Serverul încarcă PHP-ul remediat? Opt verificări.

Lista rezumă metoda din acest caz. Nu înlocuiește recomandările producătorului sau ale furnizorului de hosting și nu este un ghid pentru testarea sistemelor altora.

> Acesta este un material educațional, nu instrucțiuni pentru serverul tău: testează orice modificare mai întâi într-un mediu de test, cu o copie de rezervă actuală, și aplic-o pe propria răspundere.

1. Înainte de orice modificare, fă un dump al bazei de date și verifică-l. Păstrează-l în afara directorului aplicației.
2. Include în inventar fiecare hostname la care răspunde serverul, nu doar adresa publică.
3. Nu judeca remedierea după bannerul X-Powered-By sau după numărul versiunii. Unele distribuții adaugă remedieri fără să schimbe numărul: Ubuntu 24.04 afișează 8.3.6 și înainte, și după patch.
4. Compară data build-ului din php -v și din câmpul Build Date din phpinfo cu data remedierii din changelogul pachetului distribuției.
5. Verifică fișierul modulului PHP pe care îl încarcă serverul web. Linia de comandă și serverul web pot folosi build-uri diferite.
6. Rulează phpinfo doar pe o pagină care nu este accesibilă public și șterge-o după verificare.
7. Ai încredere doar în verificările care pot și să treacă, și să eșueze. Un script de verificare trebuie să poată întoarce atât „neremediat”, cât și „remediat”, iar când datele nu permit o concluzie, „nu se poate stabili”.
8. După fiecare restart din panoul furnizorului și după fiecare actualizare a imaginii, verifică din nou build-ul. Unele reporniri recreează containerul.

php.net recomandă setarea session.upload_progress.enabled = 0 până la următoarea lansare PHP din ciclul obișnuit, chiar și după actualizarea din 21 octombrie. Această setare închide o singură cale de atac; numai actualizarea elimină vulnerabilitatea.

Urmărește tot incidentul în laboratorul gratuit cu narațiune (în browser, fără înscriere):

cyberszkola.pl/php2shell

Surse

1. PHP security advisory GHSA-w8hw-w9fx-j79w — CVE-2026-109618, RCE via OOM corruption https://github.com/php/php-src/security/advisories/GHSA-w8hw-w9fx-j79w
2. PHP security advisory GHSA-6wmf-4rv5-p3rp — CVE-2026-109619, RCE via stale global state https://github.com/php/php-src/security/advisories/GHSA-6wmf-4rv5-p3rp
3. php.net — advance notice of 20 October 2026 https://www.php.net/archive/2026.php#2026-10-20-1
4. php.net — informational advisory: session upload progress (23 October 2026, CVE-2026-115560) https://www.php.net/archive/2026.php#2026-10-23-1
5. CERT-EU Security Advisory 2026-018 — Critical Vulnerabilities in PHP (v1.2) https://cert.europa.eu/publications/security-advisories/2026-018
6. Ubuntu Security — CVE-2026-109618 (package table, USN-8948-1/2) https://ubuntu.com/security/CVE-2026-109618
7. Launchpad — php8.3 8.3.6-0ubuntu0.24.04.13 (changelog: SECURITY UPDATE, CVE-2026-109618/109619) https://launchpad.net/ubuntu/+source/php8.3/8.3.6-0ubuntu0.24.04.13
8. Tenable — php2shell FAQ (updated 28 October 2026) https://www.tenable.com/blog/php2shell-cve-2026-109618-cve-2026-109619-frequently-asked-questions-about-critical-php-remote
9. Datadog Security Labs — PHP2Shell: what the logs and processes show (updated 23 October 2026) https://securitylabs.datadoghq.com/articles/php2shell-critical-vulnerabilities-in-php/
10. Varonis — PHP2Shell: the log pattern to hunt for https://www.varonis.com/blog/php2shell
11. NCSC-NL — third-party software that embeds PHP (inventory aid, referenced by CERT-EU) https://github.com/NCSC-NL/php2shell
12. CISA — Known Exploited Vulnerabilities Catalog https://www.cisa.gov/known-exploited-vulnerabilities-catalog
13. OWASP WSTG-CONF-01: Test Network Infrastructure Configuration https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/02-Configuration_and_Deployment_Management_Testing/01-Test_Network_Infrastructure_Configuration
14. OWASP WSTG-INFO-02: Fingerprint Web Server https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server
15. OWASP WSTG-INFO-08: Fingerprint Web Application Framework https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/01-Information_Gathering/08-Fingerprint_Web_Application_Framework