Pushed by Accident – A Mixed-Methods Study on Strategies of Handling Secrets in Source Code Repositories

Alexander Krause*, Jan H. Klemmer†, Nicolas Huaman†, Dominik Wermke*, Yasemin Acar†, ‡, Sascha Fahl*
*CISPA Helmholtz Center for Information Security, †Leibniz University Hannover, †Paderborn University, ‡The George Washington University
{alexander.krause, dominik.wermke, sascha.fahl}@cispa.de, {klemmer, huaman}@sec.uni-hannover.de, yasemin.acar@uni-paderborn.de

Are developers well-prepared to handle secrets in source code without leaking them to the public?

Motivation

The State of Secrets Sprawl 2023
GitGuardian, https://s.gwdg.de/o3Z7dz

The amount of code secret leaks on GitHub increased by 10M (67%) within one year (2021 to 2022).

"5.5 commits out of 1,000 exposed at least one secret (+50%)"

"1 in 10 authors exposed a secret in 2022"

Research Questions

1. How widespread is code secret leakage among developers?
2. What are secret leakage prevention approaches, and what are developers experiences?
3. What are developers' experiences with code secret leakage incidents?
4. What are developers' experiences with code secret remediation techniques and tools?

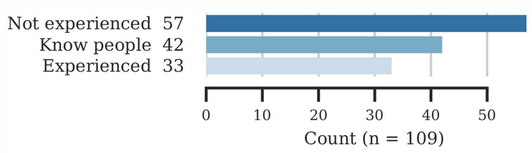
Mixed-Methods Study

- Survey with 109 developers about their experiences with code secret leakage
  - 50 freelancers from Upwork
  - 59 developers from GitHub
- Followed up by 14 semi-structured interviews with GitHub developers who experienced secret leakage to gather in-depth insights

Selected Survey Findings

How widespread is code secret leakage?

- 30.3% of our survey respondents encountered code secret leakage



- We discovered 18 approaches in total to prevent and remediate code secret leakage

Approach	Description	#	%
Prevention			
Externalize Secrets	Separation of code secrets and committed code so that secrets are loaded at runtime, e.g., storing secrets on a central server or secret management system, using environment variables or files [Hashicorp Vault, Azure Key Vault, AWS Vault, *Password, KeePass, Doppler, python-decouple, GitLab CI, GitHub CI, Travis CI]*	60	55.0%
Block Secrets	Prevent code secrets to be contained in code, config, or any other files or prevent including them in publicly available source code repositories, for example, usage of *.gitignore* files, minimizing secret usage in general or use none, remove secrets from version control before publishing to repository	32	29.4%
Encrypted Secrets	Use encryption to store secrets securely within source code repositories [git-secret, git-crypt, SOPS, GPG, kube-seal]*	30	27.5%
Restrict access	Limit the scope of entities including systems and users with access to code secrets, e.g., by user management, policies, role-based access control	19	17.4%
Monitoring	Regular scanning for code secrets and leaks both locally and remote e.g., using secrets scanners in CI/CD pipelines or pre-commit hooks, or review which entities have/had access [SonarQube, Checkmarx, GitGuardian, AWS Cloud Trail]*	16	14.7%
Education & Awareness	Raise awareness for code secret leakage and educate developers how to handle code secrets, e.g., coding guide, best practices wiki	9	8.3%
Other	Miscellaneous other approaches named by participants, not limited to secret handling	8	7.3%
Rotation	Use short-lived secrets, rotate them periodically [Doppler]*	6	5.5%
Code & Secret Reviews	Manual code reviewing which also focus on code secrets; four or more eyes principle to approve code changes	4	3.7%
Remediation			
Renew or Revoke Secret	Invalidate leaked code secrets to prevent any future misuse [Doppler]*	59	54.1%
Cleanup VCS History	Remove leaked secrets from VCSs whole history, e.g., by rewriting the history, clean caches, or reinitialize the whole repository [BFG Repo Cleaner]*	19	17.4%
Analyze Leak	Analysis and forensics on the code secret leak to identify root causes or how the leak was exploited, e.g., by auditing logs or consulting security experts,	17	15.6%
Removal from Source Code	Remove leaked secrets from the current code base. This doesn't include version history, caches or similar	12	11.0%
Notify Concerned Roles	Inform stakeholders affected or involved in the leak, e.g., security team, management, customers, providers, authorities	8	7.3%
Access Management	Re-evaluation of access control concepts and applying more restrictive access management if needed	6	5.5%
Retract Repository	Delete public repositories affected by the leak or make them private, possibly temporarily until remediation is completed	5	4.6%
Systemic Consequences	Applying consequences due to the secret leak, e.g., new processes, specific education, removal of team members or clients	3	2.8%
Server Operations	Actions taken to remediate secret leakage in running software, e.g., by backuping systems, or pruning and re-initializing servers	2	1.8%

* Tools our participants used.

Selected Interview Findings

- *"Code secret leakage happens four or five times a year"*— I5
- *"[The Leak] was probably out there for a couple of weeks. So, yes, that was not amazing."*— I11
- *"We were a startup, [we didn't had any prevention approaches in place], we took all the measures after the secret leakage."*— I2
- *"I didn't ask anyone, I knew what to do, I just responded directly."*— I5
- *"Most of the time, [the secret scanner] just raises warnings about some secrets that are really supposed to be in the code and you have to manually exclude it from being scanned."*— I13

Selected Recommendations

- Prevention: Using a combination of different approaches to decrease the likelihood of code secret leakage.
  - Externalize and block secrets from VCS (e.g., using environment variables)
  - Apply monitoring to detect leaks (e.g., secret scanners)
- Remediation: Always renew or revoke leaked secrets.
  - Analyze leak and revise access management
  - Notify concerned roles (e.g., customers or management)

More Findings

Will be presented at USENIX'23
See you there!
Website & Replication Package
https://publications.teamusec.de/2023-usenix-codesecrets/