ThoughtWorks®

Cabinet Office

Tin Tulip - Blue team

Showcase #4 - May 12

Agenda

What we achieved

Threat modelling #3 recap

What’s next?

Summary

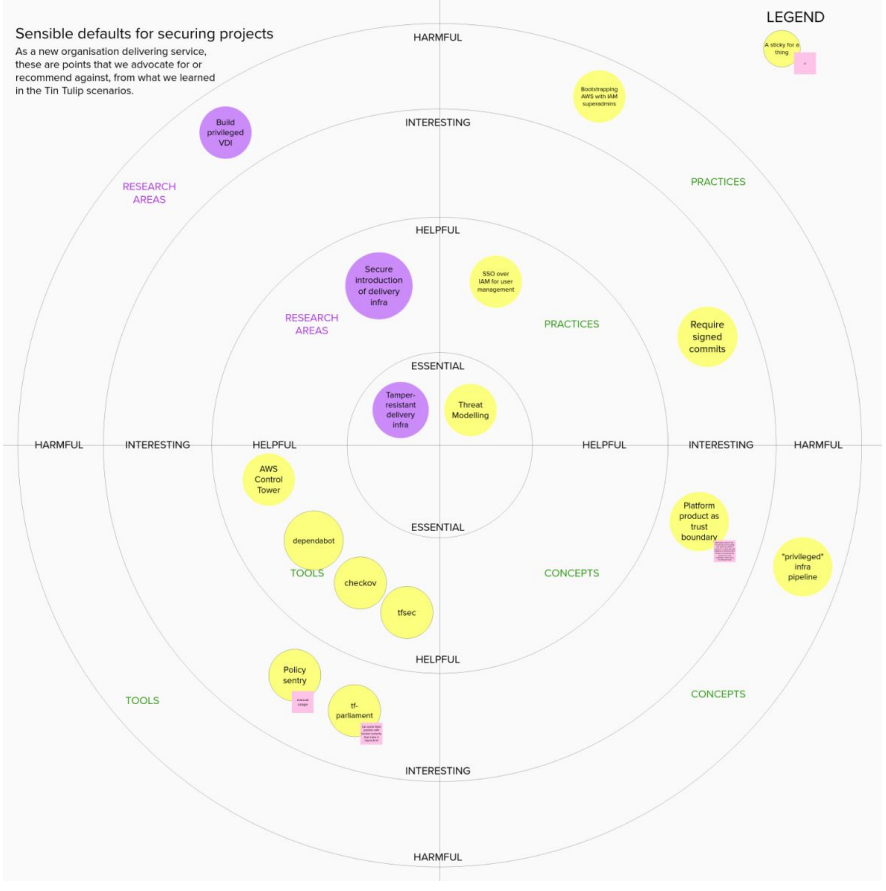
CLA's website is now available to the public.

The Platform team continues work to improve security controls across the organisation while the web team plans the build of a "licensing service".

What we achieved

What we worked on

- Pipeline for website infrastructure
- Pipeline for website content delivery
- Security tooling for IaC pipelines
- Publishing CLA's website



Zoomable diagram available on Mural at [this link](#) (signup required)

Website Infra Pipeline

What we built:

Pipeline that maintains infrastructure to host web content

Why we built it:

Changes in infrastructure are accounted for as code

What we learned from it:

- Restrict who can trigger the infra pipeline
- Pre-defined checks can break the pipeline (a good thing!)
- Manual approval for GH actions is in GitHub's Q2 2021 roadmap

*image of Colonial Pipeline taken from the BBC

Website Content Pipeline

What we built:

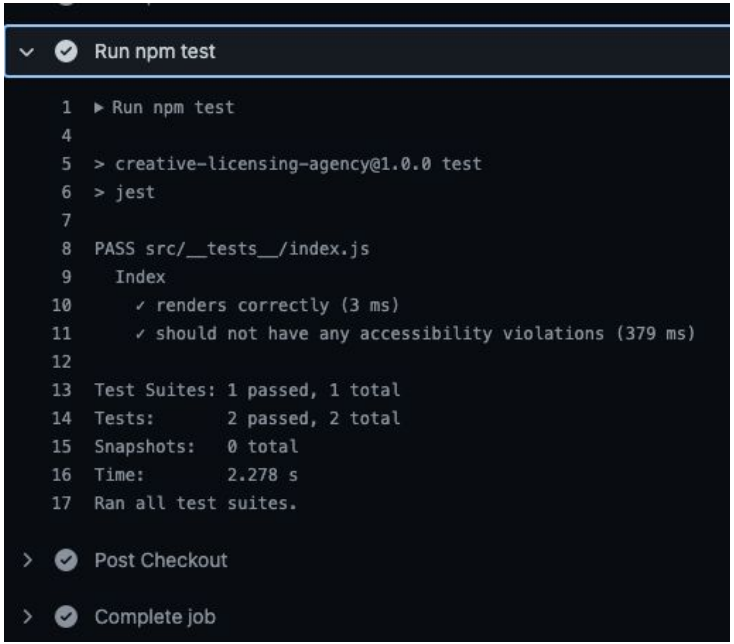
Pipeline that tests, builds and deploys a static website

Why we built it:

Develop frontend application using CI/CD securely

What we learned from it:

- Securely sharing IAM access keys and secret keys
- Finding the least privilege required
- Using dependabot to our advantage



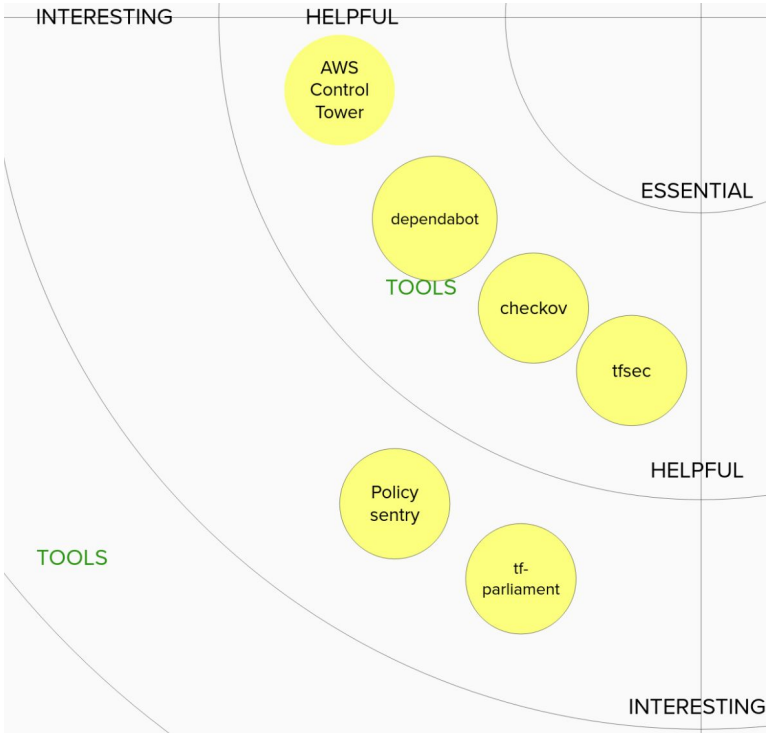
Security tooling for IaC pipelines

What we built:
Added security tooling in IaC pipelines

Why we built it:
Understanding which tools provide fast-feedback and which can be used for policy and gating

What we learned from it:

- tfsec and checkov are good for CI
- most tooling for IAM least privilege check either needs manual operation or has rough edges



Publishing CLA's website

What we built:

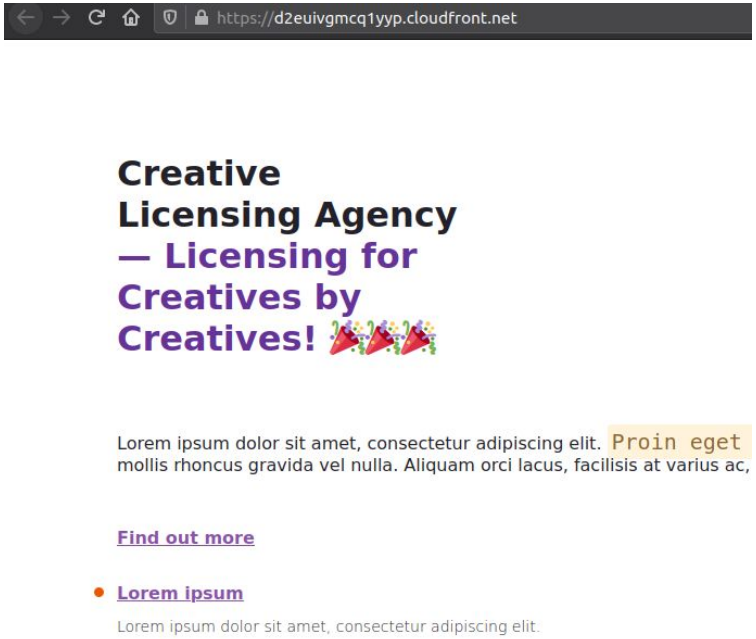
Published CLA's website to the internet!

Why we built it:

Completes Scenario 0's setup end to end

What we learned from it:

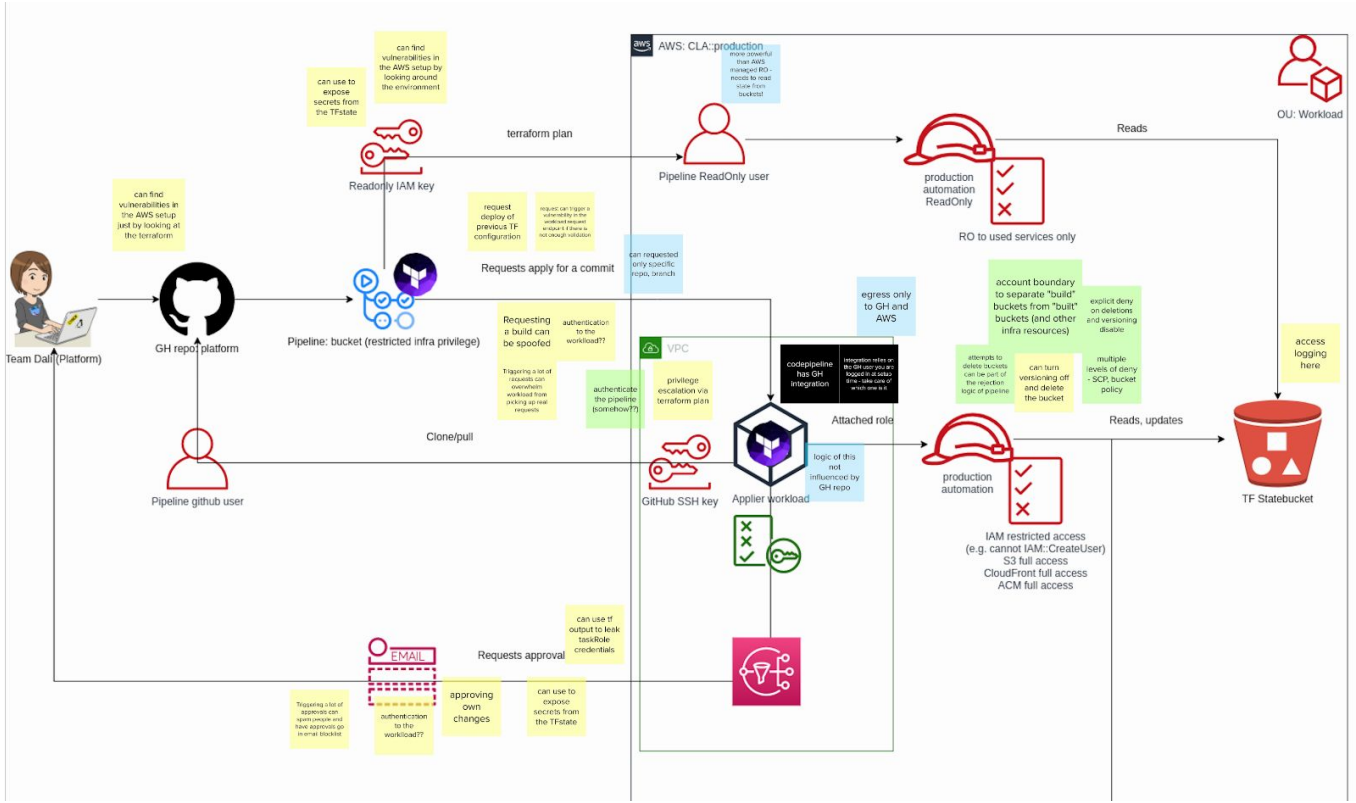
tfsec and Checkov report on slightly different AWS items - handy to have both



CLA's website is available at this link https://d2euivgmcq1yyp.cloudfront.net/

Threat modelling #3 - recap

Threat Modelling

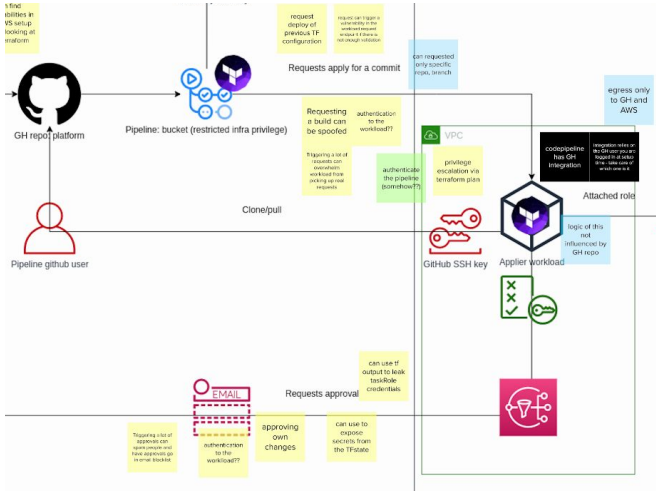


View full article [on GitHub]

Threat Modelling

Key takeaways:

- "Inverting" the interaction between GitHub and AWS mitigates many pipeline threats mentioned so far
- Requires a "TF applier workload" within AWS
  - allows to introduce trusted evaluation of policy and checks
  - needs to be locked down itself
- Account isolation from workload and prod would mitigate main risks identified



What's next?

Current next priorities

1 - Scenario 0 is "well architected":
GuardDuty, SCPs, centralised logs, GH actions controls

2 - Trustable pipelines:
From Threat Modelling - current pipelines are not tamper-resistant.
Build pipeline in AWS (high side) for assurance (Continuous Trust?)

3 - Scenario 1:

Build towards CLA's "Apply for a Creative License" service

Tradeoff Sliders review

- CLA website up - now catch up with Threat Modelling findings
  - and logging
- Introducing trust in pipelines means that we can't use SaaS for part of it

Sliders tracker (link requires access): https://app.mural.co/t/thoughtworksclientprojects1205/m/thoughtworksclientprojects1205/1620729955822

Scenario 1 proposal

- CLA's "Apply for a Creative License" service.
- Will capture contact details for users
  - users will have anonymous identity on service - no AuthN
- "Front office" only - data is captured but CLA is not ready to process it
  - Puzzle: GDPR 15 & 17? (SAR and right to erasure)
- Separate CLA Platform team (Dalí) from Governance (name TBD).
- Tech: Java Spring, TypeScript React, Postgres RDS

Scenario 1 options - target platforms

Dockerised service on ECS

- common architectural choice
- optionally run backend and frontend as API + static frontend instead

Lambda architecture:

- can take away some attack surface from Docker and a long-running service
- increasingly popular architecture choice
- Blue Team has low confidence in building this

Kubernetes service:

- main boundary between Platform and Application would move to the Kubernetes API instead of the AWS API, potentially worth exploring
- **high build cost**, the Kubernetes ecosystem adds a lot of complexity

Appendix: Guiding Principles

Guiding principle for the project

Does this teach us something new about a security control, or how to defeat it?

Guiding principle for platform implementation

In order to research the known security boundaries, the blue team will implement a test platform based on published best practices, including those published by the NCSC

Guiding principle for communicating learnings

The key audience for learnings are government departments, who want to empower their local technology teams to deliver secure systems