Jeff Williams, Co-founder and CTO

Contrast Security

@planetlevel

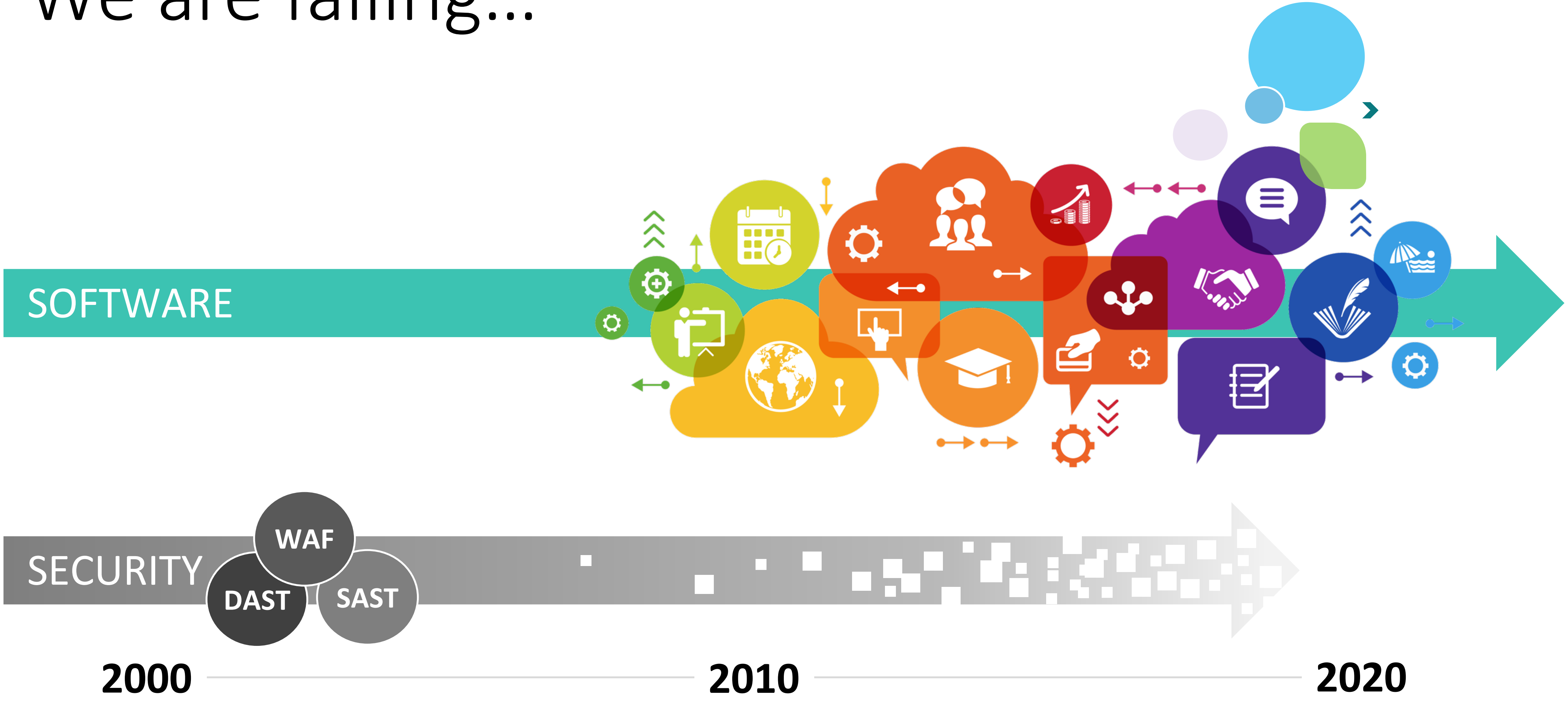
Turning
Security into Code
with
Dynamic Binary Instrumentation

CONTRAST SECURITY

OWASP

March 2017

We are failing…



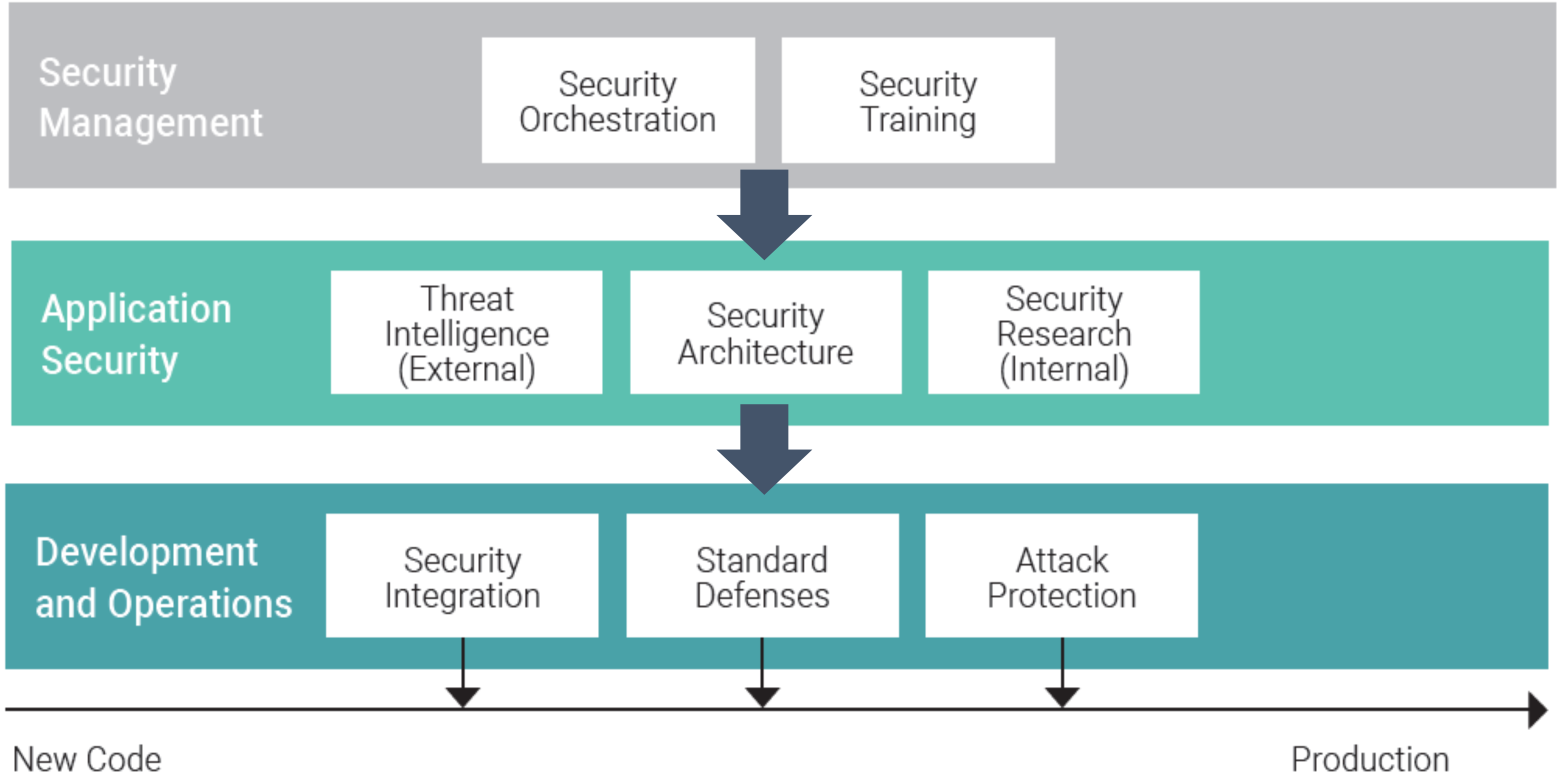
You can’t scale appsec without highly accurate tools

(both true positives and true negatives)

Because inaccuracies require experts…

…and experts don’t scale.

By turning security into code
--> we can get speed, coverage, and accuracy
--> which allows us to scale



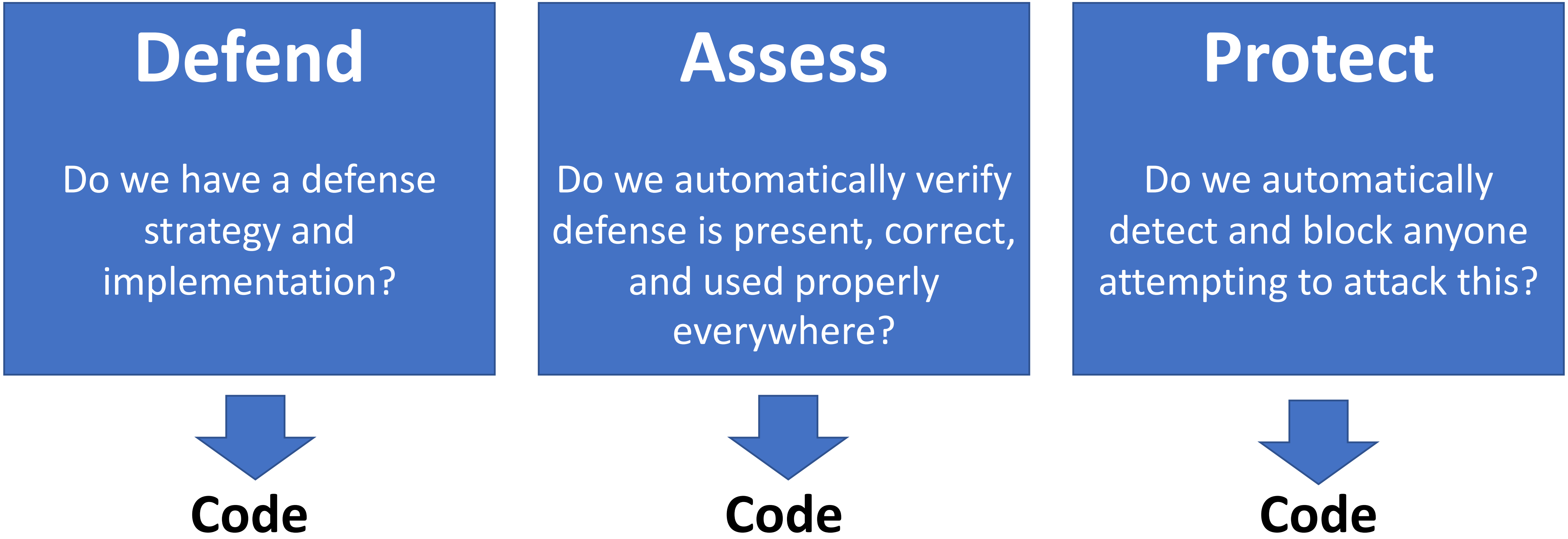
Continuous Application Security

Level 3: Management makes informed decisions with detailed security analytics

Level 2: Security experts deliver security as code

Level 1: Development and operations get fully automated security support

How do we turn “security into code”?



Defend

Do we have a defense strategy and implementation?

Code

Assess

Do we automatically verify defense is present, correct, and used properly everywhere?

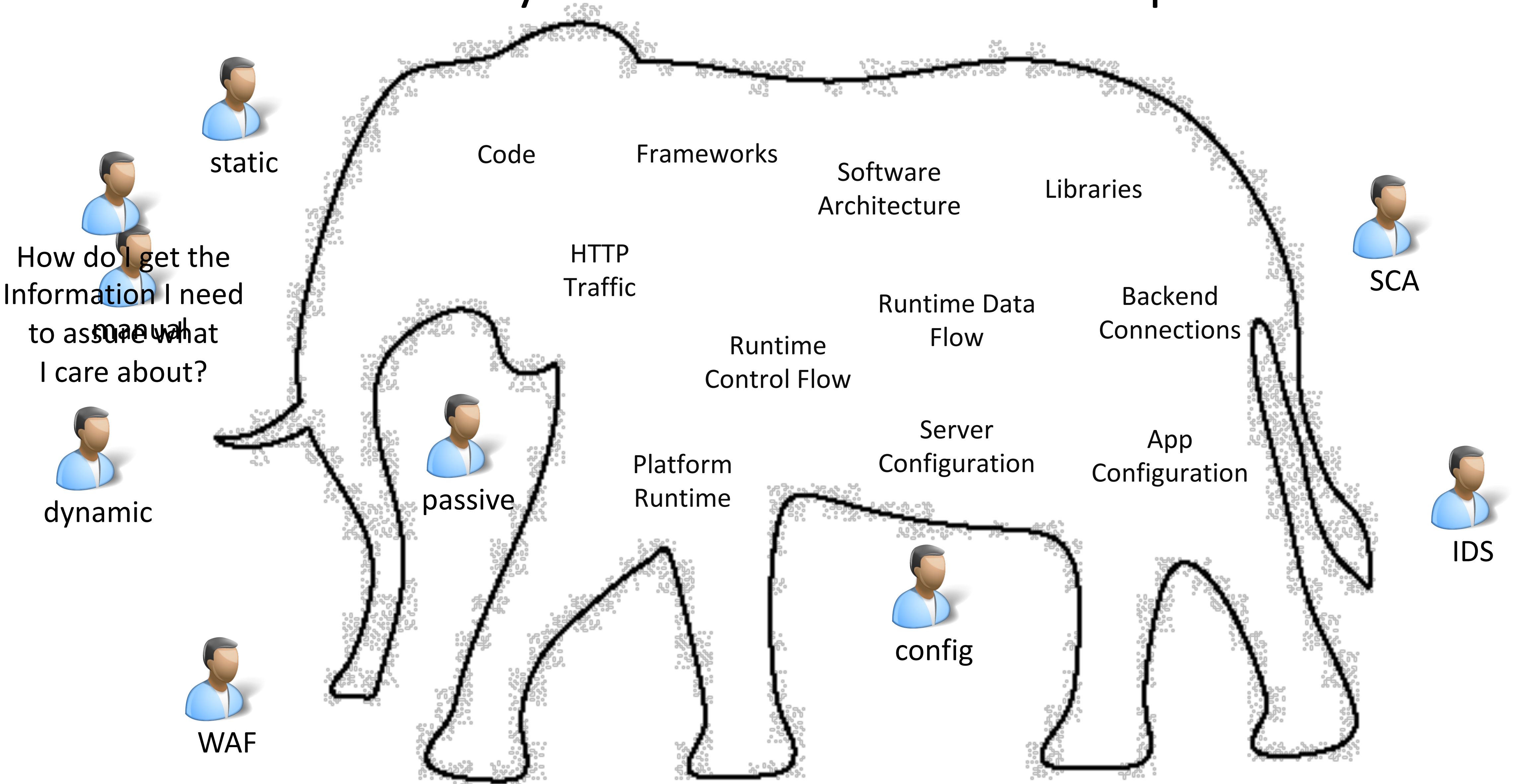
Code

Protect

Do we automatically detect and block anyone attempting to attack this?

Code

A better way to think about the problem…



static

How do I get the Information I need to assure what I care about?

manual

dynamic

WAF

Code

Frameworks

Software Architecture

Libraries

HTTP Traffic

passive

Runtime Control Flow

Runtime Data Flow

Backend Connections

Platform Runtime

Server Configuration

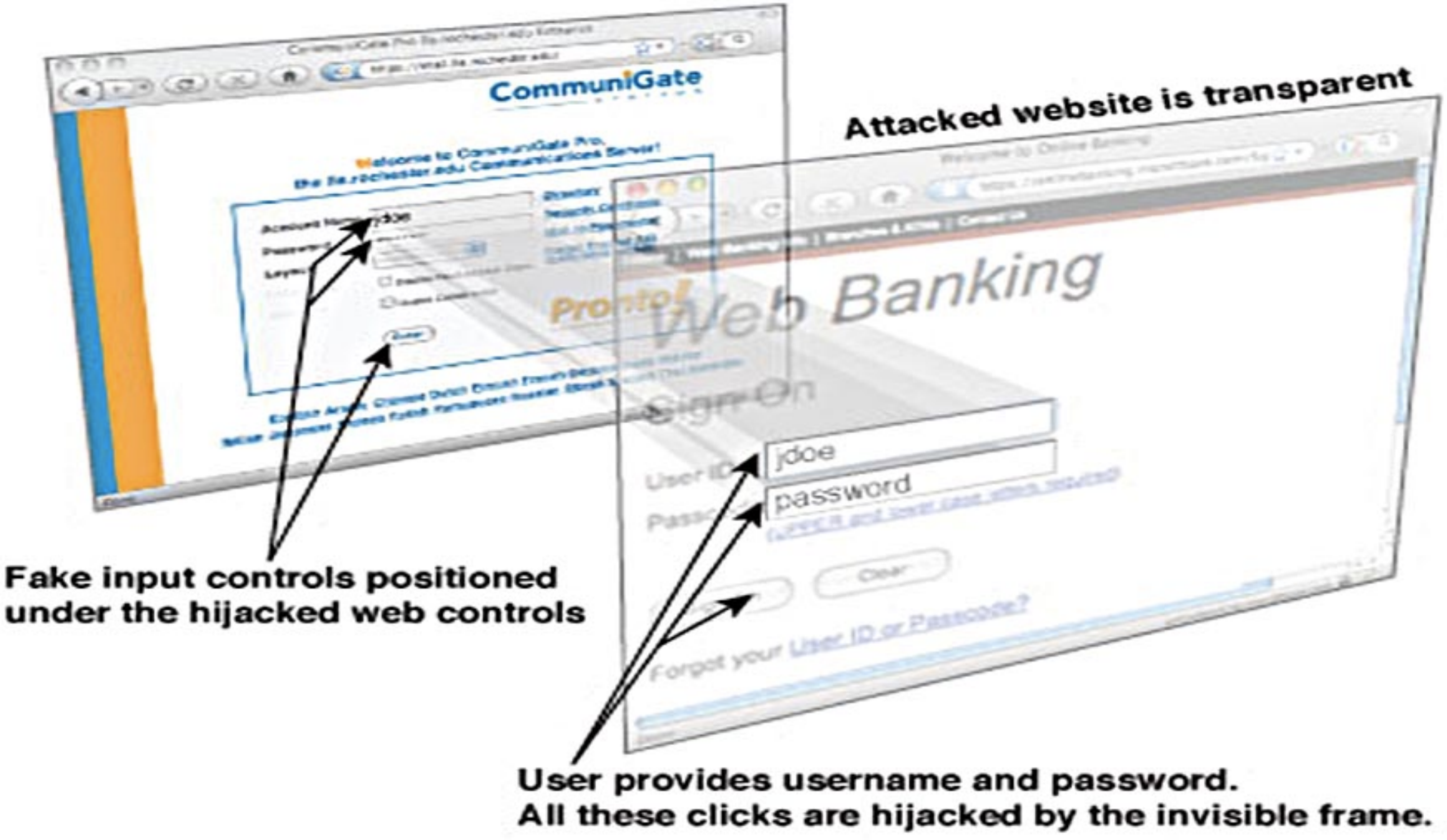
App Configuration

config

SCA

IDS

Problem: Clickjacking


Attacked website is transparent
CommuniGate
Web Banking
Sign On
jdoe
password
Clear
Forgot your User ID or Passcode?
Fake input controls positioned under the hijacked web controls
User provides username and password. All these clicks are hijacked by the invisible frame.

* Image: Igor Abade

Defend

Use X-FRAME-OPTIONS header to prevent frames

Assess

Check HTTP responses to ensure they all have X-FRAME-OPTIONS set.

Protect

Tough – looks like expected traffic.

Problem: Bypassing Verb-Based Auth’n and Auth’z (VBAAC)

```
<security-constraint>
<web-resource-collection>
    <url-pattern>/admin/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
</web-resource-collection>
<auth-constraint>
     <role-name>admin</role-name>
</auth-constraint>
</security-constraint>
```

Defend

Ensure no unauthorized HTTP verbs can be used. GET and POST only.

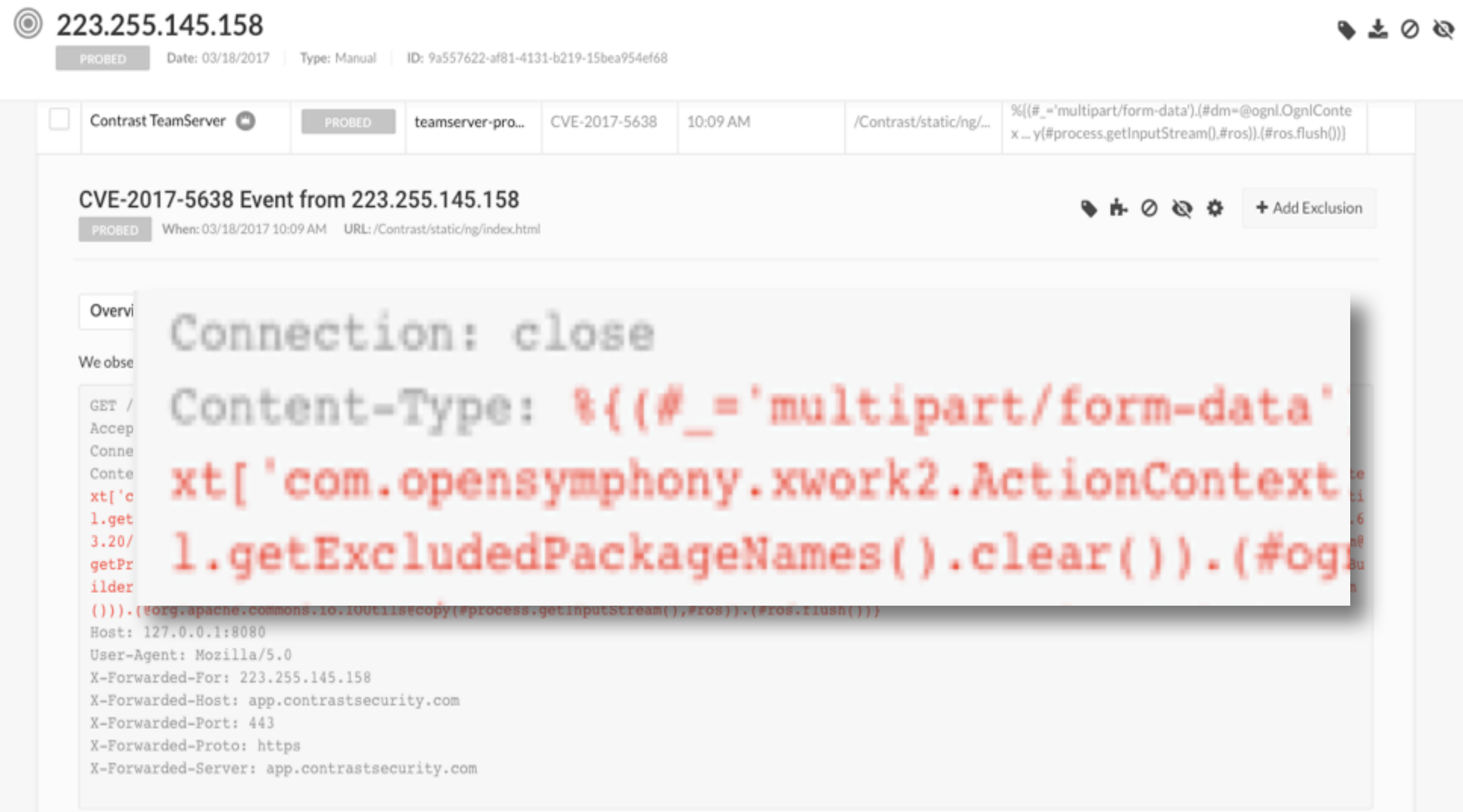
Assess

Use a tool to automatically analyze the logic of authentication and access control configurations.

Protect

Check HTTP to detect and block use of unauthorized verbs.

Problem: Insecure Libraries



223.255.145.158

PROBED Date: 03/18/2017 | Type: Manual | ID: 9a557622-af81-4131-b219-15bea954ef68

□ Contrast TeamServer	PROBED	teamserver-pro...	CVE-2017-5638	10:09 AM	/Contrast/static/ng/...	%{(#_='multipart/form-data').(#dm=@ognl.OgnlConte x ... y(#process.getInputStream(),#ros)).(#ros.flush())}

CVE-2017-5638 Event from 223.255.145.158

PROBED When: 03/18/2017 10:09 AM URL: /Contrast/static/ng/index.html

+ Add Exclusion

Overvi

We obse

```
GET /
Accep
Conne
Conte
xt['c
l.get
3.20/
getPr
ilder
())).(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}
Host: 127.0.0.1:8080
User-Agent: Mozilla/5.0
X-Forwarded-For: 223.255.145.158
X-Forwarded-Host: app.contrastsecurity.com
X-Forwarded-Port: 443
X-Forwarded-Proto: https
X-Forwarded-Server: app.contrastsecurity.com
```

```
Connection: close
Content-Type: %{(#_='multipart/form-data'
xt['com.opensymphony.xwork2.ActionContext
l.getExcludedPackageNames().clear()).(#og
```

Defend

Patch and upgrade quickly

Assess

Continuously assess libraries that are actually used for known vulnerabilities.

Protect

Deploy virtual patches that prevent vulnerability from being exploited.

Problem: Weak Crypto Algorithm

‘MD5’ is <u>everywhere</u>

Defend

Choose a strong algorithm

Assess

Watch cipher construction at runtime to ensure no weak algorithms selected.

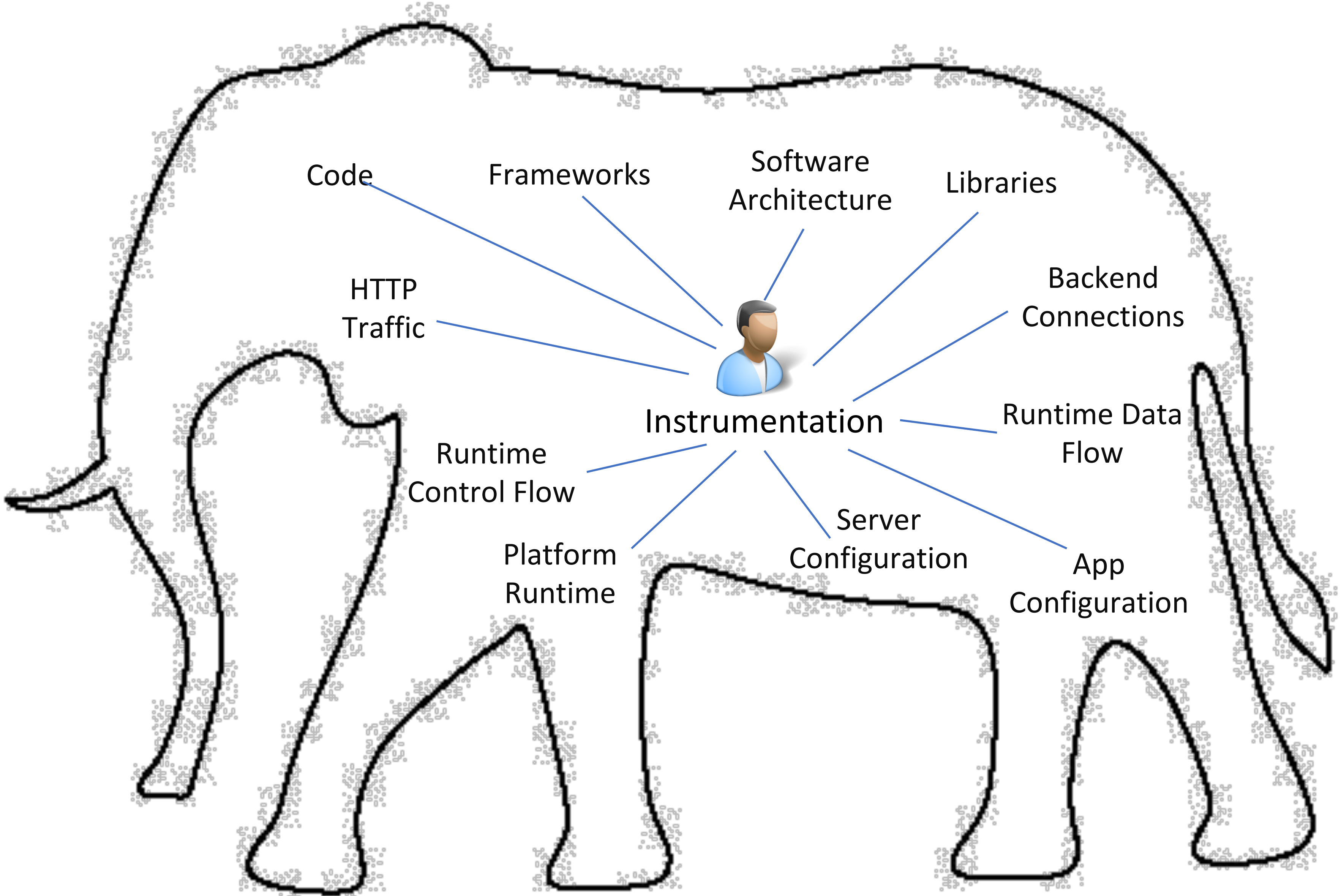
Protect

Watch running code for exceptions indicating padding oracle attacks, for example.

Summary so far…

- Clickjacking -> need HTTP headers
- Bypassable VBAAC -> need web configuration, HTTP to block
- Insecure Libraries -> need libraries, frameworks, servers, platform
- Weak Encryption -> need code, configuration, exceptions
- …

Great – so I have to run 50 tools? No.



Code

Frameworks

Software Architecture

Libraries

HTTP Traffic

Backend Connections

Instrumentation

Runtime Data Flow

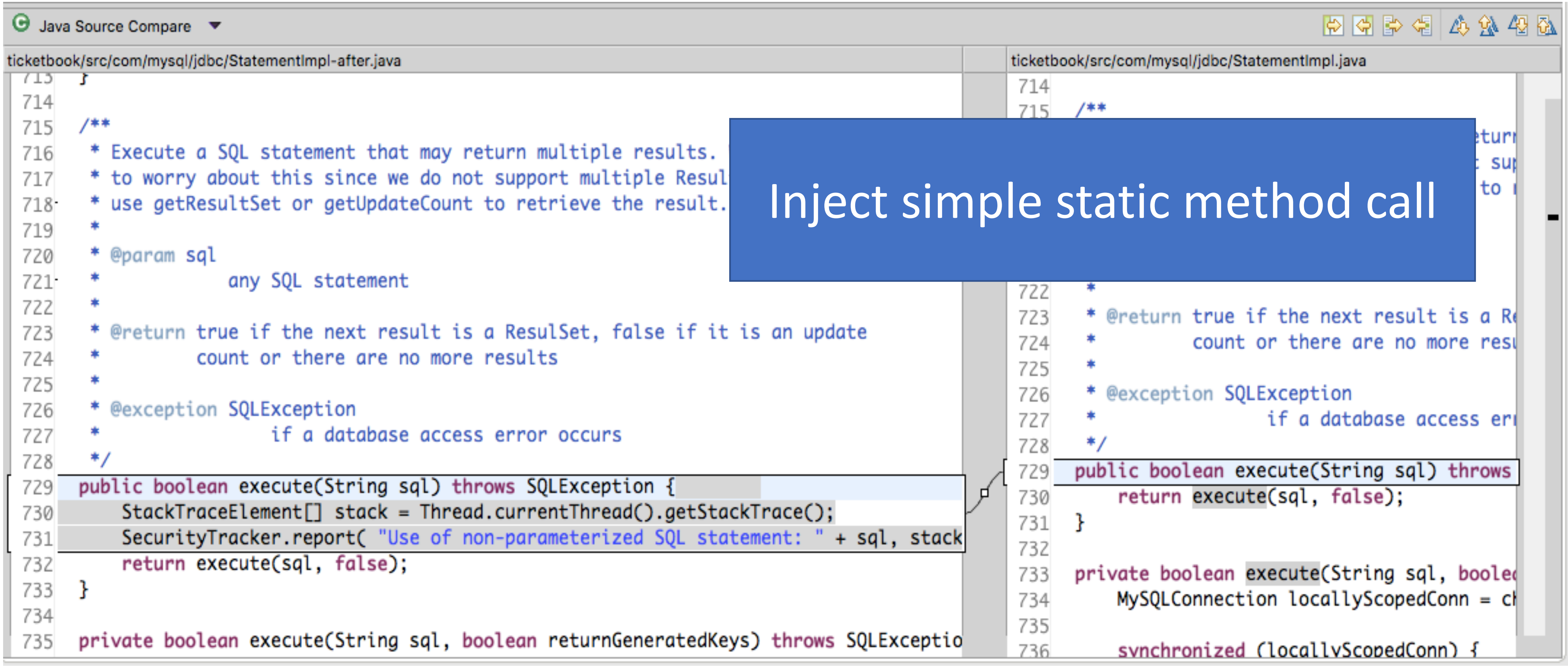
Runtime Control Flow

Platform Runtime

Server Configuration

App Configuration

Source Instrumentation



ticketbook/src/com/mysql/jdbc/StatementImpl-after.java

```
}

/**
 * Execute a SQL statement that may return multiple results. 
 * to worry about this since we do not support multiple Resul
 * use getResultSet or getUpdateCount to retrieve the result.
 *
 * @param sql
 *            any SQL statement
 *
 * @return true if the next result is a ResulSet, false if it is an update
 *         count or there are no more results
 *
 * @exception SQLException
 *                if a database access error occurs
 */
public boolean execute(String sql) throws SQLException {
    StackTraceElement[] stack = Thread.currentThread().getStackTrace();
    SecurityTracker.report( "Use of non-parameterized SQL statement: " + sql, stack
    return execute(sql, false);
}

private boolean execute(String sql, boolean returnGeneratedKeys) throws SQLExceptio
```

ticketbook/src/com/mysql/jdbc/StatementImpl.java

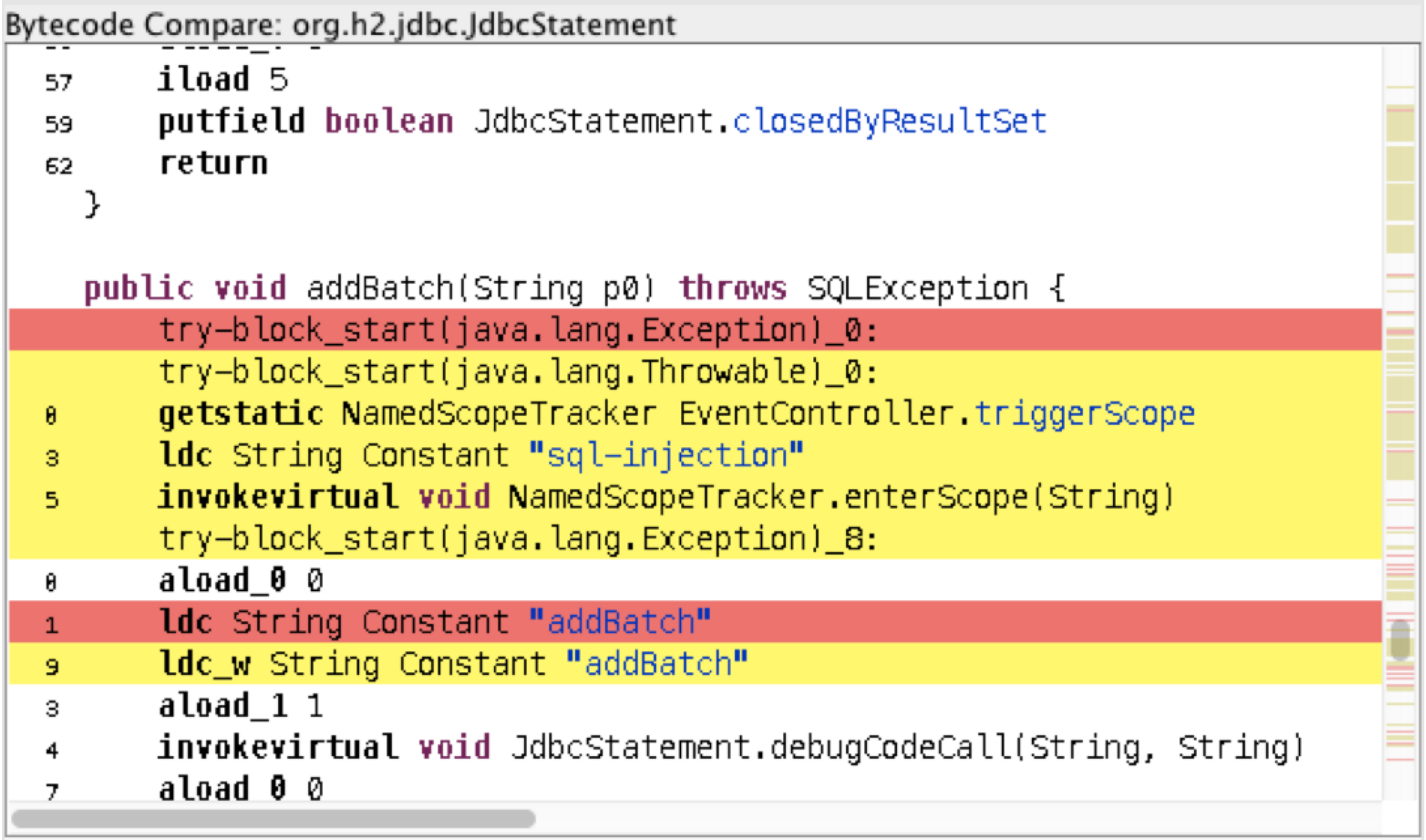
```

/**
 *
 * @return true if the next result is a Re
 *         count or there are no more resu
 *
 * @exception SQLException
 *                if a database access er
 */
public boolean execute(String sql) throws
    return execute(sql, false);
}

private boolean execute(String sql, boole
    MySQLConnection locallyScopedConn = ch

    synchronized (locallyScopedConn) {
```

Inject simple static method call

Binary Instrumentation

- Widely used
  - CPU Performance
  - Memory
  - Logging
  - Security
  - …

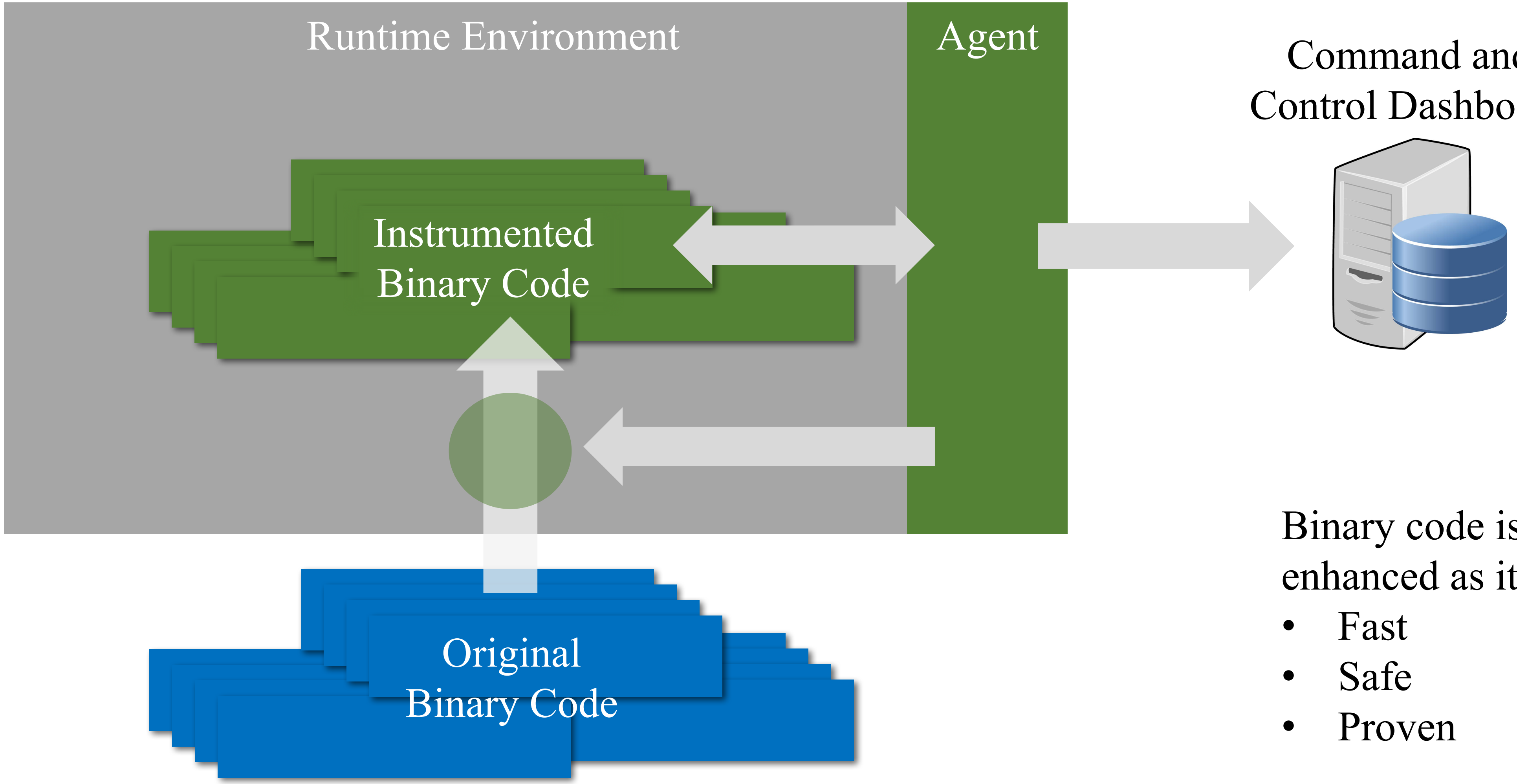
- Lots of libraries
  - ASM (Java)
  - BCEL (Java)
  - Javassist (Java)
  - MBEL (.NET)
  - RAIL (.NET)
  - …



Bytecode Compare: org.h2.jdbc.JdbcStatement

```
57      iload 5
59      putfield boolean JdbcStatement.closedByResultSet
62      return
   }

   public void addBatch(String p0) throws SQLException {
        try-block_start(java.lang.Exception)_0:
        try-block_start(java.lang.Throwable)_0:
0       getstatic NamedScopeTracker EventController.triggerScope
3       ldc String Constant "sql-injection"
5       invokevirtual void NamedScopeTracker.enterScope(String)
        try-block_start(java.lang.Exception)_8:
0       aload_0 0
1       ldc String Constant "addBatch"
9       ldc_w String Constant "addBatch"
3       aload_1 1
4       invokevirtual void JdbcStatement.debugCodeCall(String, String)
7       aload_0 0
```

Dynamic Binary Instrumentation!

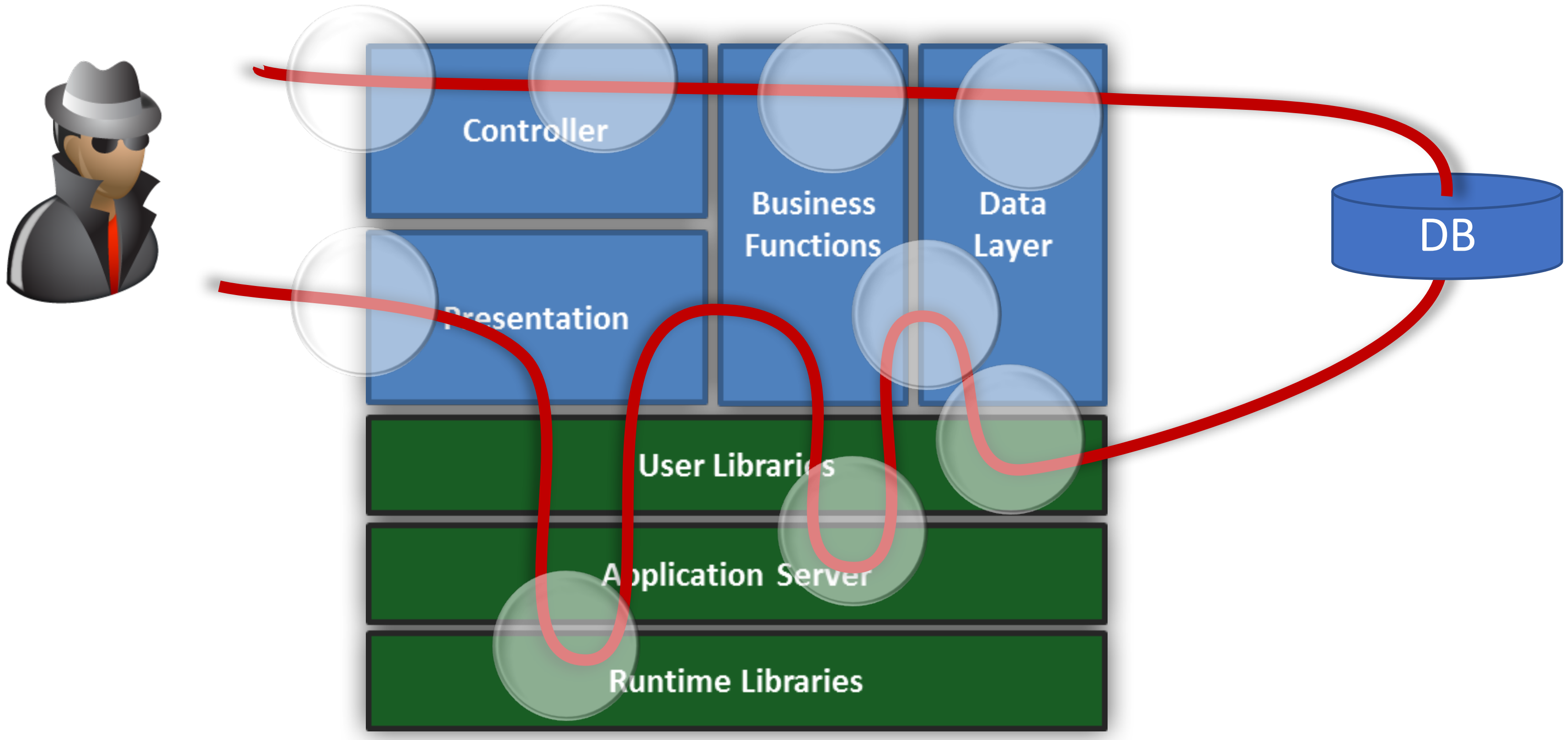


Binary code is enhanced as it loads:

- Fast
- Safe
- Proven

Problem: Injection (SQL, XSS, etc…)

- Attacker sends data that is passed to an interpreter (SQL, LDAP, EL, …)



Defend

Use escaping, parameterization correctly everywhere. Right.

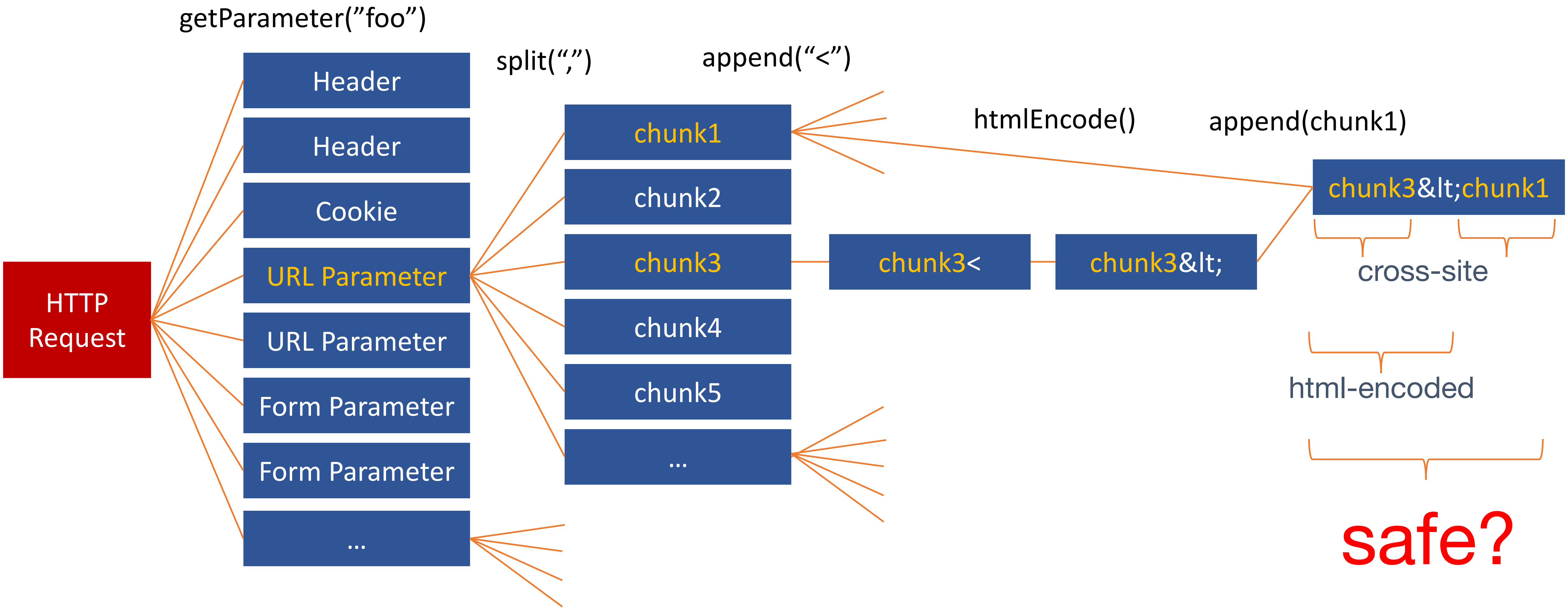
Assess

Use automated data flow analysis to track untrusted data to any queries.

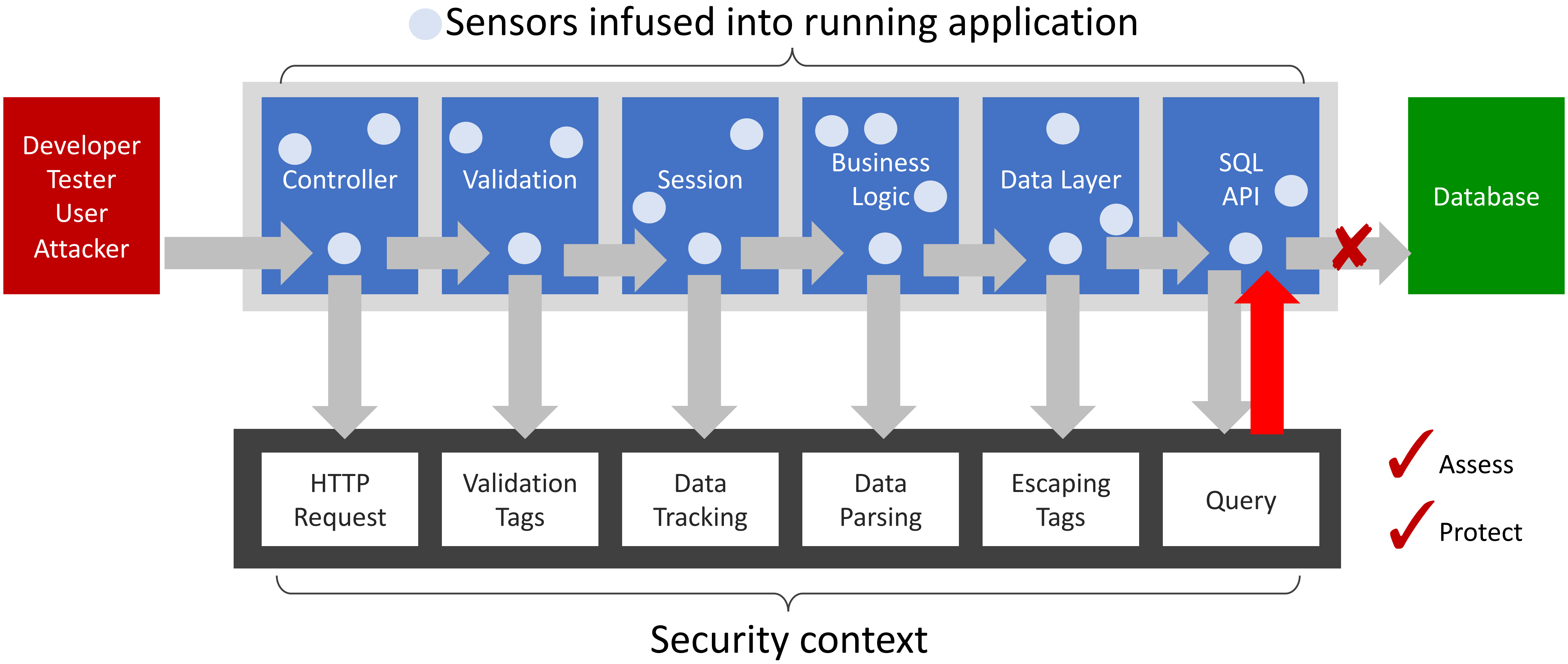
Protect

Analyze whether untrusted data flows to a query and modifies its meaning.

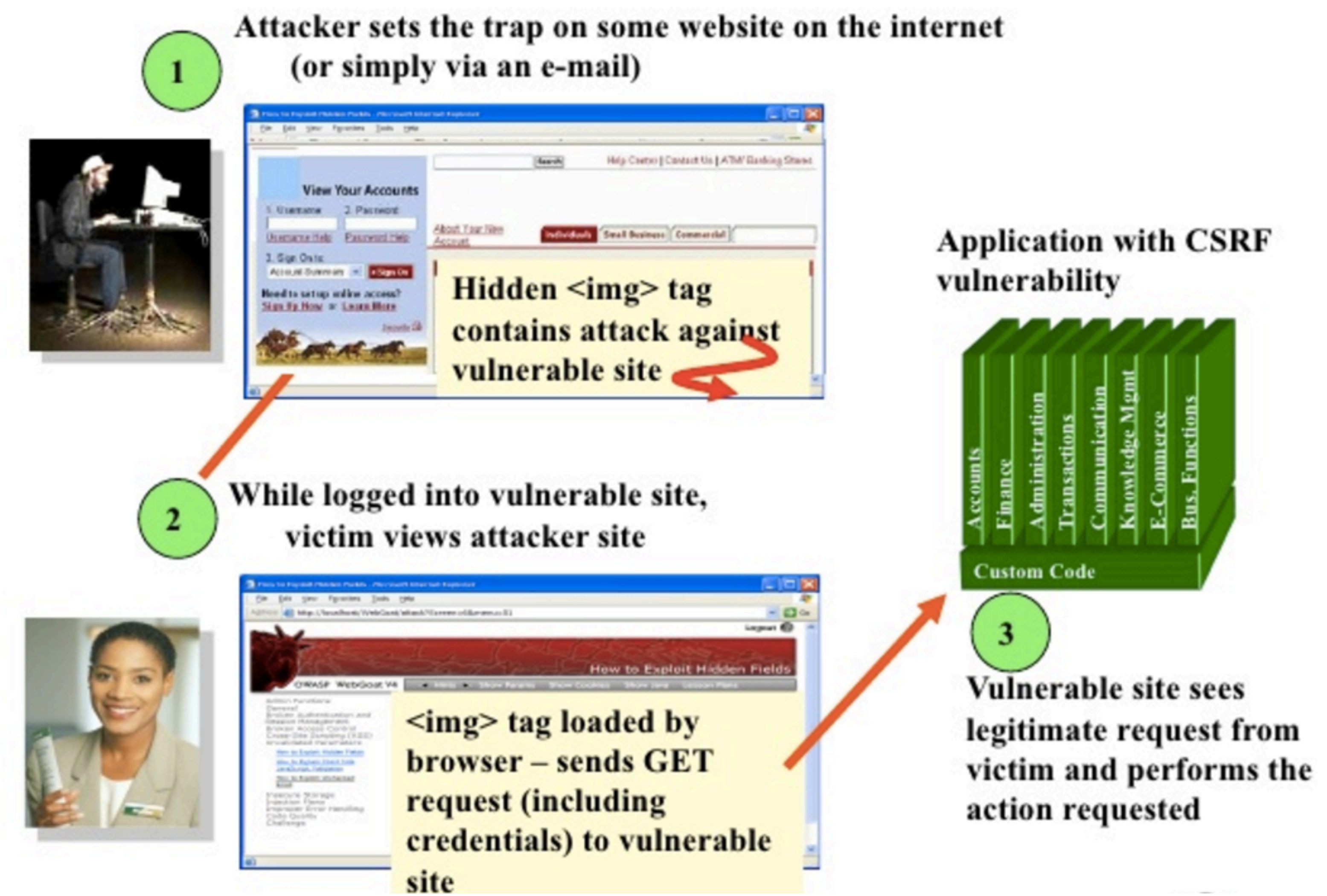
Data flow analysis (aka clusterbomb)



Solution: Instrumentation



Cross-Site Request Forgery



1 Attacker sets the trap on some website on the internet (or simply via an e-mail)

Hidden <img> tag contains attack against vulnerable site

2 While logged into vulnerable site, victim views attacker site

<img> tag loaded by browser – sends GET request (including credentials) to vulnerable site

Application with CSRF vulnerability

3 Vulnerable site sees legitimate request from victim and performs the action requested

Defend

Add a token to links and forms. Verify token is present on transactions.

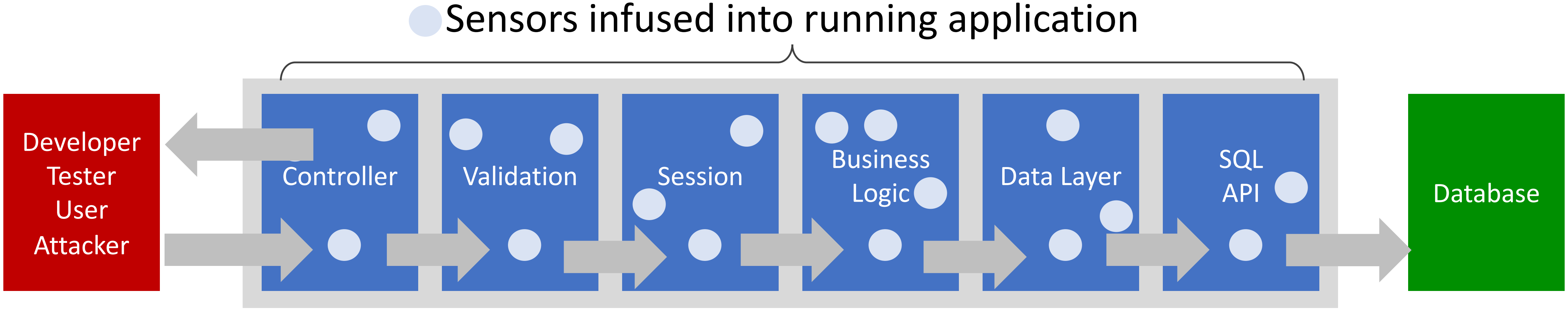
Assess

Verify non-XHR requests have token on non-idempotent transactions.

Protect

Application should detect and block use of unauthorized verbs.

Solution: Instrumentation



✓ Vulnerability

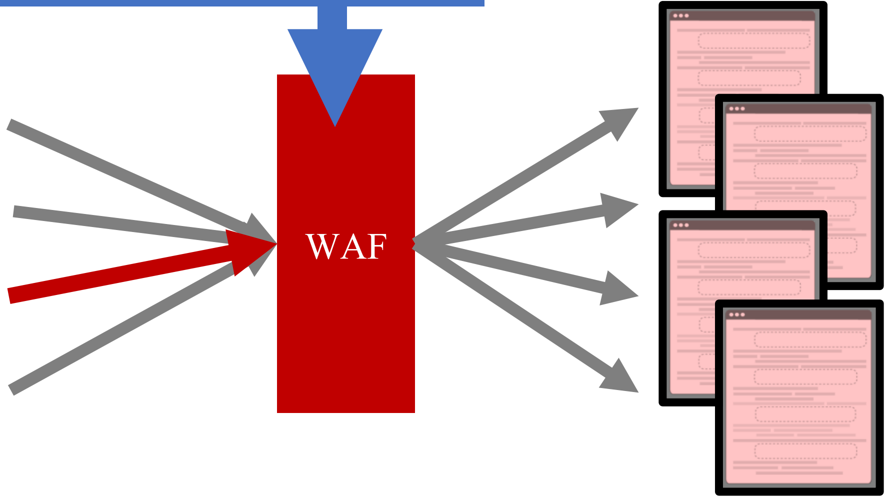
- Is not an XHR request?
- Token check fails
- Non-idempotent transaction

✓ Attack

- Add CSRF token to webpages
- Check for tokens on susceptible pages

WAF

PERIMETER DECISION POINT



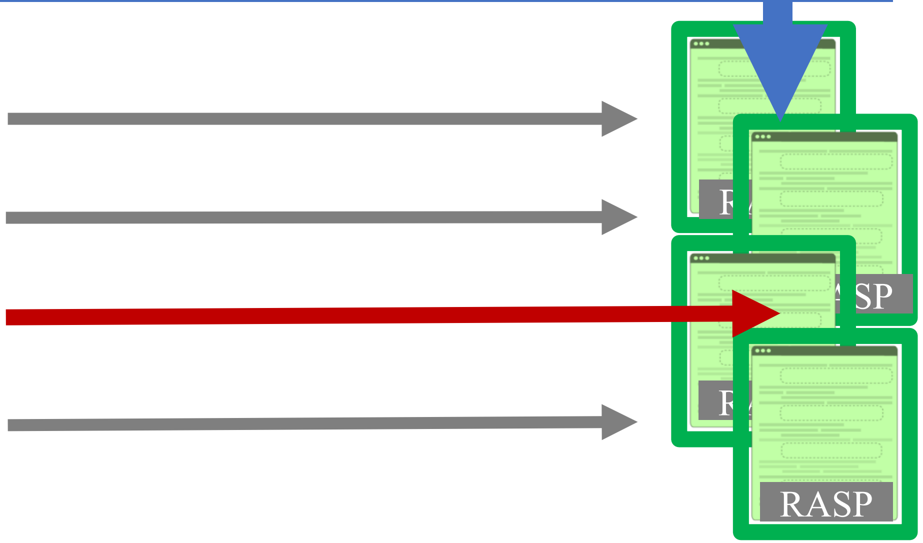
GET /foo?name='%20or%20%20'1'='1 HTTP/1.0

Three problems:

1) Bottleneck
2) No context
3) Impedance

RASP

APPLICATION DECISION POINT



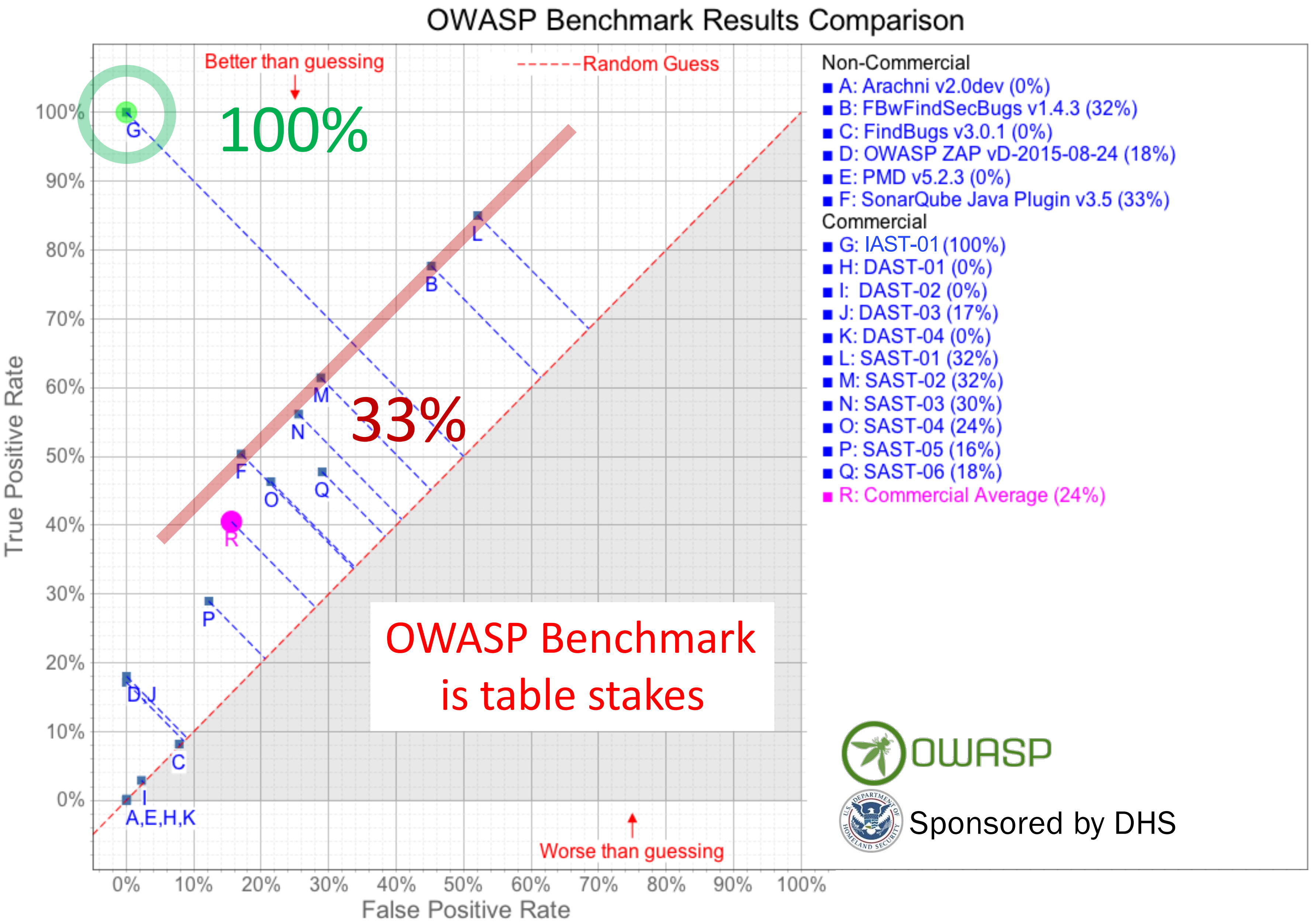
GET /foo?name='%20or%20%20'1'='1 HTTP/1.0

```
stmt.execute( "select * from table where id ='1' or '1'='1'" );
```

OWASP Benchmark – thousands of test cases across a range of true and false vulnerabilities

Free, open, reproduceable

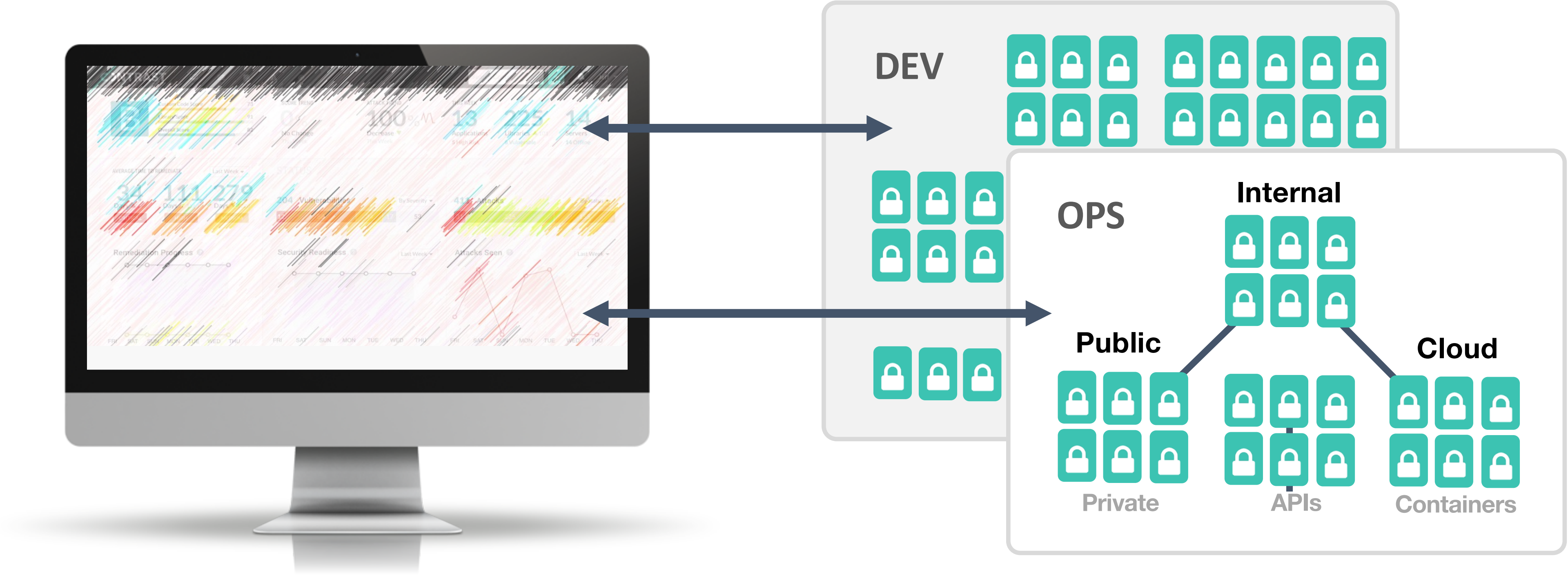
Instrumentation speed and accuracy dominates SAST and DAST



OWASP Benchmark is table stakes

Sponsored by DHS

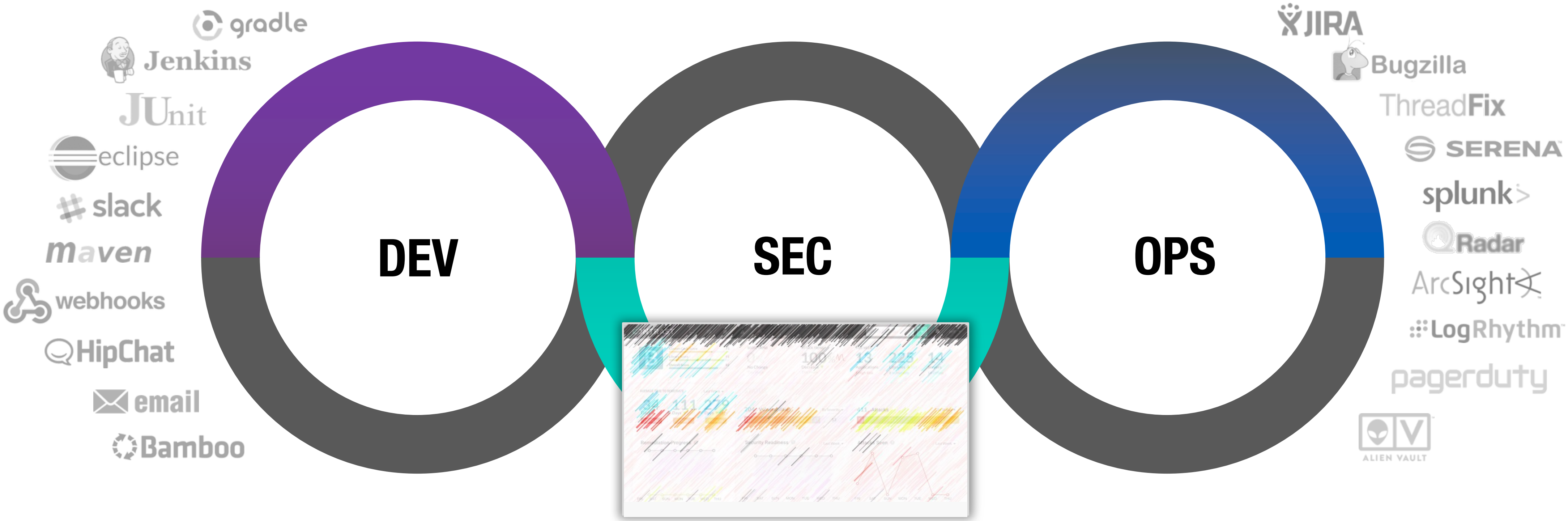
Distributed AppSec – In Parallel



Centralized Visibility and Control

Continuous Assessment and Protection

Making DevSecOps Actually Work

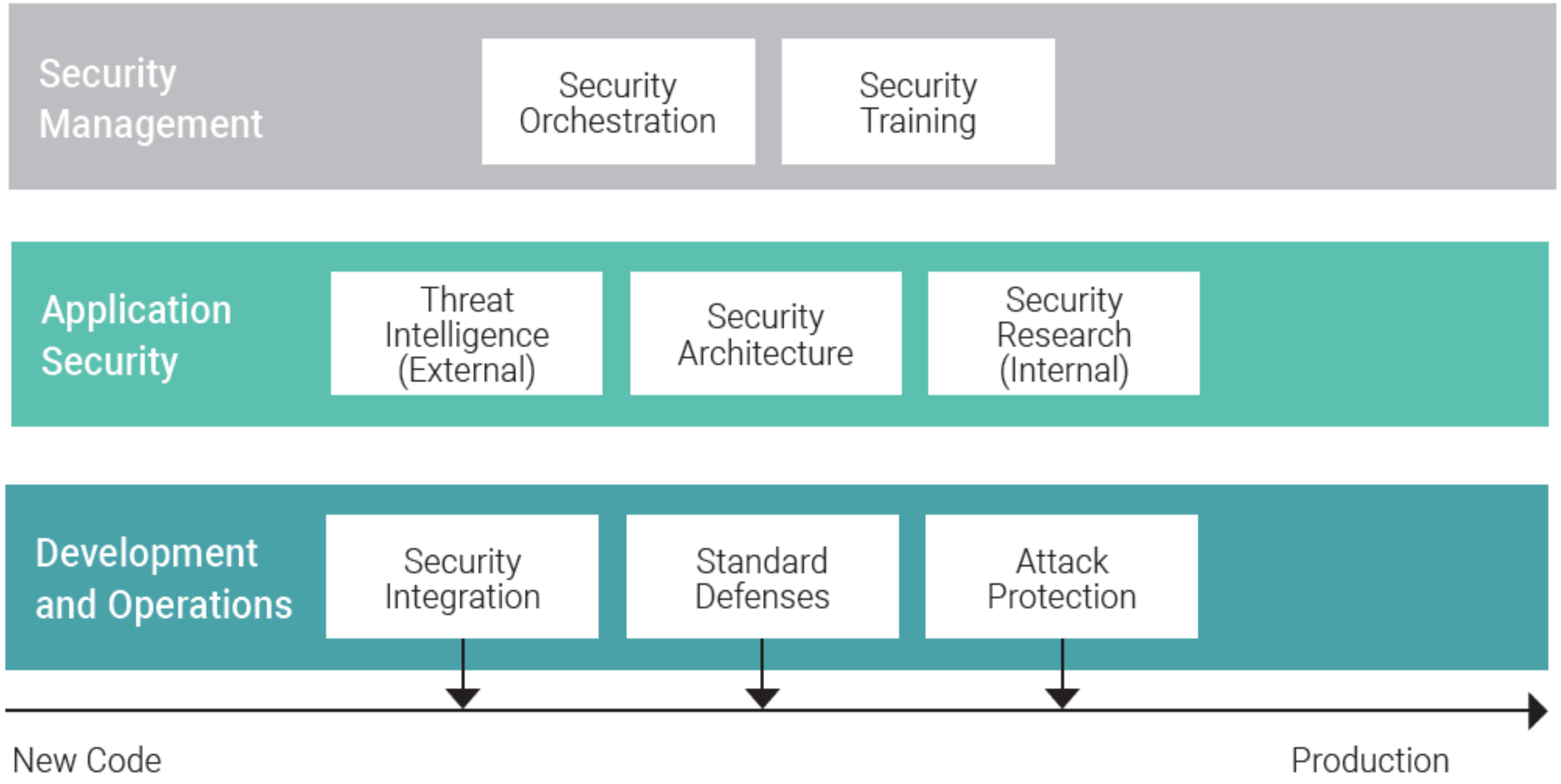


Vulnerability Alerts & Guidance

Centralized Visibility & Control

Attack Protection & Threat Intelligence

Instrumentation Powers Continuous AppSec



Continuous Application Security

Analytics across entire portfolio drive budget and priorities

A platform for modeling policy to enforce across portfolio

Instant accurate notification of problems via existing tools

CONTRAST SECURITY

Gartner®
VISIONARY

FORRESTER®
LEADER

RSAC Innovation Sandbox 2017 FINALIST

CYBER SECURITY EXCELLENCE AWARDS ★ WINNER ★ 2017

2016 SINET 16 INNOVATOR

THE CHANNEL CO. CRN SECURITY 100

Info Security Products Guide 2017 GLOBAL EXCELLENCE AWARDS ★★★★★

Winner SC 2017 awards
Honored in the U.S.

WILLIAMS-SONOMA HUAWEI CHROMERIVER AUTODESK. BT RingCentral® demandware A Salesforce Company

CONTRAST SECURITY

THANK YOU

Jeff Williams | jeff.williams@contrastsecurity.com