



عبنبي | تنبيه أمني عاجل · 21 أكتوبر 2026

دليل تحديث PHP خطوة بخطوة

CRITICAL · 9.2 / 10

ثغرتان في PHP نفسه قد تسمحان بتنفيذ أوامر على السيرفر عن بُعد بدون تسجيل دخول. فريق PHP يعتبر كل تثبيت يستخدم طلبات ويب معرض للخطر، والخطر أعلى مع mod_php.

CVE-2026-109618 · CVE-2026-109619

التصرف المطلوب: حدّث PHP لأحدث إصدار أمني مناسب لفرعك، وتأكد إن عمليات PHP التي تستخدم الموقع اشتغلت بالإصدار المصحّح. خذ نسخة احتياطية وخطة تراجع قبل تغيير بيئة الإنتاج.

١ · الإصدارات المصحّحة

أول إصدار مصحّح	فرع PHP الحالي
8.2.35	PHP 8.2
8.3.36	PHP 8.3
8.4.27	PHP 8.4
8.5.12	PHP 8.5

مهم: الإصدارات الأحدث داخل نفس الفرع تشمل الإصلاح. دعم PHP 8.2 الأمني ينتهي في 31 ديسمبر 2026.

8.1 PHP والأقدم: خرجت من الدعم الرسمي. راجع مزود النظام بشأن تحديثات الأمان الممدودة (LTS)، أو خطّط للانتقال إلى فرع مدعوم. ونظرًا لخطورة الثغرتين، فريق PHP نشر إصلاحات مصدريّة مختصرة لـ 7.4 و 8.0 و 8.1 لي بيبي PHP من السورس ومعدّوش مزود LTS.

٢ · مين لازم يراجع السيرفر؟

أي تطبيق ويب يعتمد على PHP، بما في ذلك WordPress وLaravel، سواء عبر PHP-FPM أو mod_php أو php-cgi أو FrankenPHP أو LiteSpeed. تحديث WordPress أو Laravel وحده مش كفاية، لأن الثغرة في PHP نفسه. فريق PHP يعتبر الخطر أعلى مع mod_php، وعنده استغلال مُثبت يشتغل بطلب من غير تسجيل دخول قبل ما كود التطبيق يبدأ.



التشخيص قبل التحديث

اعرف النسخة المستخدمة فعلاً

نسخة الترمينال ممكن تكون مختلفة عن PHP اللي بتخدم طلبات الموقع.

٣. فحص الإصدار

أولاً: إصدار سطر الأوامر (CLI)

```
php -v
```

ده بيعرض إصدار PHP على الـ CLI فقط، ومش دليل إن موقعك بيستخدمه.

ثانياً: إصدار PHP لخدمة الويب

```
php-fpm8.3 -v # example: installed binary  
# Inspect active FPM pools / web server config
```

افحص إعدادات ربط الموقع بالـ FPM pool أو Apache handler، أو إصدار PHP من لوحة الاستضافة. الأمر الأول مثال لفحص binary، ومش إثبات للإصدار المستخدم فعلياً.

متنشرش ملف `phpinfo()` مفتوح للعامة. ممكن يكشف مسارات ومعلومات عن إعدادات السيرفر. استخدم لوحة التحكم أو فحصاً محلياً موثقاً، ولو اضطررت له قيّد الوصول واحذف الملف فوراً.

٤. قبل أي تعديل على Production

1. خُد نسخة احتياطية من ملفات الموقع وقاعدة البيانات أو `Snapshot`، وتأكد إن الاستعادة ممكنة.
2. حدّد نظام التشغيل، ومصدر حزم PHP، والفرع المستخدم، والخدمات المرتبطة به.
3. راجع توافق الإضافات والاعتماديات، خصوصاً عند الانتقال بين فروع PHP.
4. جهّز فترة صيانة وخطة تراجع في حالة ظهور أخطاء.

تنبيه توزيعات Linux: أحياناً Ubuntu / Debian / RHEL بتضيف الإصلاح الأمني لنفس رقم إصدار PHP الأصلي (`backport`). هنا راجع نشرة الحزمة الأمنية من التوزيعة ومراجعة تحديث الحزمة، مش رقم `php -v` وحده.



أمثلة إرشادية، وليست أوامر صالحة لكل سيرفر

طريقة التحديث حسب البيئة

نقذ بعد مراجعة إعدادات السيرفر والمستودعات المستخدمة فعليًا.

٥ • سيناريوهات شائعة

استضافة مشتركة • cPanel / Plesk

افتح MultiPHP Manager أو Select PHP Version أو إعدادات PHP في Plesk. تأكد مع الدعم إن الـ runtime المثبت فيه إصلاح CVE-2026-109618 و CVE-2026-109619 ، لأن اختيار الفرع وحده مش بالضرورة بيثبت آخر تحديث.

Ubuntu / Debian • مثال لفرع 8.3 من حزم APT

```
sudo apt update
sudo apt install --only-upgrade 'php8.3*'
# If the active service is PHP 8.3 FPM:
sudo systemctl restart php8.3-fpm
```

تحقق الأول من أسماء الحزم الموجودة ومصدرها والخدمات الفعالة. لو بتستخدم Apache مع mod_php ممكن يلزم إعادة تشغيل Apache. وراجع نشرة التوزيع لو الإصلاح Backported.

RHEL / AlmaLinux / Rocky • مثال عام

```
sudo dnf upgrade 'php*'
# Restart only the services actually in use
```

راجع AppStream أو Remi أو أي مستودع خارجي، والإصدار المثبت، وخدمات php-fpm / httpd قبل إعادة التشغيل.

Docker • مثال لـ PHP 8.3 FPM

```
FROM php:8.3.36-fpm
# Then rebuild, pull and redeploy:
docker compose build --pull
docker compose up -d
```

الأوامر بتفترض إن Compose فعلاً بيبنى الـ image من Dockerfile. لو بتستخدم image: مباشرة، لازم تعمل pull وإعادة إنشاء الـ container، وتأكد من الإصدار جوه الـ container الجديد.

Windows / IIS / Apache

حمل الإصدار الأمني المناسب من windows.php.net، وتأكد من توافق الامتدادات والإعدادات، وبعدين أعد تشغيل الخدمات المرتبطة وتحقق من الإصدار الفعلي.



التأكد بعد التحديث + المصادر

اختبر... واتأكد من الحماية

تحديث ملفات الحزم وحده مش معناه بالضرورة إن عمّال PHP الفعّالين اشتغلوا بالإصدار الجديد.

٦. قائمة التحقق النهائية

1. أعد تشغيل PHP-FPM / Apache أو أعد نشر الـ containers، حسب بيئتك الفعلية.
2. راجع إصدار الـ CLI وإصدار Runtime الموقع، والنشرات الأمنية لو الإصلاح Backported.
3. جرّب تسجيل الدخول والفورمز ورفع الملفات والدفع والوظائف المهمة.
4. راجع سجلات الأخطاء والـ health checks بعد النشر.
5. لو حصلت مشكلة، استخدم خطة التراجع، وعالج التعارض بسرعة وأعد التحديث. متسببش نسخة معرّضة للخطر.

لو التحديث متعذّر مؤقتًا: الإجراءات دي من إعلان فريق PHP بتقلل التعرّض لمسارات هجوم محددة، ومش بتخلي السيرفر آمن. يراجعها مسؤول السيرفر، والتحديث يفضل الحل الكامل الوحيد.

- **CVE-2026-109619**: `session.upload_progress.enabled=0` في `php.ini` أو `FPM pool` أو إعدادات السيرفر (مش `ini_set()` ولا `.user.ini`)، وبعدها أعد تشغيل كل الـ `workers`. ده بيقلل مسارات الـ `session` المؤكدة.
- **على PHP 8.5 بس**: `fatal_error_backtraces=0` وأعد تشغيل الـ `workers`.
- **CVE-2026-109618**: `post_max_size = 8M` (أو `32M` لو فيه رفع ملفات)، و `upload_max_filesize` أقل منه، ويفضّل كمان حدود لحجم الطلب على الـ `web server` أو الـ `proxy`.
- `memory_limit=-1` (مذكور في الإعلان) بيلغي حد الذاكرة ويمكن يسبب استهلاك زيادة أو توقف الخدمة. متطبّق هوش من غير حد على مستوى الـ `container` أو `cgroup`.
- متستخدمش `RLIMIT_AS` أو `RLIMIT_DATA` كبديل، لأنها ممكن تفتح نفس المسار الضعيف.

٧. المصادر الرسمية

إعلان PHP الرسمي: php.net

github.com/php/php-src/security/advisories/GHSA-w8hw-w9fx-j79w · CVE-2026-109618

github.com/php/php-src/security/advisories/GHSA-6wmf-4rv5-p3rp · CVE-2026-109619

محتاج مساعدة في التحديث؟

ابعت نوع التطبيق والاستضافة وإصدار PHP فقط، من غير أي باسوردات أو ملفات حساسة.
كلم عبنبي على واتساب: [اضغط هنا](https://wa.me/0563081058) · 056 308 1058