GitGuardian [Best practices for managing API keys and other credentials — 2020]

Never store unencrypted secrets in .git repositories

Code repositories, even private ones, are not supposed to store secrets. Code is meant to be shared and duplicated, so your code might perfectly end up in a compromised place or in the public space.

Avoid git add * commands

Wildcard commands can easily capture sensitive files into your .git repository. Add each file by name instead, and use git status to list tracked and untracked files.

ADVANTAGES

- Complete control over what files are committed
- Reduces the risk of unwanted files entering source control

DISADVANTAGES

- Increases time for each commit

Name sensitive files in .gitignore

To prevent sensitive files ending up within git repositories always include a `.gitignore`. It should include files containing environment variables, files generated by another process (such as application logs), and files containing real data (like database extracts).

Don't rely on code reviews to discover secrets

Reviewers are only concerned with the difference between current and proposed states of the code, they do not consider the entire history of proposed changes.

Use automated secrets scanning on repositories

Implement real-time alerting on repositories and gain visibility over where your secrets are with tools like https://gitguardian.com.

ADVANTAGES

- Automated tools are difficult to circumvent and ignore
- Automated tools are fast and accurate
- Ability to detect secrets buried within logs and history
- Continuous scanning of incremental changes

Store secrets safely

There is no silver bullet solution for storing and distributing secrets, multiple solutions may need to coexist.

Use encryption to store secrets within .git repositories

If secrets are needed to be synced within a git repository, encrypt them using services such as git-secret.

ADVANTAGES

- Your secrets are synced

DISADVANTAGES

- You have to store and distribute encryption keys securely
- No audit logs
- Difficult to specify permissions for multiple users
- Hard to rotate encryption keys

Use local environment variables, when feasible

Environment variables exist outside the application and source code.

ADVANTAGES

- Secrets are easier to rotate
- No need to include secrets in source code
- They are less likely to be checked into the repository

DISADVANTAGES

- This approach may not be feasible at scale when secrets need to be synced

Use 'secrets as a service' solutions

Secrets management systems such as Hashicorp Vault are encrypted systems that can safely store your secrets and tightly control access.

ADVANTAGES

- Prevents secrets from sprawling
- Provides audit logs

DISADVANTAGES

- Single point of failure
- All code base must be changed
- Keys giving access must be closely protected

Restrict API access and permissions

Placing restrictions on APIs can help prevent malicious usage.

Default to minimal permission scope for APIs

This is a straightforward application of the Principle of Least Privilege. For example, keep read and write keys separate.

ADVANTAGES

- Limits potential damage

DISADVANTAGES

- Additional upfront setup
- Requires management of additional keys

Whitelist IP addresses when appropriate

IP whitelisting prevents access to services from untrusted sources.

ADVANTAGES

- Prevents attacks from external sources even with secret keys

DISADVANTAGES

- Not always feasible
- Can prevent legitimate information requests
- Needs to be maintained constantly

Use short-lived secrets

Long living or indefinite API keys and secrets while convenient allows attackers indefinite time to discover and exploit systems. Keys without lifespans should also be rotated often.

ADVANTAGES

- Enforces good secret hygiene
- Reduces the risk of long term threats

DISADVANTAGES

- Requires an active secrets management strategy

Don't share secrets via messaging services

While messaging systems like Slack and email are intended to keep messages secure, they are not intended to store sensitive information and can allow attackers to move laterally through systems.