PATCHING / 01 OF 02

PHP administrator patch checklist

A controlled update, from inventory to a verified running service. Tick only when completed in your authorised process.

- [ ] **01 Inventory the actual runtime**

  Identify every web PHP handler, pool, container, replica and bundled application. CLI PHP can differ from the website. Use authenticated hosting or application health information; do not publish phpinfo().

- [ ] **02 Record the source and permission**

  Record the OS release, package source, full package revision, service owner and change approver. Agree the scope, maintenance window and recovery route before changing anything.

- [ ] **03 Verify both advisory records**

  Assess CVE-2026-109618 and CVE-2026-109619 using PHP and your vendor. A distribution may backport fixes without adopting the upstream version number. Retain the exact source and evidence.

- [ ] **04 Back up and prove restore**

  Protect application files, databases and configuration. Keep a separate protected backup and test a restore in isolation. A recent backup is not automatically trusted if compromise is suspected.

- [ ] **05 Test in staging**

  Check PHP and extension compatibility, login, forms, uploads and key business flows in a representative isolated environment. Disable real payments, email delivery and external actions during testing.

> **If compromise is suspected**
>
> Pause the routine patch process and involve your host and a qualified incident responder. Agree containment and preserve evidence before cleanup or restoration.

Independent educational guidance. No security guarantee or authority endorsement.

Sources: PHP official advisory (21 Oct 2026) | UK NCSC guidance (21 Oct 2026)

Checked 22 October 2026 | bytova.co.uk | Keep records in a protected location.

BYTOVA

PATCHING / 02 OF 02

Apply the fix. Verify the result.

Installed files and a version label are not enough. Confirm what is running, then record the evidence.

- [ ] **06 Apply an approved vendor update**

  Use the established signed vendor package channel or the authorised hosting control panel. Review intended changes and vendor instructions for your exact platform. Avoid untrusted one-line update commands.

- [ ] **07 Replace every affected worker**

  Follow the vendor restart or reload procedure. Verify that old PHP workers are gone across Apache/mod_php, PHP-FPM pools, Windows application pools, containers and replicas as applicable.

- [ ] **08 Validate version and service**

  Recheck the running web PHP and full package revision. Test login, forms, uploads, background jobs and essential business functions. Exercise checkout only in a test environment. Record outcomes and errors.

- [ ] **09 Review the exposure window**

  Preserve and examine web, PHP, authentication, deployment and application logs. Investigate unexpected accounts, file changes, scheduled tasks, outbound connections or repeated crashes. Quiet logs do not rule out compromise.

- [ ] **10 Document, monitor and close**

  Record approval, package evidence, restart time, validation, remaining risks and the next review. If rollback would restore vulnerable PHP, keep the affected service restricted or offline pending an approved safe alternative.

> **Verified upstream fixes on 22 October 2026**
>
> PHP 8.2.35, 8.3.36, 8.4.27 and 8.5.12 contain the fixes for both advisories. This is a dated record, not a list of all later releases. Check exact vendor backports separately. A version result is not proof of security or compromise.

Independent educational guidance. No security guarantee or authority endorsement.

Sources: PHP official advisory (21 Oct 2026) | UK NCSC guidance (21 Oct 2026)

Checked 22 October 2026 | bytova.co.uk | Keep records in a protected location.