



San Francisco | March 4–8 | Moscone Center



SESSION ID: CSV-W02

# Protecting the Cloud with the Power of Cloud

Jay Kelath

Pranav Patel

Product Security @ Dow Jones

# What are we going to talk about...?

## DevSecOps

- How we “SaaS-ified” our on-prem security tools with Docker and DevSecOps
- Scaling security with help of Cloud enables Security automation.

## Self-healing Cloud Misconfigurations

- How native Cloud based tooling helps reduce risk
- How Automation and Self-healing works in our Cloud environment

## Open Source Technologies : [Takeaway] -- start your DevSecOps journey from Day 1

- Dow Jones Hammer: Open source tools enables security
- Project Bravos : Embedding security into Build Pipeline

# Quick Flashback....

- Traditional Security tools
- Manual Reviews
- Lack of visibility and scalability

# Challenges

Legacy

Technology Sprawl

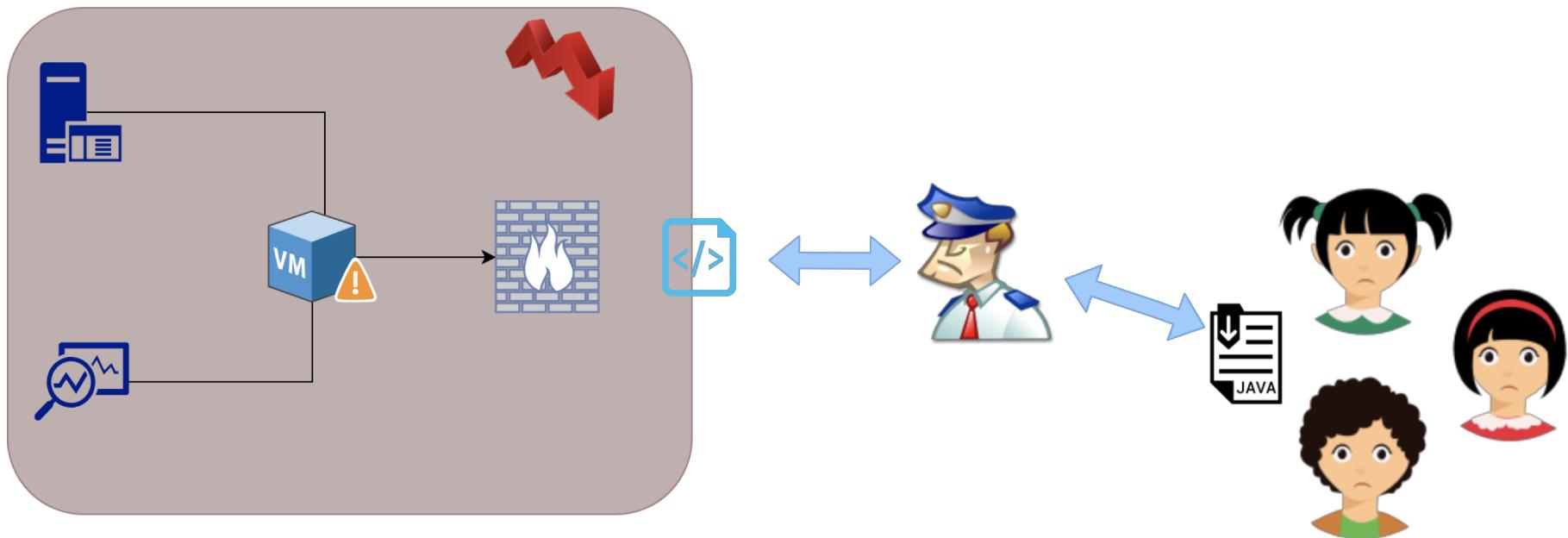
People

Process



DOW JONES

# Traditional Security -- Old School



# RSA® Conference 2019

“Don't Let a Good Crisis Go to Waste”

A complex, abstract graphic in the background, rendered in light blue, consists of numerous small circular nodes connected by thin lines, forming a dense web of curves and loops that suggest a network or a flow of information.

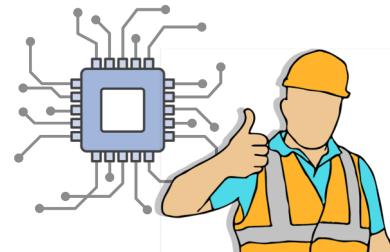
# DevSecOps

Solve a specific problem in an automated manner with well defined People, Process, Technology actions and extensive, actionable reporting

# Setting priorities right...

Risk vs Reward

End Goal: Reduce Risk



# Think about....

## Technology

API Driven

Scalable, Tunable

False Positives

KISS

## Process

Use Existing Process

Feedback Loop

High Quality Report

Optimize

## People

Support Model

Build Trust

Developer support

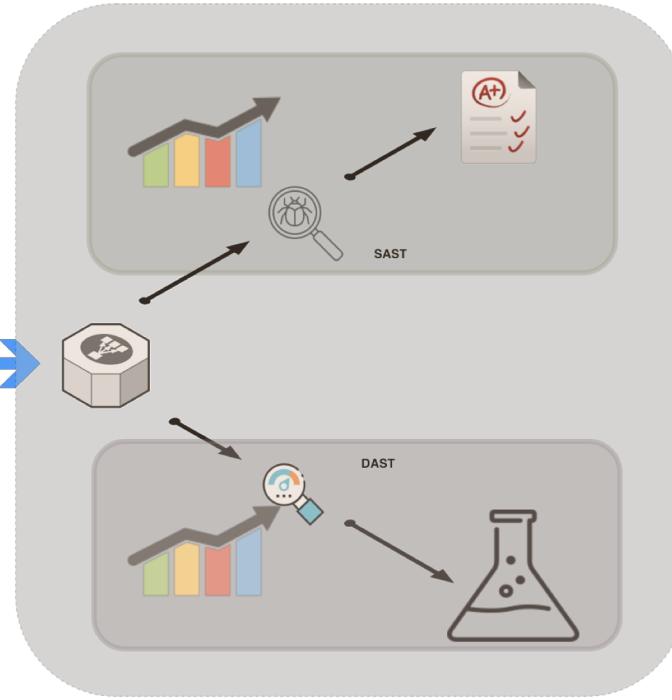
# Security as a Service



Developers

Github

/security\_test

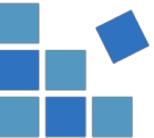


Cloud



DOW JONES

# DevSecOps pipeline



## Plan

- Threat Modelling

## Code

- Linting
- Secure code
- Software composition analysis

## Build

- SAST
- Secrets in Code
- Top 5 criticals

## Test

- Integration Testing
- Abuse case Testing

## Deploy

- Cloud Security
- Runtime security testing
- RASP/IAST/DAST
- Container scanning

# Our Solution

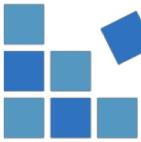
Project Braavos



Plan



Code



Build



Test



Deploy

Dow Jones Hammer

## Dow Jones Hammer

Open Sourced : <https://github.com/dowjones/hammer>



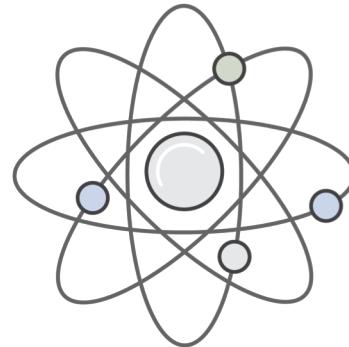
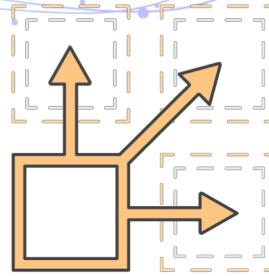
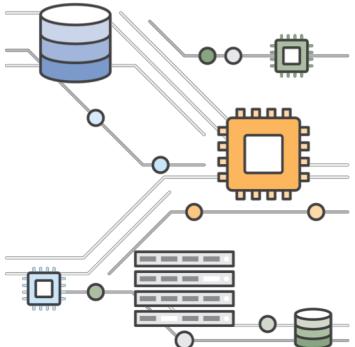
# Hammer... why?

- “Auto-fix” misconfigurations with ability to rollback
- Cloud Infrastructure visibility
- Easily pinpoint MY product’s security issues
- Tailored reporting, save analysis time
- API driven approach



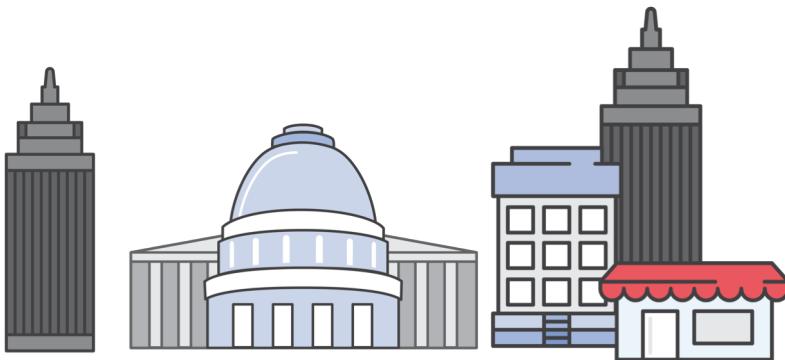
# Our Solution

- Automate
- Scalable
- Self-service
- Auditable

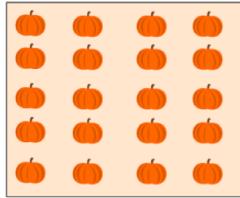
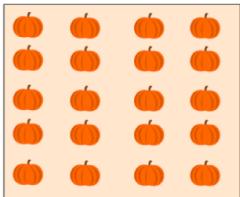
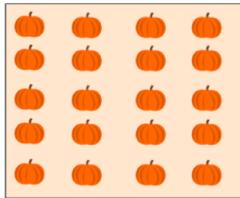
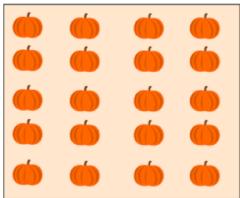
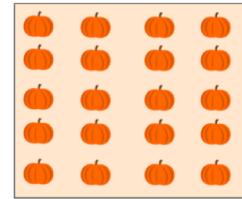
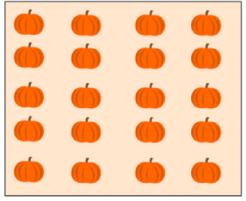
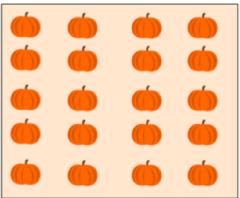
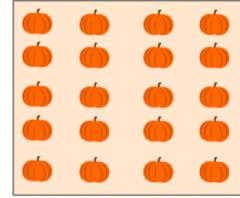
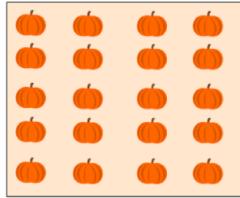
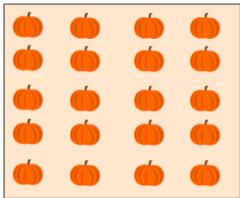
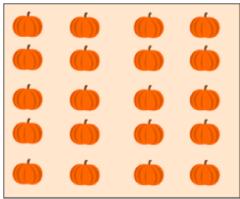


# Consumer

- Multi-account customers
- Decentralized development organizations
- Multiple business units

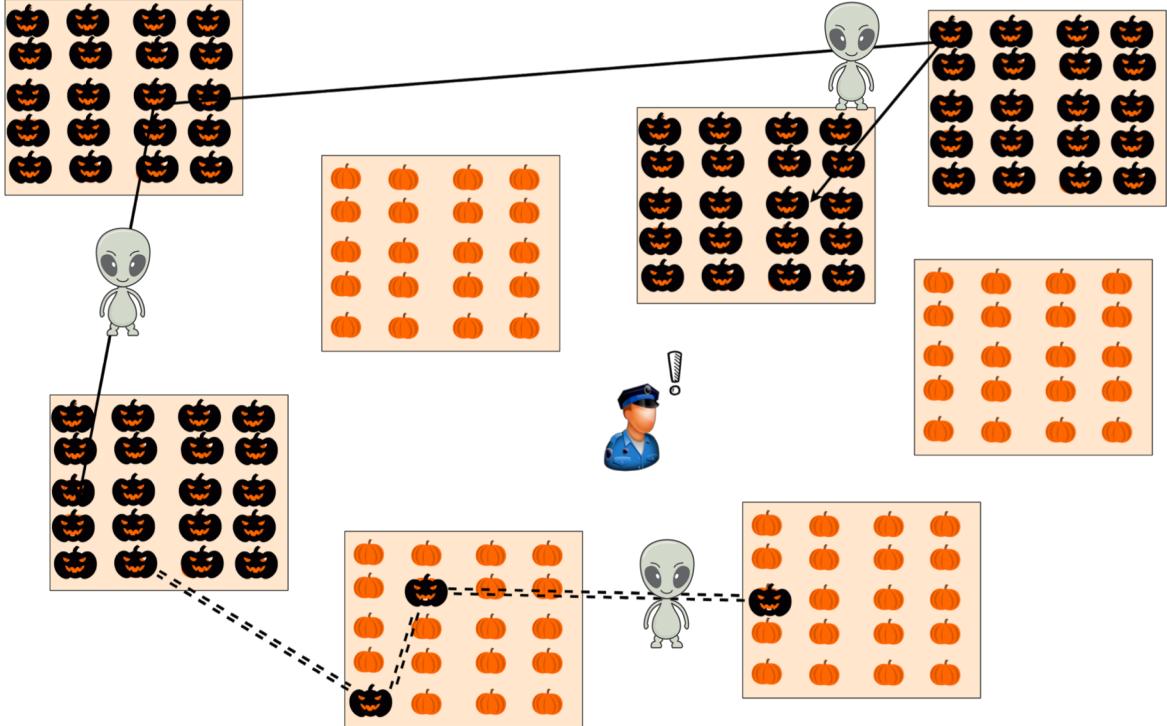


# Multi-account sprawl

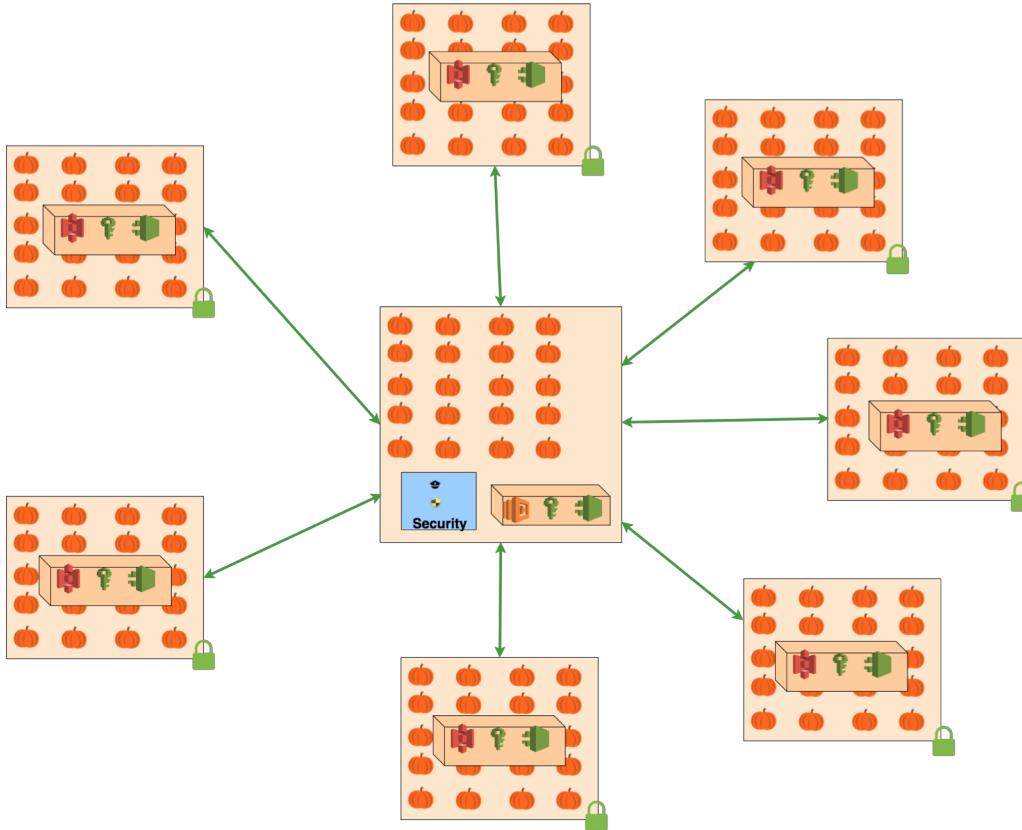


# Multi-account sprawl

- Lack of visibility
- Scalability
- “Someone is looking into it...”



# Multi-account growth : Ideal State



# Hammer: What Does it Solve?



Public Instances with Admin IAM Policies  
Exposed EC2 Instances  
Docker on EC2  
ECS



Public S3 buckets  
Unencrypted S3 buckets  
Public S3 bucket Policy



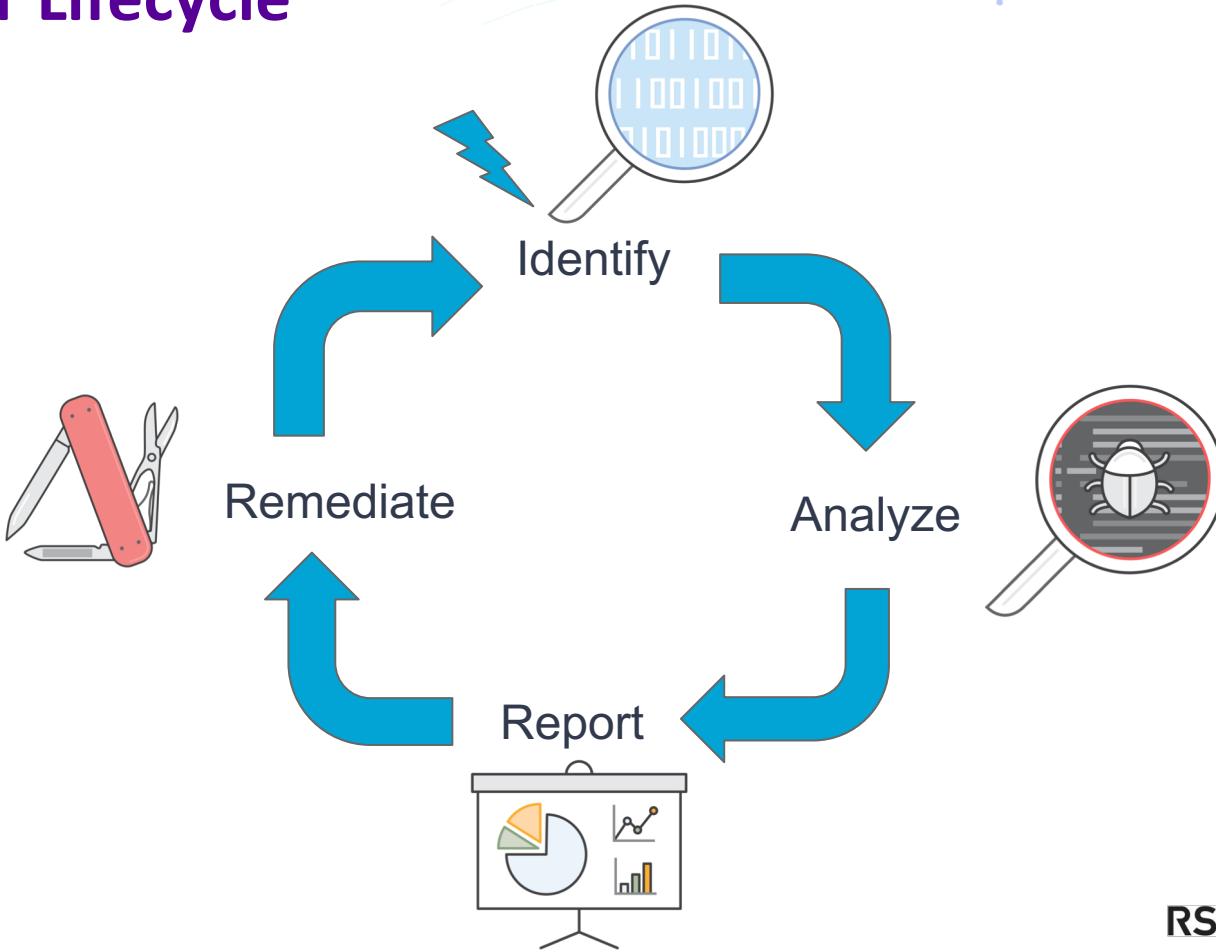
Exposed RDS instances  
Unencrypted RDS  
Public RDS Snapshots



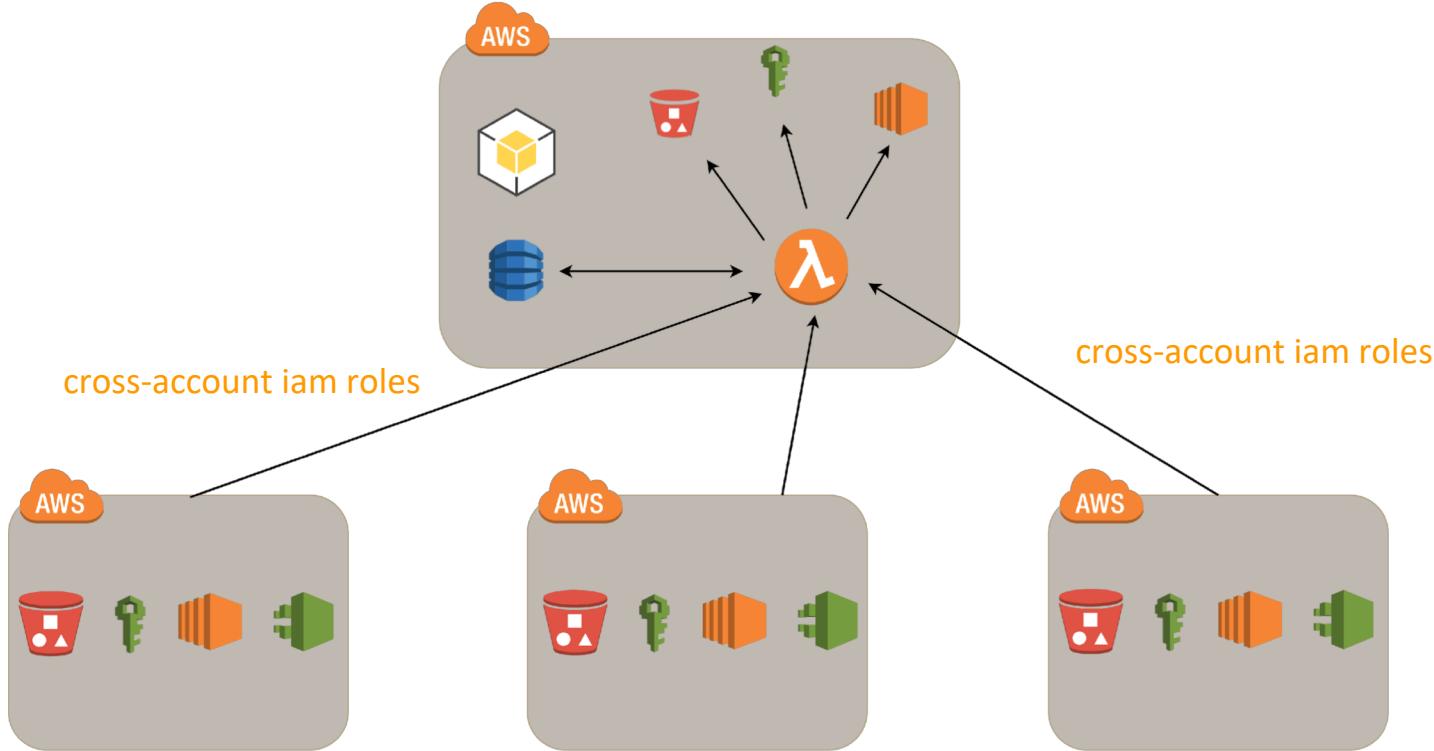
Unused IAM Keys  
Stale IAM Keys (not rotated)



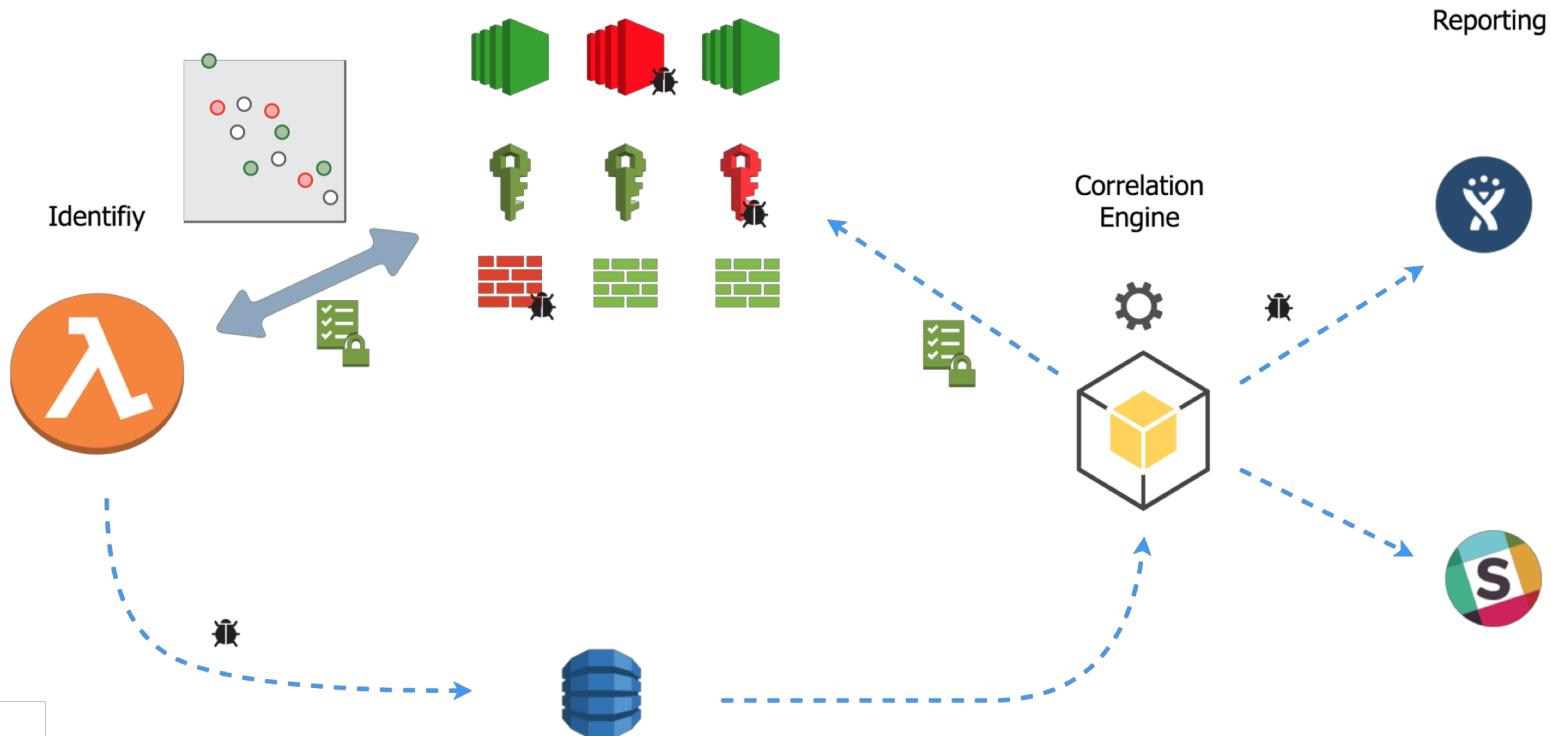
# Hammer Lifecycle



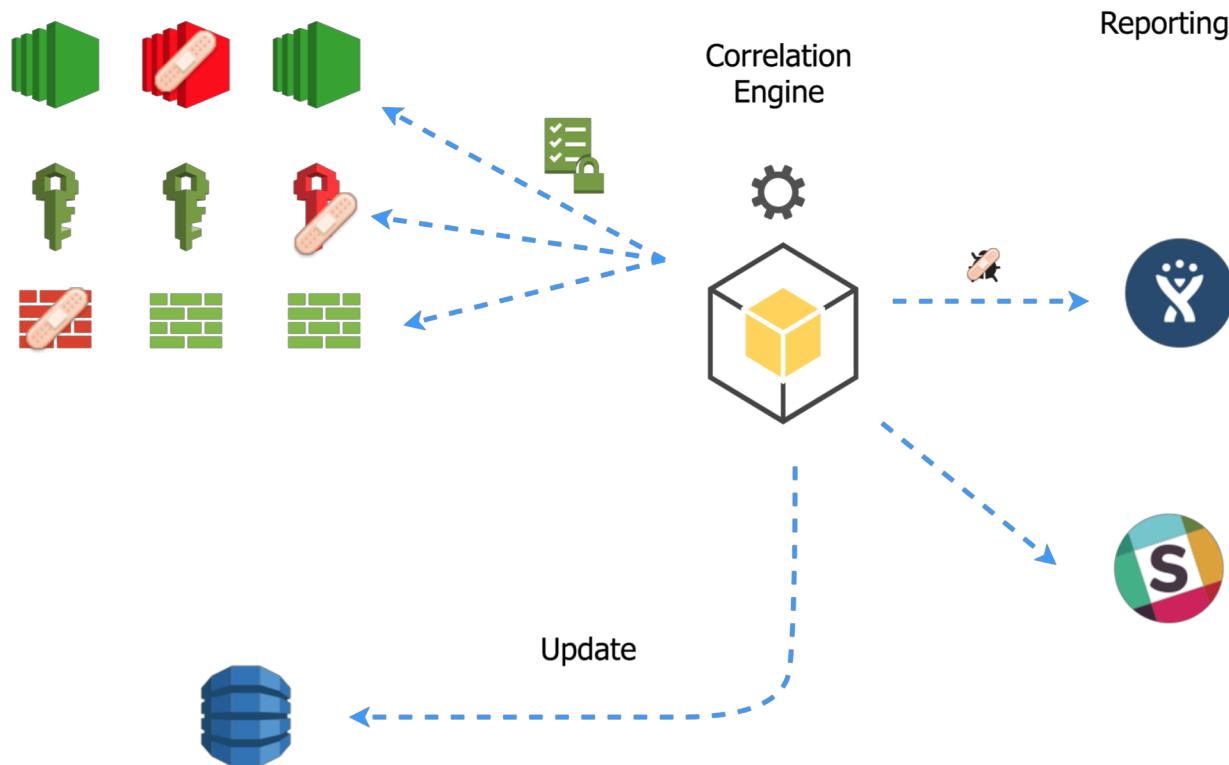
# Architecture



# Hammer - how does it work?



# Hammer - /auto-fix



# Hammer : Service Matrix

Services	Identification	Reporting	Remediation
[IAM]: Inactive IAM keys, IAM key rotations)	✓	✓	✓
[Data]: Public S3 (acl, bucket policy) Public SQS Public Snapshots(EBS, RDS, AMI) Unencrypted EBS, S3, RDS, SQS	✓	✓	✓
[Compute]: Over-exposed instances, Over-permissive IAM	✓	✓	✓
[Databases]: Over-exposed RDS, Overexposed databases (on EC2)	✓	✓	✓



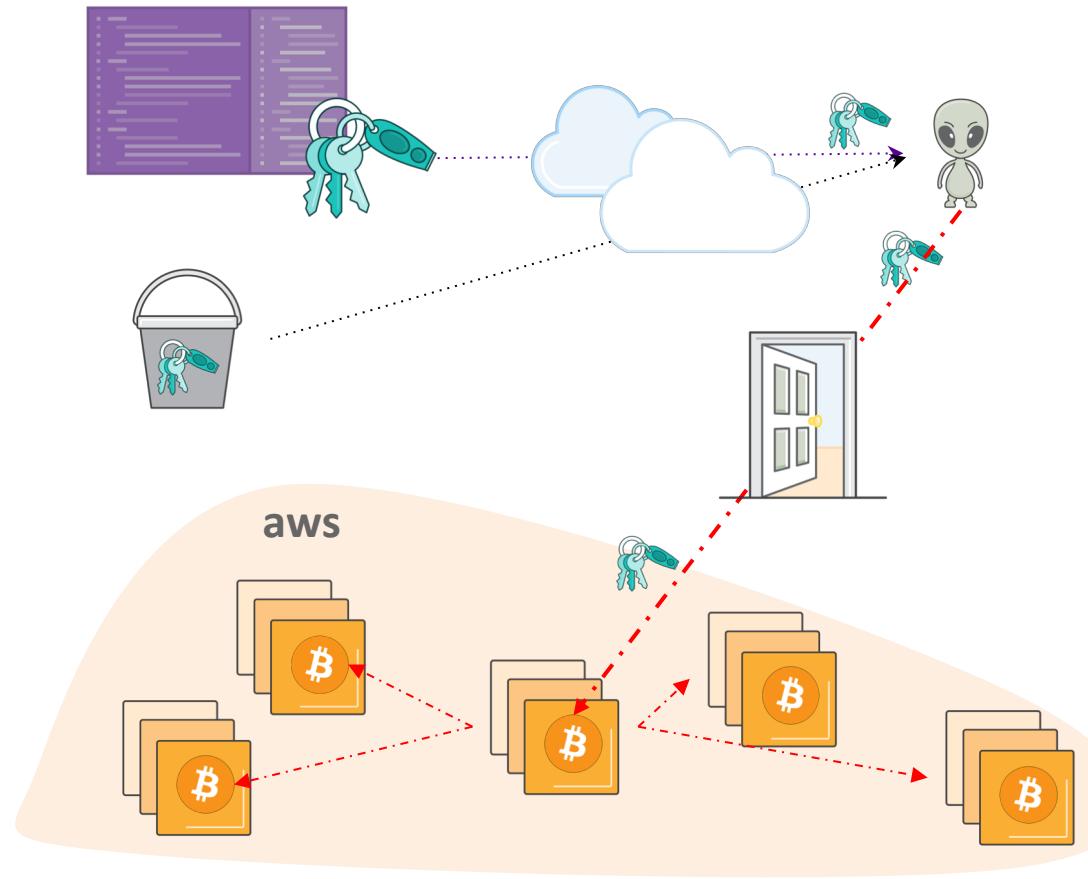
DOW JONES

# Roadmap

- Hammer API
- New Security Feature
  - Redshift
  - Containers (ECS, EKS, Fargate)
  - CloudFront ...
- Self Service Hammer bot (on-Demand scans)

# Case Study 1: Protection from bitcoin miners.....

- Stale & Exposed keys in Code
- Stale & Exposed keys in public bucket
- Ex-employees with Keys



# Case Study 1 : Example

 S [REDACTED] IAM access key 'AKIAJ [REDACTED]' for 'd [REDACTED]n' has not been used for 180 days in ' [REDACTED]' account

[Edit](#) [Comment](#) [Assign](#) [More ▾](#) [Reopen Issue](#) [In Validation](#)

**Details**

Type:	<input checked="" type="radio"/> Vulnerability	Status:	<b>CLOSED</b> (View Workflow)
Priority:	<input checked="" type="radio"/> Major	Resolution:	Fixed
Affects Version/s:	None	Fix Version/s:	None
Labels:	<a href="#">inactive-iam-keys</a>		
Risk Rating:	High		

**Description**

IAM access key has not been used for 180 days.

Risk: Low

Account Name: [REDACTED]  
 Account ID: [REDACTED]  
 User Name: [REDACTED]-production  
 Key ID: AKIAJ [REDACTED]  
 Key created: 2018-08-31 13:06  
 Key last used: 2018-08-31 13:06

**Auto-Remediation Date:** 2019-02-27

**Recommendation:** Deactivate specified inactive user access key. For any other exceptions, please follow the [whitelisting procedure](#) and provide a strong business reasoning.

**People**

Assignee: [REDACTED]  
[Assign to me](#)

Reporter: [REDACTED] (InfoSec)

Votes: 0 [Vote for this issue](#)

Watchers: 1 [Start watching this issue](#)

**Dates**

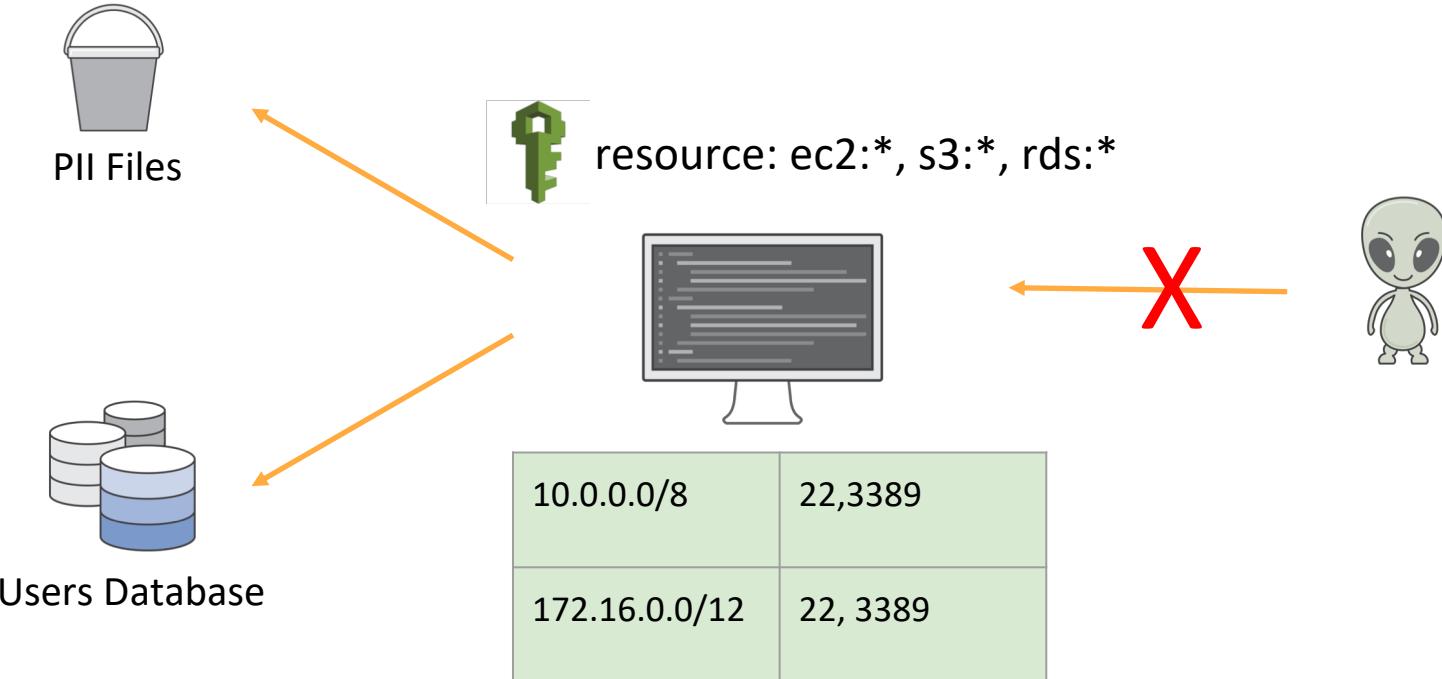
Created: 6 days ago  
 Updated: 5 days ago  
 Resolved: 5 days ago

**Collaborators**

**Agile**

[View on Board](#)

# Case Study 2: Over Exposed Instances



DOW JONES

# Case Study 2: Over Exposed Instances

Description							
Security group has EC2 instances in private subnets and source IP address with a /0 suffix for following ports:							
Click to edit	To Port	Protocol	CIDR				
<table border="1"><tr><td>22</td><td>22</td><td>tcp</td><td>0.0.0.0/0</td></tr></table>				22	22	tcp	0.0.0.0/0
22	22	tcp	0.0.0.0/0				
<b>Threat:</b> Open access within the network not only provides unrestricted access to other servers but increases opportunities for malicious activity (hacking, denial-of-service attacks, loss of data) if attacker gains access to the services within the network, thus providing lateral movement.							
<b>Risk:</b> High							
Account Name: [REDACTED]							
Account ID: [REDACTED]							
Region: [REDACTED]							

## Instance Role Unsafe Policies:

Instance Id	Role Name	Policy Name	Unsafe actions
i-[REDACTED]	c-[REDACTED]	i-[REDACTED]	autoscaling:*
i-[REDACTED]	c-[REDACTED]	i-[REDACTED]	cloudformation:Describe* cloudformation:GetTemplate cloudformation>ListStack*
i-[REDACTED]	c-[REDACTED]f	i-[REDACTED]	cloudwatch:*
i-[REDACTED]	d-[REDACTED]	i-[REDACTED]	ec2:*

# Case Study 2: Over Exposed Instances

Security group has no EC2 instances attached and allows access from some definite public ip addresses or networks for following ports:

From Port	To Port	Protocol	CIDR	Registrant
22	22	tcp	172.0.0.0/8	AT&T Corp. (AC-3280)

**Threat:** An unused SG can be leveraged to gain control/access within the network if attached to any exposed instance. This unrestricted access increases opportunities for malicious activity (hacking, denial-of-service attacks, loss of data).

**Risk:** High

**Account Name:** [REDACTED]s

**Account ID:** 2[REDACTED]

**SG Name:** r[REDACTED]sample\_ec2

**SG ID:** sg-[REDACTED]

**Region:** us-west-2

**Tags:**

Key	Value
owner	[REDACTED]@dowjones.com
servicename	[REDACTED]/sample
environment	nonprod
bu	[REDACTED]
Name	r[REDACTED]sample_ec2

**Recommendation:** Allow access only for a minimum set of required ip addresses/ranges from [RFC1918](#). For any other exceptions, please follow [whitelisting procedure](#) and provide a strong business reasoning. Be sure to delete overly permissive rules after creating rules that are more restrictive.

# Lessons Learnt

Scales over 100+ AWS Accounts

Integrates to Dev ecosystem

Easy to plug and play

Tonnes of insecurities fixed

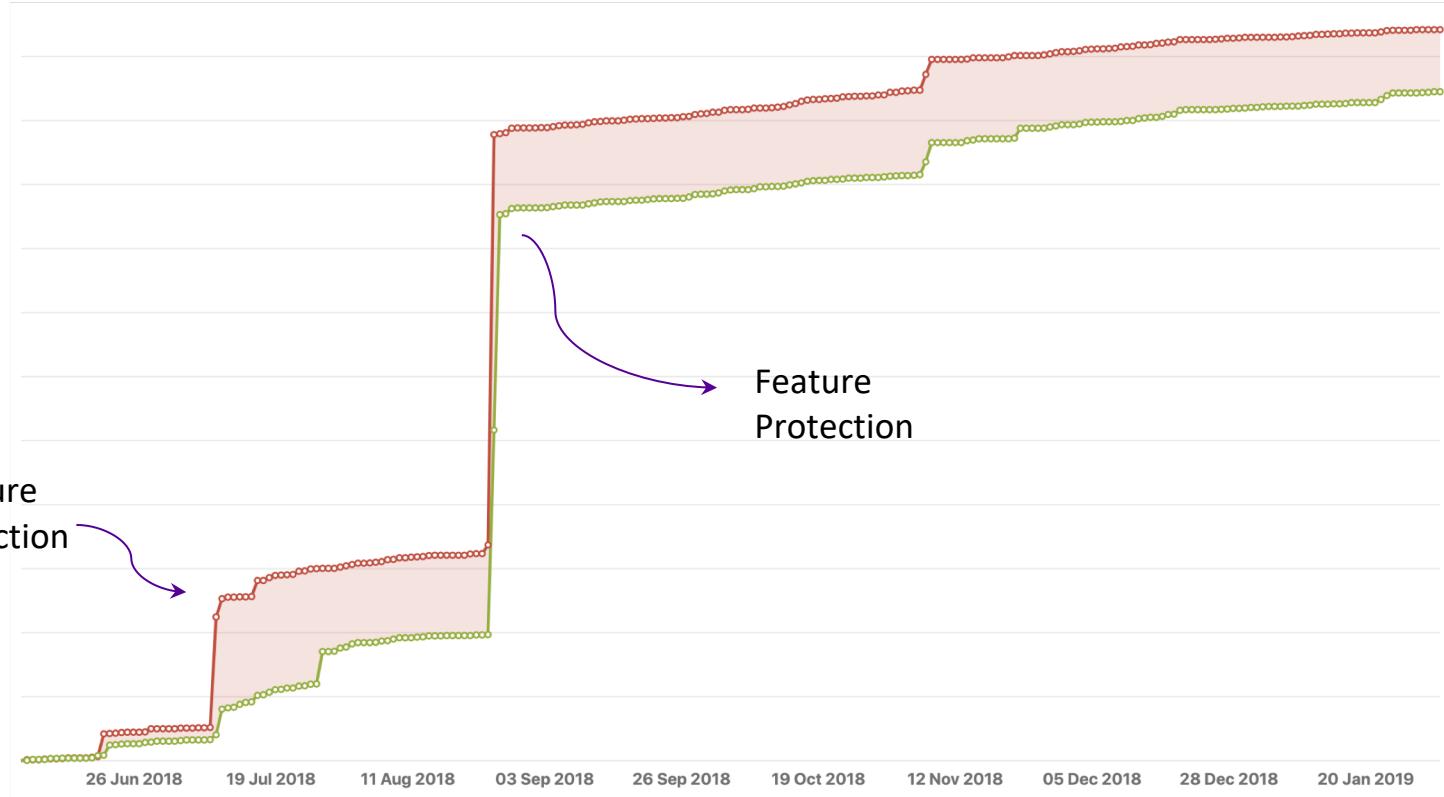
Low to minimal Impact

AWS Lambda execution limits

Server elements

Limited security checks

# Key Trends....



DOW JONES

# Implement Cloud Security : Day 1

Github : <https://github.com/dowjones/hammer>

Hammer Case Study

<https://medium.com/dowjones/introducing-dow-jones-hammer-f0121815189a>

Behind the Scenes : (Architecture design)

<https://medium.com/dowjones/behind-the-scenes-of-dow-jones-hammer-38579391f1a0>

# RSA® Conference 2019

## Scaling security and embedding into pipeline

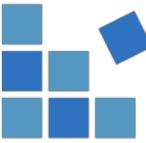
# Project Braavos



Plan



Code



Build



Test



Deploy

Dow Jones Hammer

# Project Bravos : CICD Integration

## Sensitive information in code Continuous Security

Static code analysis

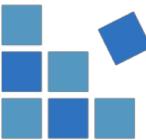
Risky function calls

Open Source Vulnerability Checks

Security test cases

## Dynamic Security Tests

# DevSecOps - reference tools



## Plan

- Threat modelling

## Code

- Cx
- Fortify
- Coverity
- Veracode

## Build

- Cx
- Coverity
- Veracode
- Git Secrets
- Snyk/  
BlackDuck

## Test

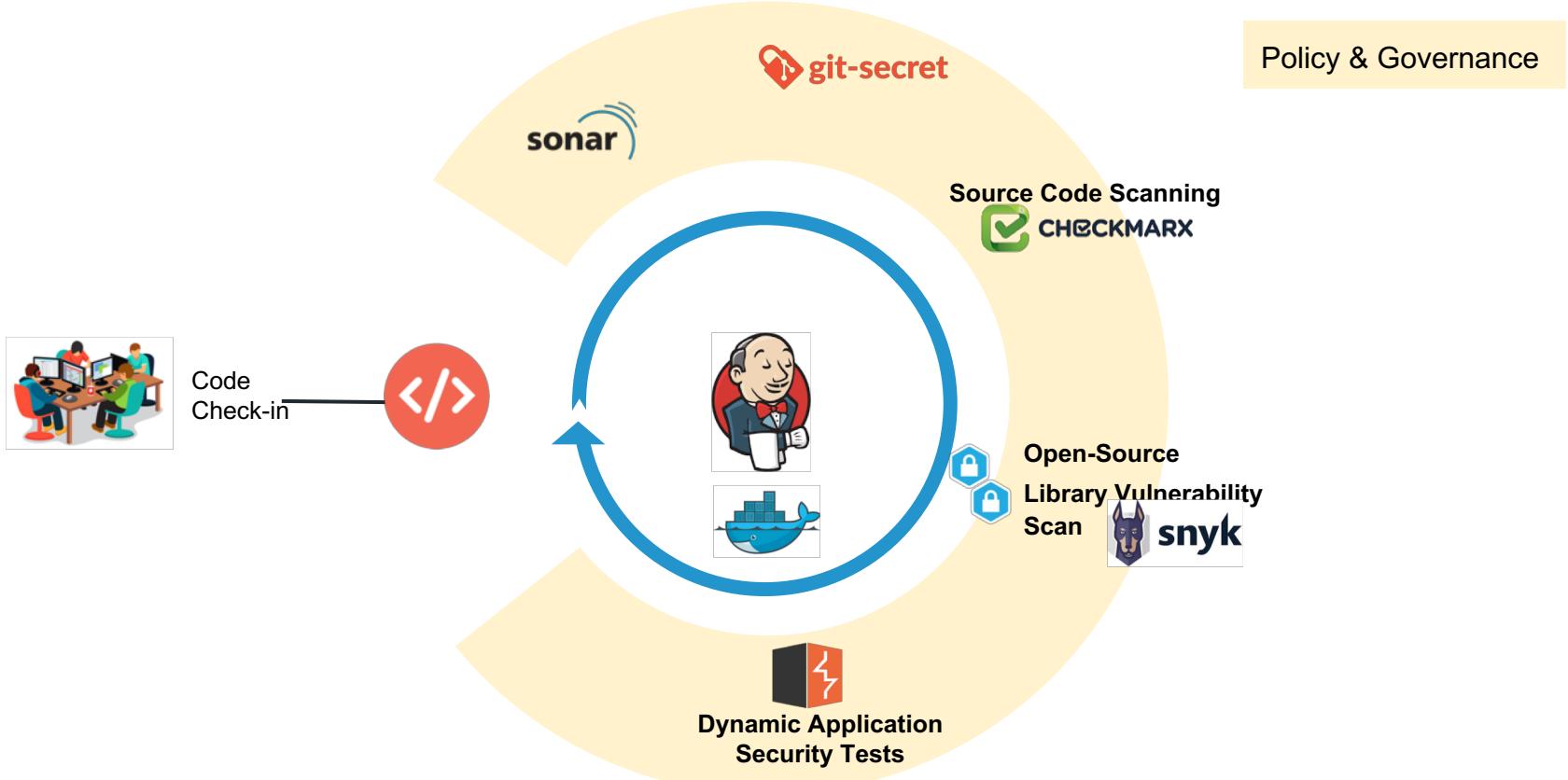
- ZAP
- Burp
- Enterprise
- Detectify

## Deploy

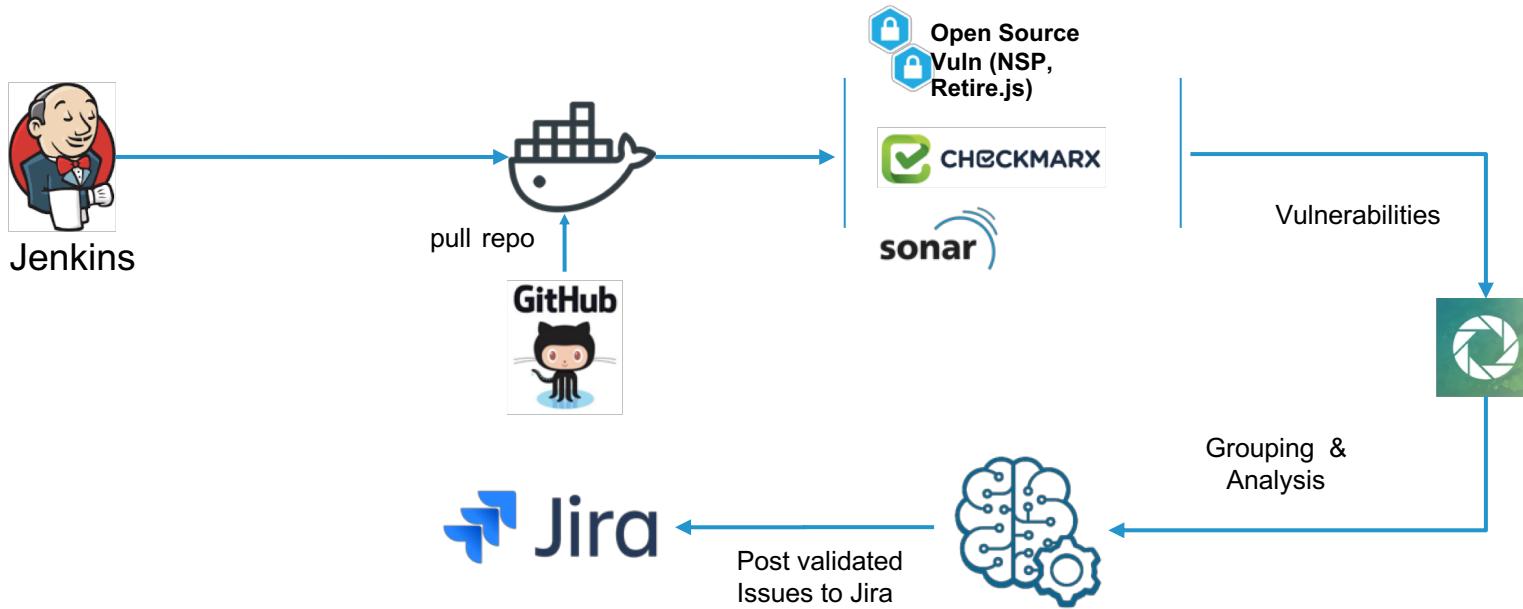
- Dow Jones
- Hammer
- Burp

Ref: [https://www.owasp.org/index.php/OWASP\\_AppSec\\_Pipeline#tab=Pipeline\\_Tools](https://www.owasp.org/index.php/OWASP_AppSec_Pipeline#tab=Pipeline_Tools)

# Our Solution : project bravos

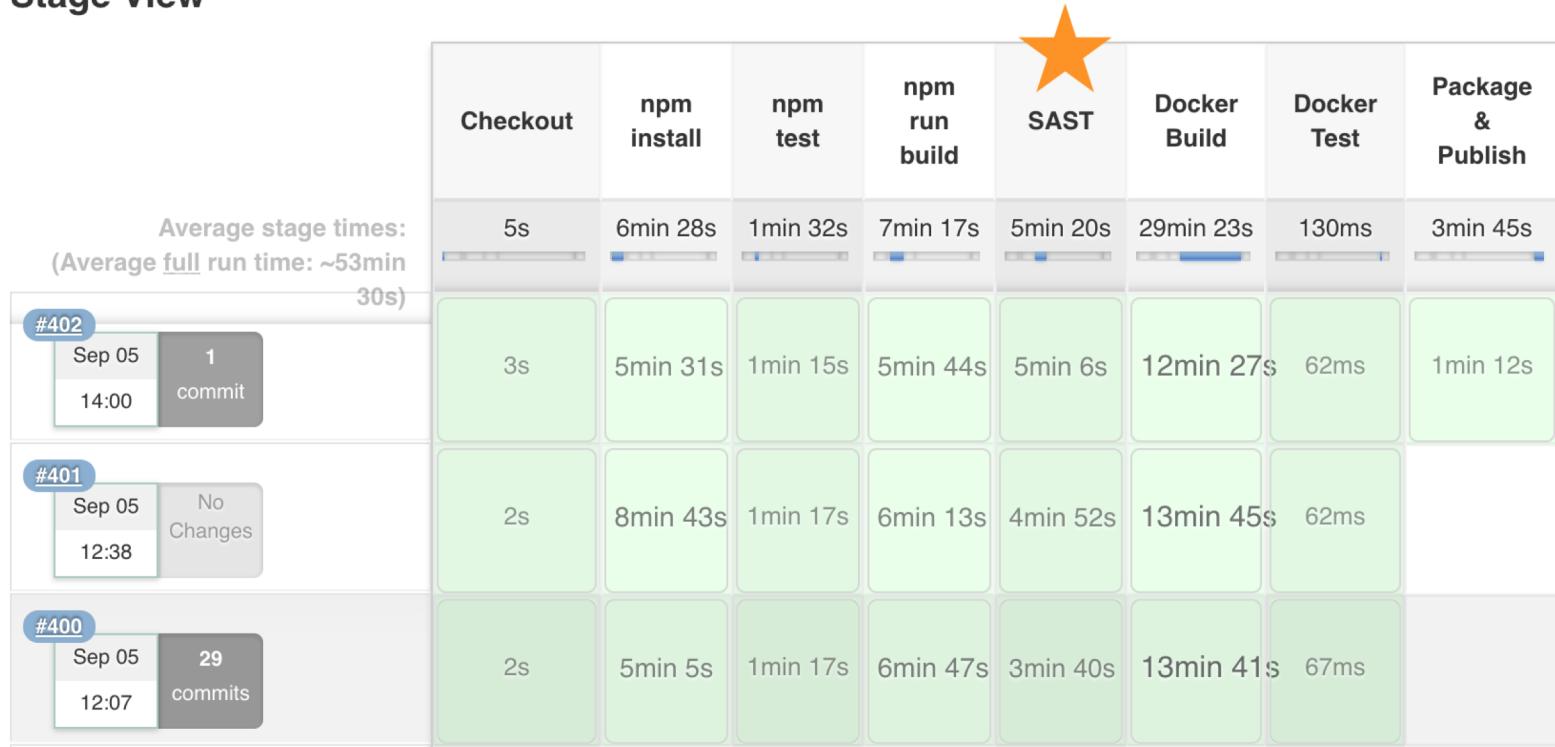


# /continuous\_security : sast



# /continuous\_security: Jenkins Pipeline

## Stage View



# /continuous\_security: sast



checkout project	Check Checkmarx	Check NPM	Check Secrets	Check Retire.Js	Check SonarQube	publish reports
6min 3s	2min 34s	23s	2s	56s	3s	12s
						
6min 44s	1min 52s	21s	2s	56s	28ms	13s

# Lessons Learnt

Easy to Integrate

Tune! Tune! Tune!

Tech needs to match the Tool

Incremental Scans FTW

More accessible to Devs

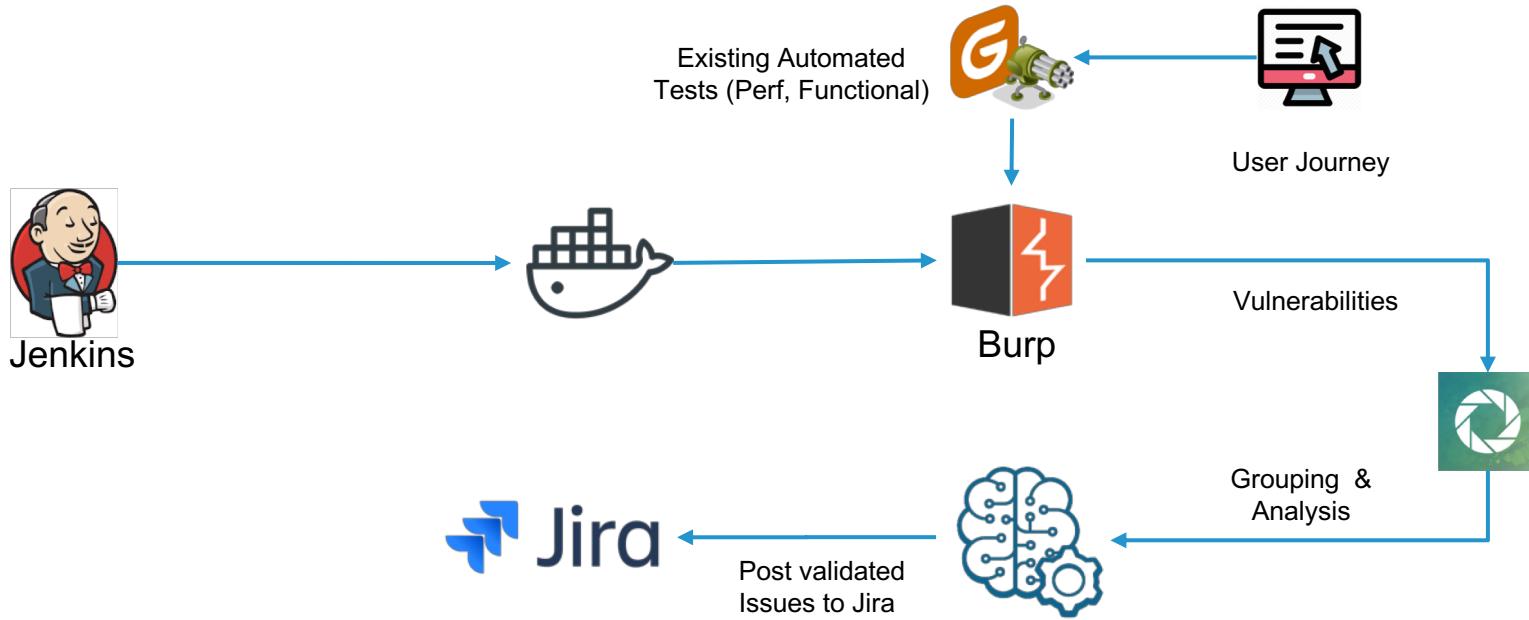
False Positives

Execution Time

Limited Tech Coverage

“Glue code” for Integration

# Dynamic Scanning in Pipeline



# Lessons Learnt

Existing Test Cases FTW

Focus on your Top 5

Deploy a “Security Test Stack”

Poor Integration Options

Execution Time

More hand-holding needed

# Breaking the build : Myth?

Reporting! Reporting! Reporting!

Container Security

Abuse Test Cases

XSS

Injection Issues

Open Source  
Components

Secrets in Code

Crawl

Walk

Run



DOW JONES

RSA® Conference 2019

# Case Study : Example 1

T	Summary	Labels	P	Status	Resolution	Created	Reporter	Due
0	Secrets exposed in /tmp/code/app/app.settings.js	GitSecrets Severity_Critical sast	✗	CLOSED	Fixed	09/24/2018	product_security	09/26/2018
0	Secrets exposed in ajaxfunctions.js	GitSecrets sast	✗	CLOSED	Won't Fix	11/06/2018	product_security	11/08/2018
0	Secrets exposed in visible.min.js	GitSecrets sast	✗	CLOSED	Not An Issue	10/10/2018	product_security	10/12/2018
0	Secrets exposed in productsscript.js	GitSecrets sast	✗	CLOSED	Not An Issue	10/10/2018	product_security	10/12/2018
0	Secrets exposed in functions.min.js	GitSecrets sast	✗	CLOSED	Not An Issue	10/10/2018	product_security	10/12/2018
0	Secrets exposed in /app/app.settings.js	GitSecrets sast	✗	RESOLVED	Done	08/17/2018	product_security	08/21/2018
0	Secrets exposed in /app/app.settings.js	GitSecrets sast	✗	RESOLVED	Done	08/17/2018	product_security	08/21/2018
0	Secrets exposed in /app/app.settings.js	GitSecrets sast	✗	RESOLVED	Done	08/20/2018	product_security	08/22/2018
0	Secrets exposed in /app/app.settings.js	GitSecrets sast	✗	RESOLVED	Done	08/20/2018	product_security	08/22/2018
0	Secrets exposed in /app/app.settings.js	GitSecrets sast	✗	RESOLVED	Fixed	08/20/2018	product_security	08/22/2018
0	Secrets exposed in app.settings.js	Braavos GitSecrets sast	✗	CLOSED	Fixed	09/10/2018	product_security	09/12/2018
0	Secrets exposed in prettyify.js	Braavos GitSecrets sast	✗	CLOSED	Fixed	09/14/2018	product_security	09/18/2018

# Resolution : Effective Reporting

**Description**

**Recommendations:**

Please use secrets management solution for storing sensitive data:

1. HashiCorp Vault
2. Parameter Store
3. KMS
4. AWS Secrets Manager

**Description:**

Sensitive data like passwords, tokens, secrets and private keys should not be committed in repository.  
Keep them secret and not commit in repository.

**Secrets exposed in AjaxFunctions.js**

**Issue Severity:** Critical  
**Overview:** Exposed secrets:  
[line 235] f\*\*\*\*\*  
[line 350] f\*\*\*\*\*  
**References:** , [REDACTED]:Functions.js

**Secrets exposed in AjaxFunctions.js**

**Issue Severity:** Critical  
**Overview:** Exposed secrets:  
[line 235] f\*\*\*\*\*  
[line 350] f\*\*\*\*\*  
**References:** , [REDACTED]:Functions.js

# Continuous Security vs Penetration Testing

Pick your Top 5 or 2

Tune ! Tune ! Tune !

Focus on Low hanging fruits

Only confirmed critical

Scan time < 10% of total build  
time

OWASP Top 10

Out of band Testing

Business logic flaws

# Thank you

## We're Hiring!

<https://dowjones.jobs>

Jay Kelath



<https://github.com/kelath>

Pranav Patel

<https://github.com/pranav1688>



@kelath

@pranav16