



Secret Detection & Responsible Disclosure

Challenges & Lessons Learnt



Gaetan FERRY & Guillaume VALADON

GitGuardian

01

Who we are

Who we are

\$ whoami



Guillaume

Cybersecurity Researcher

editor-in-chief of the **MISC**
magazine

Scapy maintainer

previously at **Quarkslab**,
ANSSI...



Gaetan

Cybersecurity Researcher

former researcher **@Sonar**

Synacktiv red teamer for 7
years

Secrets Security 🤝 NHI Governance

0% LEAKED SECRETS
Secrets Security

INTEGRATED SOURCES

Package Registries

Container Registries

Code Repositories

CI/CD Pipelines

Messaging Systems

Ticketing Systems

Logs

Knowledge Database



THE STATE OF **Secrets Sprawl** 2025



Data analysis by **GitGuardian**

SECURITY

100% MANAGED IDENTITIES & SECRETS

NHI Governance

INTEGRATED SOURCES

IAM Cloud Providers

Secret Managers

Cloud Infrastructure

Deployment Tools

Containers

Third-Party Apps



NHI Governance

check it out!

Secrets Lifecycle

Discovery, Rotation, Ownership

02

Detecting secrets in open sources

Open-Sources

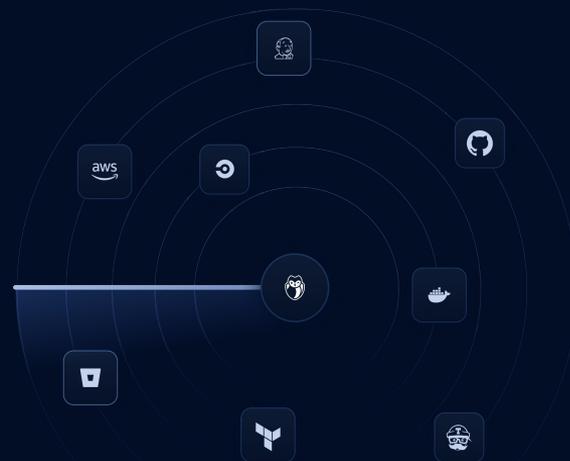
The cloud world has a lot of places for secrets to leak

- The cloud generation wants everything open and connected
- We host our data and applications over Internet (authz?)
- Open-Source is great so let's do Open-Source!

Your source code goes on GitHub

Your containers go on Docker Hub

- The frontier between public and private becomes thin!



Attackers Are Looking For Secrets

APPLICATION SECURITY

PyPI Packages Found to Expose Thousands of Secrets

GitGuardian discovered roughly 4,000 secrets in nearly 3,000 PyPI packages, including Azure, AWS, and GitHub keys.



By [Ionut Arghire](#)
November 14, 2023

[Home](#) » [Security](#)

Websites exposing over a million secrets, leaving visitors at risk

Last updated: 29 May 2024

[Home](#) > [News](#) > [Security](#) > [Nearly 12,000 API keys and passwords found in AI training dataset](#)

Nearly 12,000 API keys and passwords found in AI training dataset

By [Ionut Ilascu](#)

March 2, 2025 10:23 AM 0

Open-Sources

A secret leaks when it goes where it should not™

- Public leaks are the worse but private leaks also count (a lot)
- The “private” part gives a false sense of security

Leaking secrets is very easy!

- Hardcoding secrets is far easier than handling them safely!
- Private things will go public, PoCs will go to production
- Developers leak in personal projects

Closer to production means leakier

- Secrets are mostly needed in production
- Production > container > artifacts > source code

35%

Private repos leak

5%

For public repos

> 1M

Secrets on
Docker Hub

100,000

Found valid

Detection strategies - It's easy

Detecting secrets is as easy as a regex

- Cloud and API providers implement prefixed secrets
- They are detected with a simple regular expressions



- There are a lot of cloud providers with such keys
 - GitGuardian supports patterns for more than 450 secret formats (easy)
 - There are still more to add
 - Maintenance can become an issue

Detection strategies - ~~It's easy~~

There is a lot more than just prefixed tokens

- Some providers stick to non-formatted secrets (e.g. random 256 bits strings)
- Username/Passwords, MAC keys, ...
- In 2024, 58% of detected secrets were generics

Secrets by detector nature



Data analysis by GitGuardian

Share

Detecting secrets in open sources

Detection strategies - ~~It's easy~~

There is a lot more than just

- Some providers stick to non-formatted secrets (e.g. ...)
- Username/Passwords, MAC keys, ...
- In 2024, 58% of detected secrets were generics



Detecting generics requires efforts

Entropy based detection: A classic with a lot of false positives

Keyword based detection: Generally unreliable with lots of false positives

Usage based detection: Efficient but costly

Context based: Similar to keywords but slightly better



Data analysis by GitGuardian

Share

The problems with scanning the cloud

Volume

- Cloud sources represent TBs of data due to their adoption
 - In 2024: 1.3B commits on GitHub, 15M Docker images
 - The more data, the more secrets...

GitHub

23.8M

DockerHub

1.2M

- ...the more corner cases

The problems with scanning the cloud

False positives

- Many reasons for false positives:
 - Testing credentials, dummy values, placeholders, etc
 - People testing secret detection
 - Already revoked secrets
 - Generics !!

```
a1_username <- "t[REDACTED]m"  
a1_password <- rstudioapi::askForPassword("A1 Password: ")
```

```
sha: "g[REDACTED]g"  
node_id: "C[REDACTED]:"  
commit:  
author:  
  name: "eJwt[REDACTED]nP+jhP0oECIanCtdQ1vG/  
        HGZSX[REDACTED]h/  
        EoPrp[REDACTED]Iym8GR/  
        YGmkaQlyBAf4lT5i1KaDmJYtn8x9gTk6A"  
email: "s[REDACTED].com"  
date: "2025-02-18[REDACTED]"
```

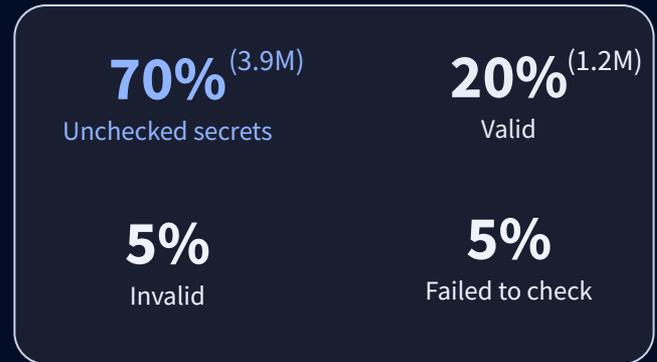
```
records = ["E0G0S16406", "MS=ms51271079", "google-site-verification=yzqA25_KO_rZYL4b-UxXDXI7x-ZWUKYHjtyxyVILvqU", "adobe-idp-site-  
verification=12745d082f0122d00a6ac369ec9edff9a2b54fd6e569dee485e26119cd5523ee", "dn0QxuQ4AjklbhQTyFA+nWix2yM5DE7xy0qbZgb1afVWAT/TczyzQZ0q7xkIsvcroCHw8YuEw/  
pw2JQGJMaZQQ==", "QuoVadis=22879b0e-362c-40bc-a726-da94acee34ed", "v=spf1 include:u2320754.wl005.sendgrid.net ip4:18.168.37.156/30 -all"]
```

You can't just send alerts for every secret detected

Breaking the wall: validity checks

If it works, it works. If not, who knows

- Specifics can be tested. **Especially in the cloud!**
 - GitHub tokens, AWS Keys, Azure secrets, etc
- Allows filtering out False Positives with 100% accuracy
- Also reduces the volume of secrets because...
- ... lots can not be checked
 - Generics
 - Unreachable/unknown hosts



Based on secrets found on GitHub in 2024

Breaking the wall: validity checks

If it works, it works. If not, who knows

- Specifics can be tested. **Especially in the cloud!**
 - GitHub tokens, AWS Keys, Azure secrets, etc
- Allows filtering out False Positives with 100% accuracy
- Also reduces the volume of secrets because...
- ... lots can not be checked
 - Generics
 - Unreachable/unknown hosts

**Still 100,000 secrets/month.
More than we can handle.**



Based on secrets found on GitHub in 2024

Breaking the wall: make the most of the secret

If it works, make it talk!

- Secrets are used to connect to services, services expose data
 - Extract metadata from the secrets' environment
 - Collect scopes, permissions, ownership information
- Help with the attribution of the secret ownership (more on that later)
 - Filter out personal / non corporate secrets
- Limitations
 - Some services are light in information (AWS, GCP)
 - Balance between collected information and impact

```
$ curl -H "Authorization:
Bearer ghp_AF***pH"
https://api.github.com/user
{
  "login": "H*****m",
  "id": 1***2,
  "type": "User",
  "user_view_type": "public",
  "site_admin": false,
  "name": "Z***",
  "company": "*****",
  "blog": "",
  "location": null,
  "email": a*****m,
  "hireable": null,
  "bio": "*****",
```

Detecting secrets in open sources

Finding who to alert

You can't just find secrets and ignore them

- Finding a secret's owner can be more or less difficult
- Sources' metadata (git committer email, Docker Hub account)
 - Powers [the good samaritan program](#) since day 1
 - Automatic email to the leaker individual
- Attachment to a company is still necessary when possible
 - Secret data extraction from services
 - Secret OSINT dark magic
 - Secret correlation

Detecting secrets in open sources

Finding who to alert

You can't just find secrets and ignore them

- Finding a secret's owner can be more or less difficult
- Sources' metadata (git committer email, Docker Hub account)
 - Powers [the good samaritan program](#) since day 1
 - Automatic email to the leaker individual
- Attachment to a company is still necessary when possible
 - Secret data extraction from services
 - Secret OSINT dark magic
 - Secret correlation

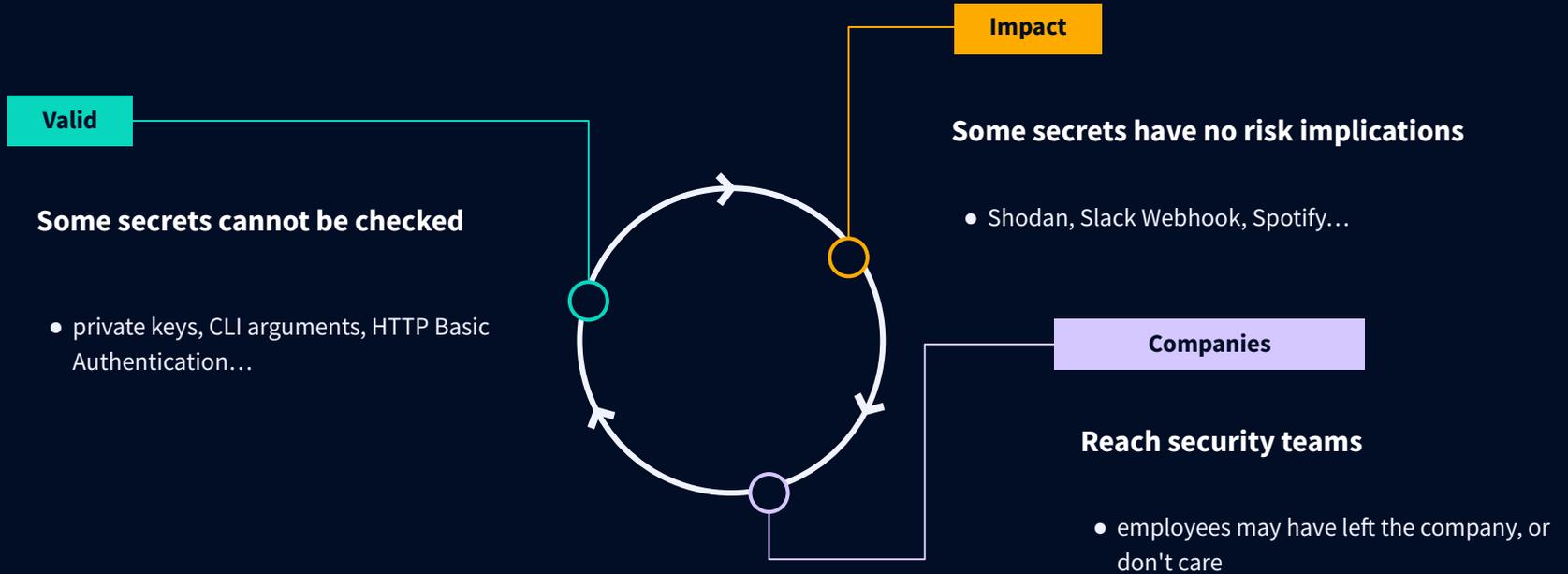
**And that's when the real fun
begins**

03

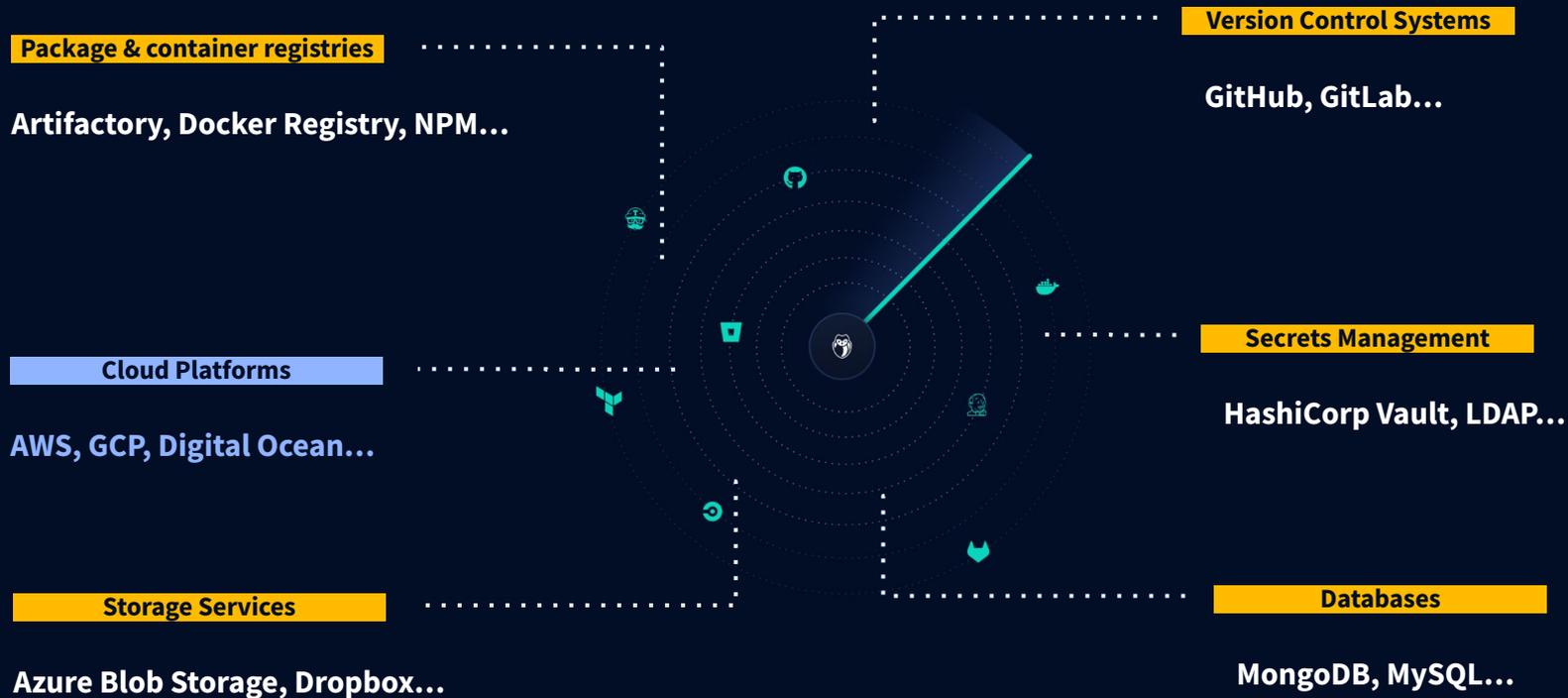
Responsible Disclosure Experiences

Scope

Corporate valid secrets with offensive impact



Secrets Categories Exploited by Attackers



MITRE ATT&CK Paths Examples

Version Control System

01

T1078 - Valid Accounts

02

T1552 - Unsecured Credentials

Cloud Platform

01

T1078 - Valid Accounts

02

T1578 - Modify Cloud Infrastructure

03

T1496 - Resource Hijacking

Statistics

6 months

26 disclosures

4 acknowledged

14 resolved

70%

of 2022 secrets still
valid

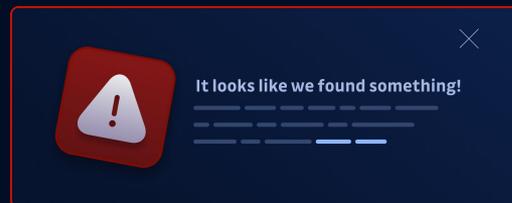
2%

of 2024 secrets related
to Cloud Providers

Typical Negative Interactions

Some are similar to generic disclosures

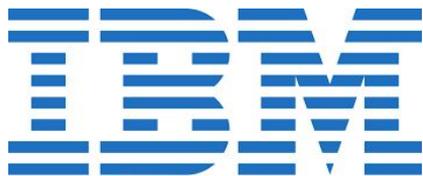
- *don't care*
- *refusing to make findings public*
- *already known*
 - remember that we disclose **valid secrets**
- *no acknowledgement / no answer*
 - escalating to National Security Agencies hardly helps



Remediation is simple & Risk Easily Evaluated

Halls of Fame

Indication that leaked secrets are underestimate?

The Oracle logo, consisting of the word "ORACLE" in a bold, red, sans-serif font.The IBM logo, consisting of the letters "IBM" in a blue, striped, sans-serif font.The Liferay logo, featuring a blue square icon with a white grid pattern to the left of the word "Liferay" in a black, sans-serif font.

- most disclosures not public
 - despite existing bounty programs

Zoo of Unexpected Interactions

- *vendor cannot reproduce*
 - <https://github.com/GitGuardian/ggshield> is open-source
- secret leaks not in Bug Bounty scope
 - reports are stopped during triaging
- secret leaks not a product issue
- no answer but internal ticket communicated to a GitGuardian sales representative

04

Taking actions

What do do next?

What to do next?

Include Secrets Leaks as a Key Security Threat

- you are probably leaking secrets
 - without realizing it
- exposures come from a wide range of sources
- consider secret leaks as part in your risk assessment
- audit your perimeter for hard-coded secrets



Prevention is more cost-effective than dealing with a breach!

What to do next?

Hardening Recommendations

Some helpful mitigations.

They could be used to [contain a leak](#), and to understand what to improve.

01

Enforce 2FA

02

Configure IP Access List

03

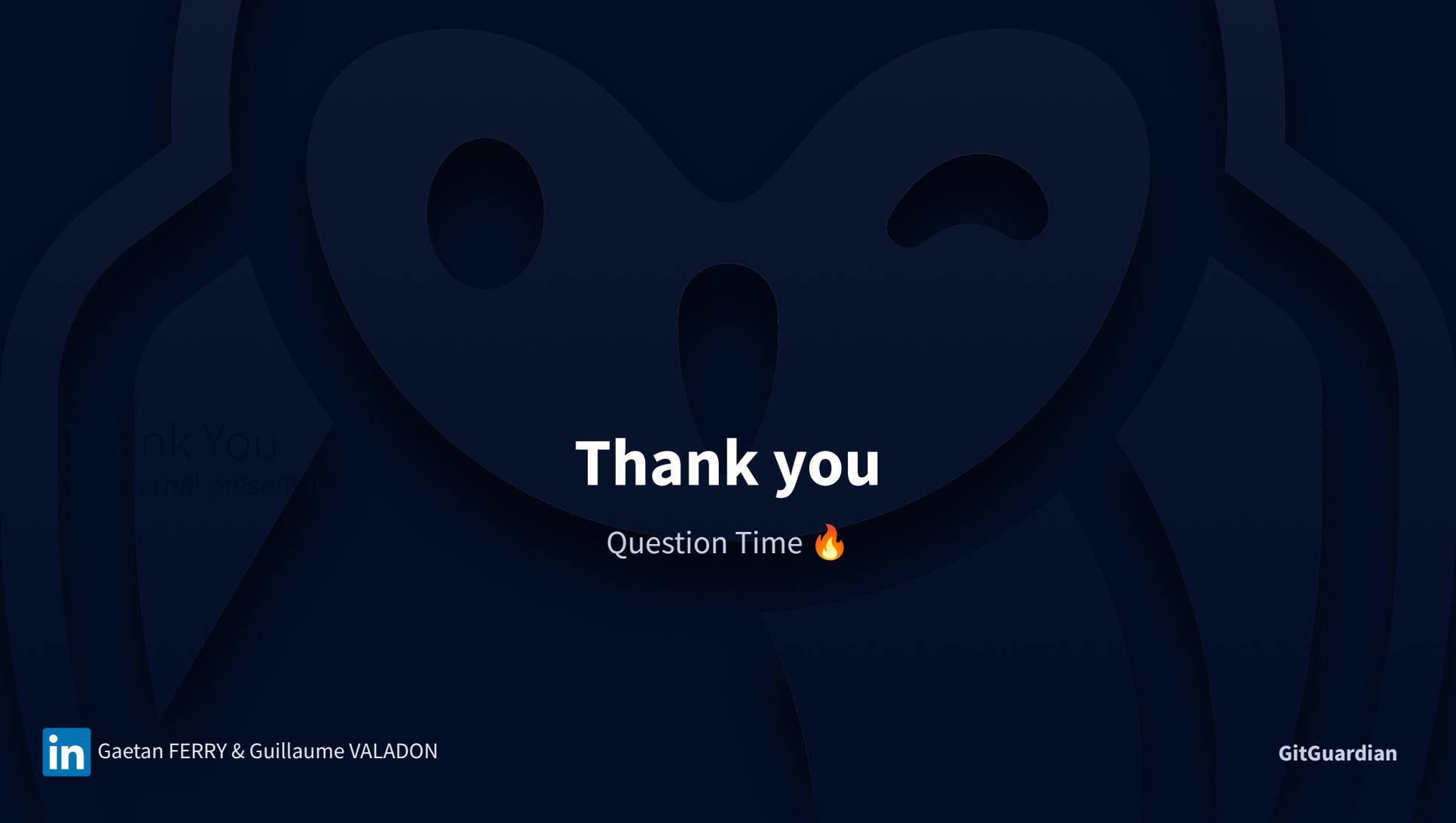
Least privilege & reduced scopes

04

Short Lived Tokens

05

Monitor API tokens Usage



Thank you

Question Time 🔥



Gaetan FERRY & Guillaume VALADON

GitGuardian