CONTRAST SECURITY

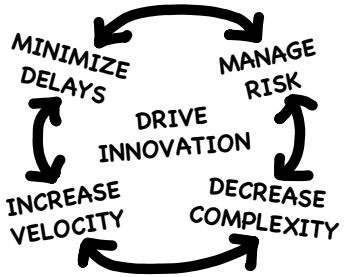
# HOW TO BUILD AWESOME SECURITY INSTRUMENTATION TO AUTOMATE APPSEC TESTING AND PROTECTION

Jeff Williams | CTO and cofounder | jeff.williams@contrastsecurity.com

OWASP Open Web Application Security Project

## THE GOAL IS SIMPLE:

# DELIVER VALUE TO CUSTOMERS, FASTER

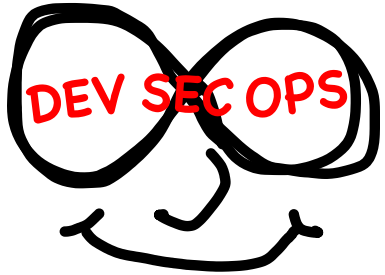



### MAJOR INITIATIVES:

- ☐ SECURITY MODERNIZATION
- ☐ DEVOPS MANDATE
- ☐ DIGITAL TRANSFORMATION
- ☐ LEGACY TOOL REVIEW
- ☐ BREACH PREVENTION
- ☐ AUTOMATION INITATIVES

## DEVELOPMENT'S RESPONSE:

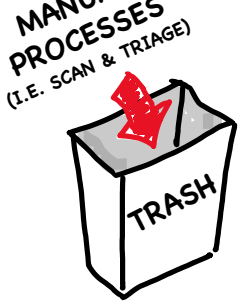
**FLOW | FEEDBACK | CONTINUOUS EXPERIMENTATION & LEARNING**




\+

- CLOUD
- CONTAINERS
- MICROSERVICES / API'S
- OPEN SOURCE
- CI/CD

\=




\=

**WHY NOT?**

**200X** MORE FREQUENT ~~DEPLOYS~~ ANALYSIS

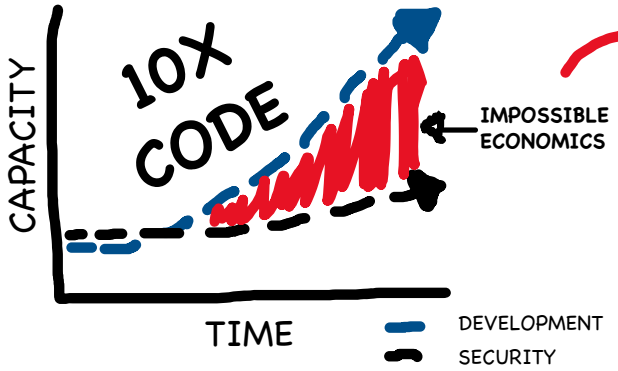
**24X** FASTER ~~RECOVERY~~ REMEDIATION TIMES

**3X** LOWER ~~CHANGE FAILURE RATE~~ SECURITY BACKLOG

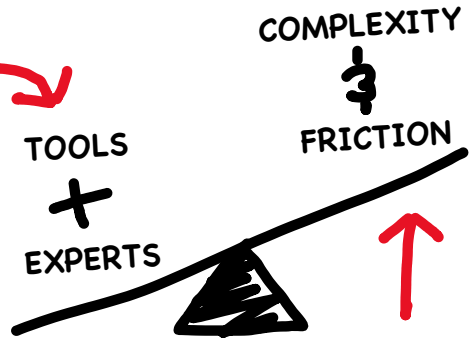
**10X** AMOUNT OF CODE SECURE

## SECURITY'S CHALLENGE:

HOW DO YOU SECURE **MODERN SOFTWARE** WITH **LEGACY TOOLS**?




TOOLS + EXPERTS

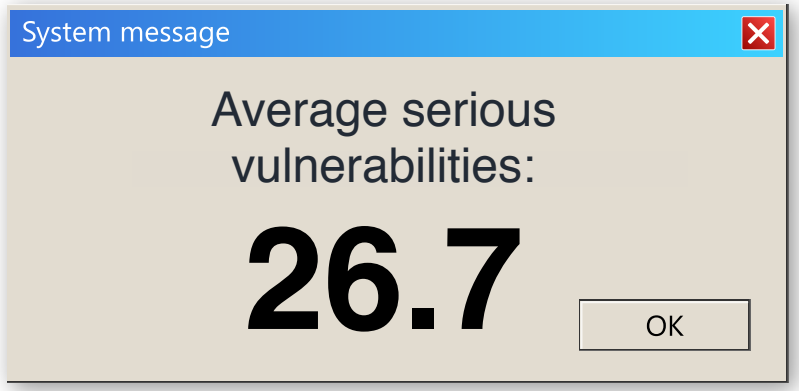



### LEGACY OUTCOMES:

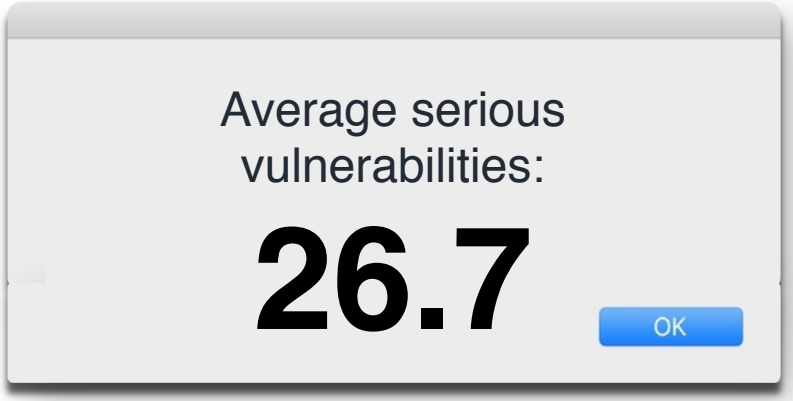
1. SECURITY SLOWS VALUE DELIVERY STREAM
2. DEVELOPMENT BYPASSES SECURITY & RELEASE INSECURE CODE
3. RUN SECURITY ASYNCHRONOUS TO SDLC AS A COMPLIANCE CHECK

# MODERN SECURITY CAN BE MORE!

# IS IT WORKING YET?




2002




2019

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Formal Methods | | SW CMM | | Vulnerability Disclosure | Common Criteria | | OWASP T10 | | WAF | | Developer Training | | | DevOps | | Shift Left |
| 1970 | 1985 | 1991 | 1997 | 1998 | 1999 | 1998 | 2001 | 2002 | 2003 | 2004 | 2005 | 2007 | 2009 | 2013 | 2015 | 2017 |
| | TCSEC DOD 5200.28 | | SSE-CMM | | Penetration Testing | | Dynamic Scanning | | Static Analysis | | Compliance | | BSIMM OpenSAMM | | DevSecOps | |

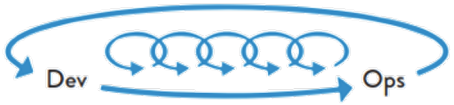
# DEVSECOPS IS ABOUT CHANGING *SECURITY* NOT *DEVOPS*


## DEVOPS

1. Establish work flow


2. Ensure instant feedback


3. Culture of experimentation

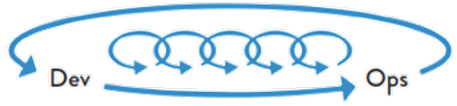



## DEVSECOPS

1. Establish security work flow


2. Ensure instant security feedback


3. Build a security culture



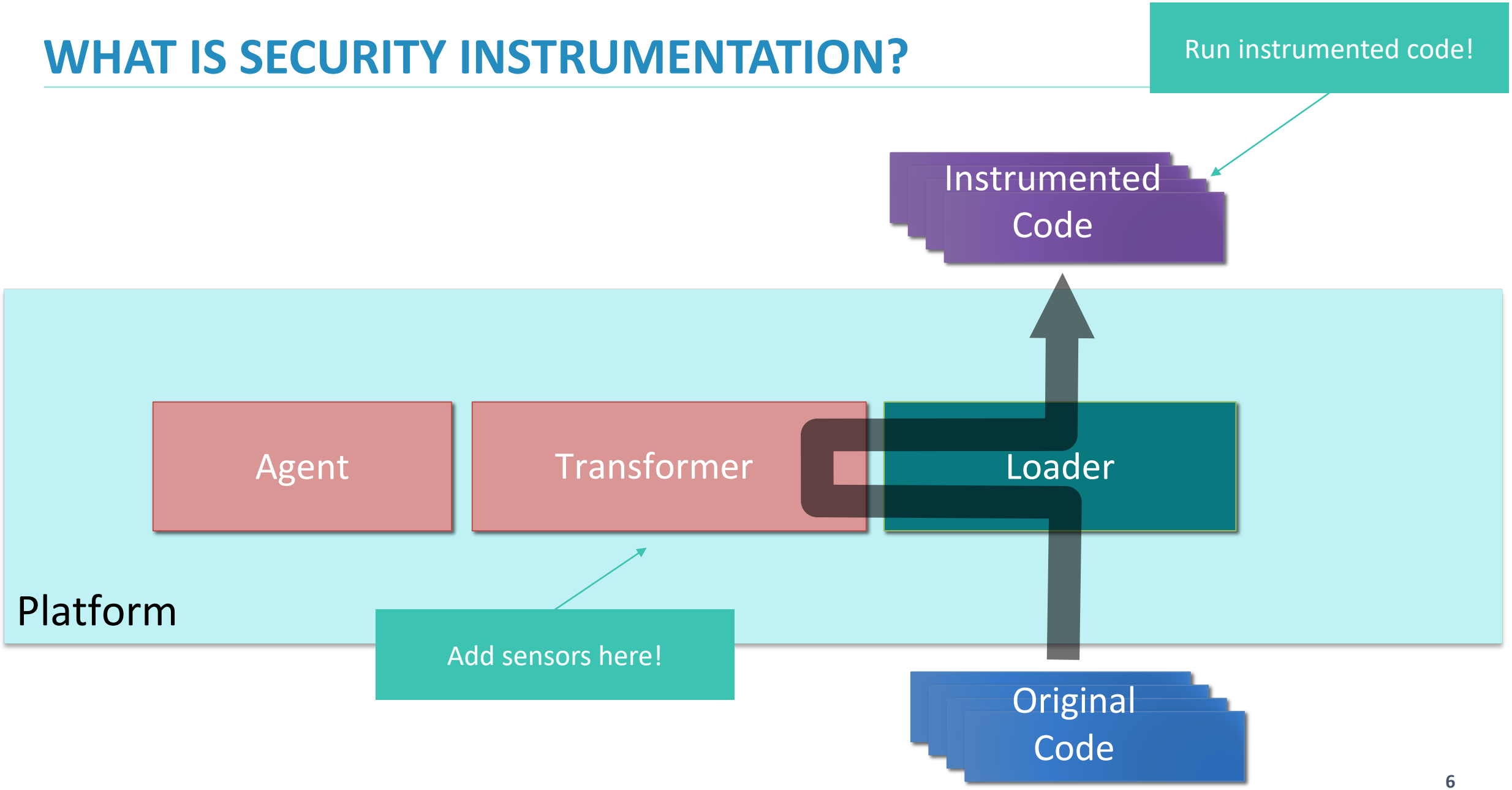

https://itrevolution.com/the-three-ways-principles-underpinning-devops/



CONTRAST SECURITY

# INSTRUMENTATION CHANGES EVERYTHING

# WHAT IS SECURITY INSTRUMENTATION?



# 1. SECURITY TESTING WITH INSTRUMENTATION

UnsafeSQLAgent.java

```
1  package com.contrastsecurity;
2
3  import static net.bytebuddy.matcher.ElementMatchers.*;
9
10 public class UnsafeSQLAgent {
11
12     public static void premain(String arg, Instrumentation inst) throws Exception {
13         System.out.println( "UnsafeSQLAgent installed" );
14         new AgentBuilder.Default()
15         .ignore(none())
16         .type(hasGenericSuperType(named("java.sql.Statement")))
17         .transform((b, td, cl, m) -> b.visit(Advice.to(SecurityAdvice.class)
18                 .on(named("execute").or(named("executeQuery")))))
19         .installOn(inst);
20     }
21
22     public static class SecurityAdvice {
23         @Advice.OnMethodExit
24         public static void exit(@Advice.Argument(0) Object p) throws Exception {
25             System.out.println( "WARNING: Unparameterized SQL -> " + p );
26             new Throwable().printStackTrace();
27         }
28     }
29 }
30
```

Put SecurityAdvice inside Statement.execute

Report rule violation when it happens

# DEMO: TESTING FOR SQL INJECTION

Terminal — java -Djava.util.logging.config.file=/Users/jeffwilliams/git/apache-tomcat-8.5.43/conf/logging.properties -Djava.util.logging.manager=org.apach...

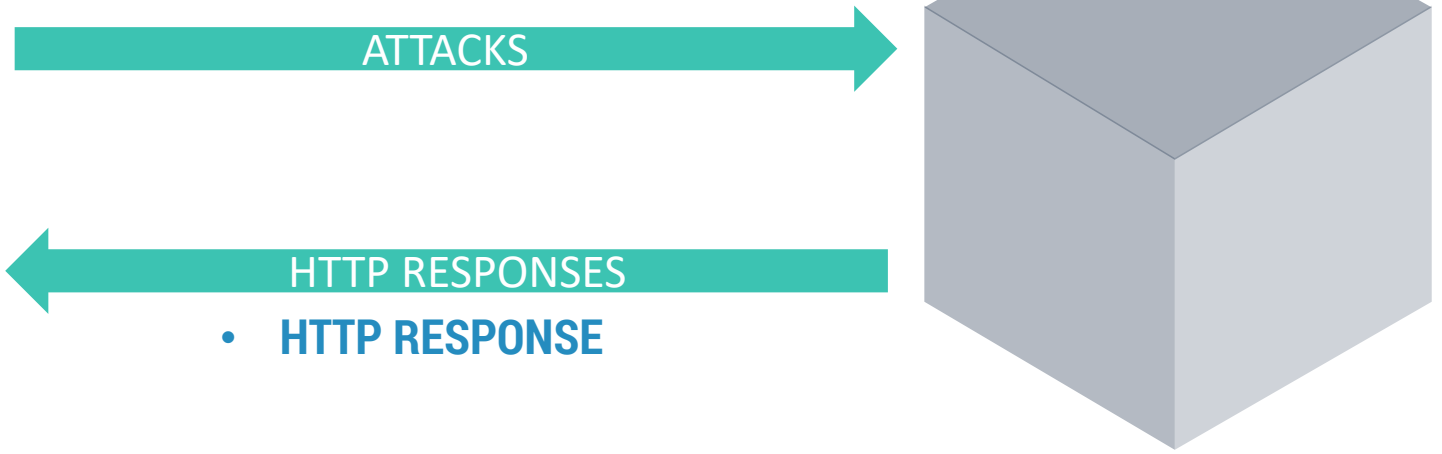
Usage of interface net.bytebuddy.asm.Advice$Return  Done

```
Jeff's-MacBook-Pro:apache-tomcat-8.5.43 jeffwilliams$ export JAVA_TOOL_OPTIONS="-javaagent:/Users/jeffwilliams/eclipse-workspace/UnsafeSQL/target/unsafe-sql-0.9-jar-with-dependencies.jar"
Jeff's-MacBook-Pro:apache-tomcat-8.5.43 jeffwilliams$
Jeff's-MacBook-Pro:apache-tomcat-8.5.43 jeffwilliams$ bin/catalina.sh run
Using CATALINA_BASE:   /Users/jeffwilliams/git/apache-tomcat-8.5.43
Using CATALINA_HOME:   /Users/jeffwilliams/git/apache-tomcat-8.5.43
Using CATALINA_TMPDIR: /Users/jeffwilliams/git/apache-tomcat-8.5.43/temp
Using JRE_HOME:        /Library/Java/JavaVirtualMachines/jdk1.8.0_211.jdk/Contents/Home
Using CLASSPATH:       /Users/jeffwilliams/git/apache-tomcat-8.5.43/bin/bootstrap.jar:/Users/jeffwilliams/git/apache-tomcat-8.5.43/bin/tomcat-juli.jar
Picked up JAVA_TOOL_OPTIONS: -javaagent:/Users/jeffwilliams/eclipse-workspace/UnsafeSQL/target/unsafe-sql-0.9-jar-with-dependencies.jar
UnsafeSQLAgent installed
16-Mar-2020 14:03:48.472 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server version:        Apache Tomcat/8.5.43
16-Mar-2020 14:03:48.474 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server built:         Jul 4 2019 20:53:15 UTC
16-Mar-2020 14:03:48.474 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server number:       8.5.43.0
16-Mar-2020 14:03:48.474 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Name:             Mac OS X
WARNING: Unparameterized SQL -> SELECT * FROM tickets WHERE ticket='10000'
java.lang.Throwable
        at org.hsqldb.jdbc.JDBCStatement.executeQuery(Unknown Source)
        at com.acme.ticketbook.Database.queryUnsafe(Database.java:151)
        at com.acme.ticketbook.Database.getTicket(Database.java:248)
        at com.acme.ticketbook.TicketService.get(TicketService.java:28)
        at org.apache.jsp.check_jsp._jspService(check_jsp.java:213)
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:741)
```

Now your normal test cases fail for security reasons if you have a vulnerability!

# TYPICAL “SCANNING” APPROACH

SCANNER SIMULATES ATTACKS TO DETECT VULNERABILITIES




## PROBLEMS WITH SPEED, SCALE, AND ACCURACY

# TESTING WITH AN AGENT ON THE INSIDE!

SECURITY SPECIALISTS TRANSLATE NEW THREATS INTO RULES

DEVELOPER USES APPLICATION NORMALLY

NORMAL APP USE

AGENT OBSERVES APPLICATION BEHAVIOR DIRECTLY

DETAILED SECURITY ANALYSIS RESULTS

- EXACT LINE OF CODE, FULL QUERY, FULL HTTP REQUEST, CONTROL FLOW, DATA FLOW, LIBRARIES, CONFIGURATION, ETC…

```
cat rules/sqli.yaml
---
# Java Sensor Toolkit (JST)
# https://openo11y.org

sensors:

- name: "get-routes"
  description: "Identifies the route for this HTTP request"
  methods:
  - "javax.servlet.Servlet.service"
  captures:
  - "#P0.getRequestURI()"

- name: "get-unsafe-queries"
  description: "Identifies unparameterized database queries"
  methods:
  - "java.sql.Statement.execute"
  - "java.sql.Statement.addBatch"
  - "java.sql.Statement.executeQuery"
  - "java.sql.Statement.executeUpdate"
  excludes:
  - "java.sql.PreparedStatement"   # not vulnerabl
  captures:
  - "#ARGS"

reports:

 - name: "SQLi"
   type: "series"
   rows: "get-routes"
   cols: "get-unsafe-queries:2"
```

# DEMO:

## “NO CODE” SECURITY OBSERVABILITY!

```
SQLi      Method Call                                                    get-routes            get-unsafe-queries
--------- -------------------------------------------------------------- --------------------- ------------------------------------------
10000     com.acme.ticketbook.Database.queryUnsafe(Database.java:151)  /ticketbook/list.jsp    [SELECT * FROM tickets]

10001     com.acme.ticketbook.Database.updateUnsafe(Database.java:173) /ticketbook/request.jsp [DROP TABLE tickets]

10002     com.acme.ticketbook.Database.updateUnsafe(Database.java:173) /ticketbook/request.jsp [CREATE TABLE tickets ( id INTEGER IDENTITY,
R(64), cc VARCHAR(64), ticket VARCHAR(16) )]
10003     com.acme.ticketbook.Database.updateUnsafe(Database.java:173) /ticketbook/request.jsp [INSERT INTO tickets(name,city,cc,ticket) VA
ton DC', 'Iffmt8k0kz/+BfWvoykdExAreMsTvM1x', '10001')]
10004     com.acme.ticketbook.Database.updateUnsafe(Database.java:173) /ticketbook/request.jsp [INSERT INTO tickets(name,city,cc,ticket) VA
ltimore', '/j7B2e388H3GVMJNNTVRMXD2JAYEf+76', '10002')]
10005     com.acme.ticketbook.Database.updateUnsafe(Database.java:173) /ticketbook/request.jsp [INSERT INTO tickets(name,city,cc,ticket) VA
delphia', 'uWtJbTHcGaGF/bvouf9w5WcVwSqa2Avr', '10003')]
10006     com.acme.ticketbook.Database.updateUnsafe(Database.java:173) /ticketbook/request.jsp [INSERT INTO tickets(name,city,cc,ticket) VA
, 'JFEu+fcb7lwvRJ3KX1DDOWrsqmPDrPvn', '10004')]
10007     com.acme.ticketbook.Database.queryUnsafe(Database.java:151)  /ticketbook/request.jsp [SELECT * FROM tickets]

10008     com.acme.ticketbook.Database.queryUnsafe(Database.java:151)  /ticketbook/check.jsp   [SELECT * FROM tickets WHERE ticket='10001']

10009     com.acme.ticketbook.Database.queryUnsafe(Database.java:151)  /ticketbook/check.jsp   [SELECT * FROM tickets WHERE ticket='1']

10010     com.acme.ticketbook.Database.queryUnsafe(Database.java:151)  /ticketbook/check.jsp   [SELECT * FROM tickets WHERE ticket='2']

10011     com.acme.ticketbook.Database.queryUnsafe(Database.java:151)  /ticketbook/check.jsp   [SELECT * FROM tickets WHERE ticket='3']
```

```
sensors:

- name: "get-routes"
  description: "Identifies the route for this HTTP request"
  methods:
  - "javax.servlet.Servlet.service"
  captures:
  - "#P0.getRequestURI()"

- name: "get-users"
  description: "Identifies user names"
  methods:
  - "javax.servlet.Servlet.service"
  captures:
  - "#P0.getRemoteUser() ?: \"Guest\""

- name: "get-role"
  description: "Identifies roles"
  methods:
  - "javax.servlet.ServletRequest.isUserInRole"
  captures:
  - "#P0"


reports:

 - name: "Test Coverage Matrix"
   type: "compare"
   rows: "get-routes"
   cols: "get-users"

 - name: "Access Control Matrix"
   type: "compare"
   rows: "get-routes"
   cols: "get-role"
```

# DEMO:

# TESTING ACCESS CONTROL!

| Test Coverage Matrix | Guest | UserB | UserC | UserD | UserE |
|---|---|---|---|---|---|
| /ticketbook/accessA.jsp | X | X | | X | X |
| /ticketbook/accessB.jsp | X | X | X | X | |
| /ticketbook/accessC.jsp | X | X | | X | |
| /ticketbook/accessD.jsp | X | X | | X | X |
| /ticketbook/accessE.jsp | X | | | | X |
| /ticketbook/architecture.jsp | X | | | | |
| /ticketbook/cmd.jsp | X | | | | |
| /ticketbook/forward.jsp | X | | | | |
| /ticketbook/hash.jsp | X | | | | |
| /ticketbook/redirect.jsp | X | | | | |
| /ticketbook/xss.jsp | X | | | | |
| /ticketbook/xxe.jsp | X | | | | |

| Access Control Matrix | RoleA | RoleB | RoleC | RoleD | RoleE |
|---|---|---|---|---|---|
| /ticketbook/accessA.jsp | X | | | | |
| /ticketbook/accessB.jsp | | X | | | |
| /ticketbook/accessC.jsp | | | X | | |
| /ticketbook/accessE.jsp | X | | | X | X |

# DEMO:

# MAKING SECURITY OBSERVABLE WITH INSTRUMENTATION

```
- name: "get-ciphers"
  description: "Identifies encryption ciphers"
  methods:
  - "javax.crypto.Cipher.getInstance"
  captures:
  - "#P0"
  matchers:
  - "!null"


- name: "native-libraries"
  description: "Identifies the use of native libraries"
  methods:
  - "java.lang.System.load"
  - "java.lang.System.loadLibrary"
  - "java.lang.System.mapLibraryName"
  captures:
  - "#P0"
  matchers:
  - "!null"


- name: "get-unsafe-queries"
  description: "Identifies unparameterized database queries"
  methods:
  - "java.sql.Statement.execute"
  - "java.sql.Statement.addBatch"
  - "java.sql.Statement.executeQuery"
  - "java.sql.Statement.executeUpdate"
  excludes:
  - "java.sql.PreparedStatement"   #these calls are harmless in PreparedStatement
  captures:
  - "#ARGS"
```

| Test Coverage Matrix | Guest | UserA | UserB | UserC | UserD | UserE |
|---|---|---|---|---|---|---|
| /ticketbook/accessA.jsp | X | X | X | X | X | X |
| /ticketbook/accessB.jsp | X | X | | | | X |
| /ticketbook/accessC.jsp | X | X | | | | X |
| /ticketbook/accessD.jsp | X | X | | | | X |
| /ticketbook/accessE.jsp | X | X | X | | | X |
| /ticketbook/check.jsp | | | | | | X |
| /ticketbook/list.jsp | | | | | | X |
| /ticketbook/profile | | | | | | X |

| Access Control Matrix | RoleA | RoleB | RoleC | RoleD | RoleE |
|---|---|---|---|---|---|
| /ticketbook/accessA.jsp | X | | | | |
| /ticketbook/accessB.jsp | | X | | | |
| /ticketbook/accessC.jsp | | | X | | |
| /ticketbook/accessD.jsp | | | | | |
| /ticketbook/accessE.jsp | X | | | X | X |
| /ticketbook/check.jsp | | | | | |
| /ticketbook/list.jsp | | | | | |
| /ticketbook/profile | | | | | |

| Cipher Matrix | AES | DES | DES/CBC/PKCS5Padding | DESede | PBEWithMD5AndTripleDES |
|---|---|---|---|---|---|
| /ticketbook/accessA.jsp | X | | | | X |
| /ticketbook/accessB.jsp | | X | | | |
| /ticketbook/accessC.jsp | | | X | | |
| /ticketbook/accessD.jsp | | | | | |
| /ticketbook/accessE.jsp | X | | | X | |
| /ticketbook/check.jsp | | X | | | |
| /ticketbook/list.jsp | | X | | | |
| /ticketbook/profile | | X | | | |

| Query Matrix | Query |
|---|---|
| /ticketbook/check.jsp | SELECT * FROM tickets WHERE ticket='OWASP' |
| /ticketbook/list.jsp | SELECT * FROM tickets |
| /ticketbook/profile | INSERT INTO tickets(name,city,cc,ticket) VALUES('OWASP', 'Everywhere', '16/RPW7OlN3H9cJofut3ig==', '10006') |

# DEMO: MAKING SECURITY OBSERVABLE WITH INSTRUMENTATION

```
# Java Sensor Toolkit (JST)
# https://openo11y.org

sensors:

- name: "get-routes"
  description: "Identifies the route for this HTTP request"
  methods:
  - "javax.servlet.Servlet.service"
  captures:
  - "#P0.getRequestURI()"

- name: "block-native-process"
  description: "Blocks attempts to start native processes"
  methods:
  - "java.lang.ProcessBuilder.<init>"
  scopes:
  - "javax.servlet.Servlet.service"
  captures:
  - "#ARGS"
  exception: "Attempt to create ProcessBuilder from within Servlet.service() prevented by JST rule 'block-native-process'"

reports:

- name: "CMDi"
  type: "series"
  rows: "get-routes"
  cols: "block-native-process:13"
```

# DEMO: PREVENTING COMMAND INJECTION EXPLOITS WITH INSTRUMENTATION!

# DEMO: PREVENTING COMMAND INJECTION

Create accurate appsec visibility in OPS


## HTTP Status 500 – Internal Server Error

**Type** Exception Report

**Message** com.contrastsecurity.advice.SensorException: Attempt to create ProcessBuilder from within Servlet.service() prevented by JST rule 'block-native-process'

**Description** The server encountered an unexpected condition that prevented it from fulfilling the request.

**Exception**

```
org.apache.jasper.JasperException: com.contrastsecurity.advice.Sensor
        org.apache.jasper.servlet.JspServletWrapper.handleJspExceptic
        org.apache.jasper.servlet.JspServletWrapper.service(JspServle
        org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServle
        org.apache.jasper.servlet.JspServlet.service(JspServlet.java:
        javax.servlet.http.HttpServlet.service(HttpServlet.java:741)
        org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter
```

**Root Cause**

```
20-Apr-2020 12:34:43.750 INFO [localhost-start
.deployDirectory Deploying web application dir
8.5.43/webapps/host-manager]
20-Apr-2020 12:34:43.762 WARNING [localhost-st
Config.validateSecurityRoles Security role nam
being defined in a <security-role>
20-Apr-2020 12:34:43.767 INFO [localhost-start
.deployDirectory Deployment of web application
cat-8.5.43/webapps/host-manager] has finished
20-Apr-2020 12:34:43.770 INFO [main] org.apach
colHandler ["http-nio-8080"]
20-Apr-2020 12:34:43.780 INFO [main] org.apach
colHandler ["ajp-nio-8009"]
20-Apr-2020 12:34:43.782 INFO [main] org.apach
Execing: /usr/bin/open -a Calculator
WARNING: Access to java.lang.ProcessBuilder.st
e() scope prevented by security sandbox.
ProtocolHandler ["http-nio-8080"]
20-Apr-2020 15:01:24.893 INFO [Thread-5] org.apache.coyote.AbstractProtocol.pause Pausing Pr
otocolHandler ["ajp-nio-8009"]
```

# DEMO: PREVENTING EXPRESSION LANGUAGE INJECTION EXPLOITS WITH INSTRUMENTATION!

```
sensors:

- name: "get-routes"
  description: "Identifies the route for this HTTP request"
  methods:
  - "javax.servlet.Servlet.service"
  captures:
  - "#P0.getRequestURI()"

- name: "sandbox-expressions"
  description: "Prevents harmful methods from being used during expresssion evaluation"
  methods:
  - "java.lang.ProcessBuilder.<init>"
  - "java.io.Socket.<init>"
  scopes:
  - "javax.el.ValueExpression.getValue"
  captures:
  - "#P0"
  exception: "Attempt to escape expression language sandbox prevented by JST rule 'sandbox-expressions'"

reports:

- name: "Expression Language Injection Attempt Log"
  type: "series"
  rows: "get-routes"
  cols: "sandbox-expressions:13"
```

# DEMO: PREVENTING EXPRESSION LANGUAGE INJECTION

Runtime Protection!


## HTTP Status 500 – Internal Server Error

**Type** Exception Report

**Message** javax.el.ELException: com.contrastsecurity.advice.SensorException: Attempt to escape expression language sandbox prevented by JST rule 'sandbox-expressions'

**Description** The server encountered an unexpected condition that prevented it from fulfilling the request.

**Exception**

```
org.apache.jasper.JasperException: javax.el.ELException: com.contrastsecurity.
        org.apache.jasper.servlet.JspServletWrapper.handleJspException(JspServ
        org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.
        org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:38
        org.apache.jasper.servlet.JspServlet.service(JspServlet.java:330)
        javax.servlet.http.HttpServlet.service(HttpServlet.java:741)
        org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)
```

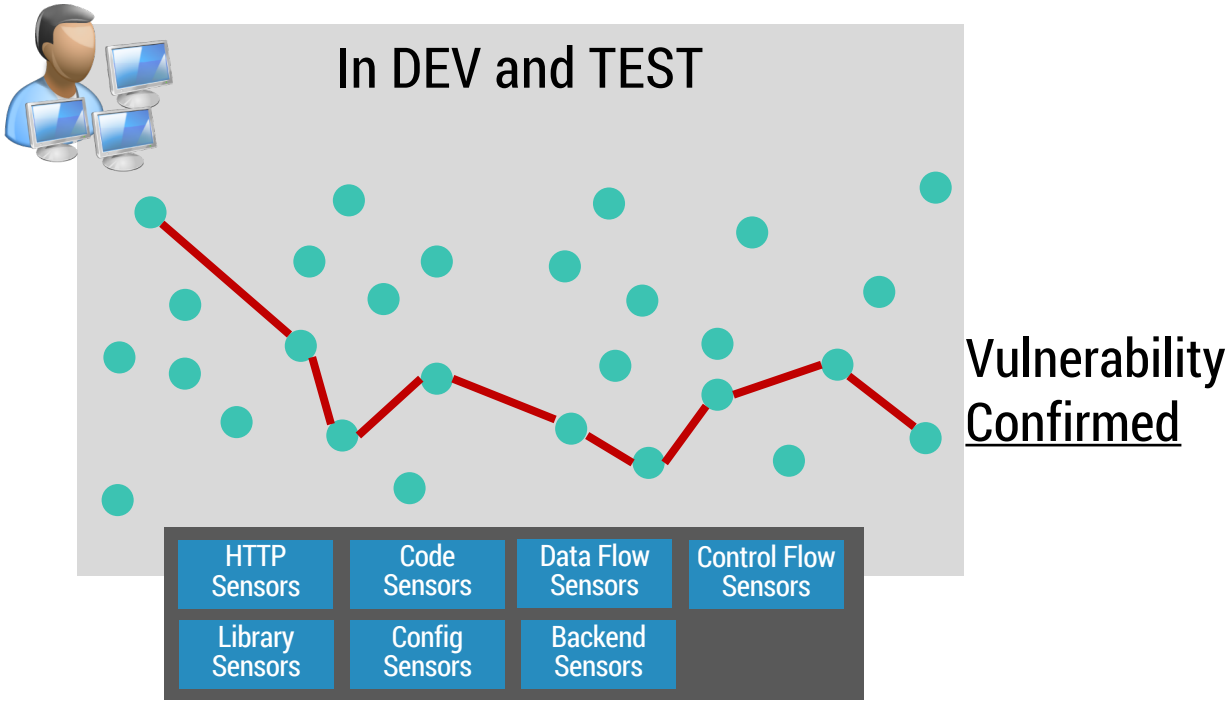
**Root Cause**

```
javax.el.ELException: com.contrastsecurity.advice.SensorException: Attempt to
```

```
20-Apr-2020 12:34:43.750 INFO [localhost-sta
.deployDirectory Deploying web application d
8.5.43/webapps/host-manager]
20-Apr-2020 12:34:43.762 WARNING [localhost-
Config.validateSecurityRoles Security role r
being defined in a <security-role>
20-Apr-2020 12:34:43.767 INFO [localhost-sta
.deployDirectory Deployment of web applicati
cat-8.5.43/webapps/host-manager] has finishe
20-Apr-2020 12:34:43.770 INFO [main] org.apa
colHandler ["http-nio-8080"]
20-Apr-2020 12:34:43.780 INFO [main] org.apa
colHandler ["ajp-nio-8009"]
20-Apr-2020 12:34:43.782 INFO [main] org.apa
Execing: /usr/bin/open -a Calculator
WARNING: Access to java.lang.ProcessBuilder.
e() scope prevented by security sandbox.
ProtocolHandler ["http-nio-8080"]
20-Apr-2020 15:01:24.893 INFO [Thread-5] org.apache.coyote.AbstractProtocol.pause Pausing Pr
otocolHandler ["ajp-nio-8009"]
```

# IAST

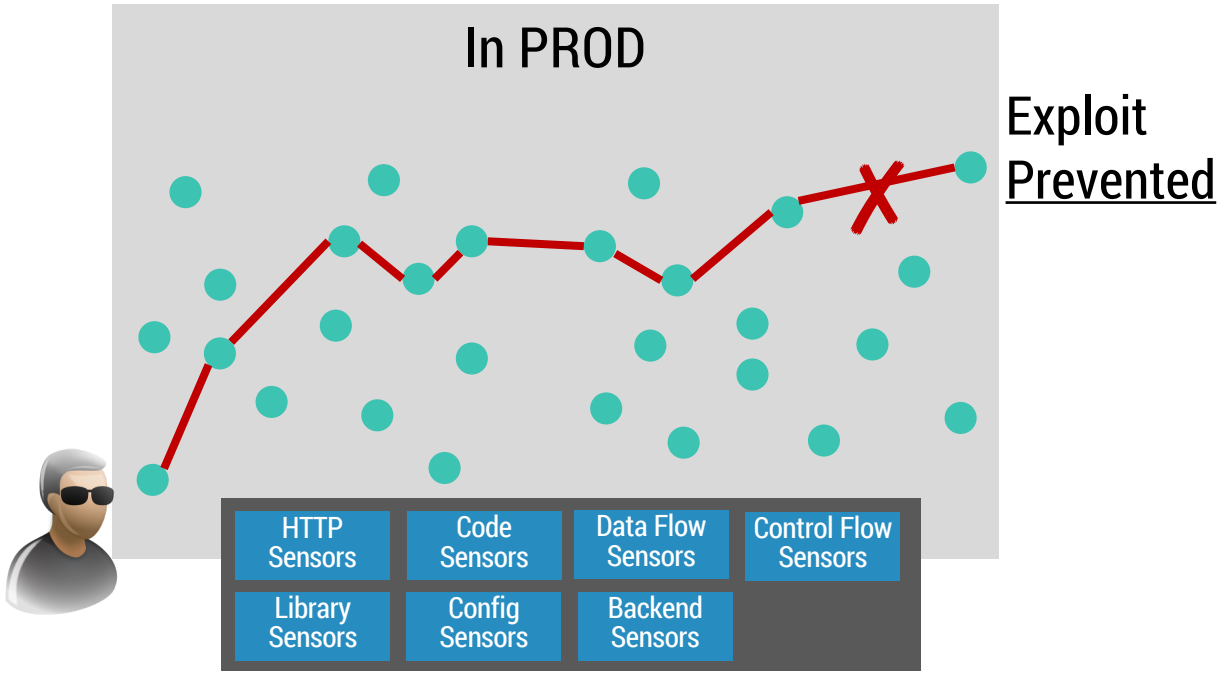
Interactive Application Security Testing detects vulnerabilities in both custom code and libraries during normal use




Vulnerability Confirmed

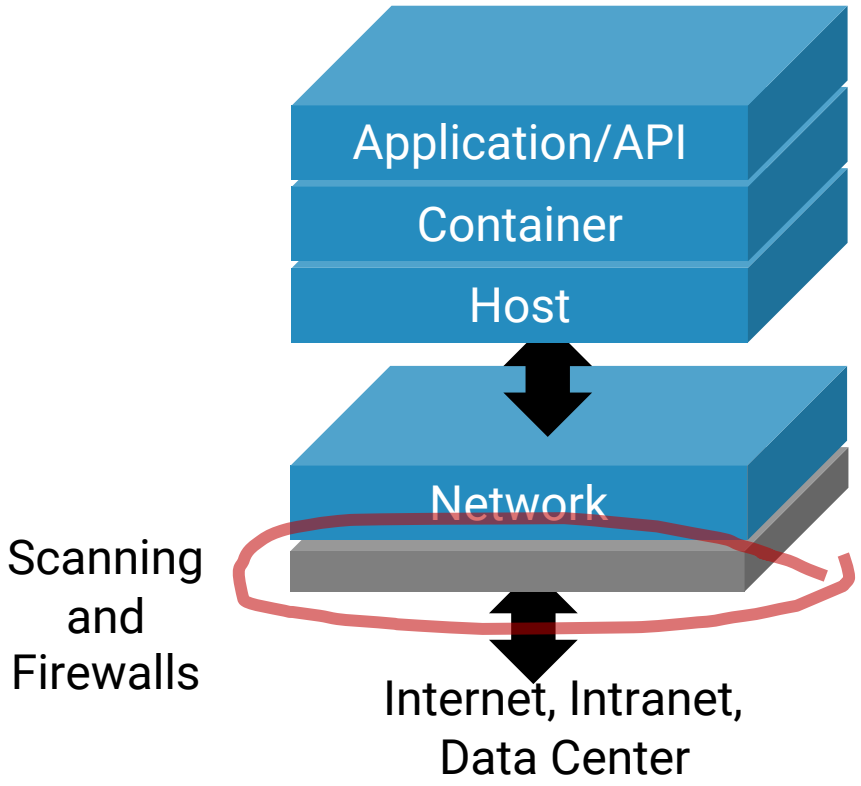
# RASP

Runtime Application Self-Protection detects attacks and prevents exploits in both custom code and libraries

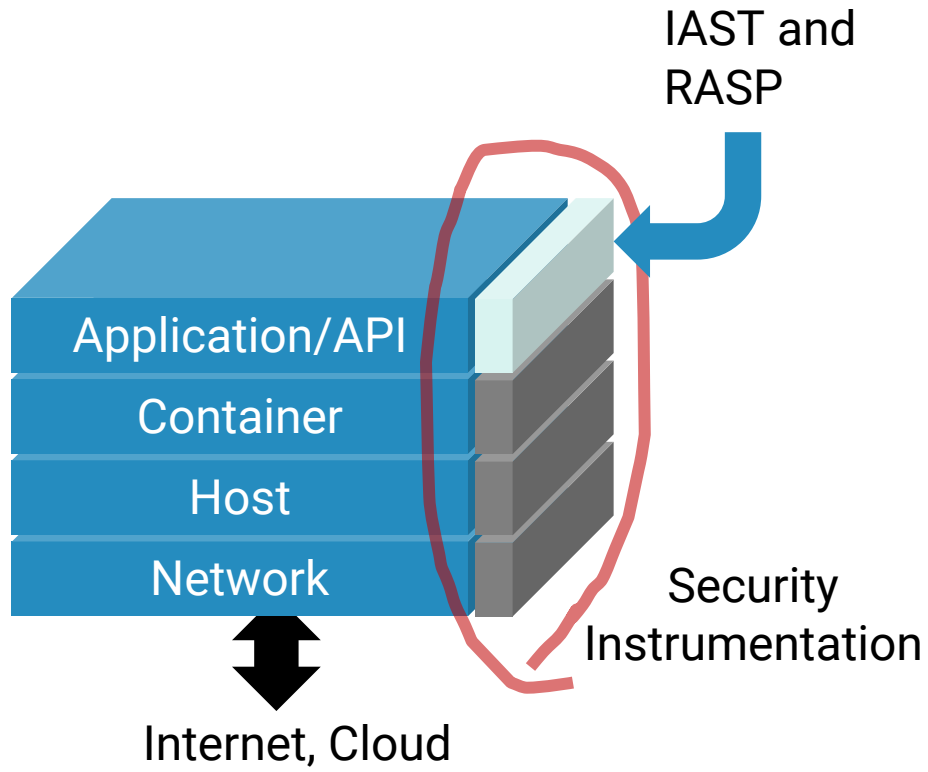



Exploit Prevented

# SECURITY WORKS BETTER FROM INSIDE-OUT




**Yesterday**: Scanning and firewalling at network layer




**Today**: Security Instrumentation means accuracy, speed, scalability

# SECURITY INSTRUMENTATION STANDARDS

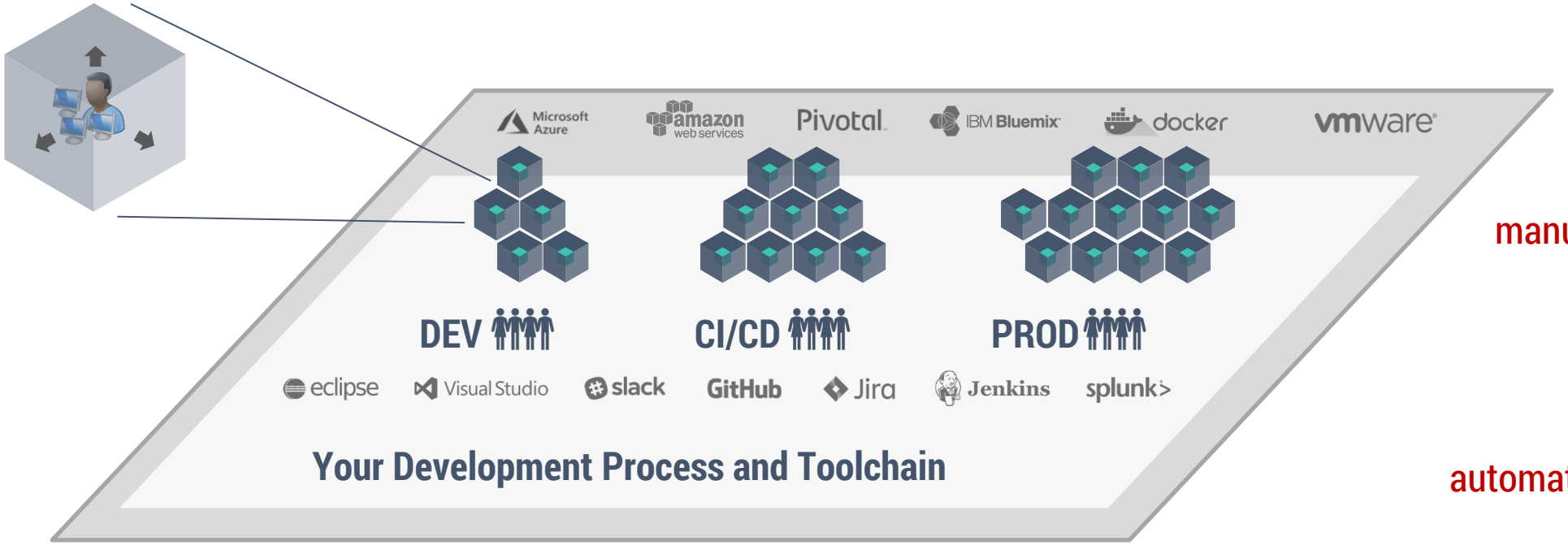

NIST 800-53

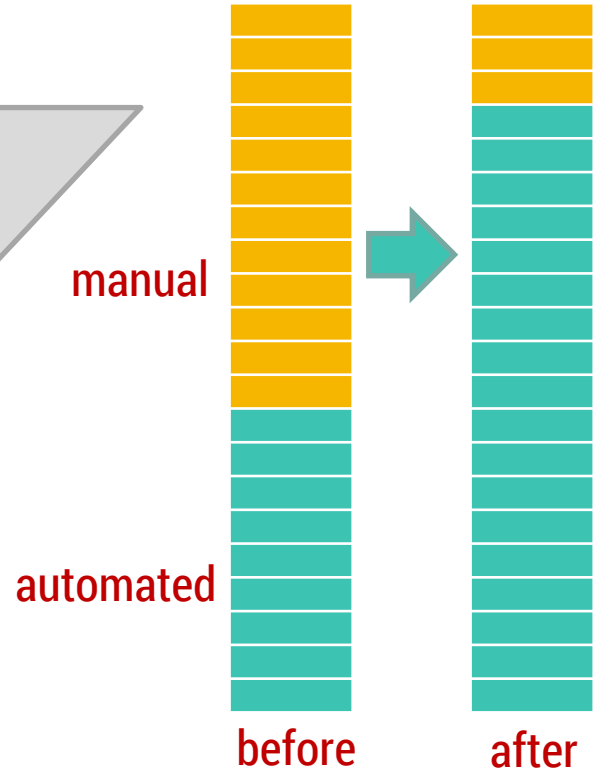
- **SA-11(9) | INTERACTIVE APPLICATION SECURITY TESTING**
  Require the developer of the system, system component, or system service to employ **interactive application security testing (IAST)** tools to identify flaws and document the results.
- **SI-7(17) | RUNTIME APPLICATION SELF-PROTECTION**
  Implement [*Assignment: organization-defined controls*] for application **self-protection at runtime (RASP)**.


SOFTWARE SECURITY STANDARD

- **SSS 9.1 | RUNTIME APPLICATION SELF-PROTECTION**
  The software detects and alerts upon detection of anomalous behavior, such as changes in postdeployment configurations or obvious **attack behavior**.
- **SSS 10.2 | INTERACTIVE APPLICATION SECURITY TESTING**
  Vulnerabilities in the software and third-party components are tested for and fixed prior to release using … [techniques including]… **interactive application security testing (IAST)** ….

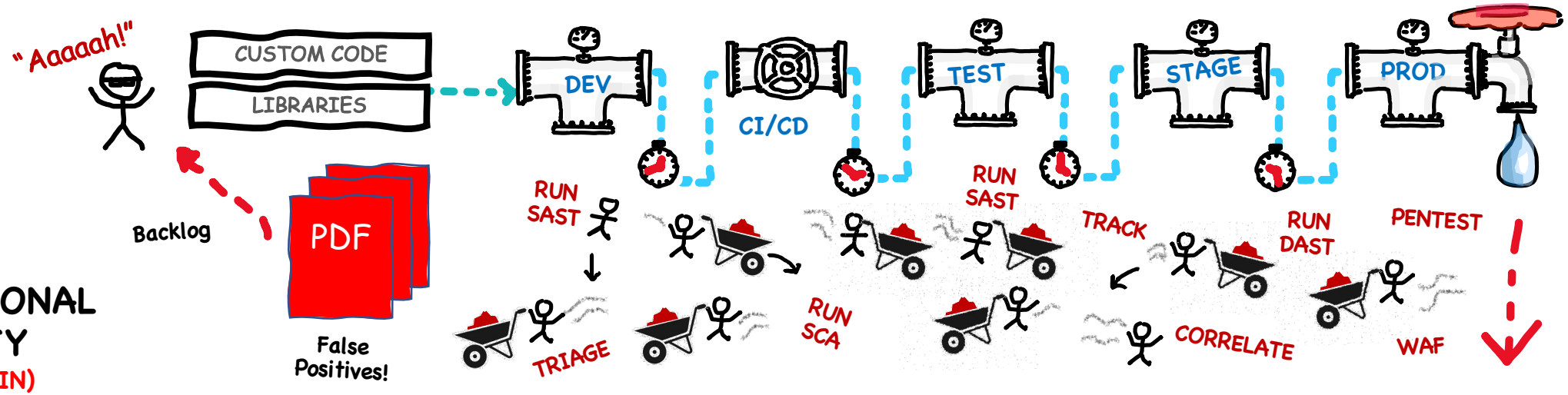
# TURNING SECURITY INTO CODE



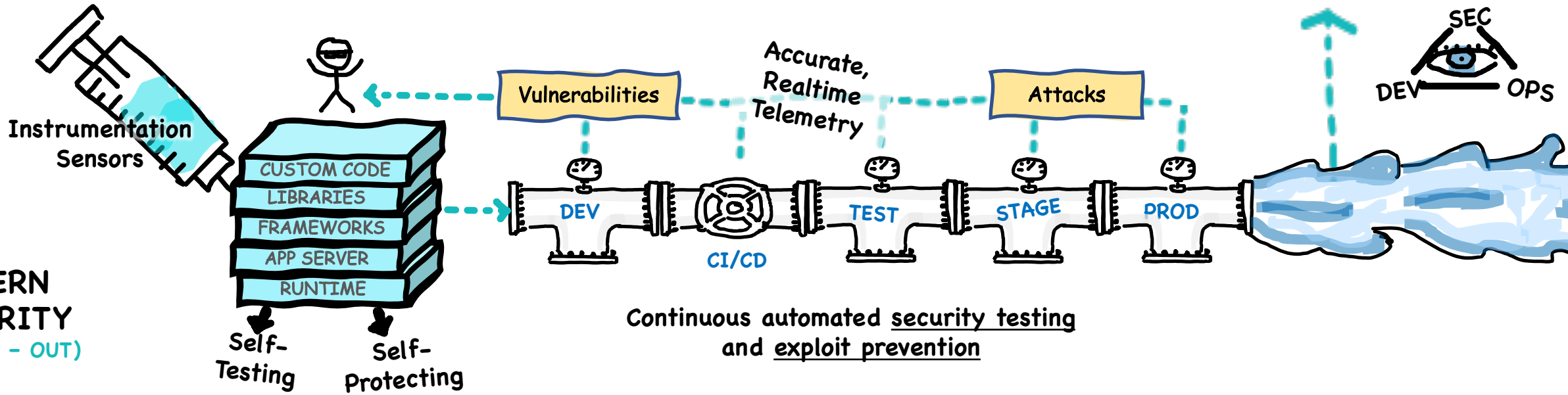

## ...ACTUALLY, THE REAL VULNERABILITIES ARE IN THE PIPELINE ITSELF

# APPSEC'S ABILITY TO
## DELIVER VALUE TO CUSTOMERS, FASTER




**TRADITIONAL SECURITY**
(OUTSIDE - IN)

FLOW
(BUSINESS VALUE)




**MODERN SECURITY**
(INSIDE - OUT)

Continuous automated security testing and exploit prevention


Development
Harmony
Security
OWASP
Open Web Application
Security Project

# CONTRAST
COMMUNITY EDITION

## A totally free and full-strength application security platform

CREATE FREE ACCOUNT

https://www.contrastsecurity.com/ce

**AVAILABLE NOW**

Java

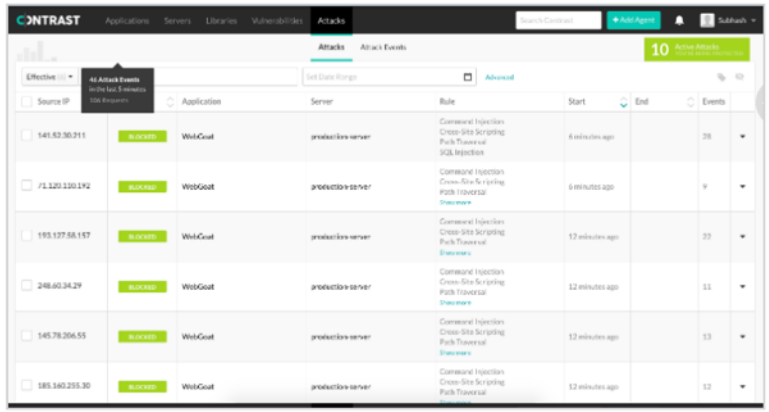
Microsoft .NET

**COMING SOON**

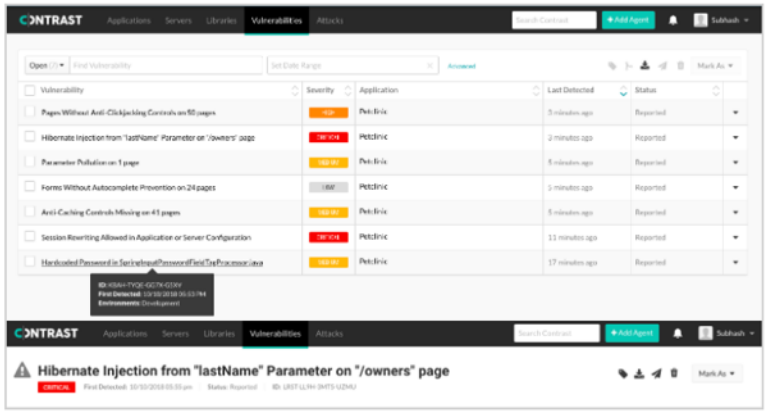
node JS

python

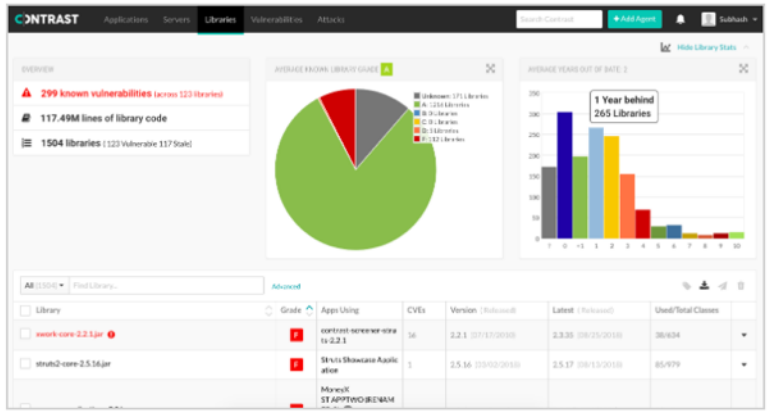
Ruby



Protect against attacks with RASP




Find vulnerabilities with IAST




Secure open-source with SCA

# ASK ME ANYTHING

Jeff Williams, Cofounder and CTO

@planetlevel

OWASP
Open Web Application Security Project