Best practices for writing repository security advisories

In this article

When you create or edit security advisories, the information you provide is easier for other users to understand when you specify the ecosystem, package name, and affected versions using the standard formats.

Anyone with admin permissions to a repository can create and edit a security advisory.

Note: If you are a security researcher, you should directly contact maintainers to ask them to create security advisories or issue CVEs on your behalf in repositories that you don't administer. However, if private vulnerabiliy reporting is enabled for the repository, you can *privately* report a vulnerability yourself. For more information, see "Privately reporting a security vulnerability."

About security advisories for repositories

Repository security advisories allow repository maintainers to privately discuss and fix a security vulnerability in a project. After collaborating on a fix, repository maintainers can publish the security advisory to publicly disclose the security vulnerability to the project's community. By publishing security advisories, repository maintainers make it easier for their community to update package dependencies and research the impact of the security vulnerabilities. For more information, see "About repository security advisories."

Best practices

We recommend you use the syntax used in the GitHub Advisory Database, especially the version formatting, when you write a repository security advisory, or make a community contribution to a global security advisory.

If you follow the syntax for the GitHub Advisory Database, especially when you define affected versions:

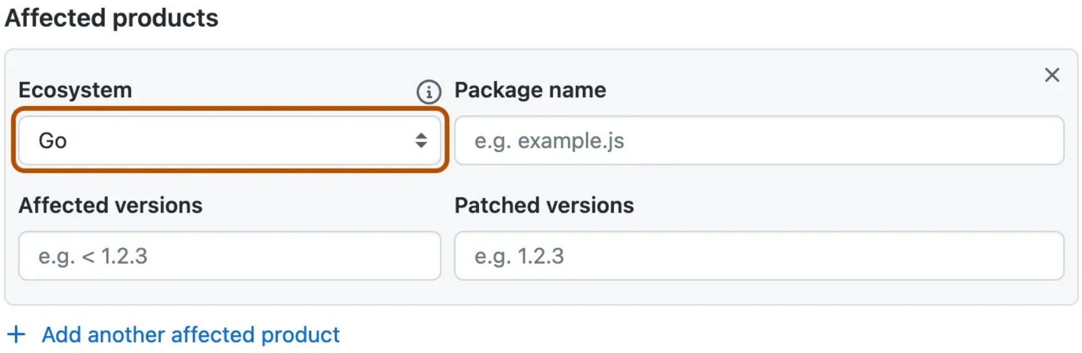
- When you publish your repository advisory, we can add your advisory to the GitHub Advisory Database as a "GitHub-reviewed" advisory, without needing to ask for more information.
- Dependabot will have the information to accurately identify repositories that are affected and send them Dependabot alerts to notify them.
- Community members are less likely to suggest edits to your advisory to fix missing or incorrect information.

You add or edit a repository advisory using the *Draft security advisory* form. For more information, see "Creating a repository security advisory."

You suggest an improvement to an existing global advisory using the *Improve security advisory* form. For more information, see "Editing security advisories in the GitHub Advisory Database."

Ecosystem

You need to assign the advisory to one of our supported ecosystems using the **Ecosystem** field. For more information about the ecosystems we support, see "Browsing security advisories in the GitHub Advisory Database."


Affected products

Ecosystem: Go

Package name: e.g. example.js

Affected versions: e.g. < 1.2.3

Patched versions: e.g. 1.2.3

Package name

We recommend that you use the **Package name** field to specify which packages are affected because package information is required for "GitHub-reviewed" advisories in the GitHub Advisory Database. Package information is optional for repository-level security advisories, but including this information early simplifies the review process when you publish your security advisory.

Affected versions

We recommend that you use the **Affected versions** field to specify which versions are affected because this information is required for "GitHub-reviewed" advisories in the GitHub Advisory Database. Version information is optional for repository-level security advisories, but including this information early simplifies the review process when you publish your security advisory.

- A valid affected version string consists of one of the following:
  - A lower bound operator sequence.
  - An upper bound operator sequence.
  - Both an upper and lower bound operator sequence.
  - A specific version sequence using the equality (`=`) operator.
- Each operator sequence must be specified as the operator, a single space, and then the version.
  - Valid operators are `=`, `<`, `<=`, `>`, or `>=`.
  - The version must begin with a number followed by any number of numbers, letters, dots, dashes, or underscores (anything other than a space or comma)
  - When specifying both an upper and lower bound sequence, the lower bound must come first, followed by a comma and a single space, then the upper bound.

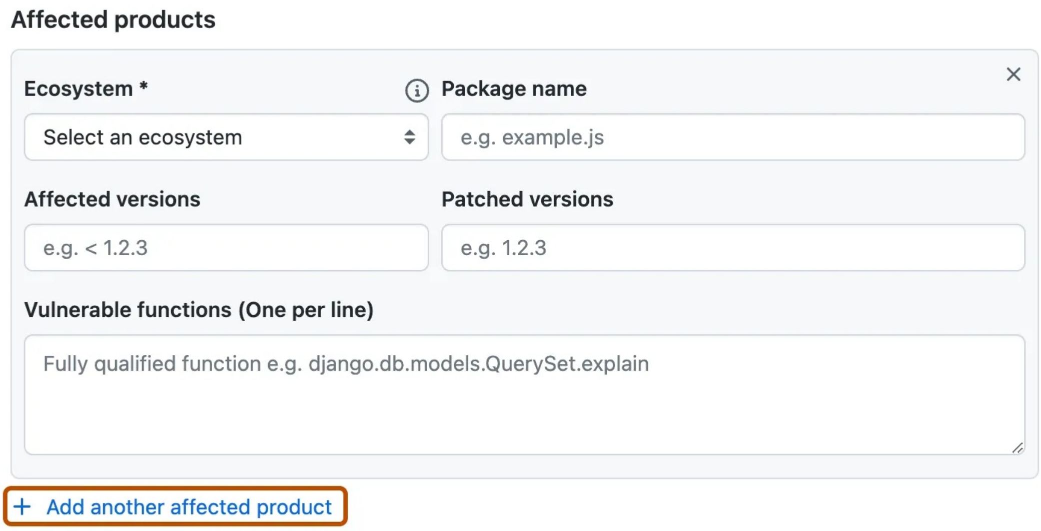
> **Note:** Affected version strings cannot contain leading or trailing spaces.

- Upper-bound operators can be inclusive or exclusive, i.e. `<=` or `<`, respectively.
- Lower-bound operators can be inclusive or exclusive, i.e. `>=` or `>`, respectively.

  However, if you publish your repository advisory, and we graduate your repository advisory into a global advisory, a different rule applies: lower-bound strings can only be inclusive, i.e. `>=`. The exclusive lower bound operator (`>`) is only allowed when the version is `0`, for example `> 0`.

  > **Notes:** The lower-bound limitation:
  >
  > - is due to incompatibilities with the OSV (Open Source Vulnerability) schema.
  > - only applies when you make a suggestion on an existing advisory in the GitHub Advisory Database.

- You cannot specify multiple affected version ranges in the same field, such as `> 2.0, < 2.3, > 3.0, < 3.2`.To specify more than one range, you must create a new **Affected products** section for each range, by clicking the **+ Add another affected product** button.



- If the affected version range includes only a single upper or lower bound:
  - The implicit value is always `> 0` if the lower bound is not explicitly specified.
  - The implicit value is always infinity if the upper bound is not explicitly specified.

For more information about the GitHub Advisory Database, see https://github.com/github/advisory-database.

Legal

Terms Privacy Status Pricing Expert services Blog