def(dev)eu

Security automation frameworks

General edition

author: Riccardo ten Cate
R1805

Introductionairy

- Security consultant at Xebia
- Hacker, trainer, speaker, security engineer, coder, project leader OWASP S.K.F
- Quirky

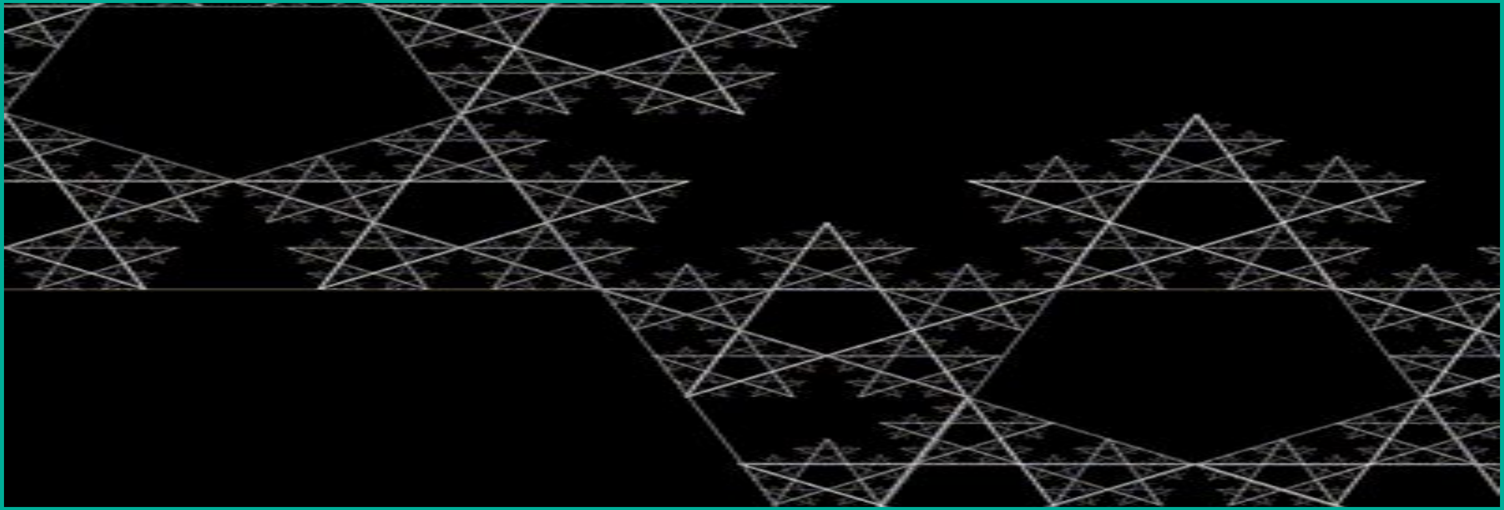
Also me

i'm not a procrastinator
i just prefer doing all my work in a deadline-induced panic
invisiblebread.com

The challenge of automation

Why do DevSecOps/Security automation

- Short feedback loop
- Empower your team
- Make applications secure by design
- Eliminate technical dept
- Regular check-ups



The (S)SDLC

- **Test automation (code quality)**
  - Dead end code
  - Over complex code
  - Repudiated code
- **Security test automation**
  - SAST tooling
  - DAST tooling
- **Manual verification**
  - Security code audit
  - Penetration test

SAST

Static Analyzer Security Tooling

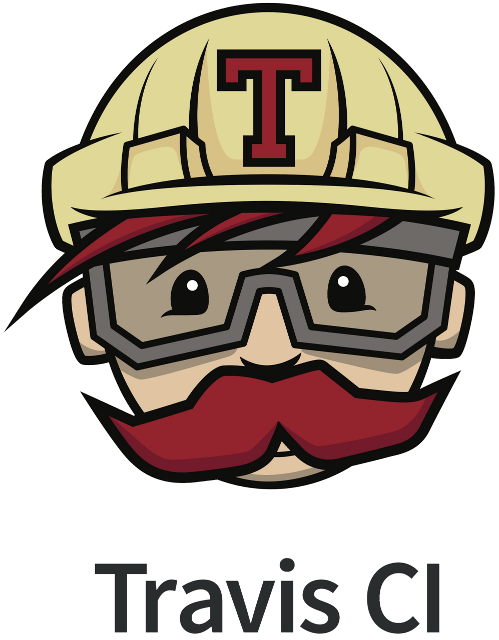
- Fortify
- Veracode
- Checkmarx
- OWASP Dependency checker
- FindBugs
- Snyk/Retire/Node security
- …

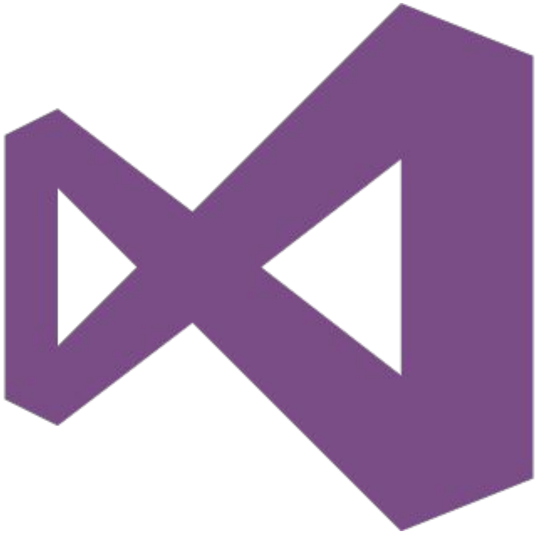
DAST

Dynamic Analyzer Security Tooling

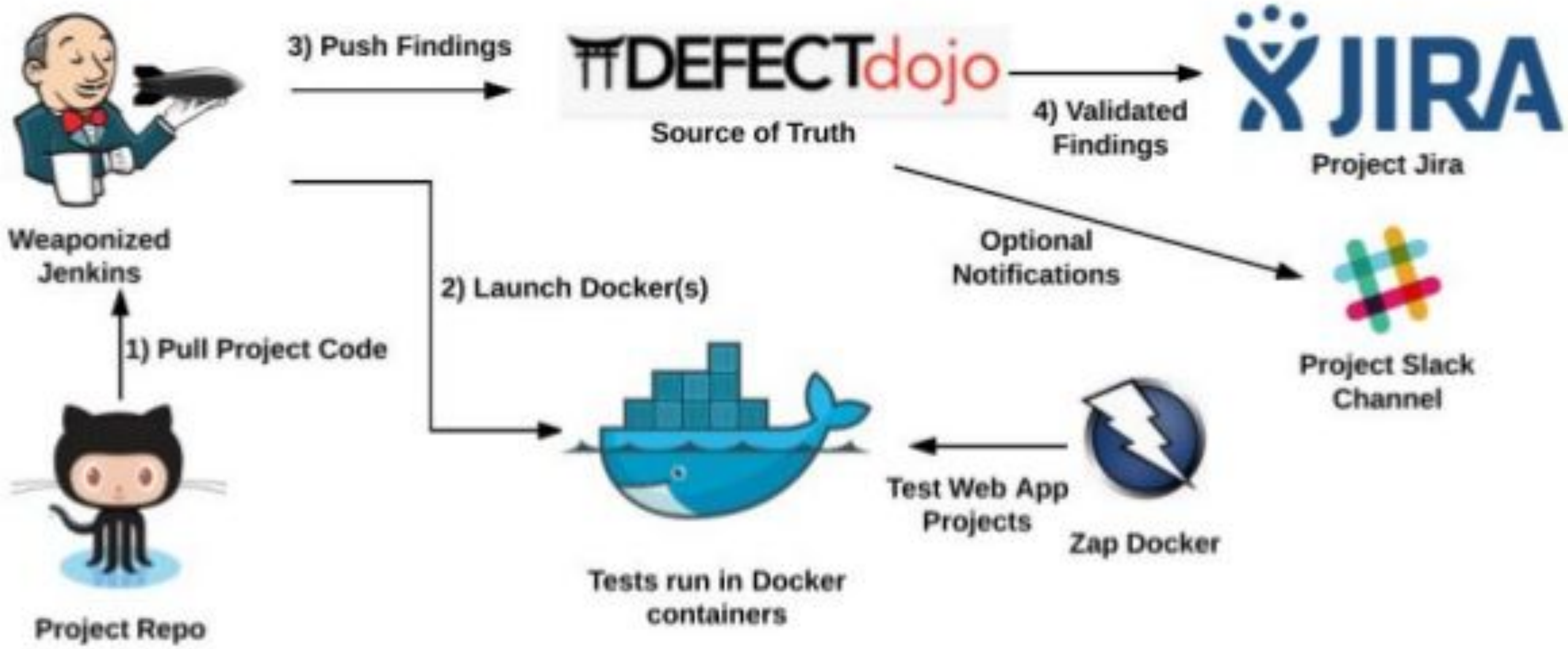
- Acunetix
- OWASP ZAP
- AppSpider
- HP WebInspect
- Burp
- Nessus
- OpenVAS

CI/CD tooling

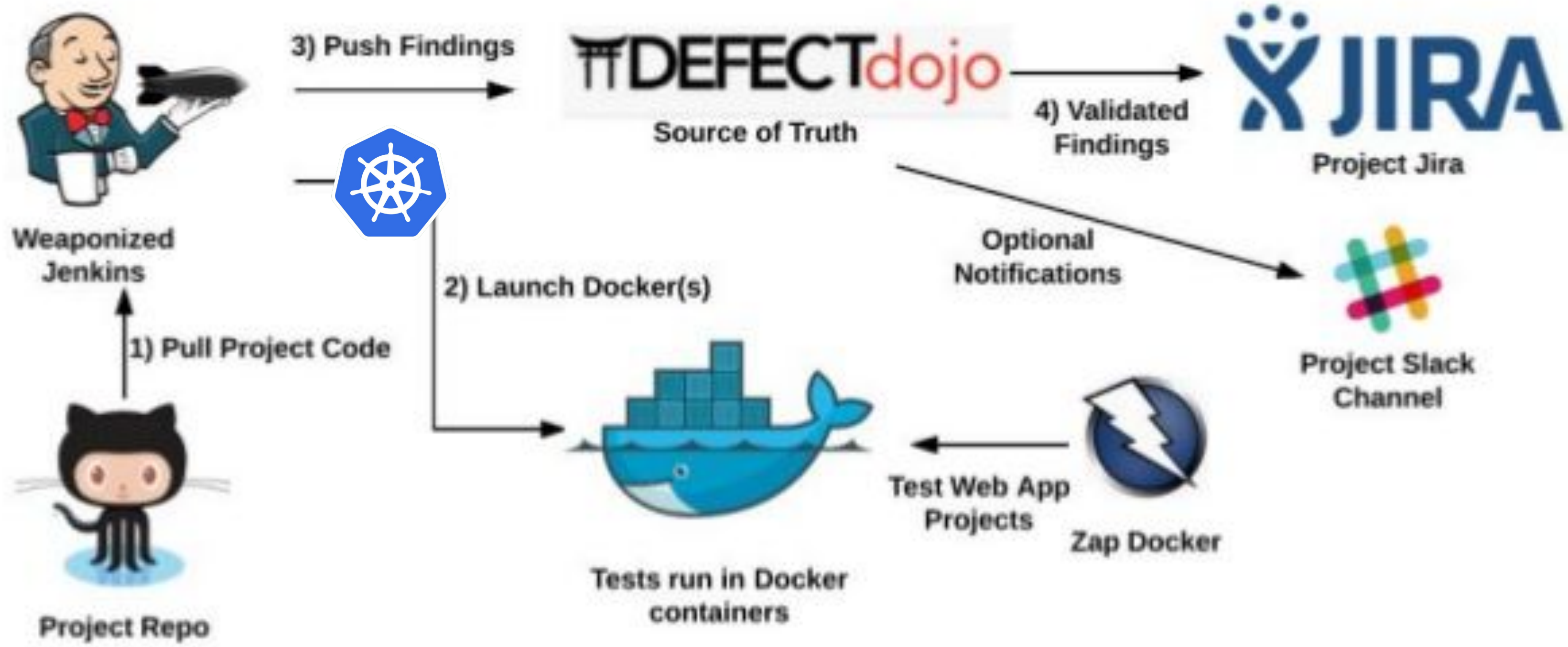




The setup

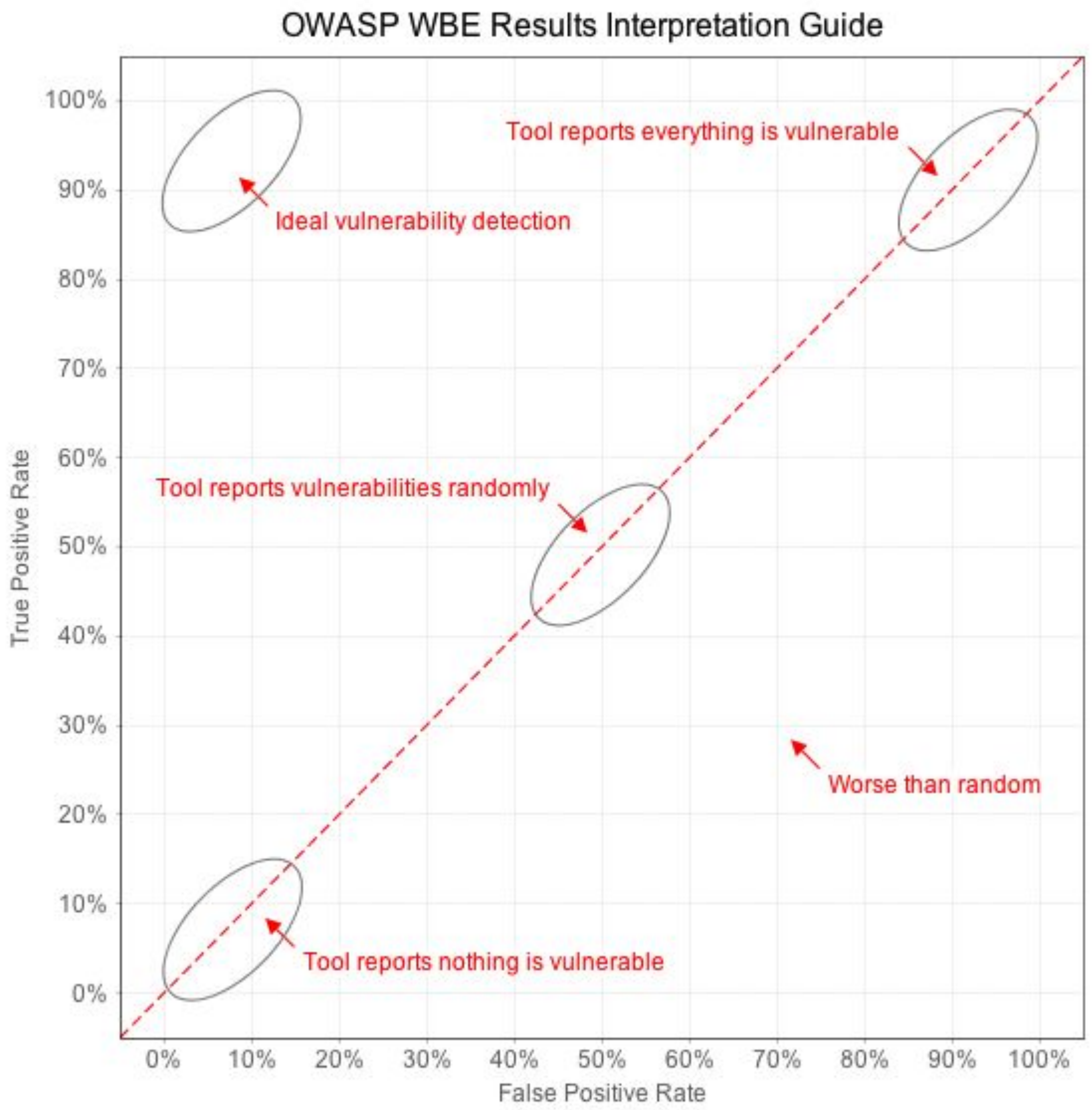

3) Push Findings
DEFECTdojo
Source of Truth
4) Validated Findings
JIRA
Project Jira
Weaponized Jenkins
2) Launch Docker(s)
Optional Notifications
Project Slack Channel
1) Pull Project Code
Test Web App Projects
Zap Docker
Tests run in Docker containers
Project Repo

But ideally!

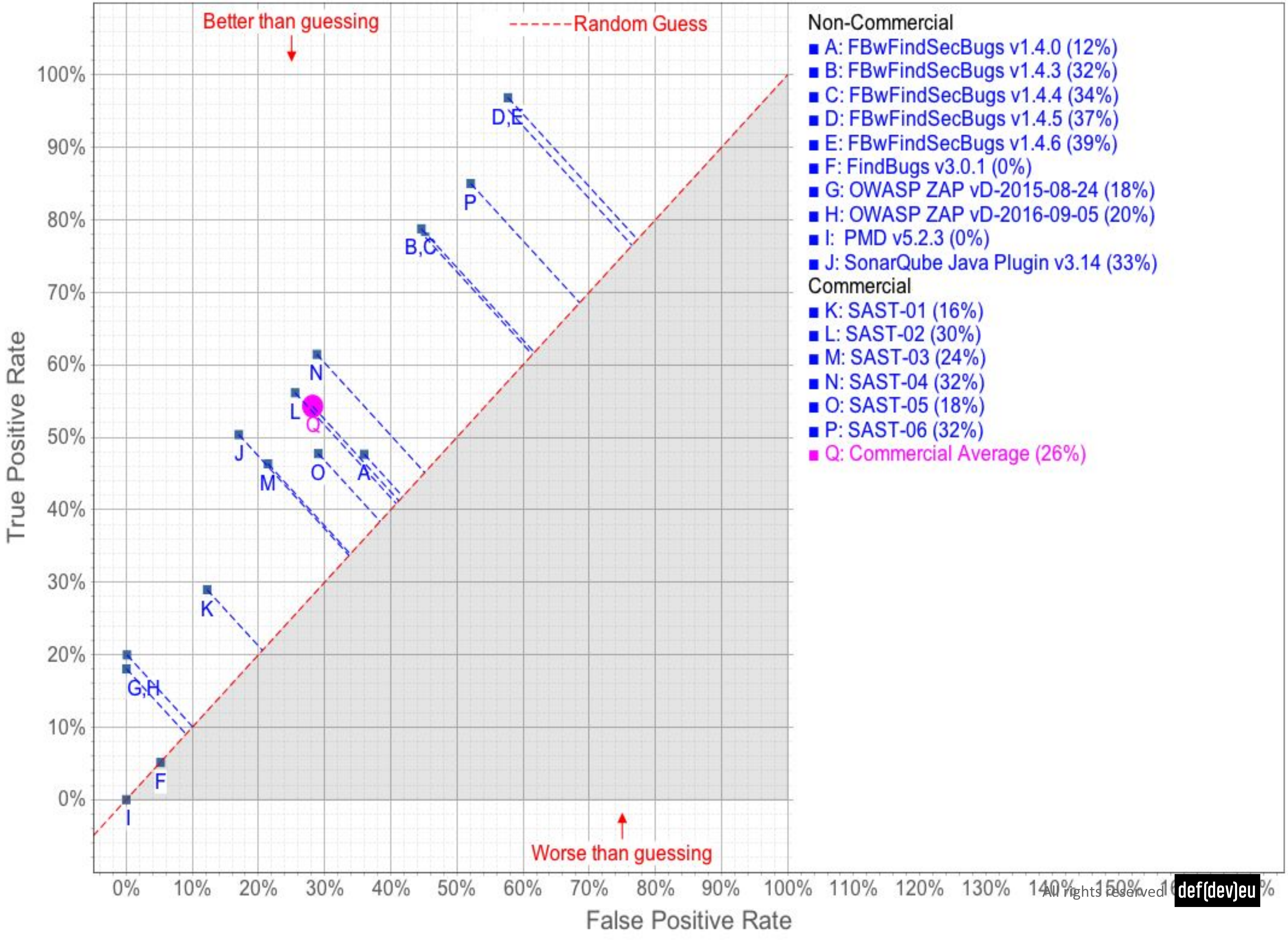

3) Push Findings
DEFECTdojo
Source of Truth
4) Validated Findings
JIRA
Project Jira
Weaponized Jenkins
2) Launch Docker(s)
Optional Notifications
Project Slack Channel
1) Pull Project Code
Test Web App Projects
Zap Docker
Tests run in Docker containers
Project Repo

False positives you say? What is that?

OWASP Benchmark project



OWASP Benchmark Results Comparison

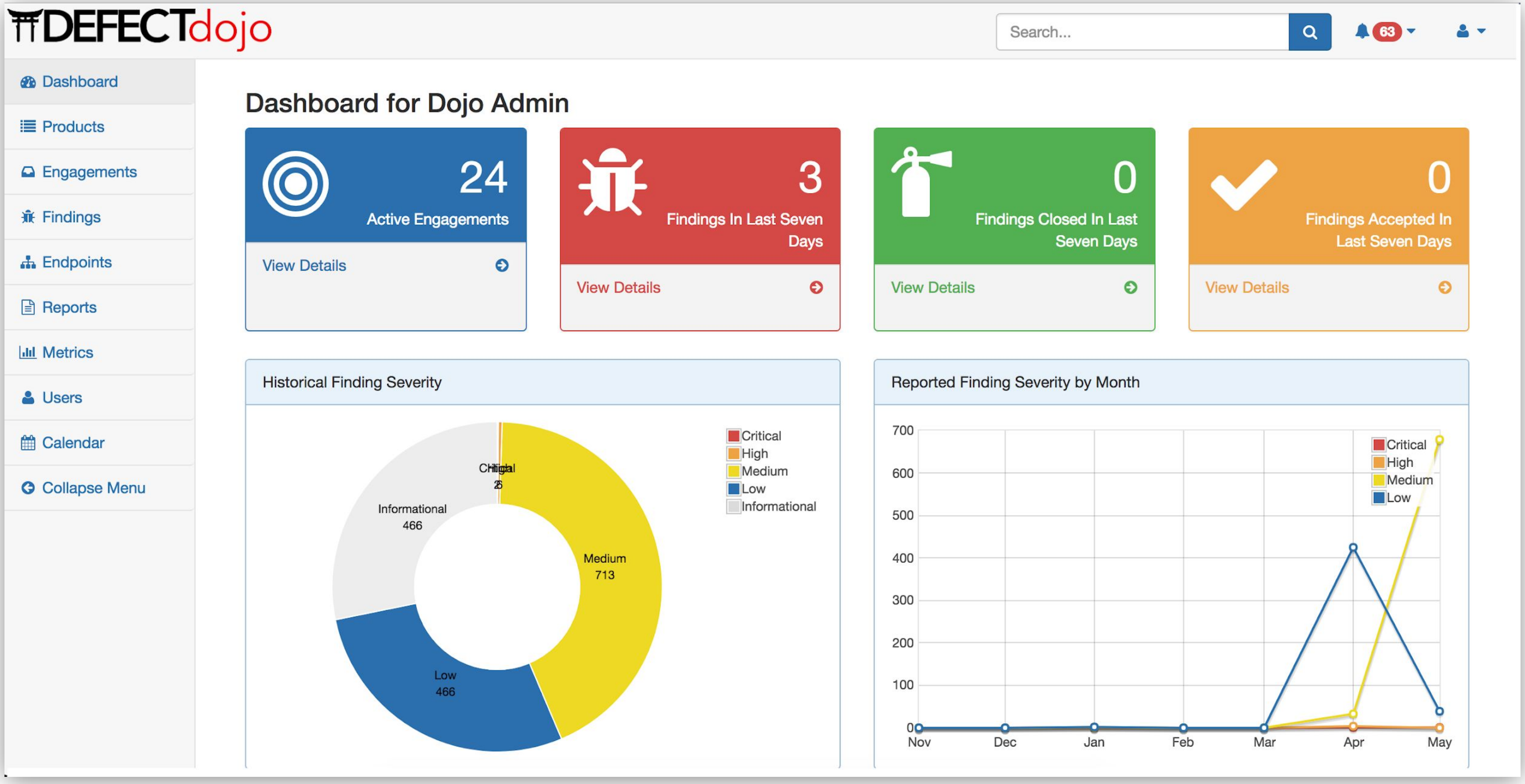


Yeah...

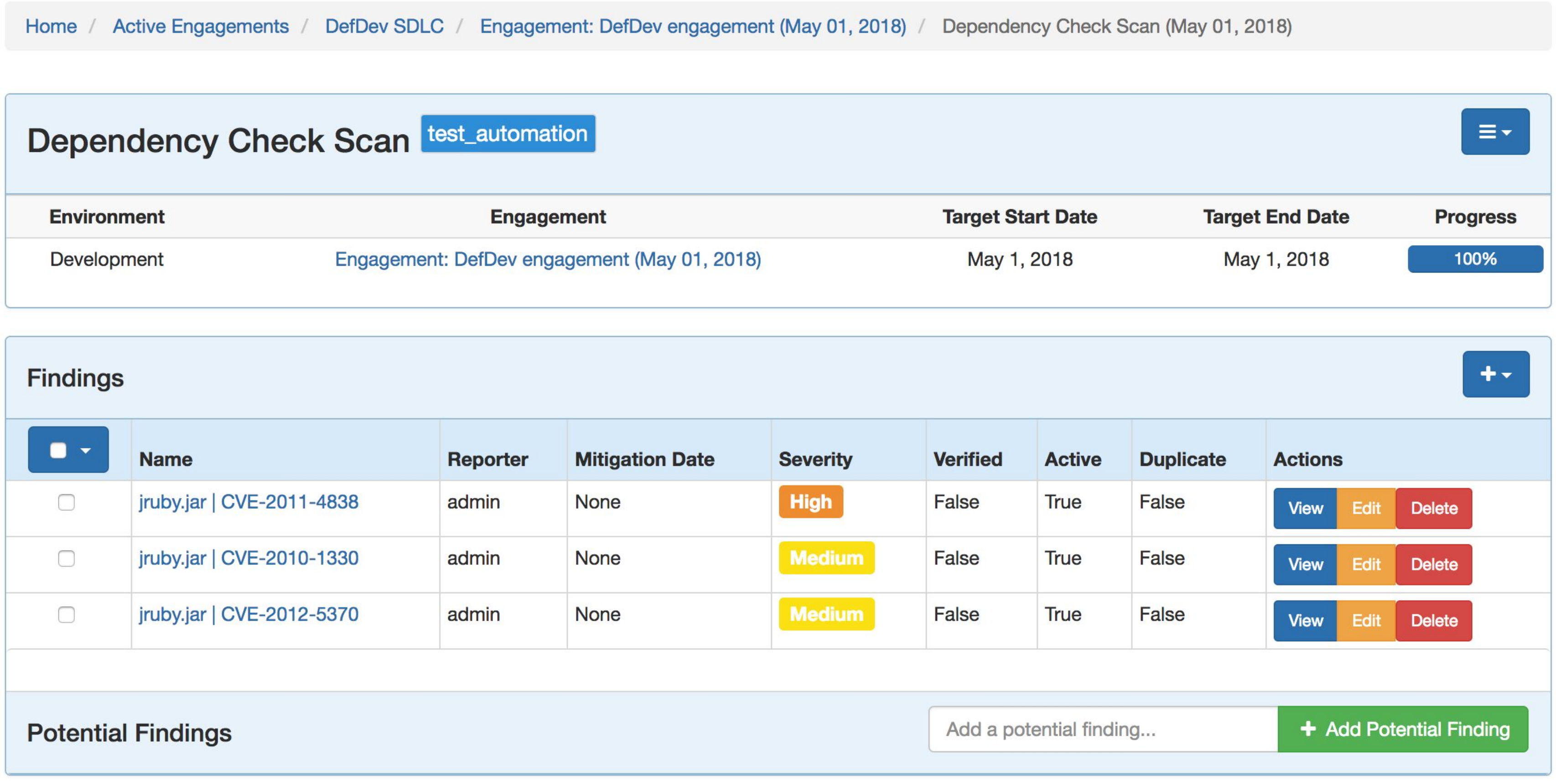
Defect Dojo

- An OWASP project
- Supports a lot of tools
- Easy to deploy
- False positive suppression
- Delta reporting

Well hello!


DEFECTdojo
Search...
63
Dashboard
Products
Engagements
Findings
Endpoints
Reports
Metrics
Users
Calendar
Collapse Menu
Dashboard for Dojo Admin
24
Active Engagements
View Details
3
Findings In Last Seven Days
View Details
0
Findings Closed In Last Seven Days
View Details
0
Findings Accepted In Last Seven Days
View Details
Historical Finding Severity
Critical
High
Medium
Low
Informational
Informational 466
Medium 713
Low 466
Reported Finding Severity by Month
700
600
500
400
300
200
100
0
Nov
Dec
Jan
Feb
Mar
Apr
May
Critical
High
Medium
Low

Show me metrics plz!



Home / Active Engagements / DefDev SDLC / Engagement: DefDev engagement (May 01, 2018) / Dependency Check Scan (May 01, 2018)

Dependency Check Scan test_automation

Environment	Engagement	Target Start Date	Target End Date	Progress
Development	Engagement: DefDev engagement (May 01, 2018)	May 1, 2018	May 1, 2018	100%

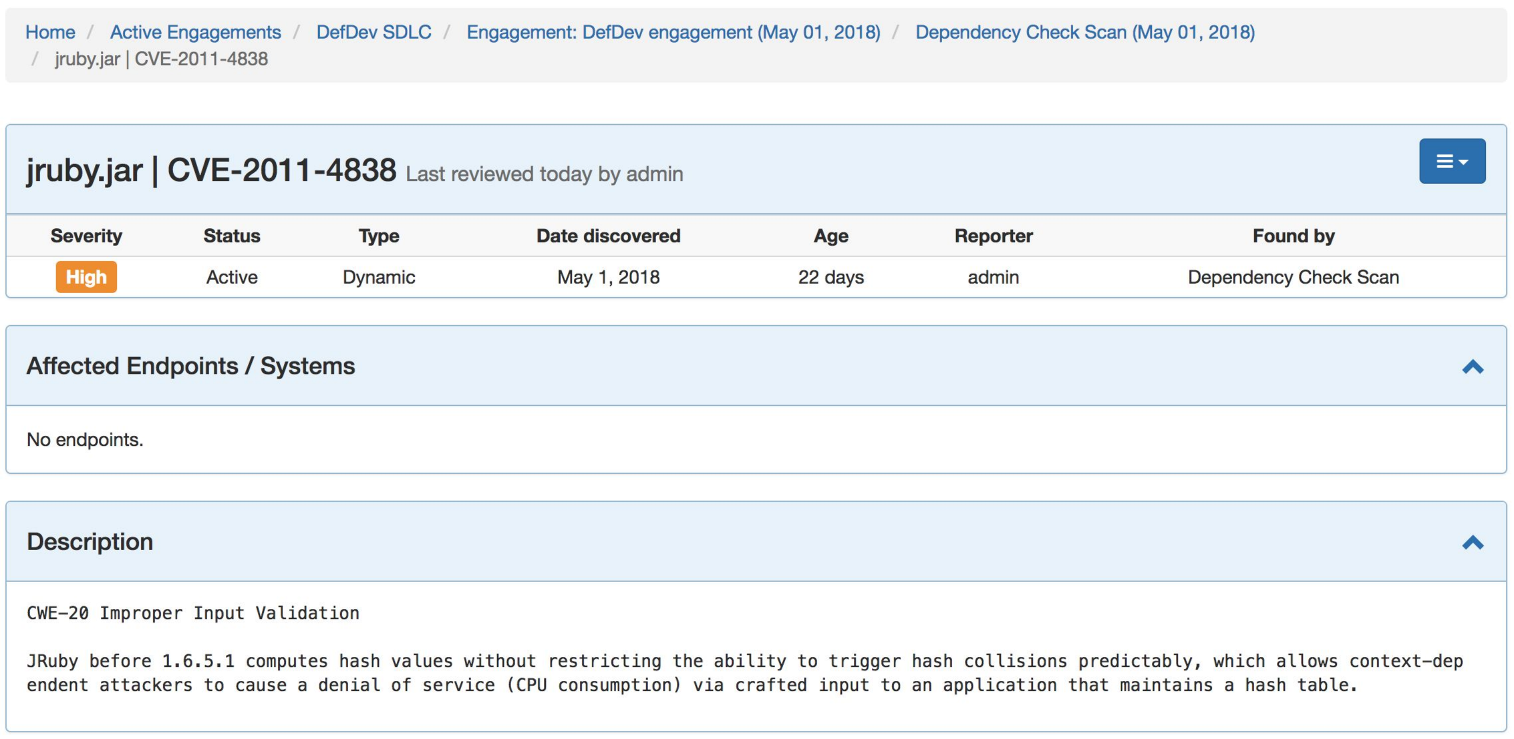
Findings

□	Name	Reporter	Mitigation Date	Severity	Verified	Active	Duplicate	Actions
□	jruby.jar | CVE-2011-4838	admin	None	High	False	True	False	View Edit Delete
□	jruby.jar | CVE-2010-1330	admin	None	Medium	False	True	False	View Edit Delete
□	jruby.jar | CVE-2012-5370	admin	None	Medium	False	True	False	View Edit Delete

Potential Findings

Add a potential finding...

Detailed information!



Home / Active Engagements / DefDev SDLC / Engagement: DefDev engagement (May 01, 2018) / Dependency Check Scan (May 01, 2018) / jruby.jar | CVE-2011-4838

jruby.jar | CVE-2011-4838 Last reviewed today by admin

Severity	Status	Type	Date discovered	Age	Reporter	Found by
High	Active	Dynamic	May 1, 2018	22 days	admin	Dependency Check Scan

Affected Endpoints / Systems

No endpoints.

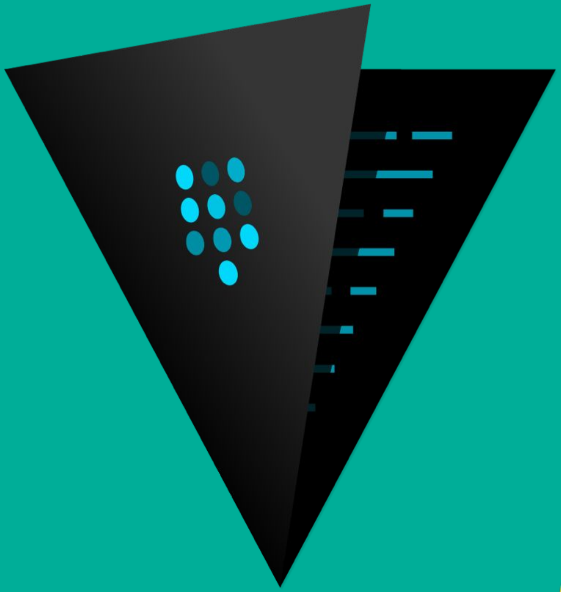
Description

CWE-20 Improper Input Validation

JRuby before 1.6.5.1 computes hash values without restricting the ability to trigger hash collisions predictably, which allows context-dependent attackers to cause a denial of service (CPU consumption) via crafted input to an application that maintains a hash table.

Insert random gif

Please handle yer secrets and keys!



Problem #1 Secret sprawl

- Plaintext
- Config files
  - Ansible, Puppet, Chef
  - Dockerfile / Entrypoint
- Source code
- Version control systems
  - Github
  - VSTS
  - Bitbucket

Look at all this new attack surface!

Soooooooo…..

- If the keys are sprawled!
- We have no auditability
- We have no good means to revoke keys
- We have applications who suck at keeping secrets!
  - Logs (splunk, syslog)
  - Error handling
  - Monitoring
- We need a Vault in our lives!

Vault can fix all of this and more!

Vault!

- Dynamic secrets
  - Ephemeral
  - So we have key rotation
    - Should an application log creds they are no longer valid
- Unique tokens
  - We now have auditability
  - We can revoke keys that were proven to be compromised
- All the information is encrypted
  - In rest
  - In transit
- This just scratches the surface of what it could do!

No more talking the talk

Hope yall had a good time!