

PHP-doc

Review of Code Snippets and User Comments in the PHP
Documentation

Reference 24-12-1913-LIV

Version 2.0

Date 2025-09-10



Quarkslab

Quarkslab SAS
10 boulevard Haussmann
75009 Paris
France



Legal notice

This report reflects the work and results obtained within the duration of the audit on the specified scope (see Section 2.1) and as agreed between PHP OSTIF and Quarkslab. Tests are not guaranteed to be exhaustive and the report does not ensure the code is bug or vulnerability free.

1. Project information

1.1. Document history

Version	Date	Details	Authors
1.0	2024-12-20	Initial audit	Mihail Kirov, Jacques Ricard
2.0	2025-09-10	Control audit (retest)	Mihail Kirov

1.2. Contacts

1.2.1. Quarkslab

Name	Role	Email
Frédéric Raynal	CEO	fraynal@quarkslab.com
Pauline Sauder	Project Manager	psauder@quarkslab.com
Ramtine Tofighi Shirazi	Project Manager	mrtofighishirazi@quarkslab.com
Mihail Kirov	R&D Engineer	mkirov@quarkslab.com
Jacques Ricard	R&D Engineer	jricard@quarkslab.com

1.2.2. PHP

Name	Role	Email
Roman Pronskiy	Operations Manager	roman.pronskiy@jetbrains.com

1.2.3. OSTIF

Name	Role	Email
Helen Woeste	Project Manager	helen@ostif.org
Amir Montazery	Managing Director	amir@ostif.org
Derek Zimmer	Executive Director	derek@ostif.org

Contents

- 1. Project information** **1**
 - 1.1. Document history 1
 - 1.2. Contacts 1

- 2. Executive Summary** **3**
 - 2.1. Context 3
 - 2.2. Objectives 4
 - 2.3. Methodology 5
 - 2.4. Findings Summary 6
 - 2.5. Retest 09/2025 – Findings status 6
 - 2.6. Recommendation and Action Plan 16
 - 2.7. Conclusions 19

- 3. Reading Guide** **20**
 - 3.1. Executive summary 20
 - 3.2. Metric definition 20

- 4. Code review** **22**
 - 4.1. Documentation with no issues discovered 22
 - 4.2. Documentation with issues discovered 181
 - 4.3. Documentation with no code 247
 - 4.4. User comments 251

- 5. Conclusions** **277**

2. Executive Summary

2.1. Context

The Open Source Technology Fund., Inc (OSTIF), engaged with Quarkslab to perform a security audit of the code snippets in [the English version of PHP documentation](#), focused on some specific pages.

The PHP documentation (i.e., manual) consists primarily of a function reference, but also contains a language reference, explanations of some of PHP's major features, and other supplemental information.

The PHP documentation also includes information covering the two most recent major releases of PHP (versions 7 and 8). Moreover, users can also post comments on some pages.

The OSTIF and Quarkslab have collaborated on several security assessments through the years, in the context of securing widely used and crucial open-source projects, such as

- [PHP-SRC security audit](#)
- [Notary project security audit](#)
- [Audit of Operator Fabric, 2024](#)
- [Cloud Native Buildpacks security audit, 2024](#)
- [Kuksa security audit, 2024](#)
- [Falco security audit, 2023](#)

This report presents the results of the security assessment, performed in 10 days.

Info

This report reflects the work and results obtained within the duration of the audit and on the specified scope, as agreed between the OSTIF, PHP Foundation, and Quarkslab.

2.2. Objectives

The defined objectives for this collaboration were to perform a security review of the PHP documentation. As the documentation is quite extensive, the scope was reduced based on priorities and the review is performed in a best-effort manner, within the allocated time frame of the review.

Quarkslab based the review on the following priorities, as agreed with the OSTIF and PHP Foundation:

- Documentation code snippet review as first priority
 - Focus on specific topics, such as filesystem, then cryptography, and more if time permits/relevant
- Then focus on user-comment code snippets
 - Starting with <https://php.net/manual/en/function.curl-setopt.php>

Info

This report specifies all reviewed documentation pages for clarity purposes.

The main purpose for Quarkslab is to review all snippets and some user comments in order to have an overall assessment of the code snippets quality.

Warning

Tests are not guaranteed to be exhaustive and the report does not ensure that the code is bug or vulnerability free.

2.3. Methodology

To assess the security of the PHP documentation and user-comment code snippets, Quarkslab's team methodology was straightforward in doing the review taking into account the context of the snippets.

The evaluation was mostly based on manual code reviews.

Overall, a single technical step was performed as listed below:

1. Step 1: Code review

- Review the source code snippets in PHP documentation.
- Review associated user posts of source code in PHP documentation.
- Given the large amount of documentation and user posts, focus on priorities defined by PHP Foundation, as previously mentioned.
- Usage of both manual and automated tools for the review when applicable.

2.4. Findings Summary

During the time frame of the security audit, Quarkslab has discovered several security issues and vulnerabilities.

Given that the scope is focused on snippets of code and user comments, no severity ranking is provided.

Overall, we have 81 observed issues.

Warning

Since the scope is focused on code examples, some issues (e.g., hardcoded secret keys) are not considered as an issue since their purpose is to illustrate the example. However, Quarkslab strongly advises PHP foundation to add disclaimers and notes on such practices.

Also discovered security issues and bad practices are marked as “ISSUE”, as observed in the table below.

2.5. Retest 09/2025 – Findings status

Quarkslab’s engineers were tasked with performing a control audit nine months after the delivery of version 1.0 of the report, in order to verify whether the discovered issues had been corrected. The GitHub commit which introduced most of the patches is <https://github.com/php/doc-en/commit/bce2cb849721c70737eb32ce958131e22677770c> and thus, the auditors focused on verifying that the issues had been corrected in this commit.

Warning

The version of php-doc at commit ``bce2cb849721c70737eb32ce958131e22677770c`` could not be built using the provided in the repository Dockerfile. Hence, the auditors, after several attempts, decided to work on a successor commit ``e3257766221ace07316deebc972f17ebc05debc2``

This subsection is included in version 2.0 of the report.

The results of the retest demonstrate that the issues were correctly addressed. However, regarding the explicit error handling within the code snippets in the doc, no visible improvements were made. An [appendix](#) file was created issuing a warning about the absence of error handling in the code snippets, but this appendix is not rendered in the published documentation. Consequently, we treat all issues related to missing error handling as unresolved.

ID	Name	Perimeter
INFO-1	Type not properly declared	https://www.php.net/manual/en/

ID	Name	Perimeter
		filter.constants.php
INFO-2	Improper value handling	https://www.php.net/manual/en/filter.constants.php
INFO-3	Filesystem operations not checked	https://www.php.net/manual/en/function.dio-open.php
INFO-4	Filesystem operations not checked	https://www.php.net/manual/en/function.dio-close.php
INFO-5	Filesystem operations not checked	https://www.php.net/manual/en/function.dio-fcntl.php
INFO-6	Filesystem operations not checked	https://www.php.net/manual/en/function.dio-seek.php
INFO-7	Filesystem operations not checked	https://www.php.net/manual/en/function.dio-tcsetattr.php
INFO-8	Weak validation/checking prone to bypass	https://www.php.net/manual/en/function.dio-tcsetattr.php
INFO-9	Filesystem operations not checked	https://www.php.net/manual/en/function.dir.php
INFO-10	Filesystem operations not checked	https://www.php.net/manual/en/function.unlink.php
INFO-11	Filesystem operations not checked	https://www.php.net/manual/en/

ID	Name	Perimeter
		function.fclos e.php
INFO-12	Filesystem operations not checked	https://www.php.net/manual/en/function.fdatasync.php
INFO-13	Filesystem operations not checked	https://www.php.net/manual/en/function fflush.php
INFO-14	Filesystem operations not checked	https://www.php.net/manual/en/function flock.php
INFO-15	Filesystem operations not checked	https://www.php.net/manual/en/function flock.php
INFO-16	Filesystem operations not checked	https://www.php.net/manual/en/function.fpassthru.php
INFO-17	Filesystem operations not checked	https://www.php.net/manual/en/function.popen.php
INFO-18	Filesystem operations not checked	https://www.php.net/manual/en/function.popen.php
INFO-19	Filesystem operations not checked	https://www.php.net/manual/en/function.rewind.php
INFO-20	Filesystem operations not checked	https://www.php.net/manual/en/function.fread.php
INFO-21	Filesystem operations not checked	https://www.php.net/manual/en/function.fread.php

ID	Name	Perimeter
INFO-22	Filesystem operations not checked	https://www.php.net/manual/en/function.fread.php
INFO-23	Filesystem operations not checked	https://www.php.net/manual/en/function.fwrite.php
INFO-24	Filesystem operations not checked	https://www.php.net/manual/en/function.fscanf.php
INFO-25	Filesystem operations not checked	https://www.php.net/manual/en/function.fseek.php
INFO-26	Filesystem operations not checked	https://www.php.net/manual/en/function.ftell.php
INFO-27	Filesystem operations not checked	https://www.php.net/manual/en/function.fputcsv.php
INFO-28	Filesystem operations not checked	https://www.php.net/manual/en/function.fstat.php
INFO-29	Filesystem operations not checked	https://www.php.net/manual/en/function.fsync.php
INFO-30	Broken link	https://www.php.net/manual/en/intro.enchant.php
INFO-31	Lack of exception handling	https://www.php.net/manual/en/function.enchant-broker-dict-exists.php
INFO-32	Lack of exception handling	https://www.php.net/manual/en/function.enchant-broker-list-dicts.php

ID	Name	Perimeter
INFO-33	Lack of exception handling	https://www.php.net/manual/en/function.enchant-broker-request-dict.php
INFO-34	Typo error	https://www.php.net/manual/en/function.enchant-broker-request-dict.php
INFO-35	Usage of deprecated function	https://www.php.net/manual/en/function.enchant-dict-add.php
INFO-36	Lack of exception handling	https://www.php.net/manual/en/function.enchant-dict-add.php
INFO-37	Lack of exception handling	https://www.php.net/manual/en/function.enchant-dict-add.php
INFO-38	Typo error	https://www.php.net/manual/en/function.enchant-dict-describe.php
INFO-39	Lack of exception handling	https://www.php.net/manual/en/function.enchant-dict-describe.php
INFO-40	Lack of exception handling	https://www.php.net/manual/en/function.enchant-dict-quick-check.php
INFO-41	Usage of deprecated function	https://www.php.net/manual/en/function.enchant-dict-suggest.php

ID	Name	Perimeter
INFO-42	Lack of exception handling	https://www.php.net/manual/en/function.enchant-dict-suggest.php
INFO-43	Lack of exception handling	https://www.php.net/manual/en/function.iconv-get-encoding.php
INFO-44	Lack of exception handling	https://www.php.net/manual/en/intl.examples-basic.php
INFO-45	Lack of exception handling	https://www.php.net/manual/en/collator.asort.php
INFO-46	Lack of exception handling	https://www.php.net/manual/en/collator.compare.php
INFO-47	Lack of exception handling	https://www.php.net/manual/en/collator.getattribute.php
INFO-48	Lack of exception handling	https://www.php.net/manual/en/collator.geterrorcode.php
INFO-49	Lack of exception handling	https://www.php.net/manual/en/collator.geterrormessage.php
INFO-50	Lack of exception handling	https://www.php.net/manual/en/collator.getlocale.php
INFO-51	Lack of exception handling	https://www.php.net/manual/en/collator.getsortkey.php

ID	Name	Perimeter
INFO-52	Lack of exception handling	https://www.php.net/manual/en/collator.setattribute.php
INFO-53	Lack of exception handling	https://www.php.net/manual/en/collator.sort.php
INFO-54	Lack of exception handling	https://www.php.net/manual/en/collator.sortwithsortkeys.php
INFO-55	Usage of undefined and/or obsolete function	https://www.php.net/manual/en/numberformatter.format.php
INFO-56	Wrong parameters	https://www.php.net/manual/en/numberformatter.format.php
INFO-57	Wrong parameters	https://www.php.net/manual/en/numberformatter.format.php
INFO-58	Usage of undefined and/or obsolete function	https://www.php.net/manual/en/numberformatter.format.php
INFO-59	Lack of exception handling	https://www.php.net/manual/en/numberformatter.getattribute.php
INFO-60	Usage of undefined and/or obsolete function	https://www.php.net/manual/en/numberformatter.geterrorcode.php
INFO-61	Usage of undefined and/or obsolete function	https://www.php.net/manual/en/numberformatter.geterrorcode.php

ID	Name	Perimeter
INFO-62	Lack of exception handling	https://www.php.net/manual/en/intlcalendar.createinstance.php
INFO-63	Lack of exception handling	https://www.php.net/manual/en/intlcalendar.equals.php
INFO-64	Lack of exception handling	https://www.php.net/manual/en/intlcalendar.fielddifference.php
INFO-65	Usage of undefined and/or obsolete function	https://www.php.net/manual/en/intlcalendar.fielddifference.php
INFO-66	Lack of exception handling	https://www.php.net/manual/en/intlcalendar.fromdatetime.php
INFO-67	Usage of undefined and/or obsolete function	https://www.php.net/manual/en/intlcalendar.fromdatetime.php
INFO-68	Loose comparison	https://www.php.net/manual/en/function.curl_setopt.php#75525
INFO-69	Potential path traversal	https://www.php.net/manual/en/function.curl_setopt.php#75525
INFO-70	Workaround of a security feature could lead to a SSRF and LFI	https://www.php.net/manual/en/function.curl_setopt.php#113682
INFO-71	Hardcoded credentials	https://www.php.net/manual/

ID	Name	Perimeter
		en/function.curl-setopt.php#80271
INFO-72	Usage of Basic authentication	https://www.php.net/manual/en/function.curl-setopt.php#80271
INFO-73	Lack of XML validation	https://www.php.net/manual/en/function.curl-setopt.php#80271
INFO-74	Unbounded recursion	https://www.php.net/manual/en/function.curl-setopt.php#59993
INFO-75	Workaround of a security feature could lead to a SSRF and LFI	https://www.php.net/manual/en/function.curl-setopt.php#95027
INFO-76	Lack of secure communications	https://www.php.net/manual/en/function.curl-setopt.php#121973
INFO-77	Workaround of a security feature could lead to a SSRF and LFI	https://www.php.net/manual/en/function.curl-setopt.php#95027
INFO-78	Unbounded recursion	https://www.php.net/manual/en/function.curl-setopt.php#95027
INFO-79	Workaround of a security feature could lead to a SSRF and LFI	https://www.php.net/manual/en/function.curl-setopt.php#102121
INFO-80	Workaround of a security feature could lead to a SSRF and LFI	https://www.php.net/manual/en/function.curl-setopt.php#71313

ID	Name	Perimeter
INFO-81	Hardcoded credentials	https://www.php.net/manual/en/function.curl-setopt.php#84374

2.6. Recommendation and Action Plan

This table provides all recommendations to fix issues discovered during the review. Quarkslab strongly suggests taking into account all the provided recommendations.

ID	Recommendation
INFO-1	Declare input type as a <code>string</code> .
INFO-2	Handle null value in the callback.
INFO-3	Check result of <code>dio_open()</code> before using <code>dio_close()</code> .
INFO-4	Check result of <code>dio_open()</code> before using <code>dio_close()</code> .
INFO-5	Check result of <code>dio_open()</code> .
INFO-6	Check result of <code>dio_open()</code> .
INFO-7	Check result of <code>dio_open()</code> .
INFO-8	Strong verification is recommended such as <code>if (\$data !== false && \$data !== '')</code>
INFO-9	Check result of <code>dir()</code> .
INFO-10	Check result of <code>fopen()</code> .
INFO-11	Check result of <code>fopen()</code> .
INFO-12	Check result of <code>fopen()</code> .
INFO-13	Check result of <code>fopen</code> .
INFO-14	Check result of <code>fopen()</code> .
INFO-15	Check result of <code>fopen()</code> .
INFO-16	Check result of <code>fopen()</code> .
INFO-17	Check result of <code>fopen()</code> .
INFO-18	Check result of <code>popen()</code> .
INFO-19	Check result of <code>fopen()</code> .
INFO-20	Check result of <code>fopen()</code> .
INFO-21	Check result of <code>fopen()</code> .
INFO-22	Check result of <code>fopen()</code> .
INFO-23	Check result of <code>fopen()</code> .
INFO-24	Check result of <code>fopen()</code> .
INFO-25	Check result of <code>fopen()</code> .
INFO-26	Check result of <code>fopen()</code> .
INFO-27	Check result of <code>fopen()</code> .
INFO-28	Check result of <code>fopen()</code> .
INFO-29	Check result of <code>fopen()</code> .
INFO-30	Update with proper link to the library.
INFO-31	Add exception handling for the case where <code>\$r</code> is <code>false</code> .

ID	Recommendation
INFO-32	Add exception handling for the case where <code>\$r</code> is <code>false</code> .
INFO-33	Add exception handling when <code>enchant_broker_init()</code> returns <code>false</code> .
INFO-34	Replace <code>\$r</code> with <code>\$broker</code> .
INFO-35	Remove deprecated functions.
INFO-36	Add exception handling for the case where <code>\$dict</code> is <code>false</code> .
INFO-37	Add exception handling for the case where <code>\$broker</code> is <code>false</code> .
INFO-38	<code>\$dict = enchant_broker_request_dict(\$r, \$tag)</code> should be replaced with <code>\$dict = enchant_broker_request_dict(\$broker, \$tag)</code> .
INFO-39	Add exception handling for the case where <code>\$dict</code> is <code>false</code> .
INFO-40	Add exception handling for the case where <code>\$d</code> is <code>false</code> .
INFO-41	Remove the deprecated functions.
INFO-42	Add exception handling for the case where <code>\$d</code> is <code>false</code> .
INFO-43	Implement proper checks for the return values of <code>iconv</code> to handle potential errors.
INFO-44	Include checks to ensure <code>collator_create()</code> does not return <code>NULL</code> before proceeding.
INFO-45	Add proper checks in the code snippet.
INFO-46	Add proper checks in the code snippet.
INFO-47	Add proper checks in the code snippet.
INFO-48	Add proper checks in the code snippet.
INFO-49	Add proper checks in the code snippet.
INFO-50	Add proper checks in the code snippet.
INFO-51	Add proper checks in the code snippet.
INFO-52	Add proper checks in the code snippet.
INFO-53	Add proper checks in the code snippet.
INFO-54	Add proper checks in the code snippet.
INFO-55	Remove obsolete functions.
INFO-56	Fix code snippet to illustrate proper usage of the <code>numfmt_format()</code> function.
INFO-57	fix code snippet to illustrate proper usage of the <code>intl_is_failure()</code> function.
INFO-58	Change the code snippet with up-to-date functions.
INFO-59	Add proper checks in the code snippet.
INFO-60	Change the code snippet with up-to-date functions.
INFO-61	Change the code snippet with up-to-date functions.
INFO-62	Add proper checks in the code snippet.
INFO-63	Add proper checks in the code snippet.
INFO-64	Add proper checks in the code snippet.

ID	Recommendation
INFO-65	Change the code snippet with up-to-date functions and with proper argument usage.
INFO-66	Add proper checks in the code snippet.
INFO-67	Change the code snippet with up-to-date functions and with proper argument usage.
INFO-68	Use <code>!==</code> for comparing the <code>\$cookies</code> variable with an empty string.
INFO-69	Clearly state that the <code>\$cookies</code> variable should not be user-controlled to prevent path traversal attacks
INFO-70	Associate a warning to the comment stating that bypassing the limitations of <code>open_basedir</code> could have severe security implications such as SSRF and LFI
INFO-71	Issue a warning to not use hardcoded credentials. Use environment variables or secure vaults instead.
INFO-72	Emit a warning stating that the following example is insecure and should not be used or at least the usage of Basic Authentication should be done over HTTPS
INFO-73	Emit a warning stating the comment is prone to XXE attacks. Urge the users to validate and sanitize the XML data to prevent XXE attacks.
INFO-74	Emit a warning stating the comment does not define a recursion limit and that it is a good practice to define a depth limit when working with deeply nested structures to prevent unbounded recursion and potential stack overflow error
INFO-75	Associate a warning to the comment stating that bypassing the limitations of <code>open_basedir</code> could have severe security implications such as SSRF and LFI.
INFO-76	Emit a warning that it is recommended to use FTPS or other secure alternatives instead of plain FTP, as it transfers data in clear text and can be intercepted.
INFO-77	Associate a warning to the comment stating that bypassing the limitations of <code>open_basedir</code> could have severe security implications such as SSRF and LFI.
INFO-78	Emit a warning saying that this code snippet could result in a stack overflow due to unbound recursion
INFO-79	Associate a warning to the comment stating that bypassing the limitations of <code>open_basedir</code> could have severe security implications such as SSRF and LFI.
INFO-80	Associate a warning to the comment stating that bypassing the limitations of <code>open_basedir</code> could have severe security implications such as SSRF and LFI.
INFO-81	Issue a warning to not use hardcoded credentials. Use environment variables or secure vaults instead.

2.7. Conclusions

Quarkslab performed the security review of several PHP documentation pages.

Also, Quarkslab focused on user comments on the most commented pages.

The review illustrates that overall:

- Code quality best-practices are not enforced (e.g., handling errors, input validations) which could be an issue if the snippet is copied into an implementation;
- Depreciated functions documentation are not always updated;
- User comments are not to be trusted by default, especially when they are quite old and the project as quite evolved since then.

Warning

Quarkslab strongly recommends applying all recommendations provided throughout all the code snippets in PHP documentation.

Retest 09/2025

Info

The PHP maintainers have incorporated Quarkslab's audit recommendations, resulting in improved overall quality of the PHP documentation. We nevertheless recommend rendering the [appendix](#) introduced in pull request 4797, thereby promoting best practices in error handling and input validation.

3. Reading Guide

This reading guide describes the different sections present in this report and gives some insights about the information contained in each of them and how to interpret it.

3.1. Executive summary

The executive summary Section 2 presents the results of the assessment in a non-technical way, summarizing all the findings and explaining the associated risks. For each vulnerability, a severity level is provided as well as a name or short description, and one or more mitigation, as shown below.

ID	Name	Category
CRITICAL	Vulnerability Name #1	Injection
HIGH	Vulnerability Name #2	Remote code injection
MEDIUM	Vulnerability Name #3	Denial of Service
LOW	Vulnerability Name #4	Information leak

Each vulnerability is identified throughout this document by a unique identifier `<LEVEL>-<ID>`, where ID is a number and LEVEL the severity (INFO, LOW, MEDIUM, HIGH or CRITICAL). Every vulnerability identifier present in the vulnerabilities summary table is a clickable link that leads to the corresponding technical analysis that details how it was found (and exploited if it was the case). Severity levels are explained in Section 3.2.

The executive summary also provides an action plan with a focus on the identified *quick wins*, some specific mitigation that would drastically improve the security of the assessed system.

3.2. Metric definition

This report uses specific metrics to rate the severity, impact and likelihood of each identified vulnerability.

3.2.1. Impact

The impact is assessed regarding the information an attacker can access by exploiting a vulnerability but also the operational impact such an attack can have. The following table summarizes the different levels of impact we are using in this report and their meanings in terms of information access and availability.

CRITICAL	Allows a total compromise of the assessed system, allowing an attacker to read or modify the data stored in the system as well as altering its behavior.
-----------------	--

HIGH	Allows an attacker to impact significantly one or more components, giving access to sensitive data or offering the attacker a possibility to pivot and attack other connected assets.
MEDIUM	Allows an attacker to access some information, or to alter the behavior of the assessed system with restricted permissions.
LOW	Allows an attacker to access non-sensitive information, or to alter the behavior of the assessed system and impact a limited number of users.

3.2.2. Likelihood

The vulnerability likelihood is evaluated by taking the following criteria in consideration:

- **Access conditions:** the vulnerability may require the attacker to have physical access to the targeted asset or to be present in the same network for instance, or can be directly exploited from the Internet.
- **Required skills:** an attacker may need specific skills to exploit the vulnerability.
- **Known available exploit:** when a vulnerability has been published and an exploit is available, the probability a non-skilled attacker would find it and use it is pretty high.

The following table summarizes the different level of vulnerability likelihood:

CRITICAL	The vulnerability is easy to exploit even from an unskilled attacker and has no specific access conditions.
HIGH	The vulnerability is easy to exploit but requires some specific conditions to be met (specific skills or access).
MEDIUM	The vulnerability is not trivial to discover and exploit, requires very specific knowledge or specific access (internal network, physical access to an asset).
LOW	The vulnerability is very difficult to discover and exploit, requires highly specific knowledge or authorized access

3.2.3. Severity

The severity of a vulnerability is defined by its impact and its likelihood, following the following table:

		Impact			
		●●●●	●●●○	●○○○	●○○○
Likelihood	●●●●	CRITICAL	CRITICAL	HIGH	MEDIUM
	●●●○	CRITICAL	HIGH	HIGH	MEDIUM
	●●○○	HIGH	HIGH	MEDIUM	LOW
	●○○○	MEDIUM	MEDIUM	LOW	LOW

4. Code review

This section presents all the findings of the English PHP documentation review.

For clarity, a section is added with all the checked links with no code to review.

4.1. Documentation with no issues discovered

This section provides all the documentation pages reviewed by Quarkslab for which no security issues were identified.

4.1.1. filter.examples.validation.php

URL: <https://www.php.net/manual/en/filter.examples.validation.php>

4.1.1.1. Example 1

Code snippet:

```
1 <?php
2 $email_a = 'joe@example.com';
3 $email_b = 'bogus';
4
5 if (filter_var($email_a, FILTER_VALIDATE_EMAIL)) {
6     echo "Email address '$email_a' is considered valid.\n";
7 }
8 if (filter_var($email_b, FILTER_VALIDATE_EMAIL)) {
9     echo "Email address '$email_b' is considered valid.\n";
10 } else {
11     echo "Email address '$email_b' is considered invalid.\n";
12 }
13 ?>
```

Success

No security issues were identified in the code snippet

4.1.1.2. Example 2

Code snippet:

```
1 <?php
2 $ip_a = '127.0.0.1';
3 $ip_b = '42.42';
4
```

```

5 if (filter_var($ip_a, FILTER_VALIDATE_IP)) {
6     echo "IP address '$ip_a' is considered valid.";
7 }
8 if (filter_var($ip_b, FILTER_VALIDATE_IP)) {
9     echo "IP address '$ip_b' is considered valid.";
10 }
11 ?>

```

Success

No security issues were identified in the code snippet

4.1.1.3. Example 3

Code snippet:

```

1 <?php
2 $int_a = '1';
3 $int_b = '-1';
4 $int_c = '4';
5 $options = array(
6     'options' => array(
7         'min_range' => 0,
8         'max_range' => 3,
9     )
10 );
11 if (filter_var($int_a, FILTER_VALIDATE_INT, $options) !== FALSE) {
12     echo "Integer A '$int_a' is considered valid (between 0 and 3).\n";
13 }
14 if (filter_var($int_b, FILTER_VALIDATE_INT, $options) !== FALSE) {
15     echo "Integer B '$int_b' is considered valid (between 0 and 3).\n";
16 }
17 if (filter_var($int_c, FILTER_VALIDATE_INT, $options) !== FALSE) {
18     echo "Integer C '$int_c' is considered valid (between 0 and 3).\n";
19 }
20
21 $options['options']['default'] = 1;
22 if (($int_c = filter_var($int_c, FILTER_VALIDATE_INT, $options)) !== FALSE) {
23     echo "Integer C '$int_c' is considered valid (between 0 and 3).";
24 }
25 ?>

```

Success

No security issues were identified in the code snippet

4.1.2. function.filter-var.php

URL: <https://www.php.net/manual/en/function.filter-var.php>

4.1.2.1. Example 1

Code snippet:

```
1 <?php
2 var_dump(filter_var('bob@example.com', FILTER_VALIDATE_EMAIL));
3 var_dump(filter_var('https://example.com', FILTER_VALIDATE_URL,
4 FILTER_FLAG_PATH_REQUIRED));
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.2.2. Example 2

Code snippet:

```
1 <?php
2 $emails = [
3     "bob@example.com",
4     "test@example.local",
5     "invalidemail"
6 ];
7
8 var_dump(filter_var($emails, FILTER_VALIDATE_EMAIL, FILTER_REQUIRE_ARRAY));
9 ?>
```

Success

No security issues were identified in the code snippet

4.1.2.3. Example 3

Code snippet:

```
1 <?php
2
3 $options = [
4     'options' => [
5         'min_range' => 10,
6     ],
```

```
7     'flags' => FILTER_FLAG_ALLOW_OCTAL,  
8 ];  
9  
10 var_dump(filter_var('0755', FILTER_VALIDATE_INT, $options));  
11 var_dump(filter_var('011', FILTER_VALIDATE_INT, $options));  
12  
13 ?>
```

Success

No security issues were identified in the code snippet

4.1.2.4. Example 4

Code snippet:

```
1 <?php  
2  
3 $str = 'string';  
4  
5 var_dump(filter_var($str, FILTER_VALIDATE_BOOLEAN, FILTER_NULL_ON_FAILURE));  
6 var_dump(filter_var($str, FILTER_VALIDATE_BOOLEAN, ['flags' =>  
7     FILTER_NULL_ON_FAILURE]));  
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.3. filter.examples.sanitization.php

URL: <https://www.php.net/manual/en/filter.examples.sanitization.php>

Code snippet:

```
1 <?php
2 $a = 'joe@example.org';
3 $b = 'bogus - at - example dot org';
4 $c = '(bogus@example.org)';
5
6 $sanitized_a = filter_var($a, FILTER_SANITIZE_EMAIL);
7 if (filter_var($sanitized_a, FILTER_VALIDATE_EMAIL)) {
8     echo "This (a) sanitized email address is considered valid.\n";
9 }
10
11 $sanitized_b = filter_var($b, FILTER_SANITIZE_EMAIL);
12 if (filter_var($sanitized_b, FILTER_VALIDATE_EMAIL)) {
13     echo "This sanitized email address is considered valid.";
14 } else {
15     echo "This (b) sanitized email address is considered invalid.\n";
16 }
17
18 $sanitized_c = filter_var($c, FILTER_SANITIZE_EMAIL);
19 if (filter_var($sanitized_c, FILTER_VALIDATE_EMAIL)) {
20     echo "This (c) sanitized email address is considered valid.\n";
21     echo "Before: $c\n";
22     echo "After:  $sanitized_c\n";
23 }
24 ?>
```

Success

No security issues were identified in the code snippet

4.1.4. function.readdir.php

URL: <https://www.php.net/manual/en/function.readdir.php>

4.1.4.1. Example 1

Code snippet:

```
1 <?php
2
```

```

3 if ($handle = opendir('/path/to/files')) {
4     echo "Directory handle: $handle\n";
5     echo "Entries:\n";
6
7     /* This is the correct way to loop over the directory. */
8     while (false !== ($entry = readdir($handle))) {
9         echo "$entry\n";
10    }
11
12    /* This is the WRONG way to loop over the directory. */
13    while ($entry = readdir($handle)) {
14        echo "$entry\n";
15    }
16
17    closedir($handle);
18 }
19 ?>

```

Success

No security issues were identified in the code snippet

4.1.4.2. Example 2

Code snippet:

```

1 <?php
2 if ($handle = opendir('.')) {
3     while (false !== ($entry = readdir($handle))) {
4         if ($entry != "." && $entry != "..") {
5             echo "$entry\n";
6         }
7     }
8     closedir($handle);
9 }
10 ?>

```

Success

No security issues were identified in the code snippet

4.1.5. function.closedir.php

URL: <https://www.php.net/manual/en/function.closedir.php>

Code snippet:

```
1 <?php
2 $dir = "/etc/php5/";
3
4 // Open a known directory, read directory into variable and then close
5 if (is_dir($dir)) {
6     if ($dh = opendir($dir)) {
7         $directory = readdir($dh);
8         closedir($dh);
9     }
10 }
11 ?>
```

Success

No security issues were identified in the code snippet

4.1.6. function.dirname.php

URL: <https://www.php.net/manual/en/function.dirname.php>

Code snippet:

```
1 <?php
2 echo dirname("/etc/passwd") . PHP_EOL;
3 echo dirname("/etc/") . PHP_EOL;
4 echo dirname(".") . PHP_EOL;
5 echo dirname("C:\\") . PHP_EOL;
6 echo dirname("/usr/local/lib", 2);
```

Success

No security issues were identified in the code snippet

4.1.7. function.is-dir.php

URL: <https://www.php.net/manual/en/function.is-dir.php>

Code snippet:

```
1 <?php
2 var_dump(is_dir('a_file.txt'));
3 var_dump(is_dir('bogus_dir/abc'));
4
5 var_dump(is_dir('../')); //one dir up
6 ?>
```

Success

No security issues were identified in the code snippet

4.1.8. function.mkdir.php

URL: <https://www.php.net/manual/en/function.mkdir.php>

4.1.8.1. Example 1

Code snippet:

```
1 <?php
2 mkdir("/path/to/my/dir", 0700);
3 ?>
```

Success

No security issues were identified in the code snippet

4.1.8.2. Example 2

Code snippet:

```
1 <?php
2 // Desired directory structure
3 $structure = './depth1/depth2/depth3/';
4
5 // To create the nested structure, the $recursive parameter
6 // to mkdir() must be specified.
7
8 if (!mkdir($structure, 0777, true)) {
9     die('Failed to create directories...');
10 }
11
12 // ...
13 ?>
```

Success

No security issues were identified in the code snippet

4.1.9. function.rmdir.php

URL: <https://www.php.net/manual/en/function.rmdir.php>

Code snippet:

```
1 <?php
2 if (!is_dir('examples')) {
3     mkdir('examples');
4 }
5
6 rmdir('examples');
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.10. function.chdir.php

URL: <https://www.php.net/manual/en/function.chdir.php>

Code snippet:

```
1 <?php
2
3 // current directory
4 echo getcwd() . "\n";
5
6 chdir('public_html');
7
8 // current directory
9 echo getcwd() . "\n";
10
11 ?>
```

Success

No security issues were identified in the code snippet

4.1.11. function.chroot.php

URL: <https://www.php.net/manual/en/function.chroot.php>

Code snippet:

```
1 <?php
2 chroot("/path/to/your/chroot/");
3 echo getcwd();
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.12. function.closedir.php

URL: <https://www.php.net/manual/en/function.closedir.php>

4.1.12.1. Example

Code snippet:

```
1 <?php
2 $dir = "/etc/php5/";
3
4 // Open a known directory, read directory into variable and then close
5 if (is_dir($dir)) {
6     if ($dh = opendir($dir)) {
7         $directory = readdir($dh);
8         closedir($dh);
9     }
10 }
11 ?>
```

Success

No security issues were identified in the code snippet

4.1.13. function.opendir.php

URL: <https://www.php.net/manual/en/function.opendir.php>

Code snippet:

```
1 <?php
2 $dir = "/etc/php5/";
3
4 // Open a known directory, and proceed to read its contents
5 if (is_dir($dir)) {
6     if ($dh = opendir($dir)) {
7         while (($file = readdir($dh)) !== false) {
8             echo "filename: $file : filetype: " . filetype($dir . $file) . "\n";
9         }
10        closedir($dh);
11    }
12 }
13 ?>
```

Success

No security issues were identified in the code snippet

4.1.14. function.getcwd.php

URL: <https://www.php.net/manual/en/function.getcwd.php>

Code snippet:

```
1 <?php
2
3 // current directory
4 echo getcwd() . "\n";
5
6 chdir('cvs');
7
8 // current directory
9 echo getcwd() . "\n";
10
11 ?>
```

Success

No security issues were identified in the code snippet

4.1.15. function.scandir.php

URL: <https://www.php.net/manual/en/function.scandir.php>

Code snippet:

```
1 <?php
2 $dir    = '/tmp';
3 $files1 = scandir($dir);
4 $files2 = scandir($dir, SCANDIR_SORT_DESCENDING);
5
6 print_r($files1);
7 print_r($files2);
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.16. function.glob.php

URL: <https://www.php.net/manual/en/function.glob.php>

4.1.16.1. Example 1

Code snippet:

```
1 <?php
2 foreach (glob("*.txt") as $filename) {
3     echo "$filename size " . filesize($filename) . "\n";
4 }
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.16.2. Example 2

Code snippet:

```
1 <?php
2 foreach (glob("path/*/*.{txt,md}", \GLOB_BRACE) as $filename) {
3     echo "$filename\n";
4 }
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.17. function.sort.php

URL: <https://www.php.net/manual/en/function.sort.php>

4.1.17.1. Example 1

Code snippet:

```
1 <?php
2
3 $fruits = array("lemon", "orange", "banana", "apple");
4 sort($fruits);
5 foreach ($fruits as $key => $val) {
6     echo "fruits[" . $key . "] = " . $val . "\n";
7 }
8
9 ?>
```

Success

No security issues were identified in the code snippet

4.1.17.2. Example 2

Code snippet:

```
1 <?php
2
3 $fruits = array(
4     "Orange1", "orange2", "Orange3", "orange20"
5 );
6 sort($fruits, SORT_NATURAL | SORT_FLAG_CASE);
7 foreach ($fruits as $key => $val) {
8     echo "fruits[" . $key . "] = " . $val . "\n";
9 }
10
11 ?>
```

Success

No security issues were identified in the code snippet

4.1.18. function.finfo-open.php

URL: <https://www.php.net/manual/en/function.finfo-open.php>

4.1.18.1. Example 1

Code snippet:

```
1 <?php
2 $finfo = new finfo(FILEINFO_MIME, "/usr/share/misc/magic"); // return mime type
   ala mimetype extension
3
4 /* get mime-type for a specific file */
5 $filename = "/usr/local/something.txt";
6 echo $finfo->file($filename);
7
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.18.2. Example 2

Code snippet:

```
1 <?php
2 $finfo = finfo_open(FILEINFO_MIME, "/usr/share/misc/magic"); // return mime type
   ala mimetype extension
3
4 if (!$finfo) {
5     echo "Opening fileinfo database failed";
6     exit();
7 }
8
9 /* get mime-type for a specific file */
10 $filename = "/usr/local/something.txt";
11 echo finfo_file($finfo, $filename);
12
13 /* close connection */
14 finfo_close($finfo);
15 ?>
```

Success

No security issues were identified in the code snippet

4.1.19. finfo-buffer.php

URL: <https://www.php.net/manual/en/function.finfo-buffer.php>

Code snippet:

```
1 <?php
2 $finfo = new finfo(FILEINFO_MIME);
3 echo $finfo->buffer($_POST["script"]) . "\n";
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.20. function.finfo-file.php

URL: <https://www.php.net/manual/en/function.finfo-file.php>

Code snippet:

```
1 <?php
2 $finfo = finfo_open(FILEINFO_MIME_TYPE); // return mime type aka mimetype
   extension
3 foreach (glob("**") as $filename) {
4     echo finfo_file($finfo, $filename) . "\n";
5 }
6 finfo_close($finfo);
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.21. function.mime-content-type.php

URL: <https://www.php.net/manual/en/function.mime-content-type.php>

Code snippet:

```
1 <?php
2 echo mime_content_type('php.gif') . "\n";
3 echo mime_content_type('test.php');
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.22. function.basename.php

URL: <https://www.php.net/manual/en/function.basename.php>

Code snippet:

```
1 <?php
2 echo "1) ".basename("/etc/sudoers.d", ".d").PHP_EOL;
3 echo "2) ".basename("/etc/sudoers.d").PHP_EOL;
4 echo "3) ".basename("/etc/passwd").PHP_EOL;
5 echo "4) ".basename("/etc/").PHP_EOL;
6 echo "5) ".basename(".").PHP_EOL;
7 echo "6) ".basename("/");
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.23. function.chgrp.php

URL: <https://www.php.net/manual/en/function.chgrp.php>

Code snippet:

```
1 <?php
2 $filename = 'shared_file.txt';
3 $format = "%s's Group ID @ %s: %d\n";
4 printf($format, $filename, date('r'), filegroup($filename));
5 chgrp($filename, 8);
6 clearstatcache(); // do not cache filegroup() results
7 printf($format, $filename, date('r'), filegroup($filename));
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.24. function.chown.ph

URL: <https://www.php.net/manual/en/function.chown.php>

Code snippet:

```
1 <?php
2
3 // File name and username to use
4 $file_name= "foo.php";
5 $path = "/home/sites/php.net/public_html/sandbox/" . $file_name ;
6 $user_name = "root";
7
8 // Set the user
9 chown($path, $user_name);
10
11 // Check the result
12 $stat = stat($path);
13 print_r(posix_getpwuid($stat['uid']));
14
15 ?>
```

Success

No security issues were identified in the code snippet

4.1.25. function.fileperms.php

URL: <https://www.php.net/manual/en/function.fileperms.php>

4.1.25.1. Example 1

Code snippet:

```
1 <?php
2 echo substr(sprintf('%o', fileperms('/tmp')), -4);
3 echo substr(sprintf('%o', fileperms('/etc/passwd')), -4);
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.25.2. Example 2

Code snippet:

```
1 <?php
2 $perms = fileperms('/etc/passwd');
3
4 switch ($perms & 0xF000) {
5     case 0xC000: // socket
6         $info = 's';
7         break;
8     case 0xA000: // symbolic link
9         $info = 'l';
10        break;
11     case 0x8000: // regular
12        $info = 'r';
13        break;
14     case 0x6000: // block special
15        $info = 'b';
16        break;
17     case 0x4000: // directory
18        $info = 'd';
19        break;
20     case 0x2000: // character special
21        $info = 'c';
22        break;
23     case 0x1000: // FIFO pipe
24        $info = 'p';
```

```

25     break;
26     default: // unknown
27         $info = 'u';
28 }
29
30 // Owner
31 $info .= (($perms & 0x0100) ? 'r' : '-');
32 $info .= (($perms & 0x0080) ? 'w' : '-');
33 $info .= (($perms & 0x0040) ?
34     (($perms & 0x0800) ? 's' : 'x' ) :
35     (($perms & 0x0800) ? 'S' : '-'));
36
37 // Group
38 $info .= (($perms & 0x0020) ? 'r' : '-');
39 $info .= (($perms & 0x0010) ? 'w' : '-');
40 $info .= (($perms & 0x0008) ?
41     (($perms & 0x0400) ? 's' : 'x' ) :
42     (($perms & 0x0400) ? 'S' : '-'));
43
44 // World
45 $info .= (($perms & 0x0004) ? 'r' : '-');
46 $info .= (($perms & 0x0002) ? 'w' : '-');
47 $info .= (($perms & 0x0001) ?
48     (($perms & 0x0200) ? 't' : 'x' ) :
49     (($perms & 0x0200) ? 'T' : '-'));
50
51 echo $info;
52 ?>

```

Success

No security issues were identified in the code snippet

4.1.26. function.stat.php

URL: <https://www.php.net/manual/en/function.stat.php>

4.1.26.1. Example 1

Code snippet:

```
1 <?php
2 /* Get file stat */
3 $stat = stat('C:\php\php.exe');
4
5 /*
6  * Print file access time, this is the same
7  * as calling fileatime()
8  */
9 echo 'Access time: ' . $stat['atime'];
10
11 /*
12  * Print file modification time, this is the
13  * same as calling filemtime()
14  */
15 echo 'Modification time: ' . $stat['mtime'];
16
17 /* Print the device number */
18 echo 'Device number: ' . $stat['dev'];
19 ?>
```

Success

No security issues were identified in the code snippet

4.1.26.2. Example 2

Code snippet:

```
1 <?php
2 /* Get file stat */
3 $stat = stat('C:\php\php.exe');
4
5 /* Did we failed to get stat information? */
6 if (!$stat) {
7     echo 'stat() call failed...';
8 } else {
9     /*
```

```
10     * We want the access time to be 1 week
11     * after the current access time.
12     */
13     $atime = $stat['atime'] + 604800;
14
15     /* Touch the file */
16     if (!touch('some_file.txt', time(), $atime)) {
17         echo 'Failed to touch file...';
18     } else {
19         echo 'touch() returned success...';
20     }
21 }
22 ?>
```

Success

No security issues were identified in the code snippet

4.1.27. function.lstat.php

URL: <https://www.php.net/manual/en/function.lstat.php>

Code snippet:

```
1 <?php
2 symlink('uploads.php', 'uploads');
3
4 // Contrast information for uploads.php and uploads
5 array_diff(stat('uploads'), lstat('uploads'));
6 ?>
```

Success

No security issues were identified in the code snippet

4.1.28. features.remote-files.php

URL: <https://www.php.net/manual/en/features.remote-files.php>

4.1.28.1. Example 1

Code snippet:

```
1 <?php
2 $file = fopen ("http://www.example.com/", "r");
3 if (!$file) {
4     echo "<p>Unable to open remote file.\n";
5     exit;
6 }
7 while (!feof ($file)) {
8     $line = fgets ($file, 1024);
9     /* This only works if the title and its tags are on one line */
10    if (preg_match ("@\<title\>(.*?)\</title\>@i", $line, $out)) {
11        $title = $out[1];
12        break;
13    }
14 }
15 fclose($file);
16 ?>
```

Success

No security issues were identified in the code snippet

4.1.28.2. Example 2

Code snippet:

```
1 <?php
2 $file = fopen ("ftp://ftp.example.com/incoming/outputfile", "w");
3 if (!$file) {
4     echo "<p>Unable to open remote file for writing.\n";
5     exit;
6 }
7 /* Write the data here. */
8 fwrite ($file, $_SERVER['HTTP_USER_AGENT'] . "\n");
9 fclose ($file);
10 ?>
```

Success

No security issues were identified in the code snippet

4.1.29. function.clearstatcache.php

URL: <https://www.php.net/manual/en/function.clearstatcache.php> Code snippet:

```
1 <?php
2 $file = 'output_log.txt';
3
4 function get_owner($file)
5 {
6     $stat = stat($file);
7     $user = posix_getpwuid($stat['uid']);
8     return $user['name'];
9 }
10
11 $format = "UID @ %s: %s\n";
12
13 printf($format, date('r'), get_owner($file));
14
15 chown($file, 'ross');
16 printf($format, date('r'), get_owner($file));
17
18 clearstatcache();
19 printf($format, date('r'), get_owner($file));
20 ?>
```

Success

No security issues were identified in the code snippet

4.1.30. function.copy.php

URL: <https://www.php.net/manual/en/function.copy.php>

Code snippet:

```
1 <?php
2 $file = 'example.txt';
3 $newfile = 'example.txt.bak';
4
5 if (!copy($file, $newfile)) {
6     echo "failed to copy $file...\n";
7 }
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.31. function.rmdir.php

URL: <https://www.php.net/manual/en/function.rmdir.php>

Code snippet:

```
1 <?php
2 if (!is_dir('examples')) {
3     mkdir('examples');
4 }
5
6 rmdir('examples');
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.32. function.unset.php

URL: <https://www.php.net/manual/en/function.unset.php>

4.1.32.1. Example 1

Code snippet:

```
1 <?php
2 function destroy_foo()
3 {
4     global $foo;
5     unset($foo);
6 }
7
8 $foo = 'bar';
9 destroy_foo();
10 echo $foo;
11 ?>
```

Success

No security issues were identified in the code snippet

4.1.32.2. Example 2

Code snippet:

```
1 <?php
2 function foo()
3 {
4     unset($GLOBALS['bar']);
5 }
6
7 $bar = "something";
8 foo();
9 ?>
```

Success

No security issues were identified in the code snippet

4.1.32.3. Example 3

Code snippet:

```
1 <?php
2 function foo(&$bar)
3 {
4     unset($bar);
5     $bar = "blah";
6 }
7
8 $bar = 'something';
9 echo "$bar\n";
10
11 foo($bar);
12 echo "$bar\n";
13 ?>
```

Success

No security issues were identified in the code snippet

4.1.32.4. Example 4

Code snippet:

```
1 <?php
2 function foo()
3 {
4     static $bar;
5     $bar++;
6     echo "Before unset: $bar, ";
7     unset($bar);
8     $bar = 23;
9     echo "after unset: $bar\n";
10 }
11
12 foo();
13 foo();
14 foo();
15 ?>
```

Success

No security issues were identified in the code snippet

4.1.33. function.disk-free-space.php

URL: <https://www.php.net/manual/en/function.disk-free-space.php>

Code snippet:

```
1 <?php
2 // $df contains the number of bytes available on "/"
3 $df = disk_free_space("/");
4
5 // On Windows:
6 $df_c = disk_free_space("C:");
7 $df_d = disk_free_space("D:");
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.34. function.disk-total-space.php

URL: <https://www.php.net/manual/en/function.disk-total-space.php>

Code snippet:

```
1 <?php
2 // $ds contains the total number of bytes available on "/"
3 $ds = disk_total_space("/");
4
5 // On Windows:
6 $ds = disk_total_space("C:");
7 $ds = disk_total_space("D:");
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.35. function.feof.php

URL: <https://www.php.net/manual/en/function.feof.php>

4.1.35.1. Example 1

Code snippet:

```
1 <?php
2 function safe_feof($fp, &$start = NULL) {
3     $start = microtime(true);
4
5     return feof($fp);
6 }
7
8 /* Assuming $fp is previously opened by fsockopen() */
9
10 $start = NULL;
11 $timeout = ini_get('default_socket_timeout');
12
13 while(!safe_feof($fp, $start) && (microtime(true) - $start) < $timeout)
14 {
15     /* Handle */
16 }
17 ?>
```

Success

No security issues were identified in the code snippet

4.1.35.2. Example 2

Code snippet:

```
1 <?php
2 // if file can not be read or doesn't exist fopen function returns FALSE
3 $file = @fopen("no_such_file", "r");
4
5 // FALSE from fopen will issue warning and result in infinite loop here
6 while (!feof($file)) {
7 }
8
9 fclose($file);
10 ?>
```

Success

No security issues were identified in the code snippet

4.1.36. function.fgetc.php

URL: <https://www.php.net/manual/en/function.fgetc.php>

Code snippet:

```
1 <?php
2 $fp = fopen('somefile.txt', 'r');
3 if (!$fp) {
4     echo 'Could not open file somefile.txt';
5 }
6 while (false !== ($char = fgetc($fp))) {
7     echo "$char\n";
8 }
9 ?>
```

Success

No security issues were identified in the code snippet

4.1.37. function.fgetcsv.php

URL: <https://www.php.net/manual/en/function.fgetcsv.php>

Code snippet:

```
1 <?php
2 $row = 1;
3 if (($handle = fopen("test.csv", "r")) !== FALSE) {
4     while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
5         $num = count($data);
6         echo "<p> $num fields in line $row: <br /></p>\n";
7         $row++;
8         for ($c=0; $c < $num; $c++) {
9             echo $data[$c] . "<br />\n";
10        }
11    }
12    fclose($handle);
13 }
14 ?>
```

Success

No security issues were identified in the code snippet

4.1.38. function.fgets.php

URL: <https://www.php.net/manual/en/function.fgets.php>

Code snippet:

```
1 <?php
2
3 $fp = @fopen("/tmp/inputfile.txt", "r");
4
5 if ($fp) {
6     while (($buffer = fgets($fp, 4096)) !== false) {
7         echo $buffer, PHP_EOL;
8     }
9
10    if (!feof($fp)) {
11        echo "Error: unexpected fgets() fail\n";
12    }
13
14    fclose($fp);
15 }
16
17 ?>
```

Success

No security issues were identified in the code snippet

4.1.39. function.fgetss.php

URL: <https://www.php.net/manual/en/function.fgetss.php>

Code snippet:

```
1 <?php
2 $str = <<<EOD
3 <html><body>
4 <p>Welcome! Today is the <?php echo(date('jS')); ?> of <?= date('F'); ?>.</p>
5 </body></html>
6 Text outside of the HTML block.
7 EOD;
8 file_put_contents('sample.php', $str);
9
10 $handle = @fopen("sample.php", "r");
11 if ($handle) {
12     while (!feof($handle)) {
13         $buffer = fgetss($handle, 4096);
14         echo $buffer;
15     }
16     fclose($handle);
17 }
18 ?>
```

Success

No security issues were identified in the code snippet

4.1.40. function.file.php

URL: <https://www.php.net/manual/en/function.file.php>

Code snippet:

```
1 <?php
2 // Get a file into an array. In this example we'll go through HTTP to get
3 // the HTML source of a URL.
4 $lines = file('http://www.example.com/');
5
6 // Loop through our array, show HTML source as HTML source; and line numbers
  too.
7 foreach ($lines as $line_num => $line) {
8     echo "Line #<b>{$line_num}</b> : " . htmlspecialchars($line) . "<br />\n";
9 }
10
11 // Using the optional flags parameter
12 $trimmed = file('somefile.txt', FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
13 ?>
```

Success

No security issues were identified in the code snippet

4.1.41. function.file-get-contents.php

URL: <https://www.php.net/manual/en/function.file-get-contents.php>

4.1.41.1. Example 1

Code snippet:

```
1 <?php
2 $homepage = file_get_contents('http://www.example.com/');
3 echo $homepage;
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.41.2. Example 2

Code snippet:

```
1 <?php
2 // If strict types are enabled i.e. declare(strict_types=1);
3 $file = file_get_contents('./people.txt', true);
4 // Otherwise
5 $file = file_get_contents('./people.txt', FILE_USE_INCLUDE_PATH);
6 ?>
```

Success

No security issues were identified in the code snippet

4.1.41.3. Example 3

Code snippet:

```
1 <?php
2 // Read 14 characters starting from the 21st character
3 $section = file_get_contents('./people.txt', FALSE, NULL, 20, 14);
4 var_dump($section);
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.41.4. Example 4

Code snippet:

```
1 <?php
2 // Create a stream
3 $opts = array(
4     'http'=>array(
5         'method'=>"GET",
6         'header'=>"Accept-language: en\r\n" .
7                 "Cookie: foo=bar\r\n"
8     )
9 );
10
11 $context = stream_context_create($opts);
12
13 // Open the file using the HTTP headers set above
14 $file = file_get_contents('http://www.example.com/', false, $context);
15 ?>
```

Success

No security issues were identified in the code snippet

4.1.42. function.file-exists.php

URL: <https://www.php.net/manual/en/function.file-exists.php>

Code snippet:

```
1 <?php
2 $filename = '/path/to/foo.txt';
3
4 if (file_exists($filename)) {
5     echo "The file $filename exists";
6 } else {
7     echo "The file $filename does not exist";
8 }
9 ?>
```

Success

No security issues were identified in the code snippet

4.1.43. function.is-readable.php

URL: <https://www.php.net/manual/en/function.is-readable.php>

Code snippet:

```
1 <?php
2 $filename = 'test.txt';
3 if (is_readable($filename)) {
4     echo 'The file is readable';
5 } else {
6     echo 'The file is not readable';
7 }
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.44. function.is-writable.php

URL: <https://www.php.net/manual/en/function.is-writable.php>

Code snippet:

```
1 <?php
2 $filename = 'test.txt';
3 if (is_writable($filename)) {
4     echo 'The file is writable';
5 } else {
6     echo 'The file is not writable';
7 }
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.45. function.is-file.php

URL: <https://www.php.net/manual/en/function.is-file.php>

Code snippet:

```
1 <?php
2 var_dump(is_file('a_file.txt')) . "\n";
3 var_dump(is_file('/usr/bin/')) . "\n";
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.46. function.file-put-contents.php

URL: <https://www.php.net/manual/en/function.file-put-contents.php>

4.1.46.1. Example 1

Code snippet:

```
1 <?php
2 $file = 'people.txt';
3 // Open the file to get existing content
4 $current = file_get_contents($file);
5 // Append a new person to the file
6 $current .= "John Smith\n";
7 // Write the contents back to the file
8 file_put_contents($file, $current);
9 ?>
```

Success

No security issues were identified in the code snippet

4.1.46.2. Example 2

Code snippet:

```
1 <?php
2 $file = 'people.txt';
3 // The new person to add to the file
4 $person = "John Smith\n";
5 // Write the contents to the file,
6 // using the FILE_APPEND flag to append the content to the end of the file
7 // and the LOCK_EX flag to prevent anyone else writing to the file at the same
8 // time
9 file_put_contents($file, $person, FILE_APPEND | LOCK_EX);
10 ?>
```

Success

No security issues were identified in the code snippet

4.1.47. function.fileatime.php

URL: <https://www.php.net/manual/en/function.fileatime.php>

Code snippet:

```
1 <?php
2
3 // outputs e.g.  somefile.txt was last accessed: December 29 2002 22:16:23.
4
5 $filename = 'somefile.txt';
6 if (file_exists($filename)) {
7     echo "$filename was last accessed: " . date("F d Y H:i:s.",
8         fileatime($filename));
9 }
10 ?>
```

Success

No security issues were identified in the code snippet

4.1.48. function.filectime.php

URL: <https://www.php.net/manual/en/function.filectime.php>

Code snippet:

```
1 <?php
2
3 // outputs e.g.  somefile.txt was last changed: December 29 2002 22:16:23.
4
5 $filename = 'somefile.txt';
6 if (file_exists($filename)) {
7     echo "$filename was last changed: " . date("F d Y H:i:s.",
8         filectime($filename));
9 }
10 ?>
```

Success

No security issues were identified in the code snippet

4.1.49. function.filegroup.php

URL: <https://www.php.net/manual/en/function.filegroup.php>

Code snippet:

```
1 <?php
2 $filename = 'index.php';
3 print_r(posix_getgrgid(filegroup($filename)));
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.50. function.posix-getgrgid.php

URL: <https://www.php.net/manual/en/function.posix-getgrgid.php>

Code snippet:

```
1 <?php
2
3 $groupid = posix_getegid();
4 $groupinfo = posix_getgrgid($groupid);
5
6 print_r($groupinfo);
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.51. function.fileinode.php

URL: <https://www.php.net/manual/en/function.fileinode.php>

Code snippet:

```
1 <?php
2 $filename = 'index.php';
3 if (getmyinode() == fileinode($filename)) {
4     echo 'You are checking the current file.';
5 }
6 ?>
```

Success

No security issues were identified in the code snippet

4.1.52. function.filemtime.php

URL: <https://www.php.net/manual/en/function.filemtime.php>

Code snippet:

```
1 <?php
2 // outputs e.g. somefile.txt was last modified: December 29 2002 22:16:23.
3
4 $filename = 'somefile.txt';
5 if (file_exists($filename)) {
6     echo "$filename was last modified: " . date ("F d Y H:i:s.",
7         filemtime($filename));
8 }
?>
```

Success

No security issues were identified in the code snippet

4.1.53. function.fileowner.php

URL: <https://www.php.net/manual/en/function.fileowner.php>

Code snippet:

```
1 <?php
2 $filename = 'index.php';
3 print_r(posix_getpwuid(fileowner($filename)));
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.54. function.fileperms.php

URL: <https://www.php.net/manual/en/function.fileperms.php>

4.1.54.1. Example 1

Code snippet:

```
1 <?php
2 echo substr(sprintf('%o', fileperms('/tmp')), -4);
3 echo substr(sprintf('%o', fileperms('/etc/passwd')), -4);
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.54.2. Example 2

Code snippet:

```
1 <?php
2 $perms = fileperms('/etc/passwd');
3
4 switch ($perms & 0xF000) {
5     case 0xC000: // socket
6         $info = 's';
7         break;
8     case 0xA000: // symbolic link
9         $info = 'l';
10        break;
11     case 0x8000: // regular
12        $info = 'r';
13        break;
14     case 0x6000: // block special
15        $info = 'b';
16        break;
17     case 0x4000: // directory
18        $info = 'd';
19        break;
20     case 0x2000: // character special
21        $info = 'c';
22        break;
23     case 0x1000: // FIFO pipe
24        $info = 'p';
```

```

25     break;
26     default: // unknown
27         $info = 'u';
28 }
29
30 // Owner
31 $info .= (($perms & 0x0100) ? 'r' : '-');
32 $info .= (($perms & 0x0080) ? 'w' : '-');
33 $info .= (($perms & 0x0040) ?
34     (($perms & 0x0800) ? 's' : 'x' ) :
35     (($perms & 0x0800) ? 'S' : '-'));
36
37 // Group
38 $info .= (($perms & 0x0020) ? 'r' : '-');
39 $info .= (($perms & 0x0010) ? 'w' : '-');
40 $info .= (($perms & 0x0008) ?
41     (($perms & 0x0400) ? 's' : 'x' ) :
42     (($perms & 0x0400) ? 'S' : '-'));
43
44 // World
45 $info .= (($perms & 0x0004) ? 'r' : '-');
46 $info .= (($perms & 0x0002) ? 'w' : '-');
47 $info .= (($perms & 0x0001) ?
48     (($perms & 0x0200) ? 't' : 'x' ) :
49     (($perms & 0x0200) ? 'T' : '-'));
50
51 echo $info;
52 ?>

```

Success

No security issues were identified in the code snippet

4.1.55. function.filesize.php

URL: <https://www.php.net/manual/en/function.filesize.php>

Code snippet:

```
1 <?php
2
3 // outputs e.g.  somefile.txt: 1024 bytes
4
5 $filename = 'somefile.txt';
6 echo $filename . ': ' . filesize($filename) . ' bytes';
7
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.56. function.filetype.php

URL: <https://www.php.net/manual/en/function.filetype.php>

Code snippet:

```
1 <?php
2
3 echo filetype('/etc/passwd');
4 echo "\n";
5 echo filetype('/etc/');
6
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.57. function.fnmatch.php

URL: <https://www.php.net/manual/en/function.fnmatch.php>

Code snippet:

```
1 <?php
2 if (fnmatch("*gr[ae]y", $color)) {
3     echo "some form of gray ...";
4 }
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.58. function.fopen.php

URL: <https://www.php.net/manual/en/function.fopen.php>

Code snippet:

```
1  
2 <?php  
3 $handle = fopen("c:\\folder\\resource.txt", "r");  
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.59. function.readfile.php

URL: <https://www.php.net/manual/en/function.readfile.php>

Code snippet:

```
1 <?php
2 $file = 'monkey.gif';
3
4 if (file_exists($file)) {
5     header('Content-Description: File Transfer');
6     header('Content-Type: application/octet-stream');
7     header('Content-Disposition: attachment; filename="'.basename($file).'");
8     header('Expires: 0');
9     header('Cache-Control: must-revalidate');
10    header('Pragma: public');
11    header('Content-Length: ' . filesize($file));
12    readfile($file);
13    exit;
14 }
15 ?>
```

Success

No security issues were identified in the code snippet

4.1.60. function.include.php

URL: <https://www.php.net/manual/en/function.include.php>

4.1.60.1. Example 1

Code snippet:

```
1 vars.php
2 <?php
3
4 $color = 'green';
5 $fruit = 'apple';
6
7 ?>
8
9 test.php
10 <?php
11
12 echo "A $color $fruit"; // A
13
14 include 'vars.php';
15
16 echo "A $color $fruit"; // A green apple
17
18 ?>
```

Success

No security issues were identified in the code snippet

4.1.60.2. Example 2

Code snippet:

```
1 <?php
2
3 function foo()
4 {
5     global $color;
6
7     include 'vars.php';
8
9     echo "A $color $fruit";
10 }
```

```

11
12 /* vars.php is in the scope of foo() so      *
13 * $fruit is NOT available outside of this *
14 * scope. $color is because we declared it *
15 * as global.                               */
16
17 foo();                                     // A green apple
18 echo "A $color $fruit"; // A green
19
20 ?>

```

Success

No security issues were identified in the code snippet

4.1.60.3. Example 3

Code snippet:

```

1 <?php
2
3 /* This example assumes that www.example.com is configured to parse .php
4 * files and not .txt files. Also, 'Works' here means that the variables
5 * $foo and $bar are available within the included file. */
6
7 // Won't work; file.txt wasn't handled by www.example.com as PHP
8 include 'http://www.example.com/file.txt?foo=1&bar=2';
9
10 // Won't work; looks for a file named 'file.php?foo=1&bar=2' on the
11 // local filesystem.
12 include 'file.php?foo=1&bar=2';
13
14 // Works.
15 include 'http://www.example.com/file.php?foo=1&bar=2';
16 ?>

```

Success

No security issues were identified in the code snippet

4.1.60.4. Example 4

Code snippet:

```

1 <?php
2 // won't work, evaluated as include('vars.php') == TRUE), i.e. include('1')
3 if (include('vars.php') == TRUE) {
4     echo 'OK';
5 }
6
7 // works
8 if ((include 'vars.php') == TRUE) {
9     echo 'OK';
10 }
11 ?>

```

Success

No security issues were identified in the code snippet

4.1.60.5. Example 5

Code snippet:

```

1 return.php
2 <?php
3
4 $var = 'PHP';
5
6 return $var;
7
8 ?>
9
10 noreturn.php
11 <?php
12
13 $var = 'PHP';
14
15 ?>
16
17 testreturns.php
18 <?php
19
20 $foo = include 'return.php';
21
22 echo $foo; // prints 'PHP'
23

```

```
24 $bar = include 'noreturn.php';
25
26 echo $bar; // prints 1
27
28 ?>
```

Success

No security issues were identified in the code snippet

4.1.60.6. Example 6

Code snippet:

```
1 <?php
2 $string = get_include_contents('somefile.php');
3
4 function get_include_contents($filename) {
5     if (is_file($filename)) {
6         ob_start();
7         include $filename;
8         return ob_get_clean();
9     }
10    return false;
11 }
12
13 ?>
```

Success

No security issues were identified in the code snippet

4.1.61. function.fsockopen.php

URL: <https://www.php.net/manual/en/function.fsockopen.php>

4.1.61.1. Example 1

Code snippet:

```
1 <?php
2 $fp = fsockopen("www.example.com", 80, $errno, $errstr, 30);
3 if (!$fp) {
4     echo "$errstr ($errno)<br />\n";
5 } else {
6     $out = "GET / HTTP/1.1\r\n";
7     $out .= "Host: www.example.com\r\n";
8     $out .= "Connection: Close\r\n\r\n";
9     fwrite($fp, $out);
10    while (!feof($fp)) {
11        echo fgets($fp, 128);
12    }
13    fclose($fp);
14 }
15 ?>
```

Success

No security issues were identified in the code snippet

4.1.61.2. Example 2

Code snippet:

```
1 <?php
2 $fp = fsockopen("udp://127.0.0.1", 13, $errno, $errstr);
3 if (!$fp) {
4     echo "ERROR: $errno - $errstr<br />\n";
5 } else {
6     fwrite($fp, "\n");
7     echo fread($fp, 26);
8     fclose($fp);
9 }
10 ?>
```

Success

No security issues were identified in the code snippet

4.1.62. function.fread.php

URL: <https://www.php.net/manual/en/function.fread.php>

4.1.62.1. Example 3.2

Code snippet:

```
1 <?php
2 $handle = fopen("http://www.example.com/", "rb");
3 if (FALSE === $handle) {
4     exit("Failed to open stream to URL");
5 }
6
7 $contents = '';
8
9 while (!feof($handle)) {
10     $contents .= fread($handle, 8192);
11 }
12 fclose($handle);
13 ?>
```

Success

No security issues were identified in the code snippet

4.1.63. function.fwrite.php

URL: <https://www.php.net/manual/en/function.fwrite.php>

4.1.63.1. Example 1

Code snippet:

```
1 <?php
2 $filename = 'test.txt';
3 $somecontent = "Add this to the file\n";
4
5 // Let's make sure the file exists and is writable first.
6 if (is_writable($filename)) {
7
8     // In our example we're opening $filename in append mode.
9     // The file pointer is at the bottom of the file hence
10    // that's where $somecontent will go when we fwrite() it.
11    if (!$fp = fopen($filename, 'a')) {
12        echo "Cannot open file ($filename)";
13        exit;
14    }
15
16    // Write $somecontent to our opened file.
17    if (fwrite($fp, $somecontent) === FALSE) {
18        echo "Cannot write to file ($filename)";
19        exit;
20    }
21
22    echo "Success, wrote ($somecontent) to file ($filename)";
23
24    fclose($fp);
25
26 } else {
27    echo "The file $filename is not writable";
28 }
29 ?>
```

Success

No security issues were identified in the code snippet

4.1.63.2. Note 1

Code snippet:

```
1 <?php
2 function fwrite_stream($fp, $string) {
3     for ($written = 0; $written < strlen($string); $written += $fwrite) {
4         $fwrite = fwrite($fp, substr($string, $written));
5         if ($fwrite === false) {
6             return $written;
7         }
8     }
9     return $written;
10 }
11 ?>
```

Success

No security issues were identified in the code snippet

4.1.64. function.unpack.php

URL: <https://www.php.net/manual/en/function.unpack.php>

4.1.64.1. Example 1

Code snippet:

```
1 <?php
2 $binarydata = "\x04\x00\xa0\x00";
3 $array = unpack("cchars/nint", $binarydata);
4 print_r($array);
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.64.2. Example 2

Code snippet:

```
1 <?php
2 $binarydata = "\x04\x00\xa0\x00";
3 $array = unpack("c2chars/nint", $binarydata);
4 print_r($array);
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.64.3. Example 3

Code snippet:

```
1 <?php
2 $binarydata = "\x32\x42\x00\xa0";
3 $array = unpack("c2/n", $binarydata);
4 var_dump($array);
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.65. function.pack.php

URL: <https://www.php.net/manual/en/function.pack.php>

Code snippet:

```
1 <?php
2 $binarydata = pack("nvc*", 0x1234, 0x5678, 65, 66);
3 ?>
```

Success

No security issues were identified in the code snippet

4.1.66. function.password-algos.php

URL: <https://www.php.net/manual/en/function.password-algos.php>

Code snippet:

```
1 <?php
2 print_r(password_algos());
3 ?>
```

Success

No security issues were identified in the code snippet

4.1.67. function.password-hash.php

URL: <https://www.php.net/manual/en/function.password-hash.php>

4.1.67.1. Example 1

Code snippet:

```
1 <?php
2 /**
3  * We just want to hash our password using the current DEFAULT algorithm.
4  * This is presently BCRYPT, and will produce a 60 character result.
5  *
6  * Beware that DEFAULT may change over time, so you would want to prepare
7  * By allowing your storage to expand past 60 characters (255 would be good)
8  */
9 echo password_hash("rasmuslerdorf", PASSWORD_DEFAULT);
10 ?>
```

Success

No security issues were identified in the code snippet

4.1.67.2. Example 2

Code snippet:

```
1 <?php
2 /**
3  * In this case, we want to increase the default cost for BCRYPT to 12.
4  * Note that we also switched to BCRYPT, which will always be 60 characters.
5  */
6 $options = [
7     'cost' => 12,
8 ];
9 echo password_hash("rasmuslerdorf", PASSWORD_BCRYPT, $options);
10 ?>
```

Success

No security issues were identified in the code snippet

4.1.67.3. Example 3

Code snippet:

```
1 <?php
2 /**
3  * This code will benchmark your server to determine how high of a cost you can
4  * afford. You want to set the highest cost that you can without slowing down
5  * you server too much. 10 is a good baseline, and more is good if your servers
6  * are fast enough. The code below aims for ≤ 350 milliseconds stretching time,
7  * which is an appropriate delay for systems handling interactive logins.
8  */
9 $timeTarget = 0.350; // 350 milliseconds
10
11 $cost = 10;
12 do {
13     $cost++;
14     $start = microtime(true);
15     password_hash("test", PASSWORD_BCRYPT, ["cost" => $cost]);
16     $end = microtime(true);
17 } while (($end - $start) < $timeTarget);
18
19 echo "Appropriate Cost Found: " . $cost;
20 ?>
```

Success

No security issues were identified in the code snippet

4.1.67.4. Example 4

Code snippet:

```
1 <?php
2 echo 'Argon2i hash: ' . password_hash('rasmuslerdorf', PASSWORD_ARGON2I);
3 ?>
```

Success

No security issues were identified in the code snippet

4.1.68. function.password-verify.php

URL: <https://www.php.net/manual/en/function.password-verify.php>

Code snippet:

```
1 <?php
2 // See the password_hash() example to see where this came from.
3 $hash = '$2y$10$.vGA109wmRjrwAVXD98HN0gsNpDczlqm3Jq7KnEd1rVAGv3Fykk1a';
4
5 if (password_verify('rasmuslerdorf', $hash)) {
6     echo 'Password is valid!';
7 } else {
8     echo 'Invalid password.';
9 }
10 ?>
```

Success

No security issues were identified in the code snippet

4.1.69. function.crypt.php

URL: <https://www.php.net/manual/en/function.crypt.php>

Code snippet:

```
1 <?php
2 $user_input = 'rasmuslerdorf';
3 $hashed_password = '$6$rounds=100000$Njy4rIPjp0aU$0ACEYGg/aKCY3v808Afyi07CTfZQ8/
  W231Qfh2tRLmfDVFD6XfHk12u6hMr9cYIA4hnpjLNSTRtUwYr9km9Ij/';
4
5 // Validate an existing crypt() hash in a way that is compatible with non-PHP
  software.
6 if (hash_equals($hashed_password, crypt($user_input, $hashed_password))) {
7     echo "Password verified!";
8 }
9 ?>
```

Success

No security issues were identified in the code snippet

4.1.70. function.hash-equals.php

URL: <https://www.php.net/manual/en/function.hash-equals.php>

Code snippet:

```
1 <?php
2 $secretKey = '8uRhAeH89naXfFXKG0Ej';
3
4 // Value and signature are provided by the user, e.g. within the URL
5 // and retrieved using $_GET.
6 $value = 'username=rasmuslerdorf';
7 $signature = '8c35009d3b50caf7f5d2c1e031842e6b7823a1bb781d33c5237cd27b57b5f327';
8
9 if (hash_equals(hash_hmac('sha256', $value, $secretKey), $signature)) {
10     echo "The value is correctly signed.", PHP_EOL;
11 } else {
12     echo "The value was tampered with.", PHP_EOL;
13 }
14 ?>
```

Success

No security issues were identified in the code snippet

4.1.71. function.hash-hmac.php

URL: <https://www.php.net/manual/en/function.hash-hmac.php>

Code snippet:

```
1 <?php
2 echo hash_hmac('sha256', 'The quick brown fox jumped over the lazy dog.',
3 'secret');
```

Success

No security issues were identified in the code snippet

4.1.72. function.hash-hmac-algos.php

URL: <https://www.php.net/manual/en/function.hash-hmac-algos.php>

Code snippet:

```
1 <?php
2 print_r(hash_hmac_algos());
```

Success

No security issues were identified in the code snippet.

4.1.73. function.hash-algos.php

URL: <https://www.php.net/manual/en/function.hash-algos.php>

Code snippet:

```
1 <?php
2 print_r(hash_algos());
3 ?>
```

Success

No security issues were identified in the code snippet

4.1.74. function.hash-hmac-file.php

URL: <https://www.php.net/manual/en/function.hash-hmac-file.php>

Code snippet:

```
1 <?php
2 /* Create a file to calculate hash of */
3 file_put_contents('example.txt', 'The quick brown fox jumped over the lazy
  dog. ');
4
5 echo hash_hmac_file('sha256', 'example.txt', 'secret');
6 ?>
```

Success

No security issues were identified in the code snippet

4.1.75. function.hash-init.php

URL: <https://www.php.net/manual/en/function.hash-init.php>

Code snippet:

```
1 <?php
2 $hash = hash('sha256', 'The quick brown fox jumped over the lazy dog.');
```



```
3
4 $ctx = hash_init('sha256');
5 hash_update($ctx, 'The quick brown fox ');
6 hash_update($ctx, 'jumped over the lazy dog.');
```

```
7 $incremental_hash = hash_final($ctx);
8
9 echo $incremental_hash, PHP_EOL;
10 var_dump($hash === $incremental_hash);
11 ?>
```

Success

No security issues were identified in the code snippet

4.1.76. function.hash-update-stream.php

URL: <https://www.php.net/manual/en/function.hash-update-stream.php>

Code snippet:

```
1 <?php
2 $fp = tmpfile();
3 fwrite($fp, 'jumped over the lazy dog. ');
4 rewind($fp);
5
6 $ctx = hash_init('sha256');
7 hash_update($ctx, 'The quick brown fox ');
8 hash_update_stream($ctx, $fp);
9 echo hash_final($ctx);
10 ?>
```

Success

No security issues were identified in the code snippet

4.1.77. function.password-needs-rehash.php

URL: <https://www.php.net/manual/en/function.password-needs-rehash.php>

Code snippet:

```
1 <?php
2
3 $password = 'rasmuslerdorf';
4 $hash = '$2y$10$YCFsG6eLYca568hBi2pZ0.3LDL5wjgxt1N8w/oLR/jfHsiQwCqTS';
5
6 $algorithm = PASSWORD_BCRYPT;
7 // bcrypt's cost parameter can change over time as hardware improves
8 $options = ['cost' => 12];
9
10 // Verify stored hash against plain-text password
11 if (password_verify($password, $hash)) {
12     // Check if either the algorithm or the options have changed
13     if (password_needs_rehash($hash, $algorithm, $options)) {
14         // If so, create a new hash, and replace the old one
15         $newHash = password_hash($password, $algorithm, $options);
16
17         // Update the user record with the $newHash
18     }
19
20     // Perform the login.
21 }
22 ?>
```

Success

No security issues were identified in the code snippet

4.1.78. function.sodium-crypto-pwhash-str.php

URL: <https://www.php.net/manual/en/function.sodium-crypto-pwhash-str.php>

Code snippet:

```
1 <?php
2 // init gnupg
3 $res = gnupg_init();
4 // not really needed. Clearsign is default
5 gnupg_setsignmode($res,GNUPG_SIG_MODE_CLEAR);
6 // add key with passphrase 'test' for signing
7 gnupg_addsignkey($res,"8660281B6051D071D94B5B230549F9DC851566DC","test");
8 // sign
9 $signed = gnupg_sign($res,"just a test");
10 echo $signed;
11 ?>
```

Success

No security issues were identified in the code snippet

4.1.79. gnupg.examples-clearsign.php

URL: <https://www.php.net/manual/en/gnupg.examples-clearsign.php>

4.1.79.1. Example 1

Code snippet:

```
1 <?php
2 // init gnupg
3 $res = gnupg_init();
4 // not really needed. Clearsign is default
5 gnupg_setsignmode($res,GNUPG_SIG_MODE_CLEAR);
6 // add key with passphrase 'test' for signing
7 gnupg_addsignkey($res,"8660281B6051D071D94B5B230549F9DC851566DC","test");
8 // sign
9 $signed = gnupg_sign($res,"just a test");
10 echo $signed;
11 ?>
```

Success

No security issues were identified in the code snippet

4.1.79.2. Example 2

Code snippet:

```
1 <?php
2 // new class
3 $gnupg = new gnupg();
4 // not really needed. Clearsign is default
5 $gnupg->setsignmode(gnupg::SIG_MODE_CLEAR);
6 // add key with passphrase 'test' for signing
7 $gnupg->addsignkey("8660281B6051D071D94B5B230549F9DC851566DC","test");
8 // sign
9 $signed = $gnupg->sign("just a test");
10 echo $signed;
11 ?>
```

Success

No security issues were identified in the code snippet

4.1.79.3. Example 3

Code snippet:

```
1 <?php
2 // create a new iterator for listing all public keys that matches 'example'
3 $iterator = new gnupg_keylistiterator("example");
4 foreach($iterator as $fingerprint => $userid){
5     echo $fingerprint." -> ".$userid."\n";
6 }
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.80. function.ngettext.php

URL: <https://www.php.net/manual/en/function.ngettext.php>

Code snippet:

```
1 <?php
2
3 setlocale(LC_ALL, 'cs_CZ');
4 printf(ngettext("%d window", "%d windows", 1), 1); // 1 okno
5 printf(ngettext("%d window", "%d windows", 2), 2); // 2 okna
6 printf(ngettext("%d window", "%d windows", 5), 5); // 5 oken
7
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.81. numberformatter.create.php

URL: <https://www.php.net/manual/en/numberformatter.create.php>

4.1.81.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = numfmt_create( 'de_DE', NumberFormatter::DECIMAL );
3 echo numfmt_format($fmt, 1234567.891234567890000)."\n";
4 $fmt = numfmt_create( 'it', NumberFormatter::SPELLOUT );
5 echo numfmt_format($fmt, 1142)."\n";
6 ?>
```

Success

No security issues were identified in the code snippet

4.1.81.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
3 echo $fmt->format(1234567.891234567890000)."\n";
4 $fmt = new NumberFormatter( 'it', NumberFormatter::SPELLOUT );
5 echo $fmt->format(1142)."\n";
6 ?>
```

Success

No security issues were identified in the code snippet

4.1.82. numberformatter.formatcurrency.php

URL: <https://www.php.net/manual/en/numberformatter.formatcurrency.php>

4.1.82.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = numfmt_create( 'de_DE', NumberFormatter::CURRENCY );
3 echo numfmt_format_currency($fmt, 1234567.891234567890000, "EUR")."\n";
4 echo numfmt_format_currency($fmt, 1234567.891234567890000, "RUR")."\n";
5 $fmt = numfmt_create( 'ru_RU', NumberFormatter::CURRENCY );
6 echo numfmt_format_currency($fmt, 1234567.891234567890000, "EUR")."\n";
7 echo numfmt_format_currency($fmt, 1234567.891234567890000, "RUR")."\n";
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.82.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new NumberFormatter( 'de_DE', NumberFormatter::CURRENCY );
3 echo $fmt->formatCurrency(1234567.891234567890000, "EUR")."\n";
4 echo $fmt->formatCurrency(1234567.891234567890000, "RUR")."\n";
5 $fmt = new NumberFormatter( 'ru_RU', NumberFormatter::CURRENCY );
6 echo $fmt->formatCurrency(1234567.891234567890000, "EUR")."\n";
7 echo $fmt->formatCurrency(1234567.891234567890000, "RUR")."\n";
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.83. numberformatter.formatcurrency.php

URL: <https://www.php.net/manual/en/numberformatter.getattribute.php>

4.1.83.1. Example 2

Code snippet:

```
1 <?php
2 $fmt = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
3 echo "Digits: ".$fmt->getAttribute(NumberFormatter::MAX_FRACTION_DIGITS)."\n";
4 echo $fmt->format(1234567.891234567890000)."\n";
5 $fmt->setAttribute(NumberFormatter::MAX_FRACTION_DIGITS, 2);
6 echo "Digits: ".$fmt->getAttribute(NumberFormatter::MAX_FRACTION_DIGITS)."\n";
7 echo $fmt->format(1234567.891234567890000)."\n";
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.84. numberformatter.geterrormessage.php

URL: <https://www.php.net/manual/en/numberformatter.geterrormessage.php>

4.1.84.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = numfmt_create( 'de_DE', NumberFormatter::DECIMAL );
3 $data = numfmt_format($fmt, 1234567.891234567890000);
4 var_dump(numfmt_get_error_message($fmt));
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.84.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
3 $fmt->format(1234567.891234567890000);
4 var_dump(numfmt_get_error_message($fmt));
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.85. numberformatter.getlocale.php

URL: <https://www.php.net/manual/en/numberformatter.getlocale.php>

Code snippet:

```
1 <?php
2 $req      = 'fr_FR_PARIS';
3 $fmt      = numfmt_create( $req, NumberFormatter::DECIMAL);
4 $res_val  = numfmt_get_locale( $fmt, Locale::VALID_LOCALE );
5 $res_act  = numfmt_get_locale( $fmt, Locale::ACTUAL_LOCALE );
6 printf( "Requested locale name: %s\nValid locale name: %s\nActual locale name:
7         %s\n",
8         $req, $res_val, $res_act );
```

Success

No security issues were identified in the code snippet

4.1.86. numberformatter.getpattern.php

URL: <https://www.php.net/manual/en/numberformatter.getpattern.php>

4.1.86.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = numfmt_create( 'de_DE', NumberFormatter::DECIMAL );
3 echo "Pattern: ".numfmt_get_pattern($fmt)."\n";
4 echo numfmt_format($fmt, 1234567.891234567890000)."\n";
5 numfmt_set_pattern($fmt, "#0.# kg");
6 echo "Pattern: ".numfmt_get_pattern($fmt)."\n";
7 echo numfmt_format($fmt, 1234567.891234567890000)."\n";
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.86.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
3 echo "Pattern: ".$fmt->getPattern()."\n";
4 echo $fmt->format(1234567.891234567890000)."\n";
5 $fmt->setPattern("#0.# kg");
6 echo "Pattern: ".$fmt->getPattern()."\n";
7 echo $fmt->format(1234567.891234567890000)."\n";
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.87. numberformatter.getsymbol.php

URL: <https://www.php.net/manual/en/numberformatter.getsymbol.php>

4.1.87.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = numfmt_create( 'de_DE', NumberFormatter::DECIMAL );
3 echo "Sep: ".numfmt_get_symbol($fmt,
  NumberFormatter::GROUPING_SEPARATOR_SYMBOL)."\n";
4 echo numfmt_format($fmt, 1234567.891234567890000)."\n";
5 numfmt_set_symbol($fmt, NumberFormatter::GROUPING_SEPARATOR_SYMBOL, "*");
6 echo "Sep: ".numfmt_get_symbol($fmt,
  NumberFormatter::GROUPING_SEPARATOR_SYMBOL)."\n";
7 echo numfmt_format($fmt, 1234567.891234567890000)."\n";
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.87.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
3 echo "Sep: ".$fmt->getSymbol(NumberFormatter::GROUPING_SEPARATOR_SYMBOL)."\n";
4 echo $fmt->format(1234567.891234567890000)."\n";
5 $fmt->setSymbol(NumberFormatter::GROUPING_SEPARATOR_SYMBOL, "*");
6 echo "Sep: ".$fmt->getSymbol(NumberFormatter::GROUPING_SEPARATOR_SYMBOL)."\n";
7 echo $fmt->format(1234567.891234567890000)."\n";
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.88. numberformatter.gettextattribute.php

URL: <https://www.php.net/manual/en/numberformatter.gettextattribute.php>

4.1.88.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = numfmt_create( 'de_DE', NumberFormatter::DECIMAL );
3 echo "Prefix: ".numfmt_get_text_attribute($fmt,
4     NumberFormatter::NEGATIVE_PREFIX)."\n";
5 echo numfmt_format($fmt, -1234567.891234567890000)."\n";
6 numfmt_set_text_attribute($fmt, NumberFormatter::NEGATIVE_PREFIX, "MINUS");
7 echo "Prefix: ".numfmt_get_text_attribute($fmt,
8     NumberFormatter::NEGATIVE_PREFIX)."\n";
9 echo numfmt_format($fmt, -1234567.891234567890000)."\n";
10 ?>
```

Success

No security issues were identified in the code snippet

4.1.88.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
3 echo "Prefix: ".$fmt->getTextAttribute(NumberFormatter::NEGATIVE_PREFIX)."\n";
4 echo $fmt->format(-1234567.891234567890000)."\n";
5 $fmt->setTextAttribute(NumberFormatter::NEGATIVE_PREFIX, "MINUS");
6 echo "Prefix: ".$fmt->getTextAttribute(NumberFormatter::NEGATIVE_PREFIX)."\n";
7 echo $fmt->format(-1234567.891234567890000)."\n";
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.89. numberformatter.parse.php

URL: <https://www.php.net/manual/en/numberformatter.parse.php>

4.1.89.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = numfmt_create( 'de_DE', NumberFormatter::DECIMAL );
3 $num = "1.234.567,891";
4 echo numfmt_parse($fmt, $num)."\n";
5 echo numfmt_parse($fmt, $num, NumberFormatter::TYPE_INT32)."\n";
6 ?>
```

Success

No security issues were identified in the code snippet

4.1.89.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
3 $num = "1.234.567,891";
4 echo $fmt->parse($num)."\n";
5 echo $fmt->parse($num, NumberFormatter::TYPE_INT32)."\n";
6 ?>
```

Success

No security issues were identified in the code snippet

4.1.90. numberformatter.parsecurrency.php

URL: <https://www.php.net/manual/en/numberformatter.parsecurrency.php>

4.1.90.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = numfmt_create( 'de_DE', NumberFormatter::CURRENCY );
3 $num = "1.234.567,89\xc2\xa0$";
4 echo "We have ".numfmt_parse_currency($fmt, $num, $curr)." in $curr\n";
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.90.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new NumberFormatter( 'de_DE', NumberFormatter::CURRENCY );
3 $num = "1.234.567,89\xc2\xa0$";
4 echo "We have ".$fmt->parseCurrency($num, $curr)." in $curr\n";
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.91. numberformatter.setattribute.php

URL: <https://www.php.net/manual/en/numberformatter.setattribute.php>

4.1.91.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = numfmt_create( 'de_DE', NumberFormatter::DECIMAL );
3 echo "Digits: ".numfmt_get_attribute($fmt,
4   NumberFormatter::MAX_FRACTION_DIGITS)."\n";
5 echo numfmt_format($fmt, 1234567.891234567890000)."\n";
6 numfmt_set_attribute($fmt, NumberFormatter::MAX_FRACTION_DIGITS, 2);
7 echo "Digits: ".numfmt_get_attribute($fmt,
8   NumberFormatter::MAX_FRACTION_DIGITS)."\n";
9 echo numfmt_format($fmt, 1234567.891234567890000)."\n";
10 ?>
```

Success

No security issues were identified in the code snippet

4.1.91.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
3 echo "Digits: ".$fmt->getAttribute(NumberFormatter::MAX_FRACTION_DIGITS)."\n";
4 echo $fmt->format(1234567.891234567890000)."\n";
5 $fmt->setAttribute(NumberFormatter::MAX_FRACTION_DIGITS, 2);
6 echo "Digits: ".$fmt->getAttribute(NumberFormatter::MAX_FRACTION_DIGITS)."\n";
7 echo $fmt->format(1234567.891234567890000)."\n";
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.92. numberformatter.setpattern.php

URL: <https://www.php.net/manual/en/numberformatter.setpattern.php>

4.1.92.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = numfmt_create( 'de_DE', NumberFormatter::DECIMAL );
3 echo "Pattern: ".numfmt_get_pattern($fmt)."\n";
4 echo numfmt_format($fmt, 1234567.891234567890000)."\n";
5 numfmt_set_pattern($fmt, "#0.# kg");
6 echo "Pattern: ".numfmt_get_pattern($fmt)."\n";
7 echo numfmt_format($fmt, 1234567.891234567890000)."\n";
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.92.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
3 echo "Pattern: ".$fmt->getPattern()."\n";
4 echo $fmt->format(1234567.891234567890000)."\n";
5 $fmt->setPattern("#0.# kg");
6 echo "Pattern: ".$fmt->getPattern()."\n";
7 echo $fmt->format(1234567.891234567890000)."\n";
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.93. numberformatter.setsymbol.php

URL: <https://www.php.net/manual/en/numberformatter.setsymbol.php>

4.1.93.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = numfmt_create( 'de_DE', NumberFormatter::DECIMAL );
3 echo "Sep: ".numfmt_get_symbol($fmt,
  NumberFormatter::GROUPING_SEPARATOR_SYMBOL)."\n";
4 echo numfmt_format($fmt, 1234567.891234567890000)."\n";
5 numfmt_set_symbol($fmt, NumberFormatter::GROUPING_SEPARATOR_SYMBOL, "*");
6 echo "Sep: ".numfmt_get_symbol($fmt,
  NumberFormatter::GROUPING_SEPARATOR_SYMBOL)."\n";
7 echo numfmt_format($fmt, 1234567.891234567890000)."\n";
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.93.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
3 echo "Sep: ".$fmt->getSymbol(NumberFormatter::GROUPING_SEPARATOR_SYMBOL)."\n";
4 echo $fmt->format(1234567.891234567890000)."\n";
5 $fmt->setSymbol(NumberFormatter::GROUPING_SEPARATOR_SYMBOL, "*");
6 echo "Sep: ".$fmt->getSymbol(NumberFormatter::GROUPING_SEPARATOR_SYMBOL)."\n";
7 echo $fmt->format(1234567.891234567890000)."\n";
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.94. numberformatter.settextattribute.php

URL: <https://www.php.net/manual/en/numberformatter.settextattribute.php>

4.1.94.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = numfmt_create( 'de_DE', NumberFormatter::DECIMAL );
3 echo "Prefix: ".numfmt_get_text_attribute($fmt,
4     NumberFormatter::NEGATIVE_PREFIX)."\n";
5 echo numfmt_format($fmt, -1234567.891234567890000)."\n";
6 numfmt_set_text_attribute($fmt, NumberFormatter::NEGATIVE_PREFIX, "MINUS");
7 echo "Prefix: ".numfmt_get_text_attribute($fmt,
8     NumberFormatter::NEGATIVE_PREFIX)."\n";
9 echo numfmt_format($fmt, -1234567.891234567890000)."\n";
10 ?>
```

Success

No security issues were identified in the code snippet

4.1.94.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
3 echo "Prefix: ".$fmt->getTextAttribute(NumberFormatter::NEGATIVE_PREFIX)."\n";
4 echo $fmt->format(-1234567.891234567890000)."\n";
5 $fmt->setTextAttribute(NumberFormatter::NEGATIVE_PREFIX, "MINUS");
6 echo "Prefix: ".$fmt->getTextAttribute(NumberFormatter::NEGATIVE_PREFIX)."\n";
7 echo $fmt->format(-1234567.891234567890000)."\n";
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.95. numberformatter.create.php

URL: <https://www.php.net/manual/en/numberformatter.create.php>

4.1.95.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = numfmt_create( 'de_DE', NumberFormatter::DECIMAL );
3 echo numfmt_format($fmt, 1234567.891234567890000)."\n";
4 $fmt = numfmt_create( 'it', NumberFormatter::SPELLOUT );
5 echo numfmt_format($fmt, 1142)."\n";
6 ?>
```

Success

No security issues were identified in the code snippet

4.1.95.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
3 echo $fmt->format(1234567.891234567890000)."\n";
4 $fmt = new NumberFormatter( 'it', NumberFormatter::SPELLOUT );
5 echo $fmt->format(1142)."\n";
6 ?>
```

Success

No security issues were identified in the code snippet

4.1.96. locale.acceptfromhttp.php

URL: <https://www.php.net/manual/en/locale.acceptfromhttp.php>

4.1.96.1. Example 1

Code snippet:

```
1 <?php
2 $locale = locale_accept_from_http($_SERVER['HTTP_ACCEPT_LANGUAGE']);
3 echo $locale;
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.96.2. Example 2

Code snippet:

```
1 <?php
2 $locale = Locale::acceptFromHttp($_SERVER['HTTP_ACCEPT_LANGUAGE']);
3 echo $locale;
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.97. locale.composelocale.php

URL: <https://www.php.net/manual/en/locale.composelocale.php>

4.1.97.1. Example 1

Code snippet:

```
1 <?php
2 $arr = array(
3     'language'=>'en',
4     'script' =>'Hans',
5     'region' =>'CN',
6     'variant2'=>'rozaj',
7     'variant1'=>'nedis',
8     'private1'=>'prv1',
9     'private2'=>'prv2',
10 );
11 echo locale_compose($arr);
12 ?>
```

Success

No security issues were identified in the code snippet

4.1.97.2. Example 2

Code snippet:

```
1 <?php
2 $arr = array(
3     'language'=>'en' ,
4     'script' =>'Hans',
5     'region' =>'CN',
6     'variant2'=>'rozaj',
7     'variant1'=>'nedis',
8     'private1'=>'prv1',
9     'private2'=>'prv2',
10 );
11 echo Locale::composeLocale($arr);
12 ?>
```

Success

No security issues were identified in the code snippet

4.1.98. locale.filtermatches.php

URL: <https://www.php.net/manual/en/locale.filtermatches.php>

4.1.98.1. Example 1

Code snippet:

```
1 <?php
2 echo (locale_filter_matches('de-DEVA','de-DE', false)) ? "Matches" : "Does not
   match";
3 echo ' ';
4 echo (locale_filter_matches('de-DE_1996','de-DE', false)) ? "Matches" : "Does not
   match";
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.98.2. Example 2

Code snippet:

```
1 <?php
2 echo (Locale::filterMatches('de-DEVA','de-DE', false)) ? "Matches" : "Does not
   match";
3 echo ' ';
4 echo (Locale::filterMatches('de-DE-1996','de-DE', false)) ? "Matches" : "Does not
   match";
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.99. locale.getallvariants.php

URL: <https://www.php.net/manual/en/locale.getallvariants.php>

4.1.99.1. Example 1

Code snippet:

```
1 <?php
2 $arr = locale_get_all_variants('sl_IT_NEDIS_ROJAZ_1901');
3 var_export( $arr );
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.99.2. Example 2

Code snippet:

```
1 <?php
2 $arr = Locale::getAllVariants('sl_IT_NEDIS_ROJAZ_1901');
3 var_export( $arr );
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.100. locale.getdefault.php

URL: <https://www.php.net/manual/en/locale.getdefault.php>

4.1.100.1. Example 1

Code snippet:

```
1 <?php
2 ini_set('intl.default_locale', 'de-DE');
3 echo locale_get_default();
4 echo ' ';
5 locale_set_default('fr');
6 echo locale_get_default();
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.100.2. Example 2

Code snippet:

```
1 <?php
2 ini_set('intl.default_locale', 'de-DE');
3 echo Locale::getDefault();
4 echo ' ';
5 Locale::setDefault('fr');
6 echo Locale::getDefault();
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.101. locale.getdisplaylanguage.php

URL: <https://www.php.net/manual/en/locale.getdisplaylanguage.php>

4.1.101.1. Example 1

Code snippet:

```
1 <?php
2 echo locale_get_display_language('sl-Latn-IT-nedis', 'en');
3 echo ";\n";
4 echo locale_get_display_language('sl-Latn-IT-nedis', 'fr');
5 echo ";\n";
6 echo locale_get_display_language('sl-Latn-IT-nedis', 'de');
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.101.2. Example 2

Code snippet:

```
1 <?php
2 echo Locale::getDisplayLanguage('sl-Latn-IT-nedis', 'en');
3 echo ";\n";
4 echo Locale::getDisplayLanguage('sl-Latn-IT-nedis', 'fr');
5 echo ";\n";
6 echo Locale::getDisplayLanguage('sl-Latn-IT-nedis', 'de');
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.102. locale.getdisplayname.php

URL: <https://www.php.net/manual/en/locale.getdisplayname.php>

4.1.102.1. Example 1

Code snippet:

```
1 <?php
2 echo locale_get_display_name('sl-Latn-IT-nedis', 'en');
3 echo ";\n";
4 echo locale_get_display_name('sl-Latn-IT-nedis', 'fr');
5 echo ";\n";
6 echo locale_get_display_name('sl-Latn-IT-nedis', 'de');
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.102.2. Example 2

Code snippet:

```
1 <?php
2 echo Locale::getDisplayname('sl-Latn-IT-nedis', 'en');
3 echo ";\n";
4 echo Locale::getDisplayname('sl-Latn-IT-nedis', 'fr');
5 echo ";\n";
6 echo Locale::getDisplayname('sl-Latn-IT-nedis', 'de');
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.103. locale.getdisplayregion.php

URL: <https://www.php.net/manual/en/locale.getdisplayregion.php>

4.1.103.1. Example 1

Code snippet:

```
1 <?php
2 echo locale_get_display_region('sl-Latn-IT-nedis', 'en');
3 echo ";\n";
4 echo locale_get_display_region('sl-Latn-IT-nedis', 'fr');
5 echo ";\n";
6 echo locale_get_display_region('sl-Latn-IT-nedis', 'de');
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.103.2. Example 2

Code snippet:

```
1 <?php
2 echo Locale::getDisplayRegion('sl-Latn-IT-nedis', 'en');
3 echo ";\n";
4 echo Locale::getDisplayRegion('sl-Latn-IT-nedis', 'fr');
5 echo ";\n";
6 echo Locale::getDisplayRegion('sl-Latn-IT-nedis', 'de');
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.104. locale.getdisplayscript.php

URL: <https://www.php.net/manual/en/locale.getdisplayscript.php>

4.1.104.1. Example 1

Code snippet:

```
1 <?php
2 echo locale_get_display_script('sl-Latn-IT-nedis', 'en');
3 echo ";\n";
4 echo locale_get_display_script('sl-Latn-IT-nedis', 'fr');
5 echo ";\n";
6 echo locale_get_display_script('sl-Latn-IT-nedis', 'de');
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.104.2. Example 2

Code snippet:

```
1 <?php
2 echo Locale::getDisplayScript('sl-Latn-IT-nedis', 'en');
3 echo ";\n";
4 echo Locale::getDisplayScript('sl-Latn-IT-nedis', 'fr');
5 echo ";\n";
6 echo Locale::getDisplayScript('sl-Latn-IT-nedis', 'de');
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.105. locale.getdisplayvariant.php

URL: <https://www.php.net/manual/en/locale.getdisplayvariant.php>

4.1.105.1. Example 1

Code snippet:

```
1 <?php
2 echo locale_get_display_variant('sl-Latn-IT-nedis', 'en');
3 echo ";\n";
4 echo locale_get_display_variant('sl-Latn-IT-nedis', 'fr');
5 echo ";\n";
6 echo locale_get_display_variant('sl-Latn-IT-nedis', 'de');
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.105.2. Example 2

Code snippet:

```
1 <?php
2 echo Locale::getDisplayVariant('sl-Latn-IT-nedis', 'en');
3 echo ";\n";
4 echo Locale::getDisplayVariant('sl-Latn-IT-nedis', 'fr');
5 echo ";\n";
6 echo Locale::getDisplayVariant('sl-Latn-IT-nedis', 'de');
7 ?>
```

Success

No security issues were identified in the code snippet

4.1.106. locale.getkeywords.php

URL: <https://www.php.net/manual/en/locale.getkeywords.php>

4.1.106.1. Example 1

Code snippet:

```
1 <?php
2 $keywords_arr = locale_get_keywords('de_DE@currency=EUR;collation=PHONEBOOK');
3 if ($keywords_arr) {
4     foreach ($keywords_arr as $key => $value) {
5         echo "$key = $value\n";
6     }
7 }
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.106.2. Example 2

Code snippet:

```
1 <?php
2 $keywords_arr = Locale::getKeywords('de_DE@currency=EUR;collation=PHONEBOOK');
3 if ($keywords_arr) {
4     foreach ($keywords_arr as $key => $value) {
5         echo "$key = $value\n";
6     }
7 }
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.107. locale.getprimarylanguage.php

URL: <https://www.php.net/manual/en/locale.getprimarylanguage.php>

4.1.107.1. Example 1

Code snippet:

```
1 <?php
2 echo locale_get_primary_language('zh-Hant');
3 ?>
```

Success

No security issues were identified in the code snippet

4.1.107.2. Example 2

Code snippet:

```
1 <?php
2 echo Locale::getPrimaryLanguage('zh-Hant');
3 ?>
```

Success

No security issues were identified in the code snippet

4.1.108. locale.getregion.php

URL: <https://www.php.net/manual/en/locale.getregion.php>

4.1.108.1. Example 1

Code snippet:

```
1 <?php
2 echo locale_get_region('de-CH-1901');
3 ?>
```

Success

No security issues were identified in the code snippet

4.1.108.2. Example 2

Code snippet:

```
1 <?php
2 echo Locale::getRegion('de-CH-1901');
3 ?>
```

Success

No security issues were identified in the code snippet

4.1.109. locale.getscript.php

URL: <https://www.php.net/manual/en/locale.getscript.php>

4.1.109.1. Example 1

Code snippet:

```
1 <?php
2 echo locale_get_script('sr-Cyrl');
3 ?>
```

PHP

Success

No security issues were identified in the code snippet

4.1.109.2. Example 2

Code snippet:

```
1 <?php
2 echo Locale::getScript('sr-Cyrl');
3 ?>
```

PHP

Success

No security issues were identified in the code snippet

4.1.110. locale.parseLocale.php

URL: <https://www.php.net/manual/en/locale.parseLocale.php>

4.1.110.1. Example 1

Code snippet:

```
1 <?php
2 $arr = locale_parse('sl-Latn-IT-nedis');
3 if ($arr) {
4     foreach ($arr as $key => $value) {
5         echo "$key : $value , ";
6     }
7 }
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.110.2. Example 2

Code snippet:

```
1 <?php
2 $arr = Locale::parseLocale('sl-Latn-IT-nedis');
3 if ($arr) {
4     foreach ($arr as $key => $value) {
5         echo "$key : $value , ";
6     }
7 }
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.111. locale.setdefault.php

URL: <https://www.php.net/manual/en/locale.setdefault.php>

4.1.111.1. Example 1

Code snippet:

```
1 <?php
2 locale_set_default('de-DE');
3 echo locale_get_default();
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.111.2. Example 2

Code snippet:

```
1 <?php
2 Locale::setDefault('de-DE');
3 echo Locale::getDefault();
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.112. normalizer.getrawdecomposition.php

URL: <https://www.php.net/manual/en/normalizer.getrawdecomposition.php>

Code snippet:

```
1 <?php
2
3 $result = "";
4 $strings = [
5     "a",
6     "\u{FFDA}",
7     "\u{FDFA}",
8     "",
9     "aa",
10    "\xF5",
11 ];
12
13 foreach ($strings as $string) {
14     $decomposition = Normalizer::getRawDecomposition($string);
15     // $decomposition = normalizer_get_raw_decomposition($string); Procedural
16     // way
17     $error_code = intl_get_error_code();
18     $error_message = intl_get_error_message();
19
20     $string_hex = bin2hex($string);
21     $result .= "-----\n";
22
23     if ($decomposition === null) {
24         $result .= "'$string_hex' has no decomposition mapping\n" ;
25     } else {
26         $result .= "'$string_hex' has the decomposition mapping '" .
27             bin2hex($decomposition) . "'\n" ;
28     }
29     $result .= "error info: '$error_message' ($error_code)\n";
30 }
31
32 echo $result;
33 ?>
```

Success

No security issues were identified in the code snippet

4.1.113. normalizer.isnormalized.php

URL: <https://www.php.net/manual/en/normalizer.isnormalized.php>

4.1.113.1. Example 1

Code snippet:

```
1 <?php
2 $char_A_ring = "\xC3\x85"; // 'LATIN CAPITAL LETTER A WITH RING ABOVE' (U+00C5)
3 $char_combining_ring_above = "\xCC\x8A"; // 'COMBINING RING ABOVE' (U+030A)
4
5 $char_orig = 'A' . $char_combining_ring_above;
6 $char_norm = normalizer_normalize( 'A' . $char_combining_ring_above,
  Normalizer::FORM_C );
7
8 echo ( normalizer_is_normalized($char_orig, Normalizer::FORM_C) ) ?
  "normalized" : "not normalized";
9 echo ' ';
10 echo ( normalizer_is_normalized($char_norm, Normalizer::FORM_C) ) ?
  "normalized" : "not normalized";
11 ?>
```

Success

No security issues were identified in the code snippet

4.1.113.2. Example 2

Code snippet:

```
1 <?php
2 $char_A_ring = "\xC3\x85"; // 'LATIN CAPITAL LETTER A WITH RING ABOVE' (U+00C5)
3 $char_combining_ring_above = "\xCC\x8A"; // 'COMBINING RING ABOVE' (U+030A)
4
5 $char_orig = 'A' . $char_combining_ring_above;
6 $char_norm = Normalizer::normalize( 'A' . $char_combining_ring_above,
  Normalizer::FORM_C );
7
8 echo ( Normalizer::isNormalized($char_orig, Normalizer::FORM_C) ) ?
  "normalized" : "not normalized";
9 echo ' ';
10 echo ( Normalizer::isNormalized($char_norm, Normalizer::FORM_C) ) ?
  "normalized" : "not normalized";
11 ?>
```

Success

No security issues were identified in the code snippet

4.1.114. normalizer.normalize.php

URL: <https://www.php.net/manual/en/normalizer.normalize.php>

4.1.114.1. Example 1

Code snippet:

```
1 <?php
2 $char_A_ring = "\xC3\x85"; // 'LATIN CAPITAL LETTER A WITH RING ABOVE' (U+00C5)
3 $char_combining_ring_above = "\xCC\x8A"; // 'COMBINING RING ABOVE' (U+030A)
4
5 $char_1 = normalizer_normalize( $char_A_ring, Normalizer::FORM_C );
6 $char_2 = normalizer_normalize( 'A' . $char_combining_ring_above,
  Normalizer::FORM_C );
7
8 echo urlencode($char_1);
9 echo ' ';
10 echo urlencode($char_2);
11 ?>
```

Success

No security issues were identified in the code snippet

4.1.114.2. Example 2

Code snippet:

```
1 <?php
2 $char_A_ring = "\xC3\x85"; // 'LATIN CAPITAL LETTER A WITH RING ABOVE' (U+00C5)
3 $char_combining_ring_above = "\xCC\x8A"; // 'COMBINING RING ABOVE' (U+030A)
4
5 $char_1 = Normalizer::normalize( $char_A_ring, Normalizer::FORM_C );
6 $char_2 = Normalizer::normalize( 'A' . $char_combining_ring_above,
  Normalizer::FORM_C );
7
8 echo urlencode($char_1);
9 echo ' ';
10 echo urlencode($char_2);
11 ?>
```

Success

No security issues were identified in the code snippet

4.1.115. messageformatter.create.php

URL: <https://www.php.net/manual/en/messageformatter.create.php>

4.1.115.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = msgfmt_create("en_US", "{0,number,integer} monkeys on {1,number,integer}
   trees make {2,number} monkeys per tree");
3 echo msgfmt_format($fmt, array(4560, 123, 4560/123));
4 $fmt = msgfmt_create("de", "{0,number,integer} Affen auf {1,number,integer}
   Bäumen sind {2,number} Affen pro Baum");
5 echo msgfmt_format($fmt, array(4560, 123, 4560/123));
6 ?>
```

Success

No security issues were identified in the code snippet

4.1.115.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new MessageFormatter("en_US", "{0,number,integer} monkeys on
   {1,number,integer} trees make {2,number} monkeys per tree");
3 echo $fmt->format(array(4560, 123, 4560/123));
4 $fmt = new MessageFormatter("de", "{0,number,integer} Affen auf
   {1,number,integer} Bäumen sind {2,number} Affen pro Baum");
5 echo $fmt->format(array(4560, 123, 4560/123));
6 ?>
```

Success

No security issues were identified in the code snippet

4.1.116. messageformatter.format.php

URL: <https://www.php.net/manual/en/messageformatter.format.php>

4.1.116.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = msgfmt_create("en_US", "{0,number,integer} monkeys on {1,number,integer}
   trees make {2,number} monkeys per tree");
3 echo msgfmt_format($fmt, array(4560, 123, 4560/123));
4 $fmt = msgfmt_create("de", "{0,number,integer} Affen auf {1,number,integer}
   Bäumen sind {2,number} Affen pro Baum");
5 echo msgfmt_format($fmt, array(4560, 123, 4560/123));
6 ?>
```

Success

No security issues were identified in the code snippet

4.1.116.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new MessageFormatter("en_US", "{0,number,integer} monkeys on
   {1,number,integer} trees make {2,number} monkeys per tree");
3 echo $fmt->format(array(4560, 123, 4560/123));
4 $fmt = new MessageFormatter("de", "{0,number,integer} Affen auf
   {1,number,integer} Bäumen sind {2,number} Affen pro Baum");
5 echo $fmt->format(array(4560, 123, 4560/123));
6 ?>
```

Success

No security issues were identified in the code snippet

4.1.117. messageformatter.getlocale.php

URL: <https://www.php.net/manual/en/messageformatter.getlocale.php>

4.1.117.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = msgfmt_create('en_US', "Number {0,number}");
3 echo msgfmt_get_locale($fmt);
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.117.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new MessageFormatter('en_US', "Number {0,number}");
3 echo $fmt->getLocale();
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.118. messageformatter.getpattern.php

URL: <https://www.php.net/manual/en/messageformatter.getpattern.php>

4.1.118.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = msgfmt_create( "en_US", "{0, number} monkeys on {1, number} trees" );
3 echo "Default pattern: '" . msgfmt_get_pattern( $fmt ) . "'\n";
4 echo "Formatting result: " . msgfmt_format( $fmt, array(123, 456) ) . "\n";
5
6 msgfmt_set_pattern( $fmt, "{0, number} trees hosting {1, number} monkeys" );
7 echo "New pattern: '" . msgfmt_get_pattern( $fmt ) . "'\n";
8 echo "Formatted number: " . msgfmt_format( $fmt, array(123, 456) ) . "\n";
9 ?>
```

Success

No security issues were identified in the code snippet

4.1.118.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new MessageFormatter( "en_US", "{0, number} monkeys on {1, number}
3 trees" );
4 echo "Default pattern: '" . $fmt->getPattern() . "'\n";
5 echo "Formatting result: " . $fmt->format(array(123, 456)) . "\n";
6
7 $fmt->setPattern("{0, number} trees hosting {1, number} monkeys" );
8 echo "New pattern: '" . $fmt->getPattern() . "'\n";
9 echo "Formatted number: " . $fmt->format(array(123, 456)) . "\n";
10 ?>
```

Success

No security issues were identified in the code snippet

4.1.119. messageformatter.parse.php

URL: <https://www.php.net/manual/en/messageformatter.parse.php>

4.1.119.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = msgfmt_create('en_US', "{0,number,integer} monkeys on {1,number,integer}
   trees make {2,number} monkeys per tree");
3 $res = msgfmt_parse($fmt, "4,560 monkeys on 123 trees make 37.073 monkeys per
   tree");
4 var_export($res);
5
6 $fmt = msgfmt_create('de', "{0,number,integer} Affen auf {1,number,integer}
   Bäumen sind {2,number} Affen pro Baum");
7 $res = msgfmt_parse($fmt, "4.560 Affen auf 123 Bäumen sind 37,073 Affen pro
   Baum");
8 var_export($res);
9 ?>
```

Success

No security issues were identified in the code snippet

4.1.119.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new MessageFormatter('en_US', "{0,number,integer} monkeys on
   {1,number,integer} trees make {2,number} monkeys per tree");
3 $res = $fmt->parse("4,560 monkeys on 123 trees make 37.073 monkeys per tree");
4 var_export($res);
5
6 $fmt = new MessageFormatter('de', "{0,number,integer} Affen auf
   {1,number,integer} Bäumen sind {2,number} Affen pro Baum");
7 $res = $fmt->parse("4.560 Affen auf 123 Bäumen sind 37,073 Affen pro Baum");
8 var_export($res);
9 ?>
```

Success

No security issues were identified in the code snippet

4.1.120. messageformatter.parsemessage.php

URL: <https://www.php.net/manual/en/messageformatter.parsemessage.php>

4.1.120.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = msgfmt_parse_message('en_US', "{0,number,integer} monkeys on
3     {1,number,integer} trees make {2,number} monkeys per tree",
4     "4,560 monkeys on 123 trees make 37.073 monkeys per
5     tree");
6 var_export($fmt);
7
8 $fmt = msgfmt_parse_message('de', "{0,number,integer} Affen auf
9     {1,number,integer} Bäumen sind {2,number} Affen pro Baum",
10    "4.560 Affen auf 123 Bäumen sind 37,073 Affen pro
11    Baum");
12 var_export($fmt);
13 ?>
```

Success

No security issues were identified in the code snippet

4.1.120.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = MessageFormatter::parseMessage('en_US', "{0,number,integer} monkeys on
3     {1,number,integer} trees make {2,number} monkeys per tree",
4     "4,560 monkeys on 123 trees make 37.073 monkeys per
5     tree");
6 var_export($fmt);
7
8 $fmt = MessageFormatter::parseMessage('de', "{0,number,integer} Affen auf
9     {1,number,integer} Bäumen sind {2,number} Affen pro Baum",
10    "4.560 Affen auf 123 Bäumen sind 37,073 Affen pro
11    Baum");
12 var_export($fmt);
13 ?>
```

Success

No security issues were identified in the code snippet

4.1.121. messageformatter.setpattern.php

URL: <https://www.php.net/manual/en/messageformatter.setpattern.php>

4.1.121.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = msgfmt_create( "en_US", "{0, number} monkeys on {1, number} trees" );
3 echo "Default pattern: '" . msgfmt_get_pattern( $fmt ) . "'\n";
4 echo "Formatting result: " . msgfmt_format( $fmt, array(123, 456) ) . "\n";
5
6 msgfmt_set_pattern( $fmt, "{0, number} trees hosting {1, number} monkeys" );
7 echo "New pattern: '" . msgfmt_get_pattern( $fmt ) . "'\n";
8 echo "Formatted number: " . msgfmt_format( $fmt, array(123, 456) ) . "\n";
9 ?>
```

Success

No security issues were identified in the code snippet

4.1.121.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new MessageFormatter( "en_US", "{0, number} monkeys on {1, number}
3 trees" );
4 echo "Default pattern: '" . $fmt->getPattern() . "'\n";
5 echo "Formatting result: " . $fmt->format(array(123, 456)) . "\n";
6
7 $fmt->setPattern("{0, number} trees hosting {1, number} monkeys" );
8 echo "New pattern: '" . $fmt->getPattern() . "'\n";
9 echo "Formatted number: " . $fmt->format(array(123, 456)) . "\n";
10 ?>
```

Success

No security issues were identified in the code snippet

4.1.122. intlgregorianCalendar.createFromDate.php

URL: <https://www.php.net/manual/en/intlgregorianCalendar.createFromDate.php>

Code snippet:

```
1 <?php
2
3 $intlCalendar = IntlGregorianCalendar::createFromDate(2023, 11, 23);
4 var_dump($intlCalendar);
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.123. intlgregorianCalendar.createFromDate.php

URL: <https://www.php.net/manual/en/intlgregorianCalendar.createFromDate.php>

Code snippet:

```
1 <?php
2
3 $intlCalendar = IntlGregorianCalendar::createFromDate(2023, 11, 23, 12, 00);
4 var_dump($intlCalendar);
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.124. intlcalendar.add.php

URL: <https://www.php.net/manual/en/intlcalendar.add.php>

Code snippet:

```
1 <?php
2 ini_set('intl.default_locale', 'fr_FR');
3 ini_set('date.timezone', 'UTC');
4
5 $cal = new IntlGregorianCalendar(2012, 0 /* January */, 31);
6 echo IntlDateFormatter::formatObject($cal), "\n";
7
8 $cal->add(IntlCalendar::FIELD_MONTH, 1);
9 echo IntlDateFormatter::formatObject($cal), "\n";
10
11 $cal->add(IntlCalendar::FIELD_DAY_OF_MONTH, 1);
12 echo IntlDateFormatter::formatObject($cal), "\n";
```

Success

No security issues were identified in the code snippet

4.1.125. intlcalendar.after.php

URL: <https://www.php.net/manual/en/intlcalendar.after.php>

Code snippet:

```
1 <?php
2 $cal1 = IntlCalendar::createInstance();
3 $cal2 = clone $cal1;
4
5 var_dump($cal1->after($cal2), //false
6          $cal2->after($cal1)); //false
7
8 $cal1->roll(IntlCalendar::FIELD_MILLISECOND, true);
9
10 var_dump($cal1->after($cal2), //true
11          $cal2->after($cal1)); //false
```

Success

No security issues were identified in the code snippet

4.1.126. intlcalendar.clear.php

URL: <https://www.php.net/manual/en/intlcalendar.clear.php>

Code snippet:

```
1 <?php
2 ini_set('intl.default_locale', 'es_ES');
3 ini_set('date.timezone', 'UTC');
4
5 $fields = array(
6     'FIELD_ERA' => 0,
7     'FIELD_YEAR' => 1,
8     'FIELD_MONTH' => 2,
9     'FIELD_WEEK_OF_YEAR' => 3,
10    'FIELD_WEEK_OF_MONTH' => 4,
11    'FIELD_DATE' => 5,
12    'FIELD_DAY_OF_YEAR' => 6,
13    'FIELD_DAY_OF_WEEK' => 7,
14    'FIELD_DAY_OF_WEEK_IN_MONTH' => 8,
15    'FIELD_AM_PM' => 9,
16    'FIELD_HOUR' => 10,
17    'FIELD_HOUR_OF_DAY' => 11,
18    'FIELD_MINUTE' => 12,
19    'FIELD_SECOND' => 13,
20    'FIELD_MILLISECOND' => 14,
21    'FIELD_ZONE_OFFSET' => 15,
22    'FIELD_DST_OFFSET' => 16,
23    'FIELD_YEAR_WOY' => 17,
24    'FIELD_DOW_LOCAL' => 18,
25    'FIELD_EXTENDED_YEAR' => 19,
26    'FIELD_JULIAN_DAY' => 20,
27    'FIELD_MILLISECONDS_IN_DAY' => 21,
28    'FIELD_IS_LEAP_MONTH' => 22,
29    'FIELD_FIELD_COUNT' => 23,
30 );
31 function getSetFields(IntlCalendar $cal) {
32     global $fields;
33     $ret = array();
34     foreach ($fields as $name => $value) {
35         if ($cal->isSet($value)) {
36             $ret[] = $name;
37         }
38     }
```

```

39     return $ret;
40 }
41
42 $cal = new IntlGregorianCalendar(2013, 2 /* March */, 15);
43 echo "After GregorianCalendar creation\n";
44 print_r(getSetFields($cal));
45 echo "\n";
46
47 echo IntlDateFormatter::formatObject($cal), "\n";
48 echo "After the formatter requested the extended year\n";
49 print_r(getSetFields($cal));
50 echo "\n";
51
52 $cal->clear(IntlCalendar::FIELD_YEAR);
53 echo "After the year has been cleared, the date stays the same\n";
54 echo IntlDateFormatter::formatObject($cal), "\n";
55 echo "because FIELD_EXTENDED_YEAR is still set\n";
56 print_r(getSetFields($cal));
57 echo "\n";
58
59 var_dump($cal->clear(IntlCalendar::FIELD_EXTENDED_YEAR));
60 echo "After the extended year has been cleared\n";
61 print_r(getSetFields($cal));
62 echo IntlDateFormatter::formatObject($cal), "\n";
63 echo "\n";
64
65 echo "After the fields are recalculated,\n"
66     . " " extended year is set again (to 1970)\n";
67 print_r(getSetFields($cal));
68 echo "\n";
69
70 $cal->clear();
71 echo "After calling variant with no arguments\n";
72 print_r(getSetFields($cal));
73 echo IntlDateFormatter::formatObject($cal), "\n";

```

Success

No security issues were identified in the code snippet

4.1.127. spoofchecker.areconfusable.php

URL: <https://www.php.net/manual/en/spoofchecker.areconfusable.php>

Code snippet:

```
1 <?php
2 $checker = new Spoofchecker();
3
4 $checker->areConfusable('google.com', 'google.com'); // true
5 // Lower l can be confused with digit one
6
7 $checker->areConfusable('google.com', 'g00gle.com'); // false
8 // Zero (0) cannot be easily confused with "o" letter
```

Success

No security issues were identified in the code snippet

4.1.128. spoofchecker.issuspicious.php

URL: <https://www.php.net/manual/en/spoofchecker.issuspicious.php>

Code snippet:

```
1 <?php
2 $checker = new Spoofchecker();
3
4 $checker->isSuspicious('google.com'); // FALSE: only ASCII characters
5
6 $checker->isSuspicious('Paypal.com'); // TRUE
7 // The first letter is from Cirylic, not a regular latin "P"
```

Success

No security issues were identified in the code snippet

4.1.129. function.bindtextdomain.php

URL: <https://www.php.net/manual/en/function.bindtextdomain.php>

Code snippet:

```
1 <?php
2 $domain = 'myapp';
3 echo bindtextdomain($domain, '/usr/share/myapp/locale');
4
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.130. function.iconv.php

URL: <https://www.php.net/manual/en/function.iconv.php>

Code snippet:

```
1 <?php
2 $text = "This is the Euro symbol '€'.";
3
4 echo 'Original : ', $text, PHP_EOL;
5 echo 'TRANSLIT : ', iconv("UTF-8", "ISO-8859-1//TRANSLIT", $text), PHP_EOL;
6 echo 'IGNORE    : ', iconv("UTF-8", "ISO-8859-1//IGNORE", $text), PHP_EOL;
7 echo 'Plain     : ', iconv("UTF-8", "ISO-8859-1", $text), PHP_EOL;
8
9 ?>
```

Success

No security issues were identified in the code snippet

4.1.131. function.iconv-mime-decode.php

URL: <https://www.php.net/manual/en/function.iconv-mime-decode.php>

Code snippet:

```
1 <?php
2 // This yields "Subject: Prüfung Prüfung"
3 echo iconv_mime_decode("Subject: =?UTF-8?B?UHLdVdGZ1bmcgUHLdVdGZ1bmc=?=",
4                          0, "ISO-8859-1");
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.132. function.ob-conv-handler.php

URL: <https://www.php.net/manual/en/function.ob-conv-handler.php>

Code snippet:

```
1 <?php
2 iconv_set_encoding("internal_encoding", "UTF-8");
3 iconv_set_encoding("output_encoding", "ISO-8859-1");
4 ob_start("ob_conv_handler"); // start output buffering
5 ?>
```

Success

No security issues were identified in the code snippet

4.1.133. function.iconv-set-encoding.php

URL: <https://www.php.net/manual/en/function.iconv-set-encoding.php>

Code snippet:

```
1 <?php
2 iconv_set_encoding("internal_encoding", "UTF-8");
3 iconv_set_encoding("output_encoding", "ISO-8859-1");
4 ?>
```

Success

No security issues were identified in the code snippet

4.1.134. function.iconv-mime-encode.php

URL: <https://www.php.net/manual/en/function.iconv-mime-encode.php>

Code snippet:

```
1 <?php
2 $preferences = array(
3     "input-charset" => "ISO-8859-1",
4     "output-charset" => "UTF-8",
5     "line-length" => 76,
6     "line-break-chars" => "\n"
7 );
8 $preferences["scheme"] = "Q";
9 // This yields "Subject: =?UTF-8?Q?Pr=C3=BCfung=20Pr=C3=BCfung?="
10 echo iconv_mime_encode("Subject", "Prüfung Prüfung", $preferences);
11
12 $preferences["scheme"] = "B";
13 // This yields "Subject: =?UTF-8?B?UHLdVgZ1bmcgUHLdVgZ1bmc=?="
14 echo iconv_mime_encode("Subject", "Prüfung Prüfung", $preferences);
15 ?>
```

Success

No security issues were identified in the code snippet

4.1.135. intl.examples.basic.php

URL: <https://www.php.net/manual/en/intl.examples.basic.php>

4.1.135.1. Example 2

Code snippet:

```
1 <?php
2 $coll = new Collator('en_US');
3 $al   = $coll->getLocale(Locale::ACTUAL_LOCALE);
4 echo "Actual locale: $al\n";
5
6 $formatter = new NumberFormatter('en_US', NumberFormatter::DECIMAL);
7 echo $formatter->format(1234567);
8 ?>
```

Success

No security issues were identified in the code snippet

4.1.136. collator.compare.php

URL: <https://www.php.net/manual/en/collator.compare.php>

4.1.136.1. Example 2:

Code snippet:

```
1 <?php
2 $c = new Collator( 'en' );
3 $c->setStrength( Collator::PRIMARY );
4
5 if ( $c->compare( 'Séan', 'Sean' ) == 0 )
6 {
7     echo "The same\n";
8 }
```

Success

No security issues were identified in the code snippet

4.1.137. collator.create.php

URL: <https://www.php.net/manual/en/collator.create.php>

Code snippet:

```
1 <?php
2 $coll = collator_create( 'en_US' );
3
4 if( !isset( $coll ) ) {
5     printf( "Collator creation failed: %s\n", intl_get_error_message() );
6     exit( 1 );
7 }
8 ?>
```

Success

No security issues were identified in the code snippet

4.2. Documentation with issues discovered

This section provides all the documentation pages reviewed by Quarkslab for which security issues or bad practices have been identified.

Since the review is focused on code snippet, no severity ranking is provided as there is no context around the reviewed code.

However, Quarkslab recommended to fix all listed issues.

Retest 09/2025

Warning

The PHP maintainers decided to extend most code examples in the PHP documentation with an [appendix](#) clarifying that snippets omit error handling for readability. However, based on the comments of the pull request introducing this appendix, rendering it in the published documentation requires additional changes that, as of now, have not been implemented. Thus, the appendix is not rendered, and consequently, we treat all issues related to missing error handling as unresolved.

4.2.1. filter.constants.php

Code snippet:

```
1 <?php
2 function validate_login($value): ?string
3 {
4     if (strlen($value) >= 5 && ctype_alnum($value)) {
5         return $value;
6     }
7     return null;
8 }
9
10 $login = "valldL0gin";
11 $filtered_login = filter_var($login, FILTER_CALLBACK, ['options' =>
12     'validate_login']);
13 var_dump($filtered_login);
14
15 $login = "f&ke login";
16 $filtered_login = filter_var($login, FILTER_CALLBACK, ['options' =>
17     'validate_login']);
18 var_dump($filtered_login);
19 ?>
```

INFO-1	Type not properly declared
Perimeter	https://www.php.net/manual/en/filter.constants.php
Description	
Input type is not properly declared.	
Recommendation	
Declare input type as a <code>string</code> .	

Retest 09/2025

Success

The issue is correctly fixed.

INFO-2	Improper value handling
Perimeter	https://www.php.net/manual/en/filter.constants.php
Description	
Callback does not handle <code>null</code> value.	
Recommendation	
Handle <code>null</code> value in the callback.	

Retest 09/2025

Success

The issue is correctly fixed.

4.2.2. function.dio-open.php

Code snippet:

```
1 <?php
2
3 $fd = dio_open('/dev/ttyS0', O_RDWR | O_NOCTTY | O_NONBLOCK);
4
5 dio_close($fd);
6 ?>
```

INFO-3	Filesystem operations not checked
Perimeter	https://www.php.net/manual/en/function.dio-open.php
Description	
No check on the result of <code>dio_open()</code> before trying to <code>dio_close()</code> .	
Recommendation	
Check result of <code>dio_open()</code> before using <code>dio_close()</code> .	

Retest 09/2025

Warning
Currently, there is no visible fix implemented for the issue thus, it remains unaddressed.

4.2.3. function.dio-close.php

Code snippet:

```
1 <?php
2 $fd = dio_open('/dev/ttyS0', O_RDWR);
3
4 dio_close($fd);
5 ?>
```

INFO-4	Filesystem operations not checked
Perimeter	https://www.php.net/manual/en/function.dio-close.php
Description	
No check on the result of <code>dio_open()</code> before trying to <code>dio_close()</code> .	
Recommendation	
Check result of <code>dio_open()</code> before using <code>dio_close()</code> .	

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue thus, it remains unaddressed.

4.2.4. function.dio-fcntl.php

Code snippet:

```
1 <?php
2
3 $fd = dio_open('/dev/ttyS0', O_RDWR);
4
5 if (dio_fcntl($fd, F_SETLK, Array("type"=>F_WRLCK)) == -1) {
6     // the file descriptor appears locked
7     echo "The lock can not be cleared. It is held by someone else.";
8 } else {
9     echo "Lock successfully set/cleared";
10 }
11
12 dio_close($fd);
13 ?>
```

INFO-5

Filesystem operations not checked

Perimeter

<https://www.php.net/manual/en/function.dio-fcntl.php>

Description

No check on the result of `dio_open()`.

Recommendation

Check result of `dio_open()`.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue thus, it remains unaddressed.

4.2.5. function.dio-seek.php

Code snippet:

```
1 <?php
2
3 $fd = dio_open('/dev/ttyS0', O_RDWR);
4
5 dio_seek($fd, 10, SEEK_SET);
6 // position is now at 10 characters from the start of the file
7
8 dio_seek($fd, -2, SEEK_CUR);
9 // position is now at 8 characters from the start of the file
10
11 dio_seek($fd, -5, SEEK_END);
12 // position is now at 5 characters from the end of the file
13
14 dio_seek($fd, 10, SEEK_END);
15 // position is now at 10 characters past the end of the file.
16 // The 10 characters between the end of the file and the current
17 // position are filled with zeros.
18
19 dio_close($fd);
20 ?>
```

INFO-6

Filesystem operations not checked

Perimeter

<https://www.php.net/manual/en/function.dio-seek.php>

Description

No check on the result of `dio_open()`.

Recommendation

Check result of `dio_open()`.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue thus, it remains unaddressed.

4.2.6. function.dio-tcsetattr.php

Code snippet:

```
1 <?php
2
3 $fd = dio_open('/dev/ttyS0', O_RDWR | O_NOCTTY | O_NONBLOCK);
4
5 dio_fcntl($fd, F_SETFL, O_SYNC);
6
7 dio_tcsetattr($fd, array(
8     'baud' => 9600,
9     'bits' => 8,
10    'stop' => 1,
11    'parity' => 0
12 ));
13
14 while (1) {
15
16     $data = dio_read($fd, 256);
17
18     if ($data) {
19         echo $data;
20     }
21 }
22
23 ?>
```

INFO-7

Filesystem operations not checked

Perimeter

<https://www.php.net/manual/en/function.dio-tcsetattr.php>

Description

No check on the result of `dio_open()`.

Recommendation

Check result of `dio_open()`.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue thus, it remains unaddressed.

INFO-8

Weak validation/checking prone to bypass

Perimeter

<https://www.php.net/manual/en/function.dio-tcsetattr.php>

Description

The check on data can be bypassed with for instance 0

Recommendation

Strong verification is recommended such as `if ($data !== false && $data !== '')`

Retest 09/2025

Success

The issue is correctly fixed.

4.2.7. function.dir.php

Code snippet:

```
1 <?php
2 $d = dir("/etc/php5");
3 echo "Handle: " . $d->handle . "\n";
4 echo "Path: " . $d->path . "\n";
5 while (false !== ($entry = $d->read())) {
6     echo $entry . "\n";
7 }
8 $d->close();
9 ?>
```

INFO-9

Filesystem operations not checked

Perimeter

<https://www.php.net/manual/en/function.dir.php>

Description

No check on the result of `dir()`.

Recommendation

Check result of `dir()`.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue thus, it remains unaddressed.

4.2.8. function.unlink.php

Code snippet:

```
1 <?php
2 $fh = fopen('test.html', 'a');
3 fwrite($fh, '<h1>Hello world!</h1>');
4 fclose($fh);
5
6 unlink('test.html');
7 ?>
```

INFO-10	Filesystem operations not checked
Perimeter	https://www.php.net/manual/en/function.unlink.php
Description	
No check on the result of <code>fopen()</code> .	
Recommendation	
Check result of <code>fopen()</code> .	

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue thus, it remains unaddressed.

4.2.9. function.fclos.php

Code snippet:

```
1 <?php
2
3 $handle = fopen('somefile.txt', 'r');
4
5 fclose($handle);
6
7 ?>
```

INFO-11	Filesystem operations not checked
Perimeter	https://www.php.net/manual/en/function.fclose.php
Description	
No check on the result of <code>fopen()</code> .	
Recommendation	
Check result of <code>fopen()</code> .	

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.10. function.fdatasync.php

Code snippet:

```
1 <?php
2
3 $file = 'test.txt';
4
5 $stream = fopen($file, 'w');
6 fwrite($stream, 'test data');
7 fwrite($stream, "\r\n");
8 fwrite($stream, 'additional data');
9
10 fdatasync($stream);
11 fclose($stream);
12 ?>
```

INFO-12	Filesystem operations not checked
Perimeter	https://www.php.net/manual/en/function.fdatasync.php
Description	
No check on the result of <code>fopen()</code> .	
Recommendation	
Check result of <code>fopen()</code> .	

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.11. function.fflush.php

Code snippet:

```
1 <?php
2 $filename = 'bar.txt';
3
4 $file = fopen($filename, 'r+');
5 rewind($file);
6 fwrite($file, 'Foo');
7 fflush($file);
8 ftruncate($file, ftell($file));
9 fclose($file);
10 ?>
```

INFO-13	Filesystem operations not checked
Perimeter	https://www.php.net/manual/en/function.fflush.php
Description	
No check on the result of <code>fopen</code> .	
Recommendation	
Check result of <code>fopen</code> .	

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.12. function.flock.php

4.2.12.1. Example 1

Code snippet:

```
1 <?php
2
3 $fp = fopen("/tmp/lock.txt", "r+");
4
5 if (flock($fp, LOCK_EX)) { // acquire an exclusive lock
6     ftruncate($fp, 0);      // truncate file
7     fwrite($fp, "Write something here\n");
8     fflush($fp);           // flush output before releasing the lock
9     flock($fp, LOCK_UN);   // release the lock
10 } else {
11     echo "Couldn't get the lock!";
12 }
13
14 fclose($fp);
15
16 ?>
```

INFO-14

Filesystem operations not checked

Perimeter

<https://www.php.net/manual/en/function.flock.php>

Description

No check on the result of `fopen()`.

Recommendation

Check result of `fopen()`.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.12.2. Example 2

Code snippet:

```
1 <?php
2 $fp = fopen('/tmp/lock.txt', 'r+');
3
4 /* Activate the LOCK_NB option on an LOCK_EX operation */
5 if(!flock($fp, LOCK_EX | LOCK_NB)) {
6     echo 'Unable to obtain lock';
7     exit(-1);
8 }
9
10 /* ... */
11
12 fclose($fp);
13 ?>
```

INFO-15

Filesystem operations not checked

Perimeter

<https://www.php.net/manual/en/function.flock.php>

Description

No check on the result of `fopen()`.

Recommendation

Check result of `fopen()`.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.13. function.fpassthru.php

Code snippet:

```
1 <?php
2
3 // open the file in a binary mode
4 $name = './img/ok.png';
5 $fp = fopen($name, 'rb');
6
7 // send the right headers
8 header("Content-Type: image/png");
9 header("Content-Length: " . filesize($name));
10
11 // dump the picture and stop the script
12 fpassthru($fp);
13 exit;
14
15 ?>
```

INFO-16

Filesystem operations not checked

Perimeter

<https://www.php.net/manual/en/function.fpassthru.php>

Description

No check on the result of `fopen()`.

Recommendation

Check result of `fopen()`.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.14. function.popen.php

4.2.14.1. Example 1

Code snippet:

```
1 <?php
2 $handle = popen("/bin/ls", "r");
3 ?>
```

INFO-17

Filesystem operations not checked

Perimeter

<https://www.php.net/manual/en/function.popen.php>

Description

No check on the result of `fopen()`.

Recommendation

Check result of `fopen()`.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.14.2. Example 2

Code snippet:

```
1 <?php
2 error_reporting(E_ALL);
3
4 /* Add redirection so we can get stderr. */
5 $handle = popen('/path/to/executable 2>&1', 'r');
6 echo "'$handle'; " . gettype($handle) . "\n";
7 $read = fread($handle, 2096);
8 echo $read;
9 pclose($handle);
10 ?>
```

INFO-18	Filesystem operations not checked
Perimeter	https://www.php.net/manual/en/function.popen.php
Description	
No check on the result of <code>popen()</code> .	
Recommendation	
Check result of <code>popen()</code> .	

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.15. function.rewind.php

Code snippet:

```
1 <?php
2 $handle = fopen('output.txt', 'r+');
3
4 fwrite($handle, 'Really long sentence. ');
5 rewind($handle);
6 fwrite($handle, 'Foo');
7 rewind($handle);
8
9 echo fread($handle, filesize('output.txt'));
10
11 fclose($handle);
12 ?>
```

INFO-19

Filesystem operations not checked

Perimeter

<https://www.php.net/manual/en/function.rewind.php>

Description

No check on the result of `fopen()`.

Recommendation

Check result of `fopen()`.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.16. function.rewind.php

4.2.16.1. Example 1

Code snippet:

```
1 <?php
2 // get contents of a file into a string
3 $filename = "/usr/local/something.txt";
4 $handle = fopen($filename, "r");
5 $contents = fread($handle, filesize($filename));
6 fclose($handle);
7 ?>
```

INFO-20

Filesystem operations not checked

Perimeter

<https://www.php.net/manual/en/function.fread.php>

Description

No check on the result of `fopen()`.

Recommendation

Check result of `fopen()`.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.16.2. Example 2

Code snippet:

```
1 <?php
2 $filename = "c:\\files\\somepic.gif";
3 $handle = fopen($filename, "rb");
4 $contents = fread($handle, filesize($filename));
5 fclose($handle);
6 ?>
```

INFO-21	Filesystem operations not checked
Perimeter	https://www.php.net/manual/en/function.fread.php
Description	
No check on the result of <code>fopen()</code> .	
Recommendation	
Check result of <code>fopen()</code> .	

4.2.16.3. Example 3.1

Code snippet:

```

1 <?php
2 $handle = fopen("http://www.example.com/", "rb");
3 $contents = stream_get_contents($handle);
4 fclose($handle);
5 ?>

```

INFO-22	Filesystem operations not checked
Perimeter	https://www.php.net/manual/en/function.fread.php
Description	
No check on the result of <code>fopen()</code> .	
Recommendation	
Check result of <code>fopen()</code> .	

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.17. function.fwrite.php

4.2.17.1. Note 2

Code snippet:

```
1 <?php
2 $fp = fopen('data.txt', 'w');
3 fwrite($fp, '1');
4 fwrite($fp, '23');
5 fclose($fp);
6
7 // the content of 'data.txt' is now 123 and not 23!
8 ?>
```

INFO-23	Filesystem operations not checked
Perimeter	https://www.php.net/manual/en/function.fwrite.php
Description	
No check on the result of <code>fopen()</code> .	
Recommendation	
Check result of <code>fopen()</code> .	

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.18. function.fscanf.php

Code snippet:

```
1 <?php
2 $handle = fopen("users.txt", "r");
3 while ($userinfo = fscanf($handle, "%s\t%s\t%s\n")) {
4     list ($name, $profession, $countrycode) = $userinfo;
5     //... do something with the values
6 }
7 fclose($handle);
8 ?>
```

INFO-24	Filesystem operations not checked
Perimeter	https://www.php.net/manual/en/function.fscanf.php
Description	
No check on the result of <code>fopen()</code> .	
Recommendation	
Check result of <code>fopen()</code> .	

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.19. function.fseek.php

Code snippet:

```
1 <?php
2
3 $fp = fopen('somefile.txt', 'r');
4
5 // read some data
6 $data = fgets($fp, 4096);
7
8 // move back to the beginning of the file
9 // same as rewind($fp);
10 fseek($fp, 0);
11
12 ?>
```

INFO-25

Filesystem operations not checked

Perimeter

<https://www.php.net/manual/en/function.fseek.php>

Description

No check on the result of `fopen()`.

Recommendation

Check result of `fopen()`.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.20. function.ftell.php

Code snippet:

```
1 <?php
2
3 // opens a file and read some data
4 $fp = fopen("/etc/passwd", "r");
5 $data = fgets($fp, 12);
6
7 // where are we ?
8 echo ftell($fp); // 11
9
10 fclose($fp);
11
12 ?>
```

INFO-26	Filesystem operations not checked
Perimeter	https://www.php.net/manual/en/function.ftell.php
Description	
No check on the result of <code>fopen()</code> .	
Recommendation	
Check result of <code>fopen()</code> .	

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.21. function.fputcsv.php

Code snippet:

```
1 <?php
2
3 $list = [
4     ['aaa', 'bbb', 'ccc', 'ddd'],
5     ['123', '456', '789'],
6     ['"aaa"', '"bbb"']
7 ];
8
9 $fp = fopen('file.csv', 'w');
10
11 foreach ($list as $fields) {
12     fputcsv($fp, $fields, ',', '"', '');
13 }
14
15 fclose($fp);
16 ?>
```

INFO-27

Filesystem operations not checked

Perimeter

<https://www.php.net/manual/en/function.fputcsv.php>

Description

No check on the result of `fopen()`.

Recommendation

Check result of `fopen()`.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.22. function.fstat.php

Code snippet:

```
1 <?php
2
3 // open a file
4 $fp = fopen("/etc/passwd", "r");
5
6 // gather statistics
7 $fstat = fstat($fp);
8
9 // close the file
10 fclose($fp);
11
12 // print only the associative part
13 print_r(array_slice($fstat, 13));
14
15 ?>
```

INFO-28

Filesystem operations not checked

Perimeter

<https://www.php.net/manual/en/function.fstat.php>

Description

No check on the result of `fopen()`.

Recommendation

Check result of `fopen()`.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.23. function.fsync.php

Code snippet:

```
1 <?php
2
3 $file = 'test.txt';
4
5 $stream = fopen($file, 'w');
6 fwrite($stream, 'test data');
7 fwrite($stream, "\r\n");
8 fwrite($stream, 'additional data');
9
10 fsync($stream);
11 fclose($stream);
12 ?>
```

INFO-29

Filesystem operations not checked

Perimeter

<https://www.php.net/manual/en/function.fsync.php>

Description

No check on the result of `fopen()`.

Recommendation

Check result of `fopen()`.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.24. intro.enchant.php

Code snippet: Not applicable (N/A) as this concerns a broken link in the documentation.

INFO-30	Broken link
Perimeter	https://www.php.net/manual/en/intro.enchant.php
Description	
The Enchant library link is broken.	
Recommendation	
Update with proper link to the library.	

Retest 09/2025

Success

The issue is correctly fixed.

4.2.25. function.enchant-broker-dict-exists.php

Code snippet:

```
1 <?php
2 $tag = 'en_US';
3 $r = enchant_broker_init();
4 if (enchant_broker_dict_exists($r,$tag)) {
5     echo $tag . " dictionary found.\n";
6 }
7 ?>
```

INFO-31

Lack of exception handling

Perimeter

<https://www.php.net/manual/en/function.enchant-broker-dict-exists.php>

Description

Unhandled case where `enchant_broker_init()` returns `false`, `enchant_broker_dict_exists` fails with:

```
1 PHP Fatal error: Uncaught TypeError: enchant_broker_dict_exists():
Argument #1 ($broker) must be of type EnchantBroker
```

Shell

Recommendation

Add exception handling for the case where `$r` is `false`.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.26. function.enchant-broker-list-dicts.php

Code snippet:

```
1 <?php
2 $r = enchant_broker_init();
3 $dicts = enchant_broker_list_dicts($r);
4 print_r($dicts);
5 ?>
```

INFO-32	Lack of exception handling
Perimeter	https://www.php.net/manual/en/function.enchant-broker-list-dicts.php
Description	
The code does not handle the case where <code>\$r</code> is <code>false</code> before calling to <code>enchant_broker_list_dicts</code> .	
Recommendation	
Add exception handling for the case where <code>\$r</code> is <code>false</code> .	

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.27. function.enchant-broker-request-dict.php

Code snippet:

```
1 <?php
2 $tag = 'en_US';
3 $broker = enchant_broker_init();
4 if (enchant_broker_dict_exists($broker,$tag)) {
5     $dict = enchant_broker_request_dict($r, $tag);
6 }
7 ?>
```

INFO-33

Lack of exception handling

Perimeter

<https://www.php.net/manual/en/function.enchant-broker-request-dict.php>

Description

The code does not handle the case where `$broker` is `false` before calling `enchant_broker_dict_exists`.

Recommendation

Add exception handling when `enchant_broker_init()` returns `false`.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

INFO-34

Typo error

Perimeter

<https://www.php.net/manual/en/function.enchant-broker-request-dict.php>

Description

There is a typo in the documentation example:

```
1 $tag = 'en_US';
```

```
2 $broker = enchant_broker_init();
3 if (enchant_broker_dict_exists($broker,$tag)) {
4     // $r below should be $broker
5     $dict = enchant_broker_request_dict($r, $tag);
6 }
```

Recommendation

Replace \$r with \$broker.

Retest 09/2025

Success

The issue is correctly fixed.

4.2.28. function.enchant-dict-add.php

Code snippet:

```
1 <?php
2
3 $filename = './my_word_list.pwl';
4 $word = 'Supercalifragilisticexpialidocious';
5
6 $broker = enchant_broker_init();
7 $dict = enchant_broker_request_pwl_dict($broker, $filename);
8
9 enchant_dict_add($dict, $word);
10
11 enchant_broker_free($broker);
12
13 ?>
```

INFO-35

Usage of deprecated function

Perimeter

<https://www.php.net/manual/en/function.enchant-dict-add.php>

Description

Contains code which could not run on recent PHP versions because it contains functions deprecated since PHP 8.0.0

Recommendation

Remove deprecated functions.

```
1 $filename = './my_word_list.pwl';
2 $word = 'Supercalifragilisticexpialidocious';
3 $broker = enchant_broker_init();
4 $dict = enchant_broker_request_pwl_dict($broker, $filename);
5 enchant_dict_add($dict, $word);
6 enchant_broker_free($broker);
```

Retest 09/2025

Success

The issue is correctly fixed.

INFO-36

Lack of exception handling

Perimeter<https://www.php.net/manual/en/function.enchant-dict-add.php>**Description**

The code does not handle the case where `$dict` is `false` before calling `enchant_dict_add`.

Recommendation

Add exception handling for the case where `$dict` is `false`.

Retest 09/2025**Warning**

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

INFO-37

Lack of exception handling

Perimeter<https://www.php.net/manual/en/function.enchant-dict-add.php>**Description**

The code does not handle the case where `$broker` is `false` before calling `enchant_broker_request_pwl_dict`.

Recommendation

Add exception handling for the case where `$broker` is `false`.

Retest 09/2025**Warning**

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.29. function.enchant-dict-describe.php

Code snippet:

```
1 <?php
2
3 $filename = './my_word_list.pwl';
4 $word = 'Supercalifragilisticexpialidocious';
5
6 $broker = enchant_broker_init();
7 $dict = enchant_broker_request_pwl_dict($broker, $filename);
8
9 enchant_dict_add($dict, $word);
10
11 enchant_broker_free($broker);
12
13 ?>
```

INFO-38	Typo error
Perimeter	https://www.php.net/manual/en/function.enchant-dict-describe.php
Description	
There is a typo in the documentation: <code>\$dict = enchant_broker_request_dict(\$r, \$tag)</code> should be replaced with <code>\$dict = enchant_broker_request_dict(\$broker, \$tag)</code> .	
Recommendation	
<code>\$dict = enchant_broker_request_dict(\$r, \$tag)</code> should be replaced with <code>\$dict = enchant_broker_request_dict(\$broker, \$tag)</code> .	

Retest 09/2025

Success

The issue is correctly fixed.

INFO-39	Lack of exception handling
----------------	----------------------------

Perimeter

<https://www.php.net/manual/en/function.enchant-dict-describe.php>

Description

The code does not check if `$dict` is `false` before calling `enchant_dict_describe($dict)`.

Recommendation

Add exception handling for the case where `$dict` is `false`.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.30. function.enchant-dict-quick-check.php

Code snippet:

```
1 <?php
2 $tag = 'en_US';
3 $r = enchant_broker_init();
4
5 if (enchant_broker_dict_exists($r,$tag)) {
6     $d = enchant_broker_request_dict($r, $tag);
7     enchant_dict_quick_check($d, 'soong', $suggs);
8     print_r($suggs);
9 }
10 ?>
```

INFO-40	Lack of exception handling
Perimeter	https://www.php.net/manual/en/function.enchant-dict-quick-check.php
Description	
There is no check if \$d is not false before calling <code>enchant_dict_quick_check(\$d, 'soong', \$suggs)</code>	
Recommendation	
Add exception handling for the case where \$d is false.	

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.31. function.enchant-dict-suggest.php

Code snippet:

```
1 <?php
2 $tag = 'en_US';
3 $r = enchant_broker_init();
4 if (enchant_broker_dict_exists($r,$tag)) {
5     $d = enchant_broker_request_dict($r, $tag);
6
7     $wordcorrect = enchant_dict_check($d, "soong");
8     if (!$wordcorrect) {
9         $suggs = enchant_dict_suggest($d, "soong");
10        echo "Suggestions for 'soong':";
11        print_r($suggs);
12    }
13    enchant_broker_free_dict($d);
14 }
15 enchant_broker_free($r);
16 ?>
```

INFO-41

Usage of deprecated function

Perimeter

<https://www.php.net/manual/en/function.enchant-dict-suggest.php>

Description

The code contains deprecated functions - `enchant_broker_free_dict($d)` and `enchant_broker_free($r)`.

Recommendation

Remove the deprecated functions.

Retest 09/2025

Success

The issue is correctly fixed.

INFO-42

Lack of exception handling

Perimeter

<https://www.php.net/manual/en/function.enchant-dict-suggest.php>

Description

There is no check if `$d` is not `false` before calling `enchant_dict_suggest($d, "soong"` which could lead to an application exception.

Recommendation

Add exception handling for the case where `$d` is `false`.

Retest 09/2025**Warning**

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.32. function.iconv-get-encoding.php

Code snippet:

```
1 <pre>
2 <?php
3 iconv_set_encoding("internal_encoding", "UTF-8");
4 iconv_set_encoding("output_encoding", "ISO-8859-1");
5 var_dump(iconv_get_encoding('all'));
6 ?>
7 </pre>
```

INFO-43

Lack of exception handling

Perimeter

<https://www.php.net/manual/en/function.iconv-get-encoding.php>

Description

The code snippets do not check the return values of the `iconv` function, which could indicate an error.

Recommendation

Implement proper checks for the return values of `iconv` to handle potential errors.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.33. intl.examples.basic.php

4.2.33.1. Example 1

Code snippet:

```
1 <?php
2 $coll = collator_create('en_US');
3 $result = collator_compare($coll, "string#1", "string#2");
4 ?>
```

INFO-44

Lack of exception handling

Perimeter

<https://www.php.net/manual/en/intl.examples.basic.php>

Description

The snippet code does not verify whether `collator_create()` returns `NULL`, which would indicate an error.

Recommendation

Include checks to ensure `collator_create()` does not return `NULL` before proceeding.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.34. collator.asort.php

Code snippet:

```
1 <?php
2 $coll = collator_create( 'en_US' );
3 $arr = array(
4     'a' => '100',
5     'b' => '50',
6     'c' => '7'
7 );
8 collator_asort( $coll, $arr, Collator::SORT_NUMERIC );
9 var_export( $arr );
10
11 collator_asort( $coll, $arr, Collator::SORT_STRING );
12 var_export( $arr );
13 ?>
```

INFO-45	Lack of exception handling
Perimeter	https://www.php.net/manual/en/collator.asort.php
Description	
Snippet code does not verify if <code>collator_create()</code> returns <code>NULL</code> , which would indicate an error.	
Recommendation	
Add proper checks in the code snippet.	

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.35. collator.compare.php

4.2.35.1. Example 1

Code snippet:

```
1 <?php
2 $s1 = 'Hello';
3 $s2 = 'hello';
4
5 $coll = collator_create( 'en_US' );
6 $res = collator_compare( $coll, $s1, $s2 );
7
8 if ( $res === false ) {
9     echo collator_get_error_message( $coll );
10 } else if( $res > 0 ) {
11     echo "s1 is greater than s2\n";
12 } else if( $res < 0 ) {
13     echo "s1 is less than s2\n";
14 } else {
15     echo "s1 is equal to s2\n";
16 }
17 ?>
```

INFO-46

Lack of exception handling

Perimeter

<https://www.php.net/manual/en/collator.compare.php>

Description

Snippet code does not verify if `collator_create()` returns `NULL`, which would indicate an error.

Recommendation

Add proper checks in the code snippet.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.36. collator.getattribute.php

Code snippet:

```
1 <?php
2 $coll = collator_create( 'en_CA' );
3 $val = collator_get_attribute( $coll, Collator::NUMERIC_COLLATION );
4 if( $val === false )
5 {
6     // Handle error.
7 }
8 ?>
```

INFO-47

Lack of exception handling

Perimeter

<https://www.php.net/manual/en/collator.getattribute.php>

Description

Snippet code does not verify if `collator_create()` returns `NULL`, which would indicate an error.

Recommendation

Add proper checks in the code snippet.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.37. collator.geterrorcode.php

Code snippet:

```
1 <?php
2 $coll = collator_create( 'en_US' );
3 if( collator_get_attribute( $coll, Collator::FRENCH_COLLATION ) === false )
4     handle_error( collator_get_error_code() );
5 ?>
```

INFO-48	Lack of exception handling
Perimeter	https://www.php.net/manual/en/collator.geterrorcode.php
Description	
Snippet code does not verify if collator_create() returns NULL, which would indicate an error.	
Recommendation	
Add proper checks in the code snippet.	

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.38. collator.geterrormessage.php

Code snippet:

```
1 <?php
2 $coll = collator_create( 'lt' );
3 if( collator_compare( $coll, 'y', 'k' ) === false ) {
4     echo collator_get_error_message( $coll );
5 }
6 ?>
```

INFO-49	Lack of exception handling
Perimeter	https://www.php.net/manual/en/collator.geterrormessage.php
Description	
Snippet code does not verify if <code>collator_create()</code> returns <code>NULL</code> , which would indicate an error.	
Recommendation	
Add proper checks in the code snippet.	

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.39. collator.getlocale.php

Code snippet:

```
1 <?php
2 $coll = collator_create( 'en_US_California' );
3 $res_val = collator_get_locale( $coll, Locale::VALID_LOCALE );
4 $res_act = collator_get_locale( $coll, Locale::ACTUAL_LOCALE );
5 printf( "Valid locale name: %s\nActual locale name: %s\n",
6         $res_val, $res_act );
7 ?>
```

INFO-50	Lack of exception handling
Perimeter	https://www.php.net/manual/en/collator.getlocale.php
Description	
Snippet code does not verify if collator_create() returns NULL, which would indicate an error.	
Recommendation	
Add proper checks in the code snippet.	

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.40. collator.getsortkey.php

Code snippet:

```
1 <?php
2 $s1 = 'Hello';
3
4 $coll = collator_create('en_US');
5 $res  = collator_get_sort_key($coll, $s1);
6
7 echo bin2hex($res);
8 ?>
```

INFO-51

Lack of exception handling

Perimeter

<https://www.php.net/manual/en/collator.getsortkey.php>

Description

Snippet code does not verify if `collator_create()` returns `NULL`, which would indicate an error.

Recommendation

Add proper checks in the code snippet.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.41. collator.setattribute.php

Code snippet:

```
1 <?php
2 $coll = collator_create( 'en_CA' );
3 $val = collator_get_attribute( $coll, Collator::NUMERIC_COLLATION );
4 if ($val === false) {
5     // Handle error.
6 } elseif ($val === Collator::ON) {
7     // Do something useful.
8 }
9 ?>
```

INFO-52

Lack of exception handling

Perimeter

<https://www.php.net/manual/en/collator.setattribute.php>

Description

Snippet code does not verify if `collator_create()` returns `NULL`, which would indicate an error.

Recommendation

Add proper checks in the code snippet.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.42. collator.sort.php

Code snippet:

```
1 <?php
2 $coll = collator_create( 'en_US' );
3 $arr = array( 'at', 'às', 'as' );
4
5 var_export( $arr );
6 collator_sort( $coll, $arr );
7 var_export( $arr );
8 ?>
```

INFO-53

Lack of exception handling

Perimeter

<https://www.php.net/manual/en/collator.sort.php>

Description

Snippet code does not verify if `collator_create()` returns `NULL`, which would indicate an error.

Recommendation

Add proper checks in the code snippet.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.43. collator.sortwithsortkeys.php

Code snippet:

```
1 <?php
2 $arr = array( 'Köpfe', 'Kyper', 'Kopfe' );
3 $coll = collator_create( 'sv' );
4
5 collator_sort_with_sort_keys( $coll, $arr );
6 var_export( $arr );
7 ?>
```

INFO-54

Lack of exception handling

Perimeter

<https://www.php.net/manual/en/collator.sortwithsortkeys.php>

Description

Snippet code does not verify if `collator_create()` returns `NULL`, which would indicate an error.

Recommendation

Add proper checks in the code snippet.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.44. numberformatter.format.php

4.2.44.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = numfmt_create( 'de_DE', NumberFormatter::DECIMAL );
3 $data = numfmt_format($fmt, 1234567.891234567890000);
4 if(intl_is_failure(numfmt_format($fmt))) {
5     report_error("Formatter error");
6 }
7 ?>
```

INFO-55

Usage of undefined and/or obsolete function

Perimeter

<https://www.php.net/manual/en/numberformatter.format.php>

Description

The `report_error()` function is undefined.

Recommendation

Remove obsolete functions.

Retest 09/2025

Success

The issue is correctly fixed.

INFO-56

Wrong parameters

Perimeter

<https://www.php.net/manual/en/numberformatter.format.php>

Description

The `numfmt_format()` takes 2 arguments rather than 1.

Recommendation

Fix code snippet to illustrate proper usage of the `numfmt_format()` function.

Retest 09/2025

Success

The issue is correctly fixed.

INFO-57

Wrong parameters

Perimeter

<https://www.php.net/manual/en/numberformatter.format.php>

Description

The `intl_is_failure()` function receives an integer error code and not a string which is returned from `numfmt_format()`.

Recommendation

fix code snippet to illustrate proper usage of the `intl_is_failure()` function.

Retest 09/2025

Success

The issue is correctly fixed.

4.2.44.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
3 $fmt->format(1234567.891234567890000);
4 if(intl_is_failure($fmt->getErrorCode())) {
5     report_error("Formatter error");
6 }
7 ?>
```

INFO-58

Usage of undefined and/or obsolete function

Perimeter

<https://www.php.net/manual/en/numberformatter.format.php>

Description

The `report_error()` function is undefined.

Recommendation

Change the code snippet with up-to-date functions.

Retest 09/2025

Success

The issue is correctly fixed.

4.2.45. numberformatter.getattribute.php

4.2.45.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = numfmt_create( 'de_DE', NumberFormatter::DECIMAL );
3 echo "Digits: ".numfmt_get_attribute($fmt,
4   NumberFormatter::MAX_FRACTION_DIGITS)."\n";
5 echo numfmt_format($fmt, 1234567.891234567890000)."\n";
6 numfmt_set_attribute($fmt, NumberFormatter::MAX_FRACTION_DIGITS, 2);
7 echo "Digits: ".numfmt_get_attribute($fmt,
8   NumberFormatter::MAX_FRACTION_DIGITS)."\n";
9 echo numfmt_format($fmt, 1234567.891234567890000)."\n";
10 ?>
```

INFO-59	Lack of exception handling
Perimeter	https://www.php.net/manual/en/numberformatter.getattribute.php
Description	
Snippet code does not verify if \$fmt is not NULL, which would indicate an error.	
Recommendation	
Add proper checks in the code snippet.	

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue thus, it remains unaddressed.

4.2.46. numberformatter.geterrorcode.php

4.2.46.1. Example 1

Code snippet:

```
1 <?php
2 $fmt = numfmt_create( 'de_DE', NumberFormatter::DECIMAL );
3 $data = numfmt_format($fmt, 1234567.891234567890000);
4 if(intl_is_failure(numfmt_get_error_code($fmt))) {
5     report_error("Formatter error");
6 }
7 ?>
```

INFO-60

Usage of undefined and/or obsolete function

Perimeter

<https://www.php.net/manual/en/numberformatter.geterrorcode.php>

Description

The `report_error()` function is undefined.

Recommendation

Change the code snippet with up-to-date functions.

Retest 09/2025

Success

The issue is correctly fixed.

4.2.46.2. Example 2

Code snippet:

```
1 <?php
2 $fmt = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
3 $fmt->format(1234567.891234567890000);
4 if(intl_is_failure($fmt->getErrorCode())) {
5     report_error("Formatter error");
6 }
7 ?>
```

INFO-61	Usage of undefined and/or obsolete function
Perimeter	https://www.php.net/manual/en/numberformatter.geterrorcode.php
Description	
The <code>report_error()</code> function is undefined.	
Recommendation	
Change the code snippet with up-to-date functions.	

Retest 09/2025

Success

The issue is correctly fixed.

4.2.47. intlcalendar.createinstance.php

Code snippet:

```
1 <?php
2 ini_set('intl.default_locale', 'es_ES');
3 ini_set('date.timezone', 'Europe/Madrid');
4
5 $cal = IntlCalendar::createInstance();
6 echo "No arguments\n";
7 var_dump(get_class($cal),
8         IntlDateFormatter::formatObject($cal, IntlDateFormatter::FULL));
9 echo "\n";
10
11 echo "Explicit timezone\n";
12 $cal = IntlCalendar::createInstance(IntlTimeZone::getGMT());
13 var_dump(get_class($cal),
14         IntlDateFormatter::formatObject($cal, IntlDateFormatter::FULL));
15 echo "\n";
16
17 echo "Explicit locale (with calendar)\n";
18 $cal = IntlCalendar::createInstance(NULL, 'es_ES@calendar=persian');
19 var_dump(get_class($cal),
20         IntlDateFormatter::formatObject($cal, IntlDateFormatter::FULL));
```

INFO-62

Lack of exception handling

Perimeter

<https://www.php.net/manual/en/intlcalendar.createinstance.php>

Description

Snippet code does not verify if `$cal` is not `NULL`, which would indicate an error.

Recommendation

Add proper checks in the code snippet.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.48. intlcalendar.equals.php

Code snippet:

```
1 <?php
2 ini_set('date.timezone', 'UTC');
3
4 $cal1 = IntlCalendar::createInstance(NULL, 'es_ES');
5 $cal2 = clone $cal1;
6
7 var_dump($cal1->equals($cal2)); //TRUE
8
9 //The locale is not included in the comparison
10 $cal2 = IntlCalendar::createInstance(NULL, 'pt_PT');
11 $cal2->setTime($cal1->getTime());
12 var_dump($cal1->equals($cal2)); //TRUE
13
14 //And set fields state is not included as well
15 $cal2->clear(IntlCalendar::FIELD_YEAR);
16 var_dump($cal1->isSet(IntlCalendar::FIELD_YEAR) ==
17         $cal2->isSet(IntlCalendar::FIELD_YEAR)); //FALSE
18 var_dump($cal1->equals($cal2)); //TRUE
19
20 //Neither is the calendar type
21 $cal2 = IntlCalendar::createInstance(NULL, 'es_ES@calendar=islamic');
22 $cal2->setTime($cal1->getTime());
23 var_dump($cal1->equals($cal2)); //TRUE
24
25 //Only the time is
26 $cal2 = clone $cal1;
27 $cal2->setTime($cal1->getTime() + 1.);
28 var_dump($cal1->equals($cal2)); //FALSE
```

INFO-63

Lack of exception handling

Perimeter

<https://www.php.net/manual/en/intlcalendar.equals.php>

Description

Snippet code does not verify if `$cal1` and `$cal2` are not NULL, which would indicate an error.

Recommendation

Add proper checks in the code snippet.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

4.2.49. intlcalendar.fielddifference.php

Code snippet:

```
1 <?php
2 ini_set('date.timezone', 'Europe/Lisbon');
3 ini_set('intl.default_locale', 'fr_FR');
4
5 $cal1 = IntlCalendar::fromDateTime('2012-02-29 09:00:11');
6 $cal2 = IntlCalendar::fromDateTime('2013-03-01 09:19:29');
7 $time = $cal2->getTime();
8
9 echo "Time before: ", IntlDateFormatter::formatObject($cal1, "\n";
10
11 printf(
12     "The difference in time is %d year(s), %d month(s), "
13     . "%d day(s), %d hour(s) and %d minute(s)\n",
14     $cal1->fieldDifference($time, IntlCalendar::FIELD_YEAR),
15     $cal1->fieldDifference($time, IntlCalendar::FIELD_MONTH),
16     $cal1->fieldDifference($time, IntlCalendar::FIELD_DAY_OF_MONTH),
17     $cal1->fieldDifference($time, IntlCalendar::FIELD_HOUR_OF_DAY),
18     $cal1->fieldDifference($time, IntlCalendar::FIELD_MINUTE)
19 );
20
21 //now it was advanced to the target time, exception for the seconds,
22 //for which we did not measure the difference
23 echo "Time after: ", IntlDateFormatter::formatObject($cal1, "\n";
```

INFO-64

Lack of exception handling

Perimeter

<https://www.php.net/manual/en/intlcalendar.fielddifference.php>

Description

Snippet code does not verify if \$cal1 and \$cal2 are NULL or not.

Recommendation

Add proper checks in the code snippet.

Retest 09/2025 Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue nor any warning thus, it remains unaddressed.

INFO-65

Usage of undefined and/or obsolete function

Perimeter

<https://www.php.net/manual/en/intlcalendar.fielddifference.php>

Description

`IntlCalendar::fromDateTime()` has an obsolete definition and it is missing an additional argument - the locale.

Recommendation

Change the code snippet with up-to-date functions and with proper argument usage.

Success

The issue is correctly fixed.

4.2.50. intlcalendar.fromdatetime.php

Code snippet:

```
1 <?php
2 ini_set('date.timezone', 'Europe/Lisbon');
3
4 //same as IntlCalendar::fromDateTime(new DateTime(...))
5 $call = IntlCalendar::fromDateTime('2013-02-28 00:01:02 Europe/Berlin');
6
7 //Note the timezone is Europe/Berlin, not the default Europe/Lisbon
8 echo IntlDateFormatter::formatObject($call, 'yyyy MMMM d HH:mm:ss VVVV',
   'de_DE'), "\n";
```

INFO-66

Lack of exception handling

Perimeter

<https://www.php.net/manual/en/intlcalendar.fromdatetime.php>

Description

Snippet code does not verify if \$call is not NULL.

Recommendation

Add proper checks in the code snippet.

Retest 09/2025

Warning

Currently, there is no visible fix implemented for the issue thus, it remains unaddressed.

INFO-67

Usage of undefined and/or obsolete function

Perimeter

<https://www.php.net/manual/en/intlcalendar.fromdatetime.php>

Description

Call of IntlCalendar::fromDateTime() has to be updated to
`$call = IntlCalendar::fromDateTime('2013-02-28 00:01:02 Europe/Berlin', 'de_DE');`.

Recommendation

Change the code snippet with up-to-date functions and with proper argument usage.

Retest 09/2025

Success

The issue is correctly fixed.

4.3. Documentation with no code

During the review, Quarkslab auditors visited these pages and marked them as not applicable for a review since they do not contain any code snippet.

- <https://www.php.net/manual/en/book.filter.php>
- <https://www.php.net/manual/en/intro.filter.php>
- <https://www.php.net/manual/en/filter.setup.php>
- <https://www.php.net/manual/en/filter.installation.php>
- <https://www.php.net/manual/en/filter.configuration.php>
- <https://www.php.net/manual/en/filter.filters.php>
- <https://www.php.net/manual/en/filter.filters.flags.php>
- <https://www.php.net/manual/en/filter.examples.php>
- <https://www.php.net/manual/en/refs.fileprocess.file.php>
- <https://www.php.net/manual/en/book.dio.php>
- <https://www.php.net/manual/en/intro.dio.php>
- <https://www.php.net/manual/en/dio.installation.php>
- <https://www.php.net/manual/en/dio.resources.php>
- <https://www.php.net/manual/en/dio.constants.php>
- <https://www.php.net/manual/en/ref.dio.php>
- <https://www.php.net/manual/en/function.dio-read.php>
- <https://www.php.net/manual/en/function.dio-stat.php>
- <https://www.php.net/manual/en/function.dio-truncate.php>
- <https://www.php.net/manual/en/function.dio-write.php>
- <https://www.php.net/manual/en/book.dir.php>
- <https://www.php.net/manual/en/dir.constants.php>
- <https://www.php.net/manual/en/class.directory.php>
- <https://www.php.net/manual/en/directory.close.php>
- <https://www.php.net/manual/en/directory.read.php>
- <https://www.php.net/manual/en/directory.rewind.php>
- <https://www.php.net/manual/en/function.rewinddir.php>
- <https://www.php.net/manual/en/directory.close.php>
- <https://www.php.net/manual/en/ref.dir.php>
- <https://www.php.net/manual/en/book.fileinfo.php>
- <https://www.php.net/manual/en/intro.fileinfo.php>
- <https://www.php.net/manual/en/fileinfo.setup.php>
- <https://www.php.net/manual/en/fileinfo.installation.php>
- <https://www.php.net/manual/en/fileinfo.resources.php>
- <https://www.php.net/manual/en/fileinfo.constants.php>
- <https://www.php.net/manual/en/ref.fileinfo.php>
- <https://www.php.net/manual/en/function.finfo-close.php>
- <https://www.php.net/manual/en/function.finfo-set-flags.php>
- <https://www.php.net/manual/en/class.finfo.php>
- <https://www.php.net/manual/en/finfo.buffer.php>
- <https://www.php.net/manual/en/finfo.construct.php>

- <https://www.php.net/manual/en/finfo.file.php>
- <https://www.php.net/manual/en/finfo.set-flags.php>
- <https://www.php.net/manual/en/book.filesystem.php>
- <https://www.php.net/manual/en/intro.filesystem.php>
- <https://www.php.net/manual/en/filesystem.setup.php>
- <https://www.php.net/manual/en/filesystem.configuration.php>
- <https://www.php.net/manual/en/filesystem.resources.php>
- <https://www.php.net/manual/en/filesystem.constants.php>
- <https://www.php.net/manual/en/ref.filesystem.php>
- <https://www.php.net/manual/en/function.chmod.php>
- <https://www.php.net/manual/en/function.delete.php>
- <https://www.php.net/manual/en/function.diskfreespace.php>
- <https://www.php.net/manual/en/function.getmyinode.php>
- <https://www.php.net/manual/en/function.ob-get-level.php>
- <https://www.php.net/manual/en/function.require.php>
- <https://www.php.net/manual/en/function.virtual.php>
- <https://www.php.net/manual/en/function.fputs.php>
- <https://www.php.net/manual/en/book.password.php>
- <https://www.php.net/manual/en/intro.password.php>
- <https://www.php.net/manual/en/password.setup.php>
- <https://www.php.net/manual/en/password.requirements.php>
- <https://www.php.net/manual/en/password.installation.php>
- <https://www.php.net/manual/en/password.constants.php>
- <https://www.php.net/manual/en/ref.password.php>
- <https://www.php.net/manual/en/function.password-get-info.php>
- <https://www.php.net/manual/en/function.hash-update.php>
- <https://www.php.net/manual/en/function.hash-update-file.php>
- <https://www.php.net/manual/en/function.hash-final.php>
- <https://www.php.net/manual/en/function.sodium-crypto-pwhash-str-verify.php>
- <https://www.php.net/manual/en/book.gnupg.php>
- <https://www.php.net/manual/en/intro.gnupg.php>
- <https://www.php.net/manual/en/gnupg.setup.php>
- <https://www.php.net/manual/en/gnupg.requirements.php>
- <https://www.php.net/manual/en/gnupg.installation.php>
- <https://www.php.net/manual/en/gnupg.constants.php>
- <https://www.php.net/manual/en/gnupg.examples.php>
- <https://www.php.net/manual/en/function.enchant-broker-describe.php>
- <https://www.php.net/manual/en/function.enchant-broker-free.php>
- <https://www.php.net/manual/en/function.enchant-broker-free-dict.php>
- <https://www.php.net/manual/en/function.enchant-broker-get-dict-path.php>
- <https://www.php.net/manual/en/function.enchant-broker-get-error.php>
- <https://www.php.net/manual/en/function.enchant-broker-init.php>
- <https://www.php.net/manual/en/function.enchant-broker-request-pwl-dict.php>
- <https://www.php.net/manual/en/function.enchant-broker-request-pwl-dict.php>

- <https://www.php.net/manual/en/function.enchant-broker-set-dict-path.php>
- <https://www.php.net/manual/en/function.enchant-broker-set-ordering.php>
- <https://www.php.net/manual/en/function.enchant-dict-add-to-personal.php>
- <https://www.php.net/manual/en/function.enchant-dict-add-to-session.php>
- <https://www.php.net/manual/en/function.enchant-dict-check.php>
- <https://www.php.net/manual/en/function.enchant-dict-get-error.php>
- <https://www.php.net/manual/en/function.enchant-dict-is-added.php>
- <https://www.php.net/manual/en/function.enchant-dict-is-in-session.php>
- <https://www.php.net/manual/en/function.enchant-dict-store-replacement.php>
- <https://www.php.net/manual/en/class.enchantbroker.php>
- <https://www.php.net/manual/en/class.enchantdictionary.php>
- <https://www.php.net/manual/en/gender-gender.connect.php>
- <https://www.php.net/manual/en/gender-gender.construct.php>
- <https://www.php.net/manual/en/gender-gender.get.php>
- <https://www.php.net/manual/en/gender-gender.isnick.php>
- <https://www.php.net/manual/en/gender-gender.similarnames.php>
- <https://www.php.net/manual/en/function.bind-textdomain-codeset.php>
- <https://www.php.net/manual/en/function.dcgettext.php>
- <https://www.php.net/manual/en/function.dcngettext.php>
- <https://www.php.net/manual/en/function.dgettext.php>
- <https://www.php.net/manual/en/function.textdomain.php>
- <https://www.php.net/manual/en/function.iconv-strlen.php>
- <https://www.php.net/manual/en/function.iconv-strpos.php>
- <https://www.php.net/manual/en/function.iconv-strrpos.php>
- <https://www.php.net/manual/en/function.iconv-substr.php>
- <https://www.php.net/manual/en/class.collator.php>
- <https://www.php.net/manual/en/class.numberformatter.php>
- <https://www.php.net/manual/en/class.locale.php>
- <https://www.php.net/manual/en/locale.canonicalize.php>
- <https://www.php.net/manual/en/class.normalizer.php>
- <https://www.php.net/manual/en/class.messageformatter.php>
- <https://www.php.net/manual/en/messageformatter.geterrorcode.php>
- <https://www.php.net/manual/en/class.intlcalendar.php>
- <https://www.php.net/manual/en/class.intlgregorianCalendar.php>
- <https://www.php.net/manual/en/intlgregorianCalendar.construct.php>
- <https://www.php.net/manual/en/intlgregorianCalendar.getgregorianchange.php>
- <https://www.php.net/manual/en/intlgregorianCalendar.isleapyear.php>
- <https://www.php.net/manual/en/intlgregorianCalendar.setgregorianchange.php>
- <https://www.php.net/manual/en/class.intltimezone.php>
- <https://www.php.net/manual/en/intlcalendar.before.php>
- <https://www.php.net/manual/en/intlcalendar.construct.php>
- <https://www.php.net/manual/en/intlcalendar.getminimum.php>
- <https://www.php.net/manual/en/intlcalendar.getmaximum.php>
- <https://www.php.net/manual/en/class.uconverter.php>

- <https://www.php.net/manual/en/uconverter.construct.php>
- <https://www.php.net/manual/en/uconverter.convert.php>
- <https://www.php.net/manual/en/uconverter.fromucallback.php>
- <https://www.php.net/manual/en/uconverter.getaliases.php>
- <https://www.php.net/manual/en/uconverter.getavailable.php>
- <https://www.php.net/manual/en/uconverter.getdestinationencoding.php>
- <https://www.php.net/manual/en/uconverter.getdestinationtype.php>
- <https://www.php.net/manual/en/uconverter.geterrorcode.php>
- <https://www.php.net/manual/en/uconverter.geterrorMessage.php>
- <https://www.php.net/manual/en/uconverter.getsourceencoding.php>
- <https://www.php.net/manual/en/uconverter.getsourcetype.php>
- <https://www.php.net/manual/en/uconverter.getstandards.php>
- <https://www.php.net/manual/en/uconverter.getsubstchars.php>
- <https://www.php.net/manual/en/uconverter.reasontext.php>
- <https://www.php.net/manual/en/uconverter.setdestinationencoding.php>
- <https://www.php.net/manual/en/uconverter.setsourceencoding.php>
- <https://www.php.net/manual/en/uconverter.setsubstchars.php>
- <https://www.php.net/manual/en/uconverter.toucallback.php>
- <https://www.php.net/manual/en/uconverter.transcode.php>
- <https://www.php.net/manual/en/class.spoofchecker.php>
- <https://www.php.net/manual/en/spoofchecker.construct.php>
- <https://www.php.net/manual/en/spoofchecker.setallowedlocales.php>
- <https://www.php.net/manual/en/spoofchecker.setchecks.php>
- <https://www.php.net/manual/en/spoofchecker.setrestrictionlevel.php>

4.4. User comments

Quarkslab was commissioned by the PHP Foundation to review the user comments associated with the most frequently commented page on the PHP manual, specifically the `curl_setopt` function documentation - https://php.net/manual/en/function.curl_setopt.php.

Warning

Quarkslab emphasizes that some of these user comments are quite old (dating back to over 20 years). Given the significant changes in the PHP language and its ecosystem since then, certain potential issues highlighted may no longer be applicable.

Retest 09/2025

Success

The PHP maintainers took into account all of Quarkslab's auditors recommendations and deleted the vulnerable comments.

4.4.1. Comment 110457

URL: https://www.php.net/manual/en/function.curl_setopt.php#110457

Success

No security issues were identified in this comment.

4.4.2. Comment 118967

URL: https://www.php.net/manual/en/function.curl_setopt.php#118967

Success

No security issues were identified in this comment.

4.4.3. Comment 75525

URL: https://www.php.net/manual/en/function.curl_setopt.php#75525

Code snippet:

```
1 <?php
2 if ($cookies != '')
3     {
```

PHP

```
4     if (substr(PHP_OS, 0, 3) == 'WIN')
5         {$cookies = str_replace('\\', '/', getcwd()).'/'.$cookies);}
6     curl_setopt($ch, CURLOPT_COOKIEJAR, $cookies);
7     curl_setopt($ch, CURLOPT_COOKIEFILE, $cookies);
8     }
9 ?>
```

INFO-68	Loose comparison
Perimeter	https://www.php.net/manual/en/function.curl_setopt.php#75525
Description	
The code uses loose comparison (!=) when checking if \$cookies is not an empty string. It's better to use strict comparison (!==) for strings.	
Recommendation	
Use !== for comparing the \$cookies variable with an empty string.	

Retest 09/2025

Success

The issue is correctly fixed (comment deleted).

INFO-69	Potential path traversal
Perimeter	https://www.php.net/manual/en/function.curl_setopt.php#75525
Description	
There is a potential risk of file path traversal if the \$cookies variable is user-controlled. A proof-of-concept is provided below.	
Recommendation	
Clearly state that the \$cookies variable should not be user-controlled to prevent path traversal attacks	

The toy example below showcases how a user can use the code provided in the comment to read the cookies of another user.

```
1 // cookies.php
2 $user = $_GET["user"];
3 $cookies = $user . "/cookies.txt";
4
5 if ($cookies !== '')
6 {
7     if (substr(PHP_OS, 0, 3) == 'WIN')
8
9         {$cookies = str_replace('\\', '/', getcwd().'/'.$cookies);}
10
11     curl_setopt($ch, CURLOPT_COOKIEJAR, $cookies);
12     curl_setopt($ch, CURLOPT_COOKIEFILE, $cookies);
13 }
```

```
1 # legitimate use -> stores in the intended directory
2 $ curl 'http://localhost:1337/cookies.php?user=koko'
3 # path traversal
4 $ curl 'http://localhost:1337/cookies.php?user=../'
```

Retest 09/2025

Success

The issue is correctly fixed (comment deleted).

4.4.4. Comment 104597

URL: <https://www.php.net/manual/en/function.curl-setopt.php#104597>

Success

No security issues were identified in this comment.

4.4.5. Comment 125490

URL: <https://www.php.net/manual/en/function.curl-setopt.php#125490>

Success

No security issues were identified in this comment.

4.4.6. Comment 66832

URL: <https://www.php.net/manual/en/function.curl-setopt.php#66832>

Success

No security issues were identified in this comment.

4.4.7. Comment 98491

URL: <https://www.php.net/manual/en/function.curl-setopt.php#98491>

Success

No security issues were identified in this comment.

4.4.8. Comment 121874

URL: <https://www.php.net/manual/en/function.curl-setopt.php#121874>

Success

No security issues were identified in this comment.

4.4.9. Comment 82418

URL: <https://www.php.net/manual/en/function.curl-setopt.php#82418>

Success

No security issues were identified in this comment.

4.4.10. Comment 108137

URL: <https://www.php.net/manual/en/function.curl-setopt.php#108137>

Success

No security issues were identified in this comment.

4.4.11. Comment 96056

URL: <https://www.php.net/manual/en/function.curl-setopt.php#96056>

Success

No security issues were identified in this comment.

4.4.12. Comment 61395

URL: <https://www.php.net/manual/en/function.curl-setopt.php#61395>

Success

No security issues were identified in this comment.

4.4.13. Comment 93686

URL: <https://www.php.net/manual/en/function.curl-setopt.php#93686>

Success

No security issues were identified in this comment.

4.4.14. Comment 118586

URL: <https://www.php.net/manual/en/function.curl-setopt.php#118586>

Success

No security issues were identified in this comment.

4.4.15. Comment 113682

URL: <https://www.php.net/manual/en/function.curl-setopt.php#113682>

Code snippet:

```
1 function curlExec(/* Array */$curlOptions='', /* Array */
2 $curlHeaders='', /* Array */$postFields='')
3 {
4     $newUrl = '';
5     $maxRedirection = 10;
6     do
7     {
8         if ($maxRedirection<1) die('Error: reached the limit of redirections');
9         $ch = curl_init();
10        if (!empty($curlOptions)) curl_setopt_array($ch, $curlOptions);
11        if (!empty($curlHeaders)) curl_setopt($ch, CURLOPT_HTTPHEADER,
12            $curlHeaders);
13        if (!empty($postFields))
14        {
15            curl_setopt($ch, CURLOPT_POST, 1);
16            curl_setopt($ch, CURLOPT_POSTFIELDS, $postFields);
17        }
18        if (!empty($newUrl)) curl_setopt($ch, CURLOPT_URL, $newUrl); // redirect
19        // needed
20        $curlResult = curl_exec($ch);
21        $code = curl_getinfo($ch, CURLINFO_HTTP_CODE);
22
23        if ($code == 301 || $code == 302 || $code == 303 || $code == 307)
24        {
25            preg_match('/Location:(.*?)\n/', $curlResult, $matches);
26            $newUrl = trim(array_pop($matches));
27            curl_close($ch);
28
29            $maxRedirection--;
30            continue;
31        }
32        else // no more redirection
33        {
34            $code = 0;
35            curl_close($ch);
```

```
36     }
37   }
38   while($code);
39   return $curlResult;
40 }
```

INFO-70

Workaround of a security feature could lead to a SSRF and LFI

Perimeter

https://www.php.net/manual/en/function.curl_setopt.php#113682

Description

User suggested workaround could potentially lead to a SSRF (Server-Side Request Forgery) and LFI (Local File Inclusion) (see https://www.php.net/manual/en/function.curl_setopt.php#116223).

Recommendation

Associate a warning to the comment stating that bypassing the limitations of `open_basedir` could have severe security implications such as SSRF and LFI

In the context of a PHP service which parses web pages based on user-supplied URLs, a malicious web page could make use of the vulnerable suggestion to read the contents of `/etc/passwd` by issuing a redirection to `Location: file:///etc/passwd`.

Retest 09/2025

Success

The issue is correctly fixed (comment deleted).

4.4.16. Comment 99082

URL: https://www.php.net/manual/en/function.curl_setopt.php#99082

Success

No security issues were identified in this comment.

4.4.17. Comment 80271

URL: https://www.php.net/manual/en/function.curl_setopt.php#80271

Code snippet:

```
1  <?php
2      $credentials = "username:password";
3      // Read the XML to send to the Web Service
4      $request_file = "./SampleRequest.xml";
5      $fh = fopen($request_file, 'r');
6      $xml_data = fread($fh, filesize($request_file));
7      fclose($fh);
8      $url = "http://www.example.com/services/calculation";
9      $page = "/services/calculation";
10
11     $headers = array(
12         "POST ".$page." HTTP/1.0",
13         "Content-type: text/xml;charset=\"utf-8\"",
14         "Accept: text/xml",
15         "Cache-Control: no-cache",
16         "Pragma: no-cache",
17         "SOAPAction: \"run\"",
18         "Content-length: ".strlen($xml_data),
19         "Authorization: Basic " . base64_encode($credentials)
20     );
21
22     $ch = curl_init();
23     curl_setopt($ch, CURLOPT_URL,$url);
24     curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
25     curl_setopt($ch, CURLOPT_TIMEOUT, 60);
26     curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
27     curl_setopt($ch, CURLOPT_USERAGENT, $defined_vars['HTTP_USER_AGENT']);
28
29     // Apply the XML to our curl call
30     curl_setopt($ch, CURLOPT_POST, 1);
31     curl_setopt($ch, CURLOPT_POSTFIELDS, $xml_data);
32
33     $data = curl_exec($ch);
34
35     if (curl_errno($ch)) {
36         print "Error: " . curl_error($ch);
37     } else {
38         // Show me the result
39         var_dump($data);
40         curl_close($ch);
41     }
```

INFO-71

Hardcoded credentials

Perimeterhttps://www.php.net/manual/en/function.curl_setopt.php#80271**Description**

Use of hardcoded credentials which is bad practice even if the comment is provided as an example.

Recommendation

Issue a warning to not use hardcoded credentials. Use environment variables or secure vaults instead.

Retest 09/2025**Success**

The issue is correctly fixed (comment deleted).

INFO-72

Usage of Basic authentication

Perimeterhttps://www.php.net/manual/en/function.curl_setopt.php#80271**Description**

The code snippet uses basic authentication over HTTP which is considered unsecure

Recommendation

Emit a warning stating that the following example is insecure and should not be used or at least the usage of Basic Authentication should be done over HTTPS

Retest 09/2025**Success**

The issue is correctly fixed (comment deleted).

INFO-73	Lack of XML validation
Perimeter	https://www.php.net/manual/en/function.curl_setopt.php#80271
Description	
The code is prone to XXE (XML External Entity) vulnerability as there is no sanitization of the XML data read from a file.	
Recommendation	
Emit a warning stating the comment is prone to XXE attacks. Urge the users to validate and sanitize the XML data to prevent XXE attacks.	

Retest 09/2025

Success

The issue is correctly fixed (comment deleted).

4.4.18. Comment 116866

URL: https://www.php.net/manual/en/function.curl_setopt.php#116866

Success

No security issues were identified in this comment.

4.4.19. Comment 123340

URL: https://www.php.net/manual/en/function.curl_setopt.php#123340

Success

No security issues were identified in this comment.

4.4.20. Comment 59993

URL: https://www.php.net/manual/en/function.curl_setopt.php#59993

Code snippet:

```
1 <?php PHP
```

```

2 function data_encode($data, $keyprefix = "", $keypostfix = "") {
3     assert( is_array($data) );
4     $vars=null;
5     foreach($data as $key=>$value) {
6         if(is_array($value)) $vars .= data_encode($value, $keyprefix.$key.
7             $keypostfix.urlencode("["), urlencode("]"));
8         else $vars .= $keyprefix.$key.$keypostfix."=".urlencode($value)."&";
9     }
10    return $vars;
11 }
12 curl_setopt($ch, CURLOPT_POSTFIELDS, substr(data_encode($_POST), 0, -1) );
13
14 ?>

```

INFO-74

Unbounded recursion

Perimeter

<https://www.php.net/manual/en/function.curl-setopt.php#59993>

Description

Deeply nested or circular data structures can cause stack overflow errors in the user-supplied comment.

Recommendation

Emit a warning stating the comment does not define a recursion limit and that it is a good practice to define a depth limit when working with deeply nested structures to prevent unbounded recursion and potential stack overflow error

Retest 09/2025

Success

The issue is correctly fixed (comment deleted).

4.4.21. Comment 87341

URL: <https://www.php.net/manual/en/function.curl-setopt.php#87341>

Success

No security issues were identified in this comment.

4.4.22. Comment 110162

URL: https://www.php.net/manual/en/function.curl_setopt.php#110162

Success

No security issues were identified in this comment.

4.4.23. Comment 95027

URL: https://www.php.net/manual/en/function.curl_setopt.php#95027

Code snippet:

```
1 <?php
2 function curl_redirect_exec($ch, &$redirects, $curl_opt_header = false) {
3     curl_setopt($ch, CURLOPT_HEADER, true);
4     curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
5     $data = curl_exec($ch);
6     $http_code = curl_getinfo($ch, CURLINFO_HTTP_CODE);
7     if ($http_code == 301 || $http_code == 302) {
8         list($header) = explode("\r\n\r\n", $data, 2);
9         $matches = array();
10        preg_match('/(Location:|URI:)(.*?)\n/', $header, $matches);
11        $url = trim(array_pop($matches));
12        $url_parsed = parse_url($url);
13        if (isset($url_parsed)) {
14            curl_setopt($ch, CURLOPT_URL, $url);
15            $redirects++;
16            return curl_redirect_exec($ch, $redirects);
17        }
18    }
19    if ($curl_opt_header)
20        return $data;
21    else {
22        list(,$body) = explode("\r\n\r\n", $data, 2);
23        return $body;
24    }
25 }
26 ?>
```

INFO-75	Workaround of a security feature could lead to a SSRF and LFI
Perimeter	https://www.php.net/manual/en/function.curl-setopt.php#95027
Description	
User suggested workaround could potentially lead to a SSRF (Server-Side Request Forgery) and LFI (Local File Inclusion) (see https://www.php.net/manual/en/function.curl-setopt.php#116223).	
Recommendation	
Associate a warning to the comment stating that bypassing the limitations of <code>open_basedir</code> could have severe security implications such as SSRF and LFI.	

Retest 09/2025

Success

The issue is correctly fixed (comment deleted).

4.4.24. Comment 108333

URL: <https://www.php.net/manual/en/function.curl-setopt.php#108333>

Success

No security issues were identified in this comment.

4.4.25. Comment 96903

URL: <https://www.php.net/manual/en/function.curl-setopt.php#96903>

Success

No security issues were identified in this comment.

4.4.26. Comment 60142

URL: <https://www.php.net/manual/en/function.curl-setopt.php#60142>

Success

No security issues were identified in this comment.

4.4.27. Comment 78046

URL: <https://www.php.net/manual/en/function.curl-setopt.php#78046>

Success

No security issues were identified in this comment.

4.4.28. Comment 116223

URL: <https://www.php.net/manual/en/function.curl-setopt.php#116223>

Success

No security issues were identified in this comment.

4.4.29. Comment 64447

URL: <https://www.php.net/manual/en/function.curl-setopt.php#64447>

Success

No security issues were identified in this comment.

4.4.30. Comment 103232

URL: <https://www.php.net/manual/en/function.curl-setopt.php#103232>

Success

No security issues were identified in this comment.

4.4.31. Comment 102519

URL: <https://www.php.net/manual/en/function.curl-setopt.php#102519>

Success

No security issues were identified in this comment.

4.4.32. Comment 121973

URL: <https://www.php.net/manual/en/function.curl-setopt.php#121973>

Code snippet:

```
1 <?php
2     $ch = curl_init();
3     $filepath = 'D:\Web\www\wuxiancheng.cn\hello.txt';
4     $basename = pathInfo($filepath, PATHINFO_BASENAME);
5     $filesize = fileSize($filepath);
6     curl_setopt_array(
7         $ch,
8         array(
9             CURLOPT_URL => 'ftp://www.wuxiancheng.cn/text/' . $basename,
10            CURLOPT_USERPWD => 'USERNAME:PASSWORD',
11            CURLOPT_PROTOCOLS => CURLPROTO_FTP,
12            CURLOPT_UPLOAD => true,
13            CURLOPT_INFILE => $filepath,
14            CURLOPT_INFILESIZE => $filesize,
15            CURLOPT_RETURNTRANSFER => true,
16            CURLOPT_HEADER => false,
17        )
18    );
19    curl_exec($ch);
20    $message = curl_errno($ch) === CURLE_OK ? 'success' : 'failure';
21    echo $message;
22 ?>
```

INFO-76	Lack of secure communications
Perimeter	https://www.php.net/manual/en/function.curl-setopt.php#121973
Description	
Usage of FTP which transfers data in clear.	
Recommendation	
Emit a warning that it is recommended to use FTPS or other secure alternatives instead of plain FTP, as it transfers data in clear text and can be intercepted.	

Retest 09/2025

Success

The issue is correctly fixed (comment deleted).

4.4.33. Comment 95027

URL: https://www.php.net/manual/en/function.curl_setopt.php#95027

Code snippet:

```
1 <?php
2 function curl_redirect_exec($ch, &$redirects, $curl_opt_header = false) {
3     curl_setopt($ch, CURLOPT_HEADER, true);
4     curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
5     $data = curl_exec($ch);
6     $http_code = curl_getinfo($ch, CURLINFO_HTTP_CODE);
7     if ($http_code == 301 || $http_code == 302) {
8         list($header) = explode("\r\n\r\n", $data, 2);
9         $matches = array();
10        preg_match('/(Location:|URI:)(.*?)\n/', $header, $matches);
11        $url = trim(array_pop($matches));
12        $url_parsed = parse_url($url);
13        if (isset($url_parsed)) {
14            curl_setopt($ch, CURLOPT_URL, $url);
15            $redirects++;
16            return curl_redirect_exec($ch, $redirects);
17        }
18    }
19    if ($curl_opt_header)
20        return $data;
21    else {
22        list(,$body) = explode("\r\n\r\n", $data, 2);
23        return $body;
24    }
25 }
26 ?>
```

INFO-77

Workaround of a security feature could lead to a SSRF and LFI

Perimeter

https://www.php.net/manual/en/function.curl_setopt.php#95027

Description

User suggested workaround could potentially lead to a SSRF (Server-Side Request Forgery) and LFI (Local File Inclusion) (see <https://www.php.net/manual/en/function.curl-setopt.php#116223>).

Recommendation

Associate a warning to the comment stating that bypassing the limitations of `open_basedir` could have severe security implications such as SSRF and LFI.

Retest 09/2025

Success

The issue is correctly fixed (comment deleted).

INFO-78

Unbounded recursion

Perimeter

<https://www.php.net/manual/en/function.curl-setopt.php#95027>

Description

The snippet uses a recursive function but does not define its recursion limit

Recommendation

Emit a warning saying that this code snippet could result in a stack overflow due to unbound recursion

Retest 09/2025

Success

The issue is correctly fixed (comment deleted).

4.4.34. Comment 110442

URL: <https://www.php.net/manual/en/function.curl-setopt.php#110442>

Success

No security issues were identified in this comment.

4.4.35. Comment 87112

URL: https://www.php.net/manual/en/function.curl_setopt.php#87112

Success

No security issues were identified in this comment.

4.4.36. Comment 95027

URL: https://www.php.net/manual/en/function.curl_setopt.php#95027

Success

No security issues were identified in this comment.

4.4.37. Comment 102121

URL: https://www.php.net/manual/en/function.curl_setopt.php#102121

Code snippet:

```
1 <?php
2 function curl_exec_follow(/*resource*/ $ch, /*int*/ &$maxredirect = null) {
3     $mr = $maxredirect === null ? 5 : intval($maxredirect);
4     if (ini_get('open_basedir') == '' && ini_get('safe_mode' == 'Off')) {
5         curl_setopt($ch, CURLOPT_FOLLOWLOCATION, $mr > 0);
6         curl_setopt($ch, CURLOPT_MAXREDIRS, $mr);
7     } else {
8         curl_setopt($ch, CURLOPT_FOLLOWLOCATION, false);
9         if ($mr > 0) {
10            $newurl = curl_getinfo($ch, CURLINFO_EFFECTIVE_URL);
11            $rch = curl_copy_handle($ch);
12            curl_setopt($rch, CURLOPT_HEADER, true);
13            curl_setopt($rch, CURLOPT_NOBODY, true);
14            curl_setopt($rch, CURLOPT_FORBID_REUSE, false);
15            curl_setopt($rch, CURLOPT_RETURNTRANSFER, true);
16            do {
17                curl_setopt($rch, CURLOPT_URL, $newurl);
18                $header = curl_exec($rch);
```

```

19         if (curl_errno($rch)) {
20             $code = 0;
21         } else {
22             $code = curl_getinfo($rch, CURLINFO_HTTP_CODE);
23             if ($code == 301 || $code == 302) {
24                 preg_match('/Location:(.*?)\n/', $header, $matches);
25                 $newurl = trim(array_pop($matches));
26             } else {
27                 $code = 0;
28             }
29         }
30     } while ($code && --$mr);
31     curl_close($rch);
32     if (!$mr) {
33         if ($maxredirect === null) {
34             trigger_error('Too many redirects. When following redirects,
35                 libcurl hit the maximum amount.', E_USER_WARNING);
36         } else {
37             $maxredirect = 0;
38         }
39         return false;
40     }
41     curl_setopt($ch, CURLOPT_URL, $newurl);
42 }
43 return curl_exec($ch);
44 }
45 ?>

```

INFO-79

Workaround of a security feature could lead to a SSRF and LFI

Perimeter

https://www.php.net/manual/en/function.curl_setopt.php#102121

Description

User suggested workaround could potentially lead to a SSRF (Server-Side Request Forgery) and LFI (Local File Inclusion) (see https://www.php.net/manual/en/function.curl_setopt.php#116223).

Recommendation

Associate a warning to the comment stating that bypassing the limitations of `open_basedir` could have severe security implications such as SSRF and LFI.

Retest 09/2025

Success

The issue is correctly fixed (comment deleted).

4.4.38. Comment 107146

Success

No security issues were identified in this comment.

4.4.39. Comment 81581

URL: [https://www.php.net/manual/en/function curl-setopt.php#81581](https://www.php.net/manual/en/function	curl-setopt.php#81581)

Success

No security issues were identified in this comment.

4.4.40. Comment 84916

URL: [https://www.php.net/manual/en/function curl-setopt.php#84916](https://www.php.net/manual/en/function	curl-setopt.php#84916)

Success

No security issues were identified in this comment.

4.4.41. Comment 101139

URL: [https://www.php.net/manual/en/function curl-setopt.php#101139](https://www.php.net/manual/en/function	curl-setopt.php#101139)

Success

No security issues were identified in this comment.

4.4.42. Comment 93108

URL: [https://www.php.net/manual/en/function curl-setopt.php#93108](https://www.php.net/manual/en/function	curl-setopt.php#93108)

Success

No security issues were identified in this comment.

4.4.43. Comment 42201

URL: <https://www.php.net/manual/en/function.curl-setopt.php#42201>

Success

No security issues were identified in this comment.

4.4.44. Comment 71313

URL: <https://www.php.net/manual/en/function.curl-setopt.php#71313>

Code snippet:

```
1 <?php
2     function curl_redir_exec($ch)
3     {
4         static $curl_loops = 0;
5         static $curl_max_loops = 20;
6         if ($curl_loops++ >= $curl_max_loops)
7         {
8             $curl_loops = 0;
9             return FALSE;
10        }
11        curl_setopt($ch, CURLOPT_HEADER, true);
12        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
13        $data = curl_exec($ch);
14        list($header, $data) = explode("\n\n", $data, 2);
15        $http_code = curl_getinfo($ch, CURLINFO_HTTP_CODE);
16        if ($http_code == 301 || $http_code == 302)
17        {
18            $matches = array();
19            preg_match('/Location:(.*?)\n/', $header, $matches);
20            $url = @parse_url(trim(array_pop($matches)));
21            if (!$url)
22            {
23                //couldn't process the url to redirect to
24                $curl_loops = 0;
25                return $data;
```

```
26     }
27     $last_url = parse_url(curl_getinfo($ch, CURLINFO_EFFECTIVE_URL));
28     if (!$url['scheme'])
29         $url['scheme'] = $last_url['scheme'];
30     if (!$url['host'])
31         $url['host'] = $last_url['host'];
32     if (!$url['path'])
33         $url['path'] = $last_url['path'];
34     $new_url = $url['scheme'] . '://' . $url['host'] . $url['path'] .
35         ($url['query']?'?'.$url['query']:'');
36     curl_setopt($ch, CURLOPT_URL, $new_url);
37     debug('Redirecting to', $new_url);
38     return curl_exec($ch);
39 } else {
40     $curl_loops=0;
41     return $data;
42 }
43 ?>
```

INFO-80	Workaround of a security feature could lead to a SSRF and LFI
Perimeter	https://www.php.net/manual/en/function.curl_setopt.php#71313
Description	
User suggested workaround could potentially lead to a SSRF (Server-Side Request Forgery) and LFI (Local File Inclusion) (see https://www.php.net/manual/en/function.curl_setopt.php#116223).	
Recommendation	
Associate a warning to the comment stating that bypassing the limitations of <code>open_basedir</code> could have severe security implications such as SSRF and LFI.	

Retest 09/2025

Success

The issue is correctly fixed (comment deleted).

4.4.45. Comment 108162

URL: <https://www.php.net/manual/en/function.curl-setopt.php#108162>

Success

No security issues were identified in this comment.

4.4.46. Comment 53445

URL: <https://www.php.net/manual/en/function.curl-setopt.php#53445>

Success

No security issues were identified in this comment.

4.4.47. Comment 69115

URL: <https://www.php.net/manual/en/function.curl-setopt.php#69115>

Success

No security issues were identified in this comment.

4.4.48. Comment 122274

URL: <https://www.php.net/manual/en/function.curl-setopt.php#122274>

Success

No security issues were identified in this comment.

4.4.49. Comment 83937

URL: <https://www.php.net/manual/en/function.curl-setopt.php#83937>

Success

No security issues were identified in this comment.

4.4.50. Comment 83937

URL: <https://www.php.net/manual/en/function.curl-setopt.php#83937>

Success

No security issues were identified in this comment.

4.4.51. Comment 102090

URL: <https://www.php.net/manual/en/function.curl-setopt.php#102090>

Success

No security issues were identified in this comment.

4.4.52. Comment 91510

URL: <https://www.php.net/manual/en/function.curl-setopt.php#91510>

Success

No security issues were identified in this comment.

4.4.53. Comment 112326

URL: <https://www.php.net/manual/en/function.curl-setopt.php#112326>

Success

No security issues were identified in this comment.

4.4.54. Comment 99082

URL: <https://www.php.net/manual/en/function.curl-setopt.php#99082>

Success

No security issues were identified in this comment.

4.4.55. Comment 84374

URL: <https://www.php.net/manual/en/function.curl-setopt.php#84374>

Code snippet:

```
1 <?php
2 curl_setopt($ch, CURLOPT_VERBOSE, '1');
3 curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, '2');
```

PHP

```
4 curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, '1');
5 curl_setopt($ch, CURLOPT_CAINFO, getcwd().'/cert/ca.crt');
6 curl_setopt($ch, CURLOPT_SSLCERT, getcwd().'/cert/mycert.pem');
7 curl_setopt($ch, CURLOPT_SSLCERTPASSWD, 'password');
8 ?>
```

INFO-81

Hardcoded credentials

Perimeter

<https://www.php.net/manual/en/function.curl-setopt.php#84374>

Description

Use of hardcoded credentials which is bad practice even if the comment is provided as an example.

Recommendation

Issue a warning to not use hardcoded credentials. Use environment variables or secure vaults instead.

Retest 09/2025

Success

The issue is correctly fixed (comment deleted).

4.4.56. Comment 112113

URL: <https://www.php.net/manual/en/function.curl-setopt.php#112113>

Success

No security issues were identified in this comment.

4.4.57. Comment 92522

URL: <https://www.php.net/manual/en/function.curl-setopt.php#92522>

Success

No security issues were identified in this comment.

4.4.58. Comment 97206

URL: <https://www.php.net/manual/en/function.curl-setopt.php#97206>

Success

No security issues were identified in this comment.

4.4.59. Comment 109319

URL: <https://www.php.net/manual/en/function.curl-setopt.php#109319>

Success

No security issues were identified in this comment.

4.4.60. Comment 109319

URL: <https://www.php.net/manual/en/function.curl-setopt.php#109319>

Success

No security issues were identified in this comment.

4.4.61. Comment 128365

URL: <https://www.php.net/manual/en/function.curl-setopt.php#128365>

Success

No security issues were identified in this comment.

4.4.62. Comment 128774

URL: <https://www.php.net/manual/en/function.curl-setopt.php#128774>

Success

No security issues were identified in this comment.

5. Conclusions

Quarkslab performed a security review of several PHP documentation pages. Furthermore, their focus was on user comments from the most extensively commented pages.

The review illustrates that overall:

- Code quality best practices are not consistently enforced (e.g., error handling, input validation), which could pose an issue if the snippet is copied into an implementation.
- Documentation for deprecated functions is not always updated.
- User comments should not be trusted by default, especially when they are quite old and the project has evolved significantly since then.

Warning

Quarkslab strongly recommends applying all recommendations provided throughout all the code snippets in PHP documentation.

Retest 09/2025

Info

The PHP maintainers have incorporated Quarkslab's audit recommendations, resulting in improved overall quality of the PHP documentation. We nevertheless recommend rendering the [appendix](#) introduced in pull request 4797, thereby promoting best practices in error handling and input validation.

Finally, Quarkslab would like to thank the OSTIF and PHP Foundation for this collaboration.