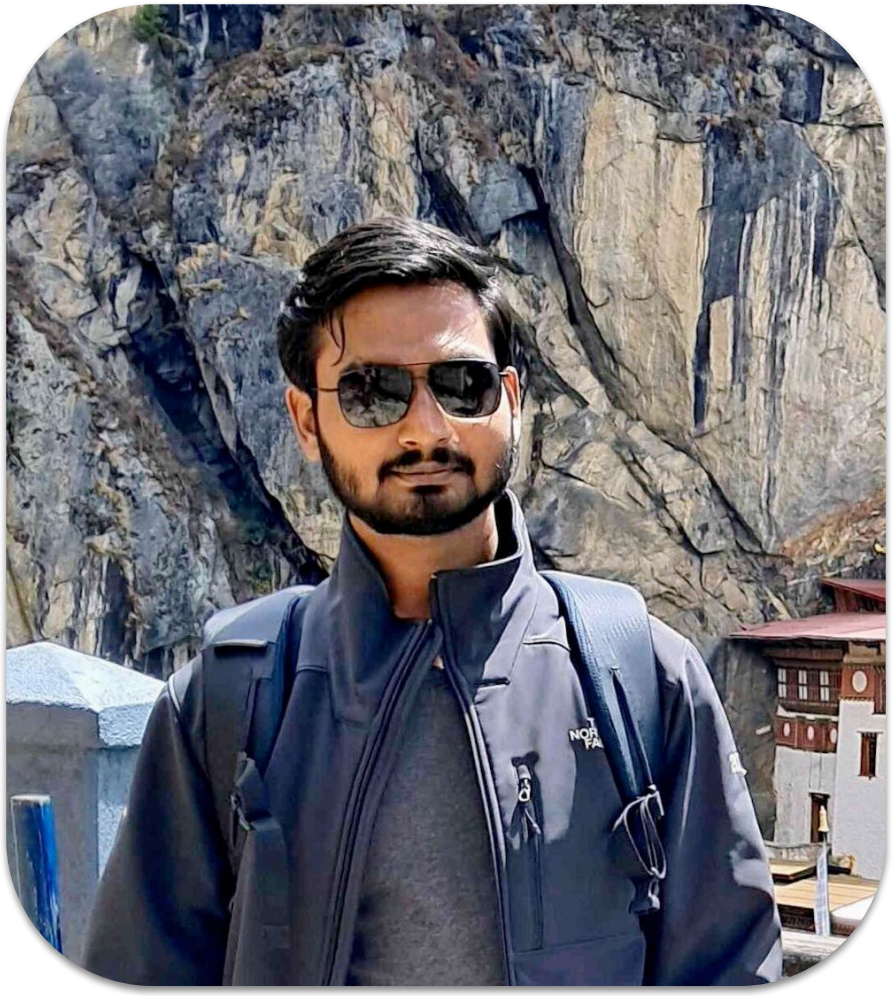
BSIDES GOA SECURITY CONFERENCE
27th April 2K24
Planet Hollywood Resort, Goa

Securing the Cloud: Detecting and Reporting Sensitive Data in ECR Images

About Me



Chandrapal Badshah

Security Researcher & Trainer

5+ Years of Experience

Cloud & Cloud Native Security

Blogs at badshah.io

- @bnchandrapal

Introduction to Amazon Elastic Container Registry

- Yet another "**elastic**" service from AWS
- Helps storing and distributing container images
- Integrates with other AWS services
- ECR supported private repositories till ECR Public was released (on 01 Dec 2020): https://gallery.ecr.aws/

Who uses ECR Public registries?

and many more...

Some ECR Terminologies

public.ecr.aws/datadog/agent:6.53.0-rc.1

Registry Alias

Repository Name

Image Tag

Some ECR Terminologies

public.ecr.aws/datadog/agent:6.53.0-rc.1

Registry Alias

Repository Name

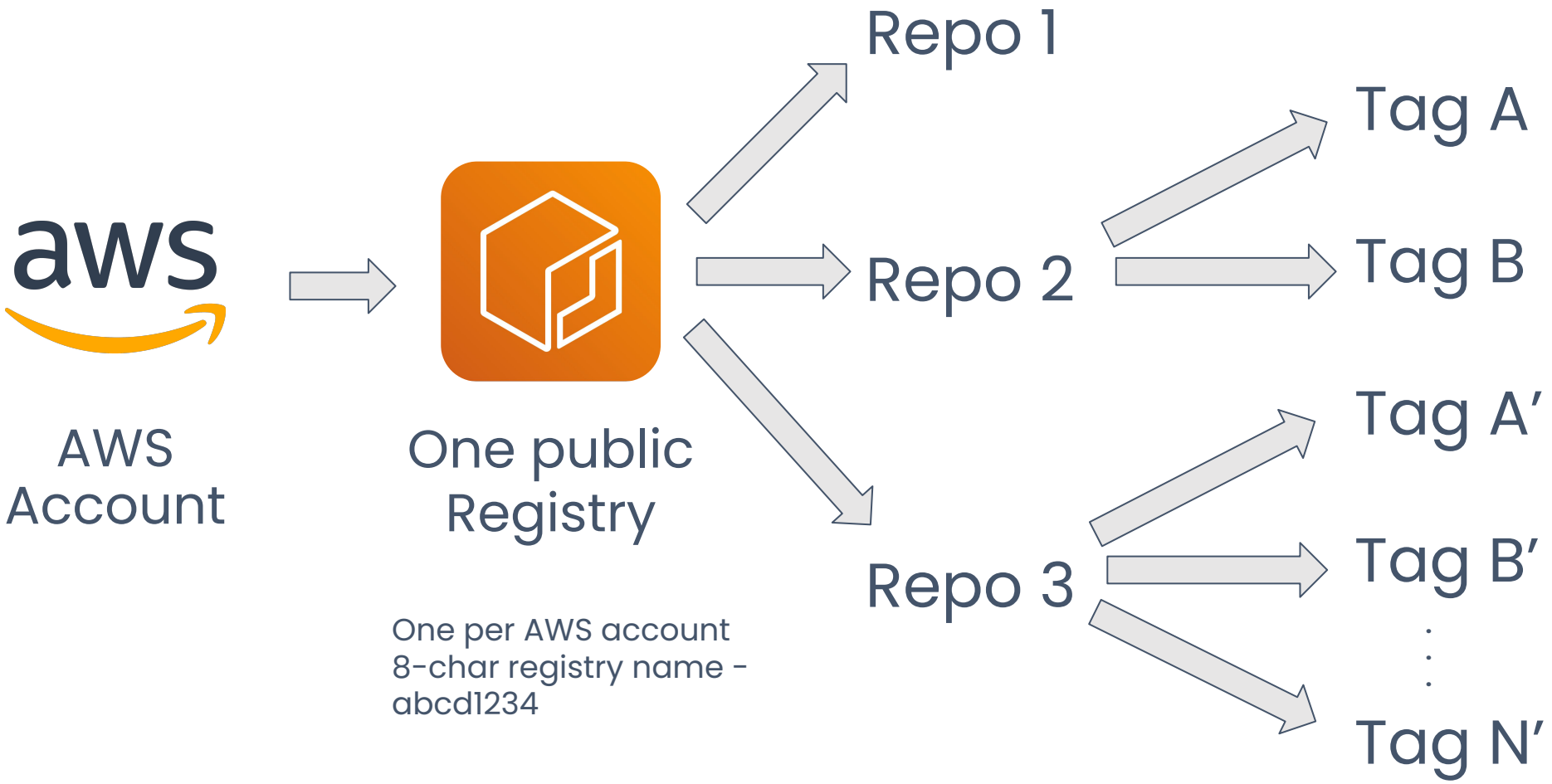
Image Tag

datadog/agent (6.5B+ downloads)

by datadog Verified Account

Contact AWS Support for Verification

Some ECR Terminologies



Problem with Container Images

- Vulnerable Packages in Base Image
- Insecure Dependencies
- Insecure Image Configurations
  - running as root
  - tonnes of unwanted software
- Hardcoded secrets
- Malware, Trojans, Backdoors

Problem with Container Images

- Vulnerable Packages in Base Image
- Insecure Dependencies
- Insecure Image Configurations
  - running as root
  - tonnes of unwanted software
- Hardcoded secrets
- Malware, Trojans, Backdoors

Hardcoded Secrets in Container Images

- A very common problem
- DockerHub images ~~contained~~ contains lots of secrets

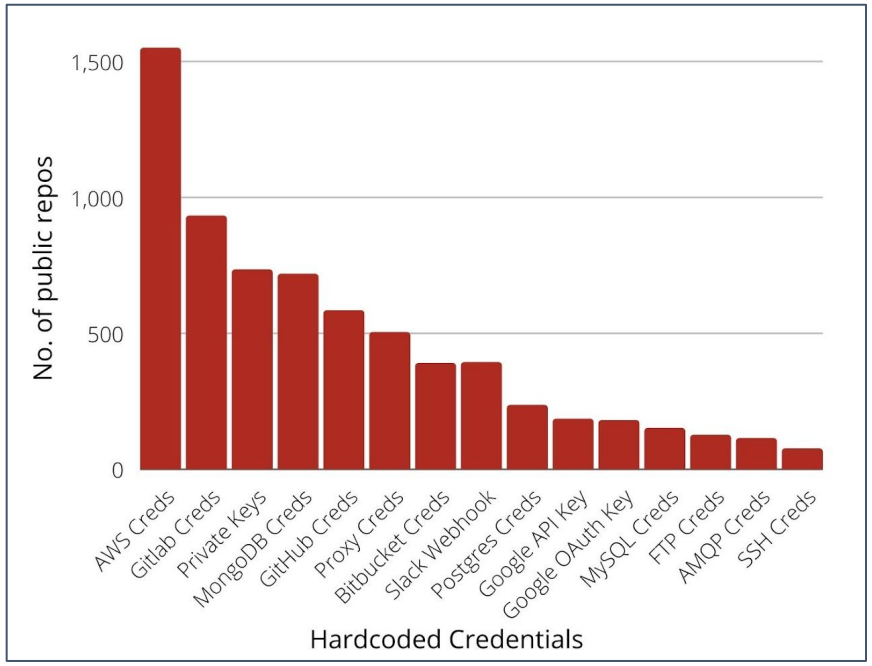
46076
Docker Containers

Leaked at least one Hardcoded Secret or Config file

Exposures in Docker Containers

15,541
Hardcoded Secrets were identified across 10,181 Repositories

57,589
Potentially Sensitive Config files copied to Docker Images across 36,176 Repos

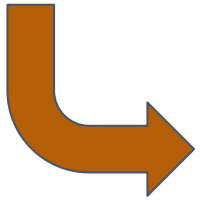


Source: https://redhuntlabs.com/blog/scanning-millions-of-publicly-exposed-docker-containers-thousands-of-secrets-leaked/ (Nov, 2021)

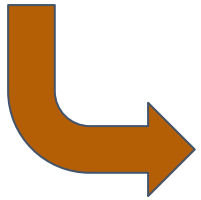
So, What about ECR?

Methodology to find secrets

Scrape ECR Registries



Fetch all image tags



Scan using OSS tools
(Trufflehog, Trivy, etc)

Methodology to find secrets

Scrape ECR Registries

ECR Public used AWS Cognito when searching for images. Scraping was complex and had issues.

Fetch all image tags

Scan using OSS tools
(Trufflehog, Trivy, etc)

Methodology to find secrets

Amazon ECR Public introduces new navigation and search features to the ECR Public Gallery

Posted On: Oct 3, 2023

Amazon Elastic Container Registry (ECR) Public has added new features that make it easier for customers to navigate the ECR Public Gallery and find the images they are looking for. New filters allow customers to search for images from well-known publishers s Docker and Amazon, and a new landing page highlights those filters as well as other frequently used repositories.

Source: https://aws.amazon.com/about-aws/whats-new/2023/10/amazon-ecr-public-navigation-search-features-gallery/

Stage 1: Scraping Registries

Scrape ECR Registries

ECR no longer used AWS Cognito.
Registries can be scraped.

Fetch all image tags

Scan using OSS tools
(Trufflehog, Trivy, etc)

Stats

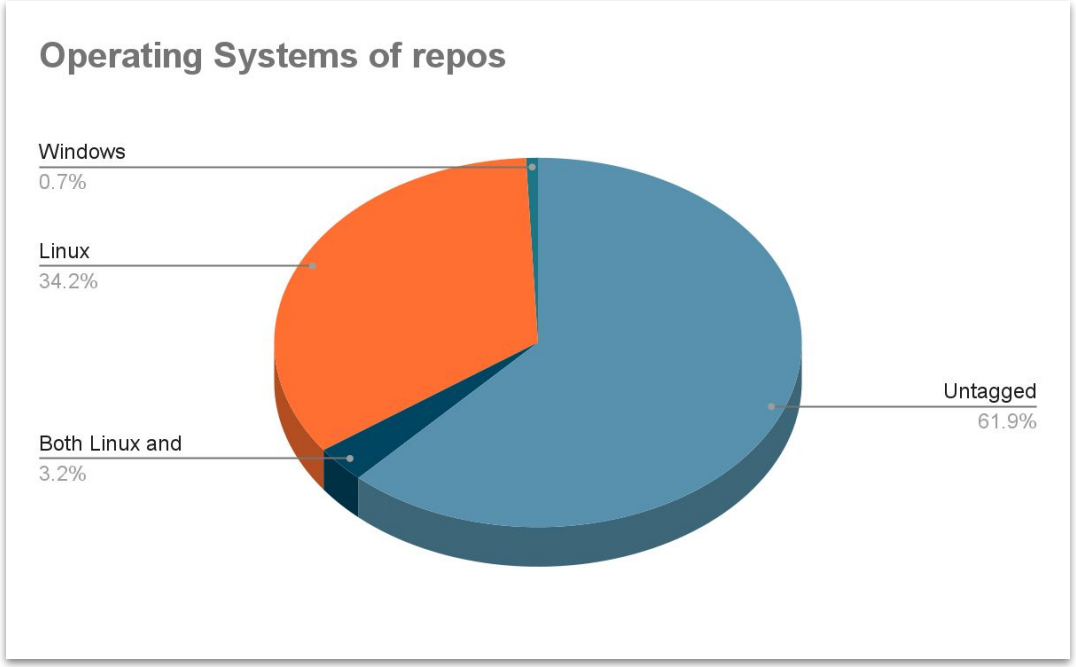
- Unique registry aliases: **>34,000**
- Unique repositories: **>106,000**
- Top 5 registry aliases having most repos:
  - biocontainers - 9149
  - y2o1b8w4 - 3905
  - h2x8n2t0 - 1578
  - d3e0i3l1 - 1411
  - k1i2y5t4 - 1251

Stats as on 10 Apr 2024

Moar Stats

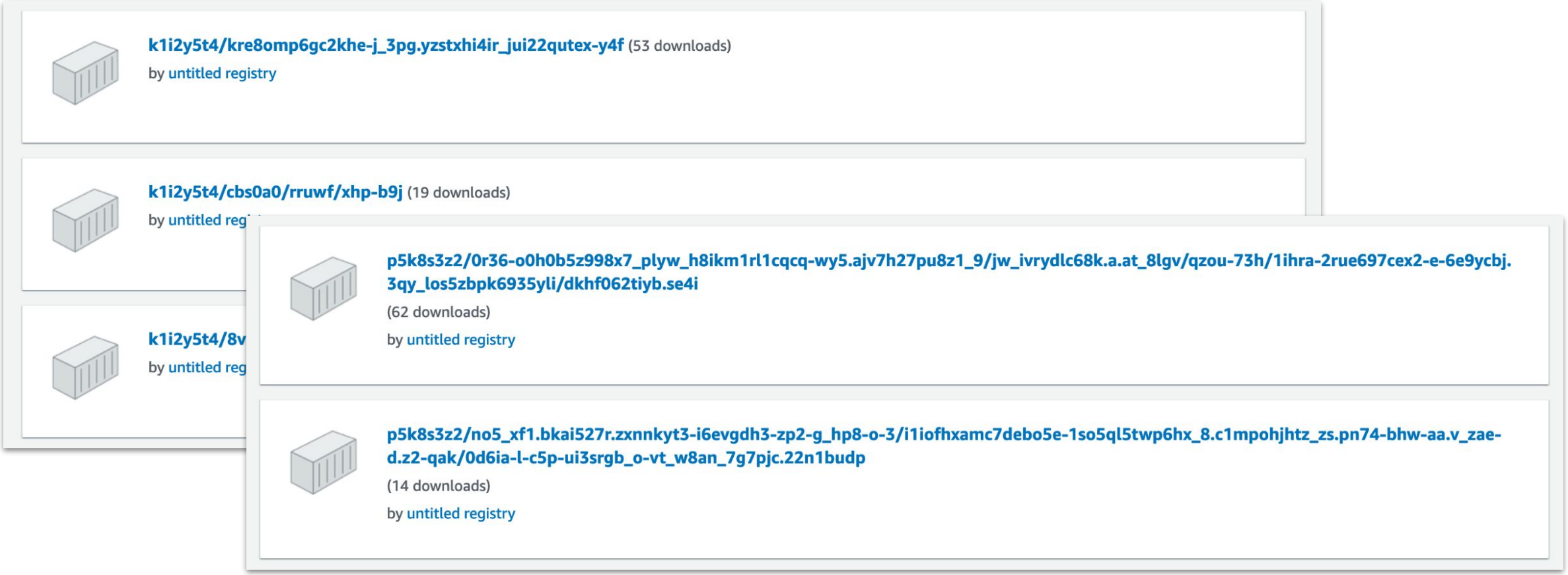
Repositories can be tagged with supported system architecture and operating systems

Remember these are just tags. Actual image might differ.



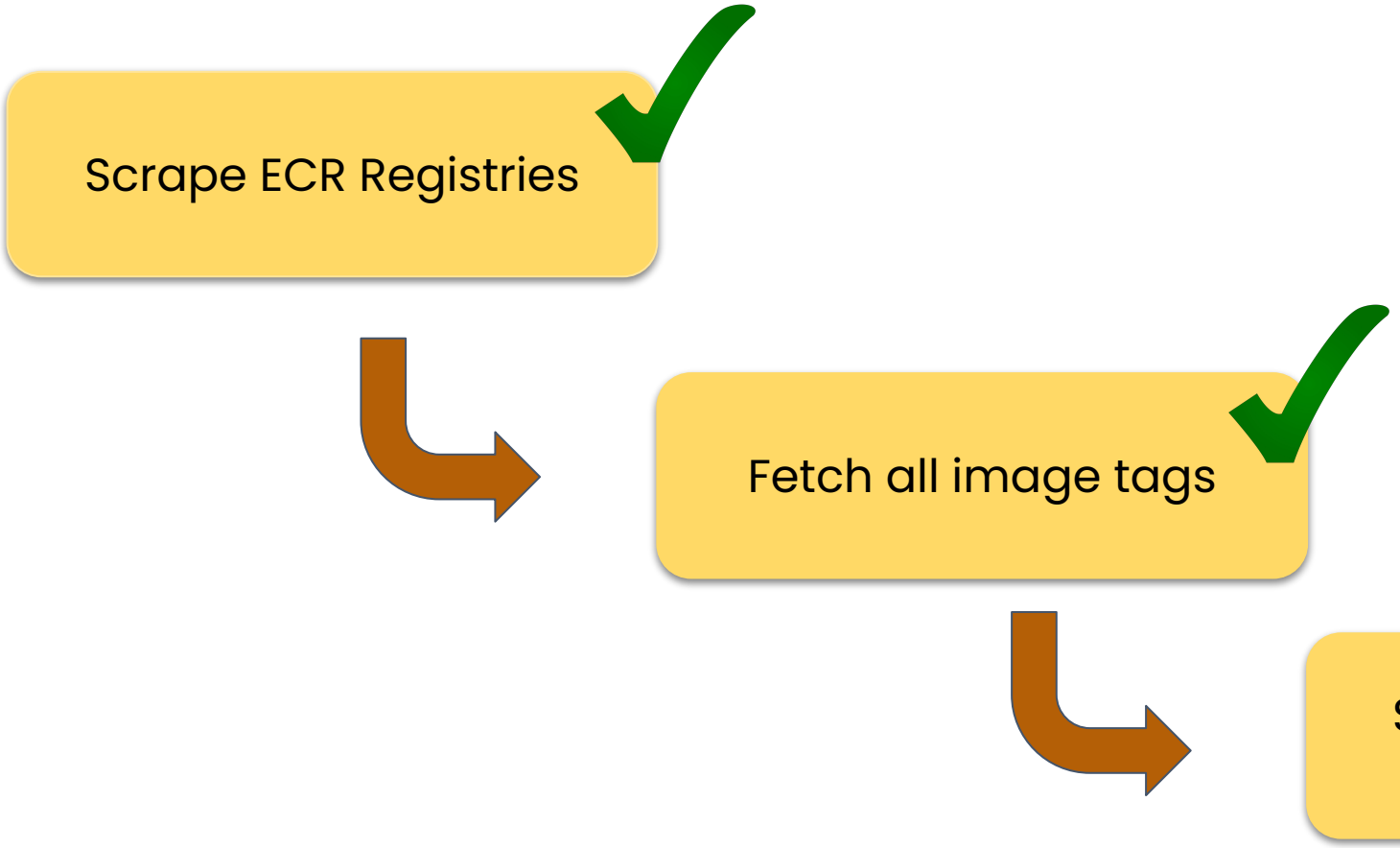
Stats as on 10 Apr 2024

Some weird things!



Some registry aliases may be abusing ECR Public.
Or probably using it for exfiltrating data. 🤦

Stage 2: Scraping Tags



Pretty straightforward

Stats

- Total unique docker images: **>1,515,000**
- Top 5 registry aliases having most tags:
  - bitnami - ~212,000
  - l0g8r8j6 - ~140,000
  - biocontainers - ~93,000
  - docker - ~75,000
  - gravitational - ~36,000

Stats as on 10 Apr 2024

Exclusions before we proceed

- Scanning all images is super costly
- Excluded the following:
  - Windows Container Images
  - Linux Container Images outside x64 and ARM architecture
  - Container Images of verified registries
  - Container Images of potential bot accounts

Stage 3: Scanning

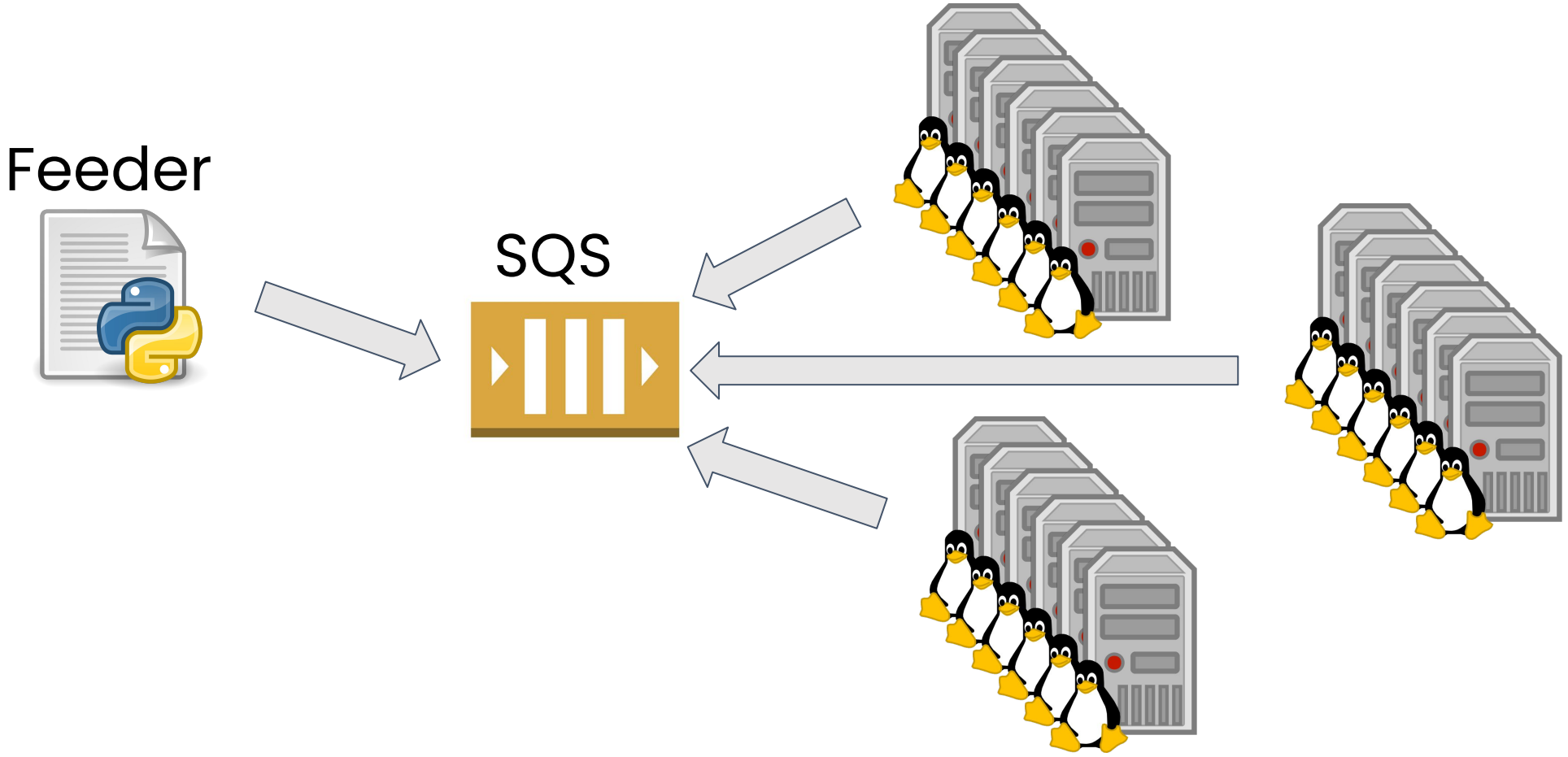
Scrape ECR Registries

Fetch all image tags

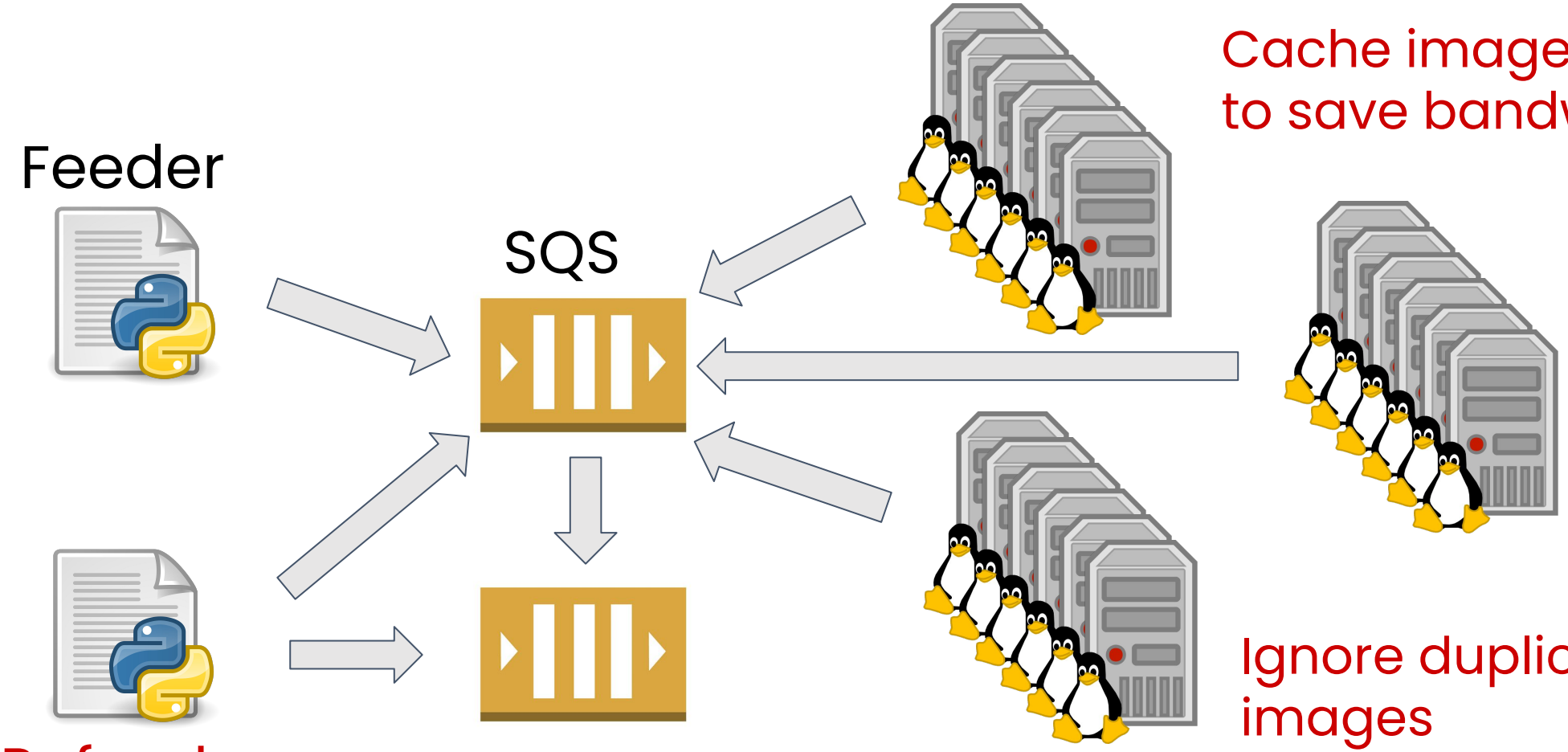
Pretty straightforward

Scan using OSS tools
(Trufflehog, Trivy, etc)

Stage 3: Scanning



Stage 3: Scanning (Optimized)



Stage 3: Scanning (Optimized)

How it started?

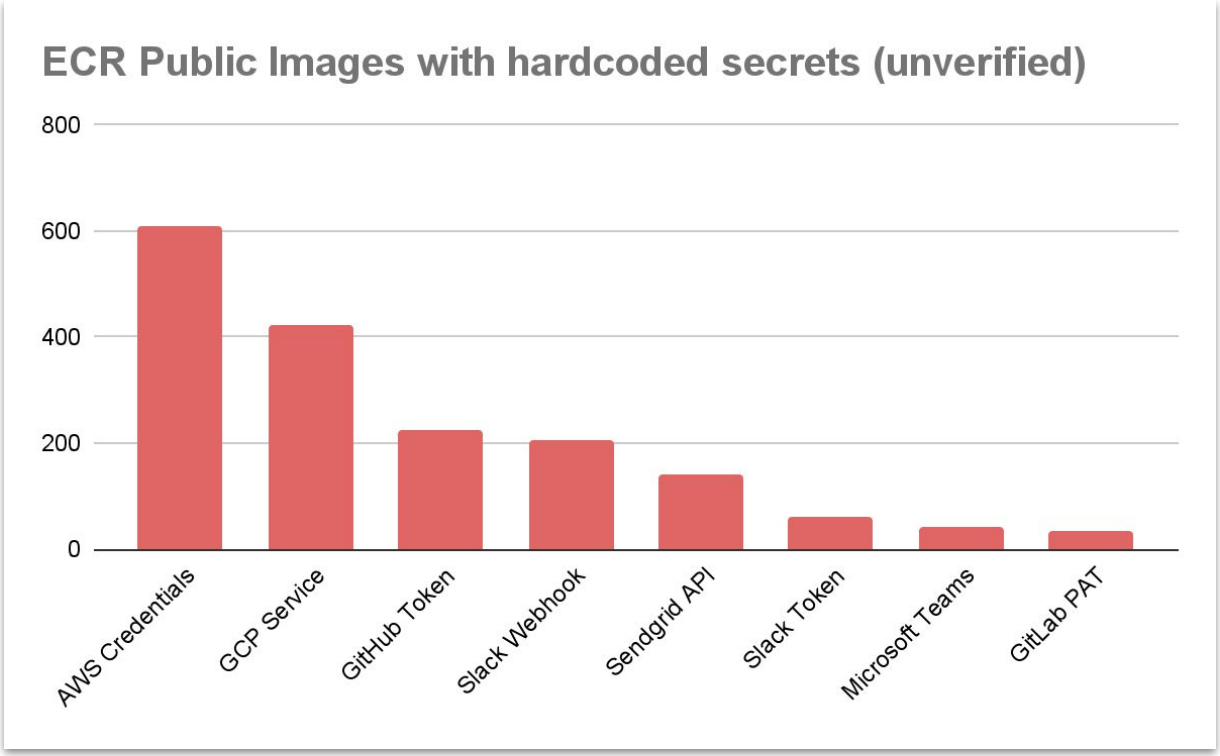
Stage 3: Scanning (Optimized)

How it started?

How it ended?

Stats

- Total images I scanned: **84,692**
- Images containing at least 1 secret: **5,900**



Hardcoded AWS Credentials

Valid AWS keys leaked: **111**

Out of which **14** belong to *root* users

Interesting IAM usernames:

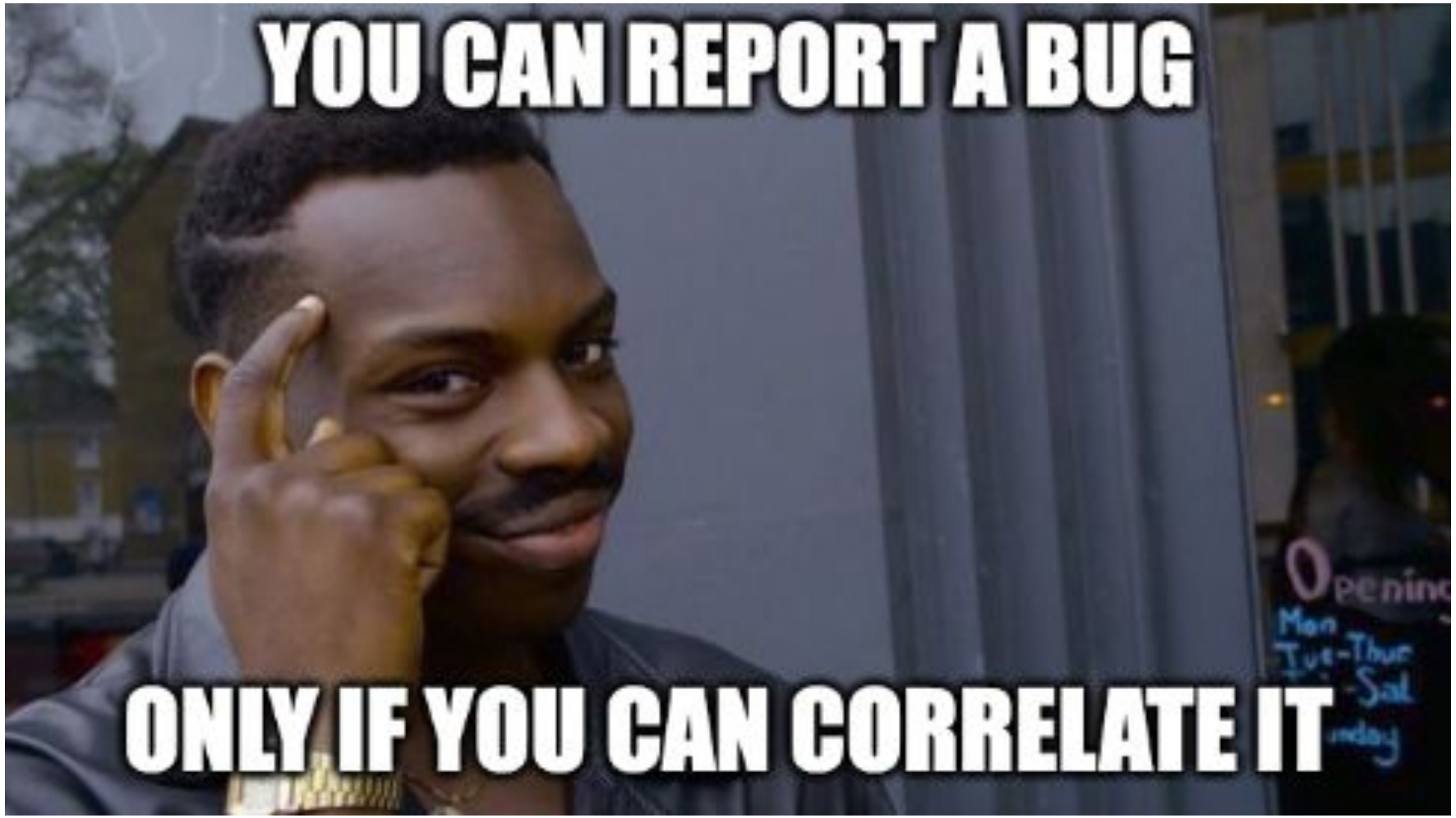
- upload-testing
- s3-role
- cicd-developer
- Administrator
- backup-user
- terraform-admin

If I were an attacker!

- Cloud Credentials -> Cryptomining
- DB Credentials -> Dump User Data
- Enterprise Communication Tools -> Phishing
- VCS credentials (GitHub, GitLab, etc) -> Dump proprietary source code

But No!

First Hurdle: Correlating



First Hurdle: Correlating

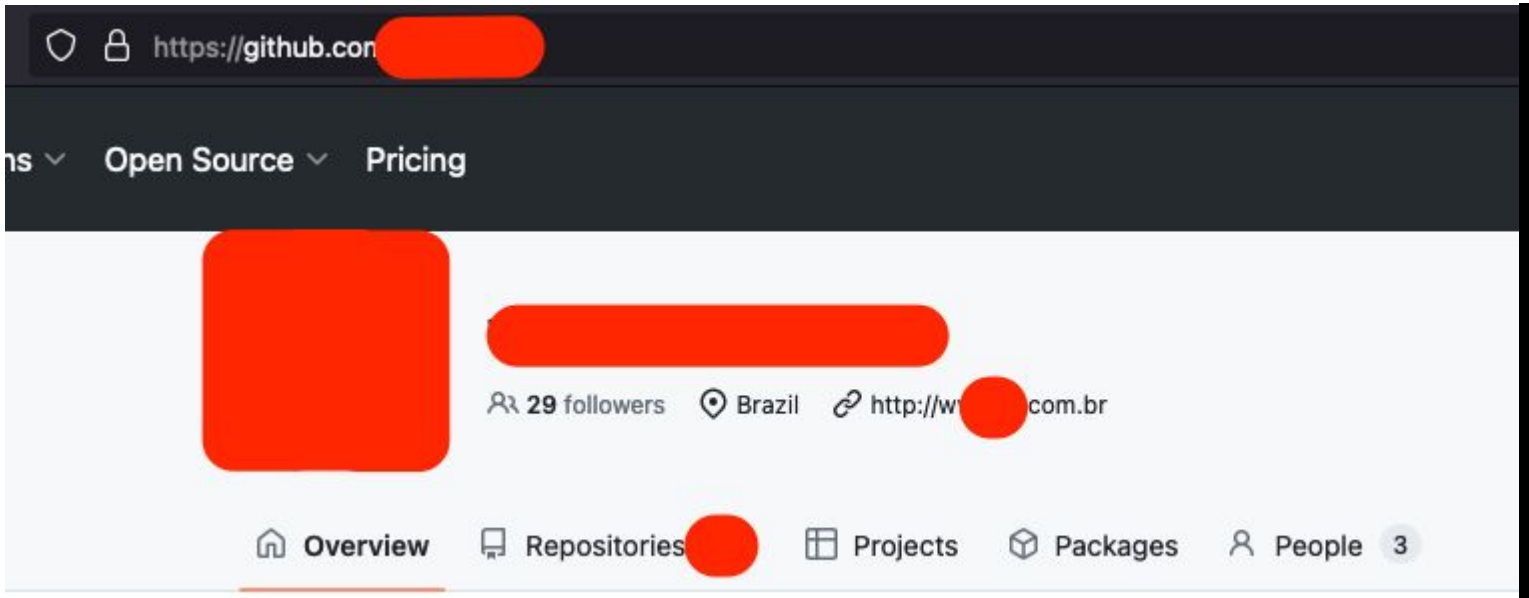
- There's *no* direct correlation between registry alias and AWS account
- Common ways to find domain name from container image
  - Maintainer label
  - Hostname used in config files
  - Git commit log

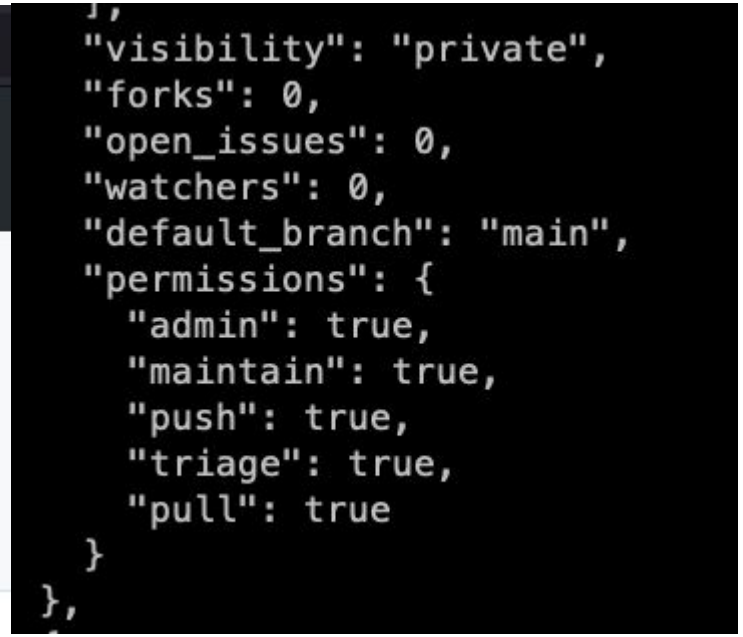
```
commit 9bd7af3e9880a2f454a46d7673c073a7eb016184 (HEAD,
Author: tiago[redacted]tc <tiago[redacted]@s[redacted]r>
Date:   Wed Jul 26 11:26:54 2023 -0300

    chore: update[redacted]-modules to 2.0.17

commit e024fe08e09ecb3f9f1f6b08e3d8709ea0efc554
Author: Cristian <cristian[redacted]@s[redacted]r>
Date:   Tue Jul 25 12:11:25 2023 -0300

    chore: change package version
```

```
ENV LOG_LEVEL=Info
RUN /bin/sh -c python3 -m pip install -r requireme
COPY requirements.txt / # buildkit
RUN /bin/sh -c sudo apt-get update &&     sudo apt
MAINTAINER Sean[redacted] <co[redacted]t@[redacted].com>
/bin/sh -c echo '#!/bin/sh.real\nbalena-info\nrm -
/bin/sh -c [ ! -d /.balena/messages ] && mkdir -p
/bin/sh -c apt-get update && apt-get install -y --
/bin/sh -c #(nop)  LABEL io.balena.device-type=ras
```

Correlating from Secrets

- Certain secrets allow correlating without causing damage
- GitHub token -> GitHub User/Org -> Email ID/Domain



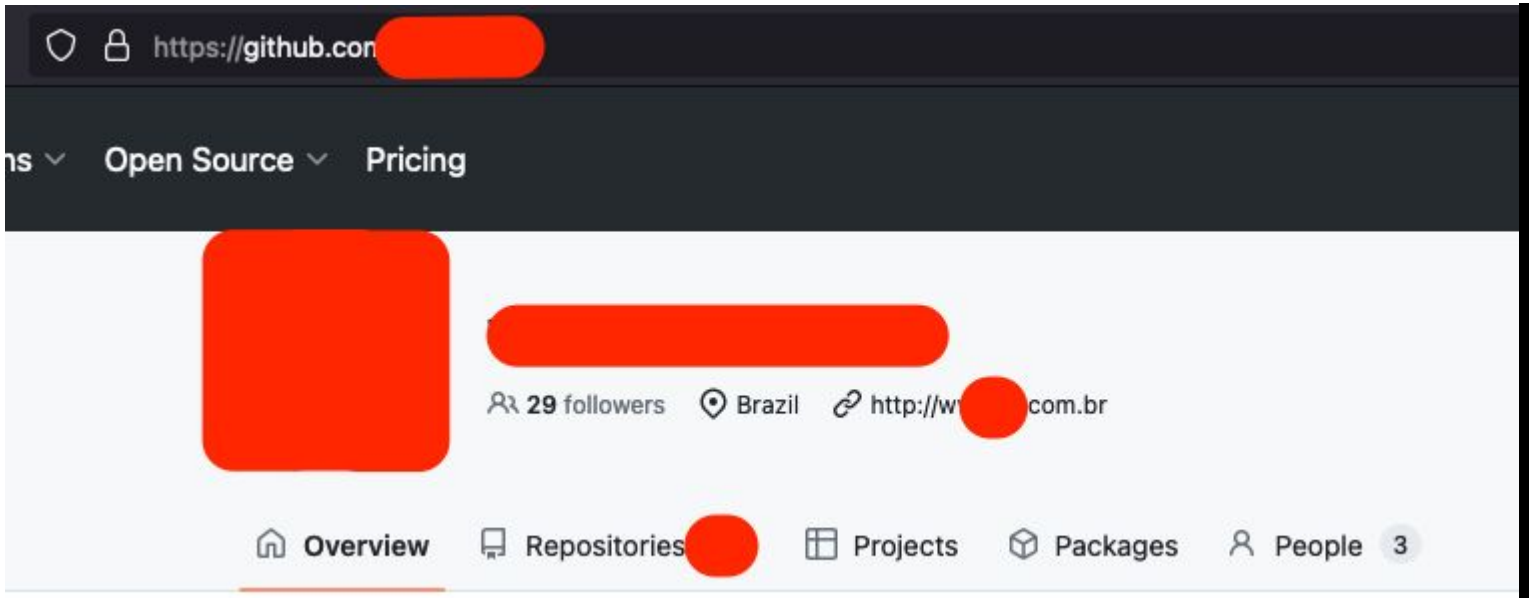


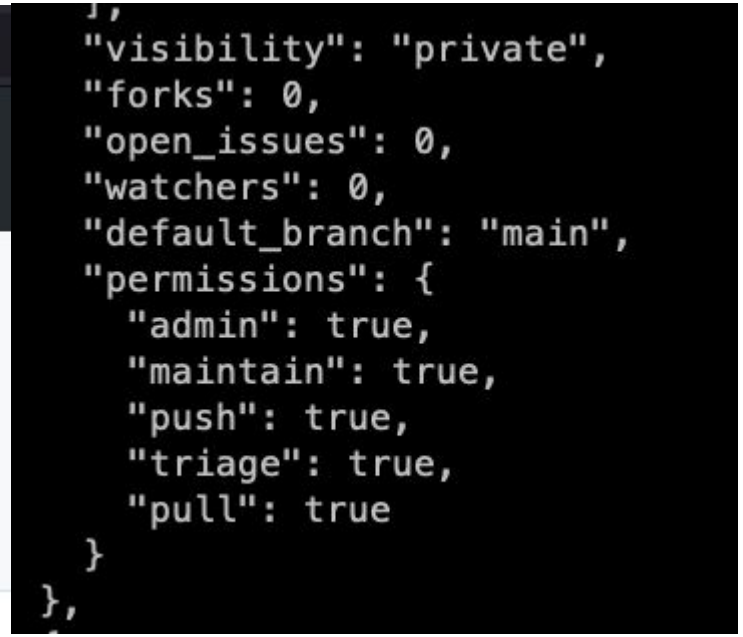
Correlating from Secrets

- Certain secrets allow correlating without causing damage
- GitHub token -> GitHub User/Org -> Email ID/Domain

Other secrets:

- GitLab Token
- Slack Token
- SSL Certificate





Fun Fact #1

A good number of images are very minimal.

All that exists are *valid secrets* and probably a binary file or generic software installation directory (ELK, Java, etc)

```
CREATED BY
/bin/sh -c #(nop)  CMD ["/bin/sh" "-c" "sh /app/bash_commands.sh"]
/bin/sh -c #(nop)  ENV AWS_SECRET_ACCESS_KEY=TdKs[redacted]u5hQ0+Sy8RN
/bin/sh -c #(nop)  ENV AWS_ACCESS_KEY_ID=AKIA[redacted]EZJT
/bin/sh -c ls -l
/bin/sh -c #(nop) ADD dir:812d6b325d[redacted]6edd06d54238b1b3c2ba4c1735898ec9fe9e7e1873 in .
/bin/sh -c #(nop)  VOLUME [/app]
/bin/sh -c #(nop) WORKDIR /app
/bin/sh -c dnf install java-1.8.0-amazon-corretto-devel -y
/bin/sh -c curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip" &&    unzip av
/bin/sh -c yum update -y && yum install jq unzip bc procps -y
/bin/sh -c #(nop)  CMD ["/bin/bash"]
/bin/sh -c #(nop) ADD file:f979bddb29b[redacted]eced160f2ab19e881619a5b67e1fbf6cdb6c3 in /
```

Hardships of Correlating

- Usage of personal emails (Gmail, Mail.ru, etc)
- Maintainer labels of base images can be misleading
- Secrets could be correlated to two or more orgs (ex: Consultancies and Freelancers)
- SaaS & PaaS providers make it trickier to correlate

```
ENV PKG_RELEASE=1
ENV NGINX_VERSION=1.24.0
LABEL maintainer=NGINX Docker Maintainers <docker-maint@nginx.com>
/bin/sh -c #(nop)  CMD ["/bin/sh"]
/bin/sh -c #(nop) ADD file:c3b6b575eb741f914ec12bd4df43de0cb044a1f2bae7ff15d
```

Second Hurdle: Reaching out safely to affected users

- Use an email ID which can be correlated to you
- Send an email saying I found a vuln in their infra (without giving much info)
- Pray your email is not marked spam and wait for a reply in 15-30 days
- Send a second final email saying it’s the final reminder
- Ignore and focus on your (next?) research

Email Format

[2] Security misconfiguration in [redacted]tg.com infrastructure

From research@badshah.io

To security@[redacted]tg.com, info@[redacted]tg.com, contact@[redacted]tg.com

Hi,

I'm Chandrapal Badshah, a security researcher. During one of my security researches, I found a potential high severity issue that impacts a[redacted]ctg.com's infrastructure. Please let me know if you have any bug bounty/vulnerability disclosure program so I can submit the issue report.

Thanks,
Badshah

First email

Last Email Format

Hi,

This email is the final reminder. Please let me know if you have any bug bounty/vulnerability disclosure program so I can submit the security issue.

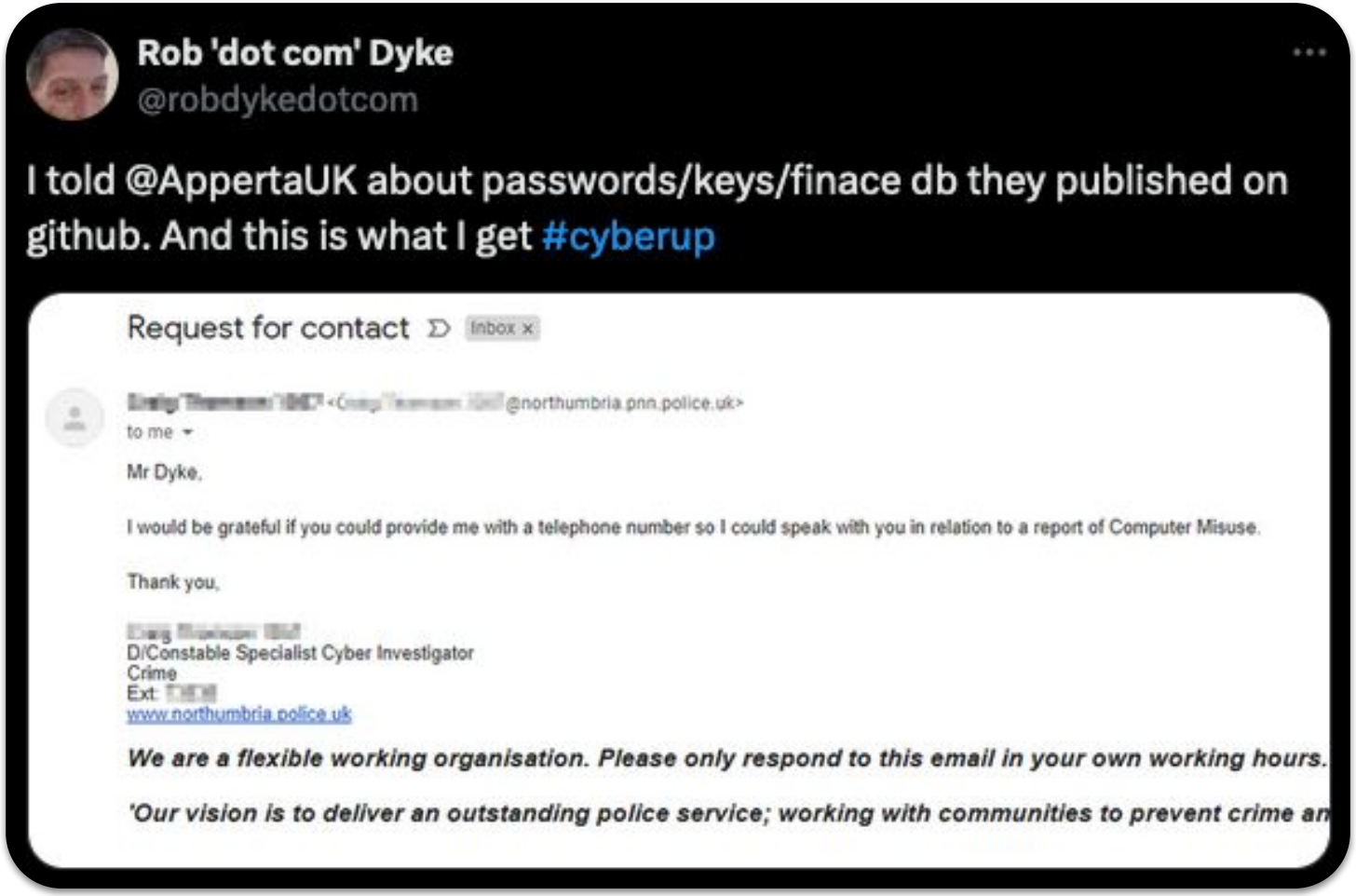
Regards,
Badshah

Final email

I need to ensure I don’t reach out to these domains again in future

BSIDES GOA
Security by the Beach 2K24

Good question



Check out https://attrition.org/errata/legal_threats/

Fun Fact #2

Security product companies are just software companies

Good number of affected security companies don't have:

- security@ email address
- security.txt file
- vulnerability disclosure/bug bounty program

Fun Fact #2

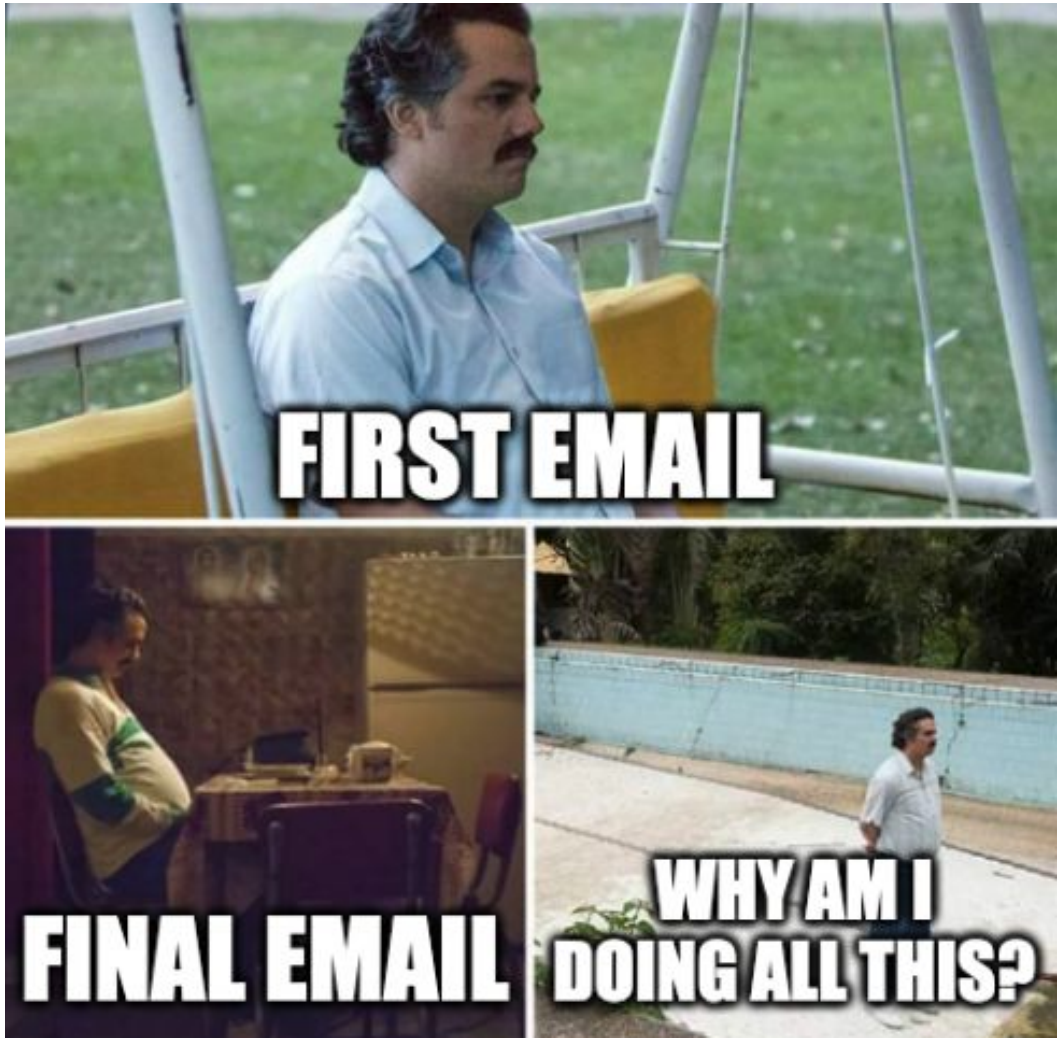
Security product companies are just software companies

Good number of affected security companies don't have:

- security@ email address
- security.txt file
- vulnerability disclosure/bug bounty program

They are still ISO 27001, etc compliant 😜

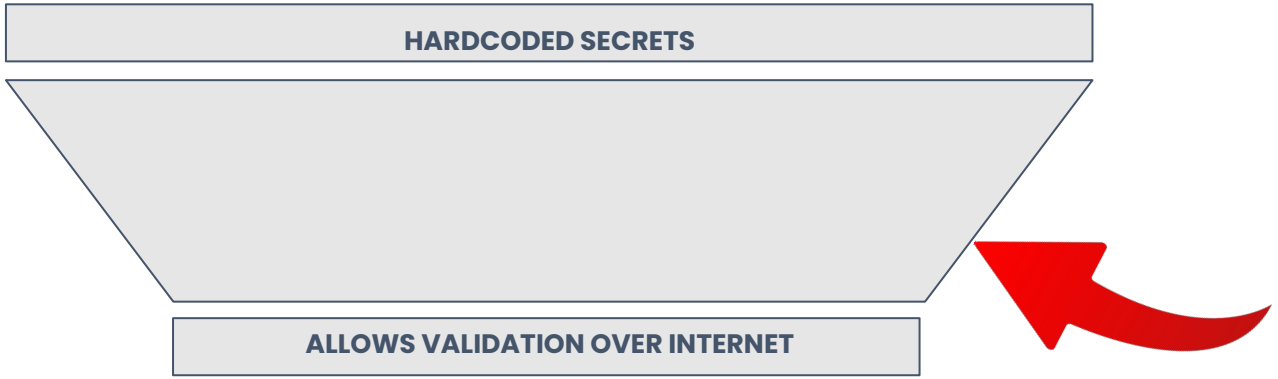
Third Hurdle: Get it fixed



My research in a nutshell

HARDCODED SECRETS

My research in a nutshell



Expired Credentials

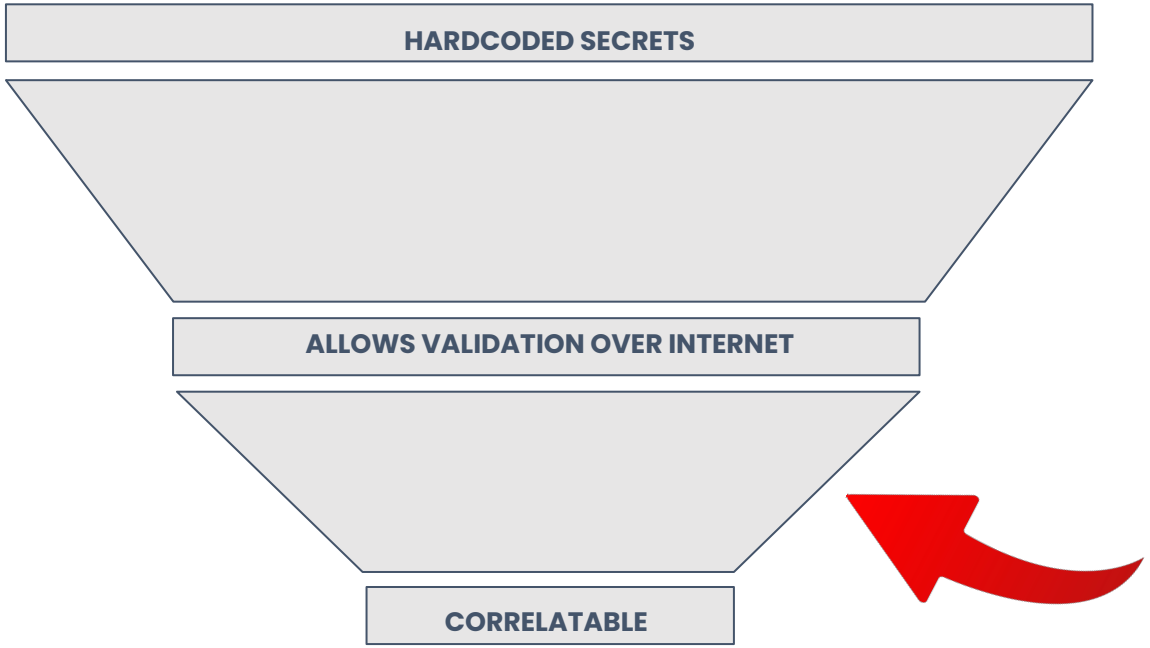
Internal passwords

SSH/Private Keys

JWT Signing Keys

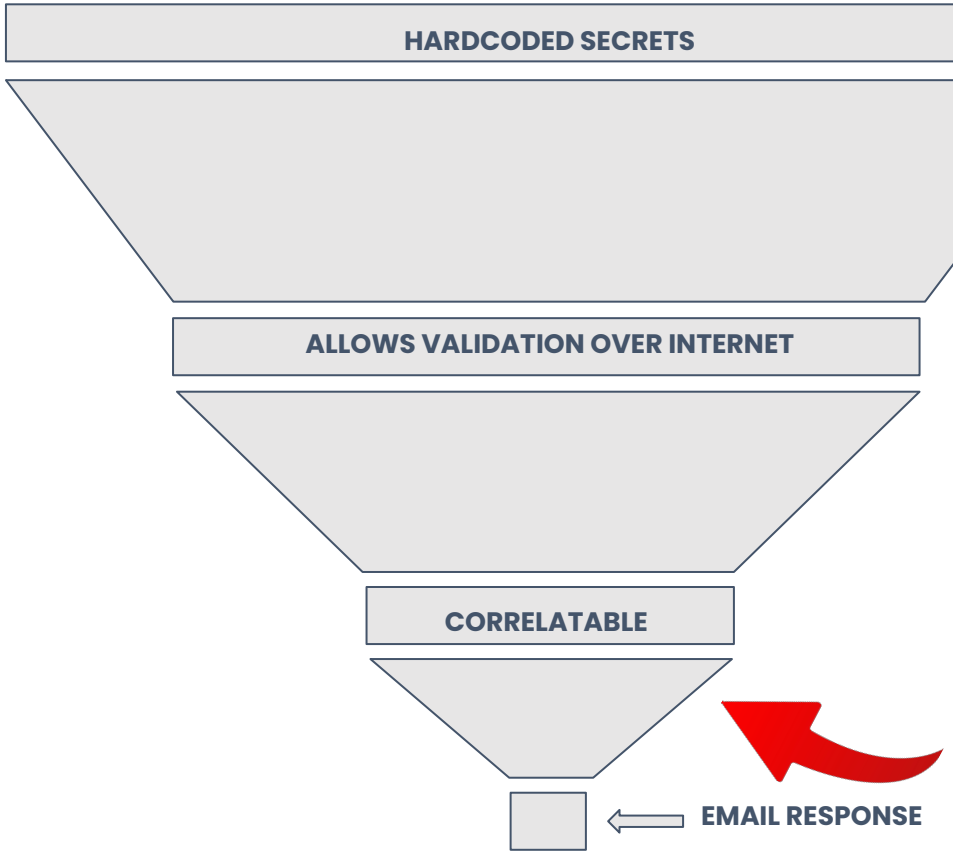
Secrets with IP Whitelisting

My research in a nutshell



Validated secrets that can't be correlated

My research in a nutshell



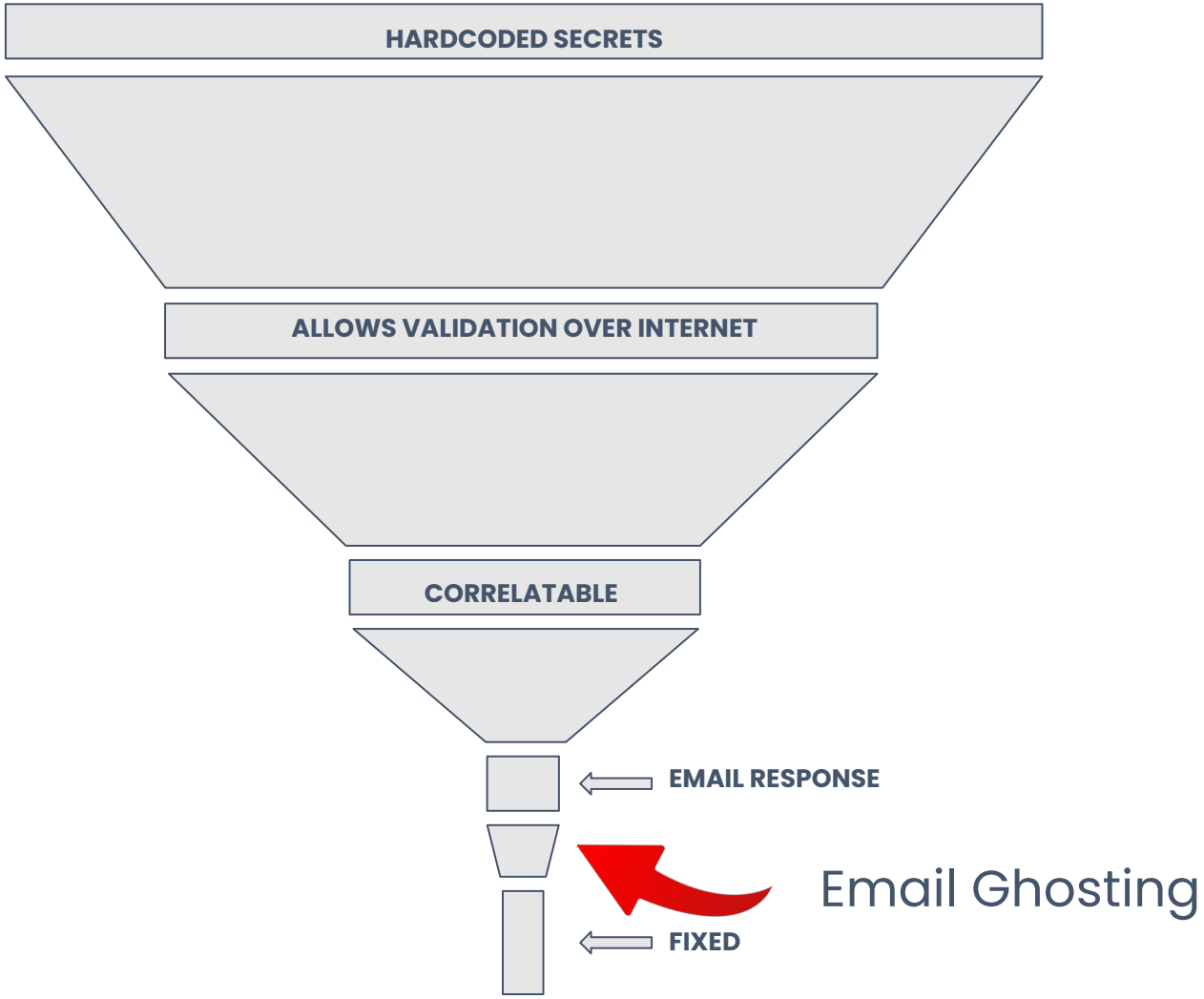
Bounced Emails

No Email Response

Auto responders

Email ID Restrictions

My research in a nutshell



What’s the greatest hardship?

What’s the greatest hardship?

NO INCENTIVES TO DO THE RIGHT THING

Case Studies

Case Study 1

```
aws configure set region $REGION
git clone https://sh[redacted]s:glpat-F1z[redacted]yes@gitlab.aws.dev/deep-visual-search/infra.git
cd infra
#pip3 install --upgrade pip
```

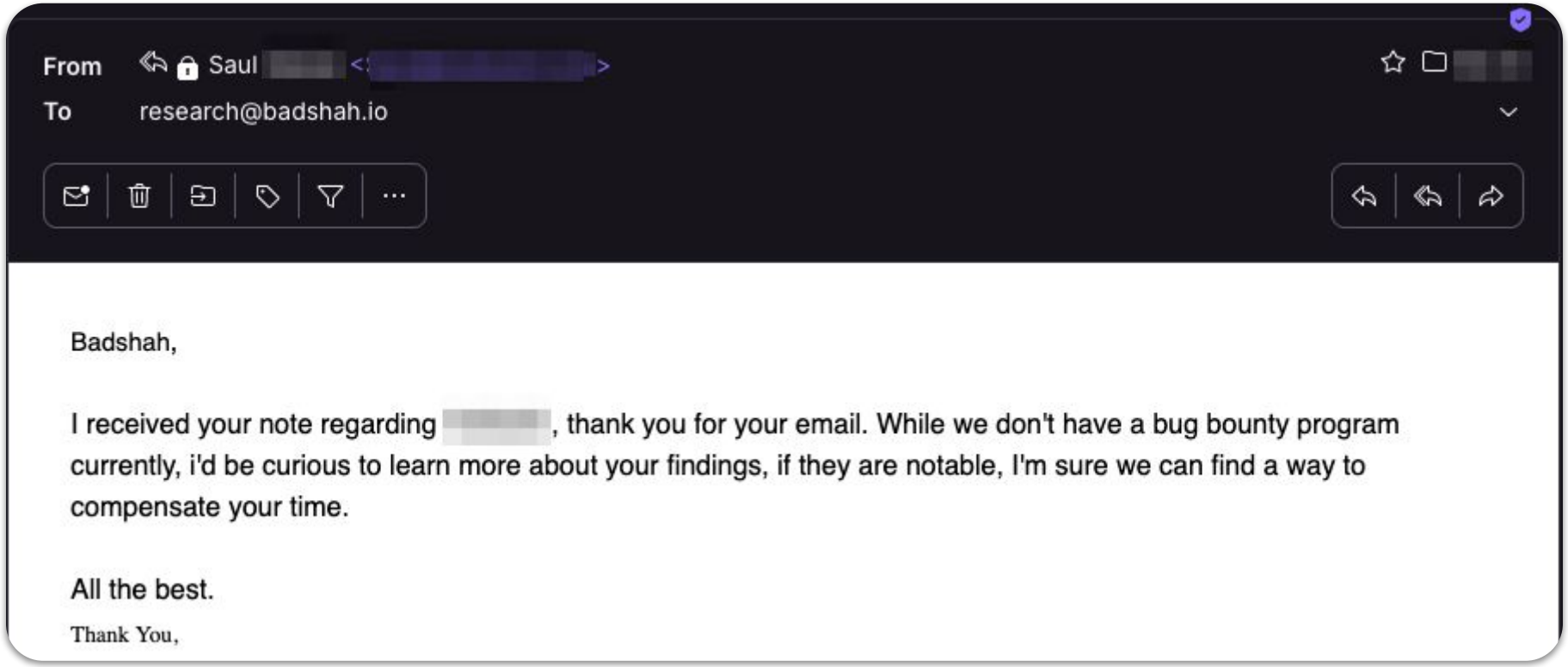
```
~ git clone https://sh[redacted]s:glpat-F1[redacted]es@gitlab.aws.dev/deep-visual-search/infra.git
Cloning into 'infra'...
remote: This GitLab instance does not allow git operations via HTTPS for security reasons. Please use Midway-signed SSH keys.
fatal: unable to access 'https://gitlab.aws.dev/deep-visual-search/infra.git/': The requested URL returned error: 403
```

Case: Developer hardcoded GitLab token

Impact: Didn't contain source code. Unable to clone source code.

Status: Fixed by AWS.

Case Study 2



From Saul [redacted] <[redacted]>

To research@badshah.io

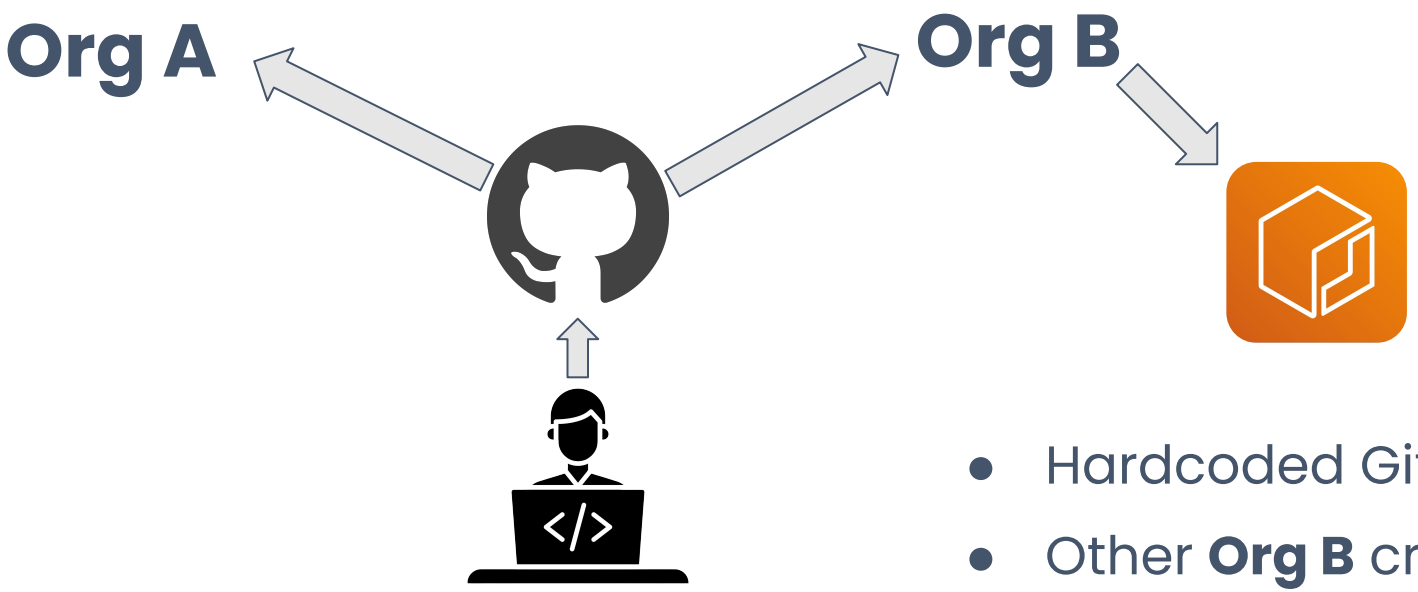
Badshah,

I received your note regarding [redacted], thank you for your email. While we don't have a bug bounty program currently, i'd be curious to learn more about your findings, if they are notable, I'm sure we can find a way to compensate your time.

All the best.

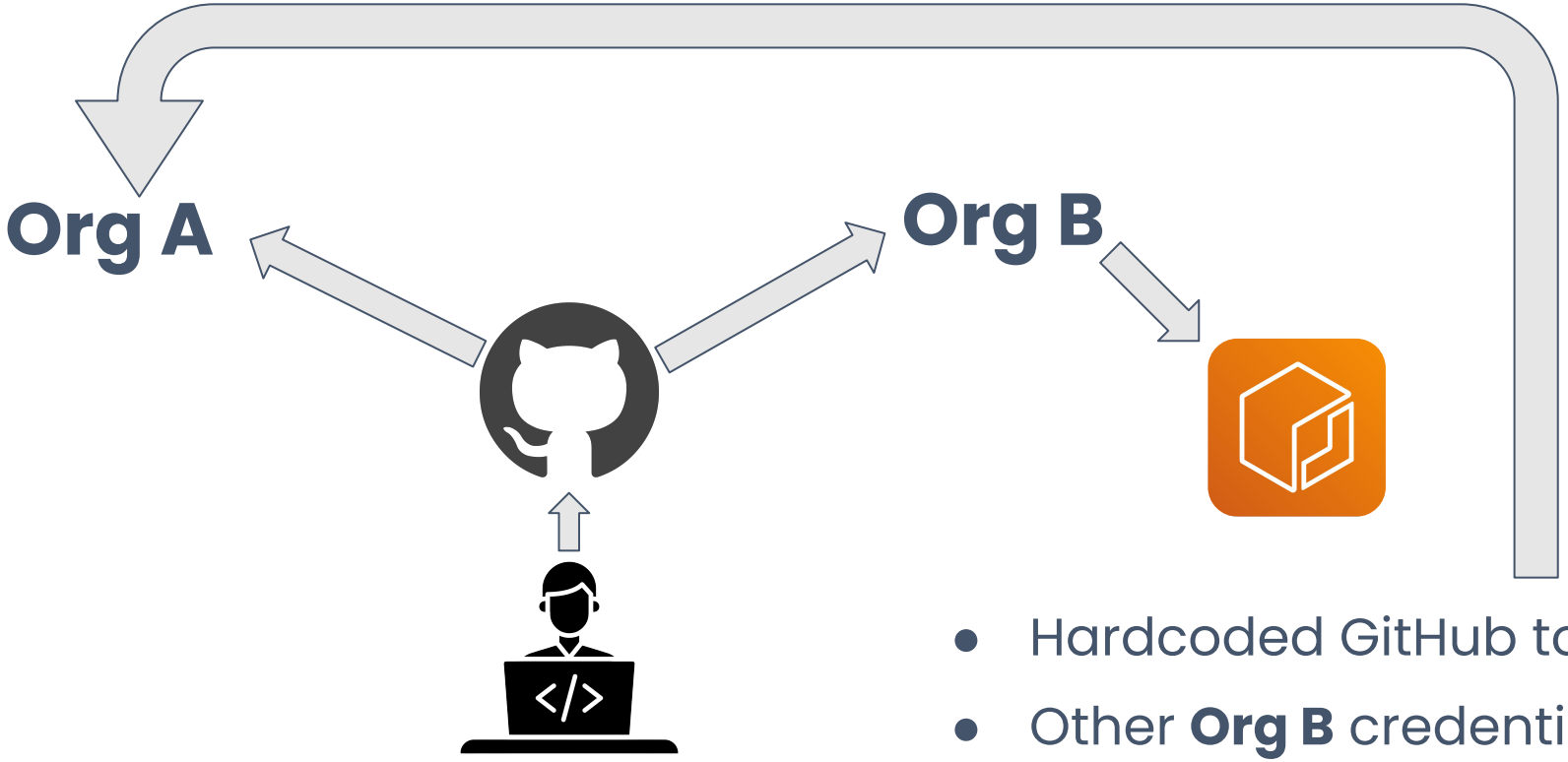
Thank You,

Case Study 2 - Scenario



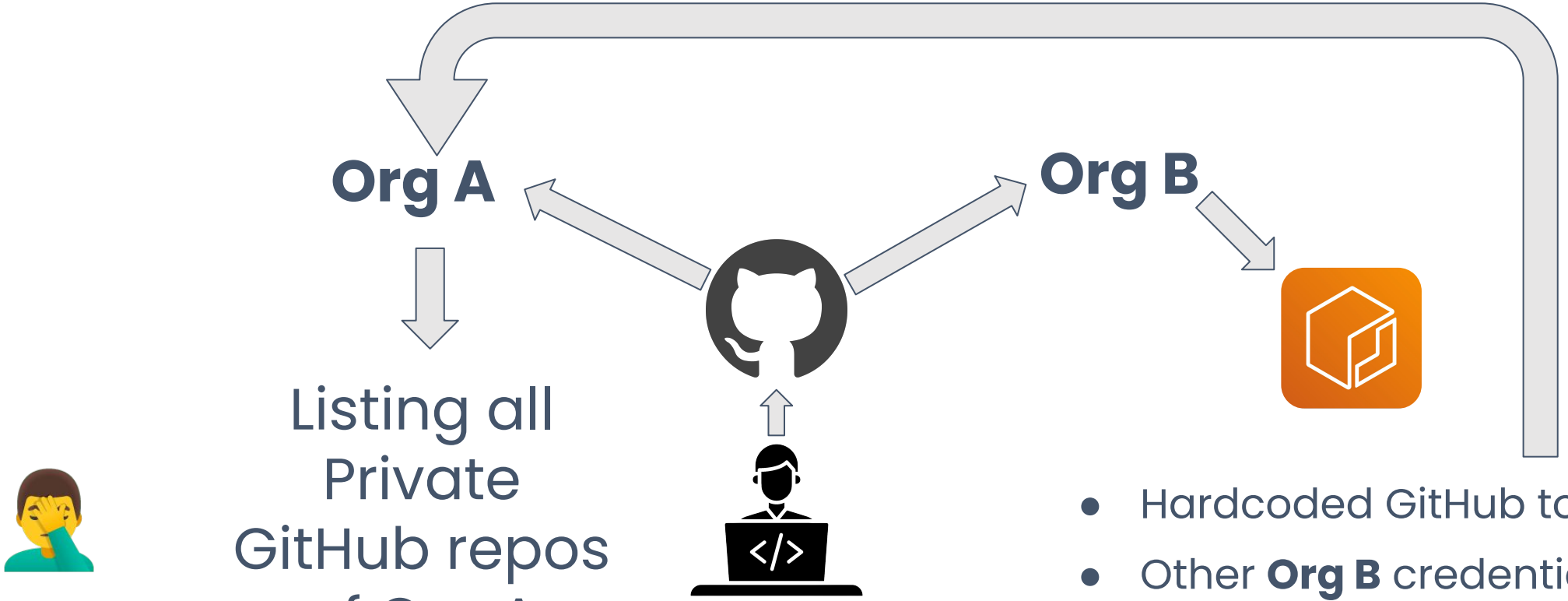
- Hardcoded GitHub token
- Other **Org B** credentials
- Private Repo Source Code

Case Study 2 - Scenario



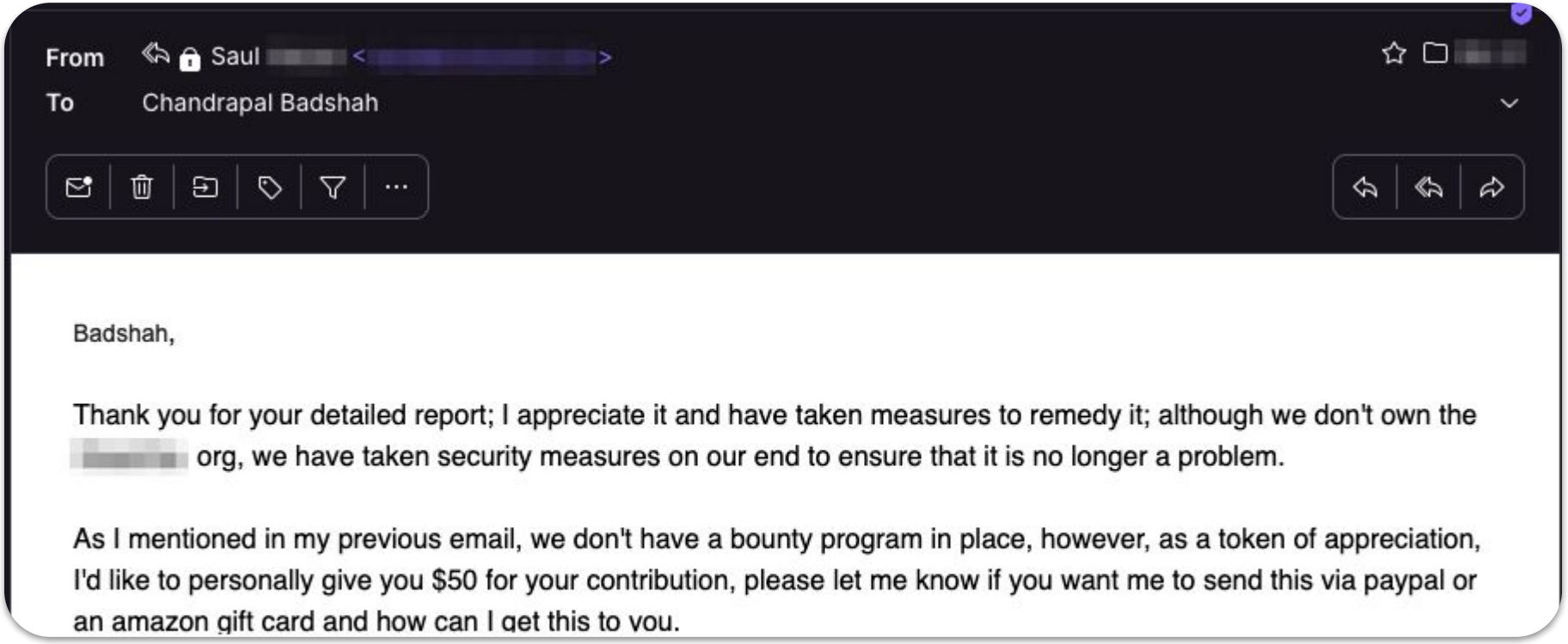
- Hardcoded GitHub token
- Other **Org B** credentials
- Private Repo Source Code

Case Study 2 - Scenario



- Hardcoded GitHub token
- Other **Org B** credentials
- Private Repo Source Code

Case Study 2



From Saul [redacted] <[redacted]>

To Chandrapal Badshah

Badshah,

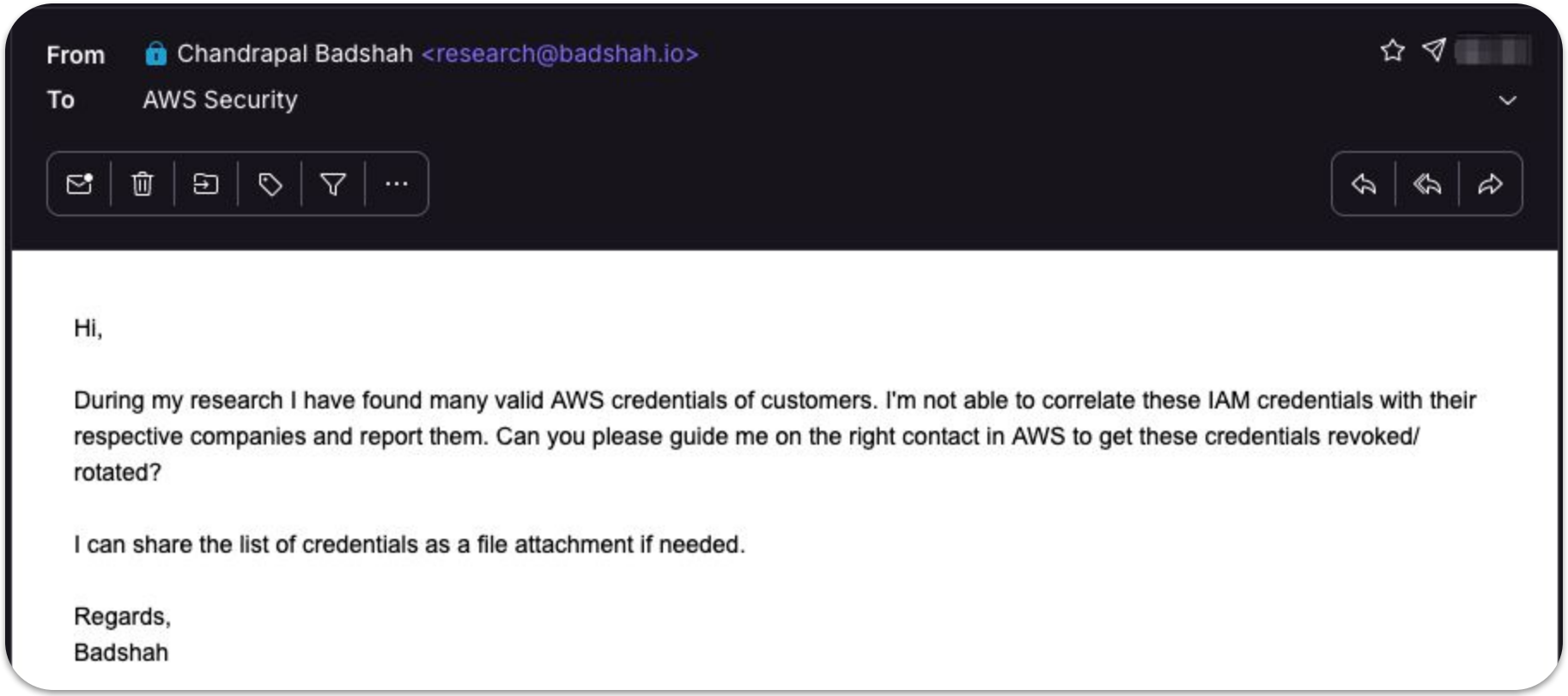
Thank you for your detailed report; I appreciate it and have taken measures to remedy it; although we don't own the [redacted] org, we have taken security measures on our end to ensure that it is no longer a problem.

As I mentioned in my previous email, we don't have a bounty program in place, however, as a token of appreciation, I'd like to personally give you $50 for your contribution, please let me know if you want me to send this via paypal or an amazon gift card and how can I get this to you.

Case: Freelance Developer hardcoded GitHub token
Impact: Access to private GitHub repos.
Status: Fixed by Org A. *No response from Org B.*

Bounty: $50

Case Study 3



From Chandrapal Badshah <research@badshah.io>

To AWS Security

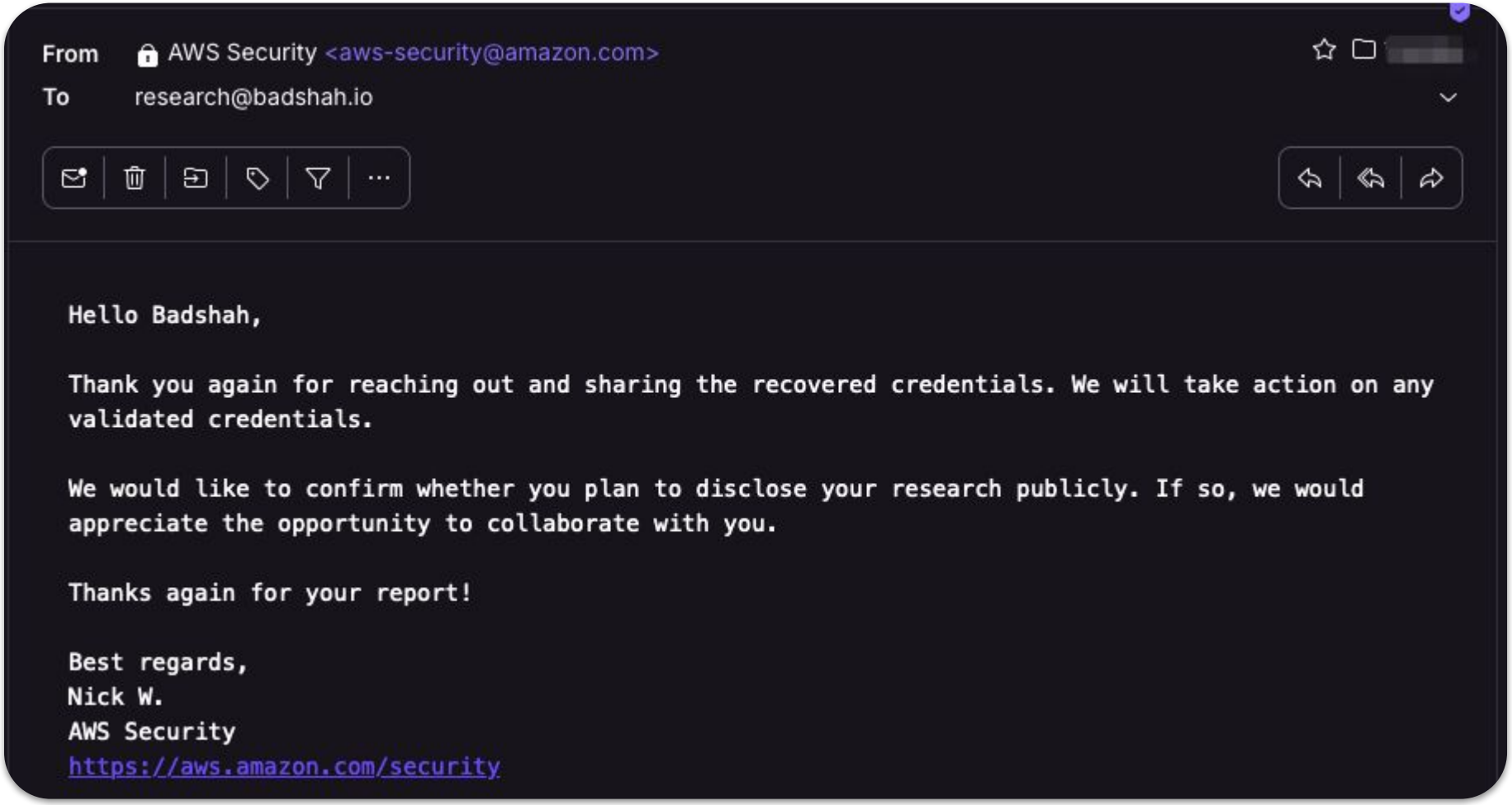
Hi,

During my research I have found many valid AWS credentials of customers. I'm not able to correlate these IAM credentials with their respective companies and report them. Can you please guide me on the right contact in AWS to get these credentials revoked/ rotated?

I can share the list of credentials as a file attachment if needed.

Regards,
Badshah

Case Study 3



From AWS Security <aws-security@amazon.com>
To research@badshah.io

Hello Badshah,

Thank you again for reaching out and sharing the recovered credentials. We will take action on any validated credentials.

We would like to confirm whether you plan to disclose your research publicly. If so, we would appreciate the opportunity to collaborate with you.

Thanks again for your report!

Best regards,
Nick W.
AWS Security
https://aws.amazon.com/security

How to defend?

How to protect yourself?

- Avoid creating public ECR registries
- AWS Inspector doesn't detect hardcoded secrets *yet*.
- Use OSS tools like Trufflehog and Trivy.
- Use container best practices
  - Avoid `COPY . .` OR `ADD . .`
  - Add dockerignore to ignore common folders like "`.git`"
  - Use multi-stage builds
- If budget allows, procure External Attack Surface Management tool which monitors new sources

How to make vuln reporting easier?

- Have a security@ email ID
  - bugbounty@, infosec@, etc can't be found unless you publish them in Privacy Policy
- Publish simple security.txt file
- Publish vulnerability disclosure policy
- Host public bug bounty programs

Less Effort

More Effort

THANK YOU

ANY QUESTIONS?