Managing security and analysis settings for your organization

In this article

You can control features that secure and analyze the code in your organization's projects on GitHub.

> **Who can use this feature**
> Organization owners can manage security and analysis settings for repositories in the organization.

About management of security and analysis settings

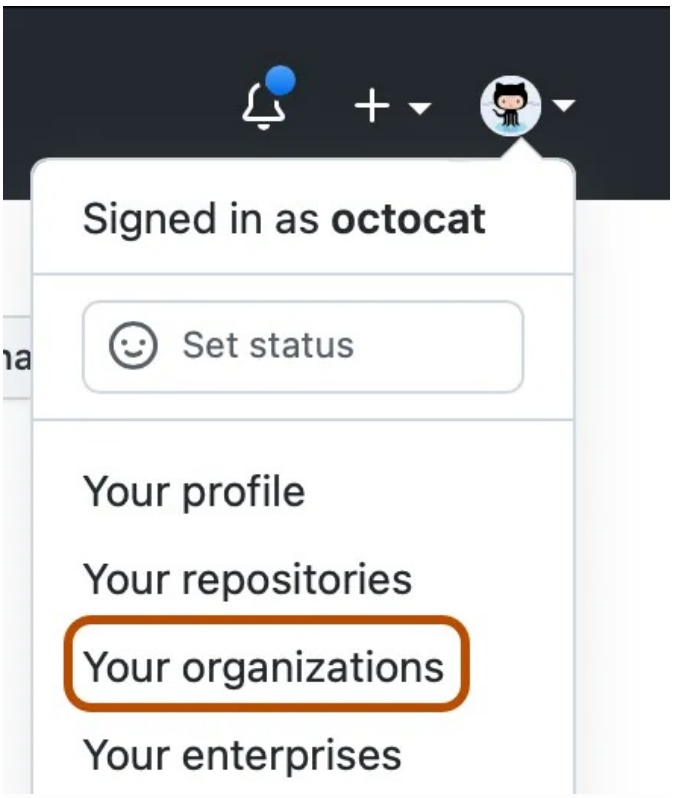
GitHub can help you to secure the repositories in your organization. You can manage the security and analysis features for all existing or new repositories that members create in your organization. If you have a license for GitHub Advanced Security then you can also manage access to these features. For more information, see "About GitHub Advanced Security."

Note: You can't disable some security and analysis features that are enabled by default for public repositories.

If you enable security and analysis features, GitHub performs read-only analysis on your repository.

Displaying the security and analysis settings

1. In the top right corner of GitHub.com, click your profile photo, then click **Your organizations**.



2. Next to the organization, click **Settings**.
3. In the "Security" section of the sidebar, click **Code security and analysis**.

The page that's displayed allows you to enable or disable all security and analysis features for the repositories in your organization.

If your organization belongs to an enterprise with a license for GitHub Advanced Security, the page will also contain options to enable and disable Advanced Security features. Any repositories that use GitHub Advanced Security are listed at the bottom of the page.

Enabling or disabling a feature for all existing repositories

You can enable or disable features for all repositories. The impact of your changes on repositories in your organization is determined by their visibility:

- **Private vulnerability reporting** - Your changes affect public repositories only.
- **Dependency graph** - Your changes affect only private repositories because the feature is always enabled for public repositories.
- **Dependabot alerts** - Your changes affect all repositories.
- **Dependabot security updates** - Your changes affect all repositories.
- **GitHub Advanced Security** - Your changes affect only private repositories because GitHub Advanced Security and the related features are always enabled for public repositories.
- **Secret scanning** - Your changes affect public repositories, and private or internal repositories where GitHub Advanced Security is enabled. This option controls whether or not secret scanning alerts for users are enabled. Secret scanning alerts for partners always runs on all public repositories.
- **Code scanning** - Your changes affect public repositories, and private or internal repositories where GitHub Advanced Security is enabled. For information about eligible repositories, see Configuring default setup for code scanning at scale. For repositories that are not eligible for default setup, you can configure advanced setup

at the repository level. For more information, see "Configuring advanced setup for code scanning."

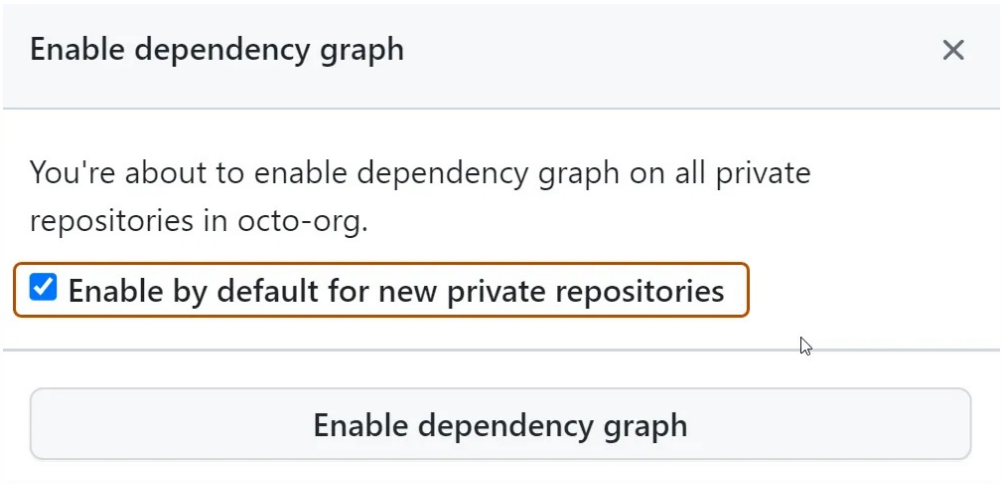
You can use security overview to find a set of repositories and enable or disable security features for them all at the same time. For more information, see "Enabling security features for multiple repositories."

Note: If you enable GitHub Advanced Security, active committers to these repositories will use GitHub Advanced Security licenses. This option is deactivated if you have exceeded your license capacity. For more information, see "About billing for GitHub Advanced Security."

Note: If you encounter an error that reads "GitHub Advanced Security cannot be enabled because of a policy setting for the organization," contact your enterprise admin and ask them to change the GitHub Advanced Security policy for your enterprise. For more information, see "Enforcing policies for code security and analysis for your enterprise."

Note: When Dependabot alerts are enabled or disabled at the enterprise level, it overrides the organization level settings for Dependabot alerts. For more information, see "Configuring Dependabot alerts."

1. Go to the security and analysis settings for your organization. For more information, see "Displaying the security and analysis settings."
2. Under "Code security and analysis", to the right of the feature, click **Disable all** or **Enable all** to display a confirmation dialog box. The control for "GitHub Advanced Security" is disabled if you have no available licenses for GitHub Advanced Security.
3. Review the information in the dialog box.
4. Optionally, if you are enabling private vulnerability reporting, dependency graph, or Dependabot, select **Enable by default for new private repositories**.



5. When you are ready to make the changes, click **Disable FEATURE** or **Enable FEATURE** to disable or enable the feature for all the repositories in your organization.
6. Optionally, in your feature's section of the security and analysis settings, select additional enablement settings. Additional enablement settings may include:
   - Automatic enablement for a specific type of repository
   - Feature-specific settings, such as recommending the extended query suite for code scanning default setup throughout your organization, or automatic secret validation for secret scanning

Notes:

- If you disable CodeQL code scanning for all repositories this change is not reflected in the coverage information shown in security overview for the organization. The repositories will still appear to have code scanning enabled in the "Security Coverage" view.
- Enabling code scanning for all eligible repositories in an organization will not override existing code scanning configurations. For information on configuring default setup with different settings for specific repositories, see "Configuring default setup for code scanning" and "Configuring default setup for code scanning at scale."

When you enable one or more security and analysis features for existing repositories, you will see any results displayed on GitHub within minutes:

- All the existing repositories will have the selected configuration.
- New repositories will follow the selected configuration if you've enabled the checkbox for new repositories.
- We use the permissions to scan for manifest files to apply the relevant services.
- If enabled, you'll see dependency information in the dependency graph.
- If enabled, GitHub will generate Dependabot alerts for vulnerable dependencies or malware.
- If enabled, Dependabot security updates will create pull requests to upgrade vulnerable dependencies when Dependabot alerts are triggered.

Enabling or disabling a feature automatically when new repositories are added

1. Go to the security and analysis settings for your organization. For more information, see "Displaying the security and analysis settings."
2. Under "Code security and analysis", locate the feature, enable or disable the feature by default for new repositories, or all new private repositories, in your organization.

Allowing Dependabot to access private dependencies

Dependabot can check for outdated dependency references in a project and automatically generate a pull request to update them. To do this, Dependabot must have access to all of the targeted dependency files. Typically, version updates will fail if one or more dependencies are inaccessible. For more information, see "About Dependabot version updates."

By default, Dependabot can't update dependencies that are located in private repositories or private package registries. However, if a dependency is in a private GitHub repository within the same organization as the project that uses that dependency, you can allow Dependabot to update the version successfully by giving it access to the host repository.

If your code depends on packages in a private registry, you can allow Dependabot to update the versions of these dependencies by configuring this at the repository level. You do this by adding authentication details to the `dependabot.yml` file for the repository. For more information, see "Configuration options for the dependabot.yml file."

Note: For the option to grant Dependabot access to private repositories to be available, you

need Dependabot version updates or Dependabot security updates to be enabled on at least one repository within the organization.

To allow Dependabot to access a private GitHub repository:

1. Go to the security and analysis settings for your organization. For more information, see "Displaying the security and analysis settings."
2. Under "Grant Dependabot private repository access", click **Add private repositories** or **Add internal and private repositories** to display a repository search field.

3. Start typing the name of the repository you want to grant Dependabot access to.
4. A list of matching repositories in the organization is displayed, click the repository you want to to allow access to and this adds the repository to the allowed list.
5. Optionally, to remove a repository from the list, to the right of the repository, click ×.

Allowing validity checks for partner patterns in an organization

Note: Validity checks for partner patterns is currently in beta and subject to change.

Validity checks for partner patterns is available on all types of repositories on GitHub.com. To use this feature, you must have a license for GitHub Advanced Security.

You can allow secret scanning to automatically check the validity of a secret by sending it to the relevant partner. When you select the checkbox in the organization settings, the feature is enabled for all repositories in the organization. Alternatively, you can enable the validity check for a single repository, or at the enterprise level. For more information, see "Allowing validity checks for partner patterns in a repository" and "Managing GitHub Advanced Security features for your enterprise."

1. Go to the security and analysis settings for your organization. For more information, see "Displaying the security and analysis settings."

2. Under Secret scanning, select the checkbox next to "Automatically verify if a secret is valid by sending it to the relevant partner".

Removing access to GitHub Advanced Security from individual repositories in an organization

You can manage access to GitHub Advanced Security features for a repository from its "Settings" tab. For more information, see "Managing security and analysis settings for your repository." However, you can also disable GitHub Advanced Security features for a repository from the "Settings" tab for the organization.

1. Go to the security and analysis settings for your organization. For more information, see "Displaying the security and analysis settings."
2. To see a list of all the repositories in your organization with GitHub Advanced Security enabled, scroll to the "GitHub Advanced Security repositories" section.

The table lists the number of unique committers for each repository. This is the number of licenses you could free up by removing access to GitHub Advanced Security. For more information, see "About billing for GitHub Advanced Security."

1. To remove access to GitHub Advanced Security from a repository and free up licenses used by any active committers that are unique to the repository, click the adjacent ×.
2. In the confirmation dialog, click **Remove repository** to remove access to the features of GitHub Advanced Security.

Note: If you remove access to GitHub Advanced Security for a repository, you should communicate with the affected development team so that they know that the change was intended. This ensures that they don't waste time debugging failed runs of code scanning.

Further reading

- "Securing your repository"
- "About secret scanning"
- "About the dependency graph"
- "About supply chain security"