Getting started

`ggshield` is a CLI application that runs in your local environment or in a CI environment to help you detect more than 400+ types of secrets.

> ggshield is open source on GitHub and accessible here.

`ggshield` can run:

- **in your local environment** to scan local files and repositories or as a pre-commit hook.
- **in a CI environment**,
- **in a pre-receive hook**, if you have a self-managed VCS instance

Note : `ggshield` uses our public API through py-gitguardian to scan files. Only metadata such as call time, request size and scan mode is stored when launching a scan with ggshield, therefore secrets incidents will not be displayed on your dashboard and **your files and secrets won't be stored**.

Step 1: Install ggshield

Requirements

`ggshield` works on macOS, Linux and Windows.

It requires **Python 3.8 and newer** (except for standalone packages) and git.

Some commands require additional programs:

- docker: to scan docker images.
- pip: to scan pypi packages.

macOS

Homebrew

You can install `ggshield` using Homebrew:

```
$ brew install gitguardian/tap/ggshield
```

Upgrading is handled by Homebrew.

Standalone .pkg package

Alternatively, you can download and install a standalone .pkg package from `ggshield` release page.

This package *does not* require installing Python, but you have to manually download new versions.

Linux

Deb and RPM packages

Deb and RPM packages are available on Cloudsmith.

Setup instructions:

- Deb packages
- RPM packages

Upgrading is handled by the package manager.

Windows

Standalone .zip archive

We provide a standalone .zip archive on `ggshield` release page.

Unpack the archive on your disk, then add the directory containing the `ggshield.exe` file to `%PATH%`.

This archive *does not* require installing Python, but you have to manually download new versions.

All operating systems

`ggshield` can be installed on all supported operating systems via its PyPI package.

Using pipx

The recommended way to install `ggshield` from PyPI is to use pipx, which will install it in an isolated environment:

```
$ pipx install ggshield
```

To upgrade your installation, run:

```
$ pipx upgrade ggshield
```

Using pip

You can also install `ggshield` from PyPI using pip, but this is not recommended because the installation is not isolated, so other applications or packages installed this way may affect your `ggshield` installation. This method will also not work if your Python installation is declared as externally managed (for example when using the system Python on operating systems like Debian 12):

```
$ pip install --user ggshield
```

To upgrade your installation, run:

```
$ pip install --user --upgrade ggshield
```

Step 2: Authenticate with your GitGuardian workspace

`ggshield` requires an API key to authenticate the CLI with your GitGuardian workspace. There are 2 different types of API keys:

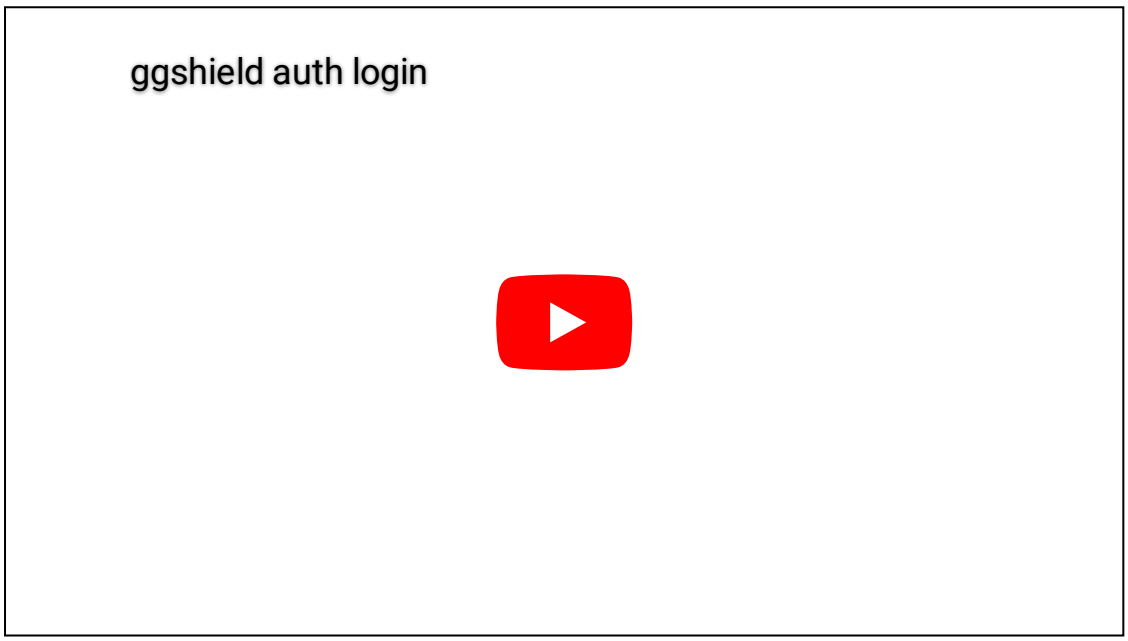
- **Service Accounts:** a special type of token intended to represent a non-human user that needs to authenticate and be authorized for scenarios such as secrets scanning in CI pipelines or batch processing open incidents.
- **Personal Access Tokens:** a token intended for the use of the GitGuardian API and command-line application ggshield by individual developers on their local workstations (e.g. pre-commit or pre-push git hooks).

Option 1: Automatically

If you want to set up **ggshield for use on your local workstation (e.g. to scan repos or in a pre-commit or pre-push git hook)**, we recommend running the following command:

```
ggshield auth login
```

This will open a new window in your web browser. Simply follow the steps to login to your workspace (or create a new account) and GitGuardian will automatically provision a personal access token and store it in your configuration.



You can find more details in the login command reference section.

Option 2: Manually

You can also provision your API key manually. This is useful when you want to set up **ggshield in your CI environment** for example.

Create your API key

To create your API key manually, please follow the steps described in the API authentication section. Once you have your API key ready, follow the rest of the guide on this page.

Source your API key in your environment

Alternatively, you can create your personal access token manually and store it in the `GITGUARDIAN_API_KEY` environment variable to complete the setup.

If you're using an on-premise version of GitGuardian, you also need to set the `GITGUARDIAN_INSTANCE` environment variable with your on-premise instance URL (eg: `https://dashboard.gitguardian.mycorp.local`).

Step 3: Scan your first content with ggshield

You can scan one of your repositories for secrets with the following command:

```
ggshield secret scan repo /path/to/your/repo
```

You can also run `ggshield -h` to get help on the CLI.

Go further with ggshield

> If you are looking to configure a CI/CD integration, take a look at our CI/CD Integrations page.

> If you are looking to use GitGuardian at the git hooks level (pre-commit, pre-receive), take a look at our Git hooks documentation page.