Viewing and updating Dependabot alerts

In this article

About updates for vulnerable dependencies in your repository
Prioritizing Dependabot alerts
Supported ecosystems and manifests for dependency scope
About the detection of calls to vulnerable functions
Viewing Dependabot alerts
Reviewing and fixing alerts
Dismissing Dependabot alerts
Viewing and updating closed alerts
Reviewing the audit logs for Dependabot alerts

If GitHub discovers insecure dependencies in your project, you can view details on the Dependabot alerts tab of your repository. Then, you can update your project to resolve or dismiss the alert.

> **Who can use this feature**
> Repository administrators, organization owners, and people with write or maintain access to a repository can view and update dependencies, as well as users and teams with explicit access.

Your repository's Dependabot alerts tab lists all open and closed Dependabot alerts and corresponding Dependabot security updates. You can filter alerts by package, ecosystem, or manifest. You can sort the list of alerts, and you can click into specific alerts for more details. You can also dismiss or reopen alerts, either one by one or by selecting multiple alerts at once. For more information, see "About Dependabot alerts."

You can enable automatic security updates for any repository that uses Dependabot alerts and the dependency graph. For more information, see "About Dependabot security updates."

About updates for vulnerable dependencies in your repository

GitHub generates Dependabot alerts when we detect that the default branch of your codebase is using dependencies with known security risks. For repositories where Dependabot security updates are enabled, when GitHub detects a vulnerable dependency in the default branch, Dependabot creates a pull request to fix it. The pull request will upgrade the dependency to the minimum possible secure version needed to avoid the vulnerability.

Each Dependabot alert has a unique numeric identifier and the Dependabot alerts tab lists an alert for every detected vulnerability. Legacy Dependabot alerts grouped vulnerabilities by dependency and generated a single alert per dependency. If you navigate to a legacy Dependabot alert, you will be redirected to a Dependabot alerts tab filtered for that package.

You can filter and sort Dependabot alerts using a variety of filters and sort options available on the user interface. For more information, see "Prioritizing Dependabot alerts" below.

You can also audit actions taken in response to Dependabot alerts. For more information, see "Auditing security alerts."

Prioritizing Dependabot alerts

GitHub helps you prioritize fixing Dependabot alerts. By default, Dependabot alerts are sorted by importance. The "Most important" sort order helps you prioritize which Dependabot alerts to focus on first. Alerts are ranked based on their potential impact, actionability, and relevance. Our prioritization calculation is constantly being improved and includes factors like CVSS score, dependency scope, and whether vulnerable function calls are found for the alert.

You can also use alert rules to prioritize Dependabot alerts. For more information, see "About Dependabot alert rules."

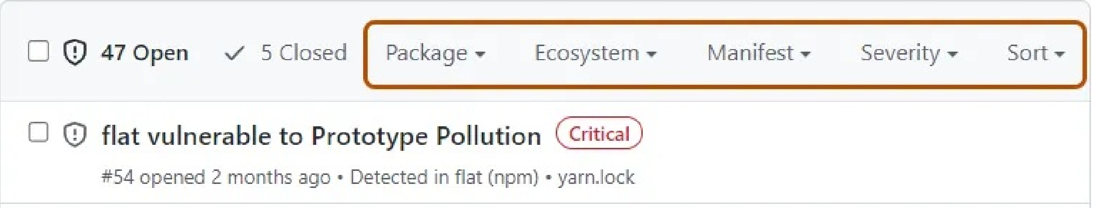
You can sort and filter Dependabot alerts by typing filters as `key:value` pairs into the search bar.

Option	Description	Example
`ecosystem`	Displays alerts for the selected ecosystem	Use `ecosystem:npm` to show Dependabot alerts for npm
`has`	Displays alerts meeting the selected filter criteria	Use `has:patch` to show alerts related to advisories that have a patch Use `has:vulnerable-calls` to show alerts relating to calls to vulnerable functions
`is`	Displays alerts based on their state	Use `is:open` to show open alerts
`manifest`	Displays alerts for the selected manifest	Use `manifest:webwolf/pom.xml` to show alerts on the pom.xml file of the webwolf application
`package`	Displays alerts for the selected package	Use `package:django` to show alerts for django
`resolution`	Displays alerts of the selected resolution status	Use `resolution:no-bandwidth` to show alerts previously parked due to lack of resources or time to fix them
`repo`	Displays alerts based on the repository they relate to Note that this filter is only available for security overview. For more information, see "About security overview"	Use `repo:octocat-repo` to show alerts in the repository called `octocat-repo`
`scope`	Displays alerts based on the scope of the dependency they relate to	Use `scope:development` to show alerts for dependencies that are only used during development

`severity`	Displays alerts based on their level of severity	Use `severity:high` to show alerts with a severity of High
`sort`	Displays alerts according to the selected sort order	The default sorting option for alerts is `sort:most-important`, which ranks alerts by importance Use `sort:newest` to show the latest alerts reported by Dependabot

In addition to the filters available via the search bar, you can sort and filter Dependabot alerts using the dropdown menus at the top of the alert list. Alternatively, to filter by label, click a label assigned to an alert to automatically apply that filter to the alert list.

The search bar also allows for full text searching of alerts and related security advisories. You can search for part of a security advisory name or description to return the alerts in your repository that relate to that security advisory. For example, searching for `yaml.load() API could execute arbitrary code` will return Dependabot alerts linked to "PyYAML insecurely deserializes YAML strings leading to arbitrary code execution" as the search string appears in the advisory description.



flat vulnerable to Prototype Pollution

#54 opened 2 months ago • Detected in flat (npm) • yarn.lock

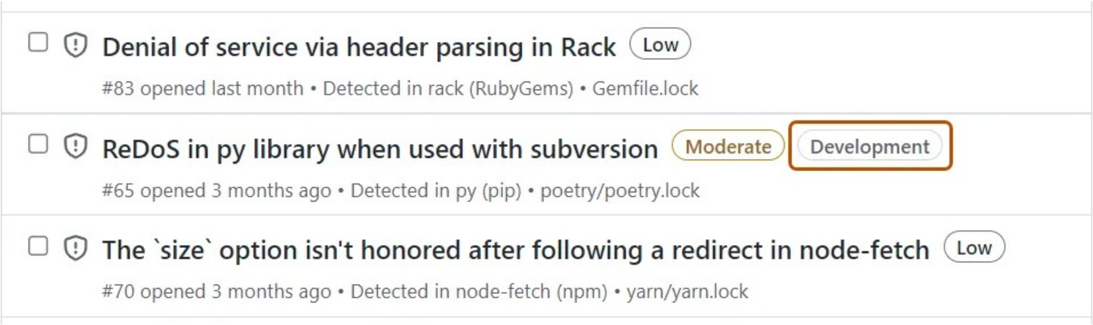
Supported ecosystems and manifests for dependency scope

The table below summarizes whether dependency scope is supported for various ecosystems and manifests, that is, whether Dependabot can identify if a dependency is used for development or production.

Language	Ecosystem	Manifest file	Dependency scope supported
Dart	pub	pubspec.yaml	✓
Dart	pub	pubspec.lock	✓
Go	Go modules	go.mod	No, defaults to runtime
Java	Maven	pom.xml	✓ `test` maps to development, else scope defaults to runtime
JavaScript	npm	package.json	✓
JavaScript	npm	package-lock.json	✓
JavaScript	npm	pnpm-lock.yaml	✓
JavaScript	yarn v1	yarn.lock	No, defaults to runtime
PHP	Composer	composer.json	✓
PHP	Composer	composer.lock	✓

PHP	Composer	composer.lock	✓
Python	Poetry	poetry.lock	✓
Python	Poetry	pyproject.toml	✓
Python	pip	requirements.txt	✓ Scope is development if the filename contains `test` or `dev`, else it is runtime
Python	pip	pipfile.lock	✓
Python	pip	pipfile	✓
Ruby	RubyGems	Gemfile	✓
Ruby	RubyGems	Gemfile.lock	No, defaults to runtime
Rust	Cargo	Cargo.toml	✓
Rust	Cargo	Cargo.lock	No, defaults to runtime
YAML	GitHub Actions	-	No, defaults to runtime
.NET (C#, F#, VB, etc.)	NuGet	.csproj / .vbproj .vcxproj / .fsproj	No, defaults to runtime
.NET	NuGet	packages.config	No, defaults to runtime
.NET	NuGet	.nuspec	✓ When the tag != runtime

Alerts for packages listed as development dependencies are marked with the `Development` label on the Dependabot alerts page and are also available for filtering via the `scope` filter.

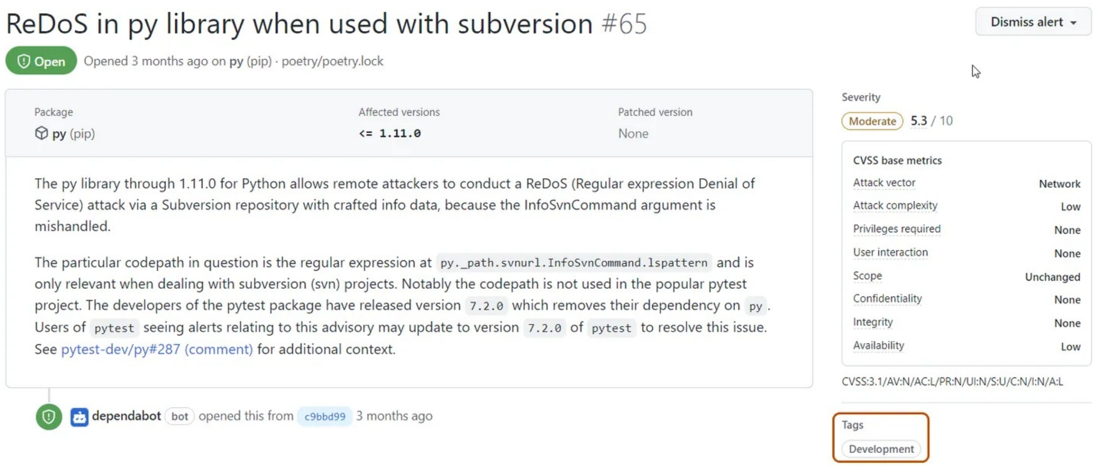


□ Denial of service via header parsing in Rack Low
#83 opened last month • Detected in rack (RubyGems) • Gemfile.lock

□ ReDoS in py library when used with subversion Moderate Development
#65 opened 3 months ago • Detected in py (pip) • poetry/poetry.lock

□ The `size` option isn't honored after following a redirect in node-fetch Low
#70 opened 3 months ago • Detected in node-fetch (npm) • yarn/yarn.lock

The alert details page of alerts on development-scoped packages shows a "Tags" section containing a `Development` label.



ReDoS in py library when used with subversion #65

Opened 3 months ago on **py** (pip) · poetry/poetry.lock

Package	Affected versions	Patched version
py (pip)	`<= 1.11.0`	None

The py library through 1.11.0 for Python allows remote attackers to conduct a ReDoS (Regular expression Denial of Service) attack via a Subversion repository with crafted info data, because the InfoSvnCommand argument is mishandled.

The particular codepath in question is the regular expression at `py._path.svnurl.InfoSvnCommand.lspattern` and is only relevant when dealing with subversion (svn) projects. Notably the codepath is not used in the popular pytest project. The developers of the pytest package have released version `7.2.0` which removes their dependency on `py`. Users of `pytest` seeing alerts relating to this advisory may update to version `7.2.0` of `pytest` to resolve this issue. See pytest-dev/py#287 (comment) for additional context.

dependabot bot opened this from c9bbd99 3 months ago

Severity
Moderate 5.3 / 10

CVSS base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	None
Integrity	None
Availability	Low

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L

Tags
Development

About the detection of calls to vulnerable functions

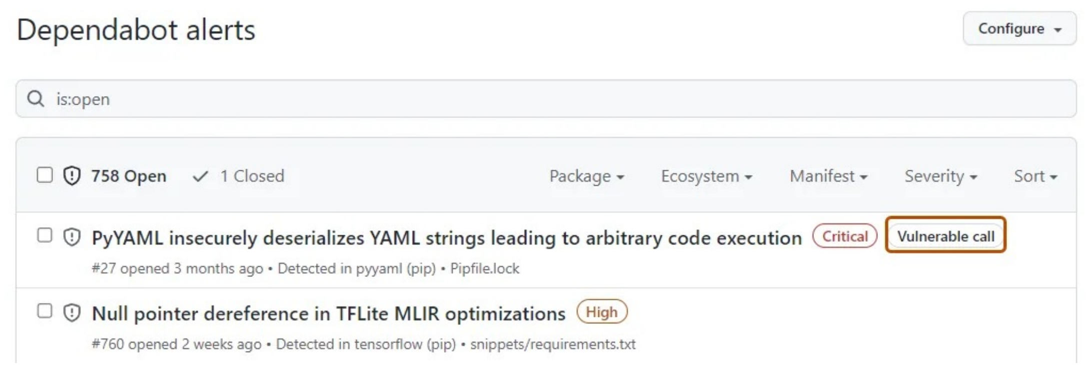
Notes:

- The detection of calls to vulnerable functions by Dependabot is in beta and subject to change.
- Detection of vulnerable calls is enabled on public repositories. This analysis is also available in private repositories owned by organizations that use GitHub Enterprise Cloud and have licensed GitHub Advanced Security. For more information, see "About GitHub Advanced Security."

When Dependabot tells you that your repository uses a vulnerable dependency, you need to determine what the vulnerable functions are and check whether you are using them. Once you have this information, then you can determine how urgently you need to upgrade to a secure version of the dependency.

For supported languages, Dependabot automatically detects whether you use a vulnerable function and adds the label "Vulnerable call" to affected alerts. You can use this information in the Dependabot alerts view to triage and prioritize remediation work more effectively.

Note: During the beta release, this feature is available only for new Python advisories created *after* April 14, 2022, and for a subset of historical Python advisories. GitHub is working to backfill data across additional historical Python advisories, which are added on a rolling basis. Vulnerable calls are highlighted only on the Dependabot alerts pages.



Dependabot alerts

□ 758 Open ✓ 1 Closed

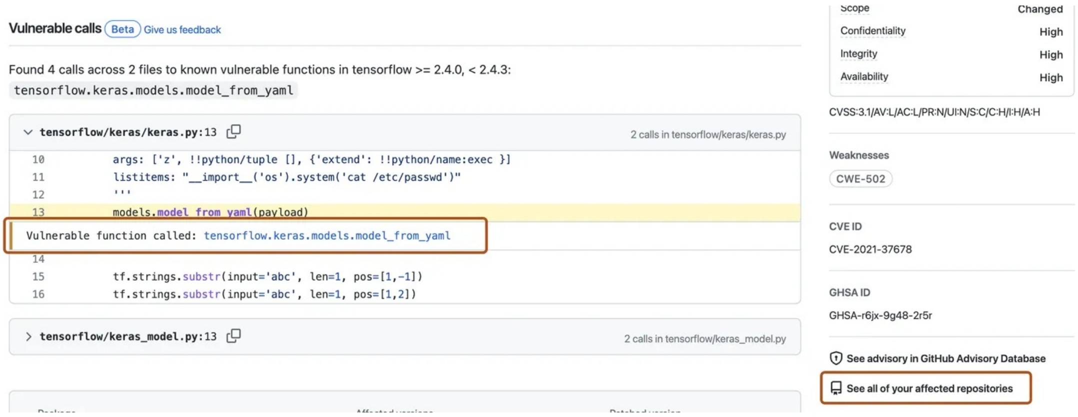
□ PyYAML insecurely deserializes YAML strings leading to arbitrary code execution Critical Vulnerable call
#27 opened 3 months ago • Detected in pyyaml (pip) • Pipfile.lock

□ Null pointer dereference in TFLite MLIR optimizations High
#760 opened 2 weeks ago • Detected in tensorflow (pip) • snippets/requirements.txt

You can filter the view to show only alerts where Dependabot detected at least one call to a vulnerable function using the `has:vulnerable-calls` filter in the search field.

For alerts where vulnerable calls are detected, the alert details page shows additional information:

- One or more code blocks showing where the function is used.
- An annotation listing the function itself, with a link to the line where the function is called.



Vulnerable calls Beta Give us feedback

Found 4 calls across 2 files to known vulnerable functions in tensorflow >= 2.4.0, < 2.4.3: `tensorflow.keras.models.model_from_yaml`

tensorflow/keras/keras.py:13 — 2 calls in tensorflow/keras/keras.py

```
        args: ['z', !!python/tuple [], {'extend': !!python/name:exec }]
        listitems: "__import__('os').system('cat /etc/passwd')"
        '''
        models.model_from_yaml(payload)
Vulnerable function called: tensorflow.keras.models.model_from_yaml

        tf.strings.substr(input='abc', len=1, pos=[1,-1])
        tf.strings.substr(input='abc', len=1, pos=[1,2])
```

tensorflow/keras_model.py:13 — 2 calls in tensorflow/keras_model.py

Scope	Changed
Confidentiality	High
Integrity	High
Availability	High

CVSS:3.1/AV:L/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

Weaknesses

CWE-502

CVE ID

CVE-2021-37678

GHSA ID

GHSA-r6jx-9g48-2r5r

See advisory in GitHub Advisory Database

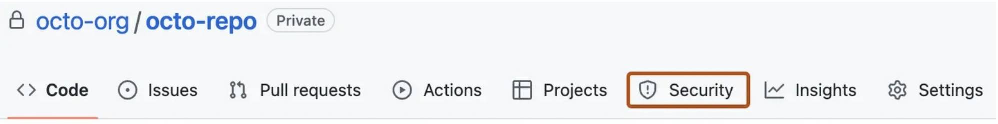
See all of your affected repositories

For more information, see "Reviewing and fixing alerts" below.

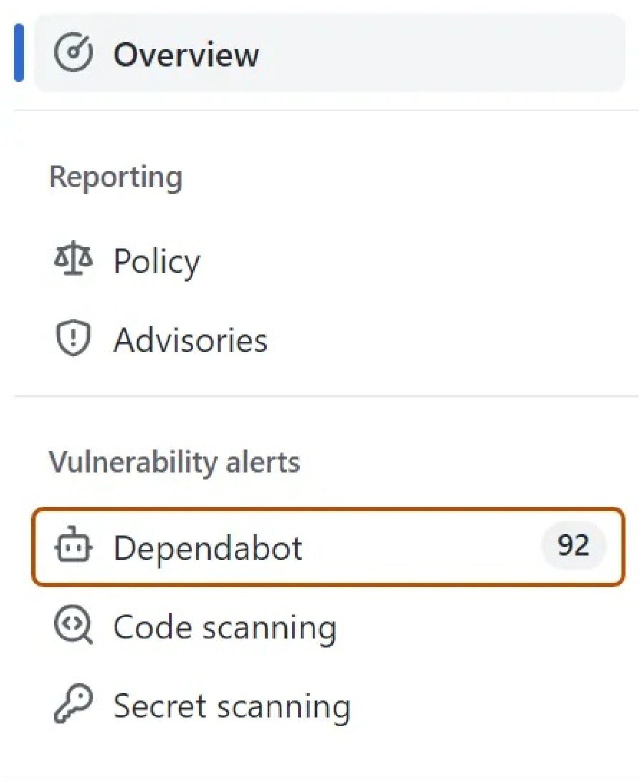
Viewing Dependabot alerts

1. On GitHub.com, navigate to the main page of the repository.

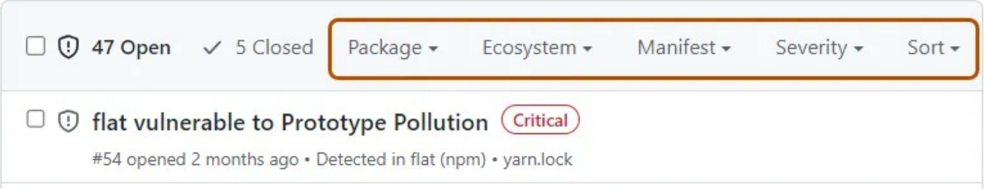
2. Under the repository name, click Security. If you cannot see the "Security" tab, select the ··· dropdown menu, and then click **Security**.



3. In the "Vulnerability alerts" sidebar of security overview, click **Dependabot**. If this option is missing, it means you don't have access to security alerts and need to be given access. For more information, see "Managing security and analysis settings for your repository."



4. Optionally, to filter alerts, select a filter in a dropdown menu then click the filter that you would like to apply. You can also type filters into the search bar. Alternatively, to filter by label, click a label assigned to an alert to automatically apply that filter to the alert list. For more information about filtering and sorting alerts, see "Prioritizing Dependabot alerts."

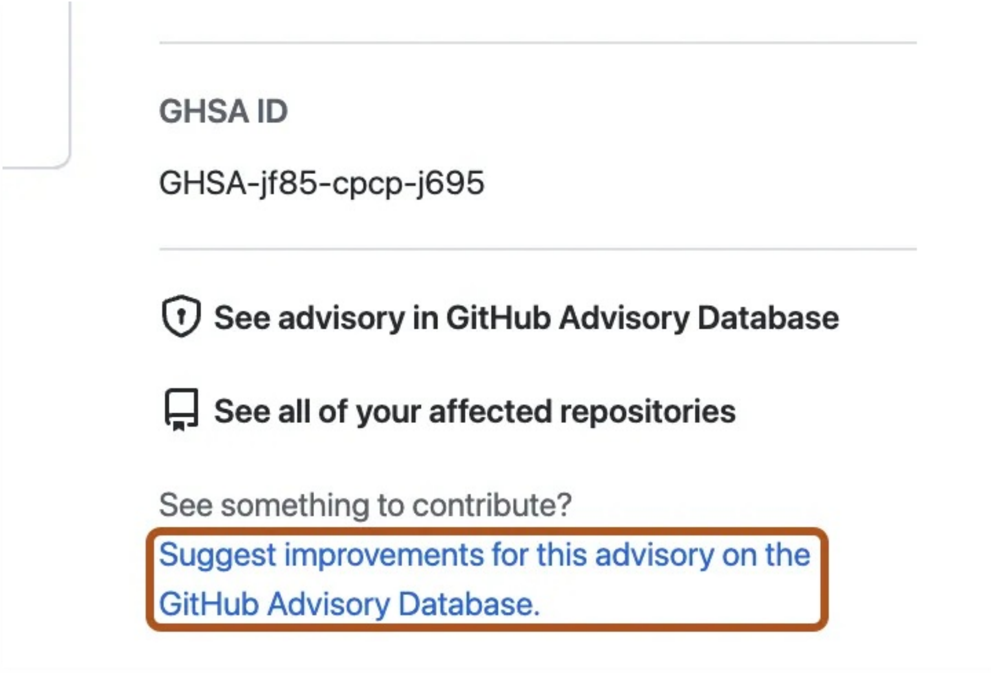


□ flat vulnerable to Prototype Pollution Critical

#54 opened 2 months ago • Detected in flat (npm) • yarn.lock

5. Click the alert that you would like to view.

6. Optionally, to suggest an improvement to the related security advisory, on the right-hand side of the alert details page, click **Suggest improvements for this advisory on the GitHub Advisory Database**. For more information, see "Editing security advisories in the GitHub Advisory Database."



GHSA ID

GHSA-jf85-cpcp-j695

See advisory in GitHub Advisory Database
See all of your affected repositories

See something to contribute?

Suggest improvements for this advisory on the GitHub Advisory Database.

Reviewing and fixing alerts

It’s important to ensure that all of your dependencies are clean of any security weaknesses. When Dependabot discovers vulnerabilities or malware in your dependencies, you should assess your project’s level of exposure and determine what remediation steps to take to secure your application.

If a patched version of the dependency is available, you can generate a Dependabot pull request to update this dependency directly from a Dependabot alert. If you have Dependabot security updates enabled, the pull request may be linked in the Dependabot alert.

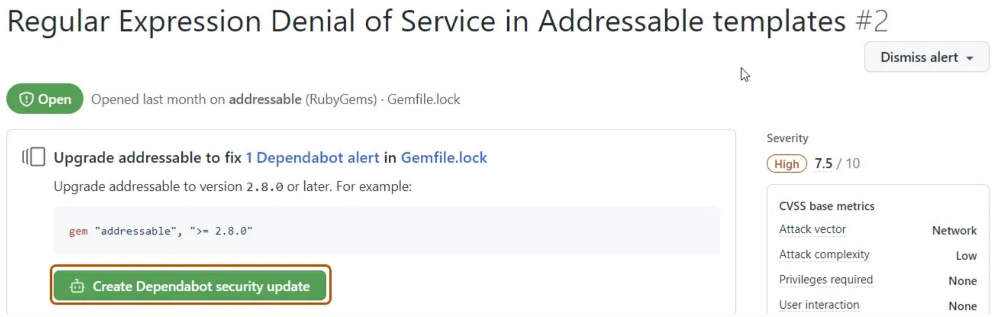
In cases where a patched version is not available, or you can’t update to the secure version, Dependabot shares additional information to help you determine next steps. When you click through to view a Dependabot alert, you can see the full details of the security advisory for the dependency including the affected functions. You can then check whether your code calls the impacted functions. This information can help you further assess your risk level, and determine workarounds or if you're able to accept the risk represented by the security advisory.

For supported languages, Dependabot detects calls to vulnerable functions for you. When you view an alert labeled as "Vulnerable call", the details include the name of the function and a link to the code that calls it. Often you will be able to take decisions based on this information, without exploring further.

Fixing vulnerable dependencies

1. View the details for an alert. For more information, see "Viewing Dependabot alerts" (above).

2. If you have Dependabot security updates enabled, there may be a link to a pull request that will fix the dependency. Alternatively, you can click **Create Dependabot security update** at the top of the alert details page to create a pull request.



Regular Expression Denial of Service in Addressable templates #2

Opened last month on **addressable** (RubyGems) · Gemfile.lock

Upgrade addressable to fix 1 Dependabot alert in Gemfile.lock

Upgrade addressable to version `2.8.0` or later. For example:

```
gem "addressable", ">= 2.8.0"
```

Severity

High 7.5 / 10

CVSS base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	None

3. Optionally, if you do not use Dependabot security updates, you can use the information on the page to decide which version of the dependency to upgrade to and create a pull request to update the dependency to a secure version.

4. When you're ready to update your dependency and resolve the vulnerability, merge the pull request.

   Each pull request raised by Dependabot includes information on commands you can use to control Dependabot. For more information, see "Managing pull requests for dependency updates."

Dismissing Dependabot alerts

Tip: You can only dismiss open alerts.

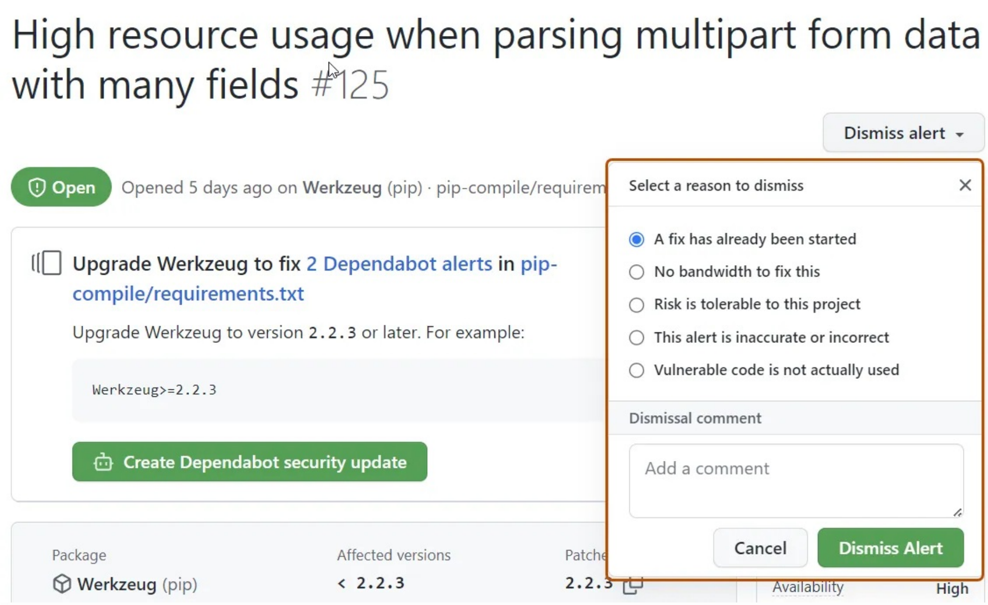
If you schedule extensive work to upgrade a dependency, or decide that an alert does not need to be fixed, you can dismiss the alert. Dismissing alerts that you have already assessed makes it easier to triage new alerts as they appear.

1. View the details for an alert. For more information, see "Viewing vulnerable dependencies" (above).

2. Select the "Dismiss" dropdown, and click a reason for dismissing the alert. Unfixed dismissed alerts can be reopened later.

3. Optionally, add a dismissal comment. The dismissal comment will be added to the alert timeline and can be used as justification during auditing and reporting. You can retrieve or set a comment by using the GraphQL API. The comment is contained

in the `dismissComment` field. For more information, see "Objects" in the GraphQL API documentation.



High resource usage when parsing multipart form data with many fields #125

Dismiss alert

Open Opened 5 days ago on Werkzeug (pip) · pip-compile/requirem

Upgrade Werkzeug to fix 2 Dependabot alerts in pip-compile/requirements.txt

Upgrade Werkzeug to version 2.2.3 or later. For example:

```
Werkzeug>=2.2.3
```

Create Dependabot security update

Package	Affected versions	Patche
Werkzeug (pip)	< 2.2.3	2.2.3

Select a reason to dismiss ✕

● A fix has already been started

○ No bandwidth to fix this

○ Risk is tolerable to this project

○ This alert is inaccurate or incorrect

○ Vulnerable code is not actually used

Dismissal comment

Add a comment

Cancel Dismiss Alert

Availability High

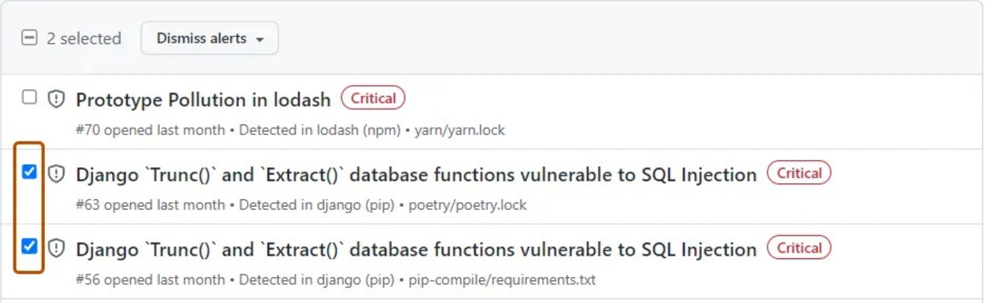
4 Click **Dismiss alert**.

Dismissing multiple alerts at once

1 View the open Dependabot alerts. For more information, see "Viewing and updating Dependabot alerts".

2 Optionally, filter the list of alerts by selecting a dropdown menu, then clicking the filter that you would like to apply. You can also type filters into the search bar.

3 To the left of each alert title, select the alerts that you want to dismiss.



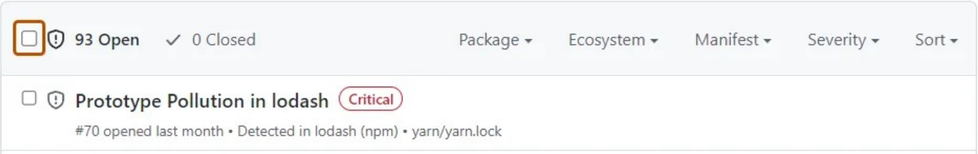
⊟ 2 selected Dismiss alerts

□ Prototype Pollution in lodash Critical
#70 opened last month • Detected in lodash (npm) • yarn/yarn.lock

☑ Django `Trunc()` and `Extract()` database functions vulnerable to SQL Injection Critical
#63 opened last month • Detected in django (pip) • poetry/poetry.lock

☑ Django `Trunc()` and `Extract()` database functions vulnerable to SQL Injection Critical
#56 opened last month • Detected in django (pip) • pip-compile/requirements.txt

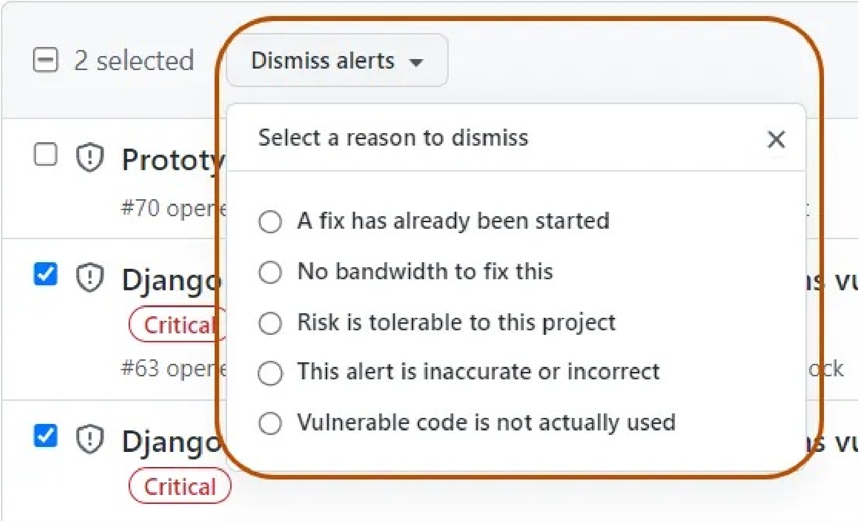
4 Optionally, at the top of the list of alerts, select all alerts on the page.



□ 93 Open ✓ 0 Closed Package Ecosystem Manifest Severity Sort

□ Prototype Pollution in lodash Critical
#70 opened last month • Detected in lodash (npm) • yarn/yarn.lock

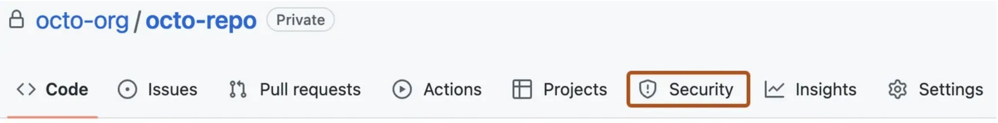
5 Select the "Dismiss alerts" dropdown, and click a reason for dismissing the alerts.


2 selected
Dismiss alerts
Select a reason to dismiss
○ A fix has already been started
○ No bandwidth to fix this
○ Risk is tolerable to this project
○ This alert is inaccurate or incorrect
○ Vulnerable code is not actually used
Prototy
#70 opene
Django
Critical
#63 opene
Django
Critical

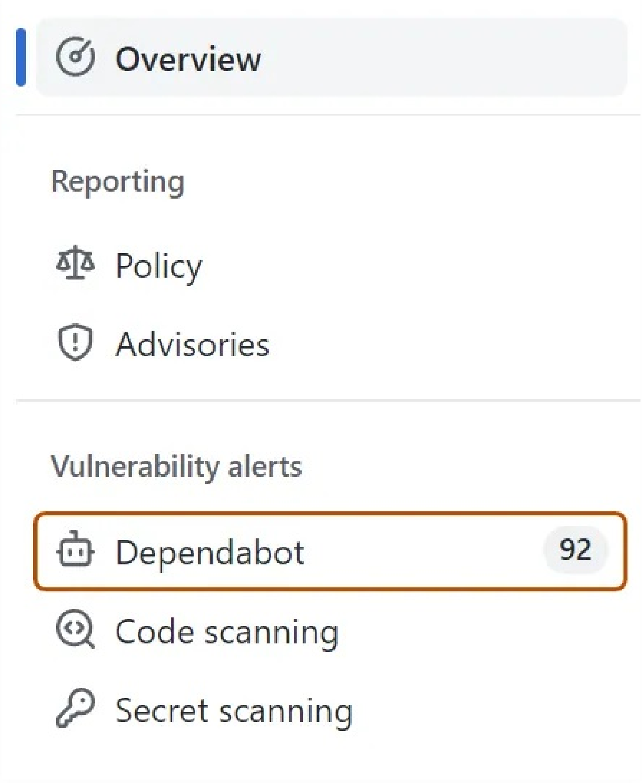
Viewing and updating closed alerts

You can view all open alerts, and you can reopen alerts that have been previously dismissed. Closed alerts that have already been fixed cannot be reopened.

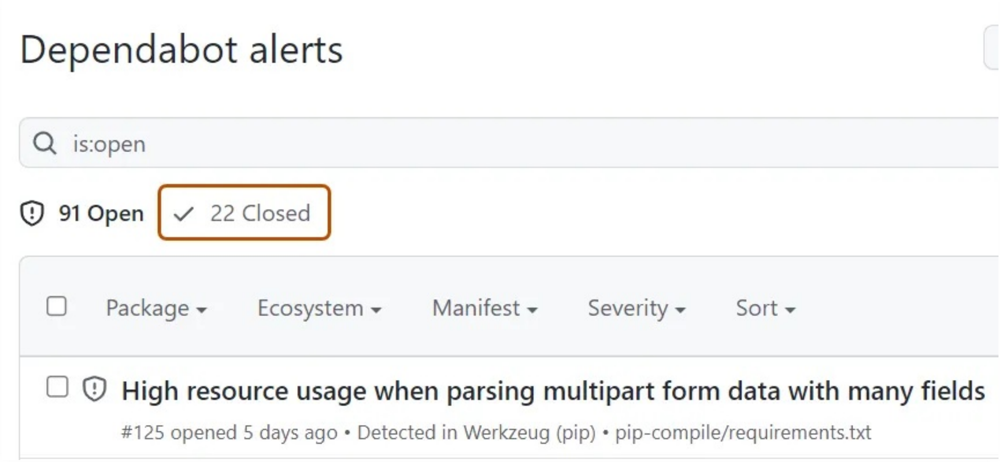
1. On GitHub.com, navigate to the main page of the repository.
2. Under the repository name, click **Security**. If you cannot see the "Security" tab, select the ··· dropdown menu, and then click **Security**.



3. In the "Vulnerability alerts" sidebar of security overview, click **Dependabot**. If this option is missing, it means you don't have access to security alerts and need to be given access. For more information, see "Managing security and analysis settings for your repository."



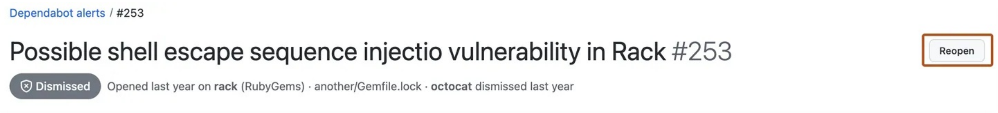
4. To just view closed alerts, click **Closed**.



□ High resource usage when parsing multipart form data with many fields
#125 opened 5 days ago • Detected in Werkzeug (pip) • pip-compile/requirements.txt

5. Click the alert that you would like to view or update.

6. Optionally, if the alert was dismissed and you wish to reopen it, click **Reopen**. Alerts that have already been fixed cannot be reopened.



Possible shell escape sequence injectio vulnerability in Rack #253

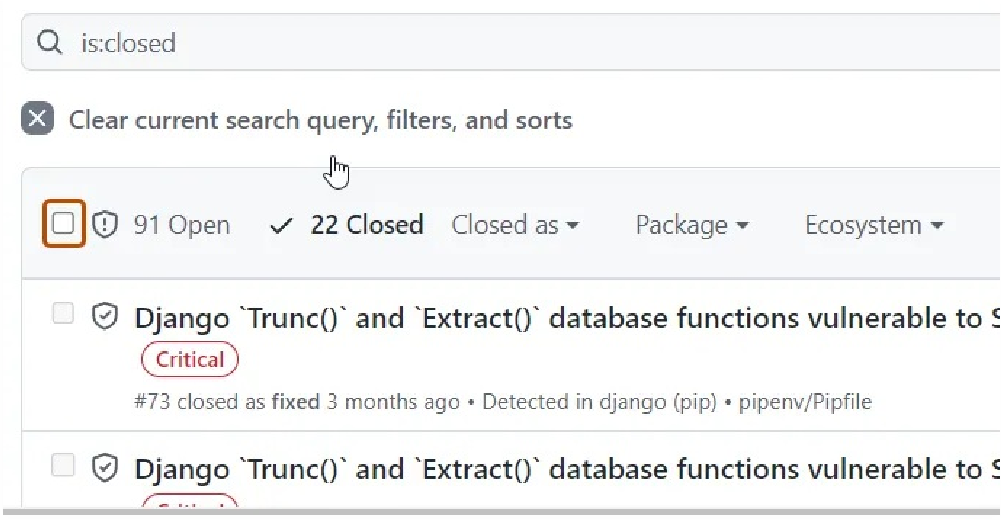
Opened last year on **rack** (RubyGems) · another/Gemfile.lock · **octocat** dismissed last year

Reopening multiple alerts at once

1. View the closed Dependabot alerts. For more information, see "Viewing and updating Dependabot alerts" (above).

2. To the left of each alert title, select the alerts that you want to reopen by clicking the checkbox adjacent to each alert.

3. Optionally, at the top of the list of alerts, select all closed alerts on the page.



is:closed

☒ Clear current search query, filters, and sorts

□ 91 Open ✓ 22 Closed Closed as ▾ Package ▾ Ecosystem ▾

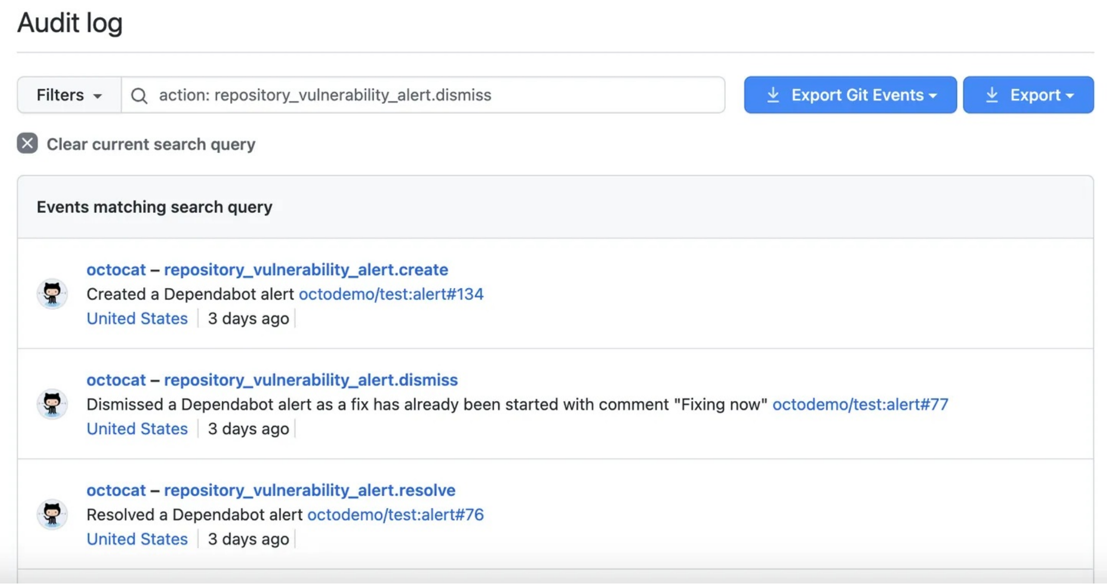
□ Django `Trunc()` and `Extract()` database functions vulnerable to S
Critical
#73 closed as fixed 3 months ago • Detected in django (pip) • pipenv/Pipfile

□ Django `Trunc()` and `Extract()` database functions vulnerable to S

4. Click **Reopen** to reopen the alerts. Alerts that have already been fixed cannot be reopened.

Reviewing the audit logs for Dependabot alerts

When a member of your organization performs an action related to Dependabot alerts, you can review the actions in the audit log. For more information about accessing the log, see "Reviewing the audit log for your organization."



Audit log

Filters ▾ action: repository_vulnerability_alert.dismiss Export Git Events ▾ Export ▾

☒ Clear current search query

Events matching search query

octocat – repository_vulnerability_alert.create
Created a Dependabot alert octodemo/test:alert#134
United States | 3 days ago

octocat – repository_vulnerability_alert.dismiss
Dismissed a Dependabot alert as a fix has already been started with comment "Fixing now" octodemo/test:alert#77
United States | 3 days ago

octocat – repository_vulnerability_alert.resolve
Resolved a Dependabot alert octodemo/test:alert#76
United States | 3 days ago

Events in your audit log for Dependabot alerts include details such as who performed the action, what the action was, and when the action was performed. The event also includes a link to the alert itself. When a member of your organization dismisses an alert, the event displays the dismissal reason and comment. For information on the Dependabot alerts actions, see the `repository_vulnerability_alert` category in "Audit log events for your organization."