Securing your repository

In this article

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 repositories on all plans. Additional features are available to enterprises that use GitHub Advanced Security. GitHub Advanced Security features are also enabled for all public repositories on GitHub.com. 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

The dependency graph is automatically generated for all public repositories. You can choose to enable it for forks and for private repositories. The dependency graph interprets manifest and lock files in a repository to identify dependencies.

1. From the main page of your repository, click **Settings**.
2. Click **Security & analysis**.
3. Next to Dependency graph, click **Enable** or **Disable**.

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. You can enable Dependabot alerts for any repository.

Additionally, you can use Dependabot alert rules to filter out false positive alerts or alerts you're not interested in, based on complex logic from a variety of contextual criteria. For more information, see "About Dependabot alert rules."

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

1. Click your profile photo, then click **Settings**.
2. Click **Security & analysis**.
3. Click **Enable all** next to Dependabot alerts.

For more information, see "About Dependabot alerts" and "Managing security and analysis settings for your personal account."

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. Dependency review is already enabled for all public repositories. To enable dependency review for a private or internal 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. If dependency graph is not already enabled, click **Enable**.
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."

1. From the main page of your repository, click **Settings**.
2. Click **Security & analysis**.
3. Next to Dependabot version updates, click **Enable** to create a basic `dependabot.yml` configuration file.
4. Specify the dependencies to update and any associated configuration options, then commit the file to the repository. 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 all public repositories, and for private repositories owned by organizations that are part of an enterprise with a license for GitHub Advanced Security.

Configuring secret scanning

Secret scanning alerts for partners runs automatically on public repositories and public npm packages to notify service providers about leaked secrets on GitHub.com.

Secret scanning alerts for users are available for free on all public repositories. Organizations using GitHub Enterprise Cloud with a license for GitHub Advanced Security can also enable secret scanning alerts for users on their private and internal repositories. For more information, see "About secret scanning" and "About GitHub Advanced Security."

For information about how you can try GitHub Advanced Security for free, see "Setting up a trial of 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".

If you have a security vulnerability, you can create a security advisory to privately discuss and fix the vulnerability. For more information, see "About repository security advisories" and "Creating a repository security advisory."