RSAConference2019
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

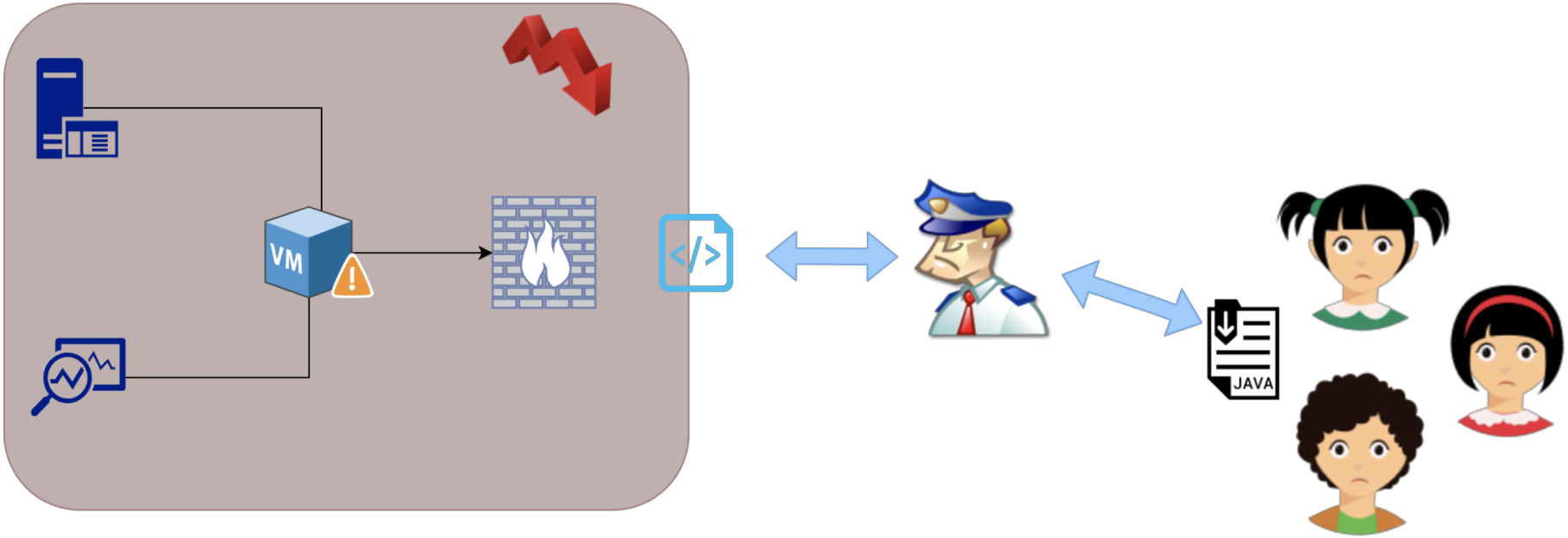
Challenges

Legacy

- Technology Sprawl
- People
- Process

DOW JONES

Traditional Security -- Old School



DOW JONES

RSAConference2019

“Don't Let a Good Crisis Go to Waste”

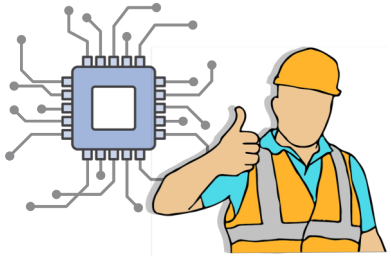
DevSecOps

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

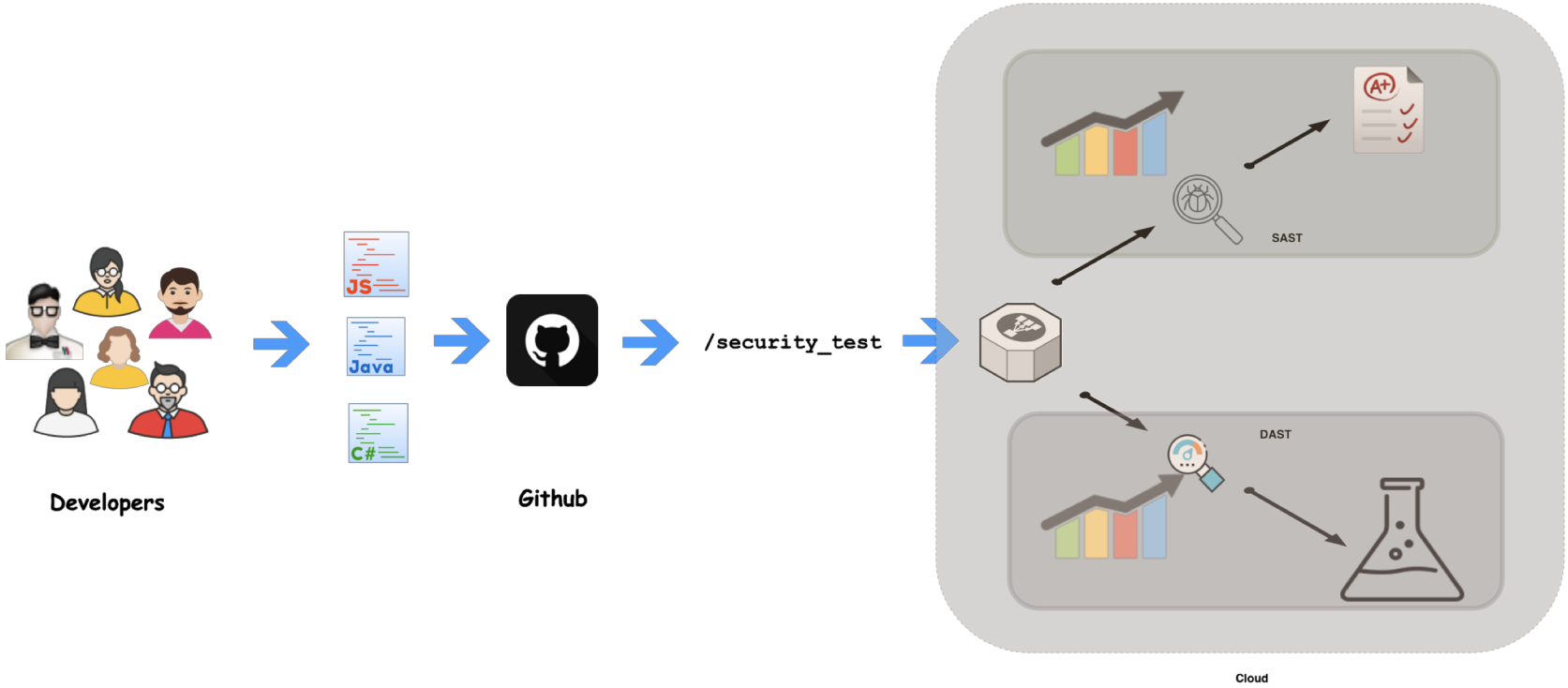
Process

- Use Existing Process
- Feedback Loop
- High Quality Report
- Optimize

People

- Support Model
- Build Trust
- Developer support

Security as a Service



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

Dow Jones Hammer

Plan

Code

Build

Test

Deploy

RSAConference2019

Dow Jones Hammer

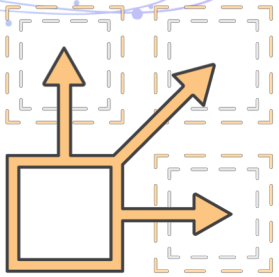
Open Sourced : https://github.com/dowjones/hammer

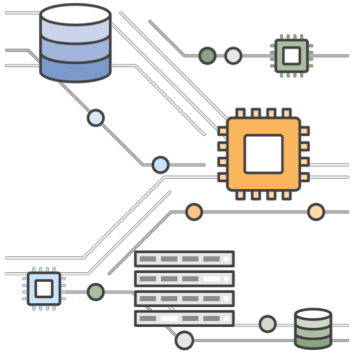
Hammer… why?

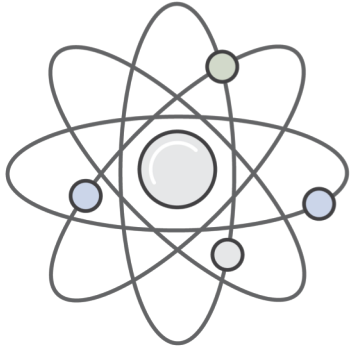
- “Auto-fix” misconfigurations with ability to rollback
- Cloud Infrastructure visibility
- Easily pinpoint MY product’s security issues
- Tailored reporting, save analysis time
- API driven approach

Our Solution

- Automate
- Scalable
- Self-service
- Auditable

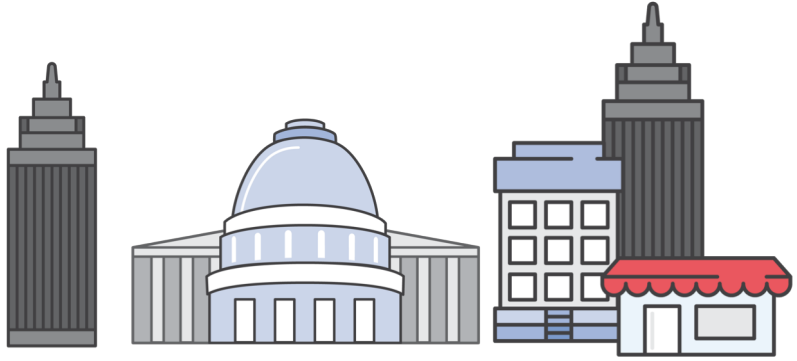




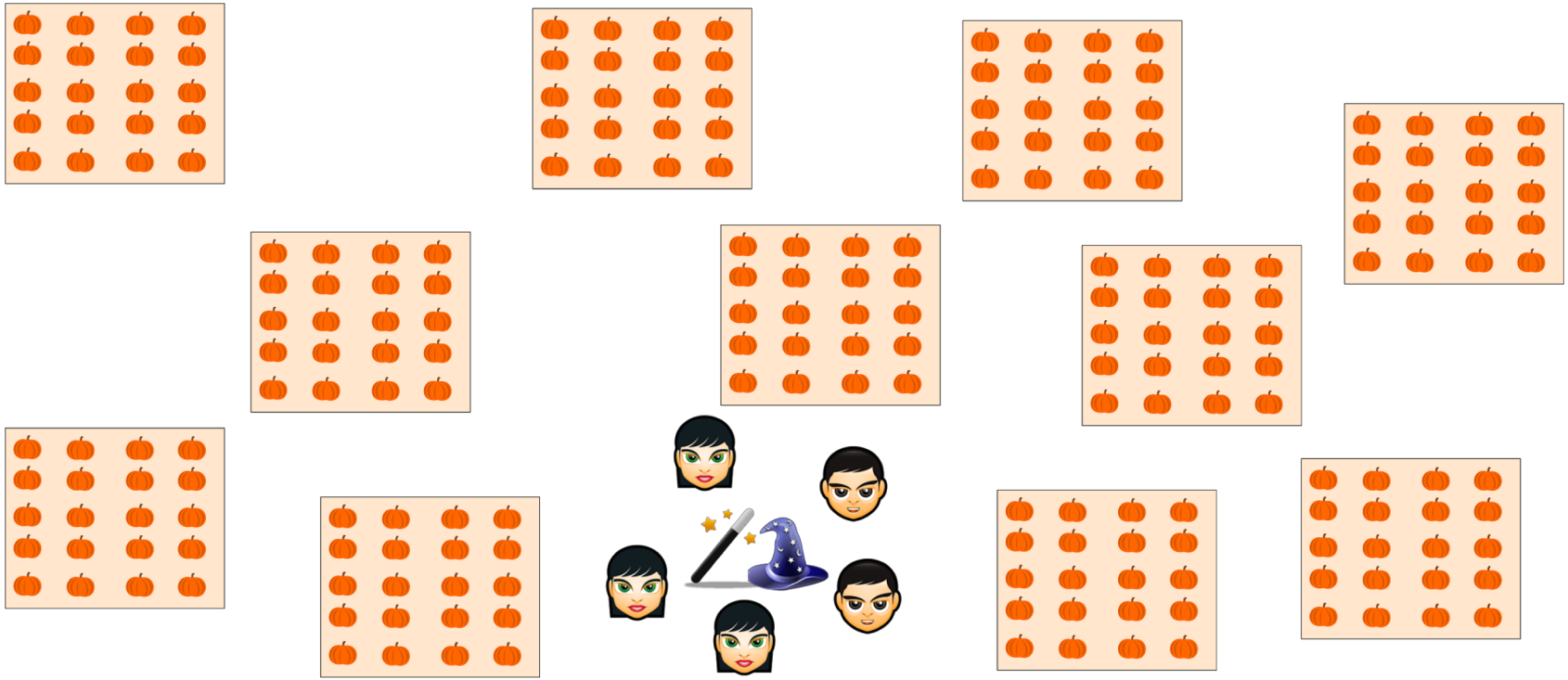


Consumer

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

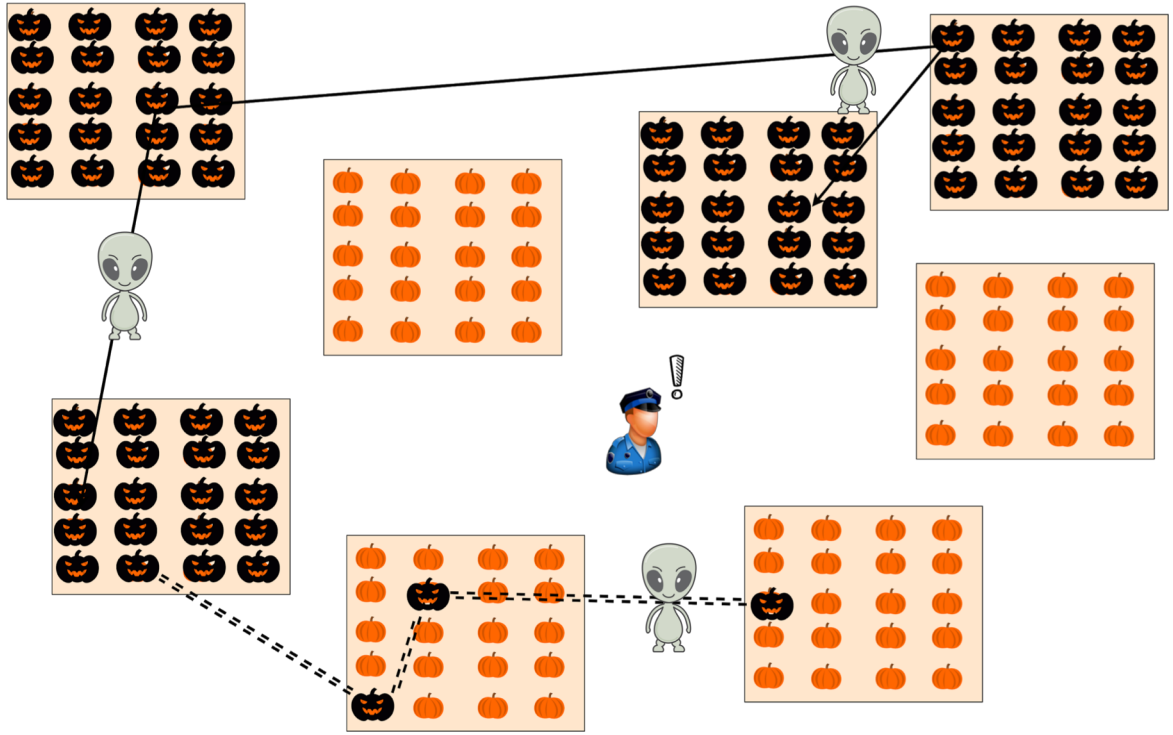


Multi-account sprawl

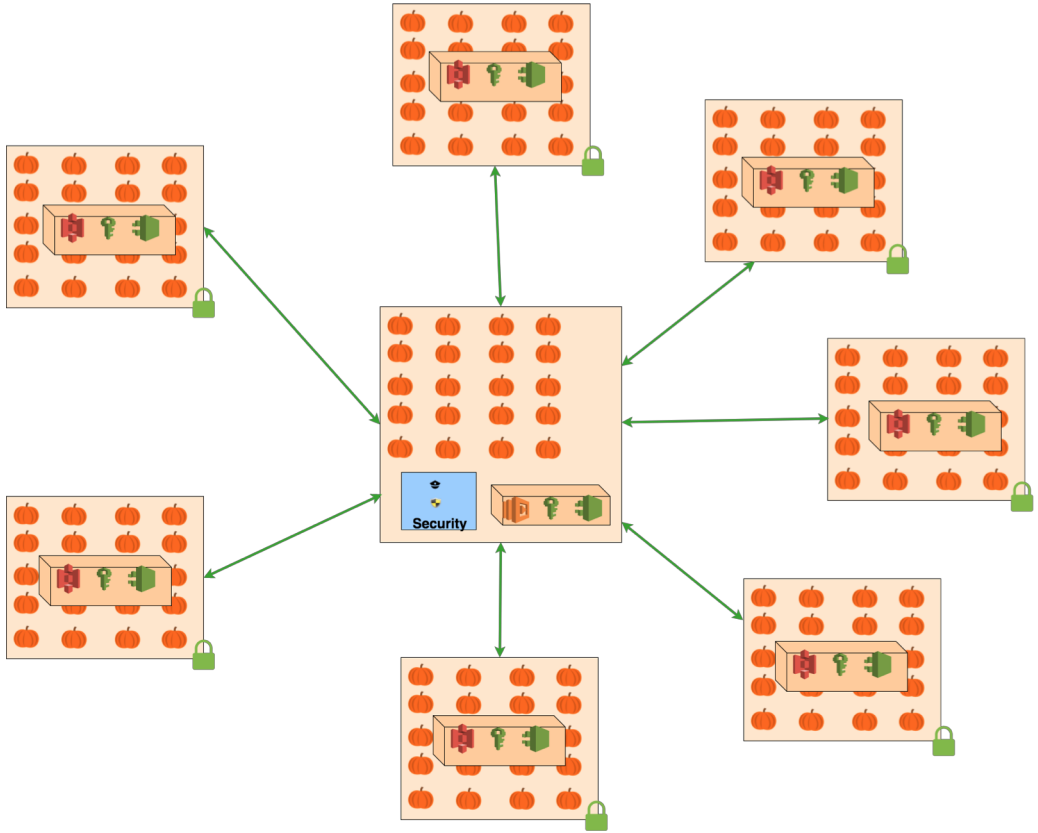


Multi-account sprawl

- 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

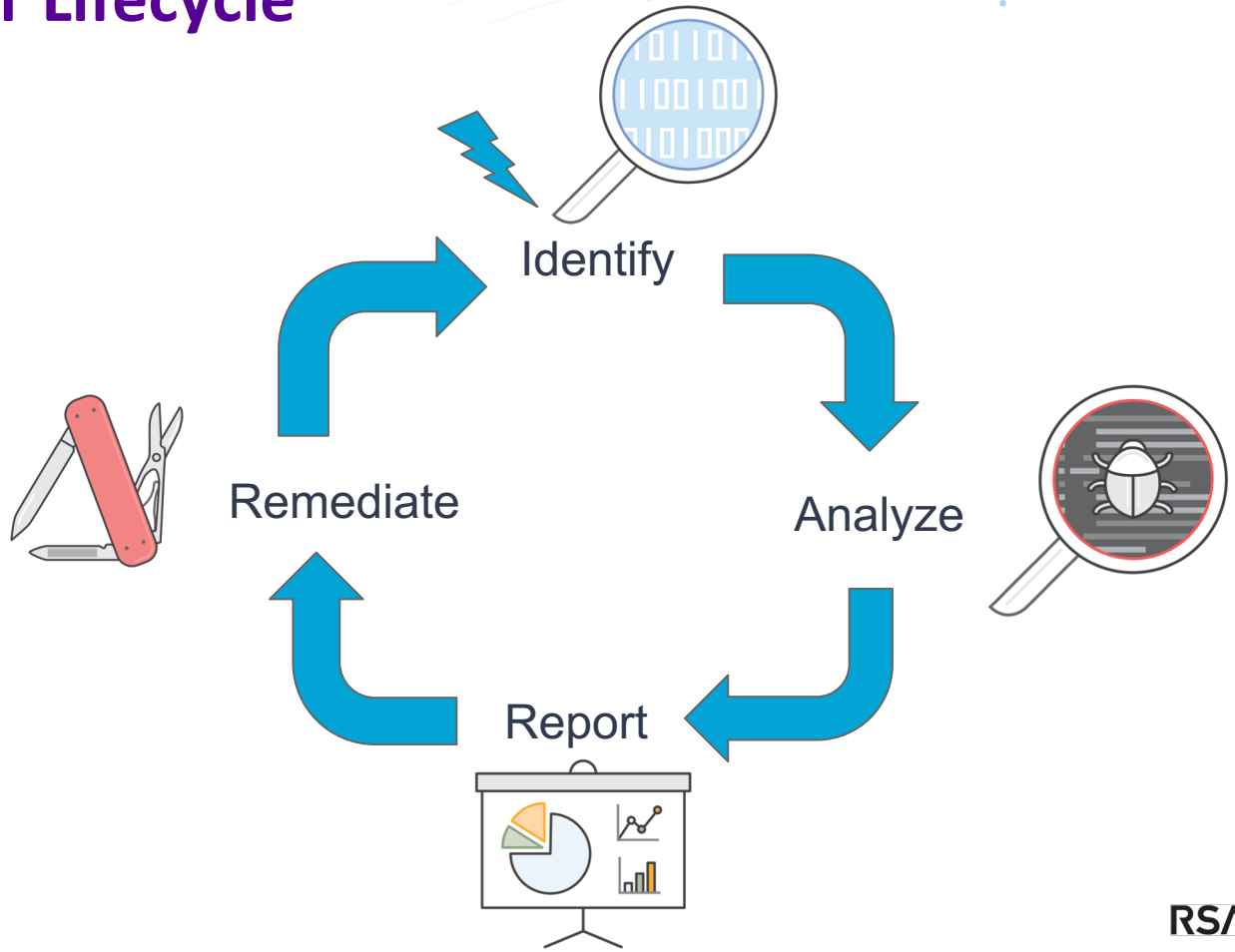
Public S3 buckets
Unencrypted S3 buckets
Public S3 bucket Policy

Exposed RDS instances
Unencrypted RDS
Public RDS Snapshots

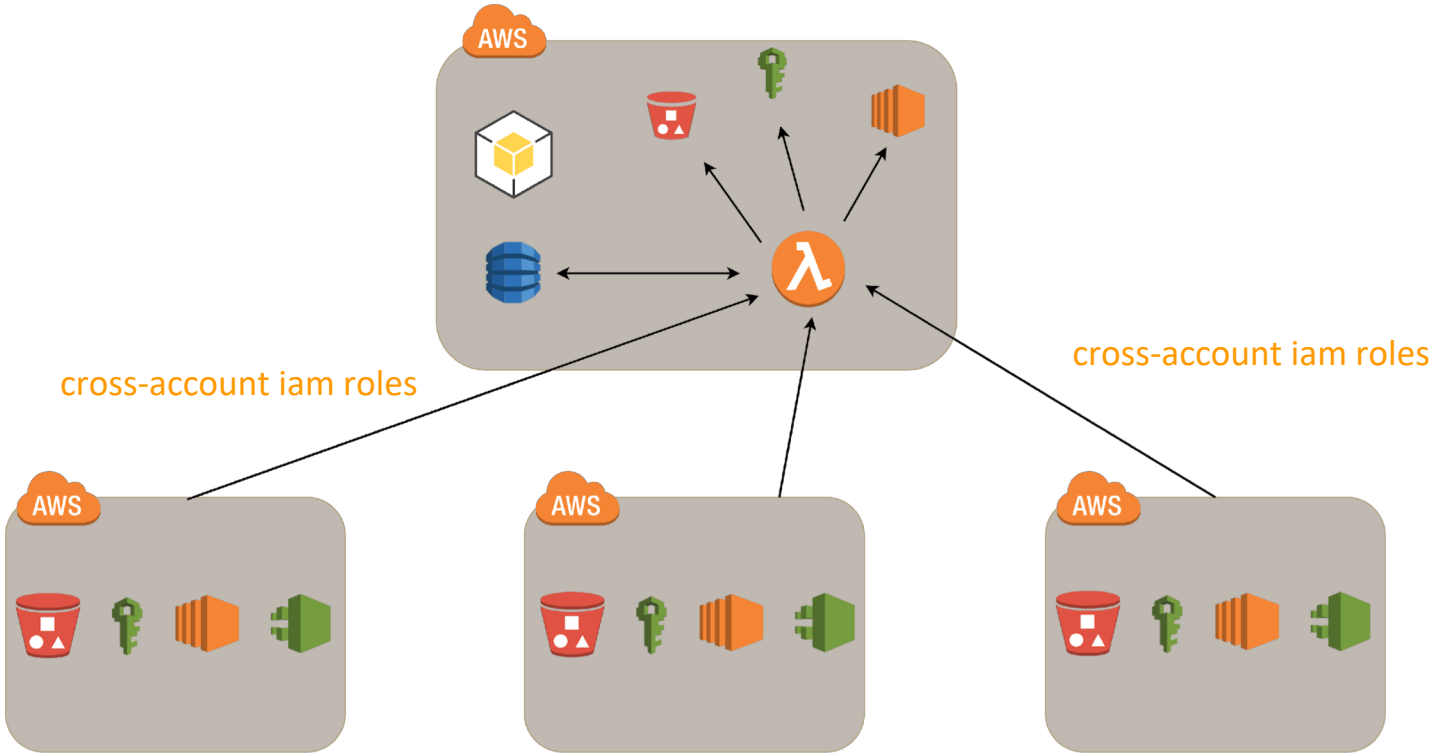
Unused IAM Keys
Stale IAM Keys (not rotated)

DOW JONES

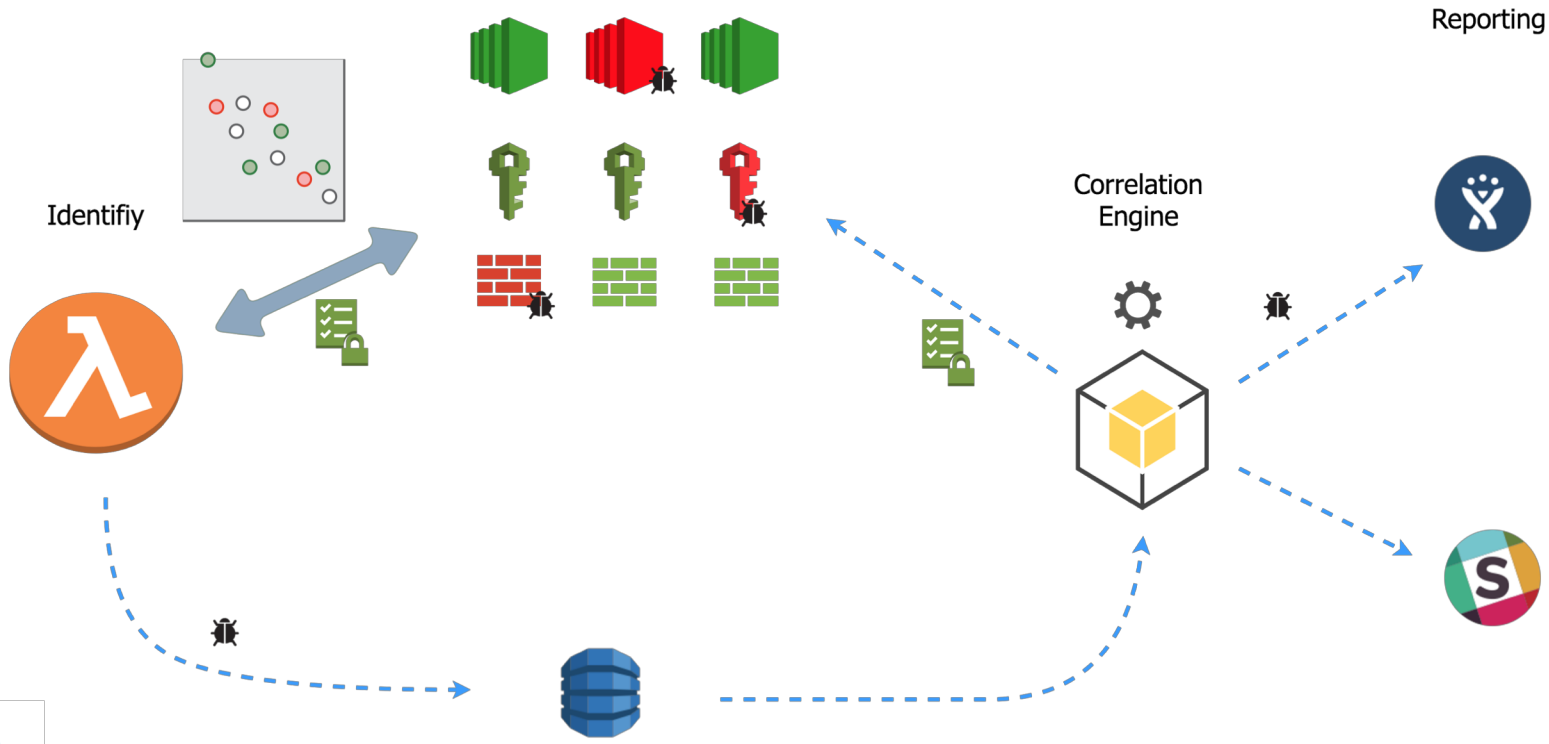
Hammer Lifecycle



Architecture

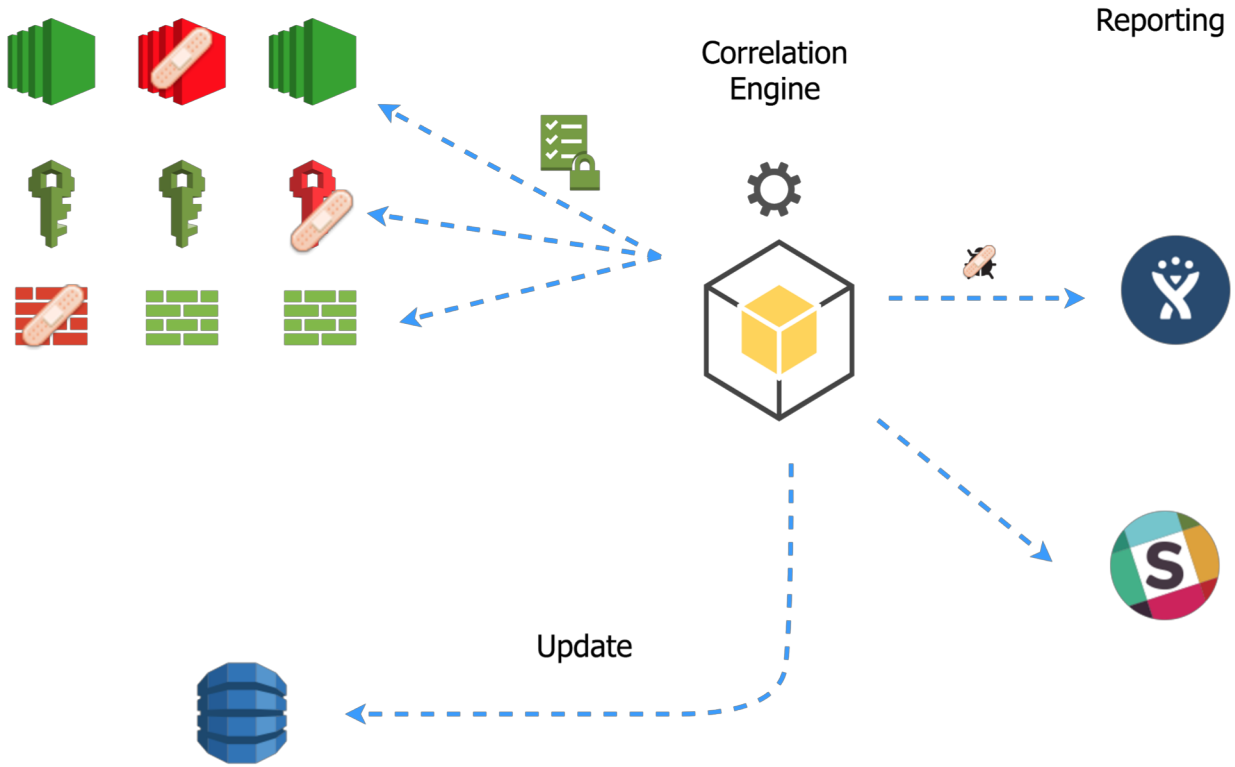


Hammer - how does it work?



DOW JONES

Hammer - /auto-fix



DOW JONES

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)	✓	✓	✓

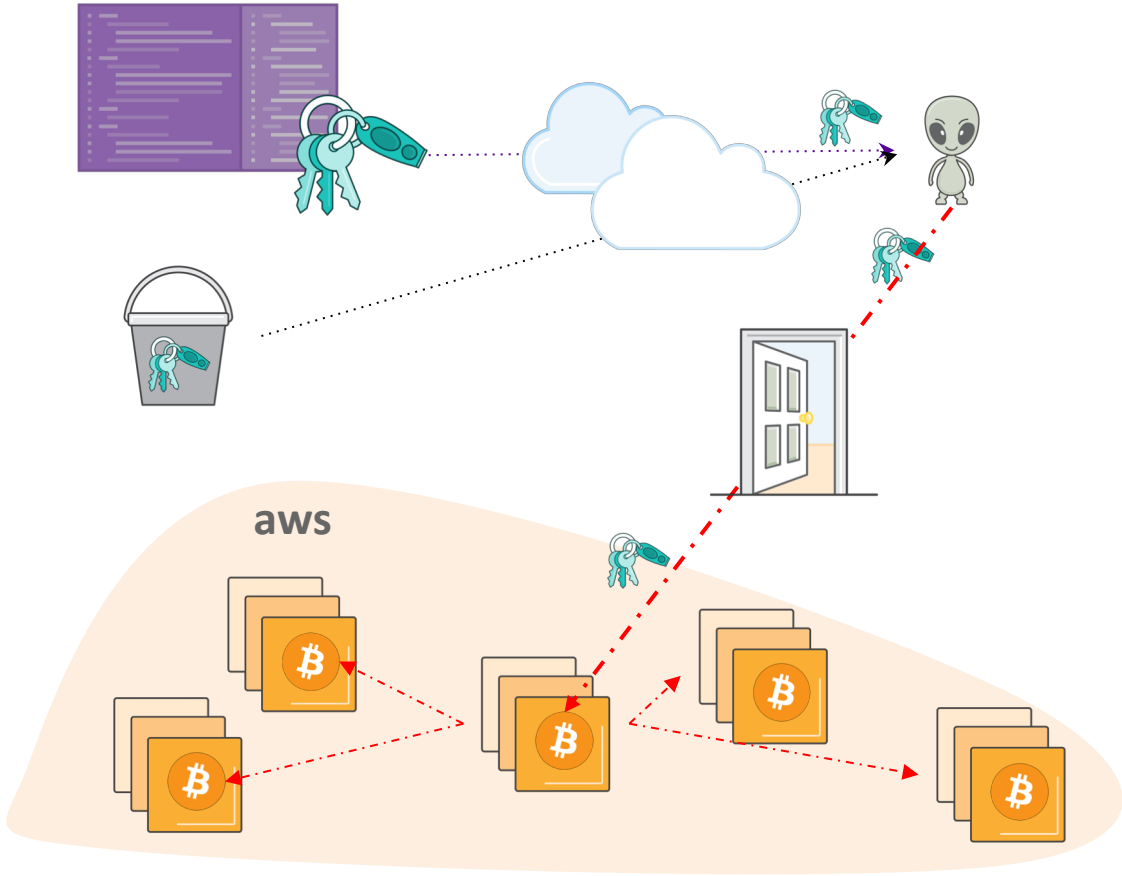
DOW JONES

Roadmap

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

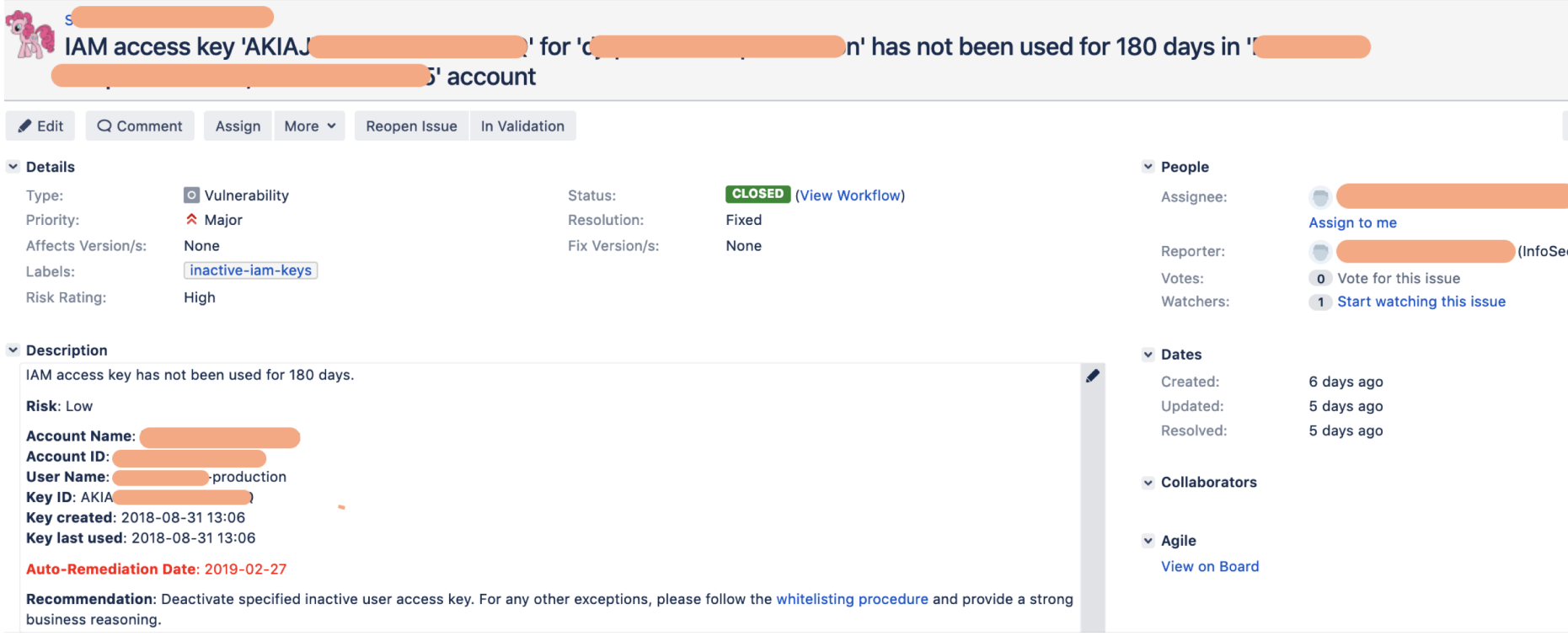
Case Study 1: Protection from bitcoin miners…..

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



DOW JONES

Case Study 1 : Example



IAM access key 'AKIAJ[redacted]' for 'c[redacted]n' has not been used for 180 days in '[redacted]' account

Edit | Comment | Assign | More | Reopen Issue | In Validation

Details

Type:	Vulnerability	Status:	CLOSED (View Workflow)
Priority:	Major	Resolution:	Fixed
Affects Version/s:	None	Fix Version/s:	None
Labels:	inactive-iam-keys		
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: AKIA[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] (InfoSe
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

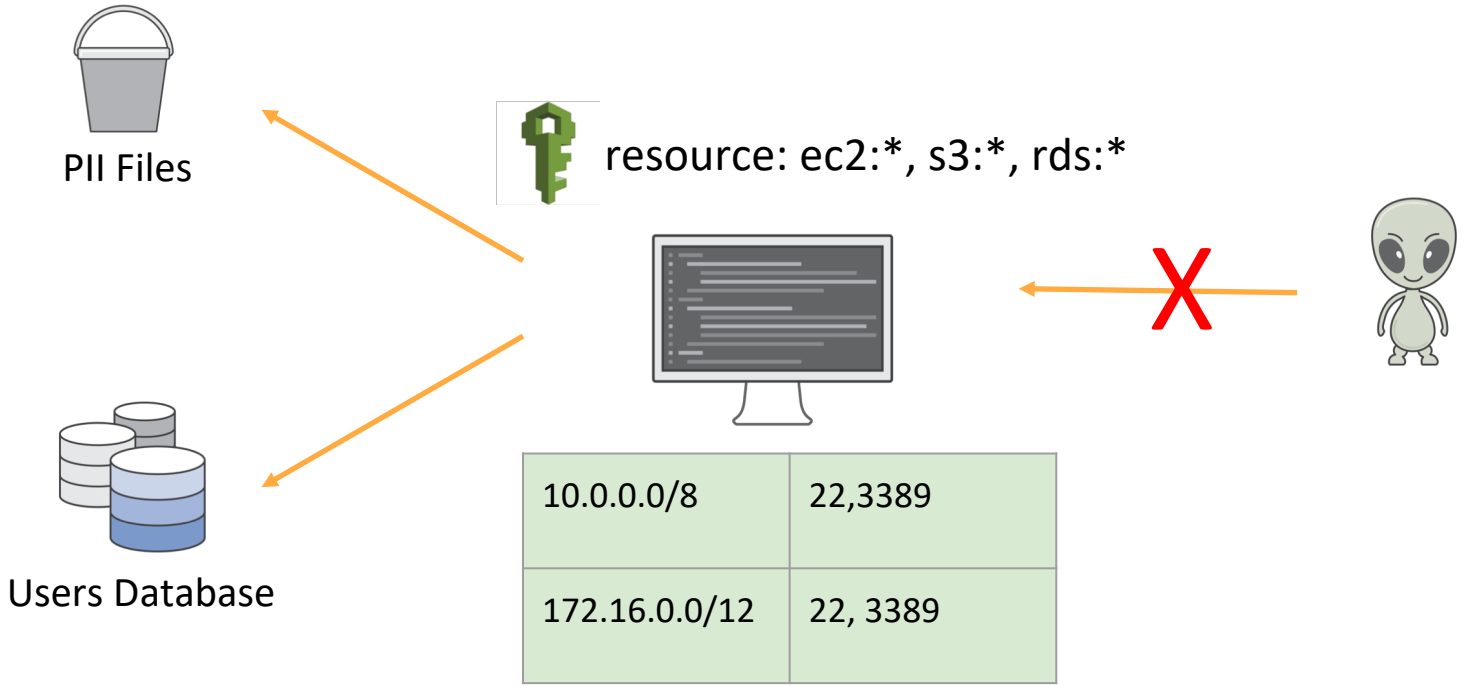
Collaborators

Agile

View on Board

DOW JONES

Case Study 2: Over Exposed Instances



DOW JONES

Case Study 2: Over Exposed Instances

Description

Security group has EC2 instances in in private subnets and source IP address with a /0 suffix for following ports:

Click to edit

To Port	Protocol	CIDR
22	tcp	0.0.0.0/0

Threat: 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.
Risk: High

Account Name:
Account ID:

Instance Role Unsafe Policies:

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

DOW JONES

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 acce
increases opportunities for malicious activity (hacking, denial-of-service attacks, loss of data).
Risk: High

Account Name: [redacted]s
Account ID: [redacted]
SG Name: [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	[redacted]sample_ec2

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

DOW JONES

Lessons Learnt

Scales over 100+ AWS Accounts

Integrates to Dev ecosystem

Easy to plug and play

Tonnes of insecurities fixed

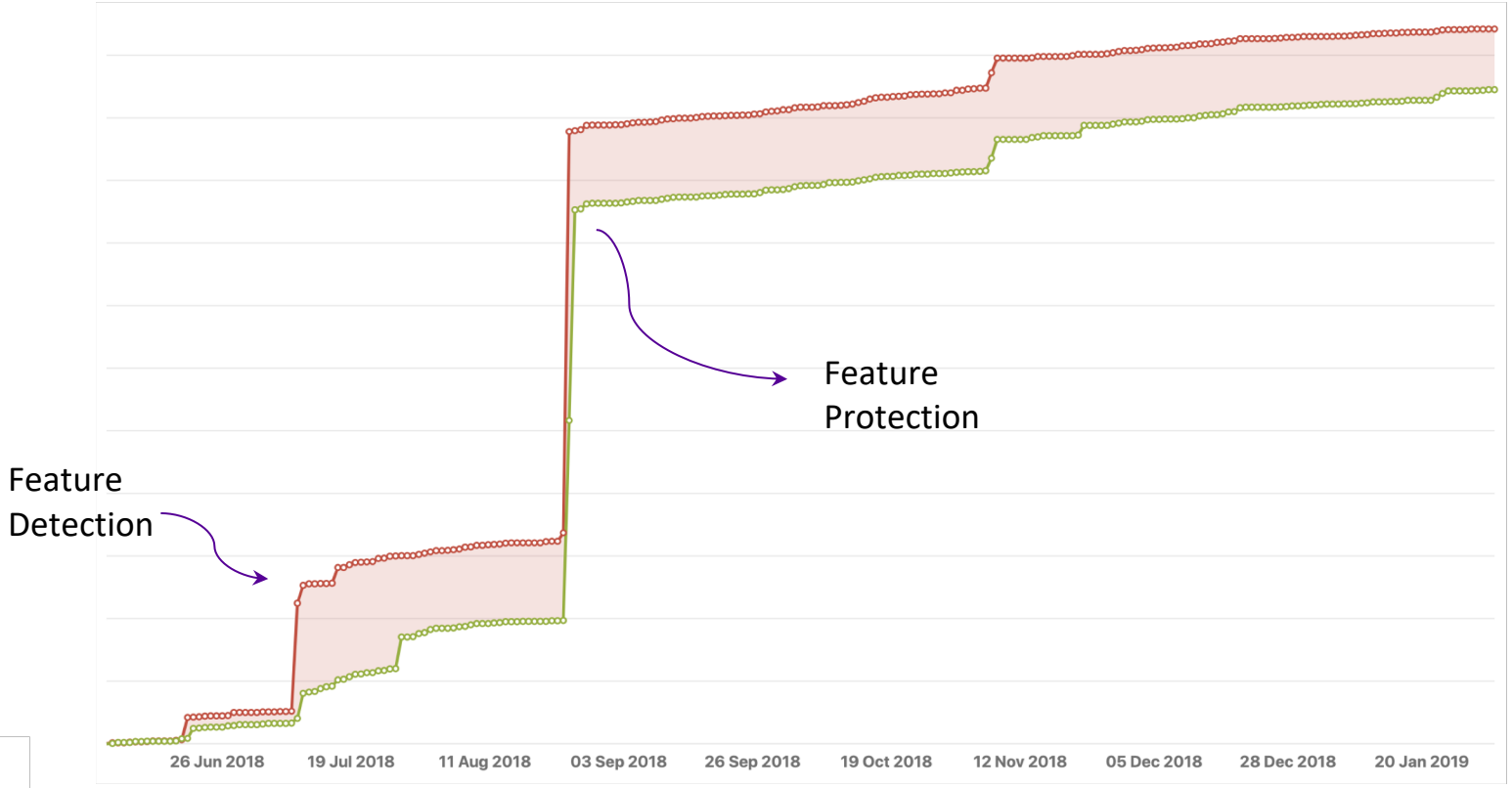
Low to minimal Impact

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

RSAConference2019

Scaling security and embedding into pipeline

Project Braavos

Dow Jones Hammer

Plan

Code

Build

Test

Deploy

Project Bravos : CICD Integration

Sensitive information in code

Continuous Security

Open Source Vulnerability Checks

Static code analysis

Risky function calls

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

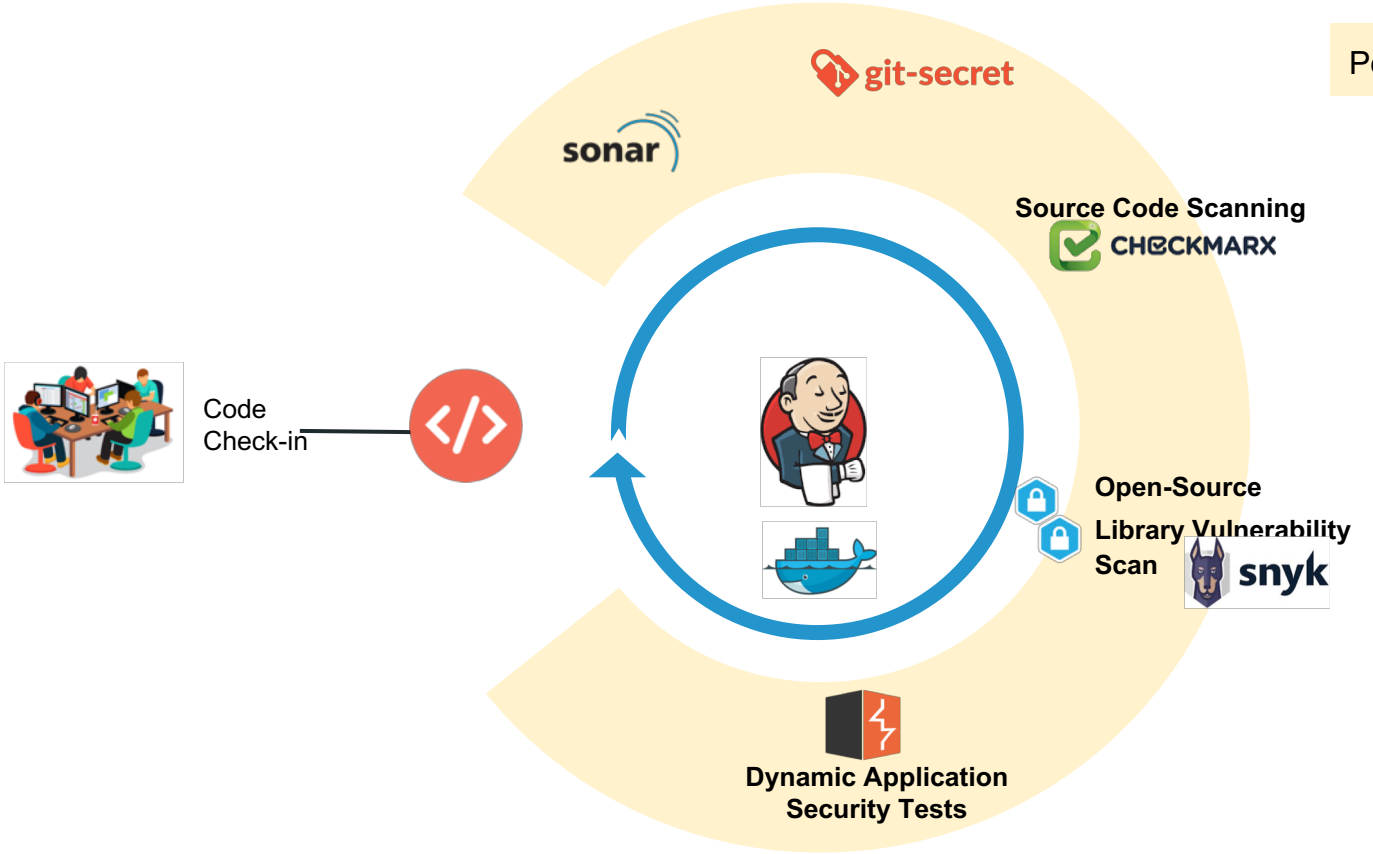
Deploy

- Dow Jones Hammer
- Burp

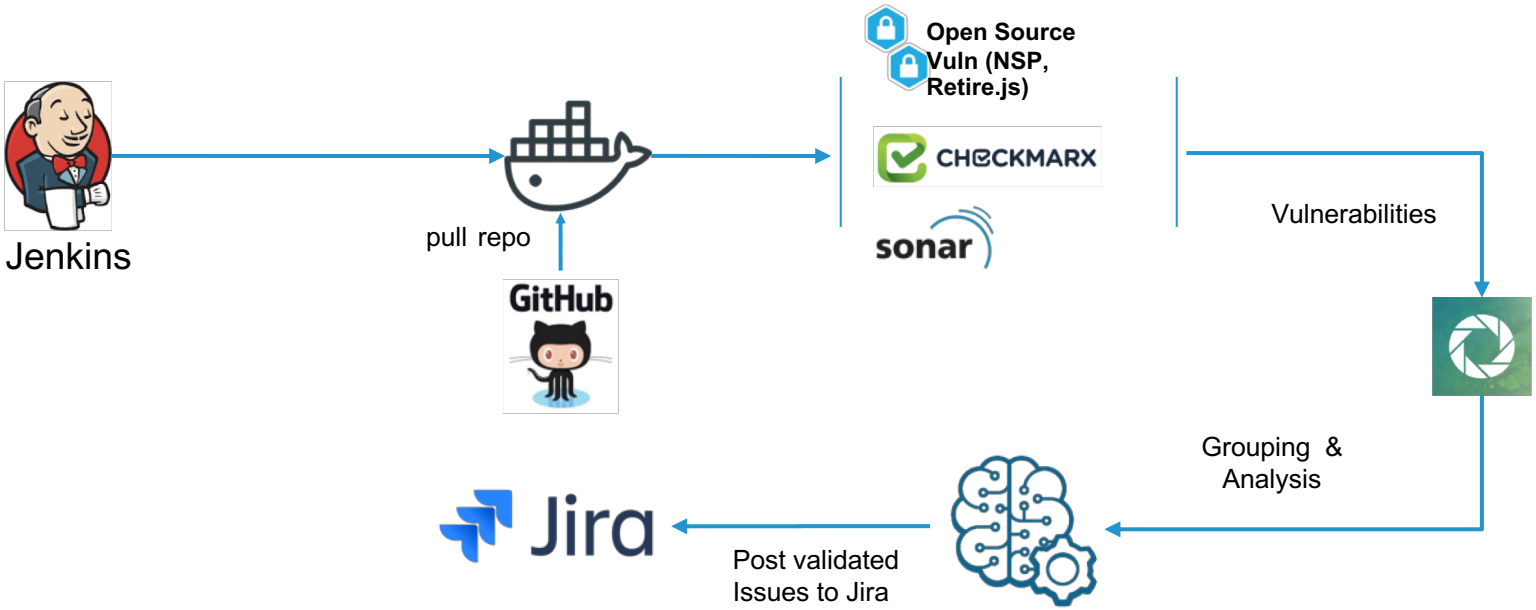
Ref: https://www.owasp.org/index.php/OWASP_AppSec_Pipeline#tab=Pipeline_Tools

DOW JONES

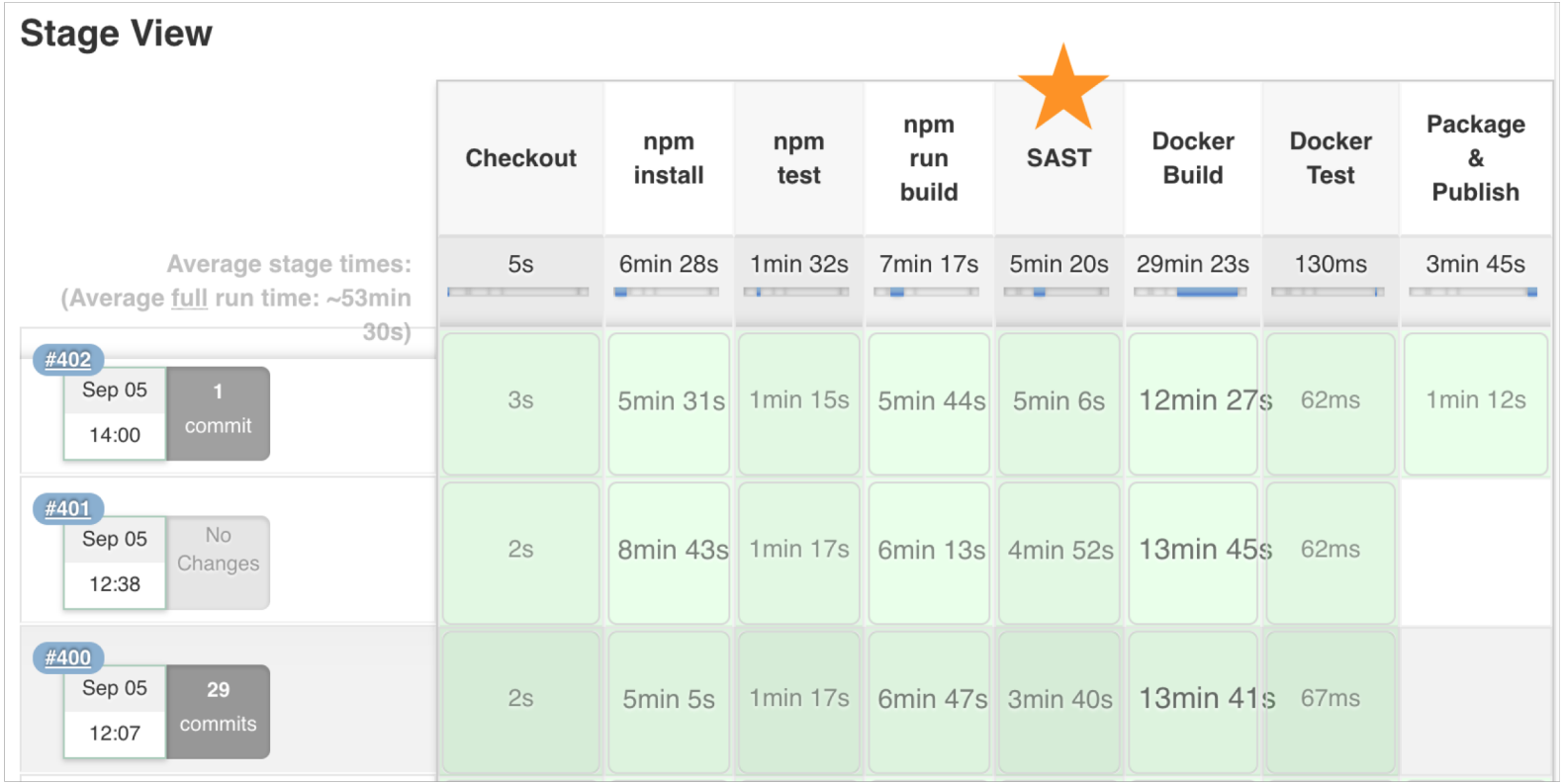
Our Solution : project bravos



/continuous_security : sast



/continuous_security: Jenkins Pipeline



Stage View

	Checkout	npm install	npm test	npm run build	SAST	Docker Build	Docker Test	Package & Publish
Average stage times: (Average full run time: ~53min 30s)	5s	6min 28s	1min 32s	7min 17s	5min 20s	29min 23s	130ms	3min 45s
#402 Sep 05 14:00 — 1 commit	3s	5min 31s	1min 15s	5min 44s	5min 6s	12min 27s	62ms	1min 12s
#401 Sep 05 12:38 — No Changes	2s	8min 43s	1min 17s	6min 13s	4min 52s	13min 45s	62ms	
#400 Sep 05 12:07 — 29 commits	2s	5min 5s	1min 17s	6min 47s	3min 40s	13min 41s	67ms	

/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

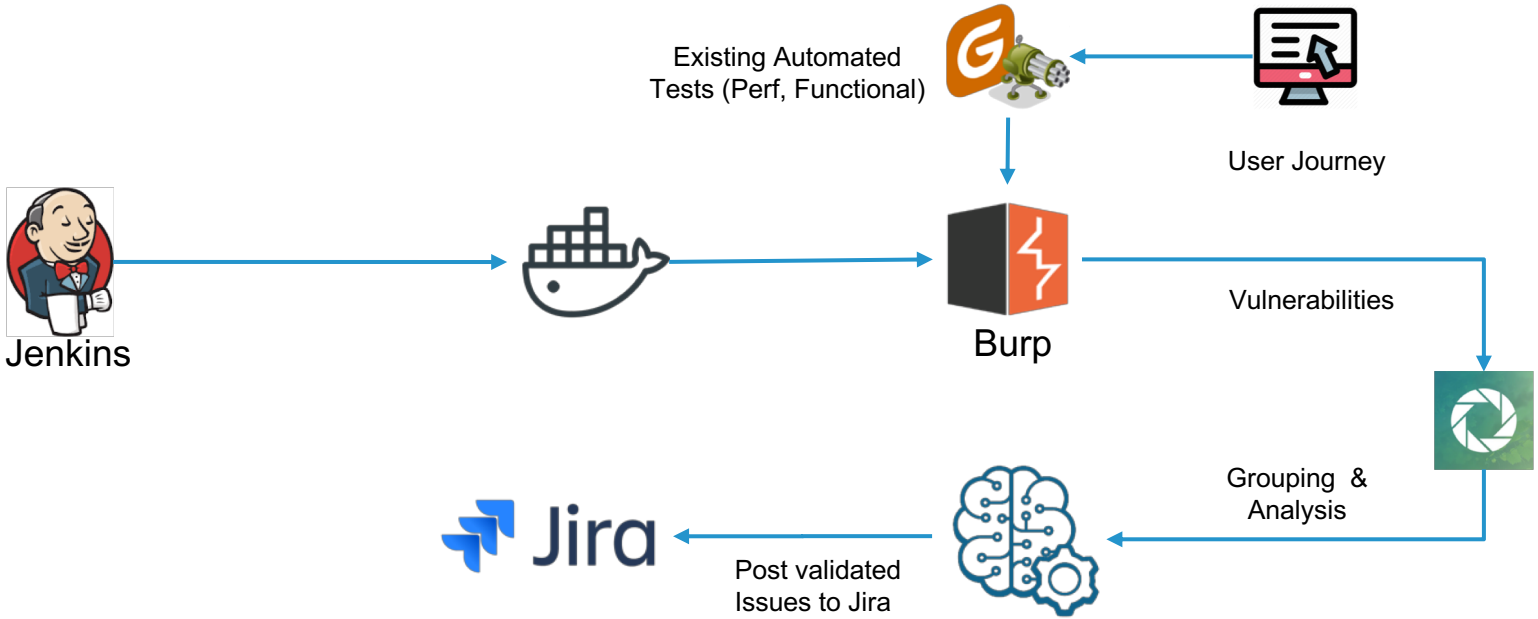
False Positives

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”

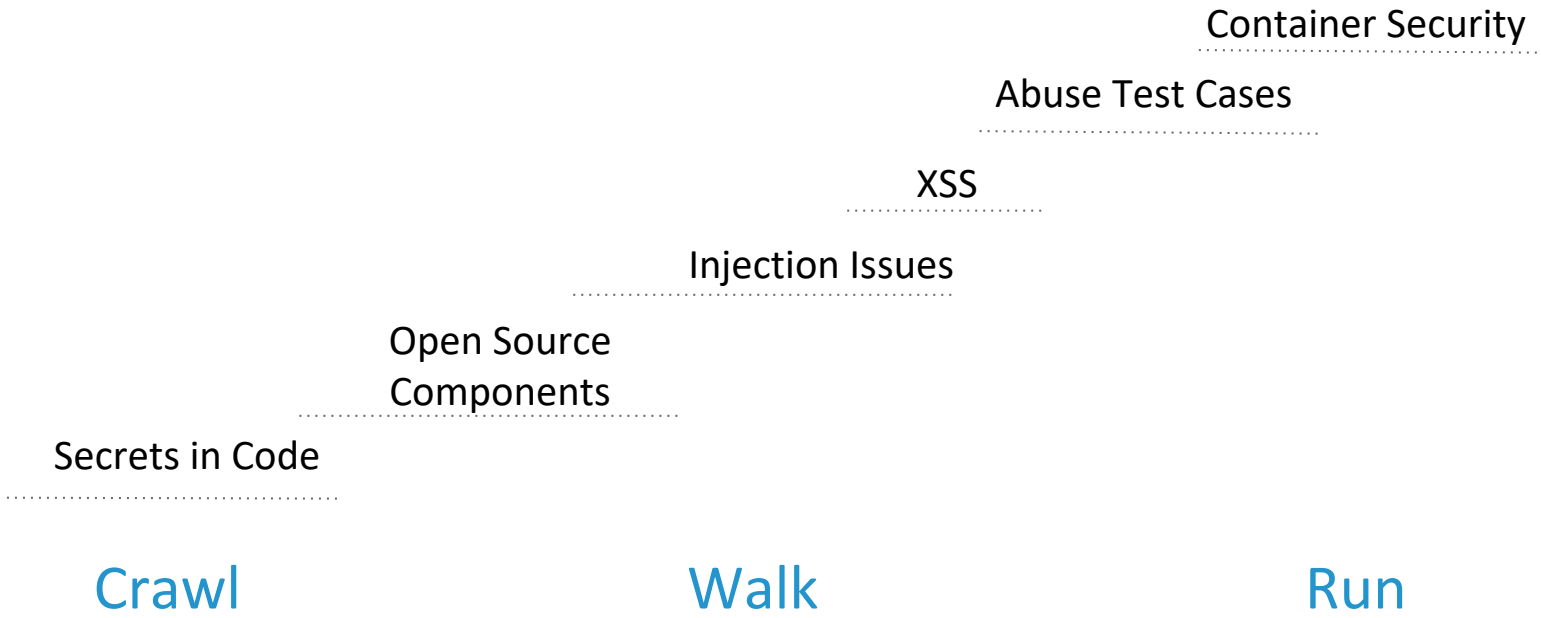
Poor Integration Options

Execution Time

More hand-holding needed

Breaking the build : Myth?

Reporting! Reporting! Reporting!



Case Study : Example 1

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

DOW JONES

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

Pranav Patel


https://github.com/kelath


https://github.com/pranav1688


@kelath


@pranav16