BYTOVA

RECOVERY PREPARATION / 01 OF 02

Website recovery checklist

Return to service from a trusted state, with the host and responder where compromise is suspected.

01 Preserve the incident record first

Keep authorised evidence copies, relevant logs and the decision history before overwriting a server. Maintain custody and integrity records. Coordinate evidence needs with containment and business continuity priorities.

02 Identify what must be restored

List the application, database, uploads, configuration, accounts, integrations and scheduled jobs. Identify essential functions, dependencies and owners. Record the acceptable recovery point and outage impact.

03 Choose a trusted source

Assess backup age, integrity and signs of compromise. Do not assume a snapshot made during the exposure window is clean. Use known-good release artifacts and trusted installation media where appropriate.

04 Recover into isolation

Rebuild or restore in a controlled environment with restricted access. Disable outgoing email, live payments, automated integrations and public traffic until approved checks pass. Retain the original incident environment as agreed.

05 Close the entry point

Apply the vendor fix to PHP and review the affected application and dependencies. Check the relevant full package revision, configuration and every worker, pool, container and replica. Remove unauthorised persistence through an approved recovery process.

Avoid a false finish

A fresh-looking homepage and a fixed PHP label do not establish a trusted system. Recovery needs evidence about the runtime, data, accounts, persistence and business functions, plus a documented decision to reconnect.

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

RECOVERY VALIDATION / 02 OF 02

Validate the service. Reconnect deliberately.

Record the evidence for each check and the owner of any outstanding risk. Do not silently return to a vulnerable runtime.

- [ ] **06 Review accounts and secrets**

  From a trusted device, rotate affected credentials and application secrets after containment, and revoke unnecessary access. Confirm administrators and integration permissions with their owners. Plan dependent changes to avoid accidental outages.

- [ ] **07 Validate data and functionality**

  Check critical data integrity, permissions, login, forms, uploads, scheduled tasks and administration. Test the intended customer journeys. Use sandbox payment and message systems for tests, then reconcile any manual requests.

- [ ] **08 Confirm visibility and monitoring**

  Ensure relevant logs are retained and accessible to the approved people. Establish a baseline for expected processes, accounts and network activity. Assign a monitoring owner and an escalation route for anomalies.

- [ ] **09 Authorise a staged return**

  Record who approved reconnection, the evidence reviewed and remaining limitations. Restore public access gradually where feasible. Keep the incident lead and host available for the agreed monitoring period.

- [ ] **10 Retain a safe recovery route**

  If the new deployment fails, use an agreed alternative that does not expose vulnerable PHP again. Keep a restricted or offline state if needed. Preserve the final record, follow-up actions and lessons for the next recovery exercise.

> **Recovery sign-off**
>
> Record the restored release and data point, verified package, worker replacement time, test outcomes, approver, unresolved risks and next review. A qualified responder should assess suspected compromise; these templates do not certify clearance.

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.