Confidence as Code:

Automated Security Testing in Cloud Environments

Brad Geesaman

@bradgeesaman

About

Previously

- Network Security Engineer
- Penetration Tester/Security Consultant

Past 8+ Years

- Cloud Infrastructure Administrator
- “DevOps” practitioner *
- Ethical Hacking Educator

Past Three Years

- Researching Cloud Security Issues with Containers and Container Orchestrators
- Independent Consulting - Securing Containers and Kubernetes

Previous Talks

- KubeCon NA 2017 - Hacking and Hardening Kubernetes Clusters by Example: https://youtu.be/vTgQLzeBfRU
- BlackHat USA 2018 - Detecting Malicious Cloud Account Behavior https://i.blackhat.com/us-18/Thu-August-9/us-18-Geesaman-Detecting-Malicious-Cloud-Account-Behavior-A-Look-At-The-New-Native-Platform-Capabilities.pdf

Contact Info: @bradgeesaman[@gmail dot com]

@bradgeesaman

Agree or Disagree

1. Securing workloads in the Cloud is easier than On Premise.

Agree or Disagree

2. Security teams have better tools and processes in the cloud.

Agree or Disagree

3. It’s easier to prove that cloud infrastructure is secure.

Traditional vs Cloud-Native Infrastructure

Traditional

- Built by humans
- Built over months/years
- IPs have identity
- Configuration issues patched “live”
- Upfront payment
- “Static” architecture diagram
- Compliance frameworks fit this model
- Security mostly understands the env
- Very mature security vendor offerings

Cloud-Native

- Built by code
- Built/rebuilt in minutes
- IPs have no identity
- Configuration issues fixed via redeploy
- Pay as you go
- “Dynamic” architecture
- Compliance frameworks struggle to fit
- Security often lacks knowledge
- Security vendors often startups

Agree or Disagree

1. Securing workloads in the Cloud is easier than On Premise.
   a. Depends on skills/experience
   b. Depends on approach
   c. Shipping velocity and complexity make it harder

Agree or Disagree

2. Security teams have better tools and processes in the cloud.
   a. Cloud APIs provide better building blocks
   b. Tools and processes often don’t keep up
   c. Vendor space is rapidly changing and maturing

Agree or Disagree

3. It’s easier to prove that cloud infrastructure is secure.
   a. Compliance frameworks starting to be cloud aware
   b. Auditors often don’t “speak cloud” well
   c. Comparing security posture to on-premise is like apples and oranges

=> Lack of Confidence

Regain Confidence through *codified testing*

Types of Security Testing

- Vulnerability Scanning
- Security Scanning
- Penetration testing
- Risk Assessment
- Security Auditing
- Posture Assessment
- Ethical hacking

https://www.guru99.com/what-is-security-testing.html

Types of Security Testing

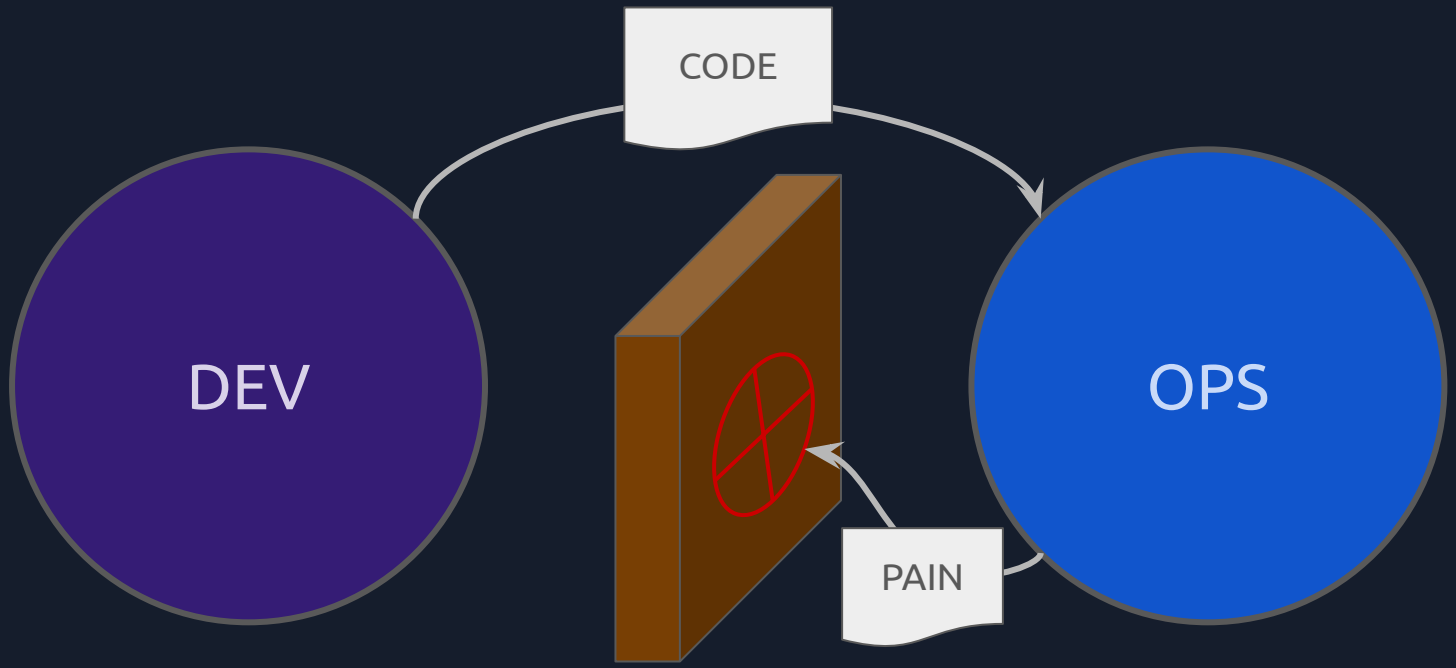
https://www.guru99.com/what-is-security-testing.html

A New Approach is Needed

1. Look for efficiencies and commonalities
2. Start small and iterate quickly
3. Embrace Automation
4. Perfect is the enemy of Good

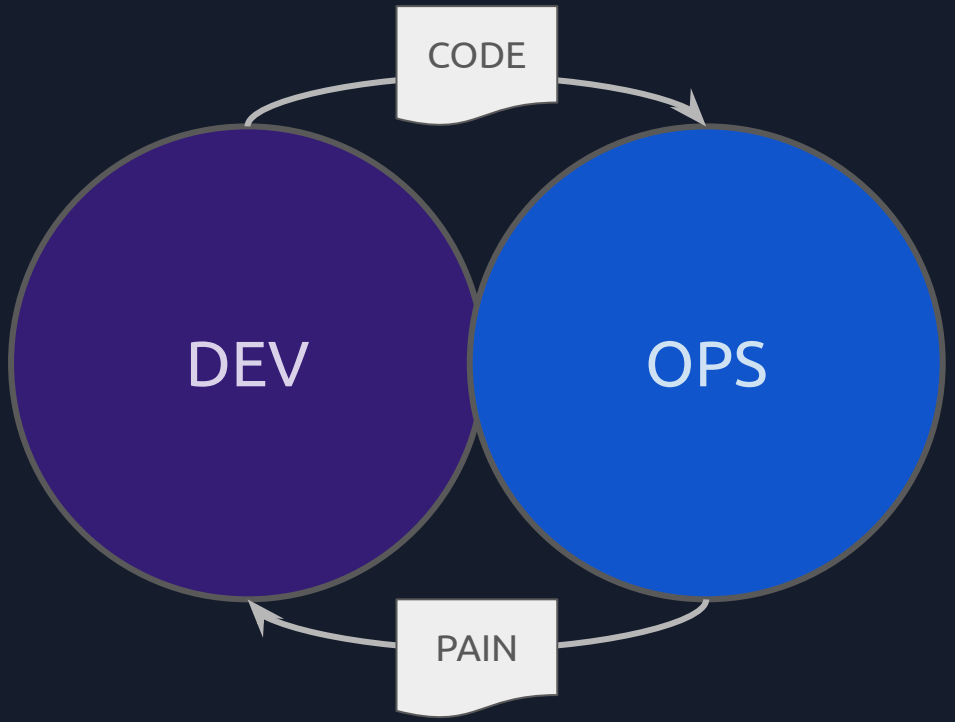
Take a page from the DevOps transformation

DevOps Transformation



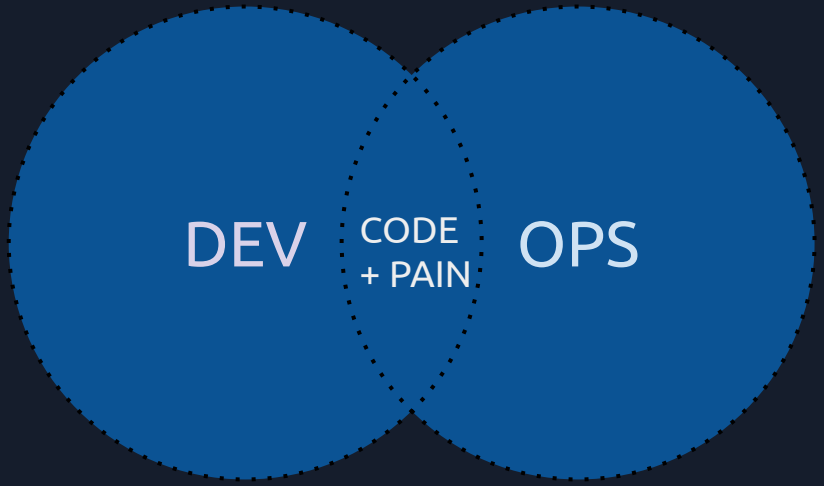
No shared pain, no feedback loop

DevOps Transformation



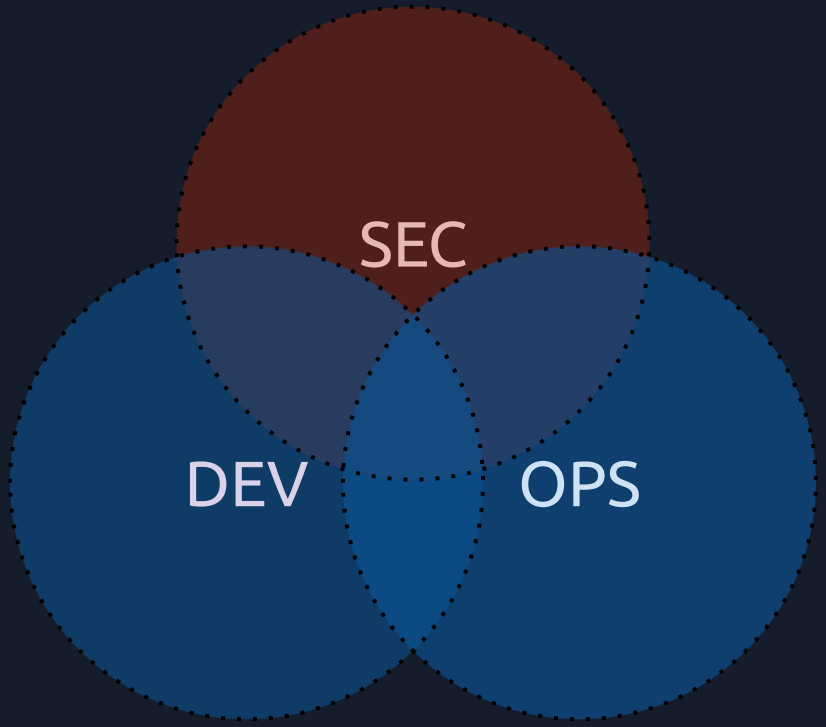
Shared “pain” is a feedback loop to improve processes

DevOps Transformation



Combined culture, automation, measurement, and sharing

DevSecOps Transformation

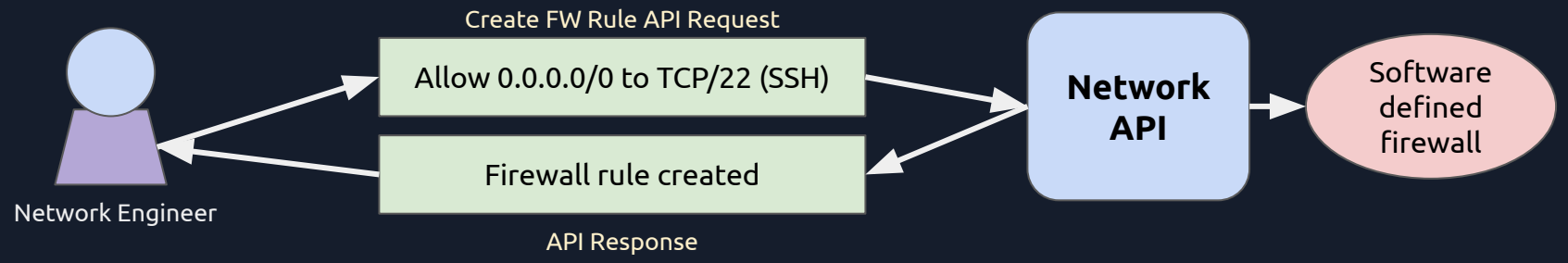


Turn security goals into *shared* security goals

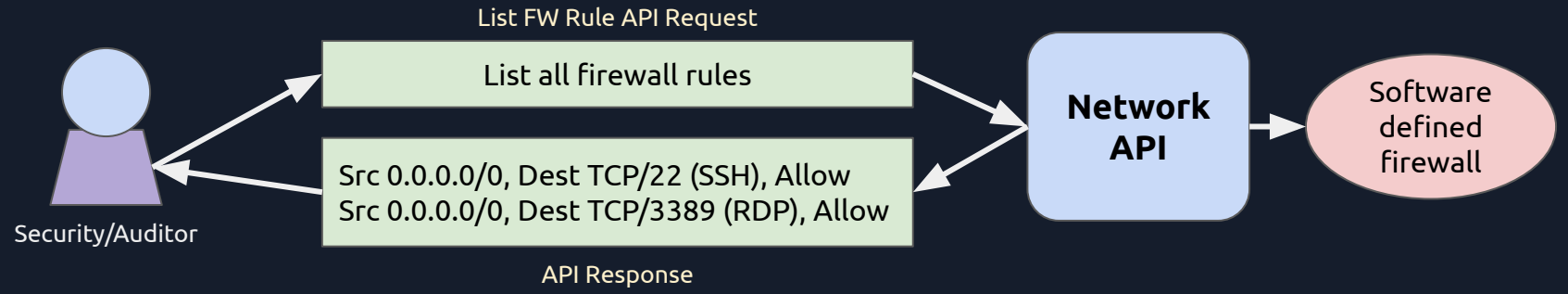
When infrastructure is created via an API...

Many cloud security tests can be done by *asking the same APIs!*

Create a firewall rule using the API



Audit the firewall rules using the same API



Paths to Security Testing Success

- Clarify your definition of compliance
- Start with simple but important questions or a small number of compliance objectives
- Codify and automate all configuration tests
- Test often and make the results visible

Paths to Security Testing Failure

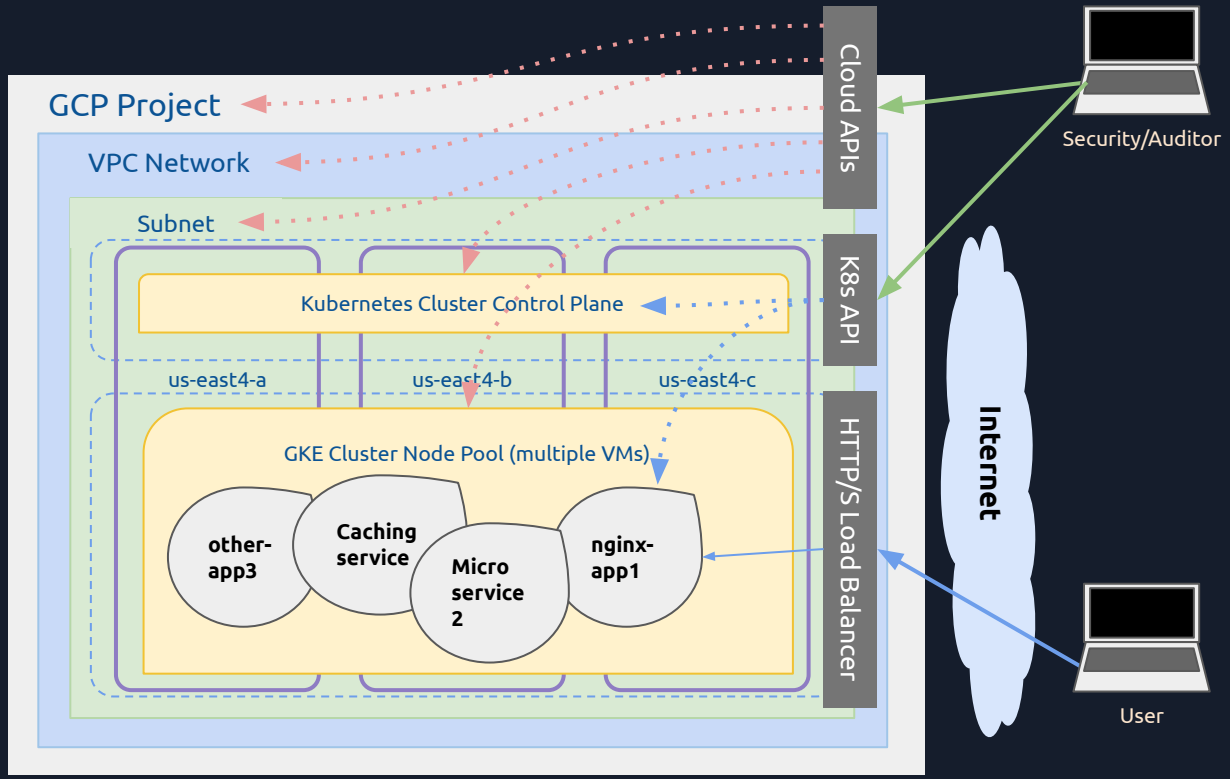
- Using spreadsheets
- Running tests manually/infrequently
- Allowing the test suite to stay broken
- Shifting Left before culture and tools mature

Chef Inspec

"Is **Compliance as Code** - a human readable language for automating the continuous testing and compliance auditing of your entire infrastructure."

https://github.com/nathenharvey/introduction-to-inspec/blob/master/pdf/03-Compliance-as-Code.pdf

Demo Environment



1. GCP Project Settings
2. Network and Subnet Settings
3. GKE/Kubernetes Cluster Configuration and Hardening
4. GKE/Kubernetes Node Pool Configuration and Hardening
5. Kubernetes Workload (nginx app)

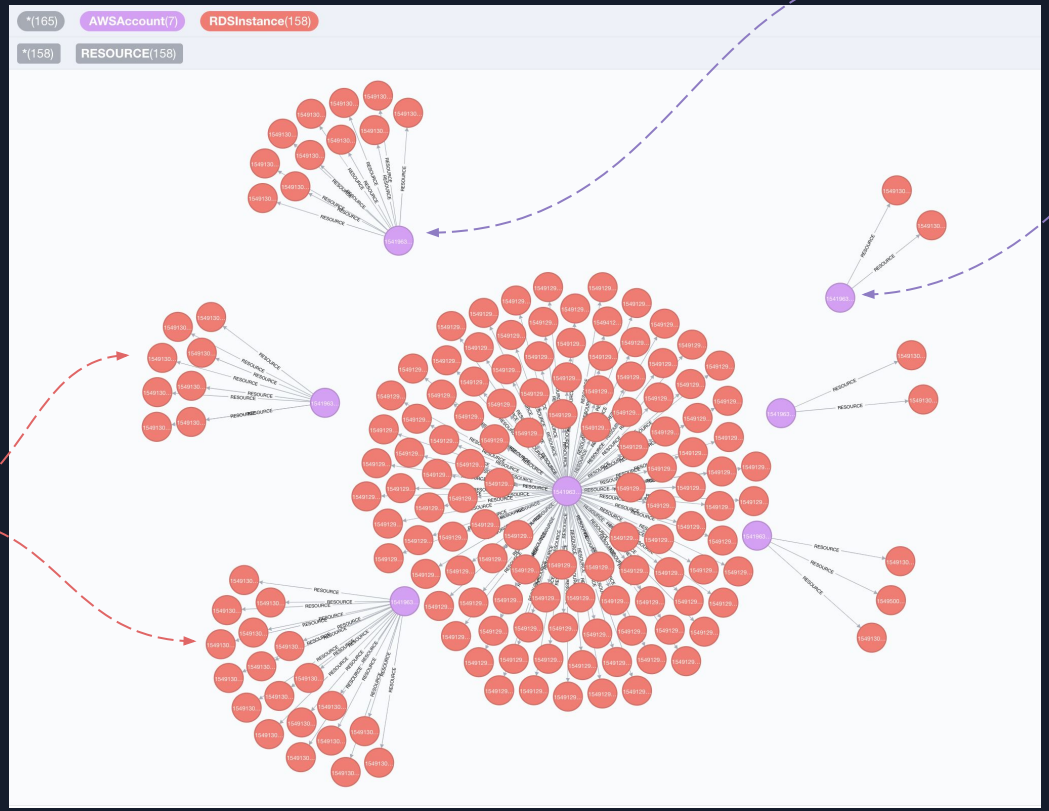
Demo

Run Inspec against a sample cloud architecture to validate proper security configuration

And then I saw Lyft's "Cartography" BsidesSF talk: https://github.com/lyft/cartography

RDS Databases per AWS Account

AWS Accounts



AWS RDS Instances

https://github.com/lyft/cartography

In Search of the “Perfect Inventory”

- We deploy infrastructure and applications from code via a declarative API
- We know about all changes in real time
- We understand and map the relationships between all cloud resources
- We have a way to query that information cleanly in code

A *"Perfect"* Configuration Inventory answers a *TON* of interesting questions

Answering Interesting Questions

- Show me all the Windows Instances with public IPs and firewall rules that allow TCP/3389 from all IPs.
- Show me all the public cloud storage buckets
- Show me all the users that have the admin role to an API service.
- Show me all the containers in my Kubernetes cluster that do not have a patch for CVE-NNNN-NNNN

Future with Accurate Inventory

- Automated Compliance Reporting
- Programmatic Attack Path Mapping
- Auto-generated threat models
- Automated Risk Scoring
- Simulating Changes and measuring Risk Score change

Thank you!

Links:

1. Chef Inspec - https://www.inspec.io/
2. Chef Inspec GCP Resource Pack - https://github.com/inspec/inspec-gcp
3. Chef Inspec AWS Resource Pack - https://github.com/inspec/inspec-aws
4. Inspec Kubernetes (K8s) Resource Pack - https://github.com/bgeesaman/inspec-k8s
5. Lyft's Cartography - https://github.com/lyft/cartography
6. AWS Config - https://aws.amazon.com/config/
7. GCP Cloud Asset Inventory - https://cloud.google.com/resource-manager/docs/cloud-asset-inventory/overview
8. GCP Continuous Scanning - https://forsetisecurity.org/