

Securing your repository

In this article

Introduction

Managing access to your repository

Managing the dependency graph

Managing Dependabot alerts

Managing dependency review

Managing Dependabot security updates

Managing Dependabot version updates

Configuring code scanning

Configuring secret scanning

Setting a security policy

Next steps

You can use a number of GitHub features to help keep your repository secure.

Who can use this feature

Repository administrators and organization owners can configure repository security settings.

Introduction [↗](#)


This guide shows you how to configure security features for a repository. You must be a repository administrator or organization owner to configure security settings for a repository.

Your security needs are unique to your repository, so you may not need to enable every feature for your repository. For more information, see "[GitHub security features](#)."

Some features are available for all repositories. Additional features are available to enterprises that use GitHub Advanced Security. For more information, see "[About GitHub Advanced Security](#)."

Managing access to your repository [↗](#)

The first step to securing a repository is to establish who can see and modify your code. For more information, see "[Managing your repository's settings and features](#)."

From the main page of your repository, click  **Settings**, then scroll down to the "Danger Zone."

- To change who can view your repository, click **Change visibility**. For more information, see "[Setting repository visibility](#)."
- To change who can access your repository and adjust permissions, click **Manage access**. For more information, see "[Managing teams and people with access to your repository](#)."

Managing the dependency graph

Enterprise owners can configure the dependency graph and Dependabot alerts for an enterprise. For more information, see "[Enabling the dependency graph for your enterprise](#)" and "[Enabling Dependabot for your enterprise](#)."

For more information, see "[Exploring the dependencies of a repository](#)."

Managing Dependabot alerts

Dependabot alerts are generated when GitHub identifies a dependency in the dependency graph with a vulnerability.

For an overview of the different features offered by Dependabot and instructions on how to get started, see "[Dependabot quickstart guide](#)."

Enterprise owners must configure the dependency graph and Dependabot alerts for an enterprise.


Once Dependabot alerts have been configured, repository administrators and organization owners can enable Dependabot alerts for private and internal repositories in their "Code security and analysis" settings page. Public repositories are enabled by default. For more information, see "[Enabling the dependency graph for your enterprise](#)", "[Enabling Dependabot for your enterprise](#)", and [Configuring Dependabot alerts](#)."

For more information, see "[About Dependabot alerts](#)."

Managing dependency review


Dependency review lets you visualize dependency changes in pull requests before they are merged into your repositories. For more information, see "[About dependency review](#)."

Dependency review is a GitHub Advanced Security feature. To enable dependency review for a repository, ensure that the dependency graph is enabled and enable GitHub Advanced Security.

- 1 From the main page of your repository, click  **Settings**.
- 2 Click **Security & analysis**.
- 3 Check that dependency graph is configured for your enterprise.
- 4 If GitHub Advanced Security is not already enabled, click **Enable**.

Managing Dependabot security updates

For any repository that uses Dependabot alerts, you can enable Dependabot security updates to raise pull requests with security updates when vulnerabilities are detected.

- 1 From the main page of your repository, click  **Settings**.
- 2 Click **Security & analysis**.
- 3 Next to Dependabot security updates, click **Enable**.

For more information, see "[About Dependabot security updates](#)" and "[Configuring](#)

[Dependabot security updates.](#)"



Managing Dependabot version updates

You can enable Dependabot to automatically raise pull requests to keep your dependencies up-to-date. For more information, see "[About Dependabot version updates.](#)"

To enable Dependabot version updates, you must create a `dependabot.yml` configuration file. For more information, see "[Configuring Dependabot version updates.](#)"

Configuring code scanning

You can configure code scanning to automatically identify vulnerabilities and errors in the code stored in your repository by using a CodeQL analysis workflow or third-party tool. Depending on the programming languages in your repository, you can configure code scanning with CodeQL using default setup, in which GitHub automatically determines the languages to scan, query suites to run, and events that will trigger a new scan. For more information, see "[Configuring default setup for code scanning.](#)"


- 1 From the main page of your repository, click  **Settings**.
- 2 In the "Security" section of the sidebar, click  **Code security and analysis**.
- 3 In the "Code scanning" section, select **Set up** ▾, then click **Default**.
- 4 In the pop-up window that appears, review the default configuration settings for your repository, then click **Enable CodeQL**.

Alternatively, you can use advanced setup, which generates a workflow file you can edit to customize your code scanning with CodeQL. For more information, see "[Configuring advanced setup for code scanning.](#)"

Code scanning is available for organization-owned repositories if your enterprise uses GitHub Advanced Security.

Configuring secret scanning


Secret scanning is available for organization-owned repositories in GitHub Enterprise Server if your enterprise has a license for GitHub Advanced Security. For more information, see "[About secret scanning](#)" and "[About GitHub Advanced Security.](#)"

- 1 From the main page of your repository, click  **Settings**.
- 2 Click **Code security & analysis**.
- 3 If GitHub Advanced Security is not already enabled, click **Enable**.
- 4 Next to Secret scanning, click **Enable**.

Setting a security policy

If you are a repository maintainer, it's good practice to specify a security policy for your repository by creating a file named `SECURITY.md` in the repository. This file instructs users about how to best contact you and collaborate with you when they want to report security vulnerabilities in your repository. You can view the security policy of a repository

from the repository's **Security** tab.

- 1 From the main page of your repository, click  **Security**.
- 2 Click **Security policy**.
- 3 Click **Start setup**.
- 4 Add information about supported versions of your project and how to report vulnerabilities.

For more information, see "[Adding a security policy to your repository](#)."

Next steps

You can view and manage alerts from security features to address dependencies and vulnerabilities in your code. For more information, see "[Viewing and updating Dependabot alerts](#)," "[Managing pull requests for dependency updates](#)," "[Managing code scanning alerts for your repository](#)," and "[Managing alerts from secret scanning](#)".

You can also use GitHub's tools to audit responses to security alerts. For more information, see "[Auditing security alerts](#)".

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)