Lessons Learned
Security at an Open Source Startup

Chris Sandulow @jfalken and **Andreas Nilsson** @agralius

```
$ whois
```

This Talk

InfoSec at a Startup

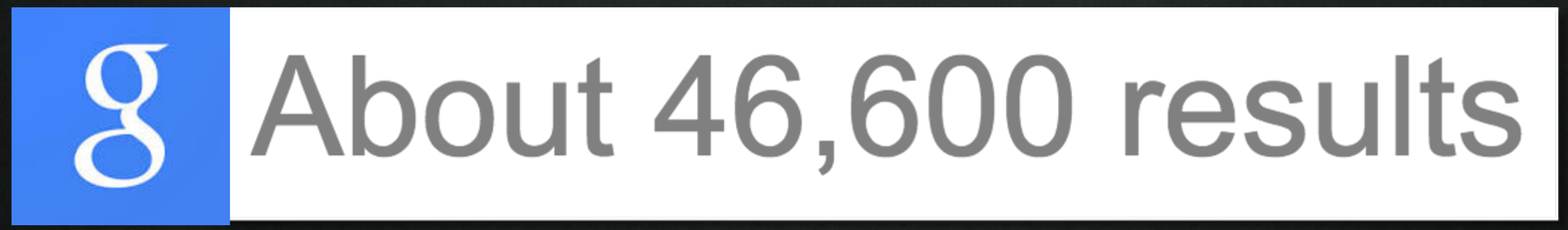
Organizational / Enterprise Security

Application Security / SDL

Tools

“We take security very seriously”

“We take security very seriously”



Size Matters

0 to mid-size-ish

Mid-Cap / Large-Cap

Banks

Retail Orgs

Visionaries

Size Matters

0 to mid-size-ish

Mid-Cap / Large-Cap

Banks

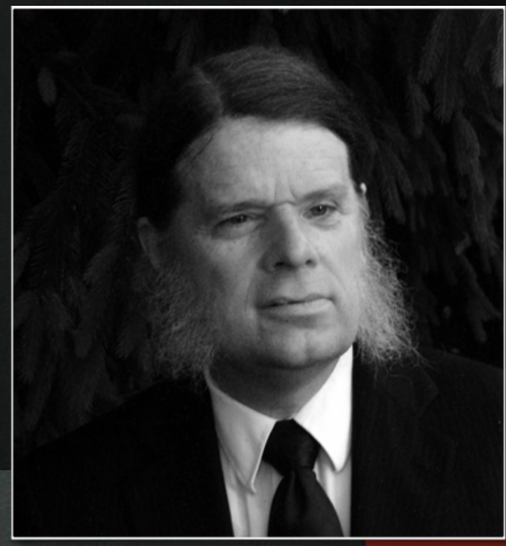
Retail Orgs

Visionaries

People

“I can’t recommend anyone to be a generalist… I don’t think it’s possible … to spectrum generalist.”

Dan Geer



http://geer.tinho.net/geer.sourceboston.txt

When to build a Security Function?

- The Sooner the Better, duh!
  - B2C, sooner than B2B; Regulated sooner than Non
- Don't let compliance solely drive the program
- **You probably need atleast 2, not just 1 person.**
- **Once you're shipping a product/service, you should have a dedicated security person**

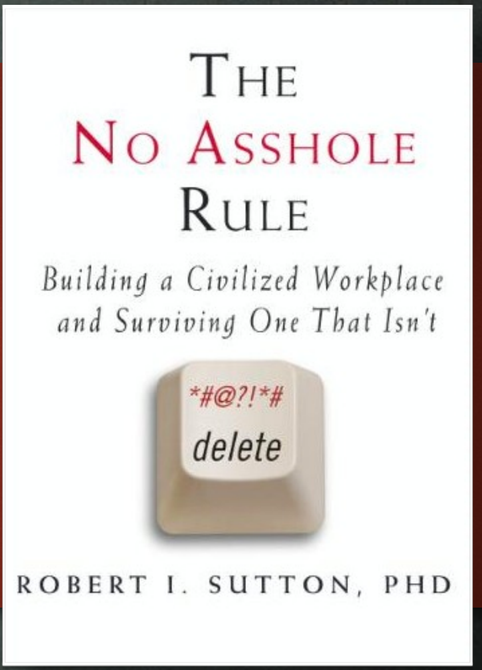
Who? and How Many?

- Purple Unicorn; field diversification

"...there is negative unemployment for highly-skilled security professionals."
— Lee Kushner

- Skills: "Code to an API"
- Capabilities: Builders vs Breakers
- Sizing and Scaling: 1% - 2% of Org*

@iodboi



https://speakerdeck.com/iodboi/crafting-an-effective-security-organisation

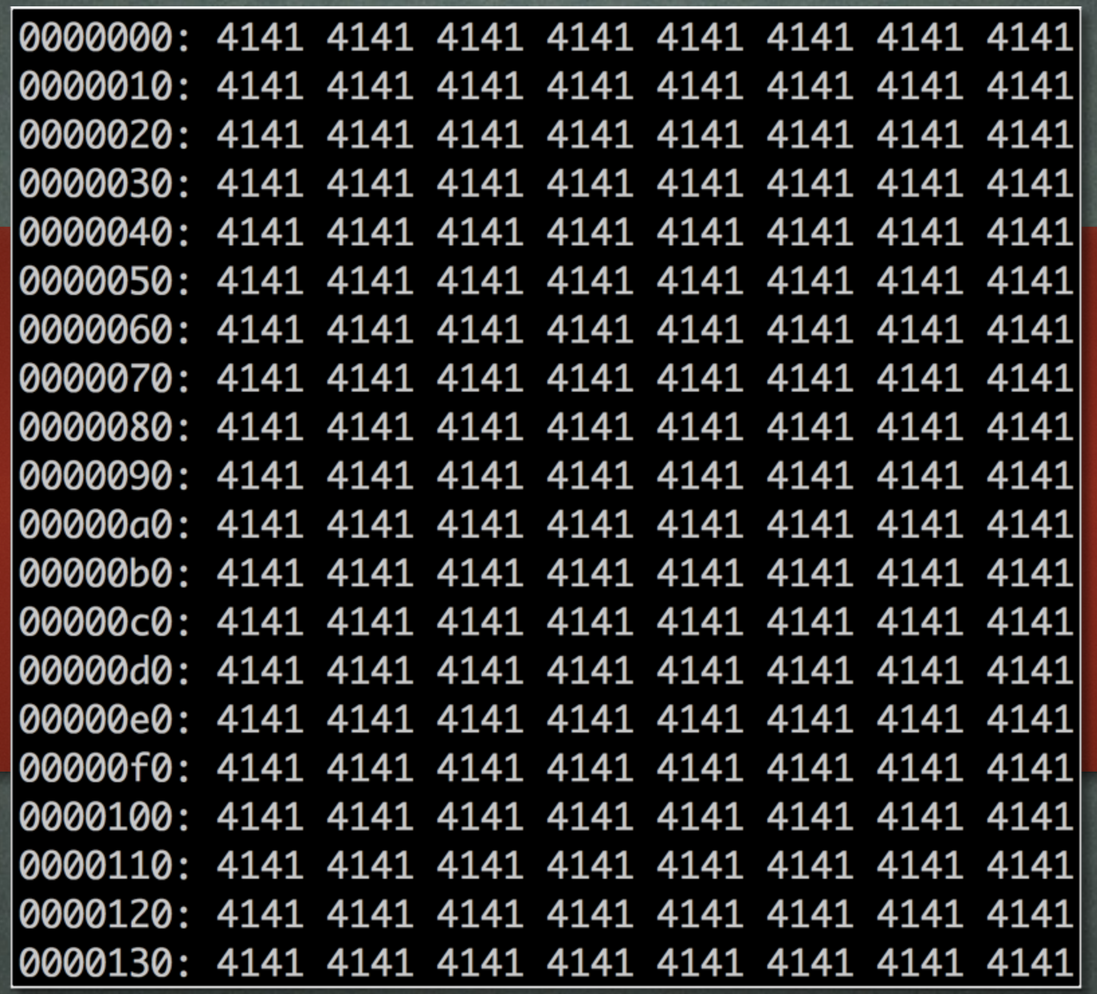
Org / Enterprise Sec

First Step:
InfoSec Program and Baseline

- Baseline program goal; vision; what to achieve
- Driving Adoption and FUD
- Measuring Progress
- Transparency
- ***Attack Driven Defense*** – Zane Lackey

@zanelackey

All the A's



AuthN
Accounts
Assets
Auditing

“…only two effective security technologies, firewalls and 2fa”

@chrisrohlf

https://twitter.com/chrisrohlf/status/574639358101352449

Curtailing Identities

cur·tail — kərˈtāl, verb

Reduce in extent or quantity;

Impose a restriction on.

Accounts: Workforce Lifecycle Management

- Onboarding, Off-boarding, Role-Change(s)
- Reduce Identities; Centralize AuthN and AuthZ
- Federated Identity (SAML)
  - Security Control reduces risk and is more convenient
- HR and IT; Documentation
- Automate process audits

Assets

- Thinking about Assets
  - Cloud vs. Meatspace
  - Servers vs Laptops/Desktop
- Tagging and Schema
  - Diffs

Asset Monitoring
Check the Deltas

- Changes in key infra (e.g., load balancers, ACLs, admin memberships, etc.)
  - Use Version Control tools
  - Develop tooling to automate this reporting
  - Stimulus-response cycle

Vulnerability Scanning

- Once assets are identified, we can scan them
  - Don't expect to find much? Expect to be surprised
  - Triage; don't patch all-things all-the-time
  - Ownership, Patching, Procedures w/ DevOps team
- Reduce exposure, Reduce exposure, Reduce exposure

AppSec

Establish an SDL

- Secure Development Lifecycle (coined by Microsoft)
- Establish one that makes sense for your org.

Requirements

Implementation

Release

Design

Verification

Response

Training

SDL for a Startup

- Internal training of engineers
- Design reviews of all security features
- No commits without peer code review
- Use static code analysis
- Use third party for security assessment

Code Reviews

- Publish and enforce coding style guide
- No commits without review
- Use a common review tool
- In person reviews is the best training

Software Verification

- No new functionality without regression tests
- Automate and run tests in CI system
- Test unexpected and abnormal cases
- Use tools, ASAN (Address Sanitizer), Valgrind

Handling Vulnerabilities

- Tell people how to contact you for Security Issues
  - Responsible Disclosure Policy
  - PGP Key for encrypting sensitive comms
- Have a process to determine when to CVE an issue
- We've all seen tons of bad examples; don't be another. Be transparent and thankful.

Community / Bug Bounty

- Great way to scale a security program
- First time? Set Ceilings
- Be prepared for everything to be hit, firehose of emails
- Be Thankful, celebrate involvement

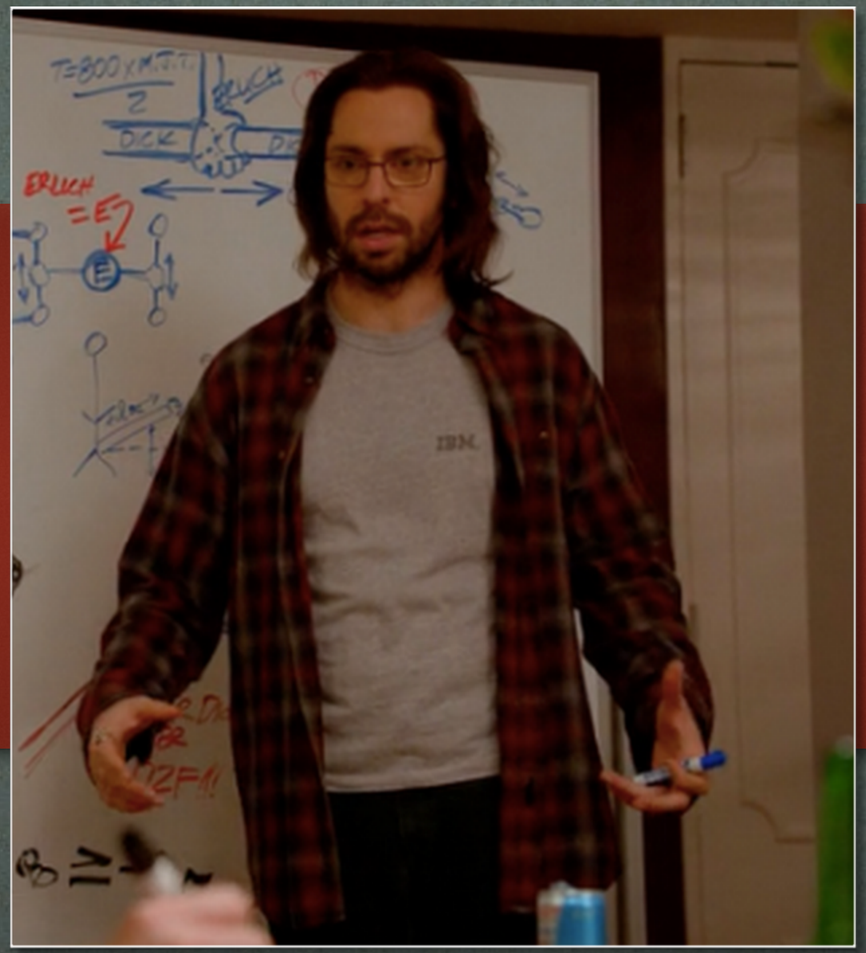
Secure and Sane Defaults

- For anything customer facing:
  - Secure Defaults save you grief later on
  - ‘Secure Defaults’ _will_ hurt adoption as usability is affected; debate this
- Non-Customer Facing (ie, infrastructure, architecture, app-design)
  - You have no excuse to no make wise decisions from the get-go
  - Measure twice, cut once.

Open Sourcing

- Blessing
  - Get testing and reviews for free
  - Not possible to cheat through obfuscation
- and a curse
  - Coordinate security fixes in a public repo
  - Serve a user community and large customers concurrently

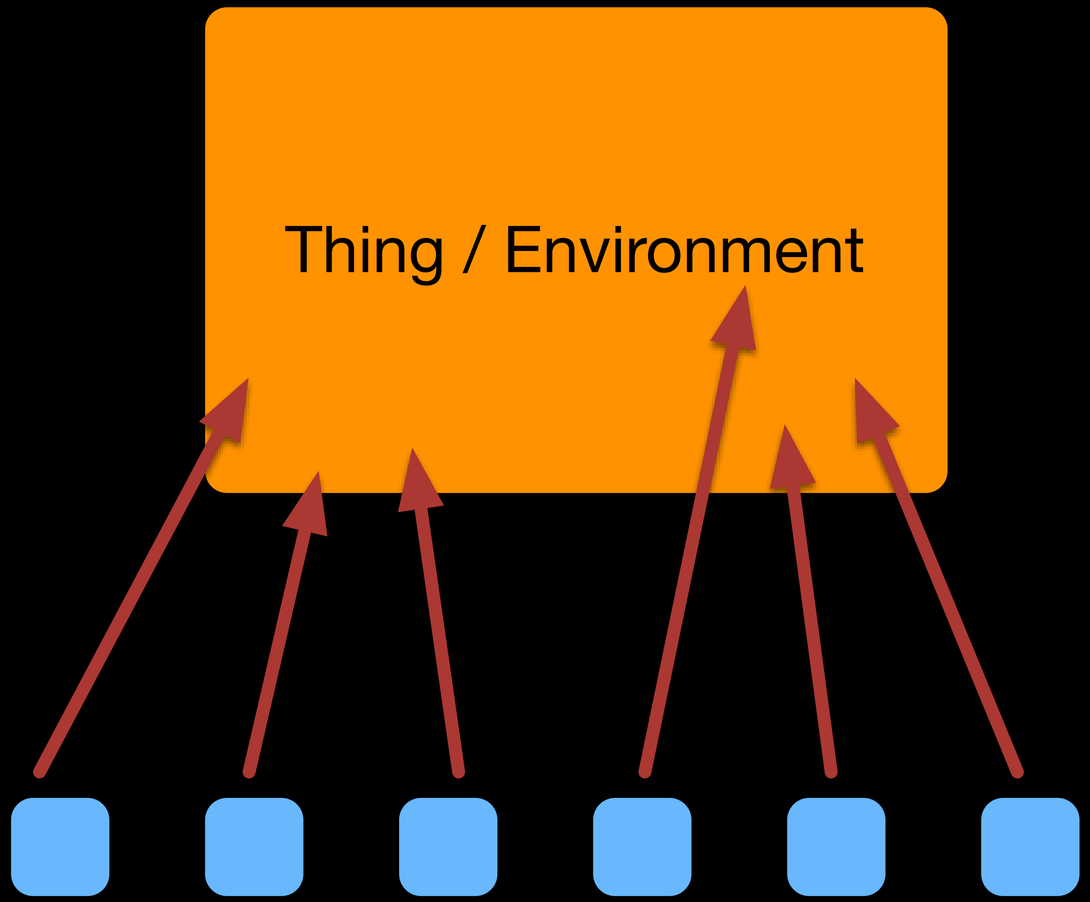
Architecting



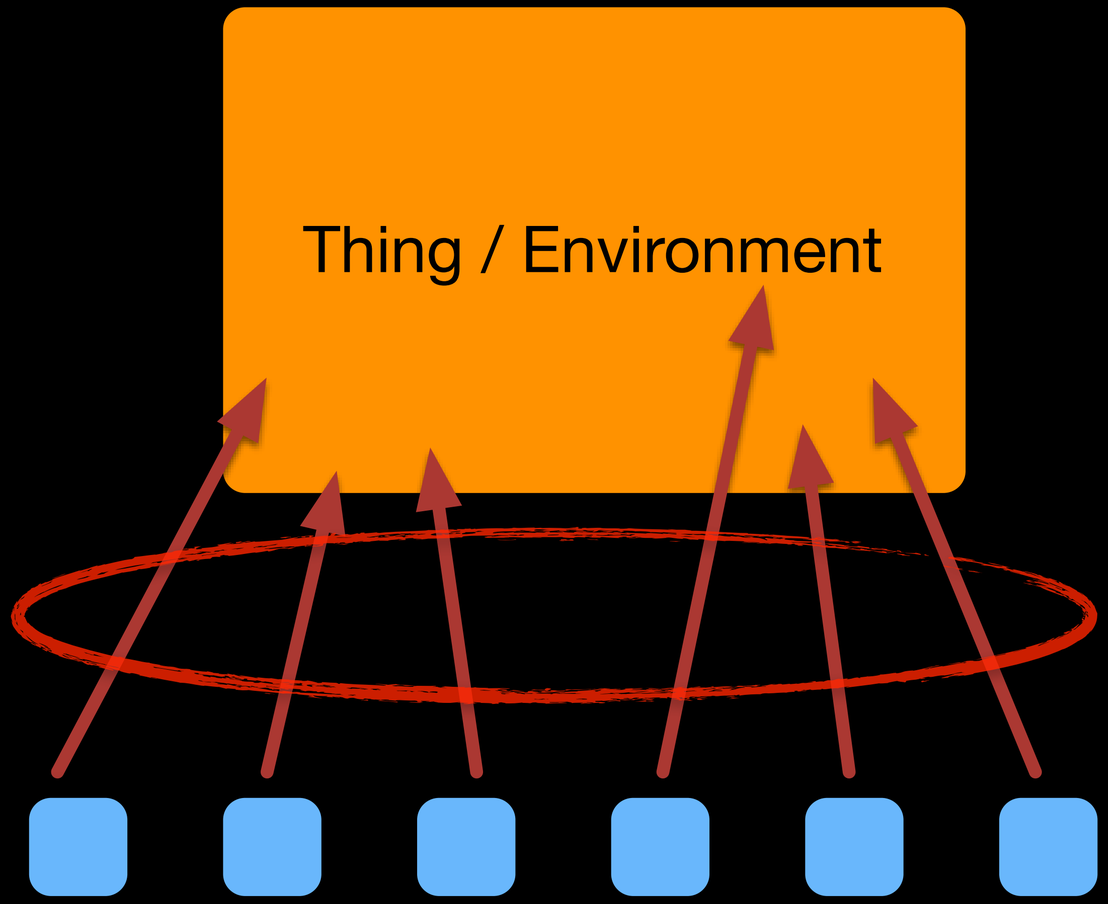
Note on Architectures

- Prefer loosely coupled applications
- Encourage Ephemeral Hosts
- Follow choke-point model
- Prefer auto-scaling / orchestration frameworks
- Find a way to not store secrets, or easily rotate them
  - e.g., ephemeral credentials

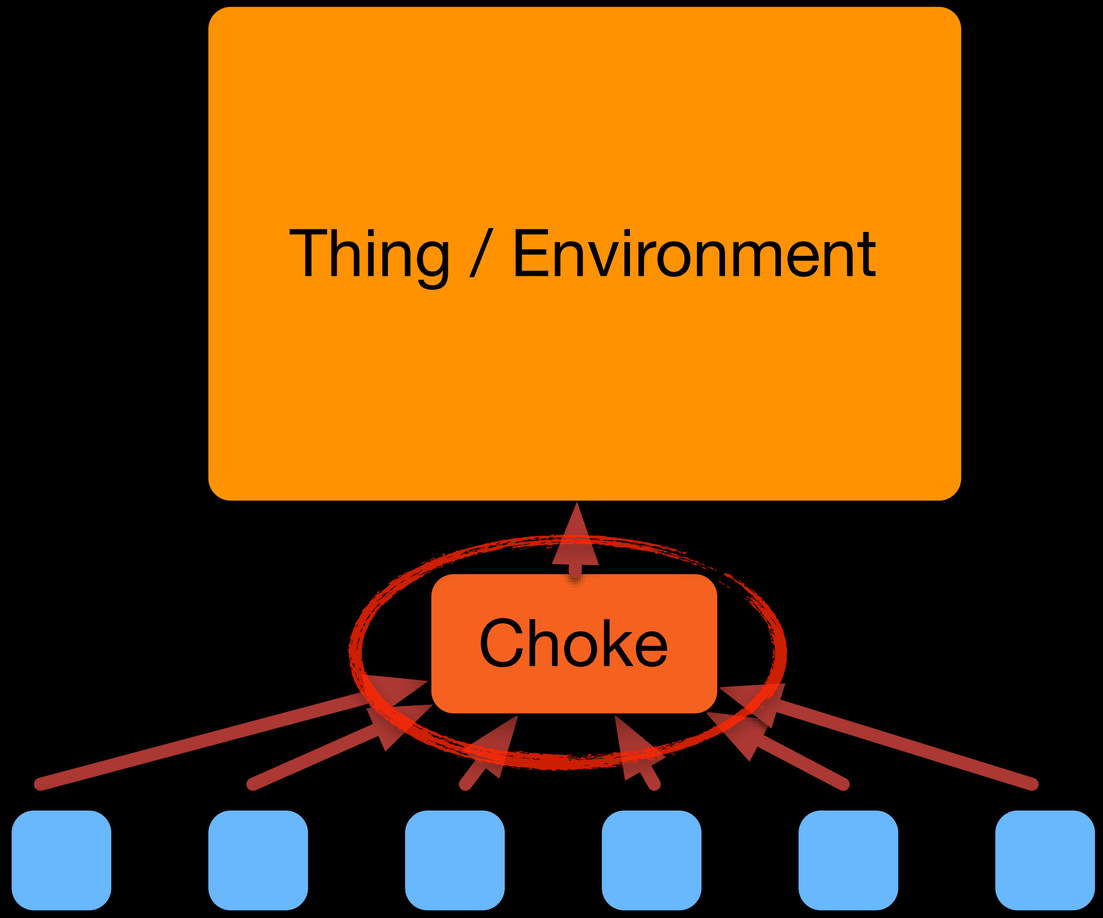
Architecting Choke Points



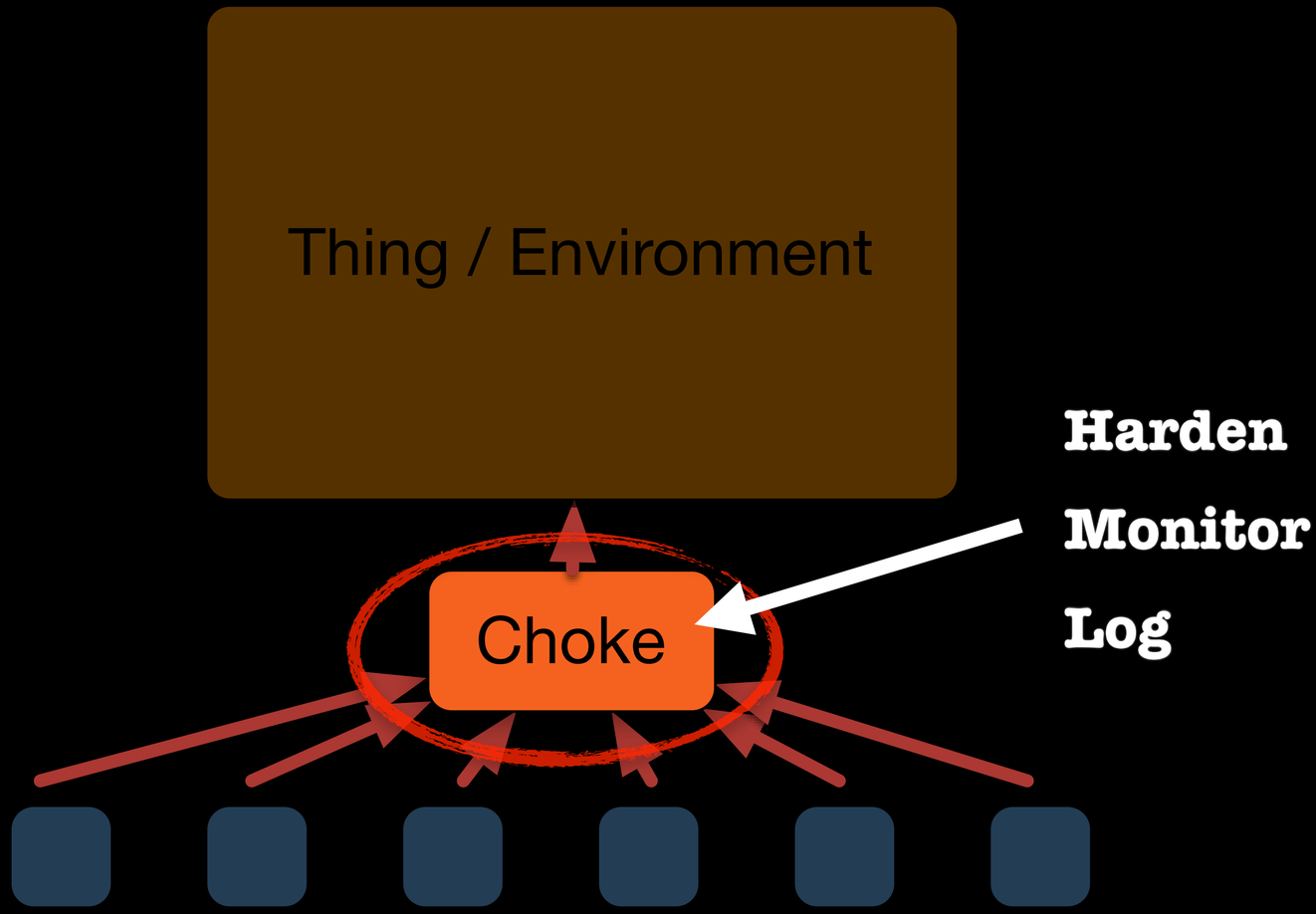
Architecting Choke Points



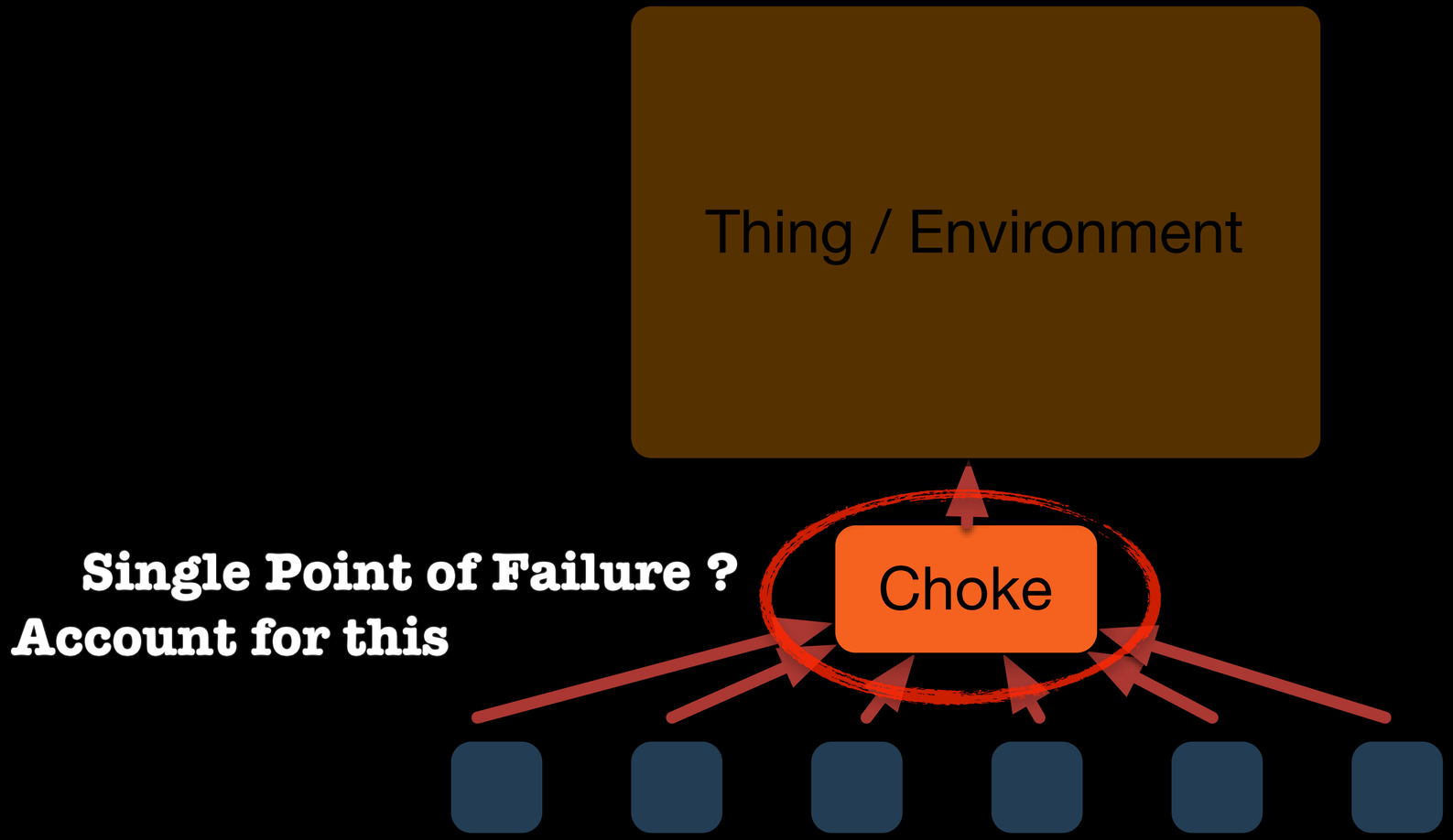
Architecting Choke Points



Architecting Choke Points



Architecting Choke Points



Single Point of Failure ?
Account for this

Assisting Sales

Sales and Customers

- Being part of Sales
- Customer Confidence
- Reducing involvement; scaling w/ sales
- Create a SIG (SharedAssessments.org)
- Create a Q&A/FAQ Repo
- Create training program for Sales Reps

Program Documentation

Reference Slide Only

- InfoSec Program and Policy Basis
- Team and Reporting Structure
- How you secure Assets
- How employees are trained (and background-checked)
- Processes Employee Use (logical controls)
- How Customer Data is Secured
- How systems are hardened and monitored
- Incident Response Plan Overview
- Security Communication and Notifications
- Development Practices / SDLC overview.
- Anything add'l relevant for regulatory reasons
- Docs / References (if applicable)

DevOps?

We’re punting on this; go see

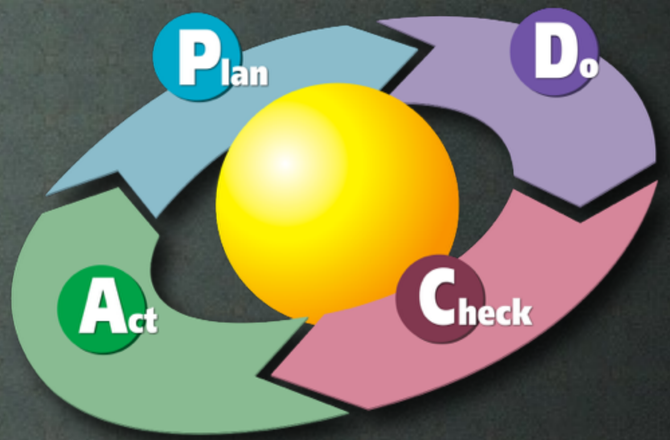
Building a Modern Security Engineering Organization - Zane Lackey

https://www.youtube.com/watch?v=aJ-RGaCiDSM

Auditing and QC

Are your automated processes working properly?

- 1. Manual. 2. Automated. 3. Verify/Improve. 4. Repeat.
- QA, Gap Analysis, API Updates
- Repeatability and Periodicity
- Docs Docs Docs
- Educate team(s) on workflow and process repeatability

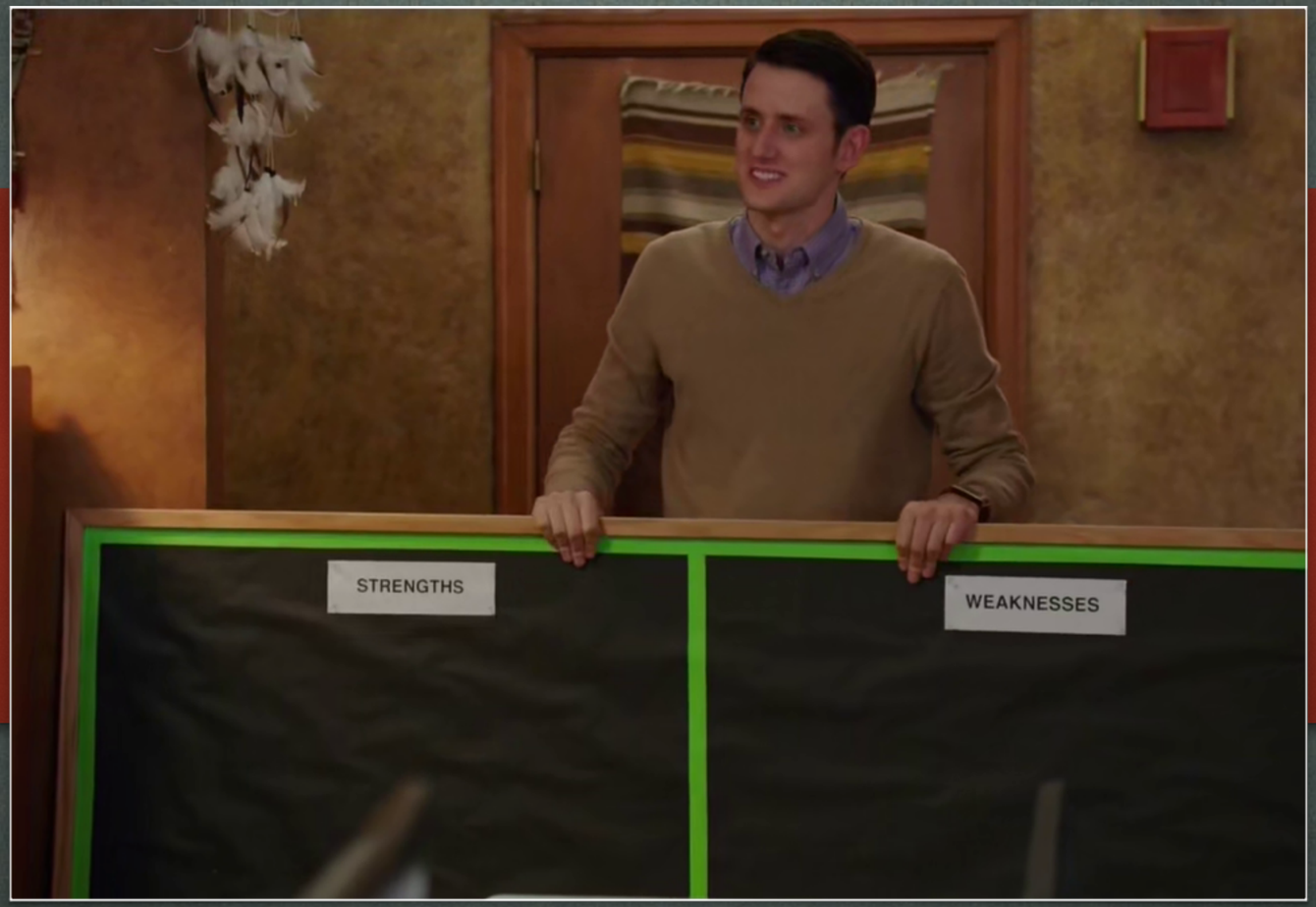


AWS Specific

Reference Slide Only

- Use IAM Roles
- Enable CloudTrail (API logs) and EC2 Config (infra change logs) everywhere
- Use Multi Accounts (prod, dev/QA, logging-account, corp-intra, etc)
- Check blogs very frequently (weekly)
- Review Credential Report (ideally, automate)
- Tons of docs and videos (terminology can be confusing, and you'll hit information overload)
  - Require Tagging; kill things that don't follow your schema
- Not all AWS best practices are realistic, recognize this.

Tools



Tools

- GitHub Commit Crawler
- S3 Key Auditor
- Nessus Process Controller
- AWS Enumeration Library
- HTTPS Uploader Tool
- Baseline Measuring Spreadsheet

Github Commit Crawler

https://github.com/jfalken/github_commit_crawler

- Monitoring for mistaken commits of secrets
- 1. Enumerate Github Org Members
- 2. Check all their recent commits
- 3. Rinse, Repeat
- Easy install and config; all packaged in a Docker Container

Github Commit Crawler

User	Audit_Date	Matched Keyword	File	Other Links
sr527	2015-04-01	password	auth_test.go	Diff Commit HTML
llvtt	2015-04-01	password	sharded_clusters.py	Diff Commit HTML
llvtt	2015-04-01	password	sharded_clusters.py	Diff Commit HTML
gabrielrussell	2015-04-01	password	nid.go	Diff Commit HTML
gabrielrussell	2015-04-01	secret	nid.go	Diff Commit HTML

- 3. Rinse, Repeat
- Easy install and config; all packaged in a Docker Container

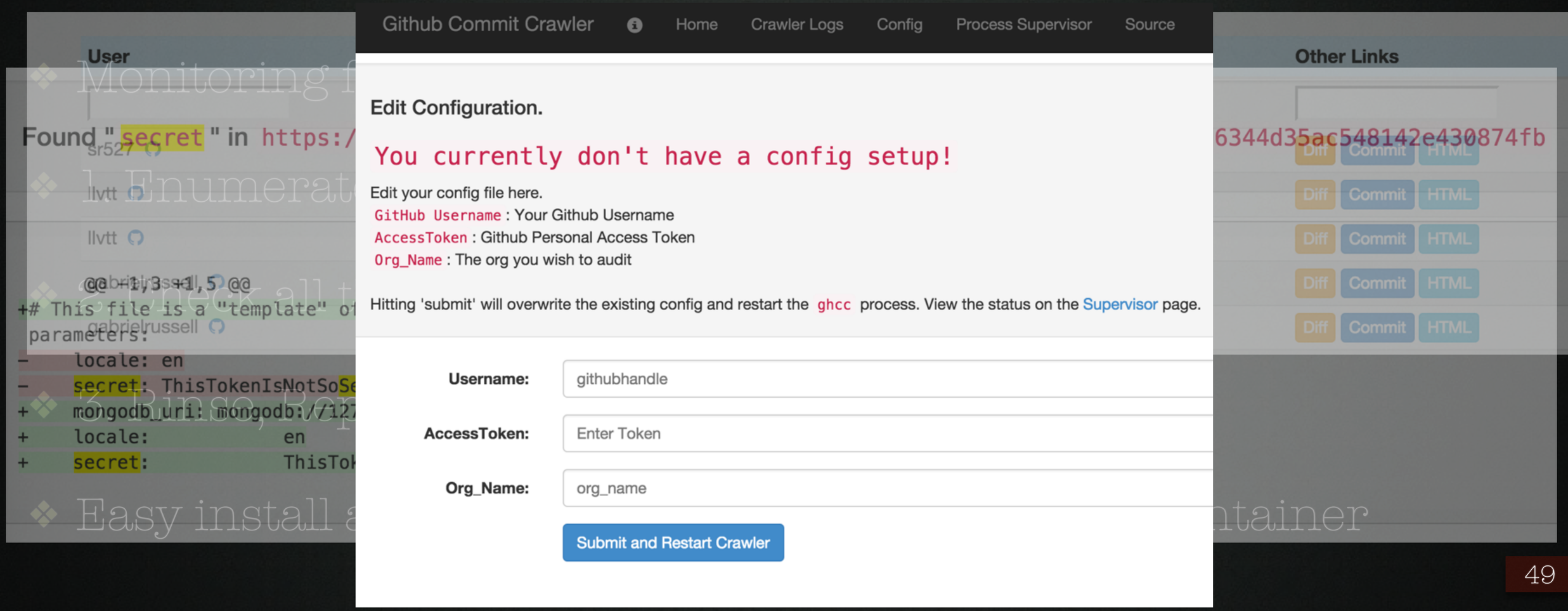
Github Commit Crawler

User	Audit_Date	Matched Keyword	File	Other Links

Found " secret " in https://api.github.com/repos/jmikola/arsjerm/commits/273308a0a265eecd6344d35ac548142e430874fb

```
@@ -1,3 +1,5 @@
+# This file is a "template" of what your parameters.yml file should look like
 parameters:
-    locale: en
-    secret: ThisTokenIsNotSoSecretChangeIt
+    mongodb_uri: mongodb://127.0.0.1:27017
+    locale:            en
+    secret:            ThisTokenIsNotSoSecretChangeIt
```

Github Commit Crawler

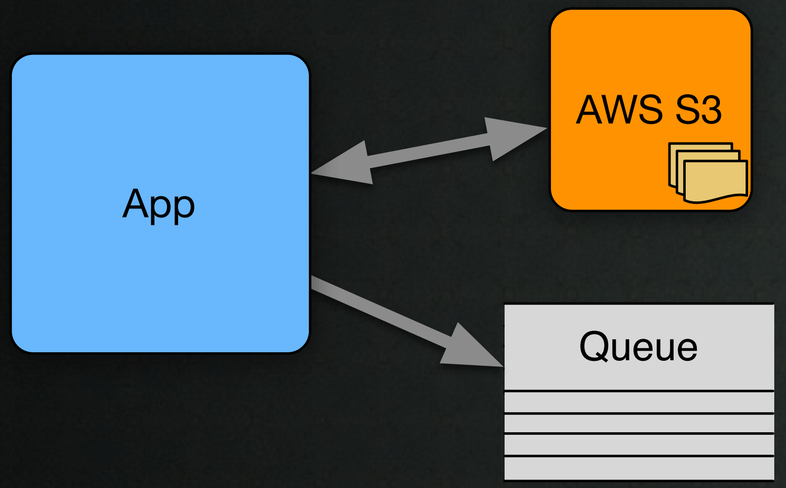


S3 Key Auditor

https://github.com/jfalken/s3_key_audit

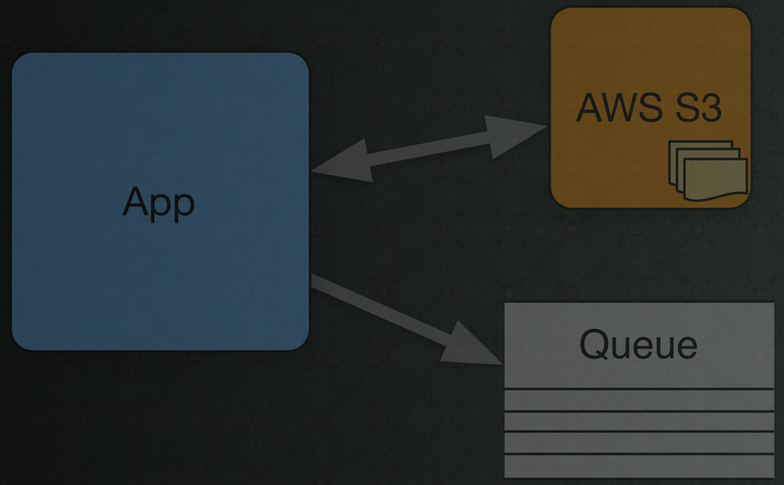
- S3 usage creeps over time; be mindful of whats going on
- ACLs and permissions on buckets don't matter, each key can have individual permissions
- Therefore; you need to audit every single key's permissions
- `s3_key_auditor.py`
  - Enumerate all keys in all buckets in all accounts
  - Multi-threaded app to burn through the queue as fast as possible
  - 2 Millions key ACL checks per hour (MBP 16GB RAM)

S3 Key Auditor



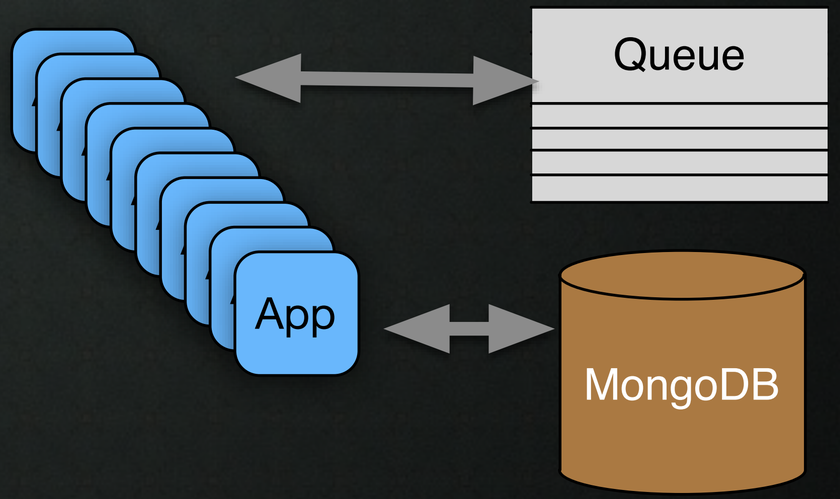
Enumerate All Accounts, Buckets and Keys, populate Queue

S3 Key Auditor

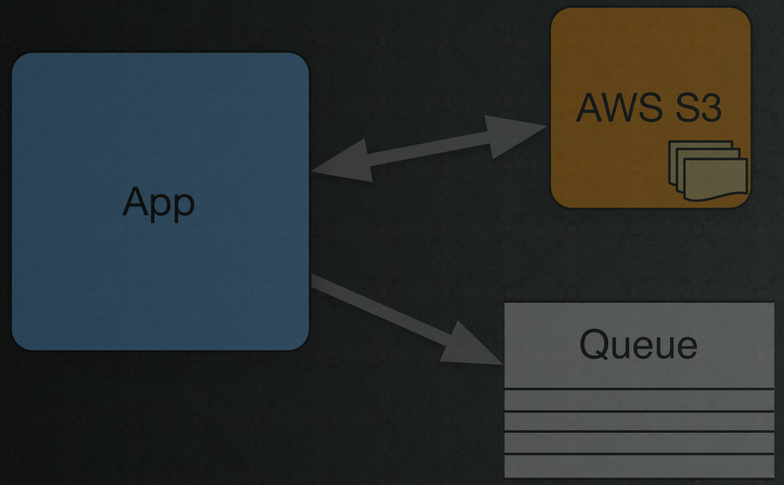


Enumerate All Accounts, Buckets and Keys, populate Queue

Burn down the queue, results stored in MongoDB.

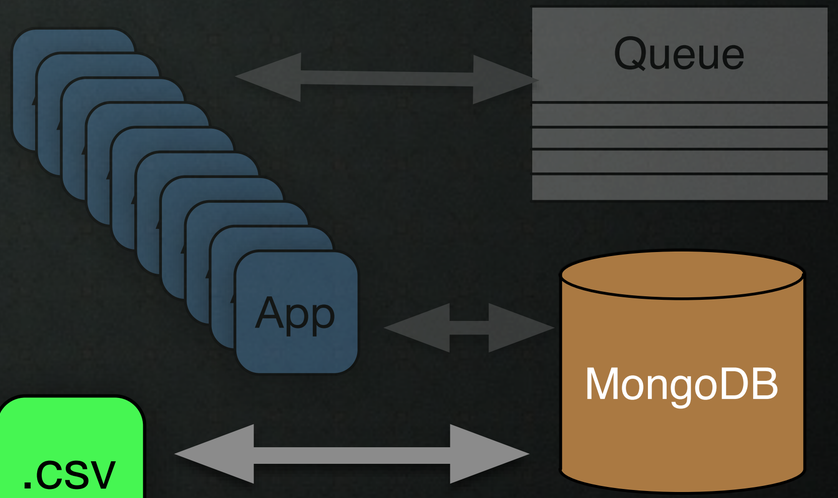


S3 Key Auditor



Enumerate All Accounts, Buckets and Keys, populate Queue

Burn down the queue, results stored in MongoDB.



Generate .csv file

Nessus Process Controller

https://github.com/jfalken/nessus_process_controller

- Enumerate, Launch Scans, Get Results, continuously.
- Supports AWS enumeration and also manually entered assets
- Scan as frequently as you like
- Get results in HTML, XML and JSON. Use output to drive other audit processes, cross-reference, etc.

AWS Enumeration Library

https://github.com/jfalken/aws_enumeration_lib

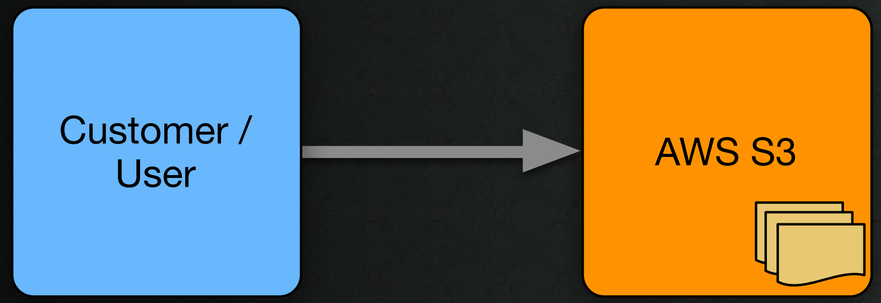
- We advocate multiple AWS Accounts
- Enumeration becomes burdensome
- AWSEnumerationObject
  - Python library that wraps `boto` to make multi account and region asset identification easier

HTTPS Uploader

https://github.com/jfalken/s3_https_upload_portal

- Enable fast uploads to S3, with a minimal maintenance as possible.
- TLS, Encryption at Rest, Data Life Cycle Management
- Direct to S3 HTTPS Uploads, “GUI” drag-drop. curl support
- Thin web app to list/download file (employees don't need API creds)
  - Py flask app on Elastic Beanstalk (ephemeral instance)

HTTPS Uploader



Customer Uploads (presigned URLs)

TLS, Encrypt At Rest, Direct to S3

Employee Access

Enable access for employees w/o S3 Creds

HTTPS Uploader

MongoDB Support File Upload

Upload from your Browser

➡ **Drop files** to upload

(or click)

Upload via curl

Alternatively, you can HTTPS POST using a tool like curl, via the following syntax. Replace `file.name` with the file you wish to upload

```
curl -F "key=uploads/test/\${filename}" \
     -F "AWSAccessKeyId=AKIAJNKYRGMCMHNLRI6Q" \
     -F "acl=private" \
     -F "x-amz-server-side-encryption=AES256" \
     -F "policy=eyJleHBpcmF0aW9uIjogIjIwMTUtMTAtMjBUMDA6MDA6MDBaIiwKICAgICAgI
CAgICAgICAgICAgImNvbmRpdGlvbnMiOiBbCiAgICAgICAgICAgICAgICAgICAgICAgICAgIC
AgICAgeyAiYnVja2V0IiA6ICJjaHJpc2ViIiB9LAogICAgICAgICAgICAgICAgICAgICAgICA
gICAgICAgIFsgInN0YXJ0cy13aXRoIiwgIiRrZXkiLCAidXBsb2Fkcy90ZXN0LyJdLAogICAgICAg
ICAgICAgICAgICAgICAgICAgICAgICAgIHsgImFjbCIgOiAicHJpdmF0ZSJ9LAogICAgICAgI
CAgICAgICAgICAgICAgICAgICAgICAgIHsgIngtYW16LXNlcnZlci1zaWRlLWVuY3J5cHRpb2
4iIDogIkFFUzI1NiIgfSwKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICBbICJjb25
0ZW50LWxlbmd0aC1yYW5nZSIsIDAsIDUzNjg3MDkxMjAgXQogICAgICAgICAgICAgICAgICAg
ICAgICAgICBdCiAgICAgICAgICAgICAgIH0=" \
     -F "signature=
     -F "file=@file.name" \
     https://chriseb.s3.amazonaws.com
```

All files uploaded here are transmitted encrypted and stored encrypted at rest. Files are only accessible by MongoDB Inc. Employees.

Baseline Spreadsheet

Fork me on GitHub

- Based on ISO 27002:2015 Controls
- Loosely establish questions based on the control domains
- Scope yourself, question yourself, do this regularly
- Reflect on the output; wheres your lowest point? Make progress on that next quarter
- REPO: TBD

Moving Forward …

Open Source ‘Book’

The content of these slides will be made into a ‘book’ format, open sourced on GitHub for collaboration. End goal: to help others starting out in this area.

References

- Give Credit Where its due
- Check out their talks and work
  - Zane Lackey
  - Rich Smith
  - Haroon Meer
  - Chris Evans
  - Dan Geer
  - Chris Rohlf

end