This article is also available in French. ✕

GitHub Docs Version: Enterprise Cloud ▾ Search GitHub Docs ···

☰ Code security / Secret scanning / Introduction / Secret scanning

About secret scanning

GitHub Enterprise Cloud scans repositories for known types of secrets, to prevent fraudulent use of secrets that were committed accidentally.

Who can use this feature?

Secret scanning is available for the following repositories:

- Public repositories (for free)
- Private and internal repositories in organizations using GitHub Enterprise Cloud with GitHub Advanced Security enabled
- User-owned repositories for GitHub Enterprise Cloud with Enterprise Managed Users

In this article

About secret scanning

Secret scanning is a security feature that helps detect and prevent the accidental inclusion of sensitive information such as API keys, passwords, tokens, and other secrets in your repository. When enabled, secret scanning scans commits in repositories for known types of secrets and alerts repository administrators upon detection.

Secret scanning scans your entire Git history on all branches present in your GitHub repository for secrets, even if the repository is archived. GitHub will also periodically run a full Git history scan of existing content in GitHub Advanced Security repositories where secret scanning is enabled.

Additionally, secret scanning scans:

- Descriptions and comments in issues
- Titles, descriptions, and comments, in open and closed *historical* issues. A notification is sent to the relevant partner when a historical partner pattern is detected.
- Titles, descriptions, and comments in pull requests
- Titles, descriptions, and comments in GitHub Discussions
- Wikis

This additional scanning is free for public repositories.

When a supported secret is leaked, GitHub Enterprise Cloud generates a secret scanning alert. Alerts are reported on the **Security** tab of repositories on GitHub Enterprise Cloud, where you can view, evaluate, and resolve them. For more information, see "Managing alerts from secret scanning."

Service providers can partner with GitHub to provide their secret formats for scanning. We automatically run secret scanning for partner patterns on all public repositories and public npm packages. To find out about our partner program, see "Secret scanning partner program."

Any strings that match patterns that were provided by secret scanning partners are reported directly to the relevant partner, and aren't displayed on GitHub. For more information about partner patterns, see "About secret scanning alerts."

For information about the secrets and service providers supported by secret scanning, see "Supported secret scanning patterns."

You can use the REST API to monitor results from secret scanning across your repositories. For more information about API endpoints, see "REST API endpoints for secret scanning."

You can also use security overview to see an organization-level view of which repositories have enabled secret scanning and the alerts found. For more information, see "About security overview."

You can audit the actions taken in response to secret scanning alerts using GitHub tools. For more information, see "Auditing security alerts."

How secret scanning works

Below is a typical workflow that explains how secret scanning works:

- **Detection**: Secret scanning automatically scans your repository's contents for sensitive data, such as API keys, passwords, tokens, and other secrets. It looks for patterns and heuristics that match known types of secrets.
- **Alerts**: When a potential secret is detected, GitHub generates an alert and notifies the relevant repository administrators and users. This notification includes details about the detected secret, such as its location in the repository. For more information about alert types and alert details, see "About secret scanning alerts."
- **Review**: When a secret is detected, you'll need to review the alert details provided.
- **Remediation**: You then need to take appropriate actions to remediate the exposure. This might include:
    - Rotating the affected credential to ensure it is no longer usable.
    - Removing the secret from the repository's history (using tools like BFG Repo-Cleaner or GitHub's built-in features).
- **Monitoring**: It's good practice to regularly audit and monitor your repositories to ensure no other secrets are exposed.
- **Integration with partners**: GitHub works with various service providers to validate secrets. When a partner secret is detected, GitHub notifies the provider so they can take appropriate action, such as revoking the credential. For more information about the partnership program, see "Secret scanning partner program."

About the benefits of secret scanning

- **Enhanced security**—Secret scanning scans your repositories for sensitive information like API keys, passwords, tokens, and other secrets. By detecting these early, you can mitigate potential security risks before they are exploited by malicious actors.
- **Automated detection**—The feature automatically scans your codebase, including commits, issues, and pull requests, ensuring continuous protection without requiring manual intervention. This automation helps in maintaining security even as your repository evolves.
- **Real-time alerts**—When a secret is detected, secret scanning provides real-time alerts to repository administrators and contributors. This immediate feedback allows for swift remediation actions.
- **Integration with service providers**—GitHub partners with various service providers to validate detected secrets. When a secret is identified, GitHub notifies the corresponding service provider to take appropriate actions, such as revoking the exposed credential. For more information, see "Secret scanning partner program."
- **Custom pattern support**—Organizations can define custom patterns to detect proprietary or unique types of secrets that may not be covered by default patterns. This flexibility allows for tailored security measures specific to your environment.
- **Ability to detect non-provider patterns**—You can expand the detection to include non-provider patterns such as connection strings, authentication headers, and private keys, for your repository or organization.

Customizing secret scanning

Once secret scanning is enabled, you can customize it further:

Detection of non-provider patterns

Scan for and detect secrets that are not specific to a service provider, such as private keys and generic API keys. For more information, see "Enabling secret scanning for non-provider patterns."

Performing validity checks

Validity checks help you prioritize alerts by telling you which secrets are `active` or `inactive`. For more information, see "Enabling validity checks for your repository" and "Evaluating alerts from secret scanning."

Defining custom patterns

Define your own patterns for secrets used by your organization that secret scanning can scan for and detect. For more information, see "Defining custom patterns for secret scanning."

Copilot secret scanning

- **Generic secret detection**: Leverage secret scanning's AI capabilities to detect unstructured secrets, such as passwords, in your repository. For more information, see "Responsible detection of generic secrets with Copilot secret scanning."
- **Regular expression generator**: Leverage secret scanning's AI capabilities to generate regular expressions that will capture all your custom patterns. For more information, see "Responsible use of AI to define regular expressions.

Further reading

- "Enabling secret scanning for your repository"
- "About push protection"
- "Working with secret scanning and push protection"
- "Best practices for preventing data leaks in your organization"
- "Quickstart for securing your repository"
- "Keeping your account and data secure"

Help and support

Did you find what you needed?

Yes No

Privacy policy

Help us make these docs great!

All GitHub docs are open source. See something that's wrong or unclear? Submit a pull request.

Make a contribution

Learn how to contribute

Still need help?

Ask the GitHub community

Contact support

Legal

Terms Privacy Status Pricing Expert services Blog