Pushed by Accident

A Mixed-Methods Study on Strategies of Handling Secrets in Source Code Repositories

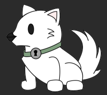
Alexander Krause[C], Jan H. Klemmer[*], Nicolas Huaman[*], Dominik Wermke[C], Yasemin Acar[†, ‡], and Sascha Fahl[C]

[C] CISPA Helmholtz Center for Information Security, Hannover, Germany

[*] Leibniz University Hannover, Hannover, Germany

[†] Paderborn University, Paderborn, Germany

[‡] The George Washington University, Washington, DC, USA



#teamusec

Developers Must Provide and Handle Secrets Securely

- Version control systems (VCSs) are an essential technology for collaborative software development
- Git-based platforms such as GitHub or GitLab are the most used source code sharing platforms
- Developers need to provide secrets to e.g., deploy software, automate interactions with third parties, or handle authentication

Credentials, Authentication Tokens, or Secret Encryption Keys

Secrets are highly sensitive, e.g.,

- credentials e.g.,

  ```
  user=admin, password=secretpwd
  ```

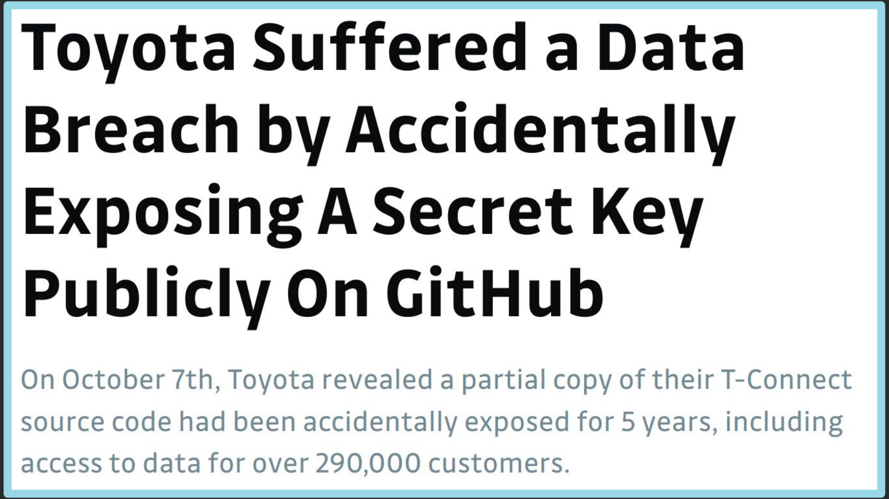
- authentication tokens e.g.,

  ```
  JalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
  ```

- secret encryption keys e.g.,

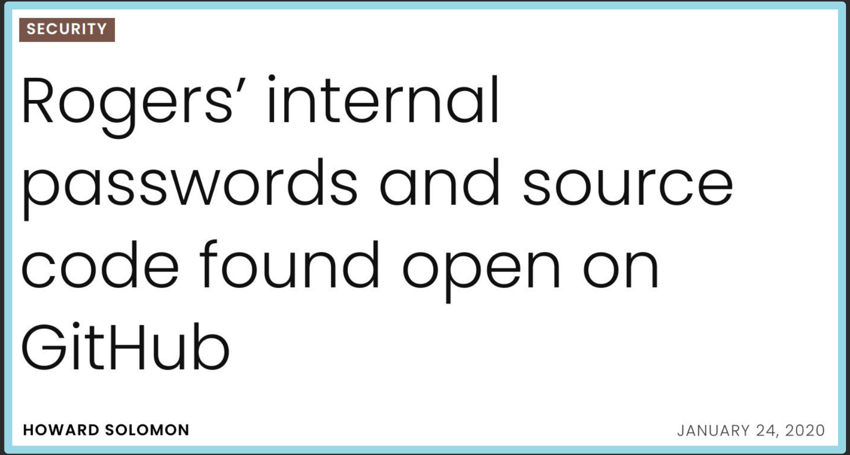
  ```
  -----BEGIN OPENSSH PRIVATE
  KEY-----b3BlbnNzaC1rZXktdjEAAAAACmFlczI1Ni1jdHIAAAAGYmNyeXB0AAAA
  GAAAABBjTZYaSZ....
  ```

Even the Big Players Fail



Toyota Suffered a Data Breach by Accidentally Exposing A Secret Key Publicly On GitHub

On October 7th, Toyota revealed a partial copy of their T-Connect source code had been accidentally exposed for 5 years, including access to data for over 290,000 customers.

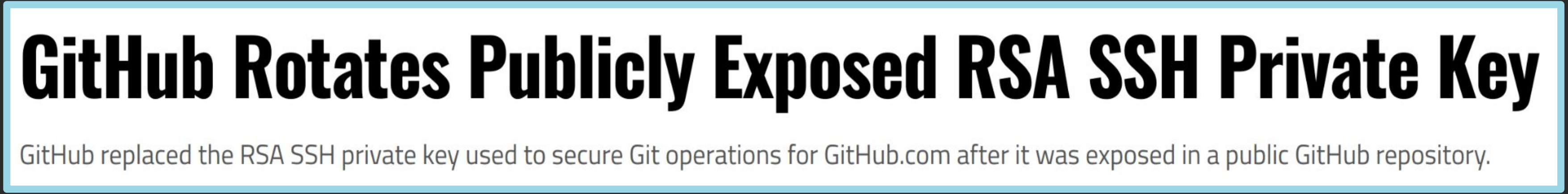


SECURITY

Rogers' internal passwords and source code found open on GitHub

HOWARD SOLOMON

JANUARY 24, 2020

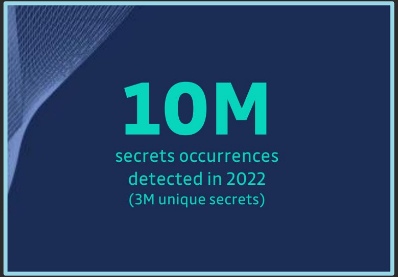


GitHub Rotates Publicly Exposed RSA SSH Private Key

GitHub replaced the RSA SSH private key used to secure Git operations for GitHub.com after it was exposed in a public GitHub repository.

Code Secret Leakage Becomes More and More Significant

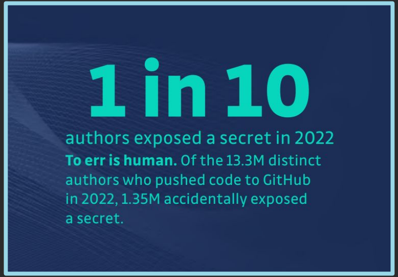
GitGuardian: The State of Secrets Sprawl 2023 [1]



10M

secrets occurrences detected in 2022

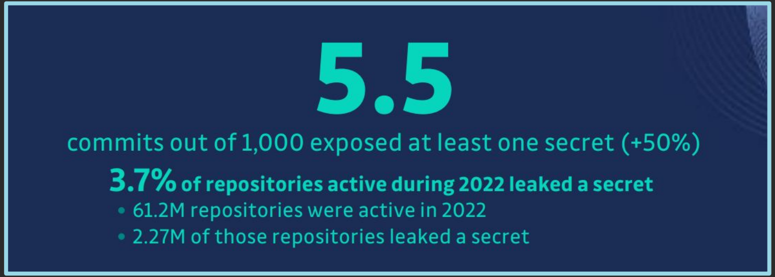
(3M unique secrets)



1 in 10

authors exposed a secret in 2022

To err is human. Of the 13.3M distinct authors who pushed code to GitHub in 2022, 1.35M accidentally exposed a secret.



5.5

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

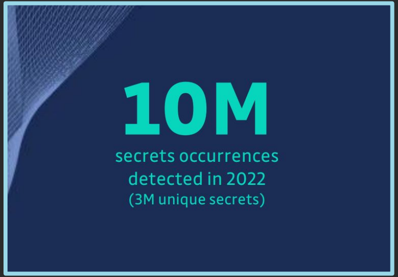
3.7% of repositories active during 2022 leaked a secret

- 61.2M repositories were active in 2022
- 2.27M of those repositories leaked a secret

[1] GitGuardian
https://www.gitguardian.com/files/the-state-of-secrets-sprawl-report-2023

Code Secret Leakage Becomes More and More Significant

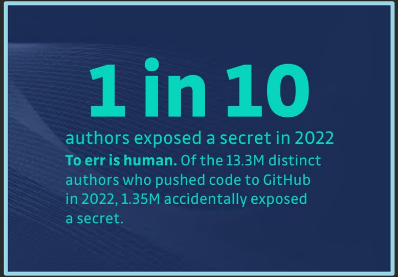
GitGuardian: The State of Secrets Sprawl 2023 [1]



10M

secrets occurrences detected in 2022

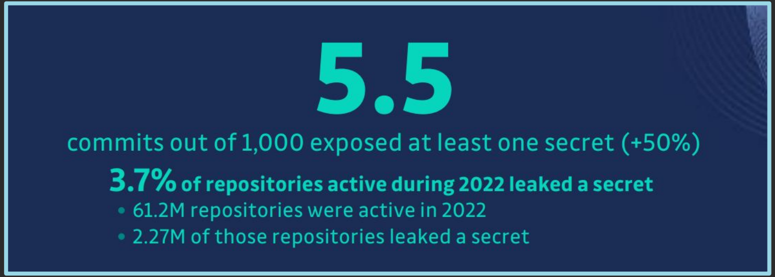
(3M unique secrets)



1 in 10

authors exposed a secret in 2022

To err is human. Of the 13.3M distinct authors who pushed code to GitHub in 2022, 1.35M accidentally exposed a secret.



5.5

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

3.7% of repositories active during 2022 leaked a secret

- 61.2M repositories were active in 2022
- 2.27M of those repositories leaked a secret

Meli et al. presented a large-scale measurement study on secret leakage in public GitHub repositories, finding more than 100,000 repositories with leaked secrets. [2]

[1] GitGuardian https://www.gitguardian.com/files/the-state-of-secrets-sprawl-report-2023

[2] Meli et al. “How Bad Can It Git? Characterizing Secret Leakage in Public GitHub Repositories”, NDSS, 2019

Research Questions

Research Questions

RQ1 How widespread is code secret leakage among developers?

Research Questions

RQ1 How widespread is code secret leakage among developers?

RQ2 What are secret leakage prevention approaches, and what are developers experiences?

Research Questions

RQ1 How widespread is code secret leakage among developers?

RQ2 What are secret leakage prevention approaches, and what are developers experiences?

RQ3 What are developers’ experiences with code secret leakage incidents?

Research Questions

RQ1 How widespread is code secret leakage among developers?

RQ2 What are secret leakage prevention approaches, and what are developers experiences?

RQ3 What are developers’ experiences with code secret leakage incidents?

RQ4 What are developers’ experiences with code secret remediation techniques and tools?

Methodology

Mixed-Methods Study

Online Developer Survey

Online Developer Interviews

Online Developer Survey

Recruitment

- n = 109 developers
  - 50 from Upwork
  - 59 from GitHub

Online Developer Survey

Recruitment

- n = 109 developers
  - 50 from Upwork
  - 59 from GitHub

Content of the Questionnaire

- Source code management
- Experience with secret information
- Threat model for secret information
- Secret leakage remediation approaches
- Secret leakage prevention approaches
- Demographics

Online Developer Survey

Recruitment

- n = 109 developers
  - 50 from Upwork
  - 59 from GitHub

Goals

- Identify the extent of code secret leakage
- Identify code secret leakage prevention & remediation approaches

Content of the Questionnaire

- Source code management
- Experience with secret information
- Threat model for secret information
- Secret leakage remediation approaches
- Secret leakage prevention approaches
- Demographics

Online Developer Interviews

Recruitment

- n = 14 developers from GitHub
  - Developers must have experienced code secret leakage

Online Developer Interviews

Recruitment

- n = 14 developers from GitHub
  - Developers must have experienced code secret leakage

Content of the Interview Guide

- Code secret leakage incidents
- Secret leakage remediation approaches
- Secret leakage prevention approaches

Online Developer Interviews

Recruitment

- n = 14 developers from GitHub
  - Developers must have experienced code secret leakage

Goals

Identify developers' problems, challenges, and needs with code secret leakage remediation & prevention approaches

Content of the Interview Guide

- Code secret leakage incidents
- Secret leakage remediation approaches
- Secret leakage prevention approaches

Selected Findings

Online Developer Survey

Online Developer Interviews

A Third Reported Secret Leakage

30.3%

of our survey respondents reported first-hand experience with secret leakage in their projects.

Places and Types of Leaked Secrets

- Places of leak
  - Public repositories
  - Restricted repositories (internal)
  - Code sharing platforms like Pastebin or GitHub gist
  - GitHub workflow logs

Places and Types of Leaked Secrets

- Places of leak
  - Public repositories
  - Restricted repositories (internal)
  - Code sharing platforms like Pastebin or GitHub gist
  - GitHub workflow logs
- Types of leak
  - Configuration files
  - API tokens
  - Access keys
  - Database passwords

> “[I was] pushing the commits to GitHub and when I pushed the remote repository, I found that my [password manager database] has gone into GitHub without me wanting it to go to there.”— I10

Detection and Impact of Incidents

- Leak Detection
  - GitHub secret scanner
  - Randomly or by others
  - Incidents discovered lately

> “It was probably out there for a couple of weeks. So, yes,
> that was not amazing.”— I11

Detection and Impact of Incidents

- Leak Detection
  - GitHub secret scanner
  - Randomly or by others
  - Incidents discovered lately
- Impact
  - For the company or software team
    - Additional workload remediating the leak
    - Financial or reputational damage
  - External stakeholders
    - Data loss or data theft

> “It was probably out there for a couple of weeks. So, yes, that was not amazing.”— I11

Root Causes of Code Secret Leakage Incidents

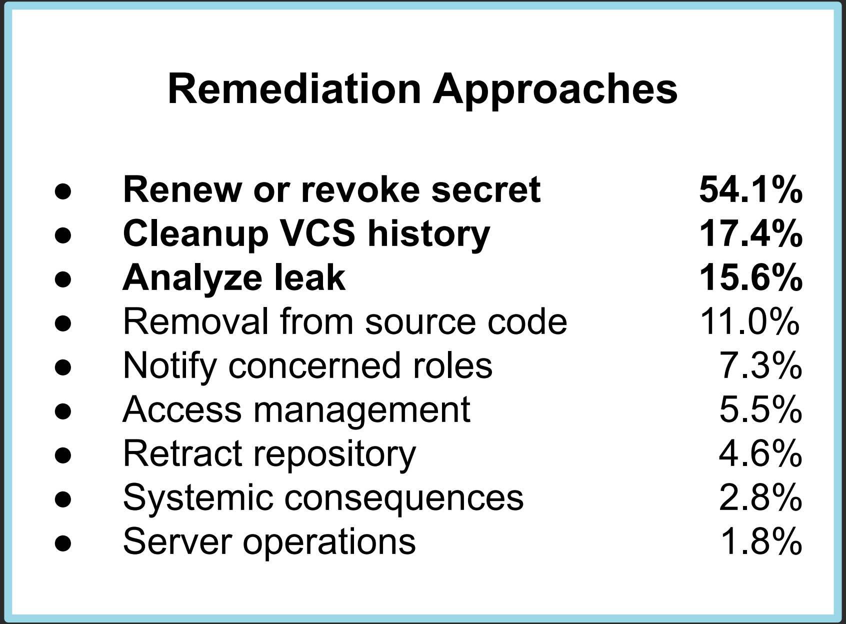
- Root Causes
  - No awareness of new developers in a team
  - No use of any prevention approaches before an incident happened
  - No use or misuse of the .gitignore file
  - Use of hard-code secrets in source code
  - Developers’ threat models and secret access process

> “Even with all the technology [. . .] to prevent secret leakage, the biggest contributor to secret leakage is the human factor, or negligence.” — I2

> “Really just any time you ask, you’ll just get access to whatever you want.” — I6

Most Survey Respondents Renewed or Revoked Leaked Secrets

What approaches did our survey respondents use to remediate code secret leakage?



Remediation Approaches

Renew or revoke secret	**54.1%**
Cleanup VCS history	**17.4%**
Analyze leak	**15.6%**
Removal from source code	11.0%
Notify concerned roles	7.3%
Access management	5.5%
Retract repository	4.6%
Systemic consequences	2.8%
Server operations	1.8%

Challenges Remediating Code Secret Leakage

- The process of remediation is cumbersome
- Complicated incident response process that was never used before
- Being not aware of all the consequences caused by the leak
- The need to select, learn, and apply different or multiple remediation approaches would be too complex and time-consuming

Survey Respondents Externalized, Blocked, and Encrypted Secrets

What approaches did our survey respondents use to <u>prevent</u> code secret leakage?

Prevention Approaches

Externalize secrets	**55.0%**
Block secrets	**29.4%**
Encrypted secrets	**27.5%**
Restrict access	17.4%
Monitoring	14.7%
Education & awareness	8.3%
Other	7.3%
Rotation	5.5%
Code & secret reviews	3.7%

Factors that Influence the Use of Prevention Approaches

- Participants reported approaches have to be:
    - Effective
    - Efficient
    - Secure
    - Usable
    - Compliant with company requirements

Challenges When Preventing Code Secret Leakage

- Cost and time constraints
  - Time to set up a new approach
  - Even more time is required to train all involved developers using the approach
  - Adopting new approaches to existing projects often requires refactoring work

Challenges When Preventing Code Secret Leakage

- Cost and time constraints
  - Time to set up a new approach
  - Even more time is required to train all involved developers using the approach
  - Adopting new approaches to existing projects often requires refactoring work
- Awareness and education

> “Someone was doing something **off the books** [. . .]:
>
> They were just **creating another repository** [. . .] **not within the organization** but maybe just under a personal account or something.
>
> Those you **can’t really fix with tooling,** at the end of the day, those are just **people's problems** [. . .] and we can **fix that through training** [. . .][**or**] **policy**.”— I6

Selected Recommendations

For Developers and Service Providers

Recommendations for Developers

Combination of different prevention approaches to decrease the likelihood of code secret leakage

- Externalize secrets e.g., using environment variables
- Block secrets from repositories, e.g., using .gitignore files
- Monitoring e.g., using secret scanners
- Encrypt secrets that need to be shared through the repository

Recommendations for Developers

Typical steps that should always be taken to remediate code secret leakage

- Renew or revoke the leaked secret
- Analyze the leak
- Revise the access management using the results from the leak analysis
- Notify the concerned roles
- In addition
  - Removal from source code
  - Cleaning up the VCS history

Recommendations for Service Providers

- Improving online information and documentation

- Provide and expand secret scanning

Alexander Krause

CISPA Helmholtz Center for Information Security

Hannover, Germany

✉ alexander.krause@cispa.de

𝕏 @akrause_de

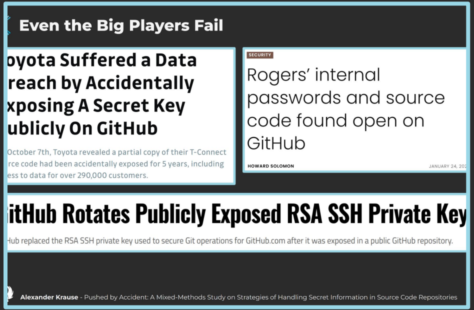
https://akrause.de

Pushed by Accident

A Mixed-Methods Study on Strategies
of Handling Secrets in Source Code Repositories

Alexander Krause[C], Jan H. Klemmer[*], Nicolas Huaman[*], Dominik Wermke[C], Yasemin Acar[†,‡], and Sascha Fahl[C]

[C]CISPA Helmholtz Center for Information Security, Hannover, Germany
[*]Leibniz University Hannover, Hannover, Germany
[†]Paderborn University, Paderborn, Germany
[‡]The George Washington University, Washington, DC, USA



Even the Big Players Fail

oyota Suffered a Data reach by Accidentally xposing A Secret Key ublicly On GitHub

October 7th, Toyota revealed a partial copy of their T-Connect rce code had been accidentally exposed for 5 years, including ess to data for over 290,000 customers.

SECURITY

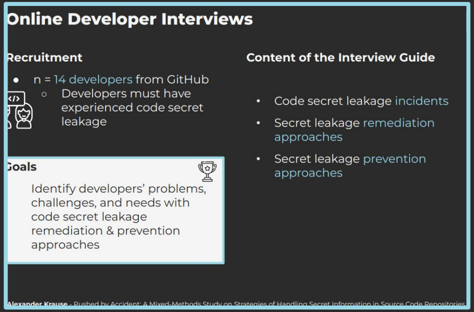
Rogers' internal passwords and source code found open on GitHub

HOWARD SOLOMON JANUARY 24, 202

itHub Rotates Publicly Exposed RSA SSH Private Key

Hub replaced the RSA SSH private key used to secure Git operations for GitHub.com after it was exposed in a public GitHub repository.

Alexander Krause - Pushed by Accident: A Mixed-Methods Study on Strategies of Handling Secret Information in Source Code Repositories



Online Developer Interviews

Recruitment

- n = 14 developers from GitHub
  - Developers must have experienced code secret leakage

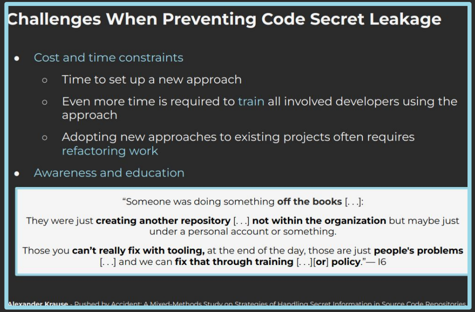
Goals

Identify developers' problems, challenges, and needs with code secret leakage remediation & prevention approaches

Content of the Interview Guide

- Code secret leakage incidents
- Secret leakage remediation approaches
- Secret leakage prevention approaches

Alexander Krause - Pushed by Accident: A Mixed-Methods Study on Strategies of Handling Secret Information in Source Code Repositories



Challenges When Preventing Code Secret Leakage

- Cost and time constraints
  - Time to set up a new approach
  - Even more time is required to train all involved developers using the approach
  - Adopting new approaches to existing projects often requires refactoring work
- Awareness and education

> "Someone was doing something **off the books** [. . .]:
>
> They were just **creating another repository** [. . .] **not within the organization** but maybe just under a personal account or something.
>
> Those you **can't really fix with tooling,** at the end of the day, those are just **people's problems** [. . .] and we can **fix that through training** [. . .][**or**] **policy**."— I6

Alexander Krause - Pushed by Accident: A Mixed-Methods Study on Strategies of Handling Secret Information in Source Code Repositories