CONTRAST SECURITY

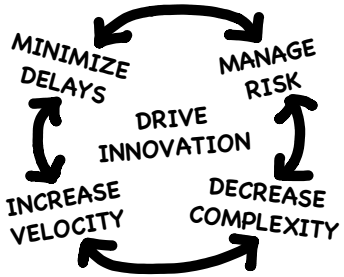
HOW TO BUILD AWESOME SECURITY INSTRUMENTATION TO AUTOMATE APPSEC TESTING AND PROTECTION

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

OWASP Open Web Application Security Project

THE GOAL IS SIMPLE:

DELIVER VALUE TO CUSTOMERS, FASTER

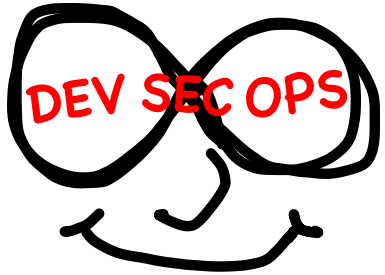


MAJOR INITIATIVES:

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

DEVELOPMENT'S RESPONSE:

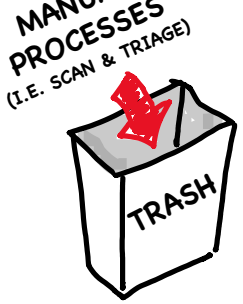
FLOW | FEEDBACK | CONTINUOUS EXPERIMENTATION & LEARNING



+

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

=



=

WHY NOT?

200X MORE FREQUENT ~~DEPLOYS~~ ANALYSIS REMEDIATION

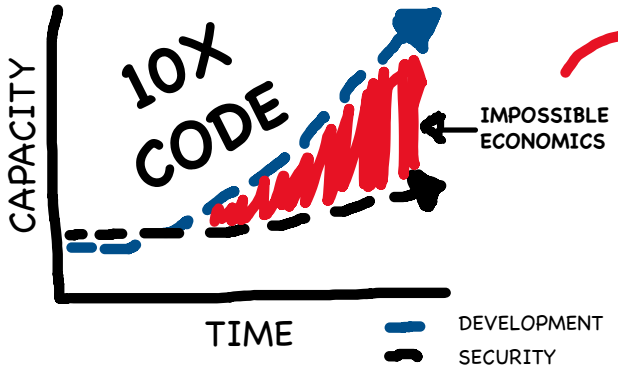
24X FASTER ~~RECOVERY~~ TIMES

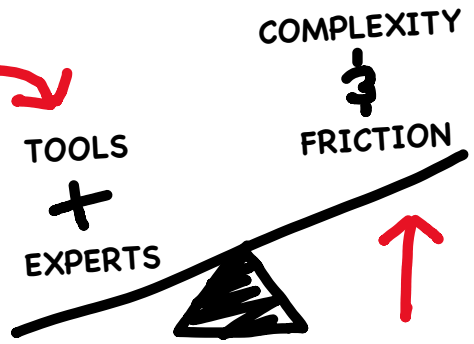
3X LOWER ~~CHANGE FAILURE RATE~~ SECURITY BACKLOG SECURE

10X AMOUNT OF CODE

SECURITY'S CHALLENGE:

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



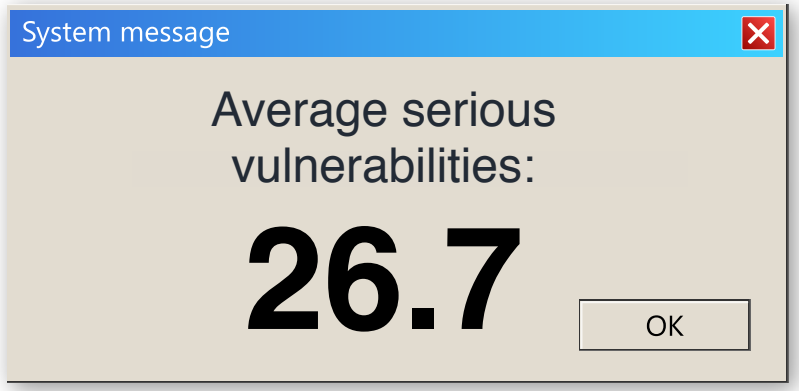


LEGACY OUTCOMES:

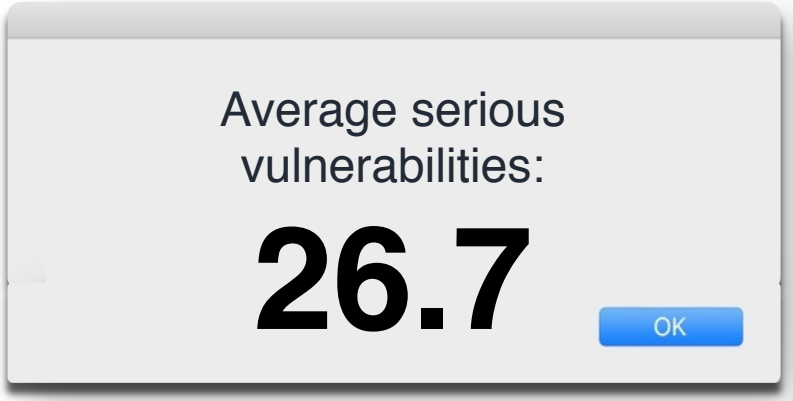
1. SECURITY SLOWS VALUE DELIVERY STREAM
2. DEVELOPMENT BYPASSES SECURITY & RELEASE INSECURE CODE
3. RUN SECURITY ASYNCHRONOUS TO SDLC AS A COMPLIANCE CHECK

MODERN SECURITY CAN BE MORE!

IS IT WORKING YET?



2002



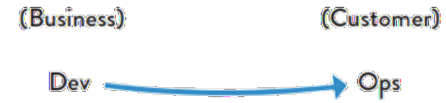
2019

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*

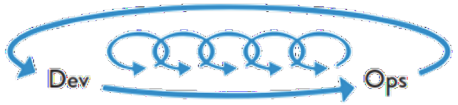
DEVOPS

1. Establish work flow



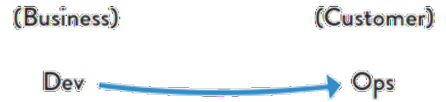
2. Ensure instant feedback

3. Culture of experimentation



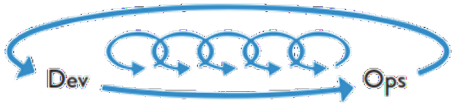
DEVSECOPS

1. Establish security work flow



2. Ensure instant security feedback

3. Build a security culture

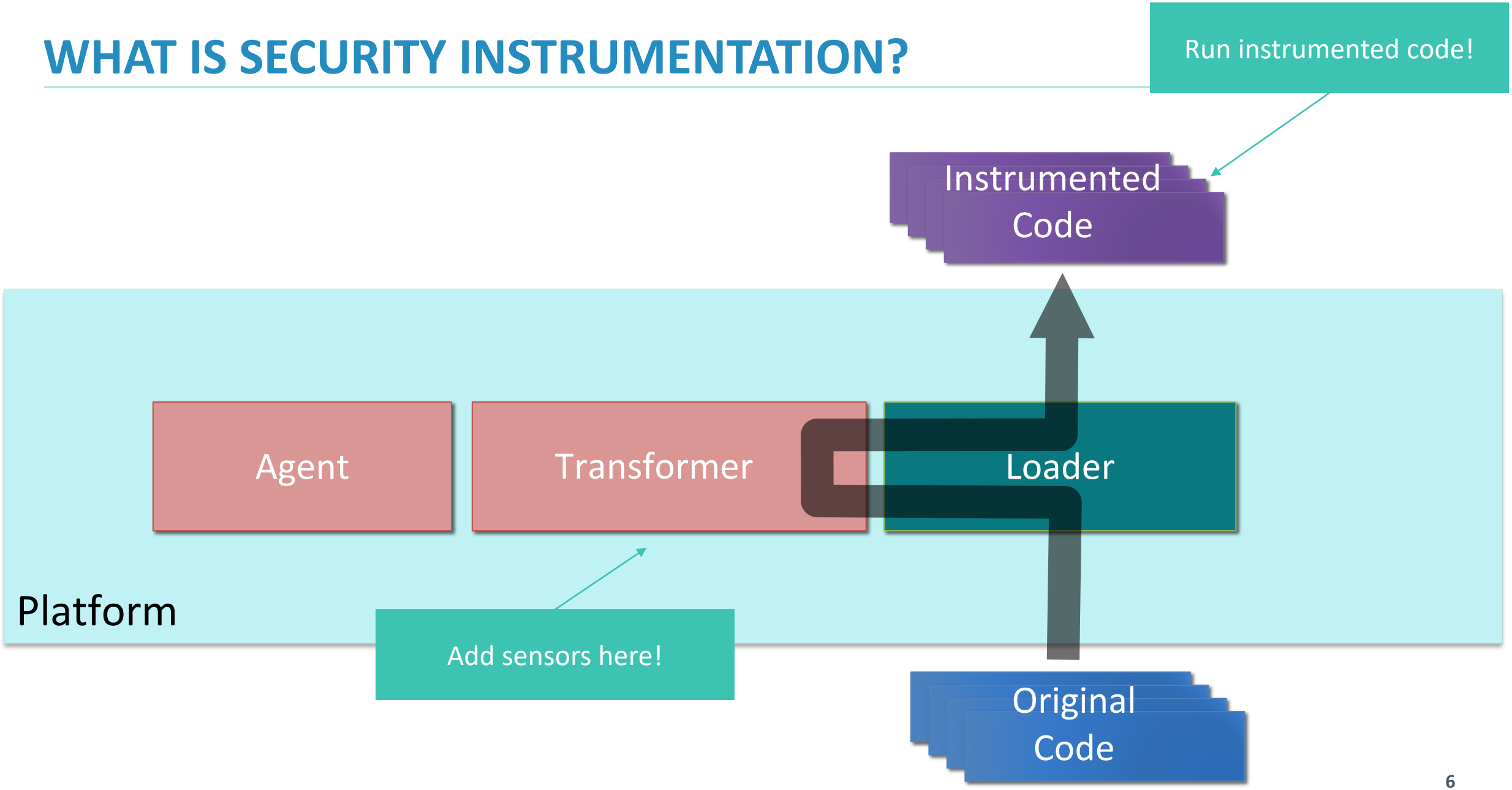


https://itrevolution.com/the-three-ways-principles-underpinning-devops/

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...

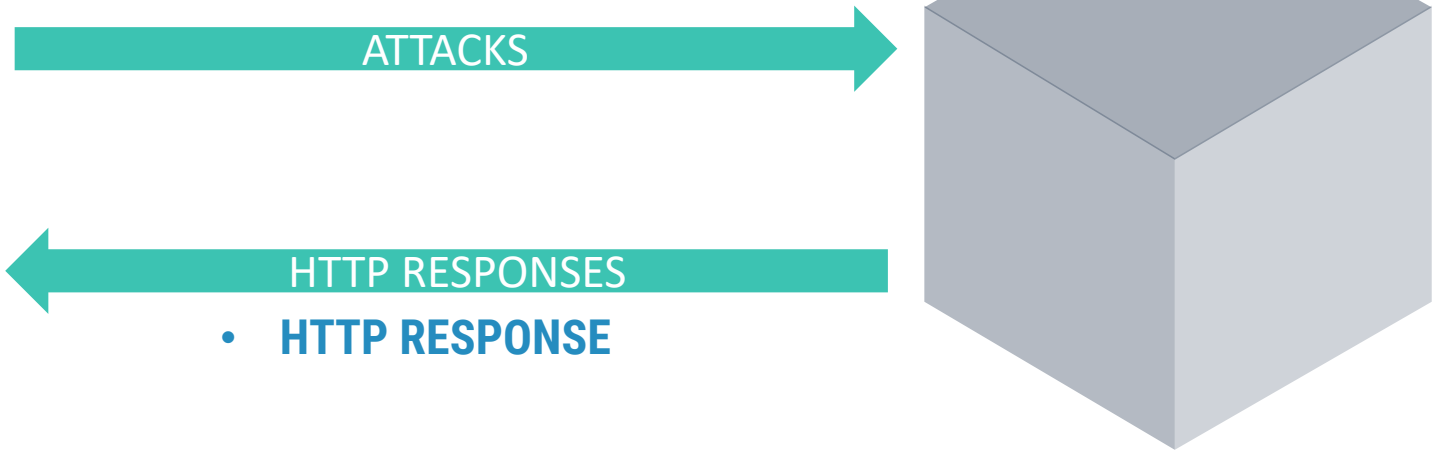
Usage of interface net.bytebuddy.asm.Advice$Return Done

```
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…

2. MAKING SECURITY OBSERVABLE WITH INSTRUMENTATION

```
28
29          // insert route sensor into Servlet.service method
30          .type(hasGenericSuperType(named("javax.servlet.Servlet")))
31          .transform((b, td, cl, m) -> b.visit(Advice.to(RouteAdvice.class)
32                  .on(named("service"))))
33
34          // insert role sensor into ServletRequest.isUserInRole method
35          .type(hasGenericSuperType(named("javax.servlet.ServletRequest")))
36          .transform((b, td, cl, m) -> b.visit(Advice.to(RoleAdvice.class)
37                  .on(named("isUserInRole"))))
38
39          // insert cipher sensor into Cipher.getInstance method
40          // Note: agent must be on bootstrap classloader
41          .type(hasGenericSuperType(named("javax.crypto.Cipher")))
42          .transform((b, td, cl, m) -> b.visit(Advice.to(CipherAdvice.class)
43                  .on(named("getInstance"))))
44
45          // insert query sensor into  method
46          .type(hasGenericSuperType(named("java.sql.Statement")))
47          .transform((b, td, cl, m) -> b.visit(Advice.to(QueryAdvice.class)
48                  .on(named("execute")
49                  .or(named("executeQuery")
50                  .or(named("executeUpdate"))))))
51
```

More sensors!

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

3. PREVENTING EXPLOITS WITH INSTRUMENTATION

SandboxAgent.java

```
16  public static void premain(String arg, Instrumentation inst) {
17      System.out.println( "Security Sandbox Agent installed");
18      System.getProperties().put("scope", new ThreadLocal<>().withInitial(() -> 0));
19      new AgentBuilder.Default().ignore(none())
20          .type(hasGenericSuperType(named("javax.servlet.Servlet")))
21          .transform((b, td, cl, m) -> b.visit(Advice.to(ServiceAdvice.class).on(named("service"))))
22          .type(hasGenericSuperType(named("java.lang.ProcessBuilder")))
23          .transform((b, td, cl, m) -> b.visit(Advice.to(SandboxAdvice.class).on(named("start"))))
24          .installOn(inst);
25  }
26
27  public static class SandboxAdvice {
28      @Advice.OnMethodEnter
29      public static void enter(@Origin Class c, @Origin Method m) {
30          boolean block = false;
31          try {
32              block = ((ThreadLocal<Integer>)System.getProperties().get("scope")).get() > 0;
33          } catch( Throwable t ) {
34              System.err.println( "WARNING: Sensor malfunction" );
35              t.printStackTrace();
36          } finally {
37              if ( block ) {
38                  String msg = "Access to " + c.getName() + "." + m.getName() + "() from within javax.servlet.Servlet.serv
39                  System.err.println( "WARNING: " + msg );
40                  throw new SecurityException( msg );
41              }
42          }
43      }
44  }
```

Start scope when you enter .service()

Add scope check to ProcessBuilder.start()

Block if start() occurs while inside service()

DEMO: PREVENTING COMMAND INJECTION

Create accurate appsec visibility in OPS

```
20-Apr-2020 12:34:43.750 INFO [localhost-star
.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.a
otocolHandler ["ajp-nio-8009"]
```

localhost:8080/ticketbook/cmd.jsp

HTTP Status 500 – Internal Server Error

Type Exception Report

Message java.lang.SecurityException: Access to java.lang.ProcessBuilder.start() from within javax.servlet.Servlet.service() scope prevented by security sandbox.

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

Exception

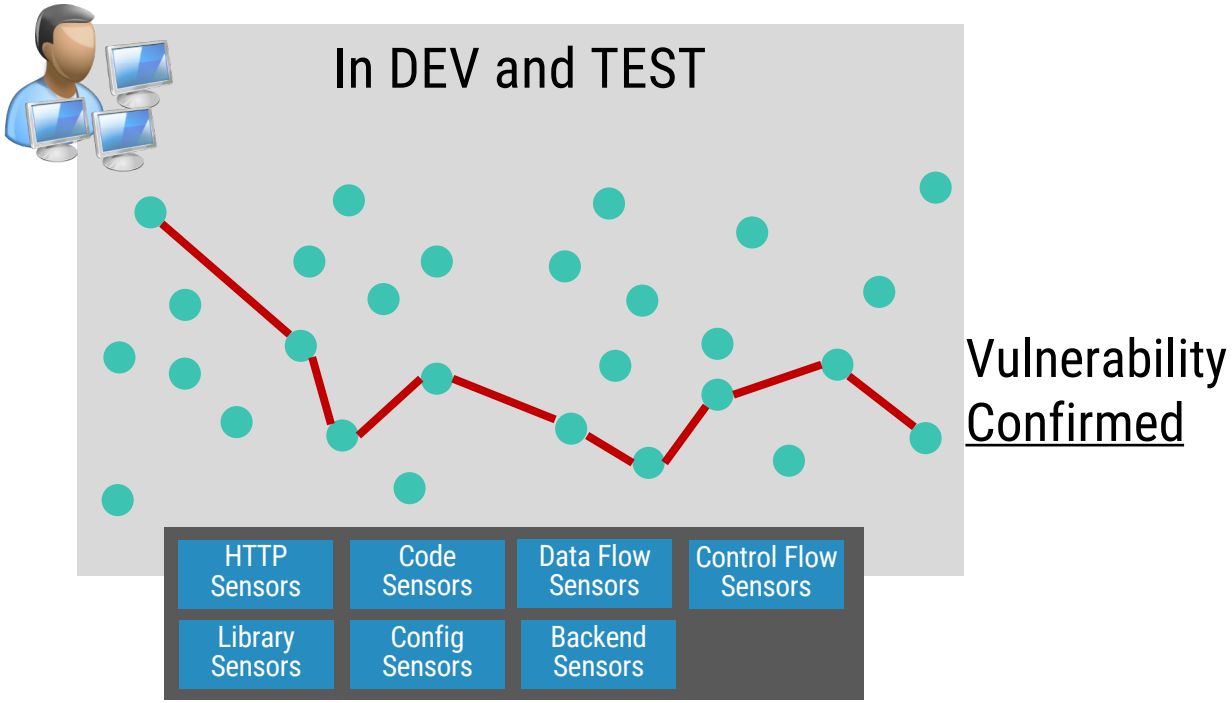
```
org.apache.jasper.JasperException: java.lang.SecurityException: Access to java.la
        org.apache.jasper.servlet.JspServletWrapper.handleJspException(JspServlet
        org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.jav
        org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:386)
        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

```
java.lang.SecurityException: Access to java.lang.ProcessBuilder.start() from with
        java.base/java.lang.ProcessBuilder.start(ProcessBuilder.java:1073)
        java.base/java.lang.Runtime.exec(Runtime.java:591)
        java.base/java.lang.Runtime.exec(Runtime.java:415)
        java.base/java.lang.Runtime.exec(Runtime.java:312)
        org.apache.jsp.cmd_jsp._jspService(cmd_jsp.java:135)
```

IAST

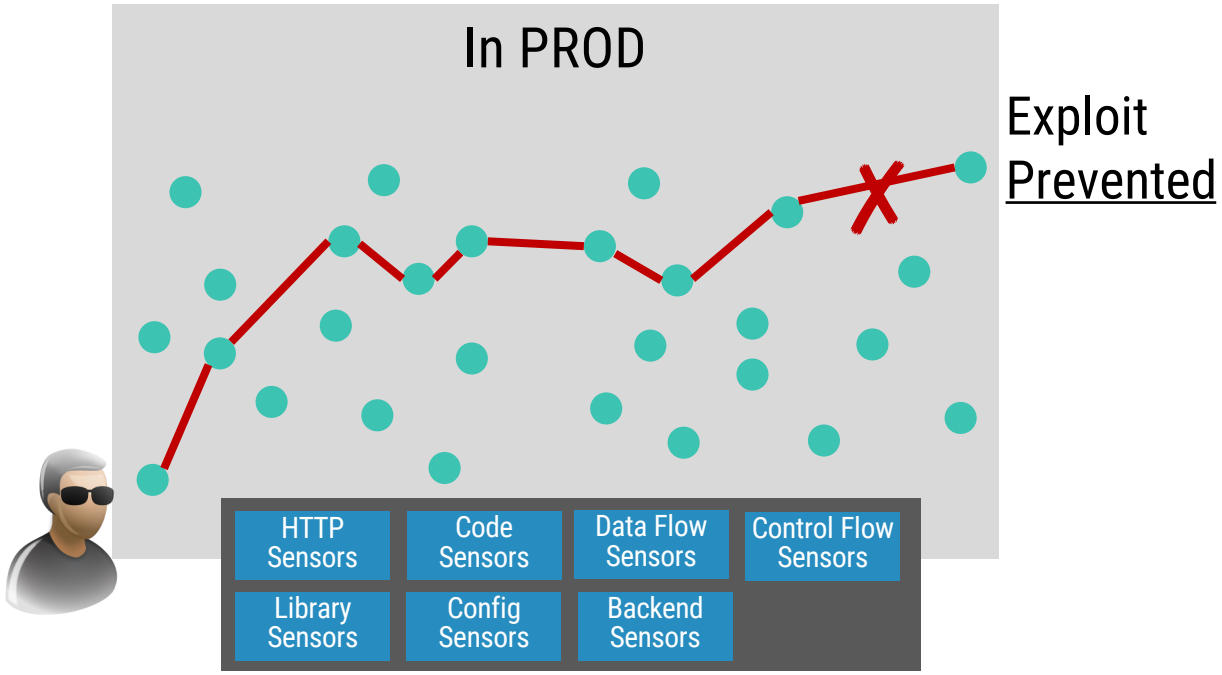
Interactive Application Security Testing detects vulnerabilities in both custom code and libraries during normal use



Vulnerability Confirmed

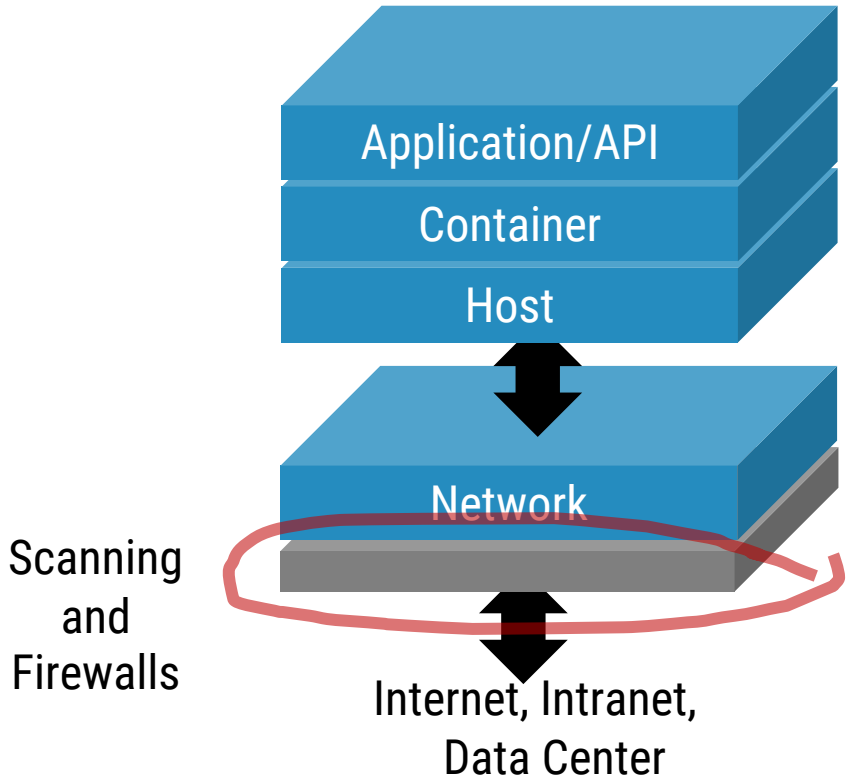
RASP

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

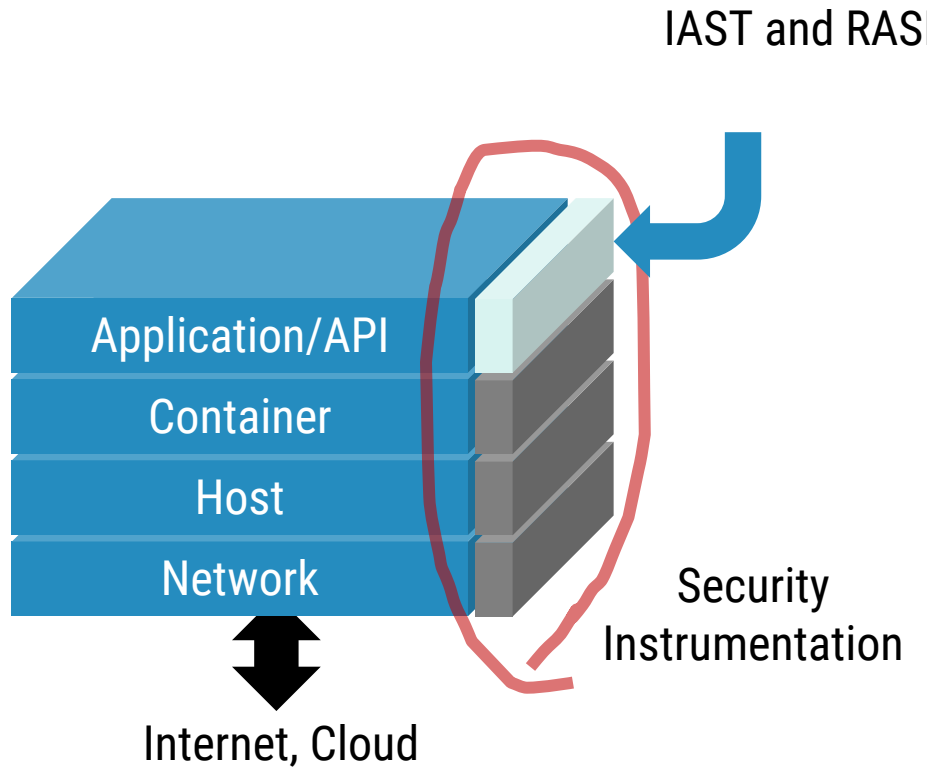


Exploit Prevented

SECURITY WORKS BETTER FROM INSIDE-OUT



Yesterday: Scanning and firewalling at network layer



Today: Security Instrumentation means accuracy, speed, scalability

SECURITY INSTRUMENTATION STANDARDS

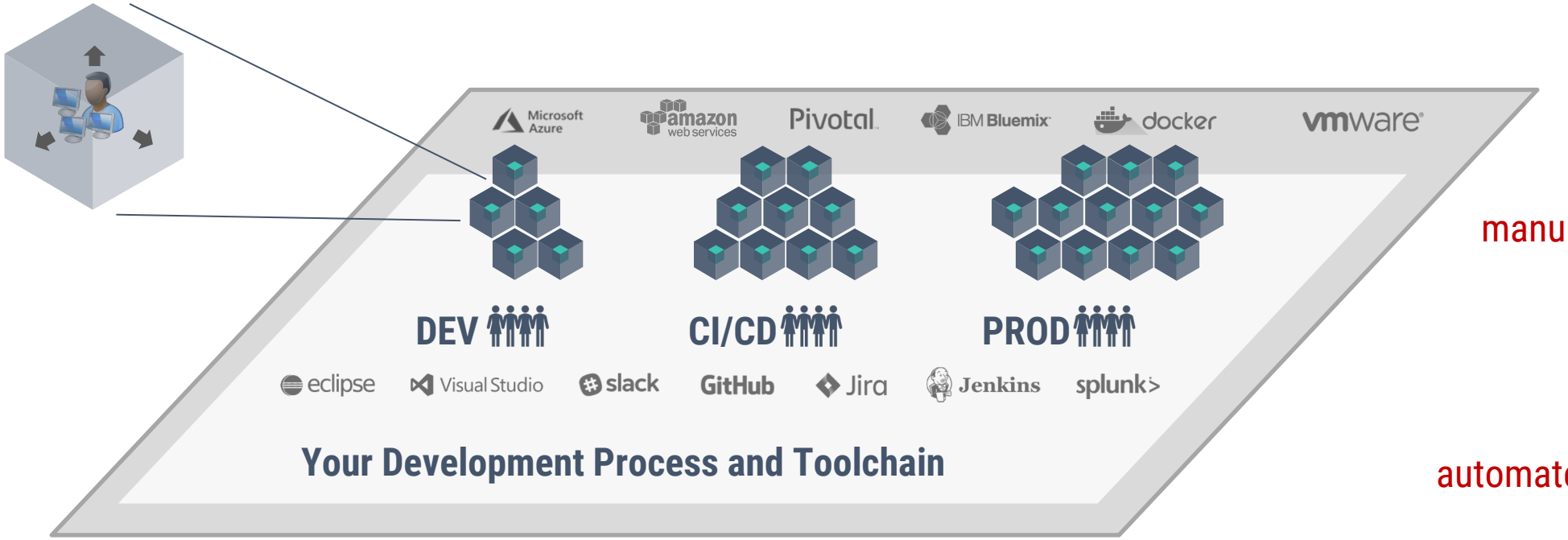
NIST 800-53

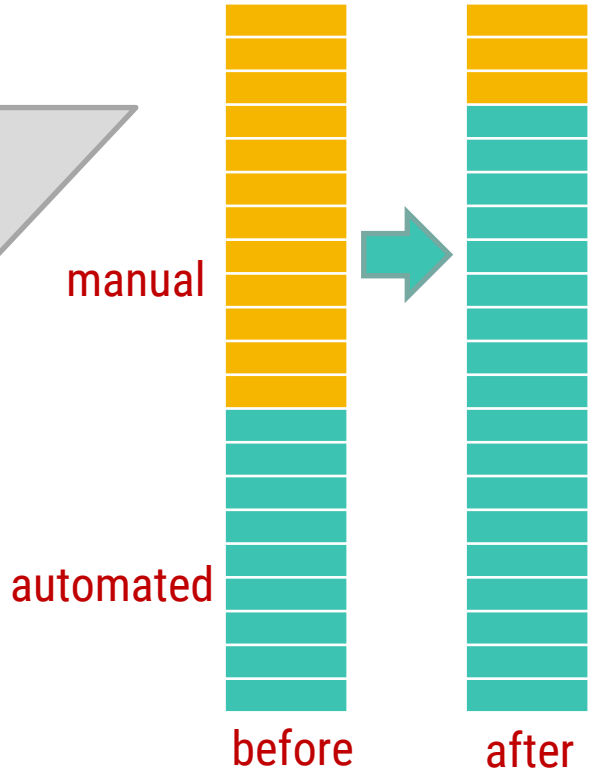
- **SA-11(9) | INTERACTIVE APPLICATION SECURITY TESTING**
  Require the developer of the system, system component, or system service to employ **interactive application security testing (IAST)** tools to identify flaws and document the results.
- **SI-7(17) | RUNTIME APPLICATION SELF-PROTECTION**
  Implement [*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)** ….

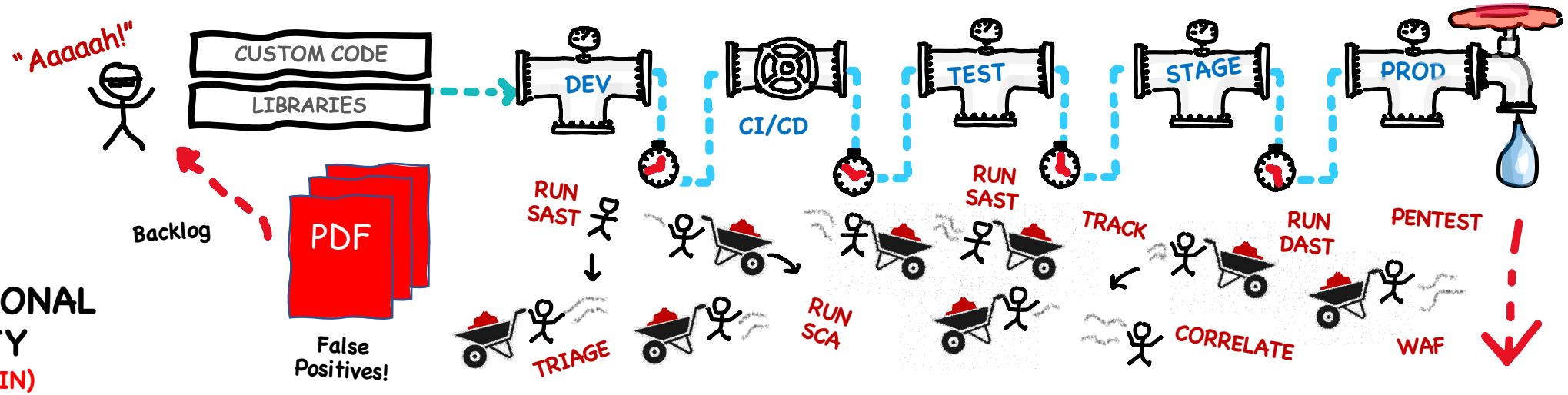
TURNING SECURITY INTO CODE





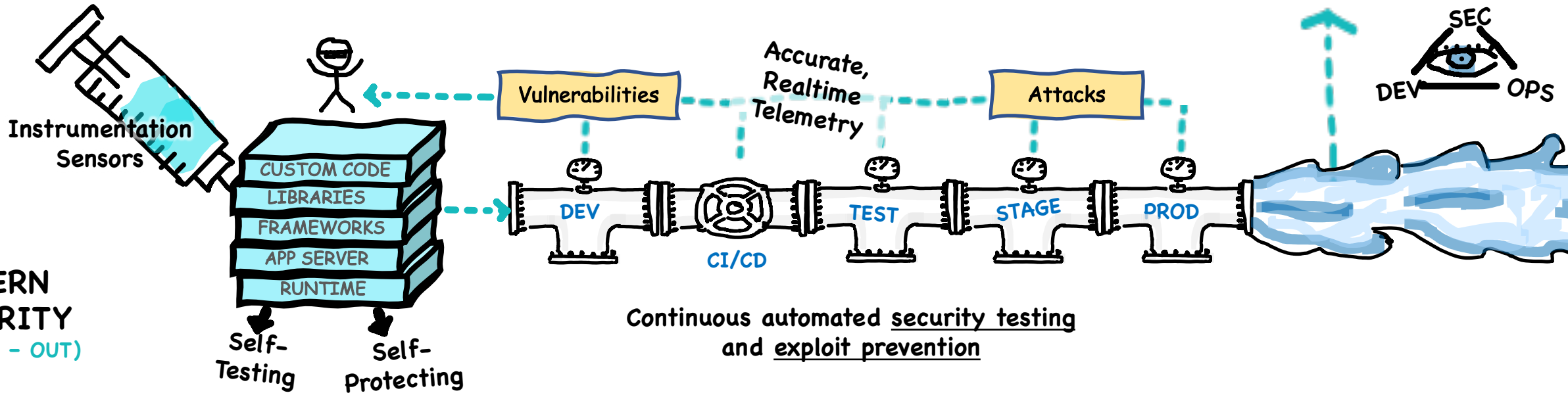
…ACTUALLY, THE REAL VULNERABILITIES ARE IN THE PIPELINE ITSELF

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

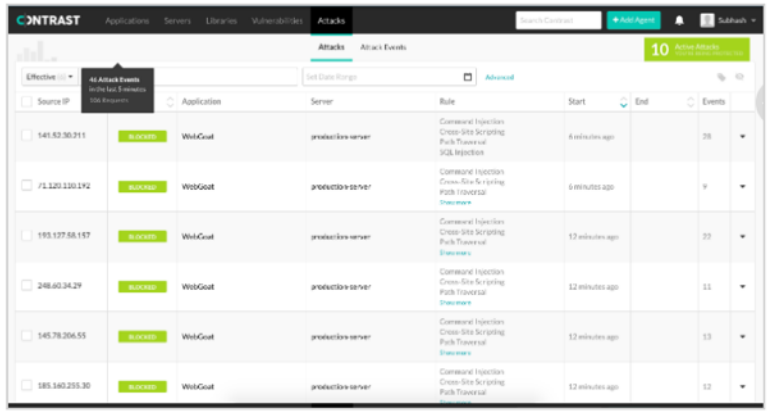
Microsoft .NET

COMING SOON

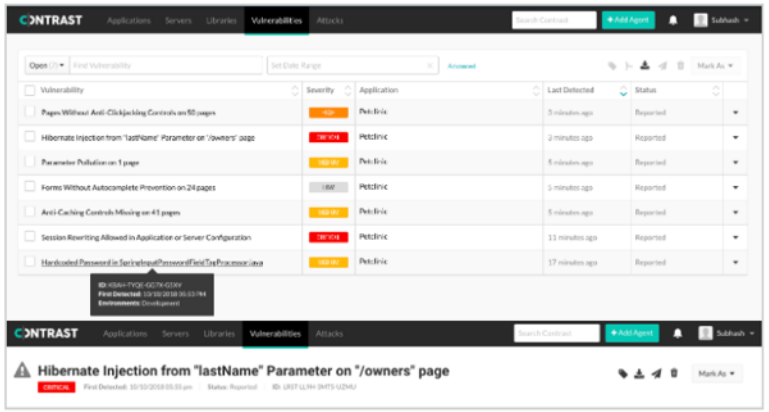
node JS

python

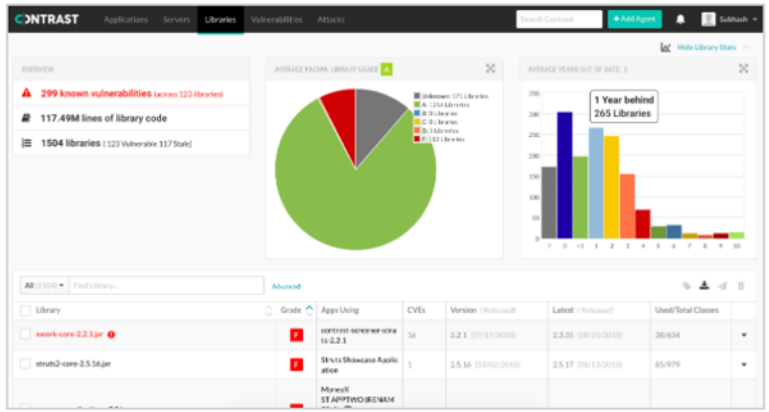
Ruby



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