About the GitHub Advisory database

In this article

The GitHub Advisory Database contains a list of known security vulnerabilities and malware, grouped in two categories: GitHub-reviewed advisories and unreviewed advisories.

About the GitHub Advisory Database

We add advisories to the GitHub Advisory Database from the following sources:

- Security advisories reported on GitHub
- The National Vulnerability database
- The npm Security advisories database
- The FriendsOfPHP database
- The Go Vulncheck database
- The Python Packaging Advisory database
- The Ruby Advisory database
- The RustSec Advisory database
- Community contributions. For more information, see https://github.com/github/advisory-database/pulls.

If you know of another database we should be importing advisories from, tell us about it by opening an issue in https://github.com/github/advisory-database.

Security advisories are published as JSON files in the Open Source Vulnerability (OSV) format. For more information about the OSV format, see "Open Source Vulnerability format."

About types of security advisories

Note: Advisories for malware are currently in beta and subject to change.

Each advisory in the GitHub Advisory Database is for a vulnerability in open source projects or for malicious open source software.

A vulnerability is a problem in a project's code that could be exploited to damage the confidentiality, integrity, or availability of the project or other projects that use its code. Vulnerabilities vary in type, severity, and method of attack. Vulnerabilities in code are usually introduced by accident and fixed soon after they are discovered. You should update your code to use the fixed version of the dependency as soon as it is available.

In contrast, malicious software, or malware, is code that is intentionally designed to perform unwanted or harmful functions. The malware may target hardware, software, confidential data, or users of any application that uses the malware. You need to remove the malware from your project and find an alternative, more secure replacement for the dependency.

GitHub-reviewed advisories

GitHub-reviewed advisories are security vulnerabilities or malware that have been mapped to packages in ecosystems we support. We carefully review each advisory for validity and ensure that they have a full description, and contain both ecosystem and package information.

Generally, we name our supported ecosystems after the software programming language's associated package registry. We review advisories if they are for a vulnerability in a package that comes from a supported registry.

- Composer (registry: https://packagist.org/)
- Erlang (registry: https://hex.pm/)
- Go (registry: https://pkg.go.dev/)
- GitHub Actions (https://github.com/marketplace?type=actions/)
- Maven (registry: https://repo.maven.apache.org/maven2)
- npm (registry: https://www.npmjs.com/)
- NuGet (registry: https://www.nuget.org/)
- pip (registry: https://pypi.org/)
- pub (registry: https://pub.dev/packages/registry)
- RubyGems (registry: https://rubygems.org/)
- Rust (registry: https://crates.io/)
- Swift (registry: N/A)

If you have a suggestion for a new ecosystem we should support, please open an issue for discussion.

If you enable Dependabot alerts for your repositories, you are automatically notified when a new GitHub-reviewed advisory reports a vulnerability or malware for a package you depend on. For more information, see "About Dependabot alerts."

Unreviewed advisories

Unreviewed advisories are security vulnerabilities that we publish automatically into the GitHub Advisory Database, directly from the National Vulnerability Database feed.

Dependabot doesn't create Dependabot alerts for unreviewed advisories as this type of advisory isn't checked for validity or completion.

About information in security advisories

In this section, you can find more detailed information about security advisories in the GitHub Advisory Database, such as:

- Advisory IDs and what format these identifiers use.
- The CVSS levels we used to assign severity levels.

About GHSA IDs

Each security advisory, regardless of its type, has a unique identifier referred to as a GHSA ID. A `GHSA-ID` qualifier is assigned when a new advisory is created on GitHub.com or added to the GitHub Advisory Database from any of the supported sources.

The syntax of GHSA IDs follows this format: `GHSA-xxxx-xxxx-xxxx` where:

- `x` is a letter or a number from the following set: `23456789cfghjmpqrvwx`.
- Outside the `GHSA` portion of the name:
  - The numbers and letters are randomly assigned.
  - All letters are lowercase.

You can validate a GHSA ID using a regular expression.

Bash

```
/GHSA(-[23456789cfghjmpqrvwx]{4}){3}/
```

About CVSS levels

Each security advisory contains information about the vulnerability or malware, which may include the description, severity, affected package, package ecosystem, affected versions and patched versions, impact, and optional information such as references, workarounds, and credits. In addition, advisories from the National Vulnerability Database list contain a link to the CVE record, where you can read more details about the vulnerability, its CVSS scores, and its qualitative severity level. For more information, see the "National Vulnerability Database" from the National Institute of Standards and Technology.

The severity level is one of four possible levels defined in the "Common Vulnerability Scoring System (CVSS), Section 5."

- Low
- Medium/Moderate
- High
- Critical

The GitHub Advisory Database uses the CVSS levels described above. If GitHub obtains a CVE, the GitHub Advisory Database uses CVSS version 3.1. If the CVE is imported, the GitHub Advisory Database supports both CVSS versions 3.0 and 3.1.

You can also join GitHub Security Lab to browse security-related topics and contribute to security tools and projects.

Further reading

- "About Dependabot alerts"
- MITRE's definition of "vulnerability"