Keeping Secrets Out of Your Pipeline

Gene Gotimer
DevOps Engineer at Praeses, LLC
@OtherDevOpsGene

The State of Secrets Sprawl Report

https://www.gitguardian.com/state-of-secrets-sprawl-report-2025

Secrets

What are secrets?

- Passwords
- Credentials
- API keys
- Signing keys
- SSH keys
- Access tokens

The scenario

- Working towards continuous delivery and DevOps
- Writing automated deployment code
- And writing automated tests
- So, you have credentials locally in your environment
- Pipeline needs those credentials as well

Oops!

- File with the credentials gets included with source control
- "*But I thought that repo was private*"
- Keys get hard-coded, "*but only while we are testing*"
- Permissions wide open so we don't need credentials in dev
- Written to an issue ticket
- Packaged with a release
- Included in a Docker image

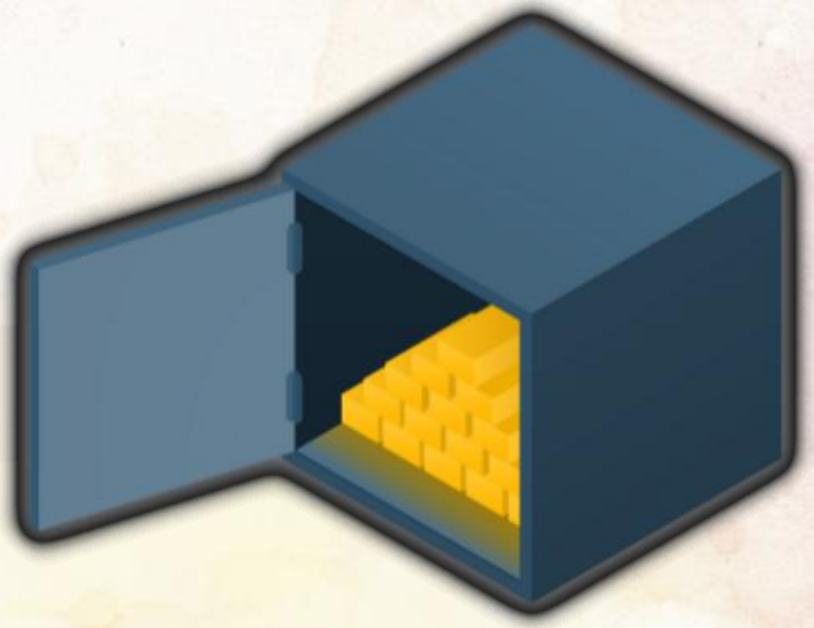
Detecting committed secrets

Some tools to try

TruffleHog

https://github.com/trufflesecurity/trufflehog

OWASP WrongSecrets



https://github.com/OWASP/wrongsecrets

Using TruffleHog

```
$ trufflehog github --repo https://github.com/OWASP/wrongsecrets \
    --issue-comments --pr-comments


$ trufflehog github --org OtherDevOpsGene \
    --issue-comments --pr-comments --gist-comments


$ trufflehog docker --image webgoat/webgoat-8.0:latest


$ trufflehog --help
```

Found some. So now what?

1. Revoke the credentials
   - Do not bother finding out first if anyone has seen them
   - Assume they are compromised,
     even when you are 100% sure it couldn't happen
2. Plug the leak
3. Rotate the credentials- https://howtorotate.com/

What about just fixing the repo?

- Don’t

If you insist

```
$ git reset HEAD~ --soft  # or HEAD@2 if it was two commits back

# Make your changes and commit the right stuff

$ git push origin --force
```

Catch secrets before commit

pre-commit hooks

https://pre-commit.com/

.pre-commit-config.yaml

```yaml
repos:
-   repo: https://github.com/trufflesecurity/trufflehog
    rev: v3.92.4
    hooks:
    -   id: trufflehog
        stages: ["commit", "push"]
```

Set up pre-commit hooks

```
$ pip install pre-commit # add -U to upgrade

$ pre-commit install
```

Storing secrets secretly

AWS Secrets Manager

Similar to

Azure Key Vault

GCP Secret Manager

https://aws.amazon.com/secrets-manager/

Avoiding secrets entirely

3 factors of authentication

- Something you know
  - password
- Something you have
  - authenticator app on smartphone
  - smart card
  - physical key
- Something you are
  - fingerprint
  - retinal scan
  - other biometric method

AWS IAM Roles

- Scenario
  - EC2 instance needs to access an RDS database
- Traditional solution
  - Create username and password and store on EC2 instance
- **Better solution**
  - **Create username and password and store in AWS Secrets Manager**
- **Even better solution**
  - **Assign IAM role to EC2 instance that has access to the RDS database**

Honeytokens

Honeypots

- Unused systems that look like valuable targets
- Aren’t referenced or linked to anywhere
- Deliberately vulnerable
- If someone accesses them, they are likely an attacker
  - They can be tracked and/or blocked elsewhere

Honeytokens

- Similar to honeypots
- Unused, but valid-looking credentials, URLs, files, API keys
- Stored somewhere that should be secure
  - private Git repo
  - S3 bucket
  - CI environment variable
  - company Slack
- If someone tries to use them, they are likely an attacker

Honeytoken providers

https://canarytokens.org/

Alternatives:

- GitGuardian ggcanary
  https://github.com/GitGuardian/ggcanary
- SpaceSiren
  https://github.com/spacesiren/spacesiren

Someone found one. So now what?

- You know that your secure location has been compromised
- Anything you thought was secure in there is not
- Check permissions
- Rotate credentials
- Replace the honeytoken

Wrap up

Software Engineer – Praeses

Role Highlights:

- Build, test, and maintain modern web applications across the full stack
- Own end-to-end delivery of complex features and functionality
- Improve existing systems and participate in architecture and design discussions
- Contribute to code reviews and SCRUM activities
- Ensure security, performance, and maintainability

What We're Looking For:

- 5+ years software development experience
- Strong communication and problem-solving skills
- Ruby on Rails, JS frameworks, Mongo/Postgres, Docker/Kubernetes
- Ability to obtain Secret clearance

Bonus Experience:

- CUI/sensitive data, defense contracting, CompTIA Security+

Salary:

- $105,000–$115,000

Full job posting available on LinkedIn:
https://www.linkedin.com/jobs/view/4358396233/

Key takeaways

- Scan your code for secrets.
- Don’t let secrets into your code.
- Assume exposed secrets are compromised and rotate them.
  - Immediately!
- Use a secrets-as-a-service solution.
- Use honeytokens to alert you to private areas being exposed.

Tools

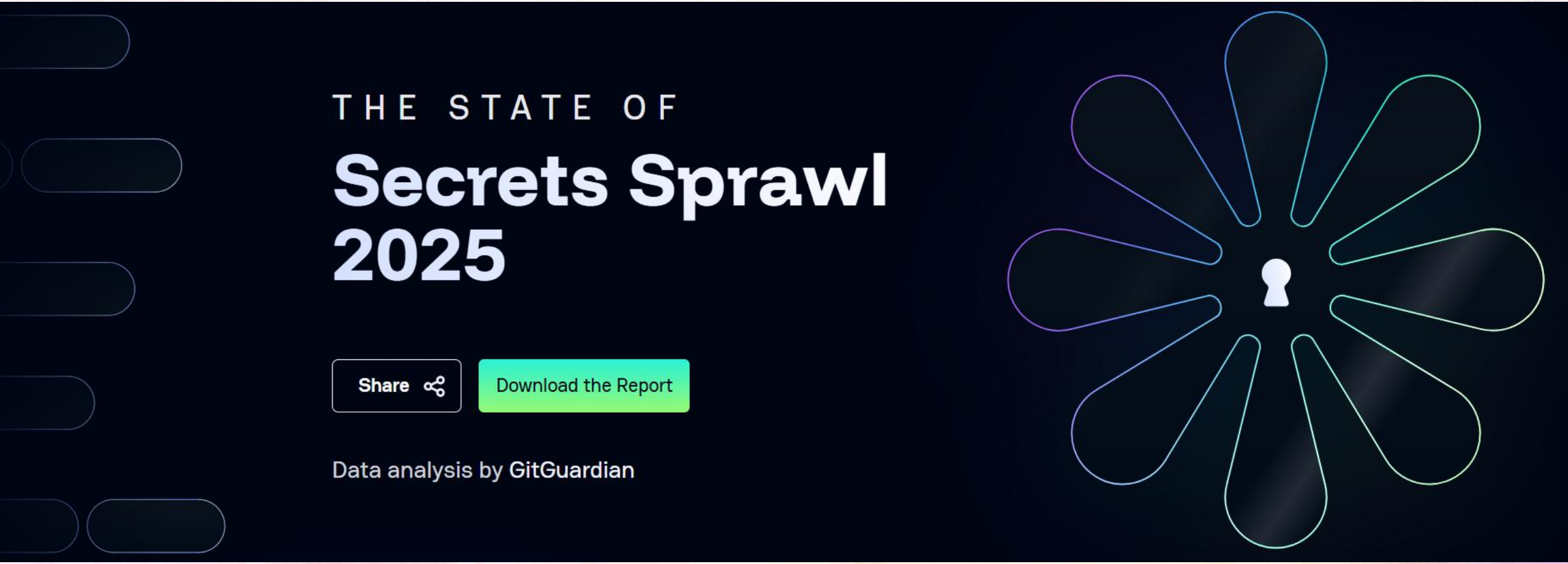
- TruffleHog- https://github.com/trufflesecurity/trufflehog
- How To Rotate- https://howtorotate.com/
- pre-commit- https://pre-commit.com/
- AWS Secrets Manager- https://aws.amazon.com/secrets-manager/
- Canary Tokens- https://canarytokens.org/
- GitGuardian ggcanary- https://github.com/GitGuardian/ggcanary
- SpaceSiren- https://github.com/spacesiren/spacesiren

Sample apps

- OWASP WrongSecrets- https://github.com/OWASP/wrongsecrets
- lotr- https://github.com/OtherDevOpsGene/lotr

The State of Secrets Sprawl Report

https://www.gitguardian.com/state-of-secrets-sprawl-report-2025



Questions?

Gene Gotimer
DevOps Engineer at Praeses, LLC
@OtherDevOpsGene