Pushing a branch blocked by push protection

In this article

The push protection feature of secret scanning proactively protects you against leaked secrets in your repositories. You can resolve blocked pushes and, once the detected secret is removed, you can push changes to your working branch from the command line or the web UI.

> 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."

About push protection for secret scanning 🔗

The push protection feature of secret scanning helps to prevent security leaks by scanning for secrets before you push changes to your repository. When you enable push protection for your organization or repository, secret scanning also checks pushes for high-confidence secrets (those identified with a low false positive rate). Secret scanning lists any secrets it detects so the author can review the secrets and remove them or, if needed, allow those secrets to be pushed. For information on the secrets and service providers supported for push protection, see "Secret scanning patterns."

If you confirm a secret is real, you need to remove the secret from your branch, *from all the commits it appears in*, before pushing again.

> **Tip** If GitHub blocks a secret that you believe is safe to push, you can allow the secret and specify the reason why it should be allowed. For more information about bypassing push protection for a secret, see "Allowing a blocked secret to be pushed" and "Bypassing push protection for a secret" for the command line and the web UI, respectively.

Organization owners can provide a custom link that will be included in the message from GitHub Enterprise Server when your push is blocked. This custom link can contain resources and advice specific to your organization and its policies.

Resolving a blocked push on the command line 🔗

When you attempt to push a supported secret to a repository or organization with secret scanning as a push protection enabled, GitHub will block the push. You can remove the secret from your branch or follow a provided URL to allow the push.

> **Notes**:
>
> - If your git configuration supports pushes to multiple branches, and not only to the current branch, your push may be blocked due to additional and unintended refs being pushed. For more information, see the `push.default` options in the Git documentation.
> - If secret scanning upon a push times out, GitHub will still scan your commits for secrets after the push.

If the blocked secret was introduced by the latest commit on your branch, you can follow the guidance below.

1. Remove the secret from your code.

2. Commit the changes, by using `git commit --amend`.

3. Push your changes with `git push`.

You can also remove the secret if the secret appears in an earlier commit in the Git history.

1. Use `git log` to determine which commit surfaced in the push error came first in history.

2. Start an interactive rebase with `git rebase -i <commit-id>~1`. is the id of the commit from step 1.

3. Identify your commit to edit by changing `pick` to `edit` on the first line of the text that appears in the editor.

4. Remove the secret from your code.

5. Commit the change with `git commit --amend`.

6. Run `git rebase --continue` to finish the rebase.

Resolving a blocked commit in the web UI 🔗

When you use the web UI to attempt to commit a supported secret to a repository or organization with secret scanning as a push protection enabled, GitHub will block the commit.

You will see a dialog box with information about the secret's location, as well as options allowing you to push the secret. The secret will also be underlined in the file so you can easily find it.

To resolve a blocked commit in the web UI, you need to remove the secret from the file, or use the options displayed in the dialog box to allow the secret. For more information about bypassing push protection from the web UI, see "[Push protection for repositories and organizations](#)."

If you confirm a secret is real, you need to remove the secret from the file. Once you remove the secret, you will be able to commit your changes.