AIRNITE

ZERO DAY

NUMERO 02/2026 · SICUREZZA INFORMATICA

Una backdoor nel cuore di PHP: come gli attaccanti hanno manipolato il codice sorgente del linguaggio del web

Nel marzo 2021 due commit manipolati sono finiti nel codice sorgente ufficiale di PHP, camuffati sotto i nomi di sviluppatori noti. Come funzionava la backdoor e che cosa dovrebbe imparare ogni team di sviluppo.

Pubblicato: 21 novembre 2026 Tempo di lettura: 6 min Autore: Redazione Airnite

SOFTWARE. SECURITY. CODE.

// LA SITUAZIONE IN CIFRE

2
commit manipolati
a nome di Rasmus Lerdorf e Nikita Popov (php.internals, 28/03/2021)

11
righe di codice malevolo
in ext/zlib/zlib.c, camuffate da “Fix typo” (GitHub)

0
autenticazioni necessarie
esecuzione di codice tramite header HTTP per qualsiasi visitatore (contenuto del commit)

2FA
obbligatoria per i permessi di scrittura
da allora solo tramite l’organizzazione PHP su GitHub (php.internals)

PHP è presente in molti CMS, negozi online e portali clienti. Il 28 marzo 2021 nel repository ufficiale del codice sorgente del progetto PHP sono comparse due modifiche dall’aspetto innocuo. Una di queste aveva il titolo “Fix typo”. In realtà inseriva una backdoor che permetteva di eseguire codice arbitrario su qualsiasi server con una versione di PHP modificata in questo modo.

// Che cosa è successo?

Secondo la comunicazione dello sviluppatore del core di PHP Nikita Popov sulla mailing list php.internals, nel repository php-src erano stati inseriti due commit malevoli, uno a nome dell’inventore di PHP **Rasmus Lerdorf**, l’altro a nome dello stesso Popov. Secondo ESET la prima modifica è stata notata durante una revisione di routine successiva al commit ed è stata rapidamente annullata; alcuni sviluppatori della community avevano chiesto che cosa facesse in realtà quel codice.

Il progetto ha reagito in modo radicale. Il server Git proprietario git.php.net non doveva più essere utilizzato e i mirror esistenti su **GitHub** sono diventati la fonte di riferimento. Da allora i permessi di scrittura spettano solo ai membri dell’organizzazione PHP su GitHub, che richiede l’**autenticazione a due fattori**. Nel testo del commit compariva il nome “Zerodium”, un broker di vulnerabilità; secondo ESET il suo amministratore ha respinto qualsiasi coinvolgimento.

// Come funziona l’attacco?

Il codice malevolo occupa undici righe nel file dedicato alla compressione zlib. Legge un header HTTP dal nome quasi impercettibilmente diverso, **“User-Agentt”** (con la doppia t). Se il valore inizia con la parola “zerodium”, PHP esegue il resto dell’header come codice di programma. A un attaccante sarebbe bastato inviare una richiesta preparata ad hoc a un sito web, senza autenticazione e senza password.

Come gli attaccanti abbiano potuto scrivere nel repository lo ha chiarito un aggiornamento del 6 aprile 2021. A quel punto il team non riteneva più che il server Git fosse stato compromesso. Le modifiche erano state caricate via HTTPS con autenticazione tramite password sul database utenti di master.php.net; era possibile che questo database fosse stato sottratto. Il team è passato ad algoritmi di hashing delle password moderni, ha **reimpostato tutte le password di php.net** e ha ricostruito il server da zero.

// Chi è coinvolto?

L’attacco non era rivolto a una singola azienda, ma alla **supply chain del software**: chi manipola il codice alla fonte raggiunge tutti coloro che lo adottano in seguito. Se la backdoor è stata scoperta presto, il merito è stato di revisori attenti, non di un blocco tecnico. Secondo quanto riferito da Popov e riportato da ESET, la modifica è stata rimossa prima di arrivare in produzione.

Lo schema riguarda qualsiasi azienda che sviluppa software proprio o utilizza librerie di terze parti. Vecchi server gestiti in proprio con una gestione utenti separata, autenticazione tramite password senza secondo fattore e la fiducia nei nomi noti nella cronologia dei commit sono esattamente le debolezze sfruttate in questo caso. Un nome in un commit non dimostra chi lo abbia scritto davvero.

// Che cosa fare adesso?

Le misure adottate da PHP sono un buon modello: gestire centralmente i permessi di scrittura, rendere obbligatoria l'autenticazione a due fattori, dismettere i sistemi obsoleti, conservare le password in modo sicuro. A questo si aggiunge un processo di review che prenda sul serio anche le modifiche più piccole. La backdoor era camuffata da correzione di un refuso: proprio commit di questo tipo vengono spesso approvati senza troppe domande.

Chi utilizza software di terze parti dovrebbe sapere da dove proviene ogni componente e reperire gli aggiornamenti da fonti definite. La checklist qui sotto riassume i passaggi più importanti.

// Come vi aiuta Airnite?

Airnite sviluppa **software** con una supply chain pulita: repository protetti, review obbligatorie, commit firmati e controllo delle dipendenze nella pipeline di build. Nel **penetration test** verifichiamo se accessi degli sviluppatori, server Git e sistemi di build sono attaccabili dall'esterno. Nella **consulenza** mostriamo al vostro team come introdurre processi che rendano visibili per tempo manipolazioni di questo tipo.

> Nota: articolo di approfondimento su un incidente del 2021, considerato un caso di scuola per gli attacchi alla supply chain del software. Ricerca aggiornata al 21 novembre 2026.

Checklist: cosa fare adesso

[+] Assegnare i permessi di scrittura sui repository di codice solo tramite account gestiti centralmente con autenticazione a due fattori; dismettere server e accessi obsoleti.

[+] Approvare ogni modifica al branch principale tramite pull request e principio dei quattro occhi, anche le "correzioni di refusi".

[+] Firmare i commit e verificare le firme nella pipeline di build.

[+] Cercare in modo mirato durante la review gli schemi sospetti: chiamate eval, nomi di header insoliti, codice nei moduli di compressione o di logging.

[+] Proteggere i database delle password degli accessi sviluppatore con algoritmi di hashing moderni e, dopo un incidente, reimpostare tutte le password.

[+] Inventariare le proprie dipendenze (SBOM) e reperirle solo da fonti affidabili e definite.

Come vi aiuta Airnite

- Software e siti web – Siti web e applicazioni web veloci, sicuri, multilingue.
- Penetration test e sicurezza – Attacchi controllati con report chiaro e piano di interventi.
- Consulenza – Lo sguardo esterno sui vostri piani e progetti.

Quanto è sicura la vostra supply chain del software?

Airnite verifica chi può scrivere nei vostri repository, come vengono approvate le modifiche e quali pacchetti di terze parti entrano nel vostro software. Ricevete un elenco chiaro di lacune e misure: dall'obbligo di autenticazione a due fattori ai commit firmati, fino al controllo delle dipendenze.

Richiedete un primo colloquio · Scriveteci un'e-mail · Su WhatsApp

Contatti

Domande su questo articolo? Scriveteci su WhatsApp: +49 174 6834842

Leggi online: https://airnite.de/it/magazine/php-codice-sorgente-backdoor-server-git/

Fonti

Ogni fatto riportato in questo articolo si basa sulle fonti seguenti. Le fonti primarie sono contrassegnate.

1. **php.internals · Nikita Popov: Changes to Git commit workflow** · 28/03/2021 · FONTE PRIMARIA
   Fonte primaria: due commit manipolati, sospetta compromissione di git.php.net, trasferimento su GitHub, obbligo di 2FA
   https://news-web.php.net/php.internals/113838
2. **php.internals · Nikita Popov: Update on git.php.net incident** · 06/04/2021 · FONTE PRIMARIA
   Fonte primaria: server probabilmente non compromesso, possibile fuga del database utenti, reimpostazione delle password
   https://news-web.php.net/php.internals/113981
3. **GitHub · php/php-src commit c730aa2 "[skip-ci] Fix typo"** · 28/03/2021 · FONTE PRIMARIA
   Fonte primaria: contenuto del codice malevolo in ext/zlib/zlib.c
   https://github.com/php/php-src/commit/c730aa26bd52829a49f2ad284b181b7e82a68d7d
4. **ESET WeLiveSecurity · Backdoor added to PHP source code in Git server breach** · 30/03/2021
   Ricerca sulla sicurezza: scoperta durante la review, spiegazione del funzionamento, smentita di Zerodium
   https://www.welivesecurity.com/2021/03/30/backdoor-php-source-code-git-server-breach/

Informazione redazionale di carattere generale, non costituisce consulenza legale né una verifica specifica dei vostri sistemi. Aggiornato alla data di pubblicazione.

Airnite GmbH · Am Waldbach 7 · 77654 Offenburg · airnite.de