ThoughtWorks®

Cabinet Office

Tin Tulip - Blue team

Showcase #5 - May 19

Agenda

What we achieved

What’s next?

Summary

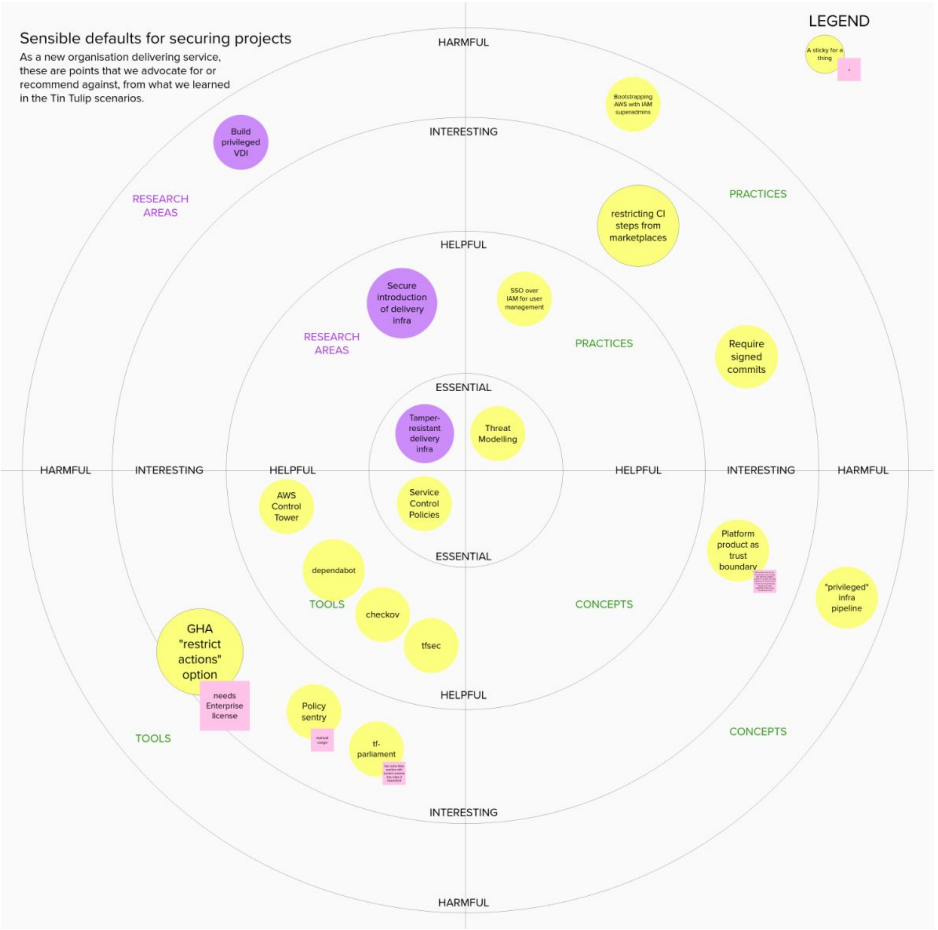
CLA's website is now available to the public.

The platform team improved security controls across the organisation and is now planning for improvements to pipeline security and the build of a "licensing service".

What we achieved

What we worked on

- GuardDuty
- Service Control Policies
- Extending Control Tower
- Restricting GitHub Actions



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

GuardDuty

What we built:

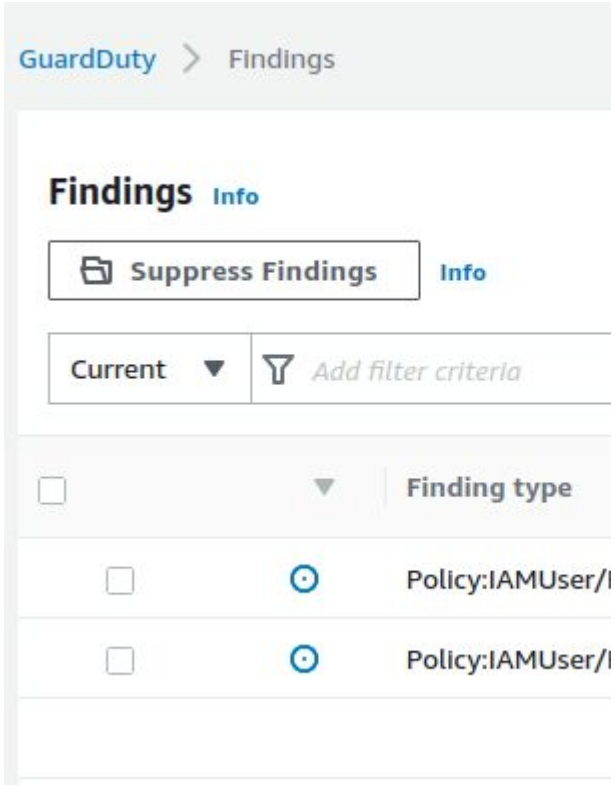
Enabled GuardDuty for all AWS accounts in the organisation from a centralised security account.

Why we built it:

Provides a governance framework that monitors threats and issues detailed findings of affected resource.

What we learned from it:

- Better to have enabled guard duty from the start.
- Easy to enable GuardDuty in other regions (best practice for all regions)



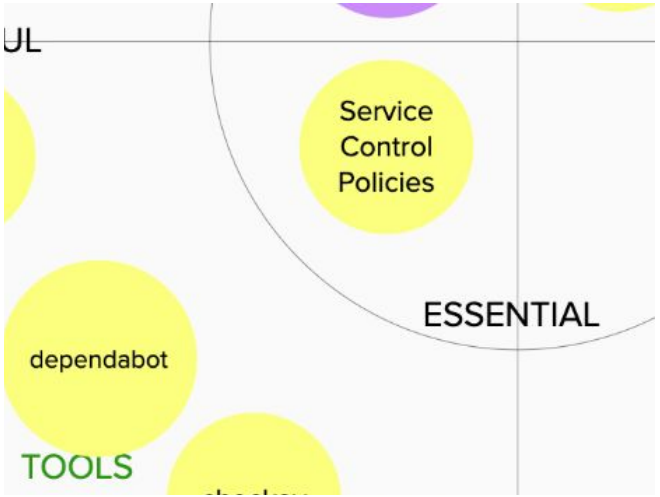
Service Control Policies

What we built:
A repository containing SCPs that were applied at the root organization

Why we built it:
Provide central control over all available permissions in all accounts within an organization

What we learned from it:

- Can deny root in child organizations from doing anything except logging in
- Can deny IAM users the ability to create passwords
- Can deny any actions in unused regions which combines well with AWS GuardDuty



Repository containing SCPs: https://github.com/tintulip/service-control-policies

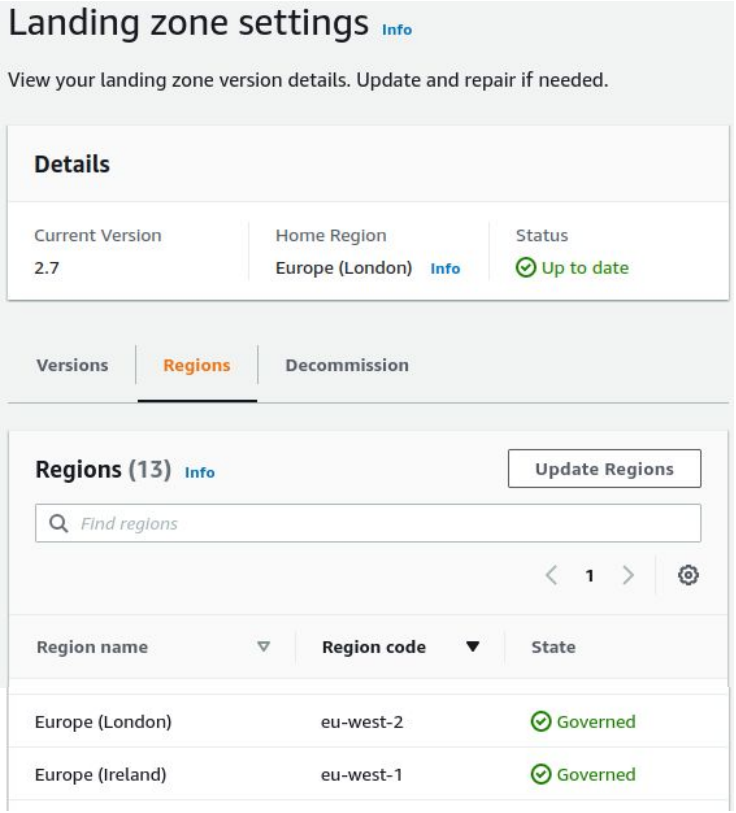
Extending Control Tower

What we built:
Enable Control Tower to govern and control eu-west-1 (Ireland). Eu-west-2 (London) is still home region.

Why we built it:
Easy way to set up and govern a secure multi-account AWS environment (called landing zone).

What we learned from it:

- “One-click” to set up a new landing zone
- OU may need to be re-registered to enroll the accounts within.



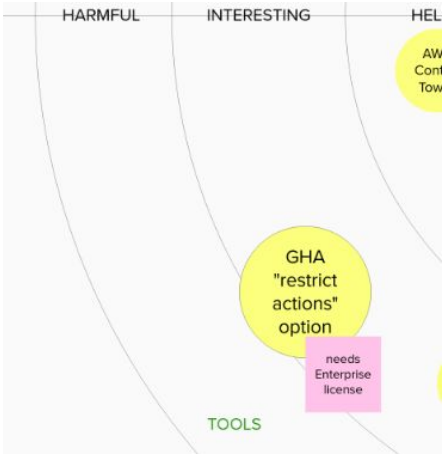
Restricting GitHub Actions

What we built:
Enabled GHA restrictions on GH org

Why we built it:
Preventing arbitrary actions from being used

What we learned from it:

- need GH Enterprise for controls to be effective
  - safelist option not available
  - cannot prevent arbitrary forking of actions
- Action verification process does not include any security assurance or in-depth due diligence



What's next?

Current next priorities

1 - Trustable pipelines:
From Threat Modelling - current pipelines are not tamper-resistant.
Build CD pipeline in AWS (high side) to enforce security controls.
Limited to groundwork to make it work.

2 - Scenario 1:

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

- Will capture user details to apply for a license
- Java service using SSR, Docker container in ECS, RDS database
- GDPR considerations out of scope - option for future scenario

Tradeoff Sliders review

- Timebox for detective controls concluded - back to focus on engineering

Sliders tracker (link requires access):

https://app.mural.co/t/thoughtworksclientprojects1205/m/thoughtworksclientprojects1205/1620729955822

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