ADELAIDE2019

#DDDADELAIDE / @DDDADELAIDE

SUPPORT OUR SPONSORS!

Telstra Purple

University of South Australia

aws

Comunet

THANKS TO OUR COMMUNITY PARTNER

YOW!

ADELAIDE2019

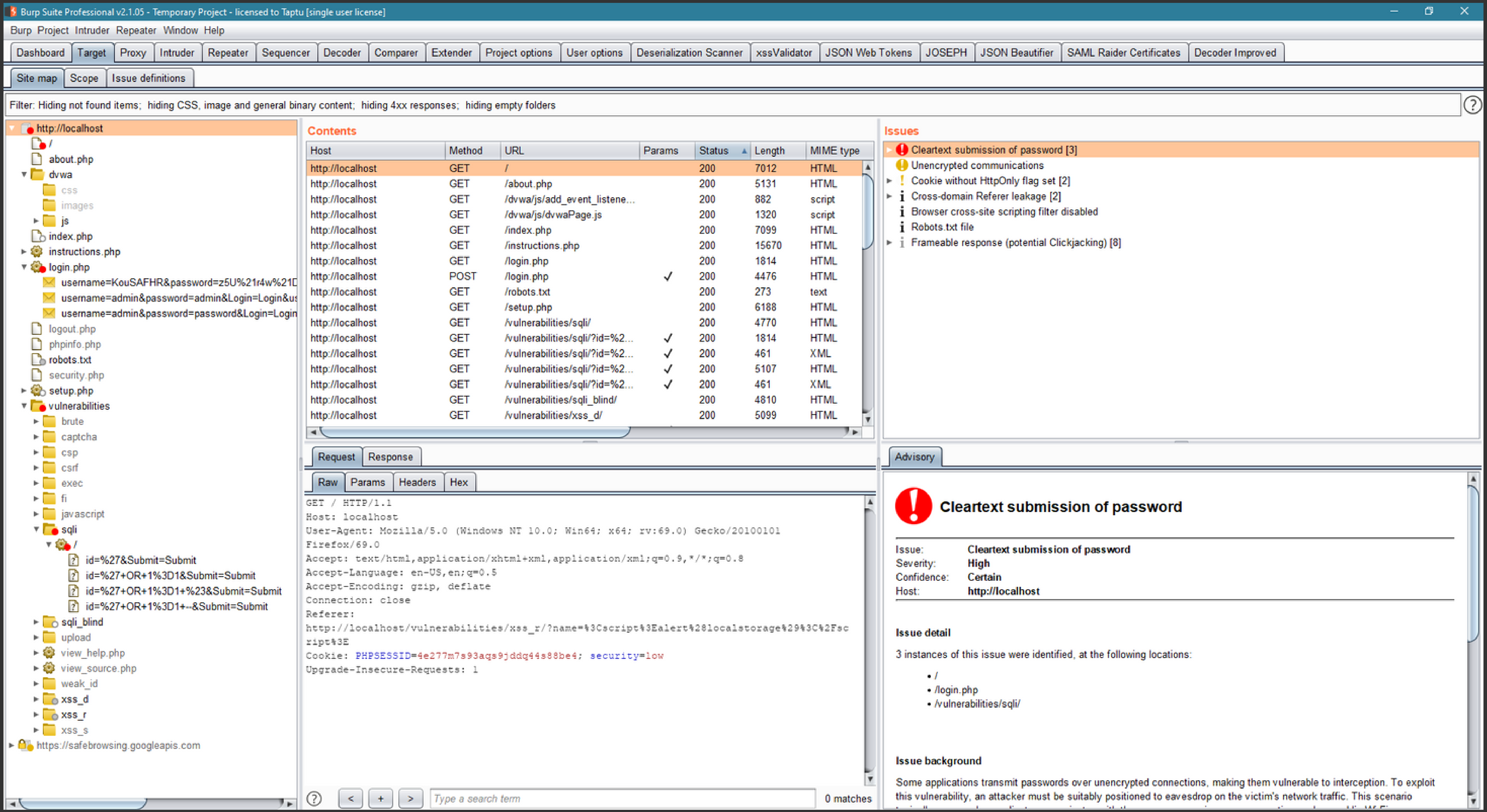
Shifting Left

DevSecOps as an Approach to Building Secure Products

Jakob Pennington
@JakobTheDev

My path to AppSec

ace


Burp Suite Professional v2.1.05 - Temporary Project - licensed to Taptu [single user license]
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Deserialization Scanner xssValidator JSON Web Tokens JOSEPH JSON Beautifier SAML Raider Certificates Decoder Improved
Site map Scope Issue definitions

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

```
http://localhost
  /
  about.php
  dvwa
    css
    images
    js
  index.php
  instructions.php
  login.php
    username=KouSAFHR&password=z5U%21r4w%21D
    username=admin&password=admin&Login=Login&us
    username=admin&password=password&Login=Login
  logout.php
  phpinfo.php
  robots.txt
  security.php
  setup.php
  vulnerabilities
    brute
    captcha
    csp
    csrf
    exec
    fi
    javascript
    sqli
      /
        id=%27&Submit=Submit
        id=%27+OR+1%3D1&Submit=Submit
        id=%27+OR+1%3D1+%23&Submit=Submit
        id=%27+OR+1%3D1+--&Submit=Submit
    sqli_blind
    upload
    view_help.php
    view_source.php
    weak_id
    xss_d
    xss_r
    xss_s
https://safebrowsing.googleapis.com
```

Contents

Host	Method	URL	Params	Status	Length	MIME type
http://localhost	GET	/		200	7012	HTML
http://localhost	GET	/about.php		200	5131	HTML
http://localhost	GET	/dvwa/js/add_event_listene...		200	882	script
http://localhost	GET	/dvwa/js/dvwaPage.js		200	1320	script
http://localhost	GET	/index.php		200	7099	HTML
http://localhost	GET	/instructions.php		200	15670	HTML
http://localhost	GET	/login.php		200	1814	HTML
http://localhost	POST	/login.php	✓	200	4476	HTML
http://localhost	GET	/robots.txt		200	273	text
http://localhost	GET	/setup.php		200	6188	HTML
http://localhost	GET	/vulnerabilities/sqli/		200	4770	HTML
http://localhost	GET	/vulnerabilities/sqli/?id=%2...	✓	200	1814	HTML
http://localhost	GET	/vulnerabilities/sqli/?id=%2...	✓	200	461	XML
http://localhost	GET	/vulnerabilities/sqli/?id=%2...	✓	200	5107	HTML
http://localhost	GET	/vulnerabilities/sqli/?id=%2...	✓	200	461	XML
http://localhost	GET	/vulnerabilities/sqli_blind/		200	4810	HTML
http://localhost	GET	/vulnerabilities/xss_d/		200	5099	HTML

Issues

- Cleartext submission of password [3]
- Unencrypted communications
- Cookie without HttpOnly flag set [2]
- Cross-domain Referer leakage [2]
- Browser cross-site scripting filter disabled
- Robots.txt file
- Frameable response (potential Clickjacking) [8]

```
GET / HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101
Firefox/69.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer:
http://localhost/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28localstorage%29%3C%2Fsc
ript%3E
Cookie: PHPSESSID=4e277m7s93aqs9jddq44s88be4; security=low
Upgrade-Insecure-Requests: 1
```

Cleartext submission of password

Issue:	Cleartext submission of password
Severity:	High
Confidence:	Certain
Host:	http://localhost

Issue detail

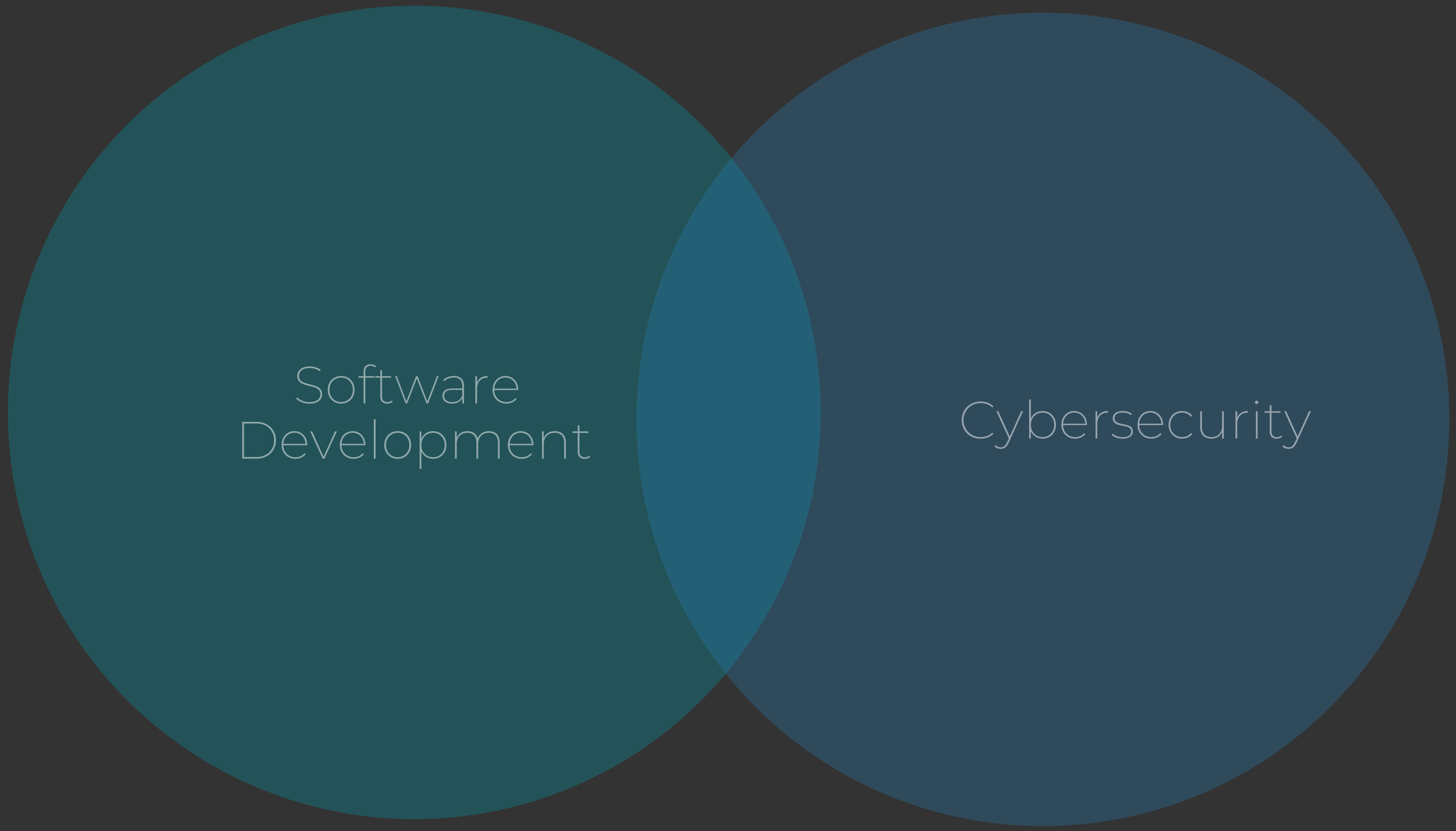
3 instances of this issue were identified, at the following locations:

- /
- /login.php
- /vulnerabilities/sqli/

Issue background

Some applications transmit passwords over unencrypted connections, making them vulnerable to interception. To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario

Put me in the middle



MADE WITH
beautiful.ai

Application Security

So what's the problem?

Data Breach

Fines

GDPR

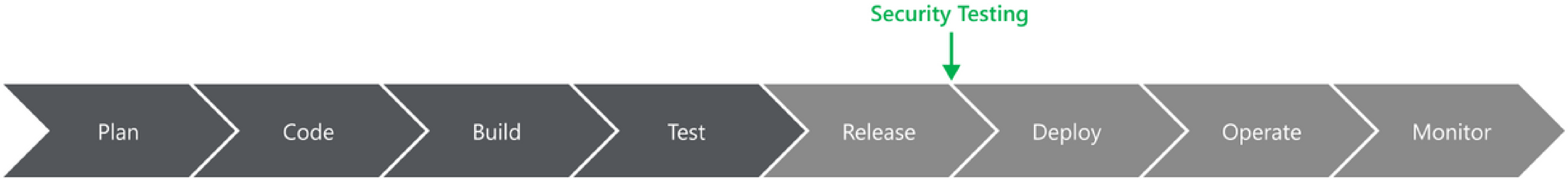
Privacy

Reputational damage

Australian Privacy Principles

Media coverage

Traditional Security Testing



100 / 10 / 1

Enter DevOps

To summarise...

- Application security is a complex problem
- Security testing is too late in the lifecycle
- Not enough security specialists
- Security can't keep up with DevOps

What's next for Application Security?

Dev Sec Ops

DevSecOps is doing
for application security what
DevOps did for
development and operations

WHY

- Defects are cheaper to fix early
- Fewer delays at release time
- Repeatable security at scale
- More secure result

Three aspects of DevSecOps

Culture

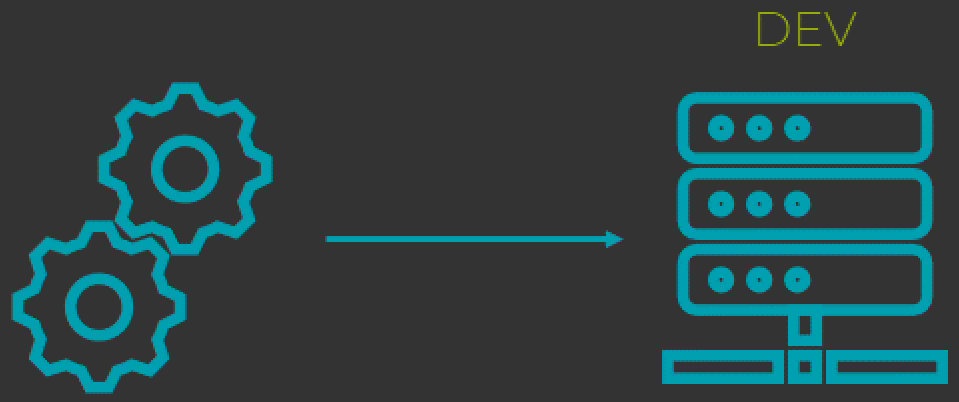
Process

Tooling

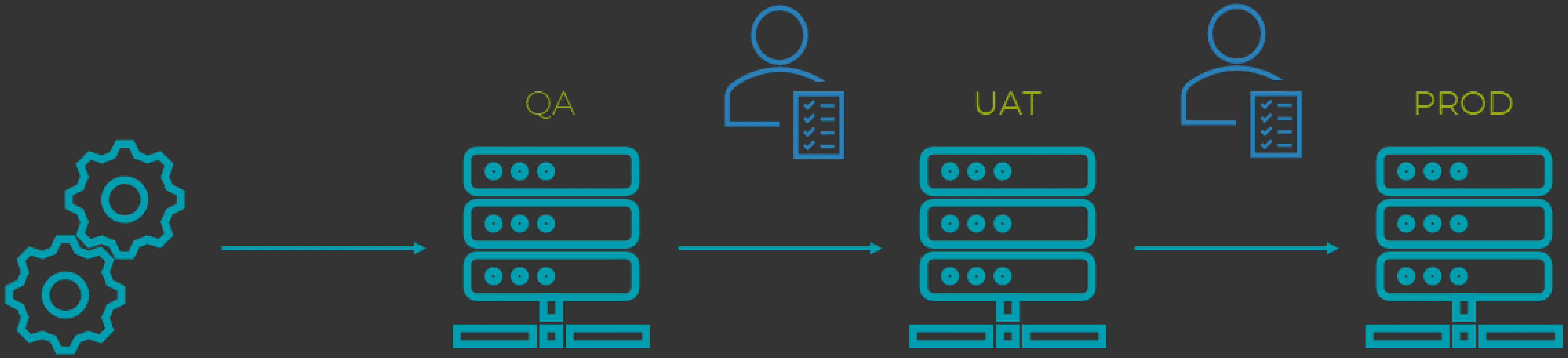
Tooling

DevOps took operations tooling and made it developer centric

Continuous Integration

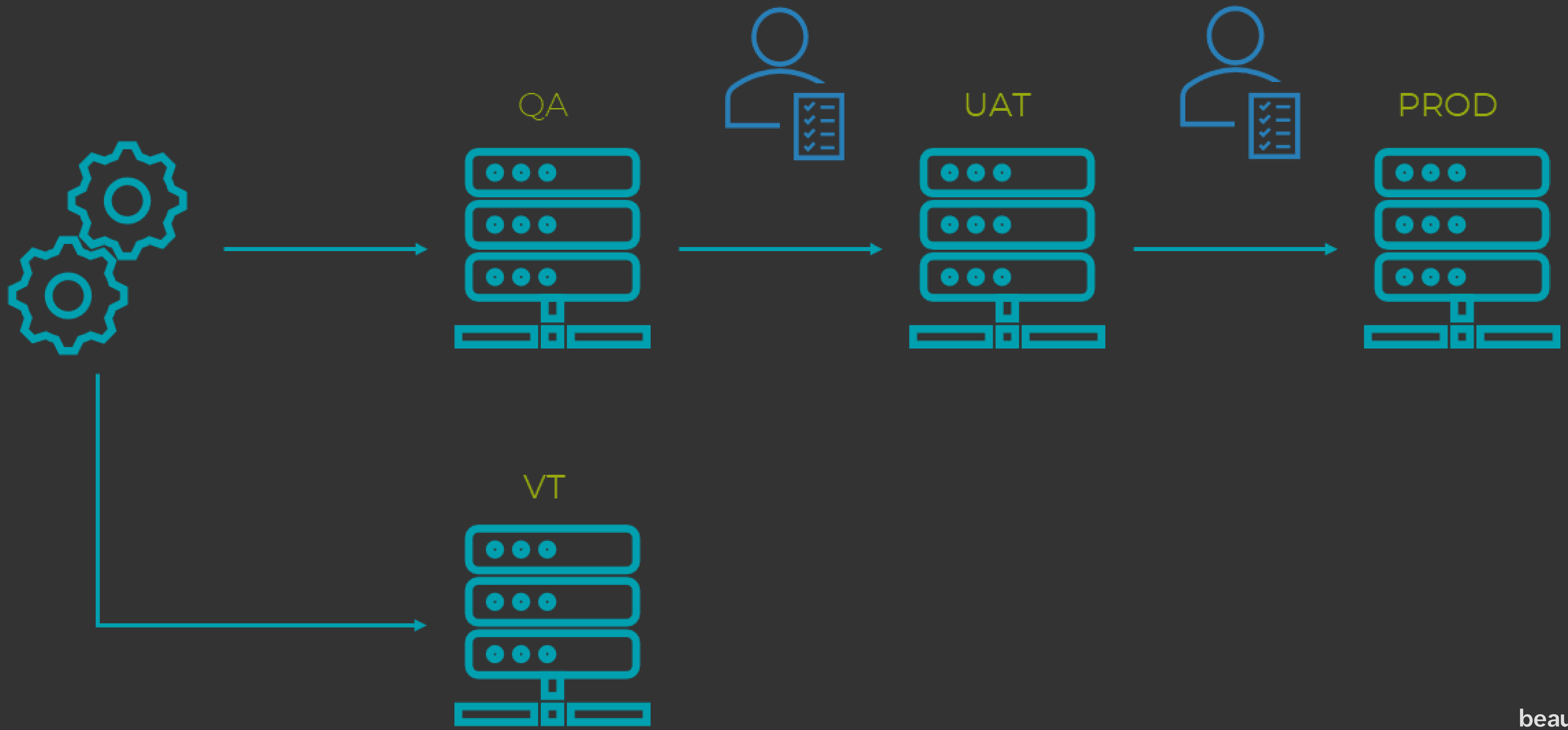


Continuous Delivery

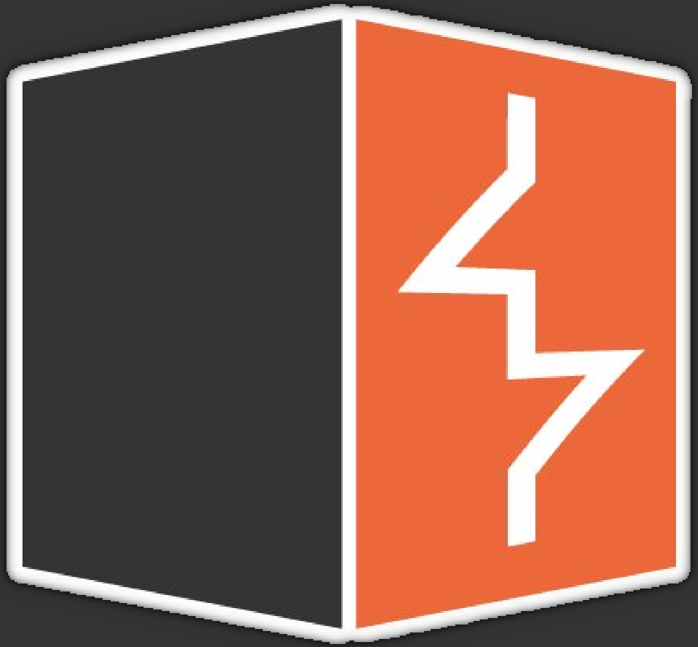


Think like a hacker
Automate like a pentester

CD with Out of Band Validation Testing



Application Scanning



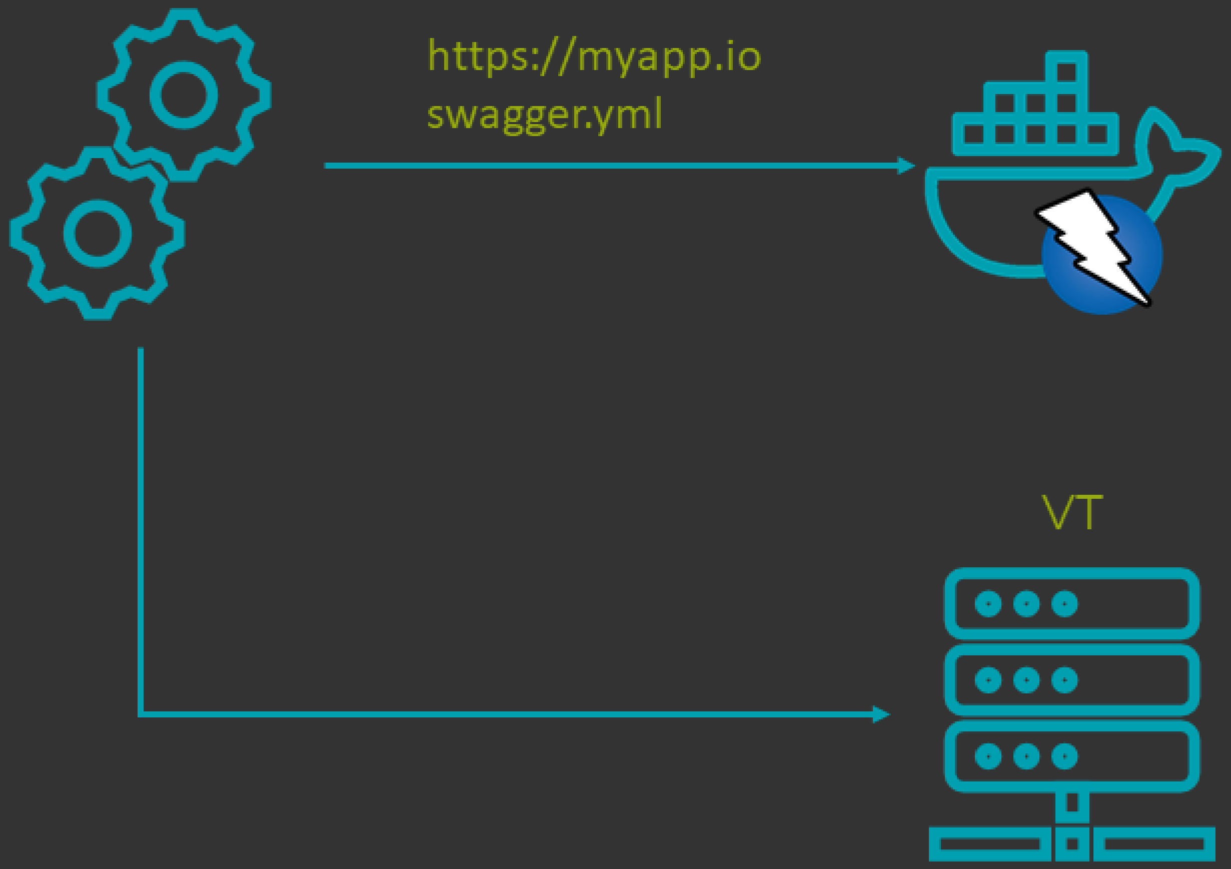
BurpSuite

$$ ENTERPRISE EDITION $$

OWASP ZAP

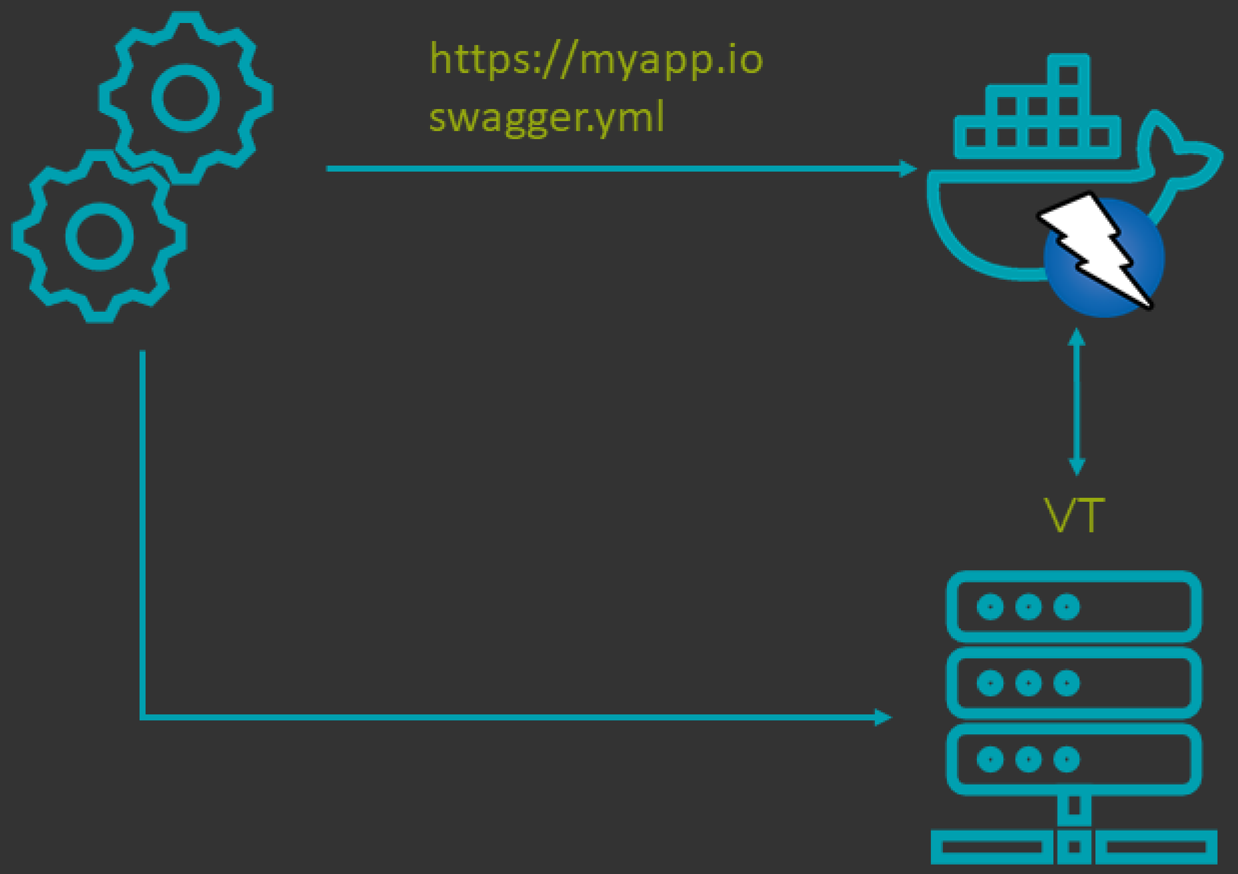
FREE

VT



https://myapp.io
swagger.yml

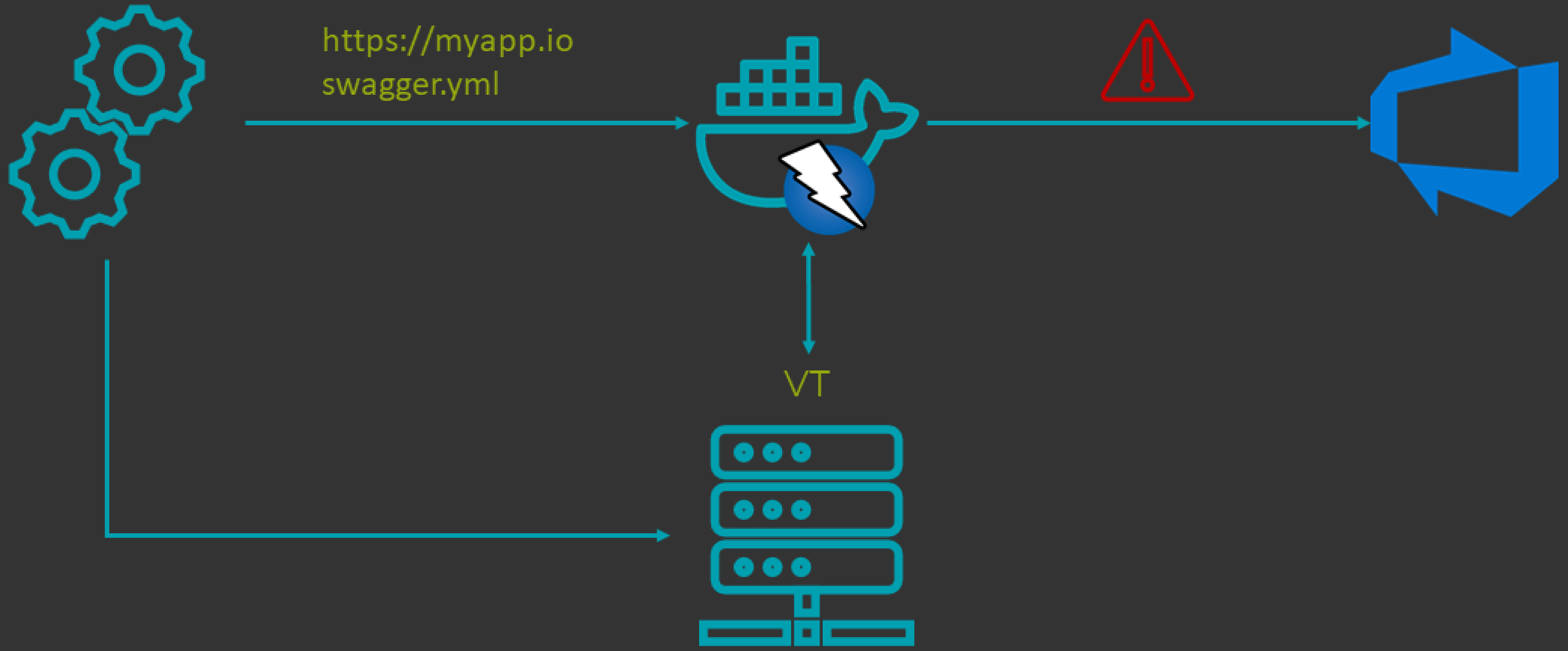
VT

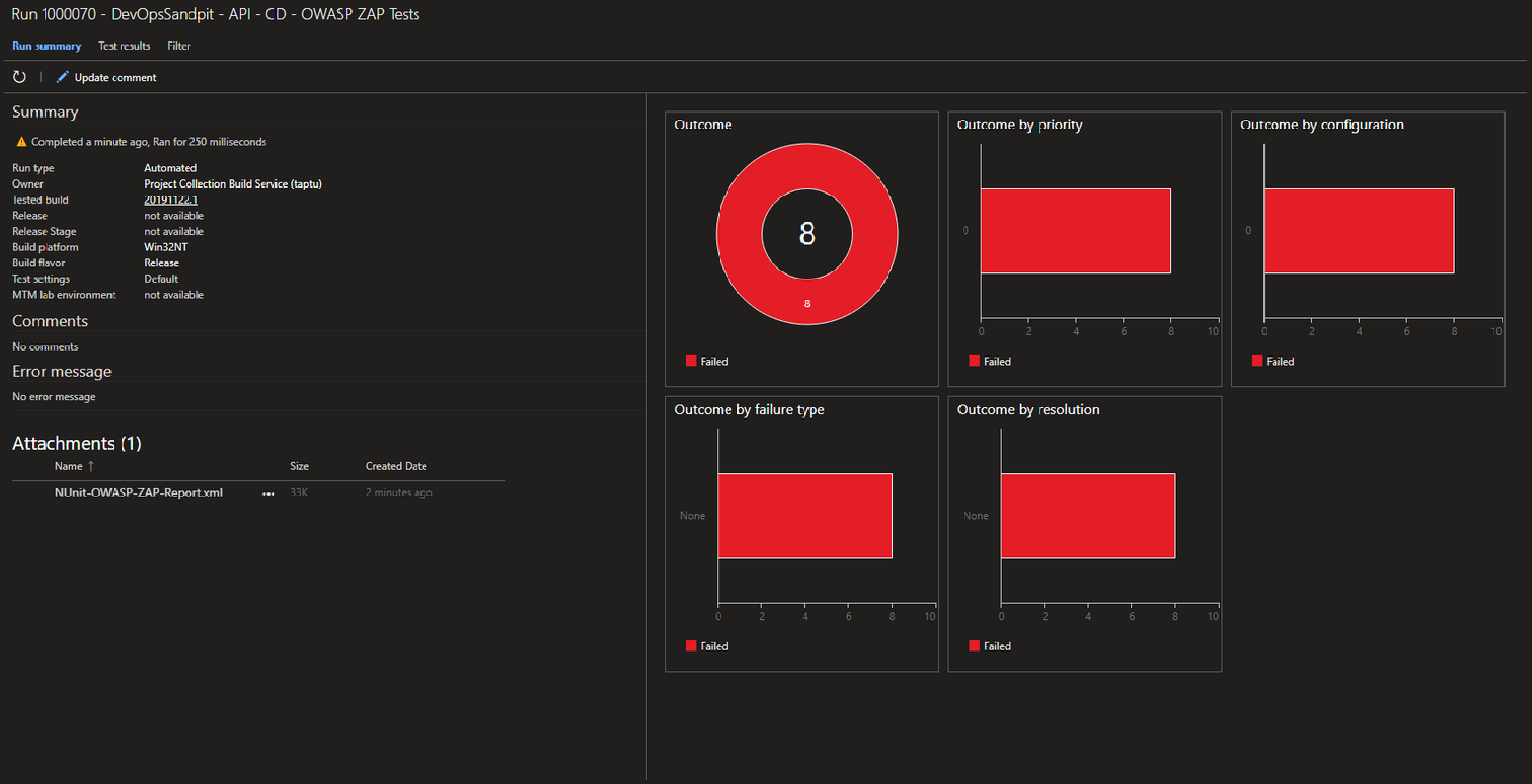


https://myapp.io

swagger.yml

VT


https://myapp.io
swagger.yml
VT



Run 1000070 - DevOpsSandpit - API - CD - OWASP ZAP Tests

Summary

⚠ Completed a minute ago, Ran for 250 milliseconds

Run type	Automated
Owner	Project Collection Build Service (taptu)
Tested build	20191122.1
Release	not available
Release Stage	not available
Build platform	Win32NT
Build flavor	Release
Test settings	Default
MTM lab environment	not available

Comments

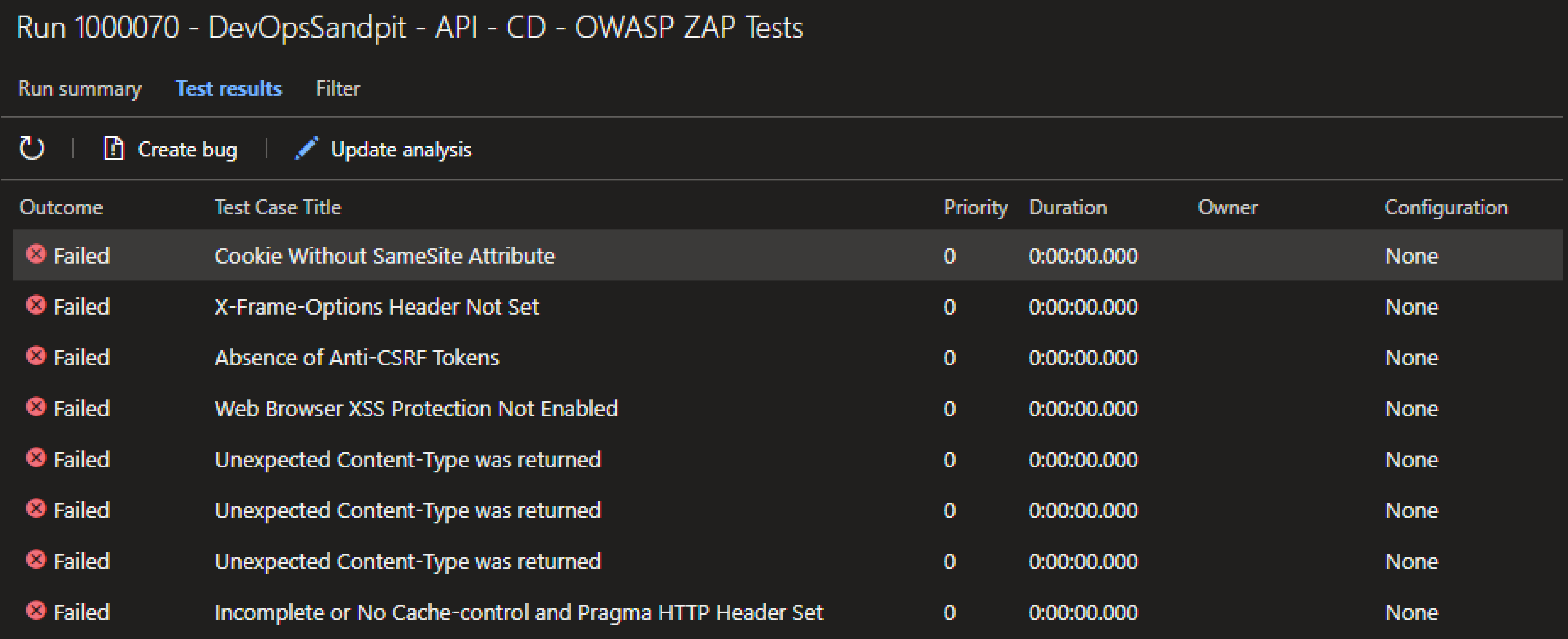
No comments

Error message

No error message

Attachments (1)

Name ↑	Size	Created Date
NUnit-OWASP-ZAP-Report.xml •••	33K	2 minutes ago



Run 1000070 - DevOpsSandpit - API - CD - OWASP ZAP Tests

Outcome	Test Case Title	Priority	Duration	Owner	Configuration
⊗ Failed	Cookie Without SameSite Attribute	0	0:00:00.000		None
⊗ Failed	X-Frame-Options Header Not Set	0	0:00:00.000		None
⊗ Failed	Absence of Anti-CSRF Tokens	0	0:00:00.000		None
⊗ Failed	Web Browser XSS Protection Not Enabled	0	0:00:00.000		None
⊗ Failed	Unexpected Content-Type was returned	0	0:00:00.000		None
⊗ Failed	Unexpected Content-Type was returned	0	0:00:00.000		None
⊗ Failed	Unexpected Content-Type was returned	0	0:00:00.000		None
⊗ Failed	Incomplete or No Cache-control and Pragma HTTP Header Set	0	0:00:00.000		None

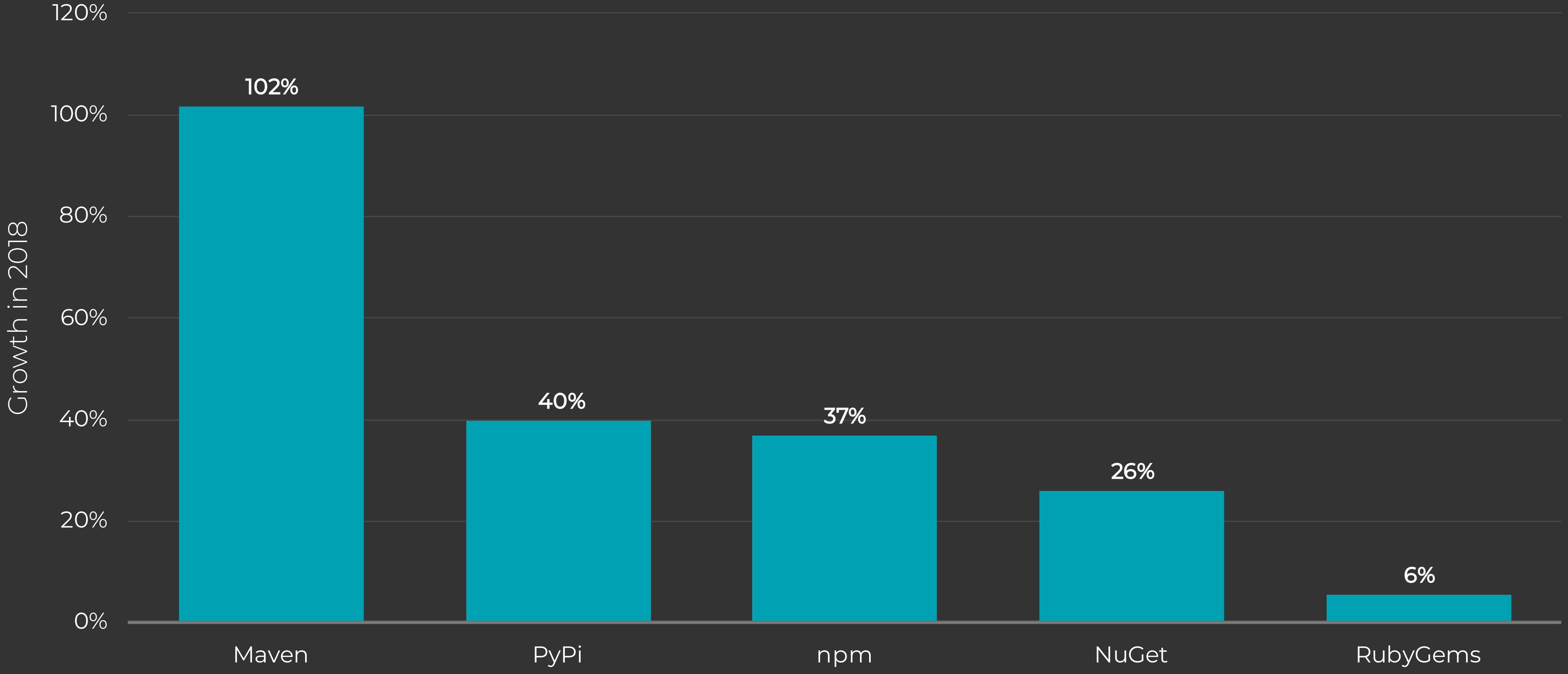
Find things like…

- SQL Injection
- Cross-Site Scripting
- Cross Site Request Forgery
- Path Traversal
- OS Command Injection
- XML External Entity Injection
- SSL / TLS Issues
- Cookie attributes
- Web Cache Poisoning
- Cacheable HTTPS Response
- Cross-Origin Resource Sharing
- …

Developers can test things that pentesters can't

Dependency Management

Open source ecosystem is exploding



Source: State of Open Source Security 2019 - Snyk

A tale of two threats

Vulnerable Code

Supply chain attacks

Snyk

Dependabot

Dependency management four ways

Audit packages list

CI Integration

Repo monitoring

Automatic pull requests

Want to test all the things?

Secure DevOps Toolchain

Pre-Commit

Security activities before code is checked in to version control

Threat Modeling/Attack Mapping:
- Attacker personas
- Evil user stories
- Raindance
- Mozilla Rapid Risk Assessment
- OWASP ThreatDragon

Security and Privacy Stories:
- OWASP ASVS
- SAFECode Security Stories

IDE Security Plugins:
- DevSkim
- FindSecurityBugs
- Puma Scan
- SonarLint

Pre-Commit Security Hooks:
- git-hound
- git-secrets
- Repo-supervisor
- ThoughtWorks Talisman

Secure Coding Standards:
- CERT Secure Coding Standards
- OWASP Proactive Controls

Manual and Peer Reviews:
- Gerrit
- GitHub pull request
- GitLab merge request
- Review Board

Commit (Continuous Integration)

Fast, automated security checks during the build and Continuous Integration steps

Static Code Analysis (SCA):
- FindSecurityBugs
- Brakeman
- ESLint
- Phan

Security Unit Tests:
- JUnit
- Mocha
- xUnit

Infrastructure as Code Analysis:
- ansible-lint
- Foodcritic
- puppet-lint
- cfn_nag

Dependency Management:
- OWASP Dependency Check
- Bundler-Audit
- Gemnasium
- PHP Security Checker
- Retire.JS
- Node Security Platform

Container Security:
- Actuary
- Anchore
- Clair
- Dagda
- Docker Bench
- Falco

Container Hardening:
- Bane
- CIS Benchmarks
- grsecurity

Acceptance (Continuous Delivery)

Automated security acceptance, functional testing, and deep out-of-band scanning during Continuous Delivery

Infrastructure as Code:
- Ansible
- Chef
- Puppet
- SaltStack
- Terraform
- Vagrant

Immutable Infrastructure:
- Docker
- rkt

Security Scanning:
- Arachni
- nmap
- sqlmap
- sslyze
- ZAP
- ssh_scan

Cloud Configuration Management:
- AWS CloudFormation
- Azure Resource Manager
- Google Cloud Deployment Manager

Security Acceptance Testing:
- BDD-Security
- Gauntlt
- Mittn

Infrastructure Tests:
- Serverspec
- Test Kitchen

Infrastructure Compliance Checks:
- HubbleStack
- InSpec

Production (Continuous Deployment)

Security checks before, during, and after code is deployed to production

Security Smoke Tests:
- ZAP Baseline Scan
- nmap
- ssllabs-scan

Configuration Safety Checks:
- AWS Config
- AWS Trusted Advisor
- Microsoft Azure Advisor
- Security Monkey
- OSQuery

Secrets Management:
- Ansible Vault
- Blackbox
- Chef Vault
- Docker Secrets
- Hashicorp Vault
- Pinterest Knox

Cloud Secrets Management:
- AWS KMS
- Azure Key Vault
- Google Cloud KMS

Cloud Security Testing:
- CloudSploit
- Nimbostratus

Server Hardening:
- dev-sec.io
- SIMP

Host Intrusion Detection System (HIDS):
- fail2ban
- OSSEC
- Samhain

Operations

Continuous security monitoring, testing, audit, and compliance checks

Fault Injection:
- Chaos Kong
- Chaos Monkey

Cyber Simulations:
- Game day exercises
- Tabletop scenarios

Penetration Testing:
- Attack-driven defense
- Bug Bounties
- Red team exercises

Threat Intelligence:
- Diamond Model
- Kill Chain
- STIX
- TAXII

Continuous Scanning:
- OpenSCAP
- OpenVAS
- Prowler
- Scout2
- vuls

Blameless Postmortems:
- Etsy Morgue

Continuous Monitoring:
- grafana
- graphite
- statsd
- seyren
- sof-elk
- ElastAlert
- 411

Cloud Monitoring:
- CloudWatch
- CloudTrail
- Reddalert

Cloud Compliance:
- Cloud Custodian
- Compliance Monkey
- Forseti Security

Building a DevSecOps Program (CALMS)

Culture

Break down barriers between Development, Security, and Operations through education and outreach

Automation

Embed self-service automated security scanning and testing in continuous delivery

Lean

Value stream analysis on security and compliance processes to optimize flow

Measurement

Use metrics to shape design and drive decisions

Sharing

Share threats, risks, and vulnerabilities by adding them to engineering backlogs

SANS AppSec
APPLICATION & SOFTWARE SECURITY

SANS APPSEC CURRICULUM

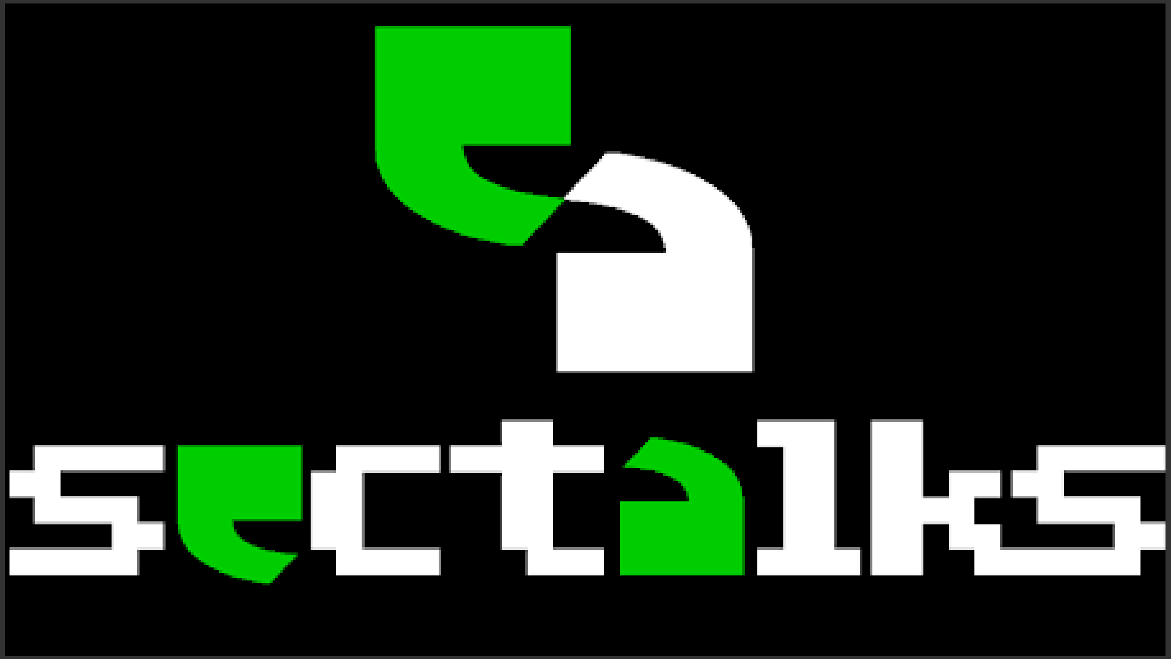
PLATFORM SECURITY CORE SPECIALIZATION

Tips for successful DevSecOps

1. Don't get in the developers' way
2. Security defects should be tracked as tickets
3. Learn when to break builds
4. Maximise automation, minimise false-positives
5. Celebrate your security wins

@heapsgooddev

heapsgood.dev



@sectalks_ADL

sectalks.org

Thanks

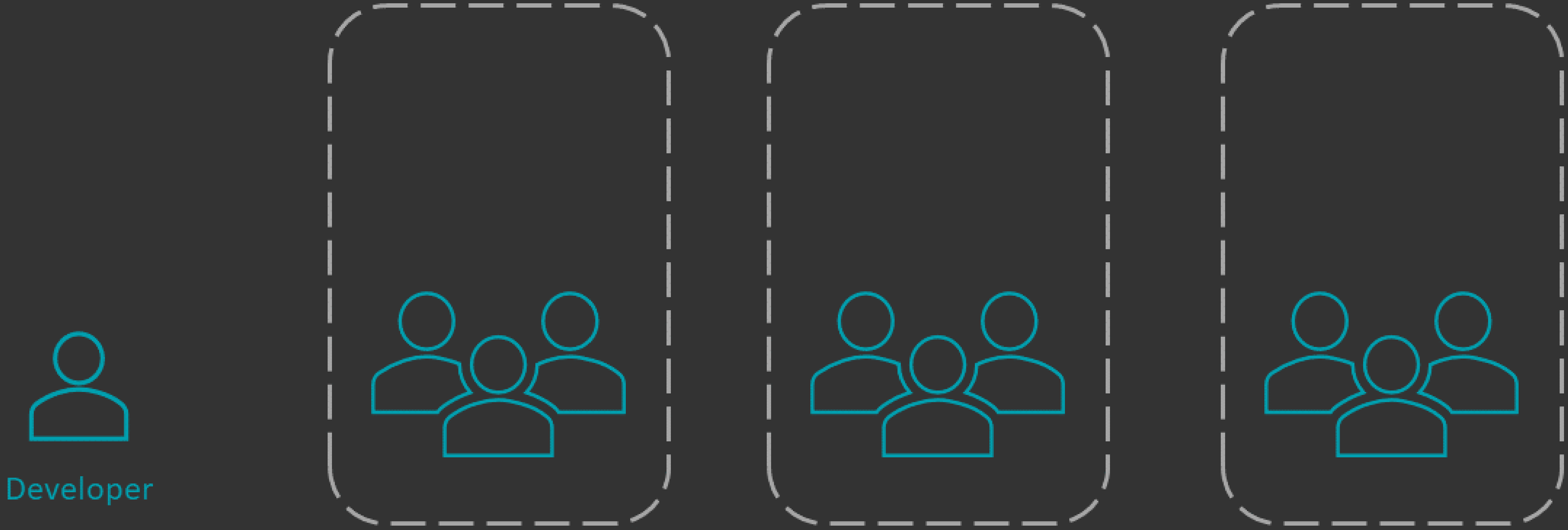
IF YOU SEE ME, COME SAY HI

@JakobTheDev

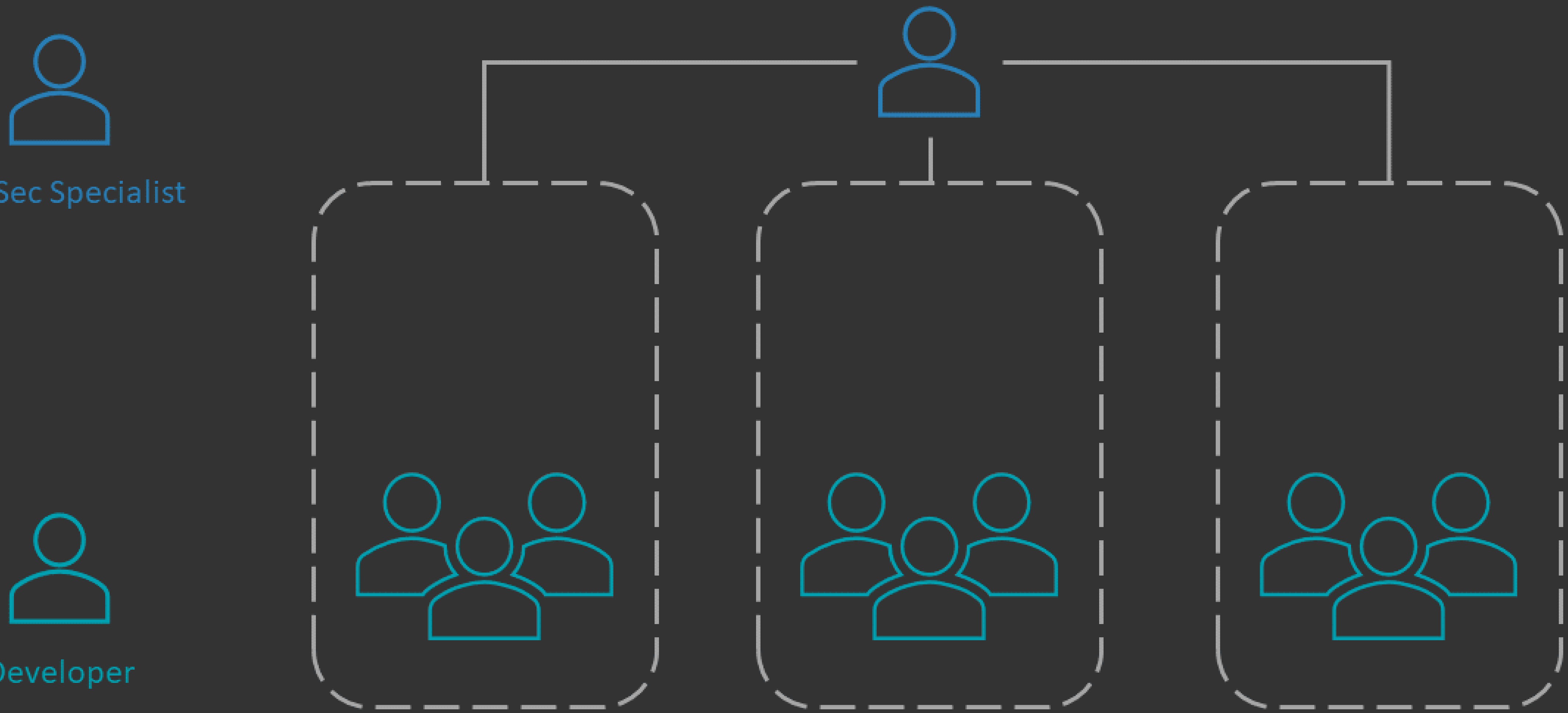
Culture

Remember 100 / 10 / 1?

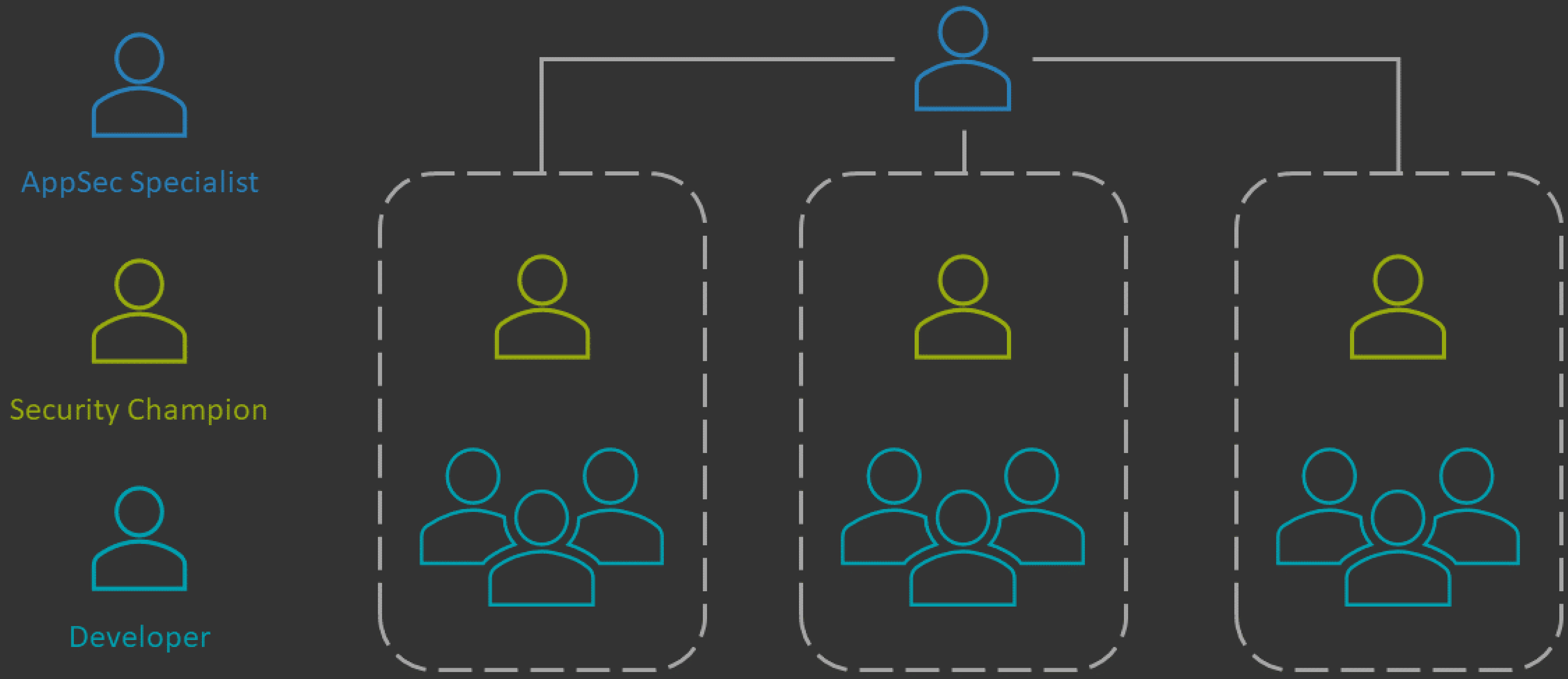
Team structure



Team structure



Team structure



MADE WITH
beautiful.ai

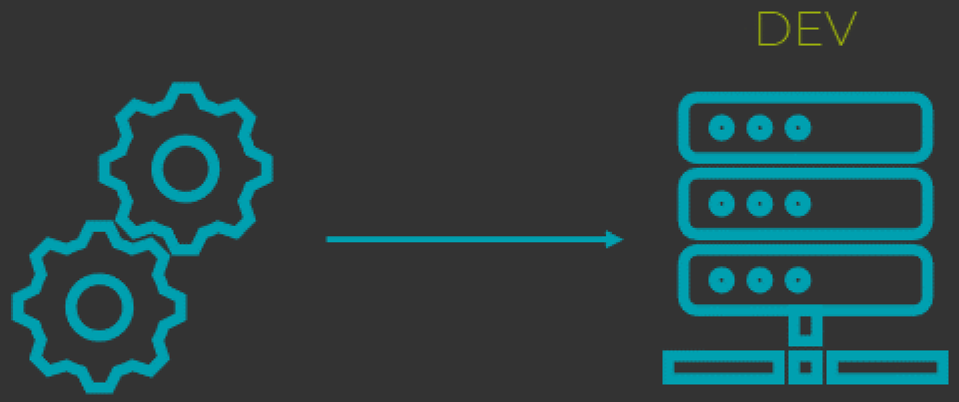
Process

Robust Release Management

Robust release management

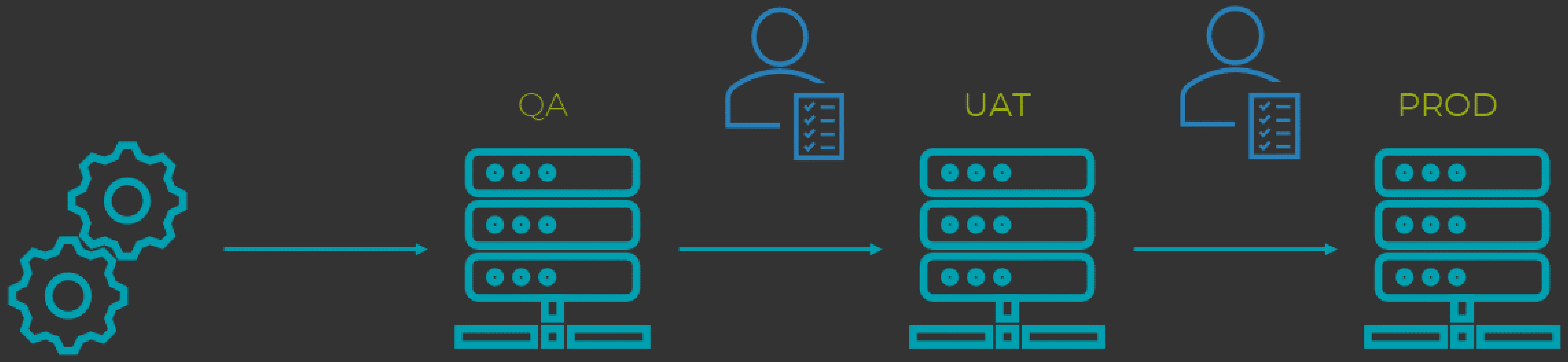
- Git workflow
- Branching policy
- CI / CD

Continuous Integration



MADE WITH
beautiful.ai

Continuous Delivery



- Secrets management
- Secure code reviews