa "Maximum execution time exceeded". Infine i malfunzionamenti circoscritti a rotte, tipi di contenuto, file, implementazioni di callback od opzioni di plugin controllabili da un attaccante.

Segue l'avvertenza che andrebbe letta due volte, perché limita il valore di tutto l'esercizio. Correggere gli errori osservati riduce l'esposizione, ma l'assenza di un errore nei log non è prova di sicurezza. I log storici possono essere stati raccolti escludendo le deprecazioni, un audit con `E_ALL` copre solo le rotte effettivamente esercitate, e il malfunzionamento può avvenire prima che PHP scriva il messaggio.

La lettura della redazione è che questa avvertenza sia il punto più scomodo della vicenda. L'autorità italiana chiede threat hunting retrospettivo, la fonte primaria avverte che quel threat hunting può non trovare nulla pur in presenza di compromissione. Chi risponde deve saperlo prima di firmare una relazione.

Una nota che cambia la portata del danno. Sansec osserva che la CVE-2026-109618 può anche far trapelare memoria di processo, con un paragone esplicito a Heartbleed. Fra i dati esposti, scrive, possono esserci password, token di sessione, chiavi API e chiavi private TLS. Se quel percorso è stato esercitato, l'aggiornamento non chiude l'incidente. La rotazione dei segreti diventa parte della risposta, non un di più.

La finestra da coprire parte almeno dal 17 ottobre, che è la data del primo tentativo documentato. Chi conserva log più lunghi farebbe bene a partire dall'inizio di ottobre. È il momento in cui Cloudflare dice di avere iniziato a sviluppare le protezioni, quindi il momento in cui l'informazione circolava già in forma ristretta.

Che cosa guardare

Se il CSIRT Italia valorizzi le colonne sulla presenza di proof of concept e sullo sfruttamento nell>alert AL07/261021, e se pubblichi un bollettino su compromissioni rilevate nel contesto nazionale. È la differenza fra un avviso di prodotto e un fatto italiano.

L'ingresso delle due CVE nel catalogo KEV della CISA, che per molte organizzazioni è l'interruttore che fa scattare la priorità.

Se emergano elementi su come i dettagli siano usciti il 20 ottobre. È la domanda che nessuna delle fonti consultate risponde, e da cui dipende se questo sia un incidente isolato o un problema del modello di divulgazione coordinata.

La risposta del PHP security team all'obiezione di Greenbone sui rami a fine supporto, perché finché non arriva gli atti nazionali e il record CVE dicono cose diverse.

Una seconda questione aperta sulla stessa lista, da trattare

con cautela perché tutte le fonti sono di seconda mano e nessuna è confermata. Il 21 ottobre Aaron Rainbolt scrive che, secondo una conversazione nella chat Matrix di Ubuntu, Canonical avrebbe chiesto a php.org se un problema circolato su Mastodon fosse una vulnerabilità reale, ricevendo risposta negativa. Nello stesso thread Yves-Alexis Perez riferisce il parere opposto di un'altra ricercatrice, secondo cui si tratterebbe di una vulnerabilità diversa. Nessuna conferma ufficiale risulta pubblicata.

Il repository NCSC-NL, che cambia più volte al giorno e che sta aggiungendo materiale di rilevamento e una modellazione VEX con ambito sul runtime. È lo strumento più utile per capire quali apparati e prodotti di terzi incorporino PHP, ed è anche il primo caso pratico di VEX applicato a un interprete.

La raccolta ENISA degli advisory dei CSIRT nazionali europei, utile per confrontare le posizioni nazionali sulle versioni a fine supporto, dove italiani e francesi divergono dal fornitore.

La scadenza del 31 dicembre 2026 per PHP 8.2, che sposta in avanti di pochi mesi lo stesso problema per chi oggi si è fermato alla 8.2.35.

<https://www.ictsecuritymagazine.com/cyber-security/php2shell-e-mbargo-rotto-finestra-prima-della-patch/>