RSAConference2019
San Francisco | March 4–8 | Moscone Center

SESSION ID: CSV-W02

Protecting the Cloud with the Power of Cloud

Jay Kelath
Head of Product Security

Pranav Patel
Lead Security Engineer

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 Insecurities

- Discuss various insecurities in Cloud environments
- How our 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*

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

Quick Flashback….

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

Challenges

Legacy

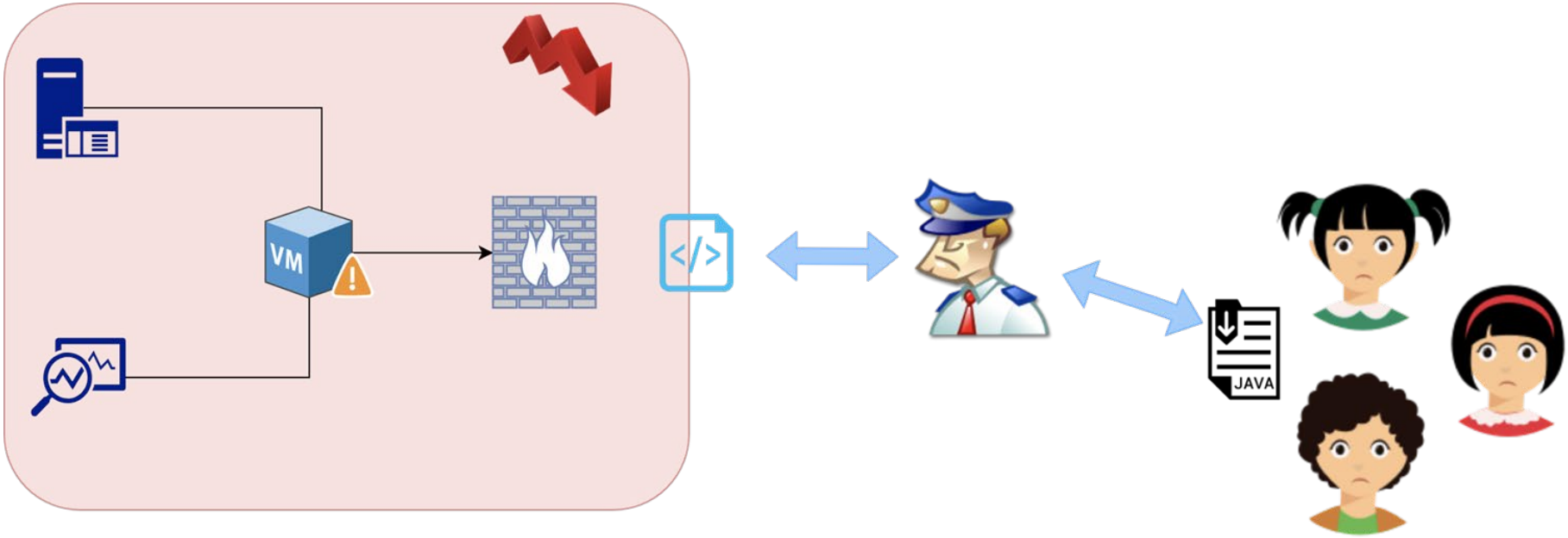
Technology Sprawl

People

Process

DOW JONES

Traditional Security



DOW JONES

RSAConference2019

“Don't Let a Good Crisis Go to Waste”

DevSecOps

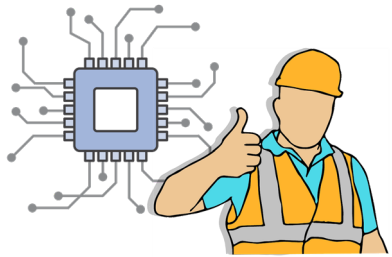
Solve a specific problem in an automated manner with well defined People, Process, Technology actions and extensive, actionable reporting

DOW JONES

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

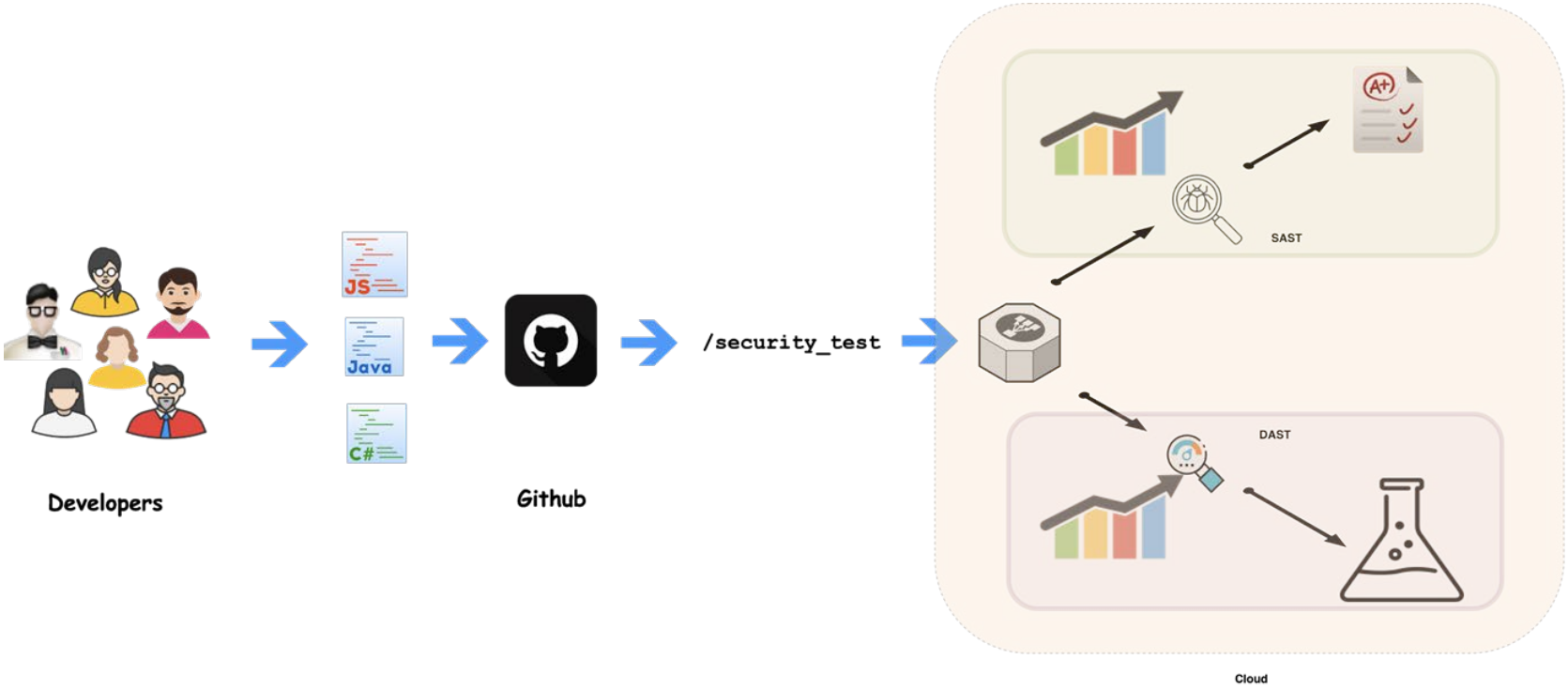
People

- Support Model
- Build Trust
- Developer support

Continuous Security

https://memegenerator.net/instance/82149927/saltbae-roast-security

DevSecOps way….



DOW JONES

RSAConference2019

Dow Jones Hammer

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

Cloud migration journey

- Agile
- Success story out of migration
- “Lift & Shift” applications
- Cost management
- “Fail forward fast...”
- “Act now, apologize later….”
- “Move Fast Break Things….”

https://memegenerator.net/instance/80795081

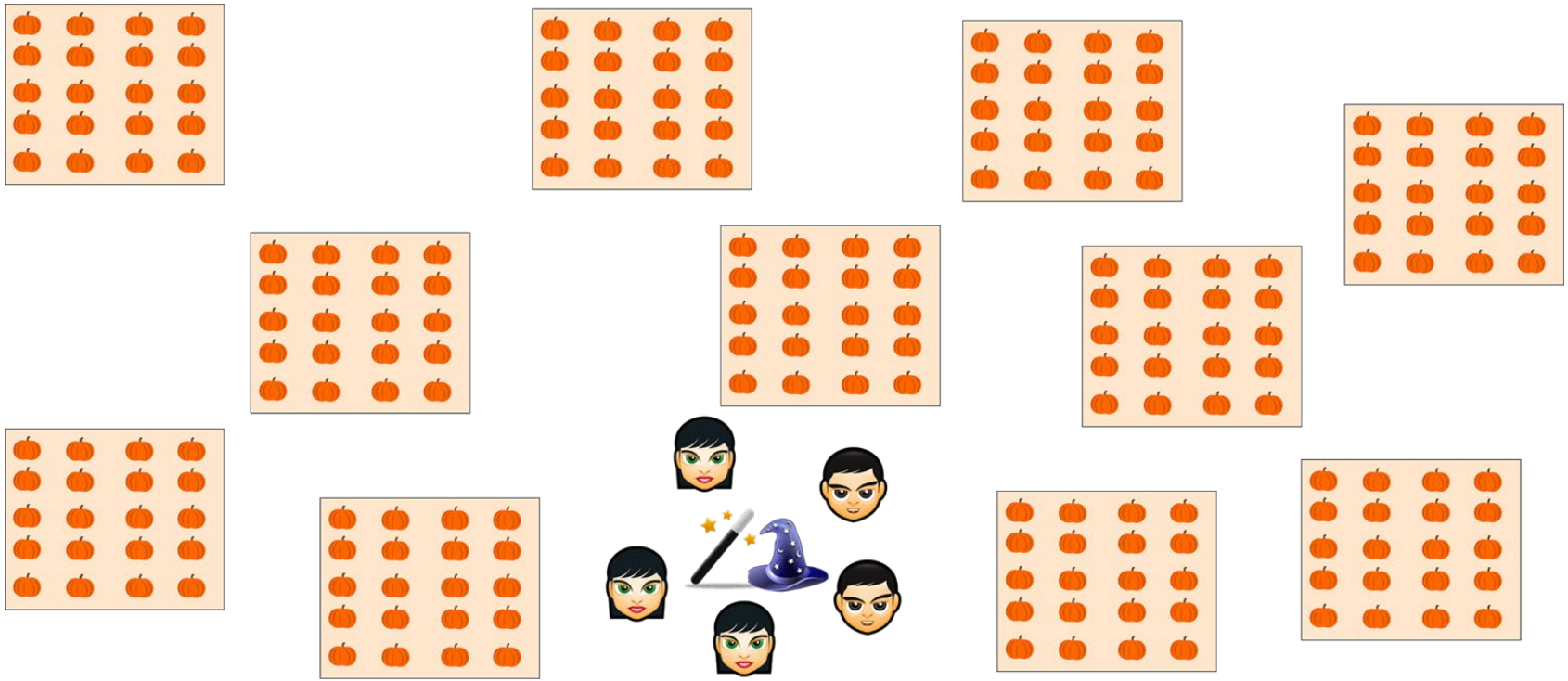
Security ??

- Legacy cloud (....already?)
- Traditional security tools do not measure up
- Multi-Account visibility and controls
- Shared Responsibility Model
- Change in Landscape

"While moving fast and breaking things,
We forget to fix things!!!"

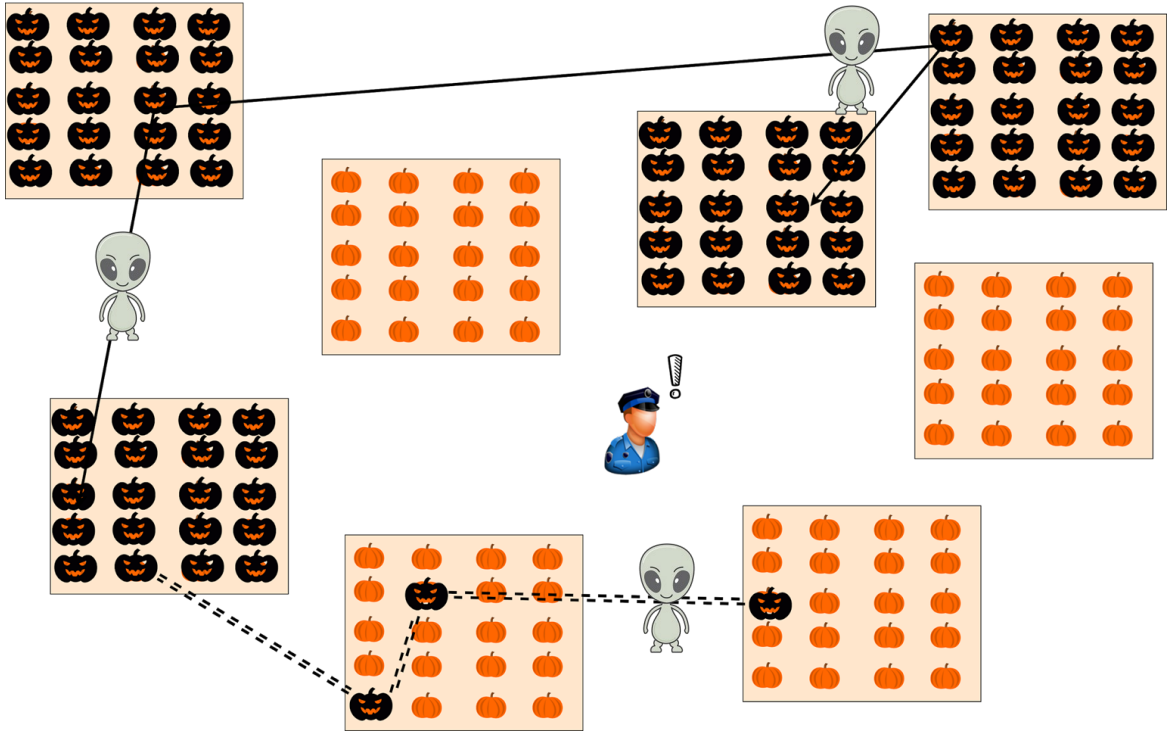
https://memegenerator.net/instance/80794879

Multi-account sprawl

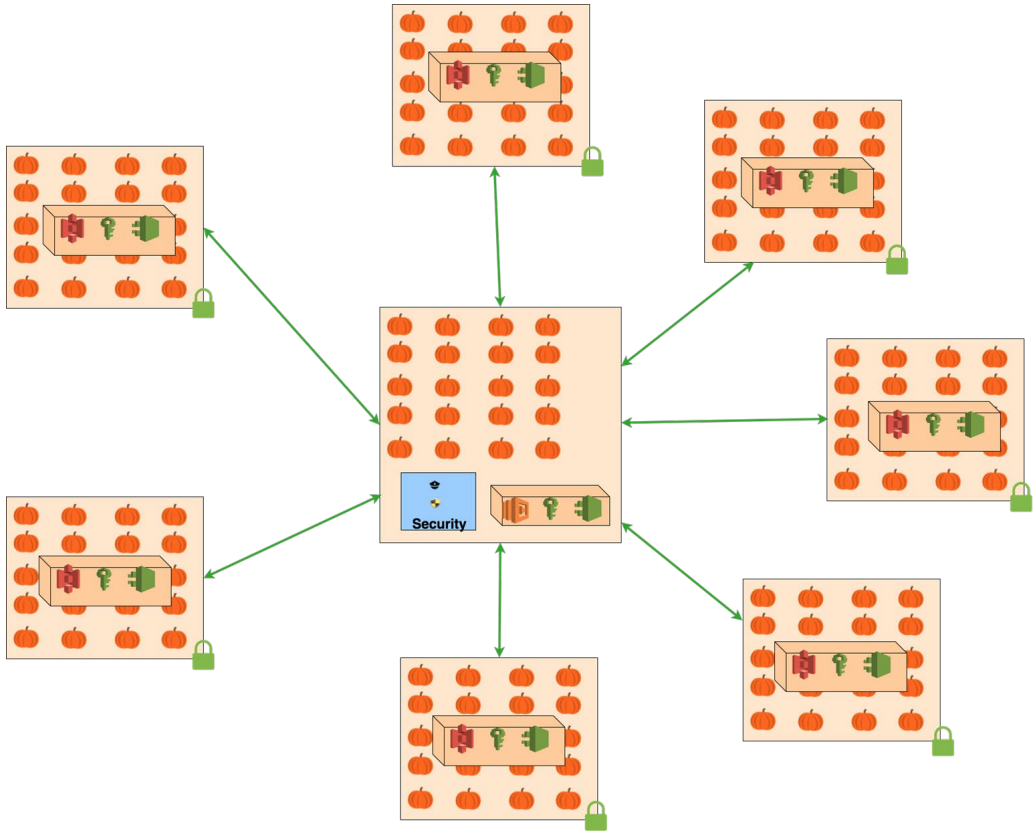


Multi-account sprawl

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



Multi-account growth : Ideal State

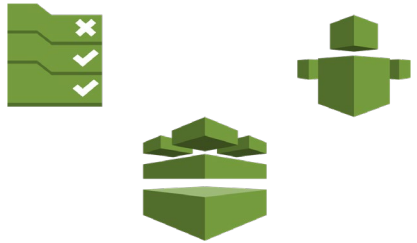


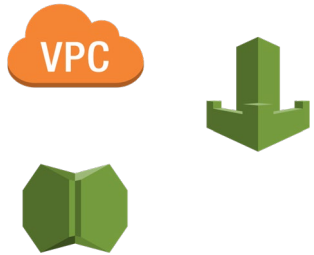
Security: Defense-in-depth

Detective Controls

Proactive Controls

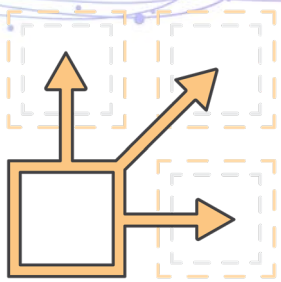
Reactive Controls

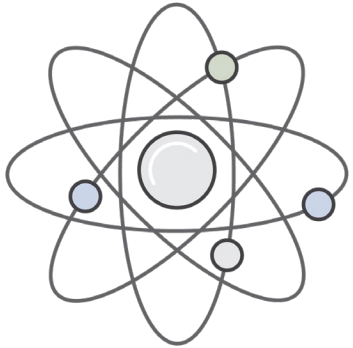




Our Solution

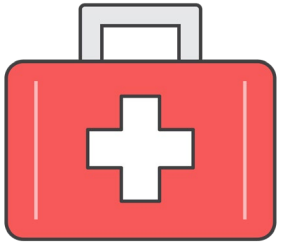
- Automate
- Scalable
- Self-service
- Auditable





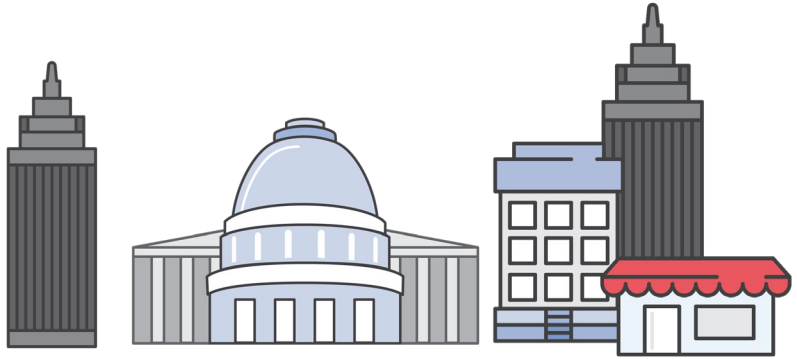
Hammer… why?

- Cloud Infrastructure visibility
- Easily pinpoint MY product's security issues
- Tailored reporting, save analysis time
- "Auto-fix" misconfigurations with ability to rollback



Consumer

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



Hammer: What Does it Solve?

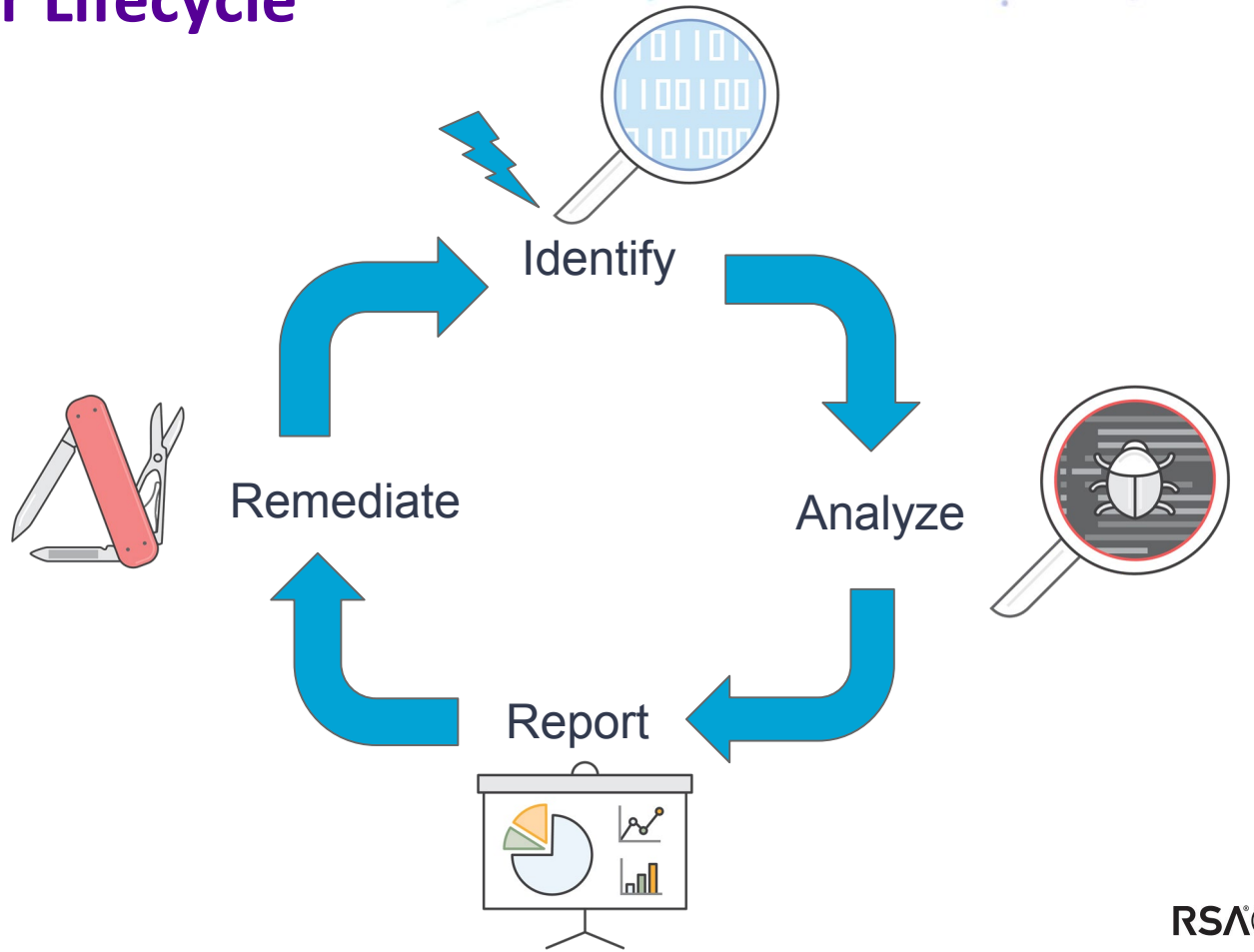
Public Instances with Admin IAM Policies
Exposed EC2 Instances
Docker on EC2
ECS

Public S3 buckets
Unencrypted S3 buckets
Public S3 bucket Policy

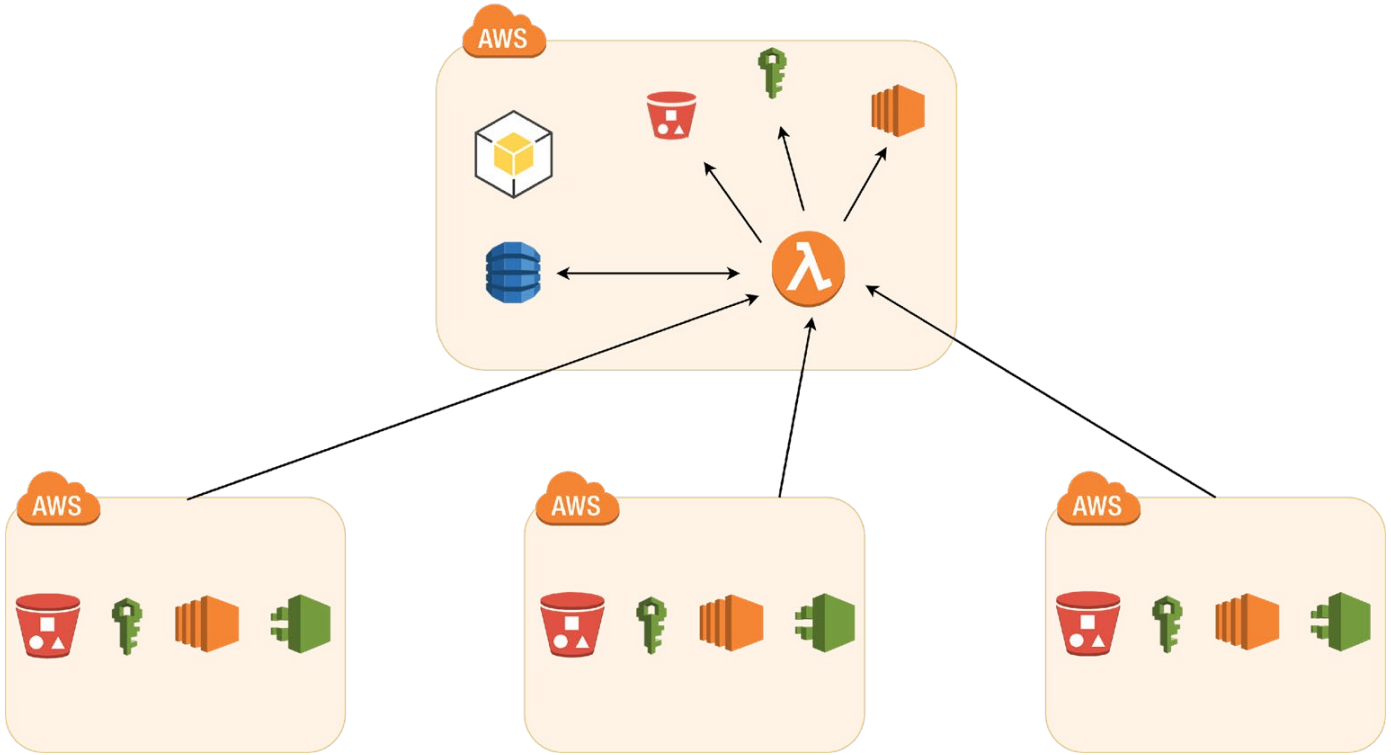
Exposed RDS instances
Unencrypted RDS
Public RDS Snapshots

Unused IAM Keys
Stale IAM Keys (not rotated)

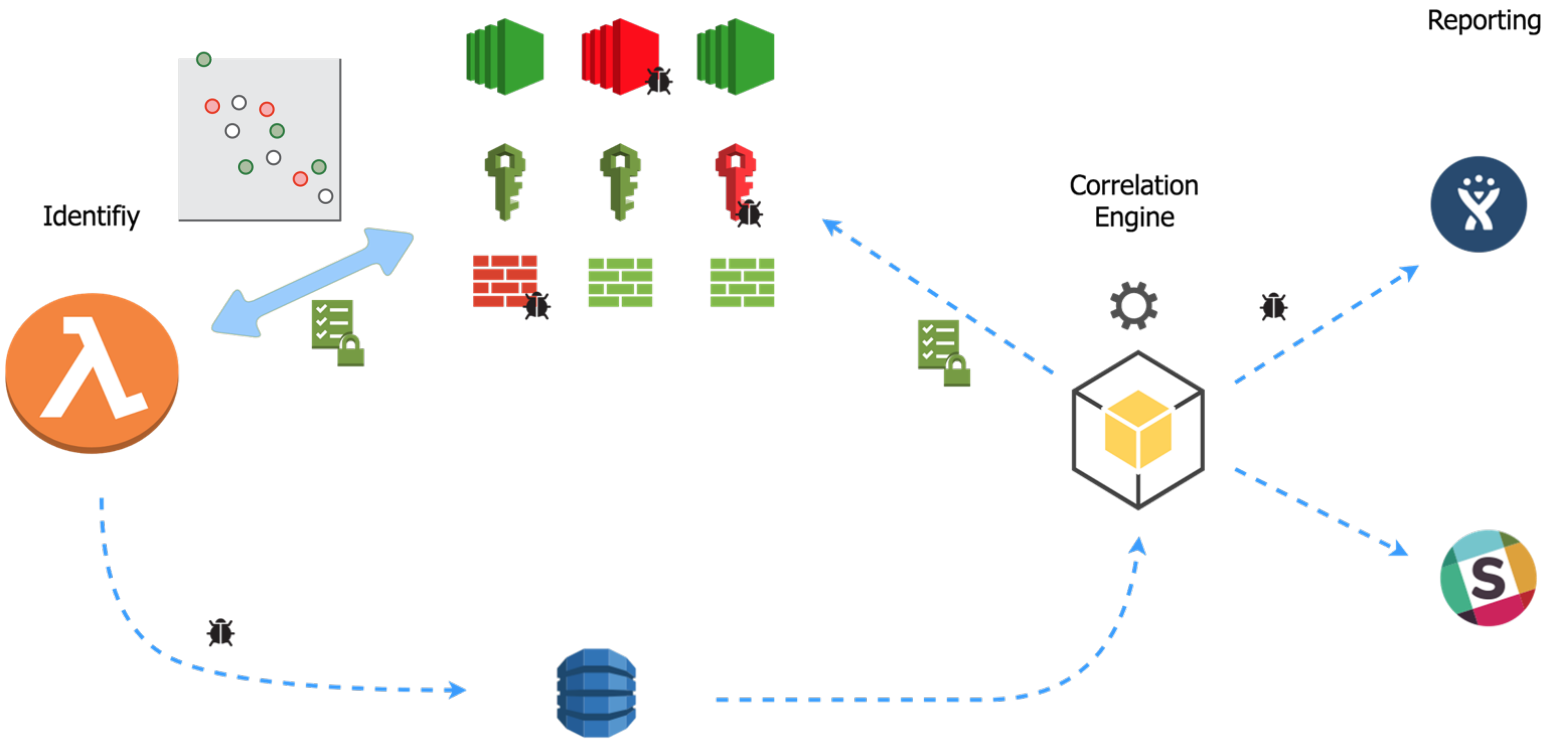
Hammer Lifecycle



Architecture

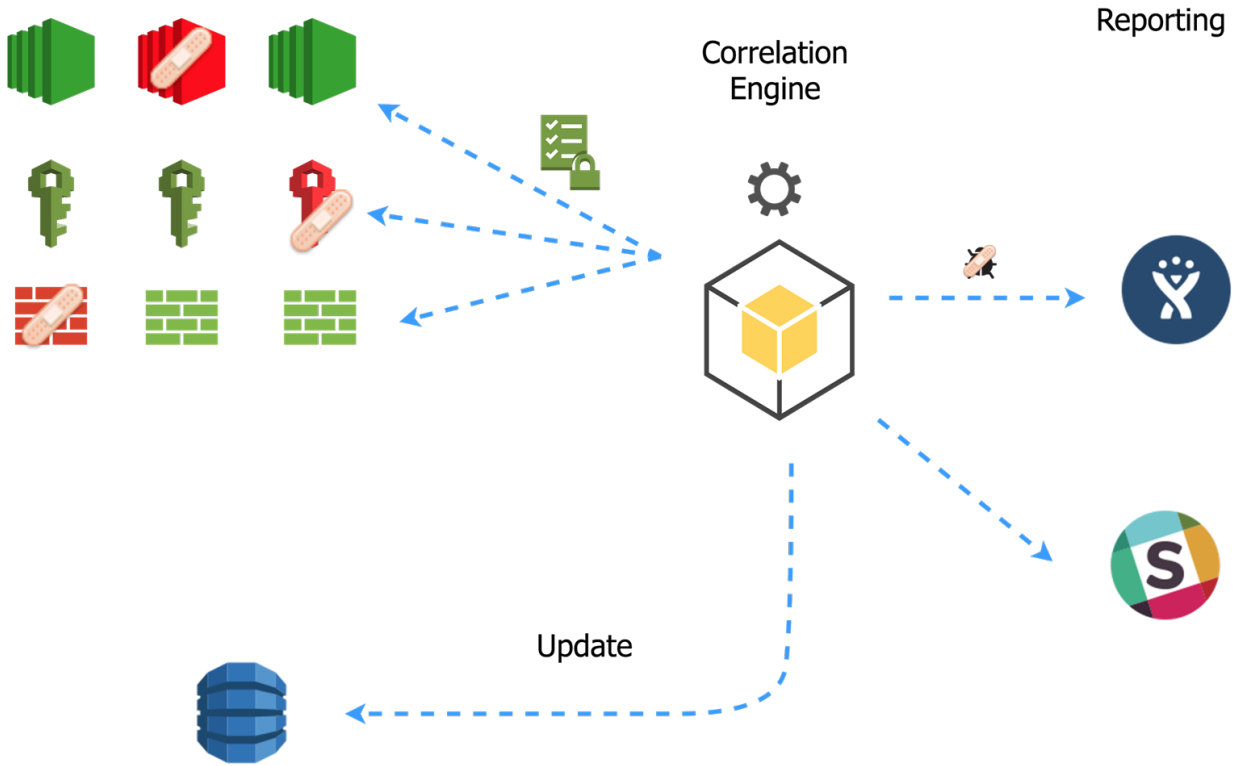


Hammer - how does it work?



DOW JONES

Hammer - /auto-fix

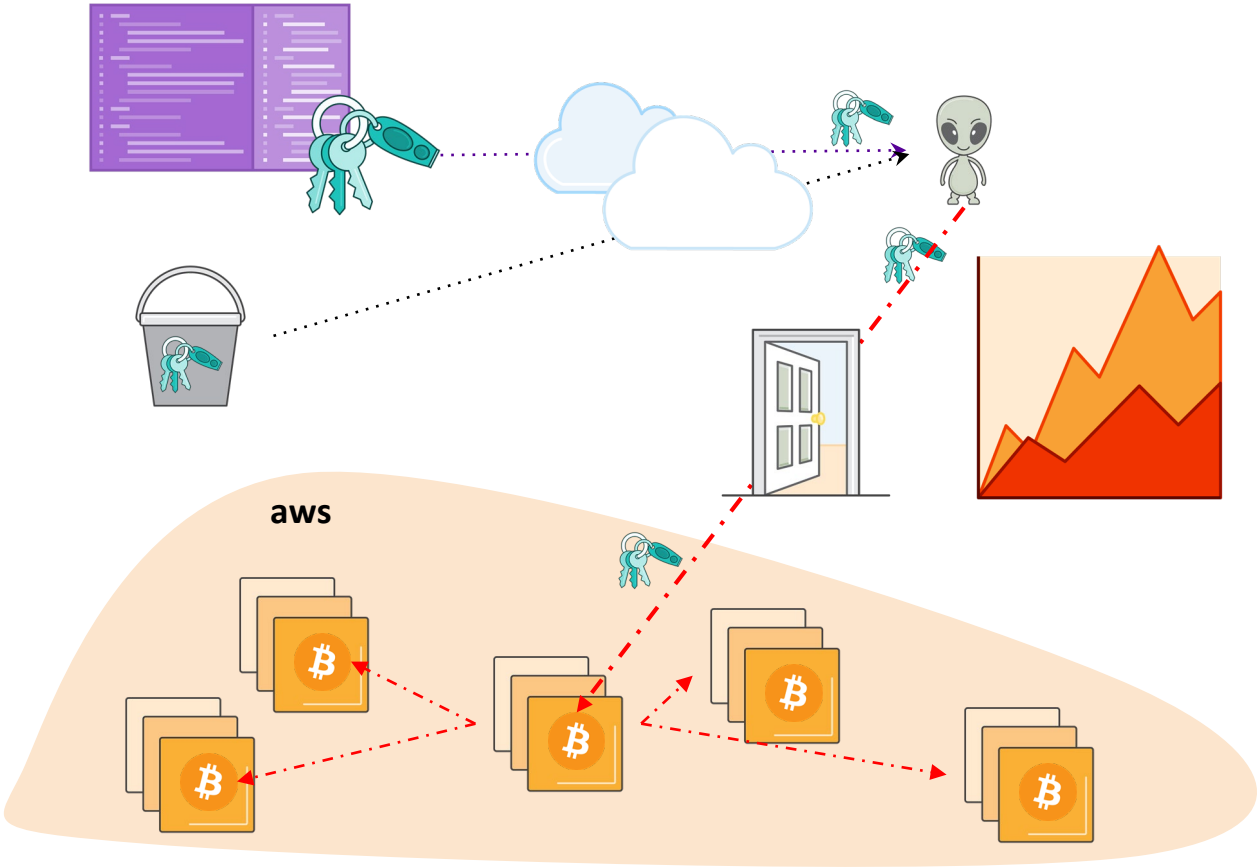


DOW JONES

Case Study 1: Protection from bitcoin miners…..

- Stale & Exposed keys in Code
- Stale & Exposed keys in public bucket
- “Exes” with Keys

Action :: Just Deactivate unused Keys



DOW JONES

Case Study 2: Exposed Instances

Action : Lock down Non-Web Ports to Private DMZ

www.acme.com

Static Files

0.0.0.0	80/443
0.0.0.0	22

resource: s3:Get*
resource: rds:*

Users Database

Case Study: Example 1

SECVULN / SECVULN-7775

Intranet open security group 'int_p le_ec2' in ehpdhj-stack-AWSEBSe

Ec2 Instances:

Instance ID	State	Private Ip Address	Public Ip Address	Owner	Business unit	Product	Component	Subnet
i-033[redacted]64a	stopped	10.20[redacted]	-	Sl[redacted]@dowjones.com	djs[redacted]	[redacted]tsvc	othr	private
i-070[redacted]1028	running	10.20[redacted]	34.22[redacted]	Sl[redacted]@dowjones.com	djs[redacted]	[redacted]svc	othr	private

Instance Role Unsafe Policies:

Instance Id	Role Name	Policy Name	Unsafe actions

Risk Rating: High

ess attached and allows access from any IP addr

Description

Security group has EC2 instances in in private subnets and allows access from some definite public ip addre and make them vulnerable to blind injection bas the internet incoming, traffic will reach instance an IGW.

From Port	To Port	Protocol	CIDR	Registrant
22	22	tcp	172.0.0.0/8	AT&T Internet Services (SIS-80)

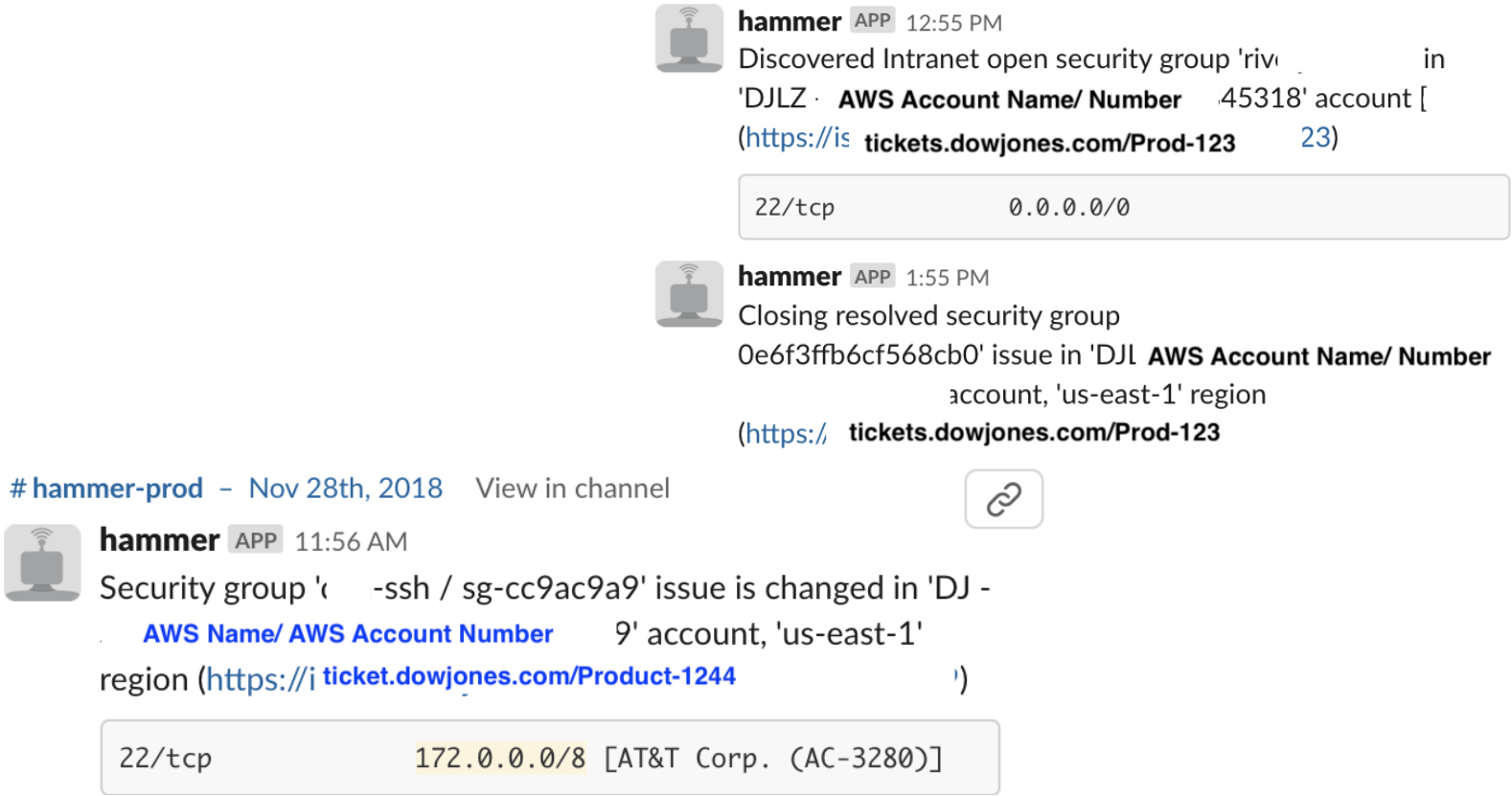
Threat: Open access within the network not only provides unrestricted access to other servers but increase (hacking, denial-of-service attacks, loss of data) if attacker gains access to the services within the network CFY4L

Risk: High

DOW JONES

Auto-Remediation Date: 2018-10-02

Case Study : Example 2



RSAConference2019

Apply : From day 1

https://github.com/dowjones/hammer

Hammer : Service Matrix

Services	Identification	Reporting	Remediation
Public S3 (ACL)	√	√	√
Public S3 (bucket policy)	√	√	√
Inactive-IAM-keys (older than 180 days)	√	√	√
IAM-key-rotation (older than 2 years)	√	√	[Roadmap]
Insecure Services (database, rdp, ssh, telnet ports)	√	√	√
Cloud Trail Disabled	√	√	n/a
Unencrypted EBS Volumes (Compliance Accounts	√	√	n/a
Public EBS Snapshots	√	√	√

DOW JONES

Hammer : Service Matrix

Services	Identification	Reporting	Remediation
[RDS]Public Snapshots	√	√	√
[RDS]Security Groups	√	√	√
[ECS] Security Groups	√	√	√
[SQS] Insecure config	√	√	√
Cleanup - Inactive [IAM] keys	√	√	√
[IAM] Privilege Escalation (IAMadmin roles)	√	√	√
Public AMIs	√	√	√

DOW JONES

Lessons Learnt

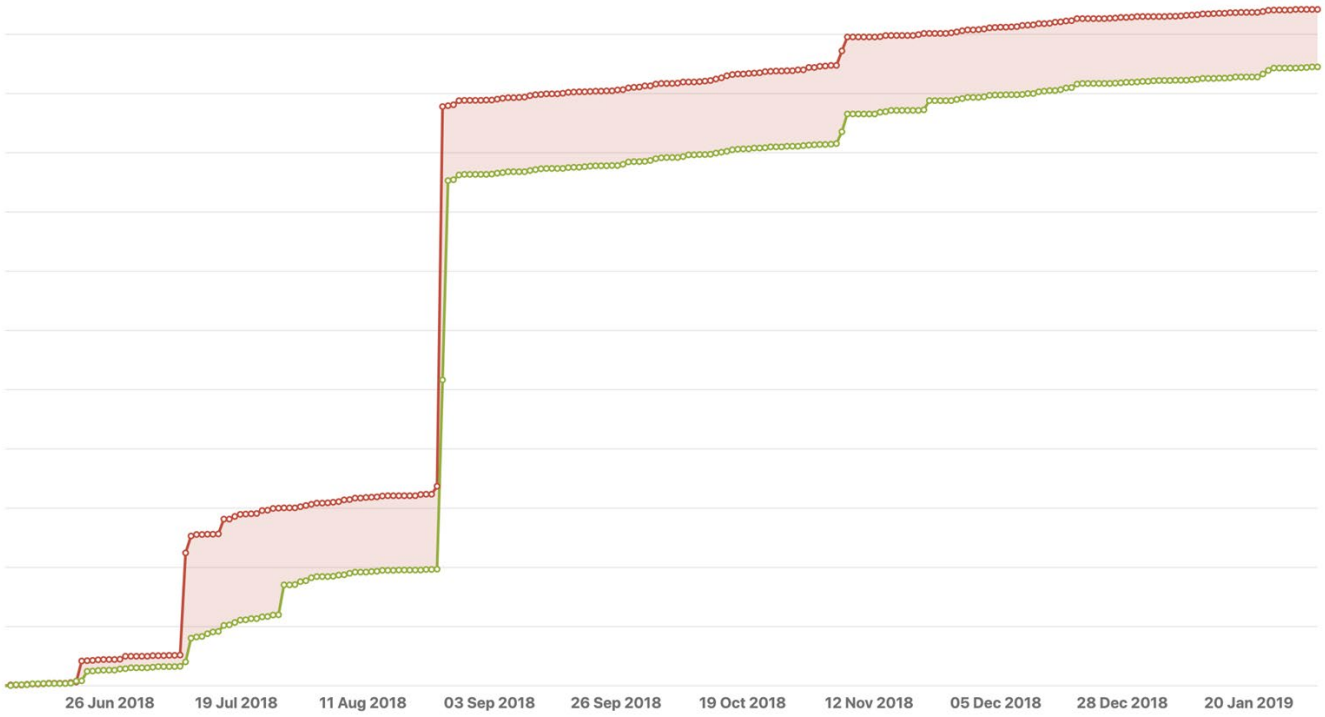
What worked..??

- Scales over 100+ AWS Accounts
- Integrates to Dev ecosystem
- Easy to plug and play
- Tonnes of insecurities fixed
- Low to minimal Impact

What did not worked..??

- Event Driven
- AWS Lambda execution limits
- Server elements
- Limited security checks

Key Trends….



DOW JONES

RSAConference2019

Scaling security and embedding into pipeline

https://github.com/dowjones/sast

Project Braavos : CICD Integration

Sensitive information in code

Continuous Security

Open Source Vulnerability Checks

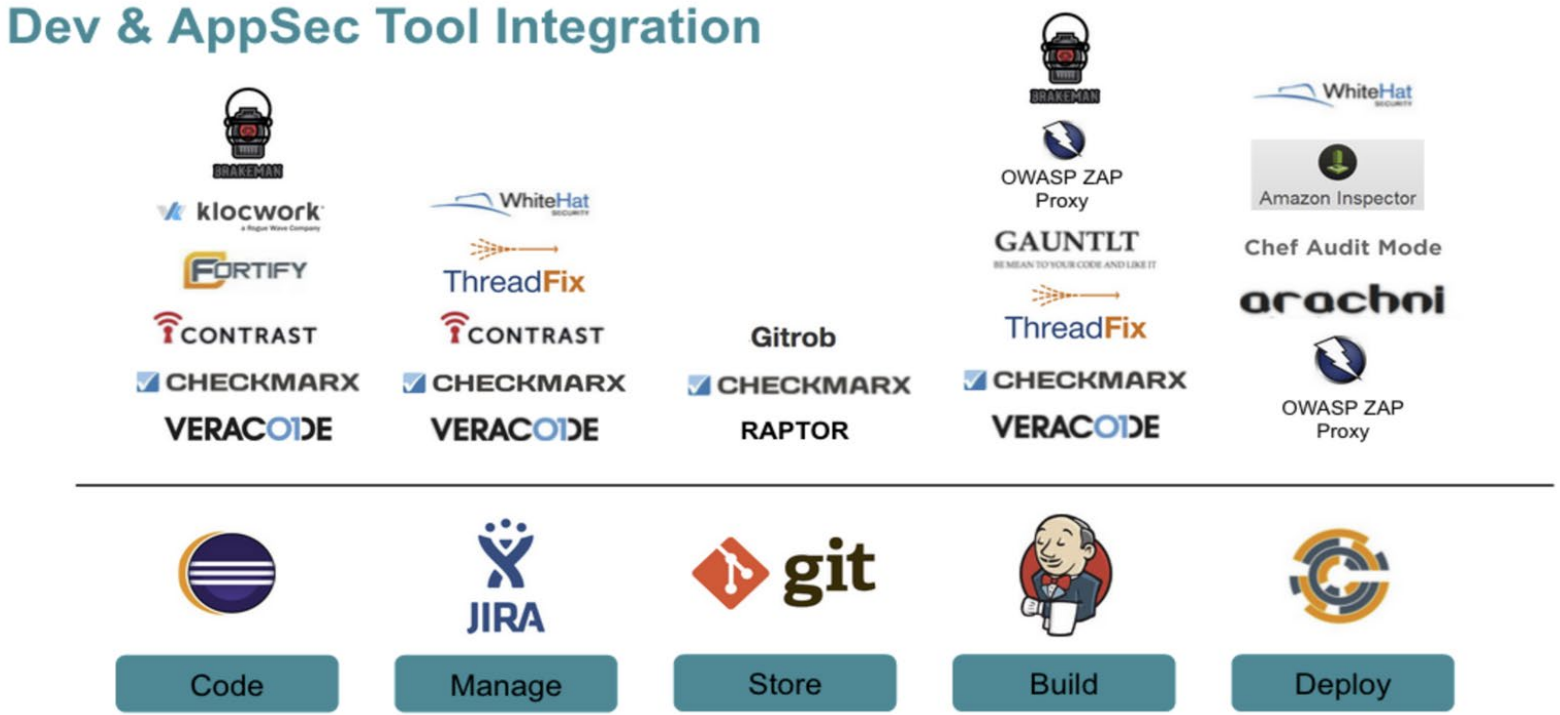
Static code analysis

Risky function calls

Security test cases

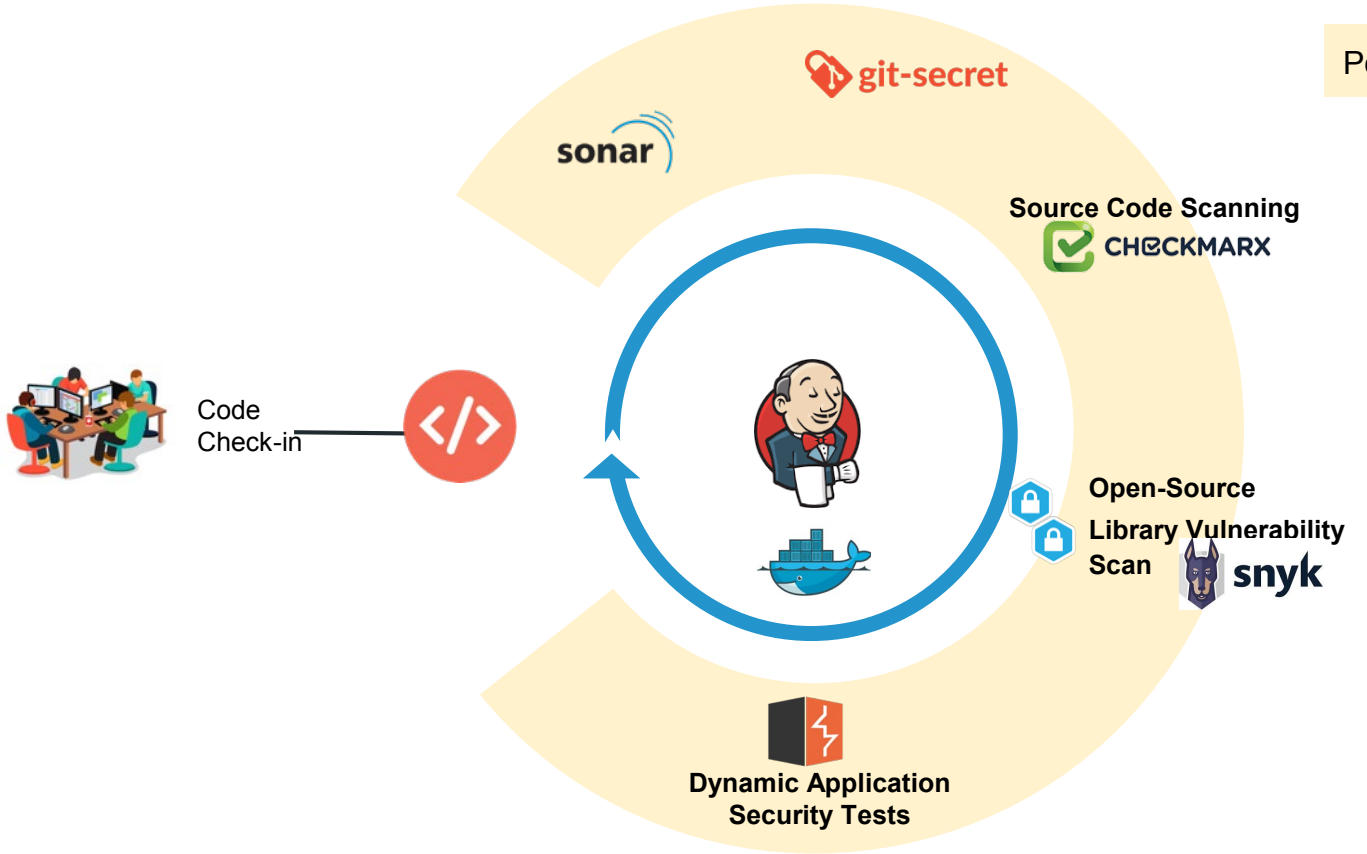
Dynamic Security Tests

OWASP Example

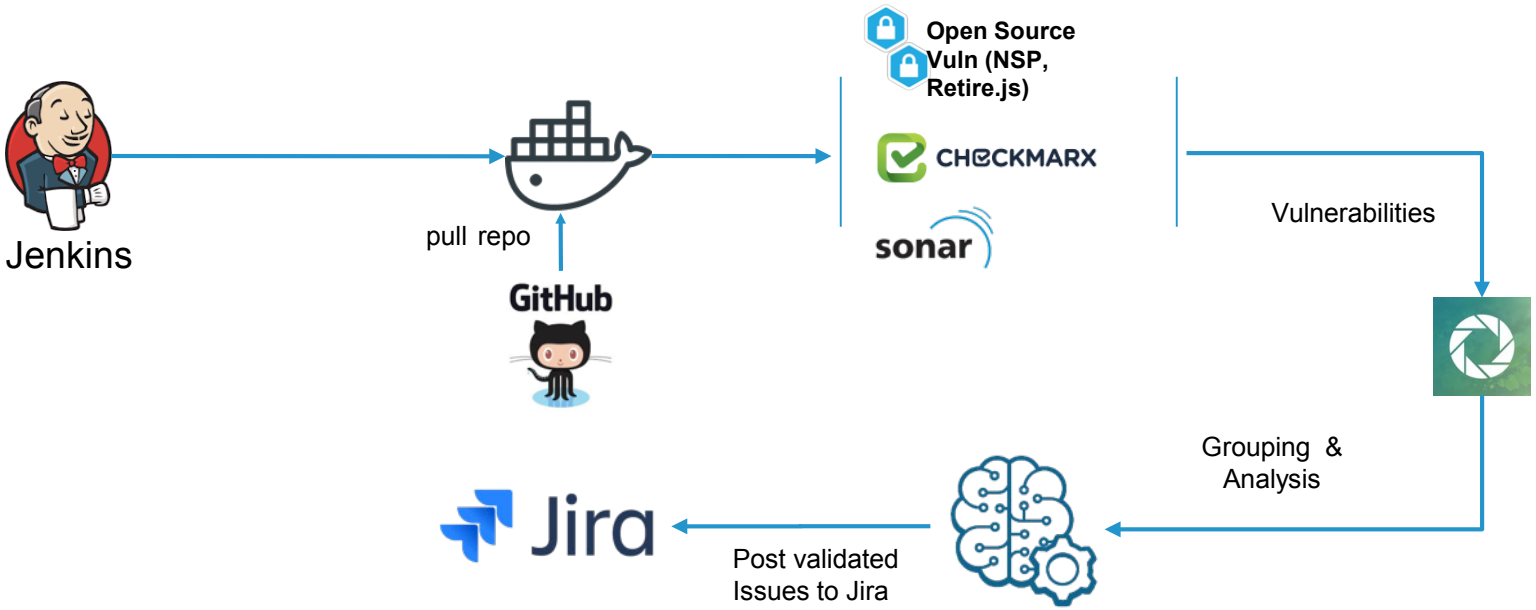


Ref: https://www.owasp.org/index.php/OWASP_AppSec_Pipeline#tab=Pipeline_Tools

Our Solution : project bravos



/continuous_security



/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(static code security testing)

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

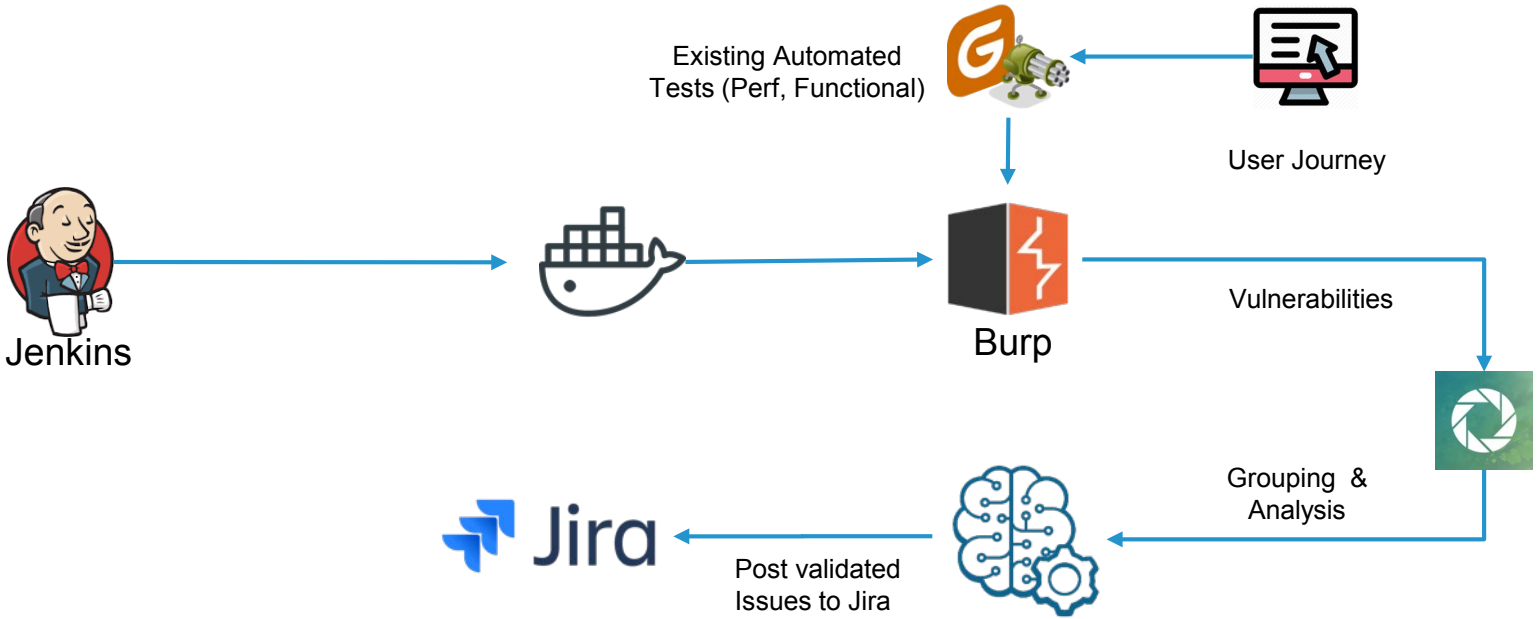
False Positives

Execution Time

Limited Tech Coverage

“Glue code” for Integration

Dynamic Scanning in Pipeline



Lessons Learnt

Existing Test Cases FTW

Focus on your Top 5

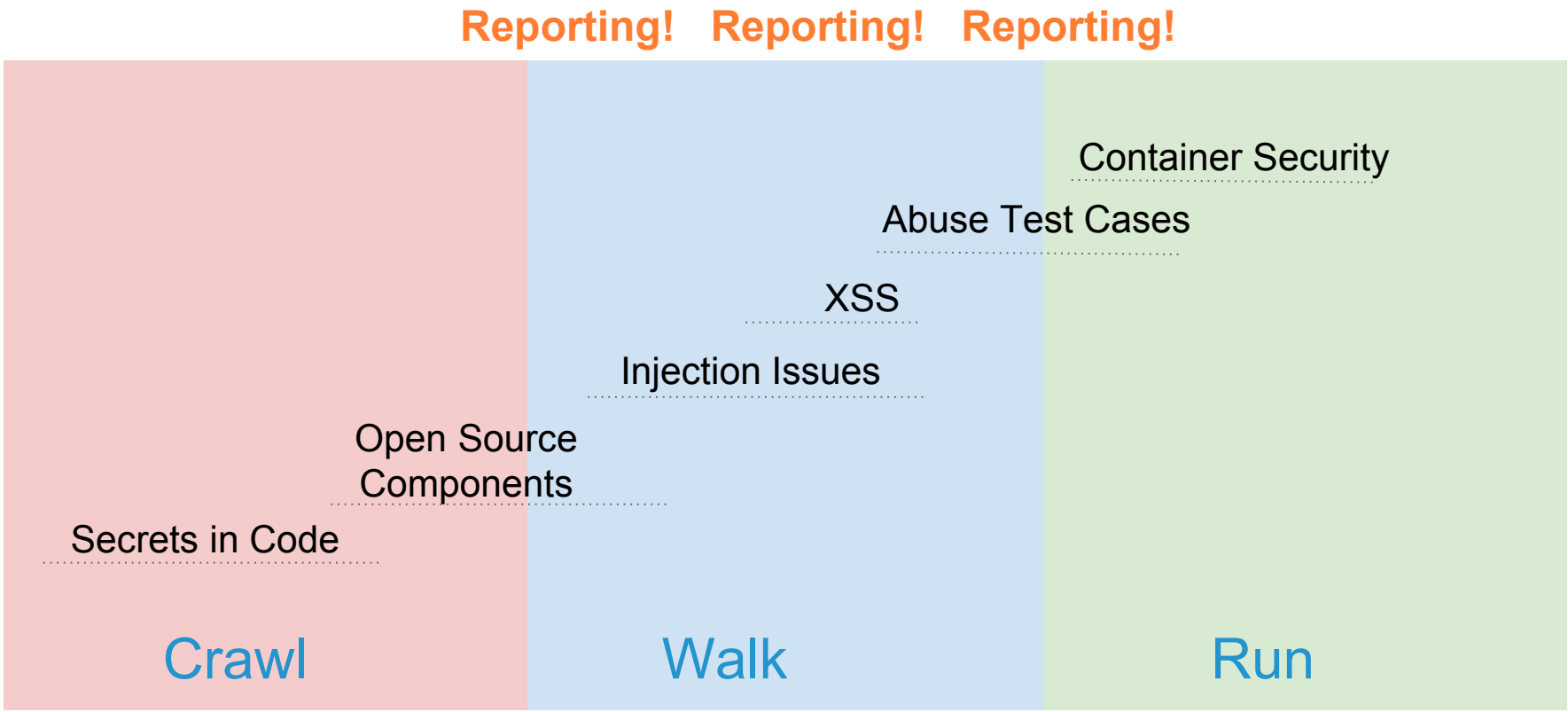
Deploy a “Security Test Stack”

Poor Integration Options

Execution Time

More hand-holding needed

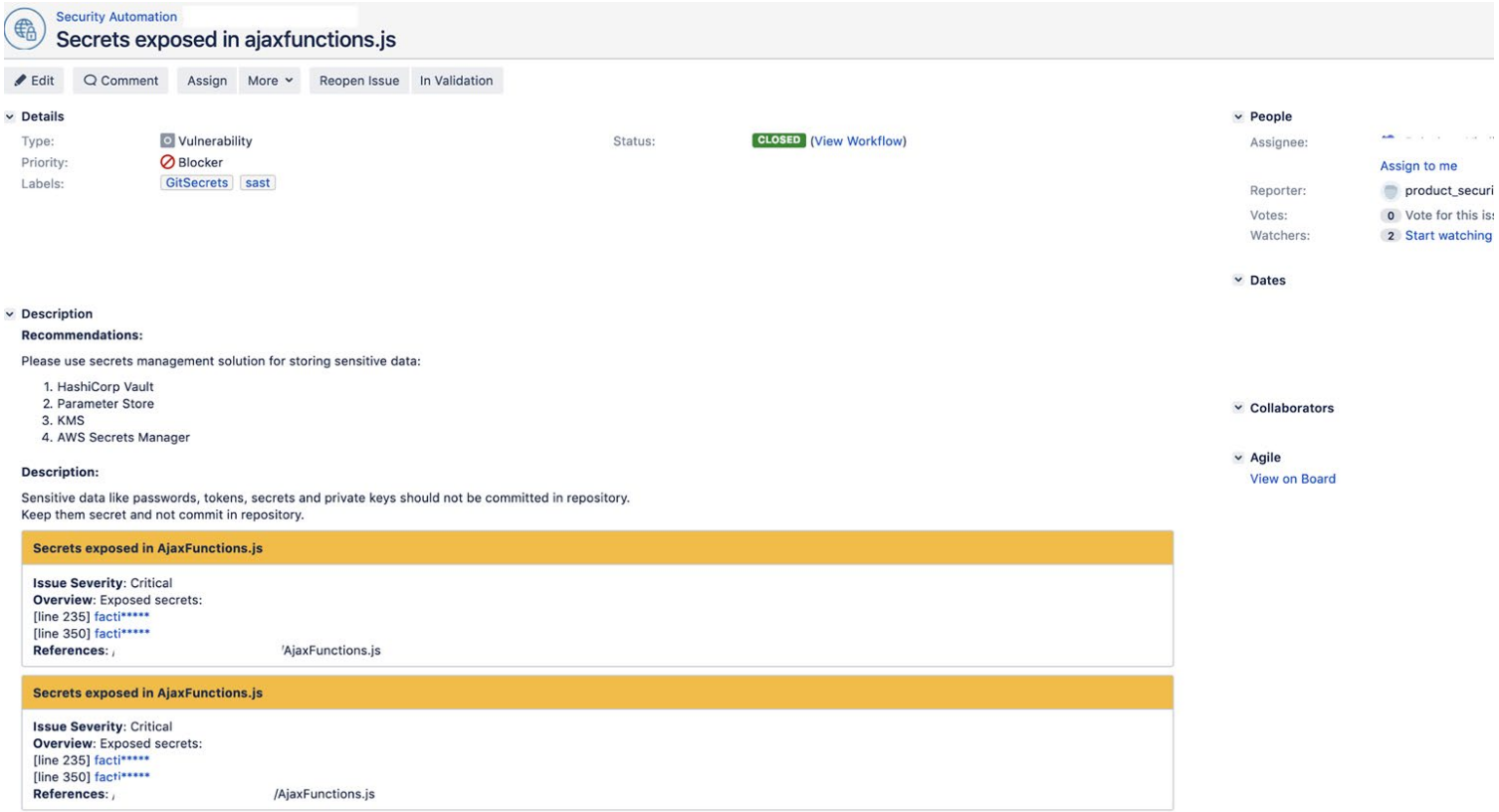
Breaking the build : Myth?



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 : Day 1



Security Automation

Secrets exposed in ajaxfunctions.js

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

Details

Type: Vulnerability
Priority: Blocker
Labels: GitSecrets sast
Status: CLOSED (View Workflow)

People

Assignee:
Assign to me
Reporter: product_securi
Votes: 0 Vote for this is
Watchers: 2 Start watching

Dates

Collaborators

Agile

View on Board

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] facti*****
[line 350] facti*****
References: , 'AjaxFunctions.js

Secrets exposed in AjaxFunctions.js

Issue Severity: Critical
Overview: Exposed secrets:
[line 235] facti*****
[line 350] facti*****
References: , /AjaxFunctions.js

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

Thank you

We're Hiring!

https://dowjones.jobs

Jay Kelath

Pranav Patel

https://github.com/kelath

https://github.com/pranav1688

@kelath

@pranav16