



Guide

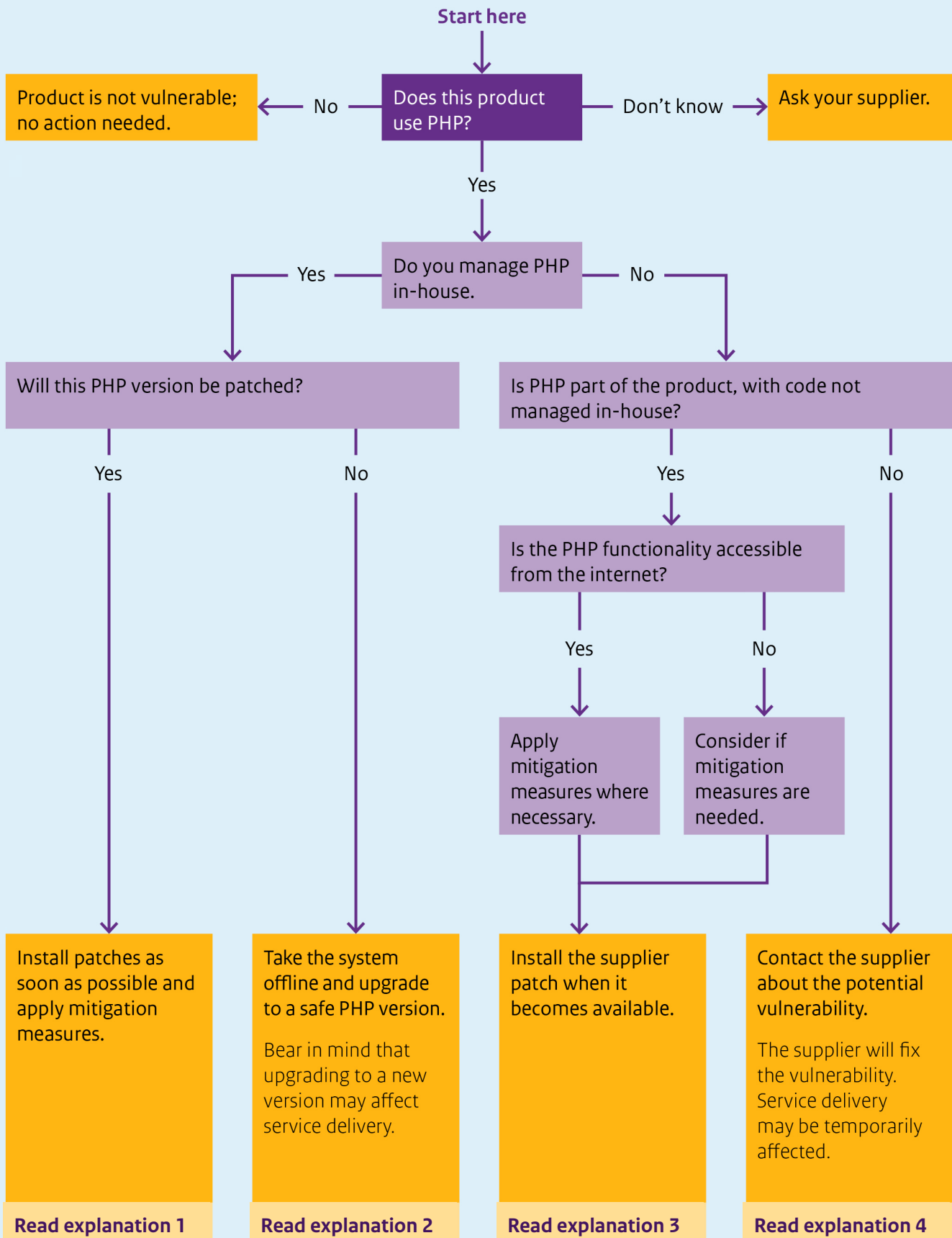
PHP vulnerabilities: what can you do?

Decision tree + explanation

22 October 2026



Decision tree



Explanations for the decision tree

Explanation 1 - PHP is managed in-house, version 8

Your organisation currently uses PHP version 8.2 or later and manages the product in-house. Please note the following:

Patches for PHP version 8 were released on Wednesday 21 October 2026. Install these patches as soon as possible.

Context

- If you are running a recent release of PHP 8.2 or later, patch as soon as possible, then investigate potential exploitation where necessary.
- Bear in mind that your organisation's services may be delayed due to the mitigation measures taken in relation to the patch and the investigation into any exploitation following the patch..
- Several mitigation options are available. The most disruptive is temporarily taking systems offline until patches have been installed..

Explanation 2 - PHP is managed in-house, version is outdated

Your organisation currently uses an outdated PHP version earlier than 8.2 and manages the product in-house. Please note the following:

- If you manage a PHP product in-house but it will not be patched because its version is not covered by the patch, take the product offline.
- This is essential: as long as the product is accessible from the internet, its vulnerabilities can be actively exploited.
- Upgrade to the latest patched PHP version as soon as possible, then start investigating potential exploitation.
- Remaining or going online without upgrading is very risky, as attempts are currently being made to exploit vulnerable PHP versions.
- Therefore, upgrade before going online. This may have side effects: new PHP versions can change how interconnected systems communicate and respond to each other.

Context

- Organisations using old PHP versions that will not be patched will experience the most downtime. No patch will be released for PHP versions earlier than 8.2. It is therefore important to take these systems offline as soon as possible. To prevent exploitation of the vulnerability in these old PHP versions, upgrade to a safe, non-vulnerable version.
- Upgrading to a new PHP version may cause conflicts with other software used by your organisation. Bear this in mind, as it may have a prolonged impact on the availability of your system or application.

Explanation 3 - Product using PHP is managed in-house, PHP code itself is not

Your organisation currently uses a product that uses PHP. You do not manage the PHP code itself in-house, but you do manage the product that uses it. Please note the following:

- Check whether the PHP functionality is externally accessible. For example, a web interface on an edge device may use PHP to enable users to log in.
- If so, take appropriate mitigation measures based on a risk assessment. If not, consider whether mitigation measures are needed.
- Next, wait for the supplier to release a patch. Install it as soon as possible.

Context

PHP can be used in many different types of products. Organisations need to know which type of product uses PHP and what its functions are. This allows them to assess the risks and determine the most appropriate measures if the PHP functionality is externally accessible. Even if the functionality is not externally accessible, assess the risks and consider possible mitigation measures.

Explanation 4 - Neither the product using PHP nor PHP itself is managed in-house

Your organisation currently uses a product that uses PHP. Neither the PHP code itself nor the product running PHP is managed in-house; both are supplied by an external party (for example, a cloud product). Please note the following:

- Contact the supplier to establish the nature of the vulnerability.
- The supplier will release a new version of the product in which the vulnerabilities have been fixed.

Context

Allow for potential disruption to service delivery until the supplier has fixed the vulnerabilities. Ask your supplier about this where necessary and prepare accordingly.

Where do we go from here?

The NCSC expects the current PHP situation to require sustained attention. PHP is so widely used across many different product categories that it will take time to update most vulnerable systems and products to a safe version.

Upgrading older PHP versions (5.x and 7.x) that have reached end of life (EOL) will also take time and affect connected applications and functionality. This means there will be a prolonged period during which malicious actors can identify vulnerable systems and organisations and carry out cyberattacks.

Suppliers of systems that use PHP will not have a patch available immediately: they must first test the patches released by PHP internally on their products. This may delay patches for products that use PHP. Organisations should therefore consider whether mitigation measures are needed.

Experience with a similar vulnerability in Log4j (Log4Shell) shows that attackers continue exploiting such vulnerabilities for a long time. This is likely to be the case for these PHP vulnerabilities too.

Published by
National Cyber Security Centre (NCSC)
PO Box 117, 2501 CC The Hague
Turfmarkt 147, 2511 DP The Hague
070 751 5555

More information
www.ncsc.nl
info@ncsc.nl
[@ncsc_nl](https://twitter.com/ncsc_nl)

October 2026