

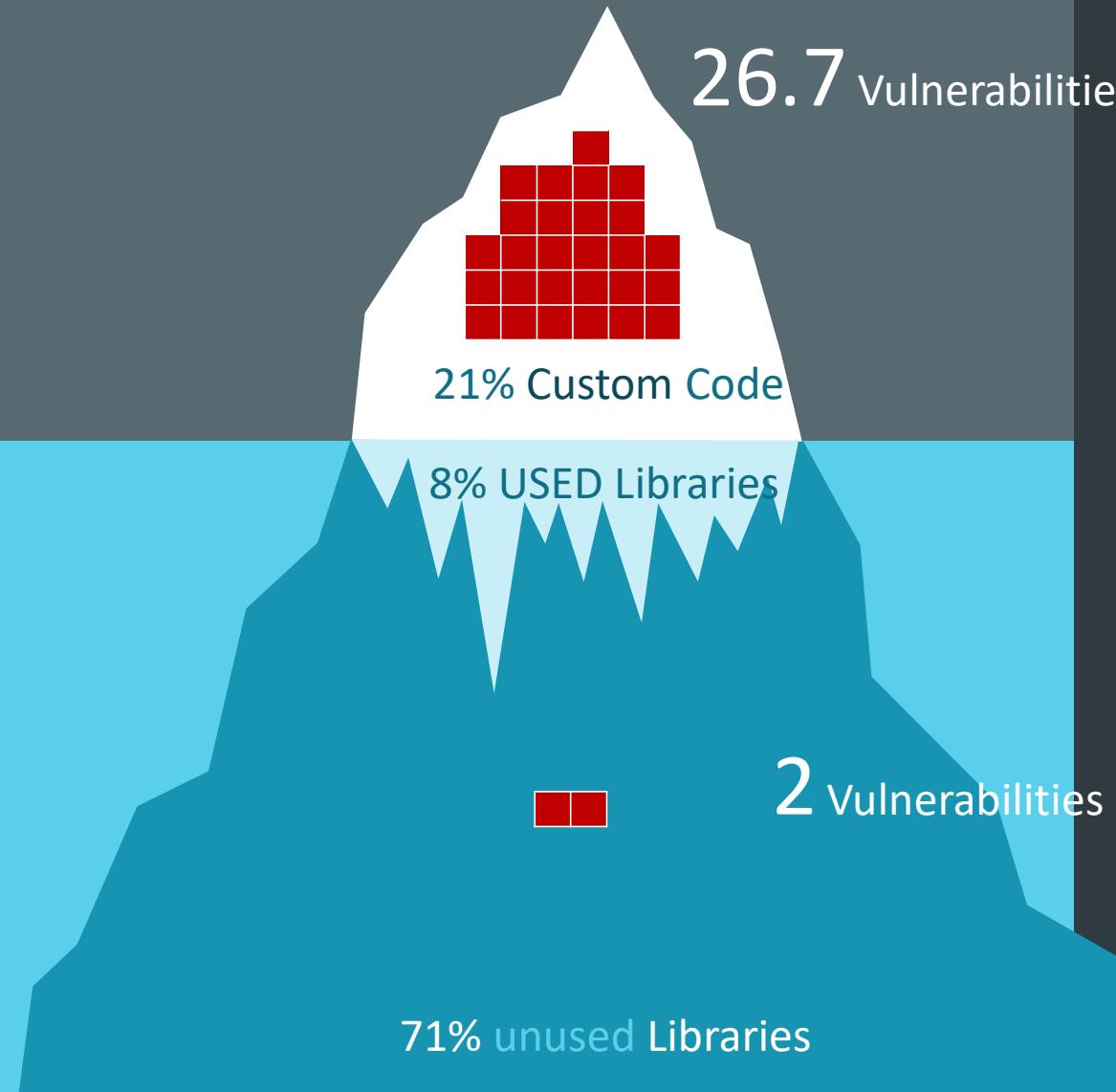


The Three ways of DevSECOps: Building a security pipeline with free tools



Jeff Williams – @planetlevel
CTO and Co-FOUNDER – Contrast



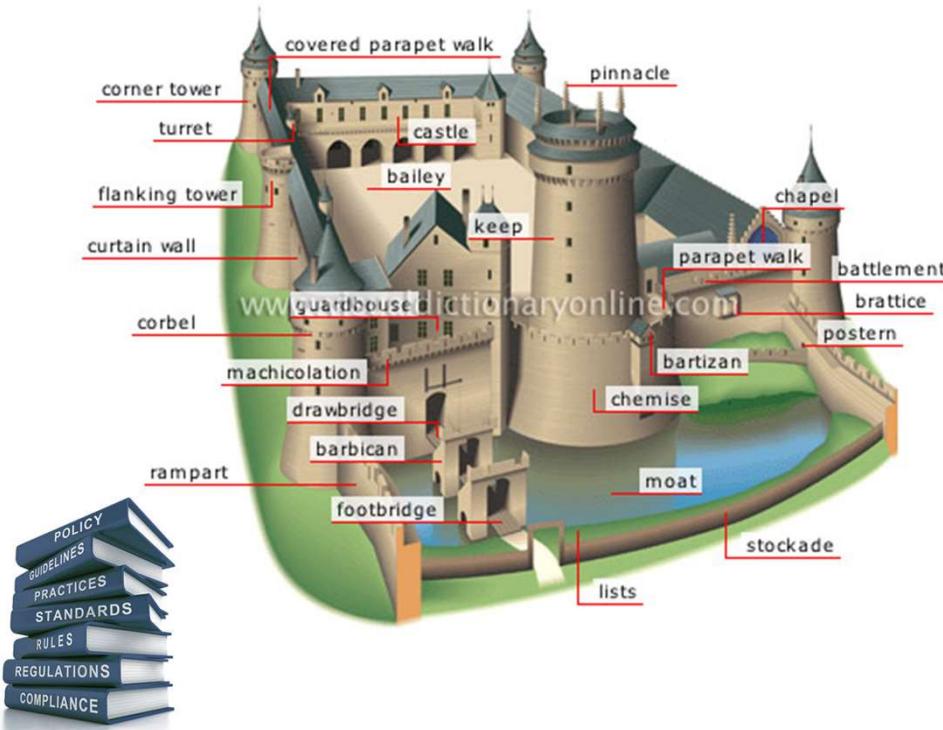


The Average application is **extremely vulnerable**

You are Under Attack

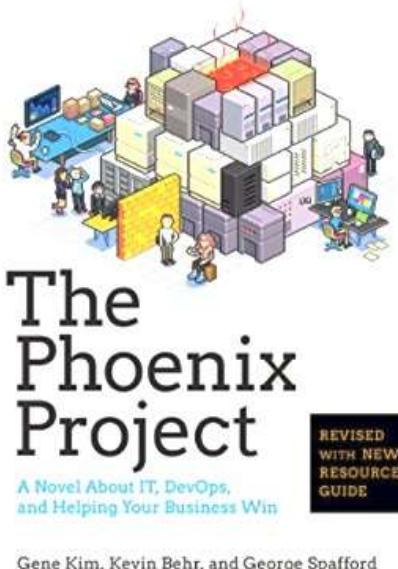
reflected-xss	58.6%	-8.1%
path-traversal	57.9%	-21.3%
sql-injection	53.8%	-11.6%
method-tampering	50.9%	23.8%
cve-2017-5638	27.2%	4.9%
cmd-injection	24.9%	-24.3%
csrf	17.1%	14.7%
cve-2017-9791	16.7%	-8.3%
cve-2017-12616	12.8%	0.3%
cve-2016-4438	8.3%	0.0%
ognl-injection	8.2%	-8.5%
cve-2013-2251	8.0%	-4.5%
padding-oracle	4.3%	4.3%
VP: patch forJBoss Remote Exploit	4.1%	0.1%
cve-2016-3081	1.0%	-3.2%
cve-2014-0112	0.0%	-8.3%

Please, Don't let anyone Blame Velocity



SECURING FAST-CHANGING THINGS IS DIFFERENT

DevSecOps is very promising...



DEVOPS

1. Establish work flow



2. Ensure instant feedback



3. Culture of experimentation



DEVSECOPS

1. Establish **security** work flow



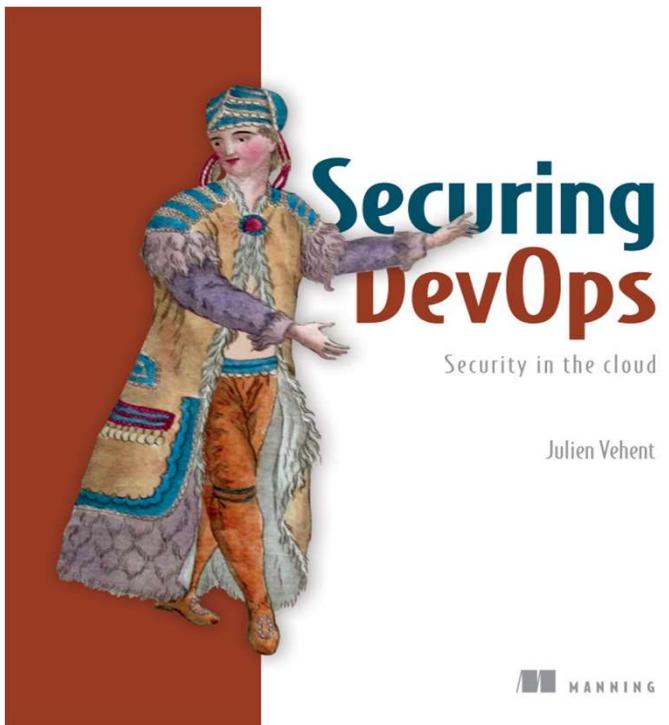
2. Ensure instant **security** feedback



3. Build a **security culture**



Recommended resources



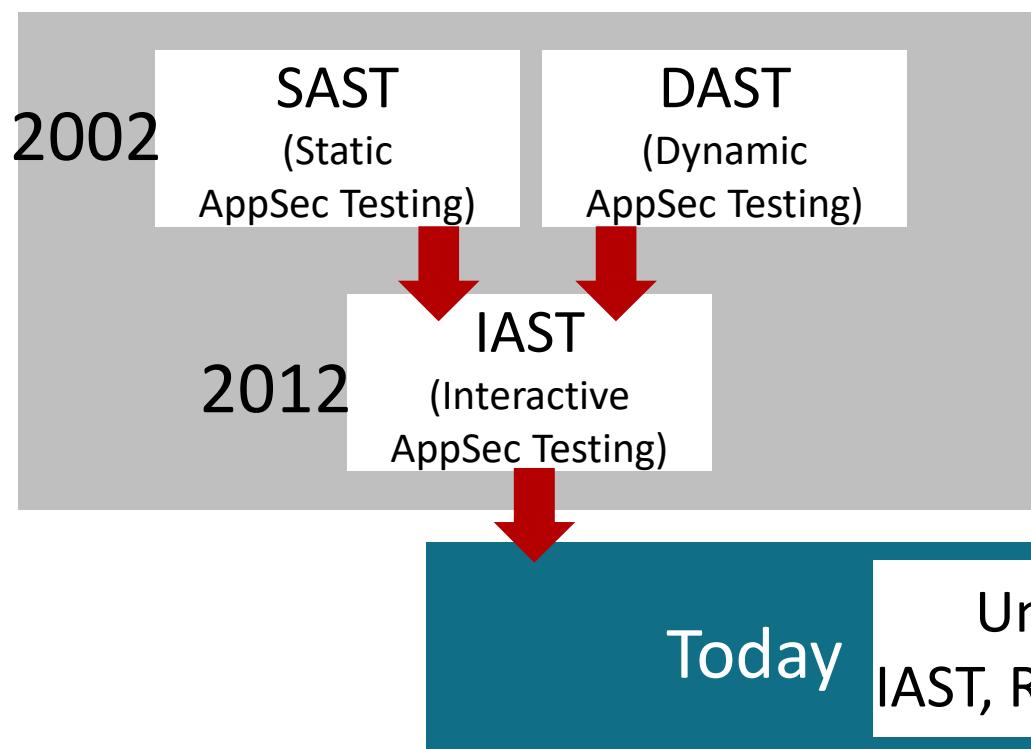
<https://www.manning.com/books/securing-devops>



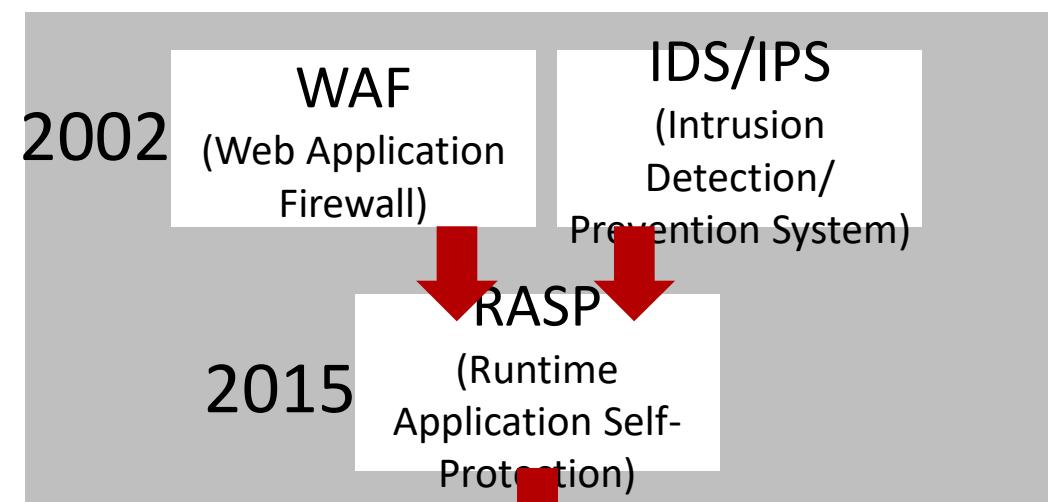
<https://dzone.com/refcardz/introduction-to-devsecops>

Evolution of appsec Automation

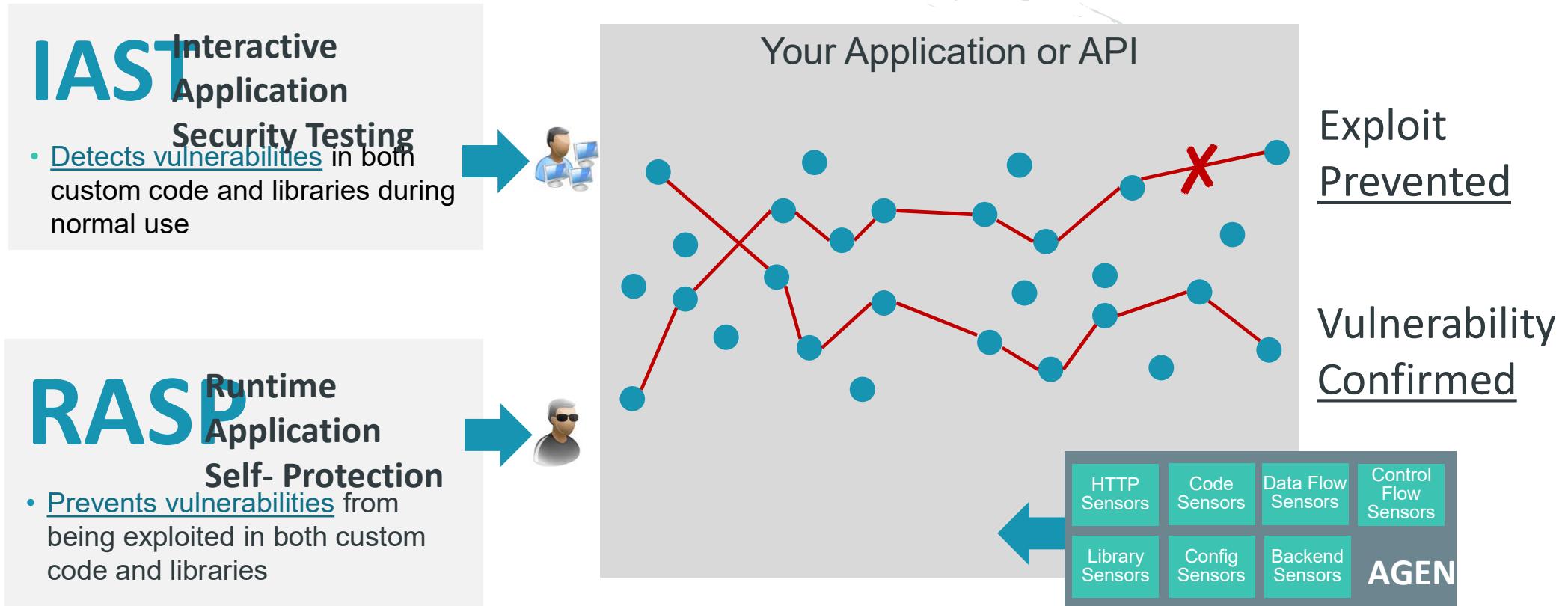
Development (find vulnerabilities)



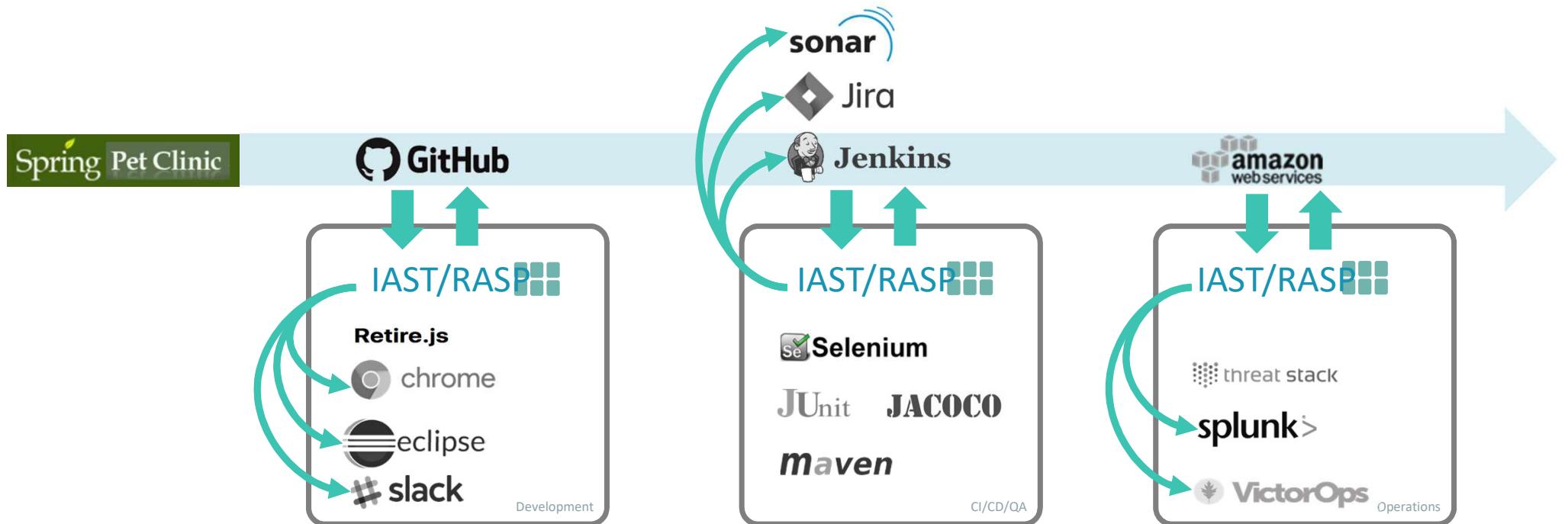
Operations (Prevent Exploit)



How IAST and RASP Work



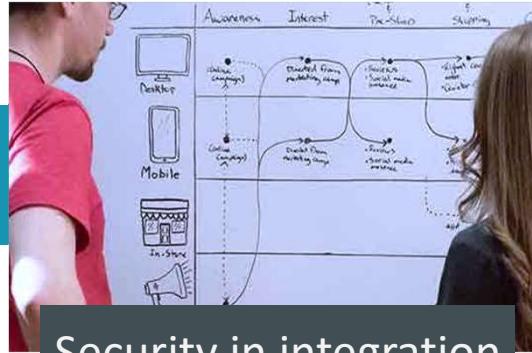
A developer's continuous pipeline



Shift left, right, and everywhere!



Security in development



Security in integration



Security in operations

EMPOWER

- Test my custom code and libraries
- Realtime feedback through my tools
- Don't slow me down

ASSURE

- Don't slow down my builds
- Integrate with my testing tools
- Real vulns break my build

PROTECT

- Tell me who is attacking and how
- Stop vulns from being exploited
- Don't create alert fatigue



Security in Development

Test my custom code AND libraries

- Must be extremely accurate
- Must work on modern apps with APIs and OSS

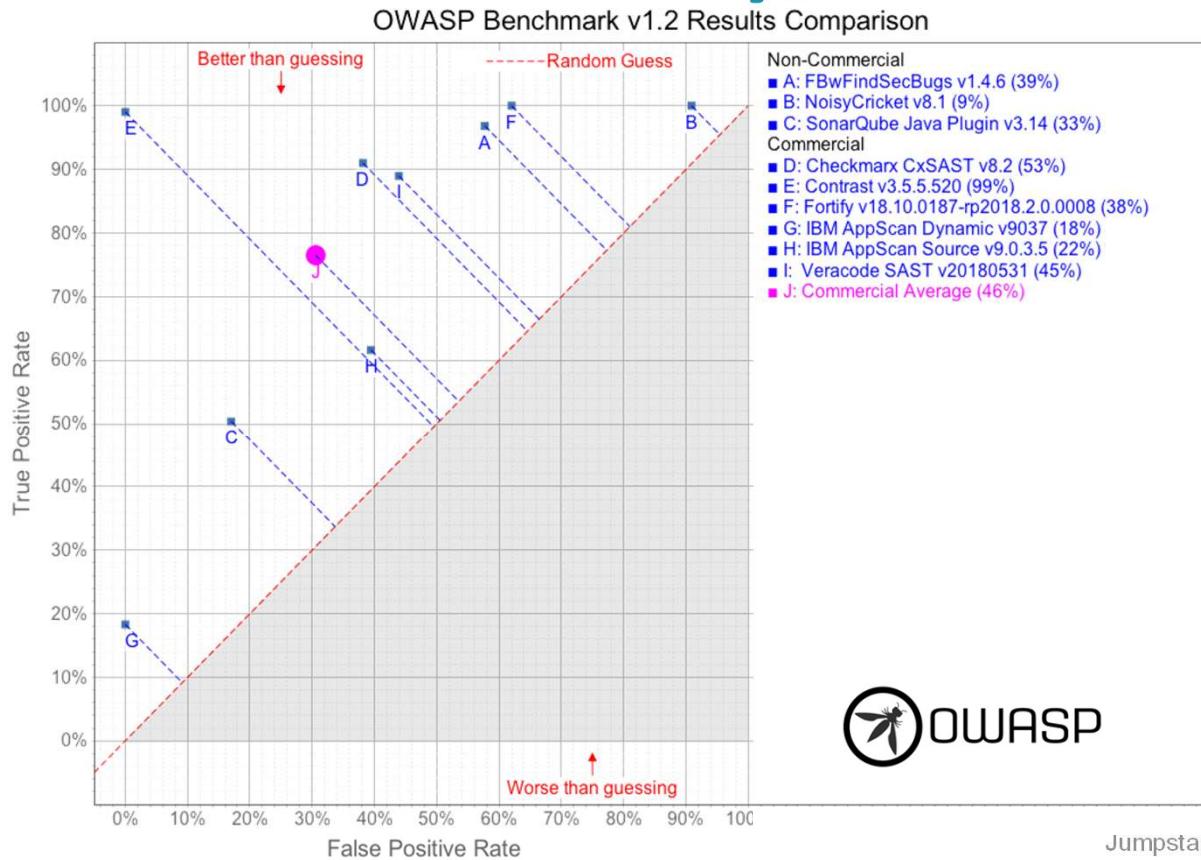
Realtime feedback through my tools

- Must integrate with tools I'm already using – NO PDF!

Don't slow me down

- Must not create bottleneck – NO SCANNING!

Automation is all about speed, accuracy, and ease-of-use



OWASP Benchmark

Free and open
application
benchmark
with thousands
of security test
cases

DevSecOps with OWASP

agent

 STEP 1
Download the Agent

Select A Language

- Java (3.5.2.303) Custom Agent Profile
- Java 1.5 (3.5.3.331)
- .NET (18.6.28)
- .NET Profiler (4.3.42)
- Node.js (1.20.0)
- Ruby (1.2.2)

1. Download

 STEP 2
Install On Your Server

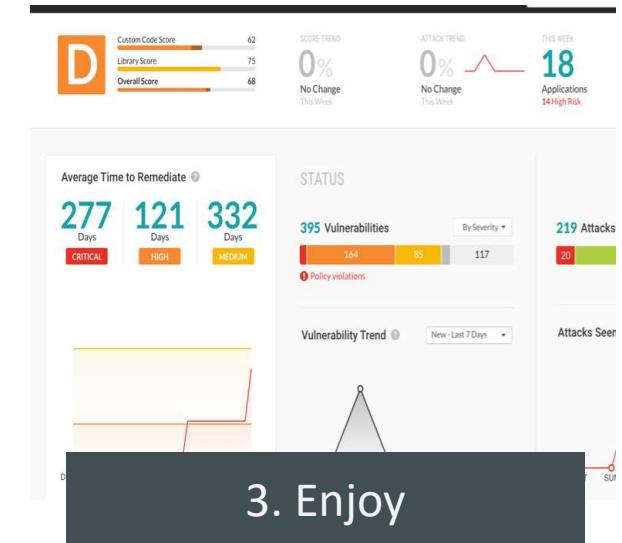
Select Your Container

- Tomcat 

JVM setting

```
-javaagent:/path/to/server/ agent .jar
```

2. install





Empower developers to assess their own security!

Custom code
- and -
Open Source

Security ... everyday

environment

The screenshot shows the Eclipse IDE interface. A code editor window displays Java code related to a Spring PetClinic application. Below the code editor is a 'Contrast' plugin window showing a list of vulnerabilities:

Severity	Vulnerability
Critical	Hibernate Injection from 'lastName' Parameter on 'owners' page
High	Cross-Site Request Forgery on 4 pages
High	Path Traversal from URL on '/favicon.ico' page
High	Path Traversal from URL on '/resources/css/petclinic.css' page
High	Path Traversal from URL on '/webjars/jquery/jquery.min.js' page
High	Cross-Site Scripting from Parameter Name on '/owners/{n}/pets/new' page
High	Stored Cross-Site Scripting from Tainted Database Value on '/owners/{n}' page
Medium	MD5 hash algorithm used at NamingHelper.java

IDE

OTHERS:

- eclipse
- Visual Studio
- IntelliJ IDEA
- slack
- HipChat

slack

A new **Critical** vulnerability was found in Spring PetClinic on qa.

Rule
Hibernate Injection

What Happened
We tracked the following data from "lastName" Parameter:

```
GET /owners?lastName=ffffffffffff
```

...which was accessed within the following code:

```
org.hibernate.jpa.spi.AbstractEntityManagerImpl#createQuery(), line 305
```

...and ended up in this database query:

```
SELECT DISTINCT owner FROM Owner owner left join fetch owner.pets WHERE owner.lastName LIKE 'ffffffffffff%'
```

Chatops

chrome

Retire.js

bootstrap 3.3.6 Found in http://localhost:8080/webjs/bootstrap/3.3.6/js/bootstrap.min.js Vulnerability Info: Medium 20184 XSS in data-target attribute

jquery-ui-dialog 1.11.4 Found in http://localhost:8080/webjs/jquery-ui/1.11.4/jquery-ui.min.js Vulnerability Info: High 281 XSS Vulnerability on closestText option

jQuery 2.2.4 jquery.min.js may execute CVE-2014-0161

Contrast Discovered Vulnerabilities

Critical Hibernate Injection
NOTE Anti Caching Controls Missing
NOTE Pages Without Anti Clickjacking Controls
NOTE Forms Without Autocomplete Prevention

spring

Find Owners

Last name: Find Owner

Add Owner

spring

Browser

Jira

Visual Studio Team Foundation Server

GitHub

Bugzilla

SERENA

Jenkins

Bamboo

maven

gradle

JUnit

nunit

Selenium

webhooks

splunk

AlienVault

LogRhythm

ArcSight

IBM Bluemix

Amazon Web Services

Microsoft Azure

Pivotal

VictorOps

Security In Integration

Don't slow down **my** builds

- Fully automated security testing with every build

Integrate with **my** testing tools

- Plugins, integrations, webhooks, and FULL REST API

Break **my** build... but only for real issues

- Set criteria for when to break the build

Fail the build

 Jenkins

Jenkins > spring-petclinic >

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[GitHub Hook Log](#)

[GitHub](#)

Project spring-petclinic

A vulnerable version of the venerable Spring Petclinic

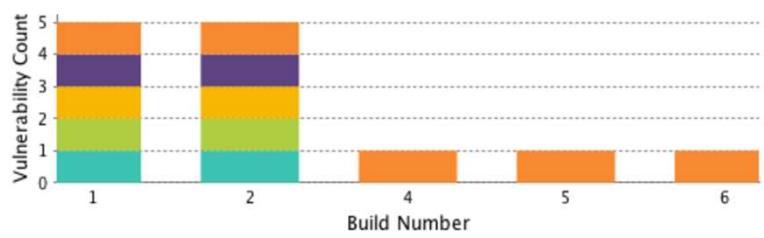
 [Workspace](#)

 [Recent Changes](#)

Permalinks

- [Last build \(#90\), 19 hr ago](#)
- [Last stable build \(#90\), 19 hr ago](#)
- [Last successful build \(#90\), 19 hr ago](#)
- [Last failed build \(#87\), 15 days ago](#)
- [Last unsuccessful build \(#87\), 15 days ago](#)
- [Last completed build \(#90\), 19 hr ago](#)

Vulnerability Trends Across Builds



Build Number	Vulnerability Count
1	5
2	5
4	1
5	1
6	1

Legend:

- session-timeout
- clickjacking-control-missing
- cache-controls-missing
- autocomplete-missing
- insecure-jsp-access

Build History

[trend](#)

find

#91 Sep 7, 2018 2:43 PM

#90 Sep 6, 2018 6:31 PM

Automatic bugtracking integration

The screenshot shows a bug tracking system interface. On the left, a list of vulnerabilities is displayed, ordered by updated. The first few items are:

- SP-36 CONTRAST: Session Rewriting All...
- SP-35 CONTRAST: Regular Expression ...
- SP-34 CONTRAST: Stored Cross-Site S...
- SP-33 CONTRAST: Cross-Site Request ...
- SP-32** CONTRAST: Hibernate Injection f...
- SP-31 CONTRAST: Path Traversal from ...
- SP-30 CONTRAST: Path Traversal from ...
- SP-29 CONTRAST: Path Traversal from ...
- SP-28 CONTRAST: IMEI hash algorithm ...

The item "SP-32" is highlighted with a blue border. On the right, a detailed view of this specific bug is shown:

Spring-Petclinic / SP-32

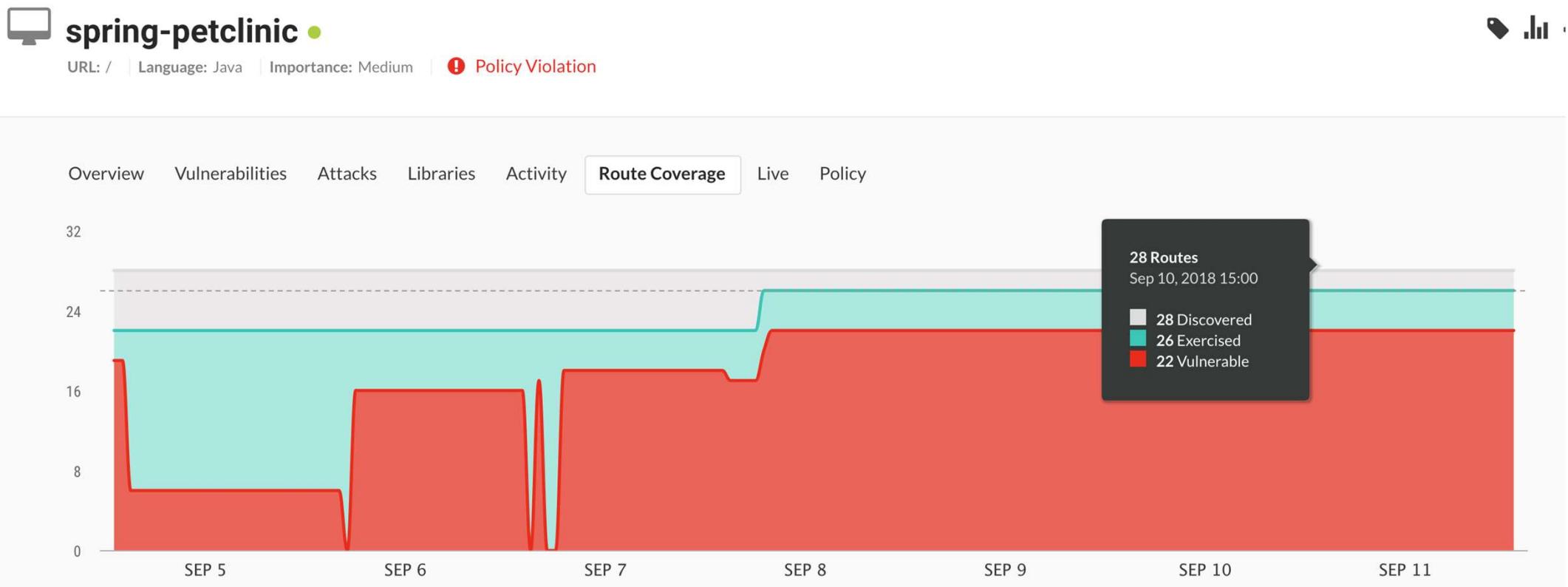
CONTRAST: Hibernate Injection from "lastName"
Parameter on "/owners" page

Buttons: Edit, Comment, Assign, Backlog, Selected for Development, Work

Type: Bug
Status: BACKLOG (View workflow)
Priority: Critical
Resolution: Unresolved
Labels: None

Description
Trace ID: C2BX-5SXK-K24Y-RJLB
Trace Link:
<https://apptwo.contrastsecurity.com/Contrast/static/ng/index.html#/d3b189b6-c17d-46e8-b56e-4ba40794630e/applications/bc014c44-e92b-4bdf-9939-a3b1300c57d8/vulns/C2BX-5SXK-K24Y-RJLB>

Measuring attack surface



Overview

On new code

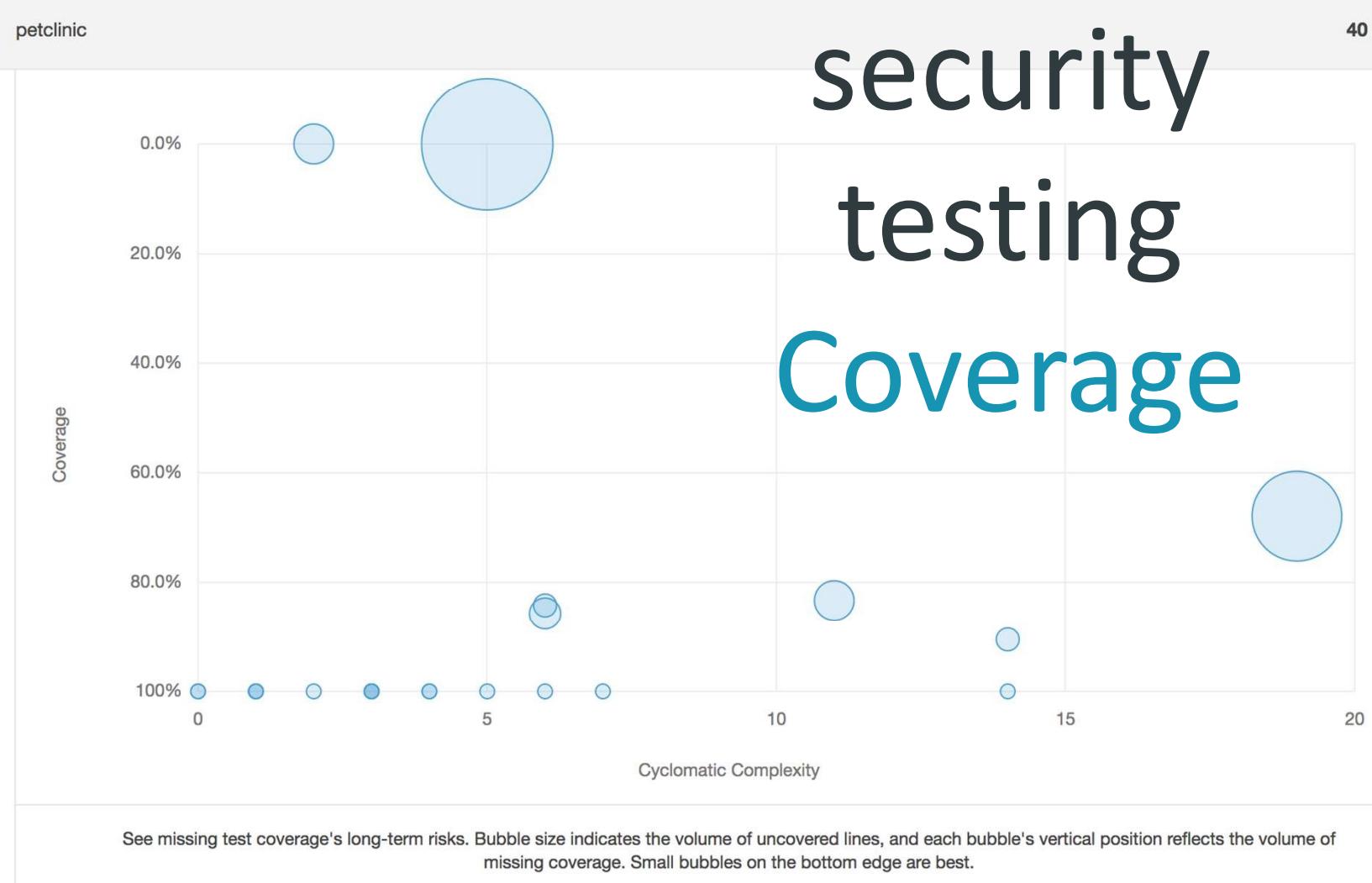
Coverage	41.7%
Lines to Cover	30
Uncovered Lines	17
Line Coverage	43.3%
Conditions to Cover	6
Uncovered Conditions	4
Condition Coverage	33.3%

Overall

Coverage	84.4%
Lines to Cover	267
Uncovered Lines	33
Line Coverage	87.6%
Conditions to Cover	60
Uncovered Conditions	18
Condition Coverage	70.0%

Tests

Unit Tests	40
Errors	0
Failures	0
Skipped	1
Success	100%
Duration	4s





Security in operations

Tell me who is attacking and how

- I need actionable threat intelligence

Stop vulnerabilities from being exploited

- Must not overblock (FP) or underblock (FN)

Don't create alert fatigue

- Don't warn me about meaningless probes

PROTECT Applications and APIs WITH R

- PREVENTS Known Vulnerabilities from exploit
- Prevents LATENT Vulnerabilities from exploit
- Fast response to prevent NOVEL vulnerabilities from exploit



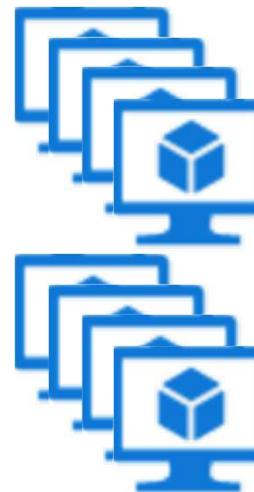
Custom Code

Open Source
Libraries and
Frameworks

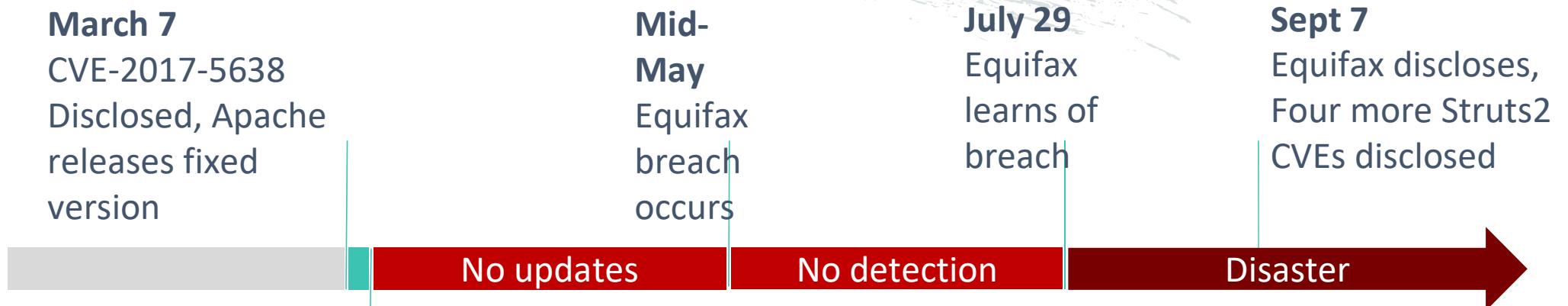
Application Server
and Platform

RASP deploys with your application

- Ansible
- Puppet
- Docker
- Kubernetes
- Whatever...



Protect against latent vulnerabilities



March 8
We observe widespread attacks

You must have infrastructure to respond **within hours**.

RASP supports complex protocols

Bad Guy

A screenshot of an Eclipse IDE window titled "Java - Restaurants.java - Eclipse Platform". The code editor shows the following Java code:

```
import java.util.ArrayList;
import java.util.List;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.PreparedStatement;
import java.sql.ResultSetMetadata;
```

The code includes imports for `java.util.ArrayList`, `java.util.List`, `java.sql.Connection`, `java.sql.DriverManager`, `java.sql.ResultSet`, `java.sql.Statement`, `java.sql.PreparedStatement`, and `java.sql.ResultSetMetadata`. Below the code, there are three snippets of code highlighted in red:

- AcmeInternalType#cmd: `java.lang.Runtime`
- AcmeInternalType#mtd: `getRuntime().exec`
- AcmeInternalType#args: `'cmd.exe','/C','calc'`

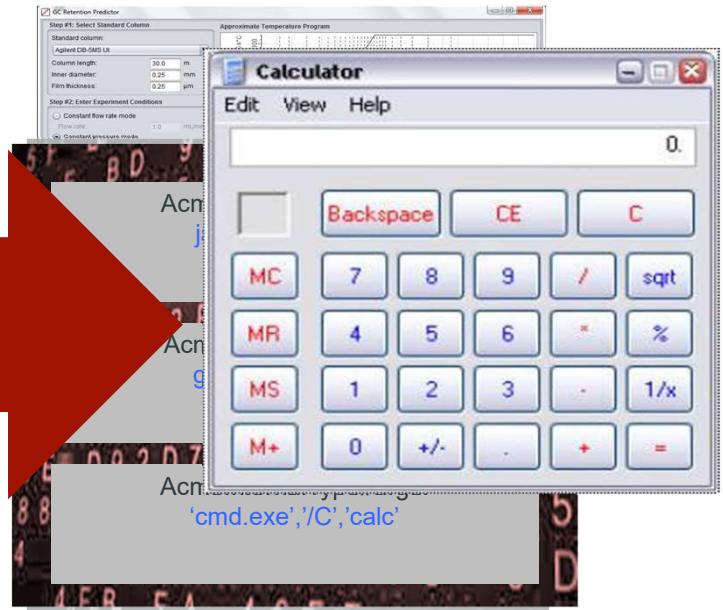
Untrusted deserialization

Attacker sends malicious object

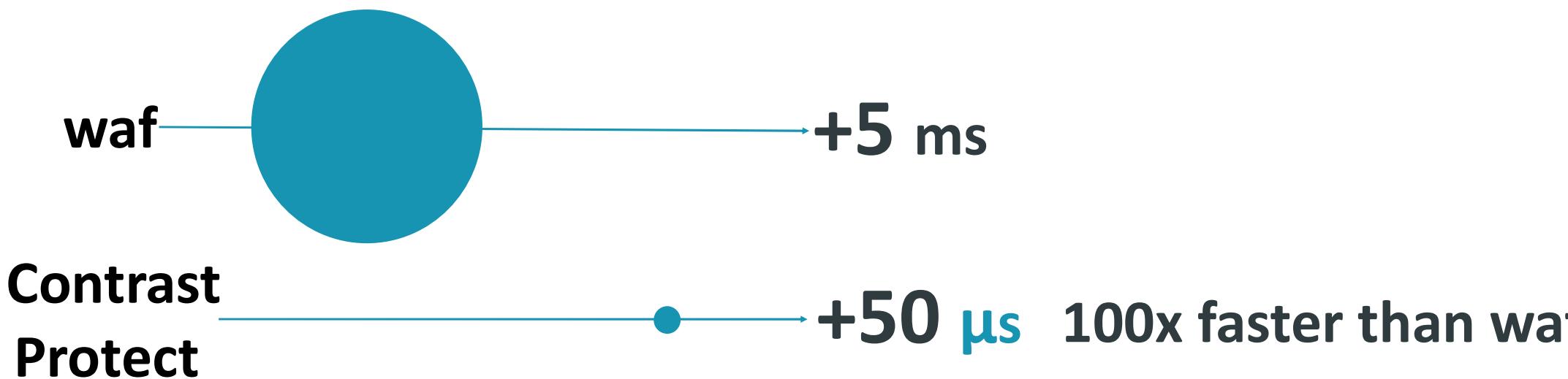
```
POST / HTTP/1.1
User-Agent: Java/1.8.0_74
Host: localhost
Accept: text/html, image/gif, image/jpeg, *, *; q=.2, */*; q=.2
Content-type: application/x-www-form-urlencoded
Content-Length: 1876
Connection: close

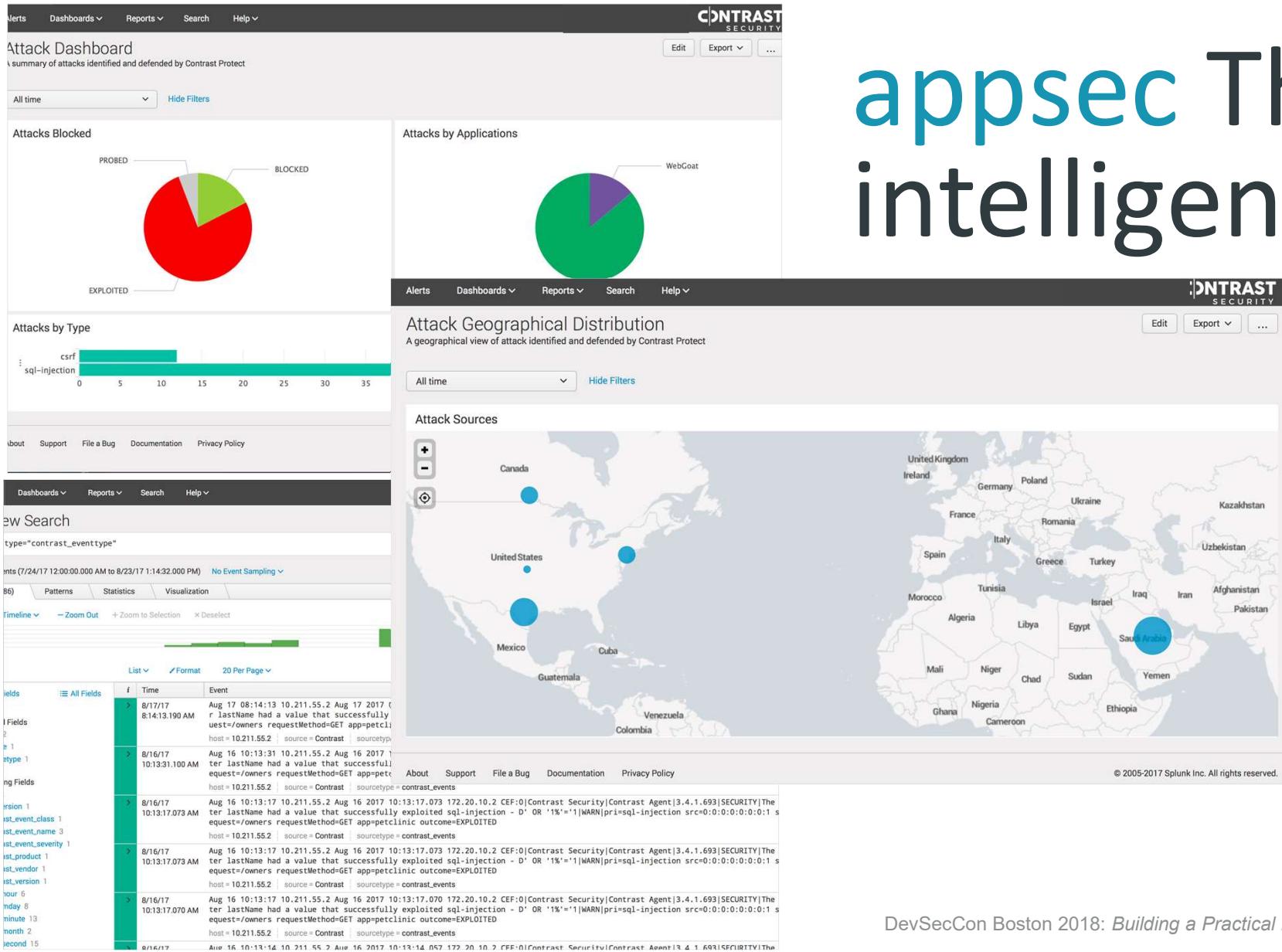
rOOABXNyADJzdW4ucmVmbGVjdC5hbm5vdGF0aW9uLkFubm9OYXRpb25Jbn2vY2F0aW9uSGFu2
Gx1c1XK9Q8Vy361AgACTAAMbVVtYmVfFdWVzdAAPTGphdmEvdXRpbCSNYXA7TAAEdHivZX
QAEUxqYXZhL2xhbmcvQ2xhc3M7eHBzfQAAAAEADWphdmEudXRpbCSNYXB4cgAXamF2YS5sYW5
```

Application



RASP IS FAST





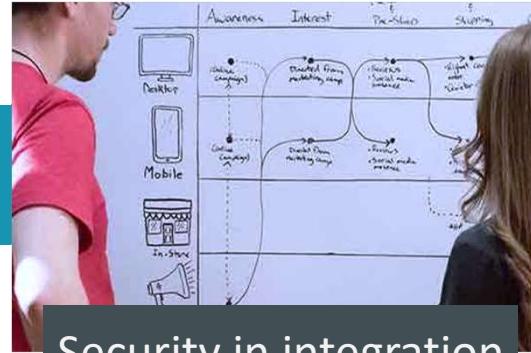
appsec Threat intelligence

DevSecCon Boston 2018: *Building a Practical DevSecOps Pipeline for Free*

DevSecOps Pipeline



Security in development



Security in integration



Security in operations

EMPOWER

- Test my custom code and libraries
- Realtime feedback through my tools
- Don't slow me down

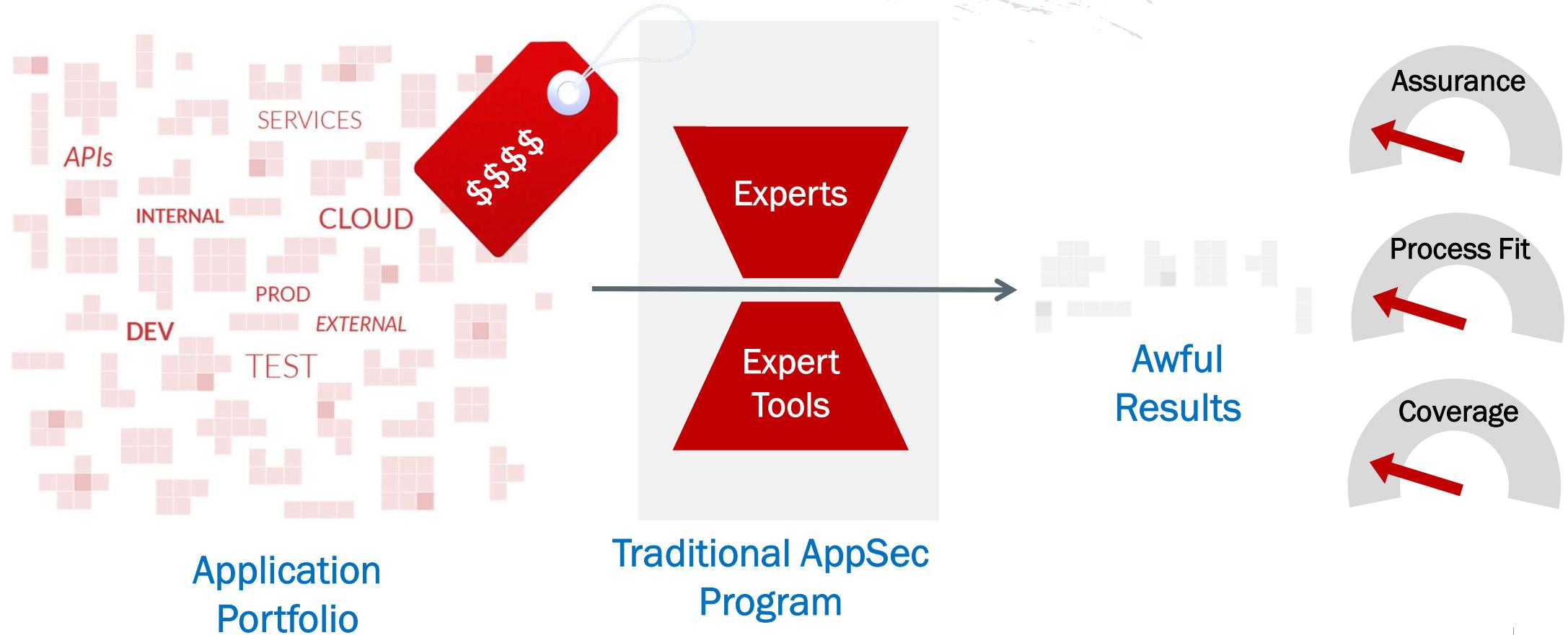
ASSURE

- Don't slow down my builds
- Integrate with my testing tools
- Real vulns break my build

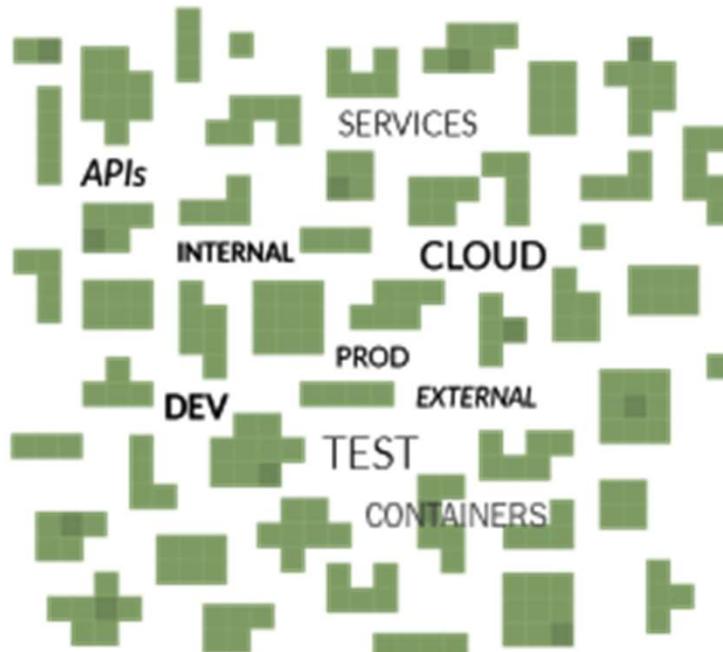
PROTECT

- Tell me who is attacking and how
- Stop vulns from being exploited
- Don't create alert fatigue

SUMMARY: Scanners and Firewalls don't Scale



Summary: devops + security works at scale



Enable application portfolio
with IAST/RASP agents



Continuous assessment
and protection in parallel



Free DevSecOps Tools

OWASP Dependency check

- Free SCA tool to scan for known vulnerabilities in libraries.
- https://www.owasp.org/index.php/OWASP_Dependency_Check

Retire.js

- Free SCA tool to scan for known vulnerabilities in javascript libraries
- <https://retirejs.github.io/retire.js>

Contrast CE

- Free and full-strength IAST, RASP, and SCA for Java applications and APIs.
- <http://contrastsecurity.com/ce>



OWASP
SINGAPORE

THANK YOU!

Ask me anything

Jumpstarting Your DevSecOps Pipeline with IAST and RASP
Jeff Williams @planetlevel

CONTRAST
SECURITY