CONTRAST
SECURITY

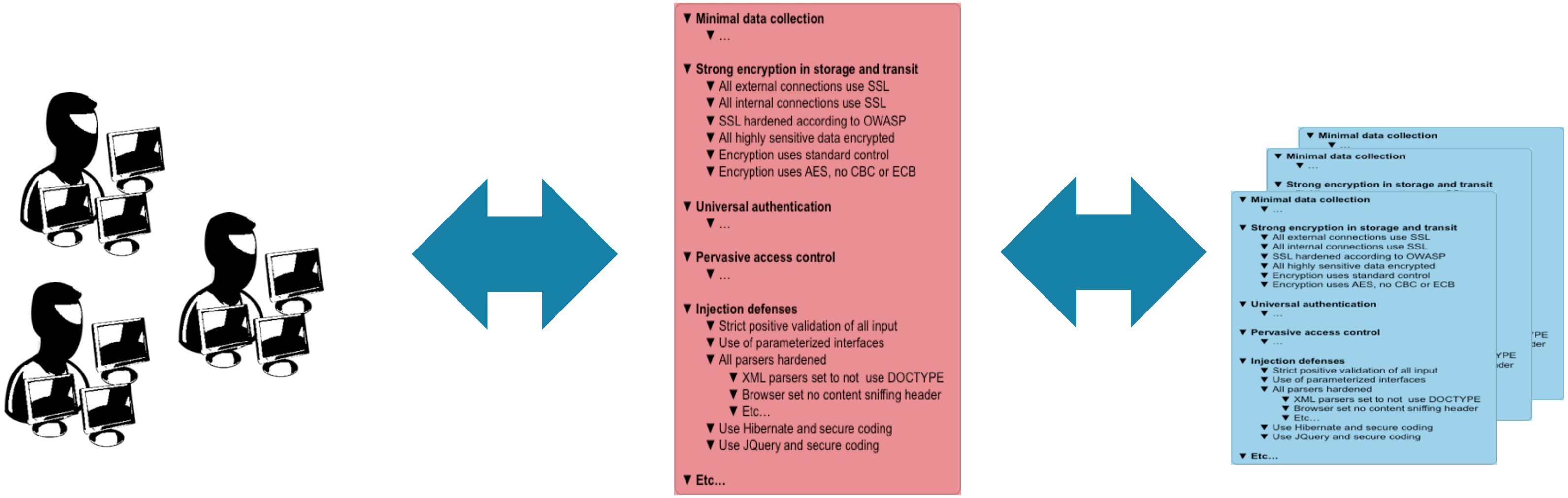
THE FUTURE OF APPSEC AUTOMATION

WHY YOUR APPSEC EXPERTS ARE KILLING YOU

Jeff Williams, CTO
@planetlevel

CONTRAST SECURITY
291 Lambert Avenue
Palo Alto, California 94306
www.contrastsecurity.com

ARE YOU “SECURE”?



- Minimal data collection
  - ...
- Strong encryption in storage and transit
  - All external connections use SSL
  - All internal connections use SSL
  - SSL hardened according to OWASP
  - All highly sensitive data encrypted
  - Encryption uses standard control
  - Encryption uses AES, no CBC or ECB
- Universal authentication
  - ...
- Pervasive access control
  - ...
- Injection defenses
  - Strict positive validation of all input
  - Use of parameterized interfaces
  - All parsers hardened
    - XML parsers set to not use DOCTYPE
    - Browser set no content sniffing header
    - Etc…
  - Use Hibernate and secure coding
  - Use JQuery and secure coding
- Etc...

- Minimal data collection
  - ...
- Strong encryption in storage and transit
  - All external connections use SSL
  - All internal connections use SSL
  - SSL hardened according to OWASP
  - All highly sensitive data encrypted
  - Encryption uses standard control
  - Encryption uses AES, no CBC or ECB
- Universal authentication
  - ...
- Pervasive access control
  - ...
- Injection defenses
  - Strict positive validation of all input
  - Use of parameterized interfaces
  - All parsers hardened
    - XML parsers set to not use DOCTYPE
    - Browser set no content sniffing header
    - Etc…
  - Use Hibernate and secure coding
  - Use JQuery and secure coding
- Etc...

Threats

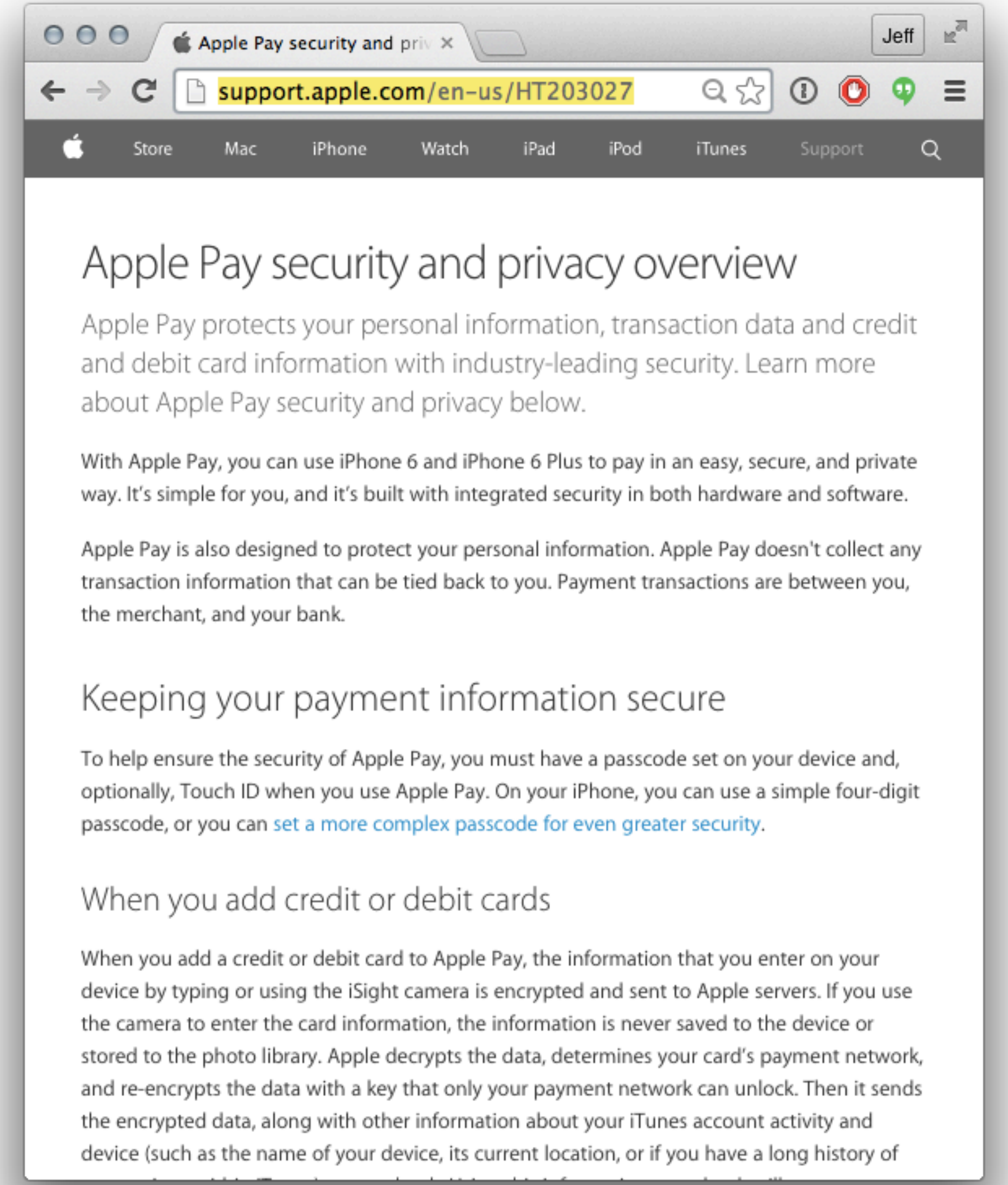
Expected Security Model (claims)

Verified Defenses (evidence)

Claims.

Not evidence.

But still very cool.



Apple Pay security and privacy overview

Apple Pay protects your personal information, transaction data and credit and debit card information with industry-leading security. Learn more about Apple Pay security and privacy below.

With Apple Pay, you can use iPhone 6 and iPhone 6 Plus to pay in an easy, secure, and private way. It's simple for you, and it's built with integrated security in both hardware and software.

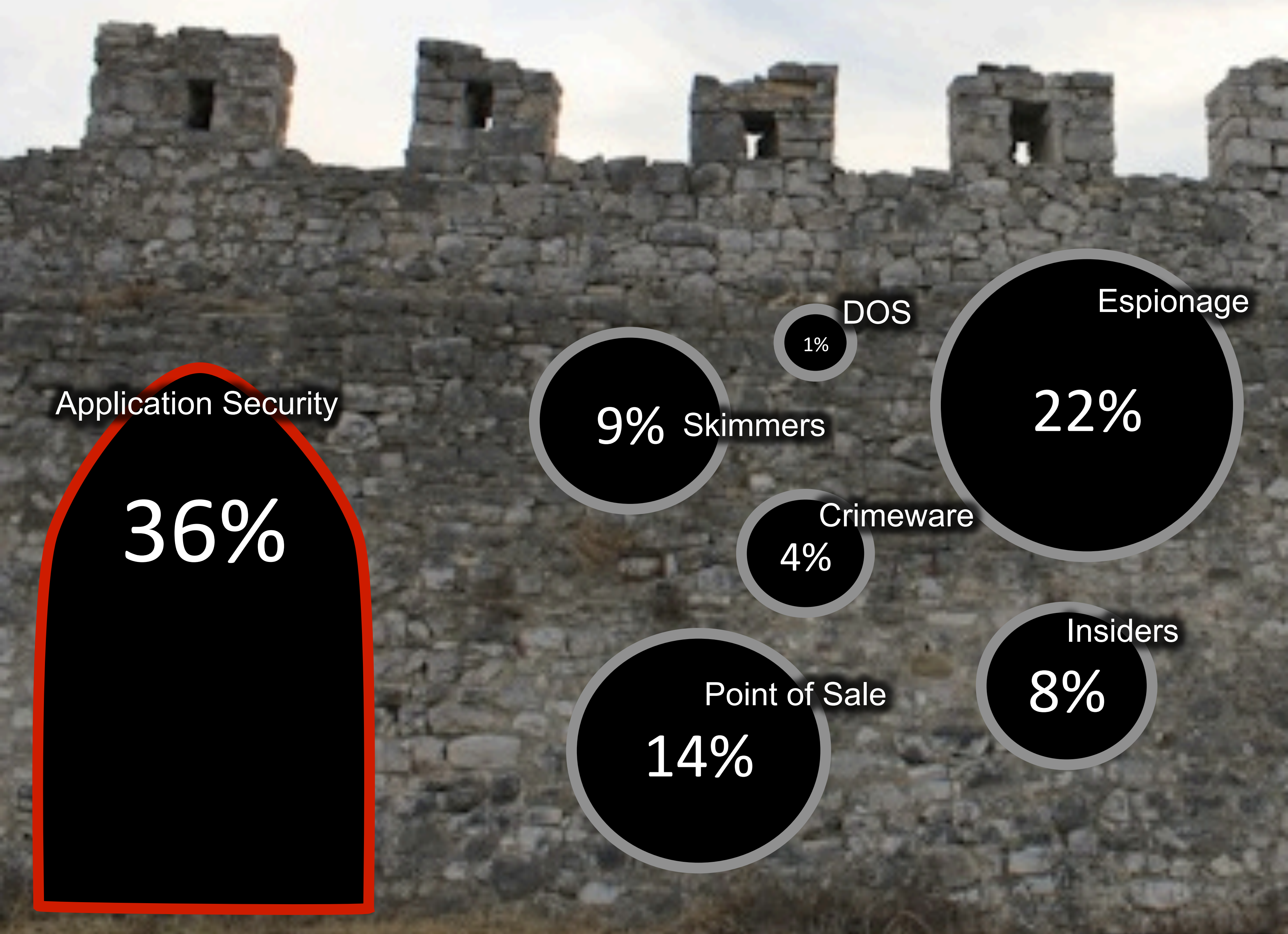
Apple Pay is also designed to protect your personal information. Apple Pay doesn't collect any transaction information that can be tied back to you. Payment transactions are between you, the merchant, and your bank.

Keeping your payment information secure

To help ensure the security of Apple Pay, you must have a passcode set on your device and, optionally, Touch ID when you use Apple Pay. On your iPhone, you can use a simple four-digit passcode, or you can set a more complex passcode for even greater security.

When you add credit or debit cards

When you add a credit or debit card to Apple Pay, the information that you enter on your device by typing or using the iSight camera is encrypted and sent to Apple servers. If you use the camera to enter the card information, the information is never saved to the device or stored to the photo library. Apple decrypts the data, determines your card's payment network, and re-encrypts the data with a key that only your payment network can unlock. Then it sends the encrypted data, along with other information about your iTunes account activity and device (such as the name of your device, its current location, or if you have a long history of



Application Security 36%

DOS 1%

Espionage 22%

Skimmers 9%

Crimeware 4%

Insiders 8%

Point of Sale 14%

*2014 Verizon DBIR

Application Security

36%

1.7% of
security spending

DOS

Espionage

Skim

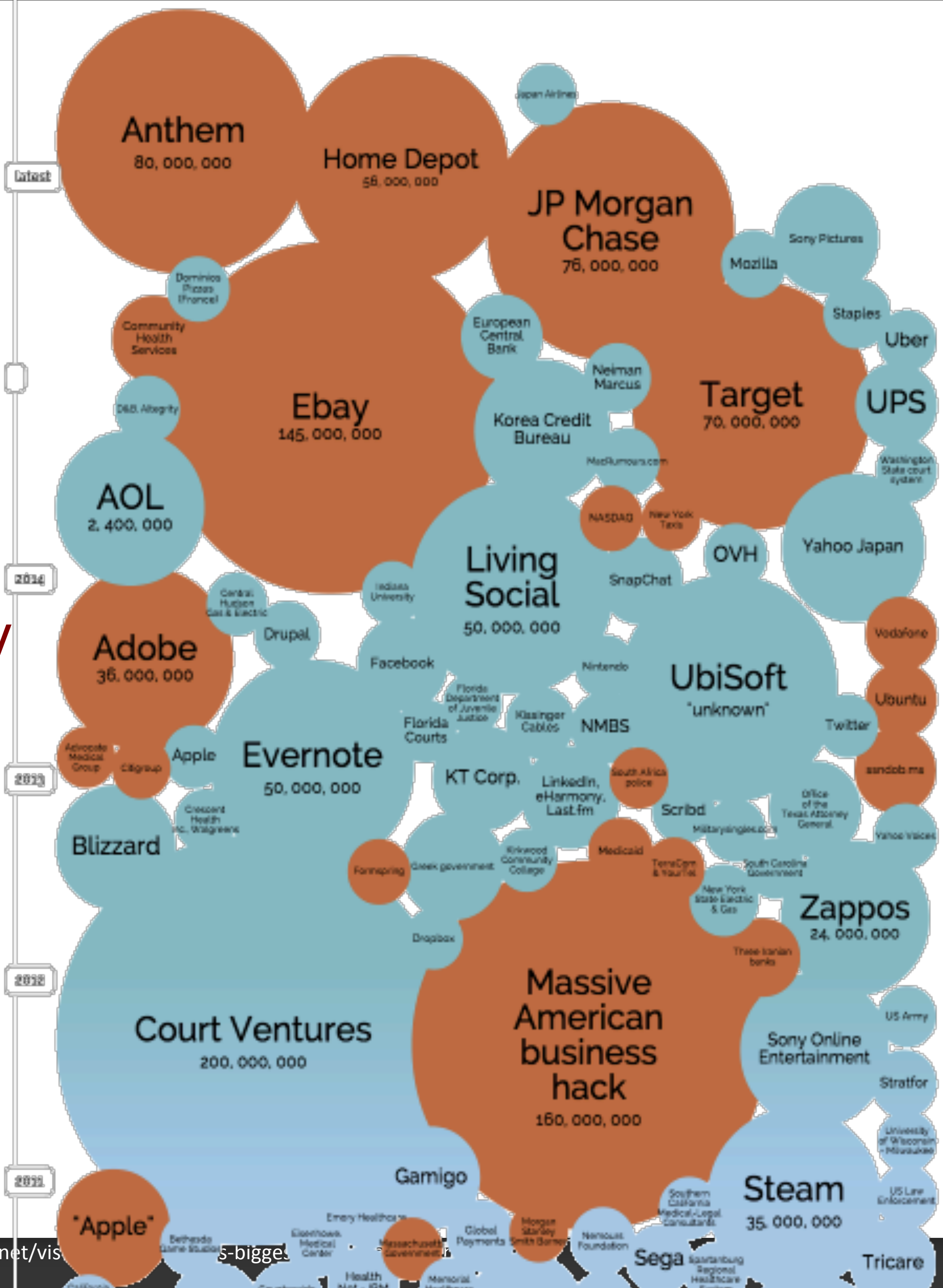
ware

Point of Sale

Insiders

*2014 Verizon DBIR

The frequency and sophistication of attacks is increasing…



…as the complexity, connectivity, and criticality of our code is increasing

* http://www.informationisbeautiful.net/vis s-bigge

“Application security is eating security” – Alex Stamos (Yahoo CISO)

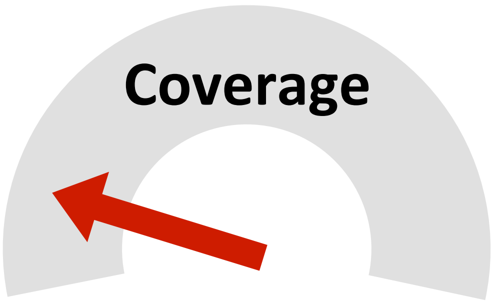
How Are We Doing?

1. Assurance?

22.4

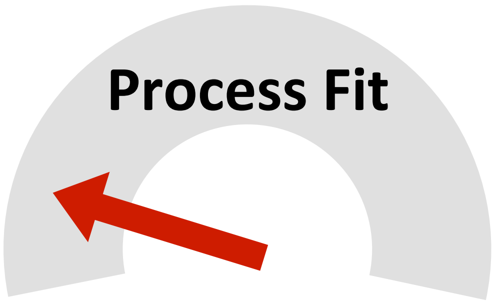
2. Coverage?

10%



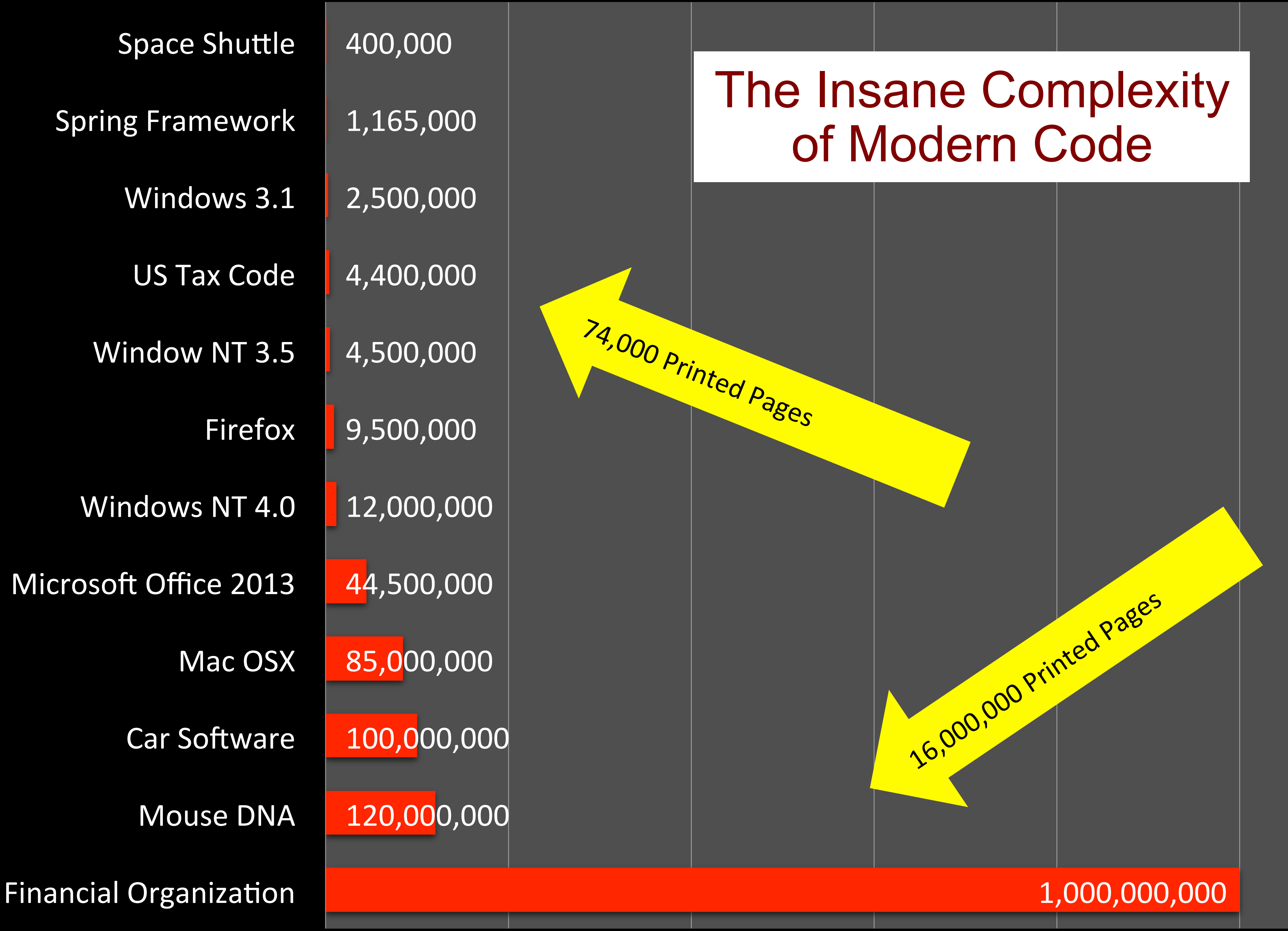
3. Process Fit?

Mad



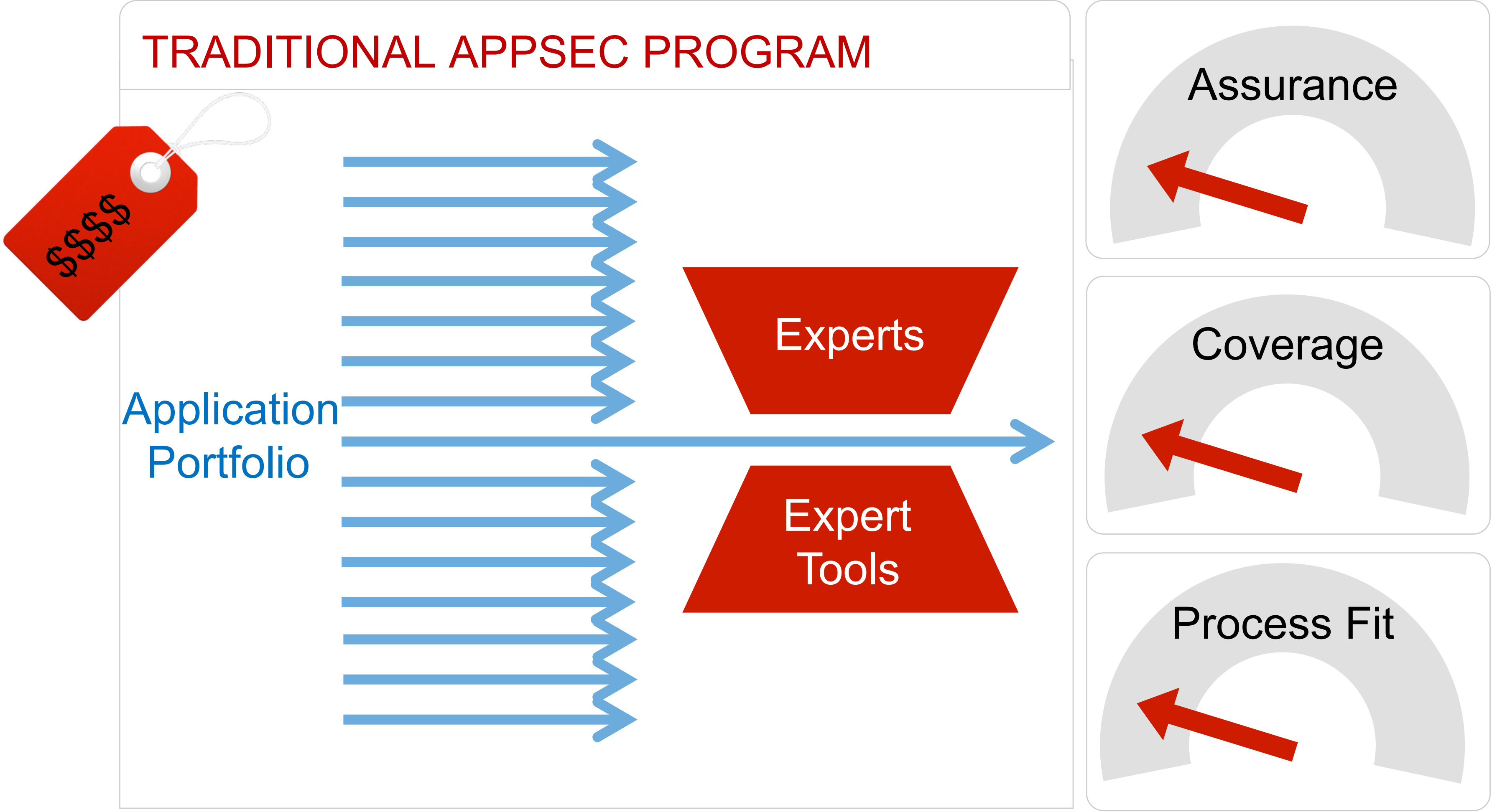
*2014 Aspect Security Global Application Security Risk Report

The Insane Complexity of Modern Code



74,000 Printed Pages

16,000,000 Printed Pages

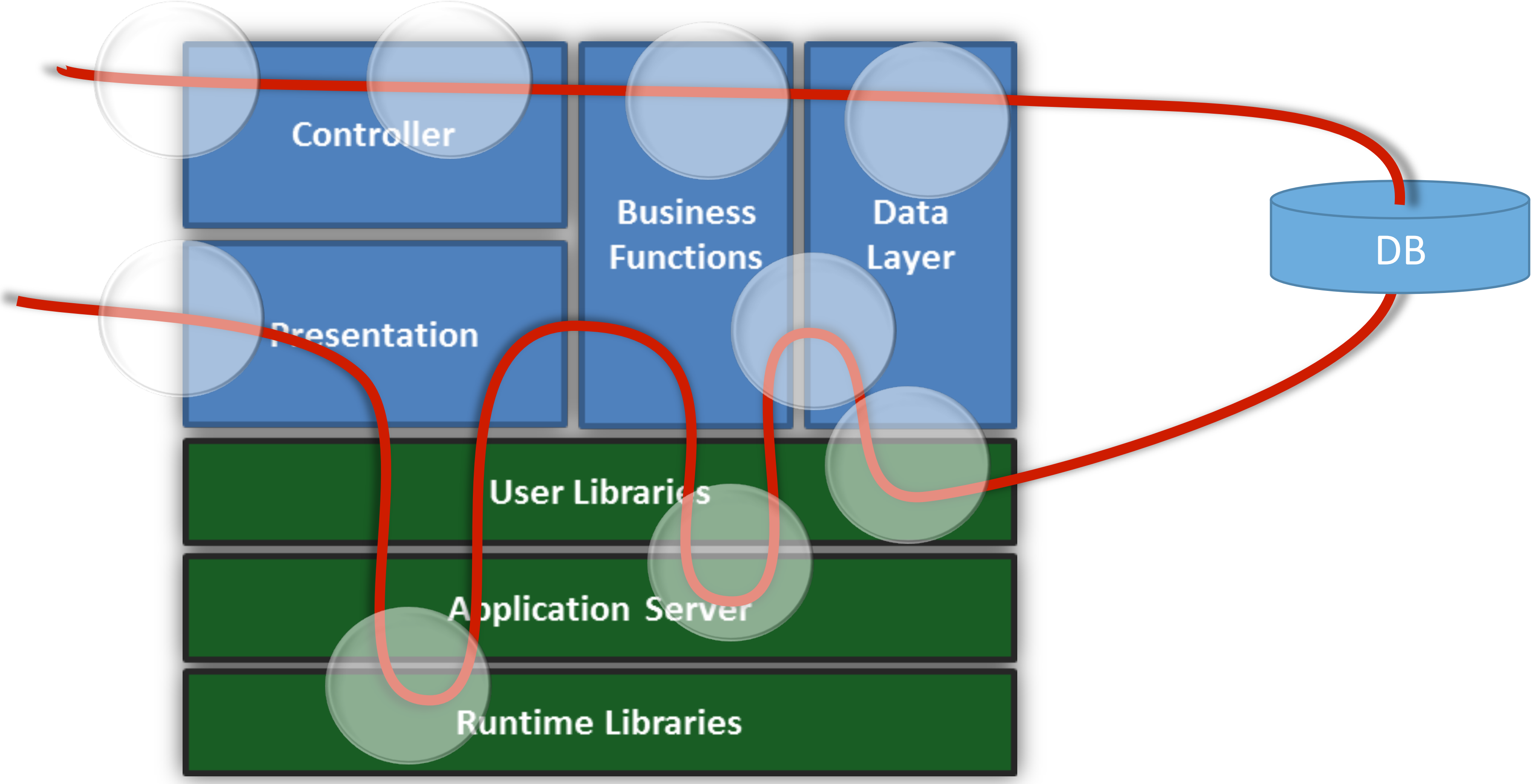


Development organizations interpret DELAYS as DAMAGE and route around them.

CONTEXT

~~CONTENT~~ IS KING!

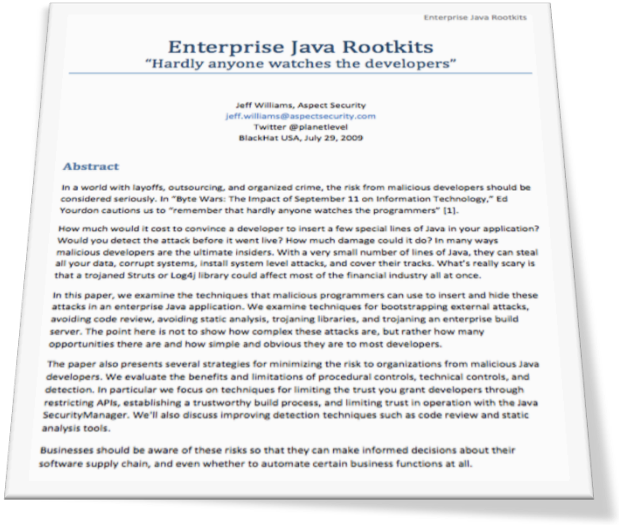
A Vulnerability Is a Pattern of Events



Rootkits Aren’t Always Evil!

- “Enterprise Java Rootkits” - BlackHat 2009

- The holy grail of backdoors
- The Java Instrumentation API



Enterprise Java Rootkits
“Hardly anyone watches the developers”

Jeff Williams, Aspect Security
jeff.williams@aspectsecurity.com
Twitter @planetlevel
BlackHat USA, July 29, 2009

Abstract

In a world with layoffs, outsourcing, and organized crime, the risk from malicious developers should be considered seriously. In “Byte Wars: The Impact of September 11 on Information Technology,” Ed Yourdon cautions us to “remember that hardly anyone watches the programmers” [1].

How much would it cost to convince a developer to insert a few special lines of Java in your application? Would you detect the attack before it went live? How much damage could it do? In many ways malicious developers are the ultimate insiders. With a very small number of lines of Java, they can steal all your data, corrupt systems, install system level attacks, and cover their tracks. What’s really scary is that a trojaned Struts or Log4j library could affect most of the financial industry all at once.

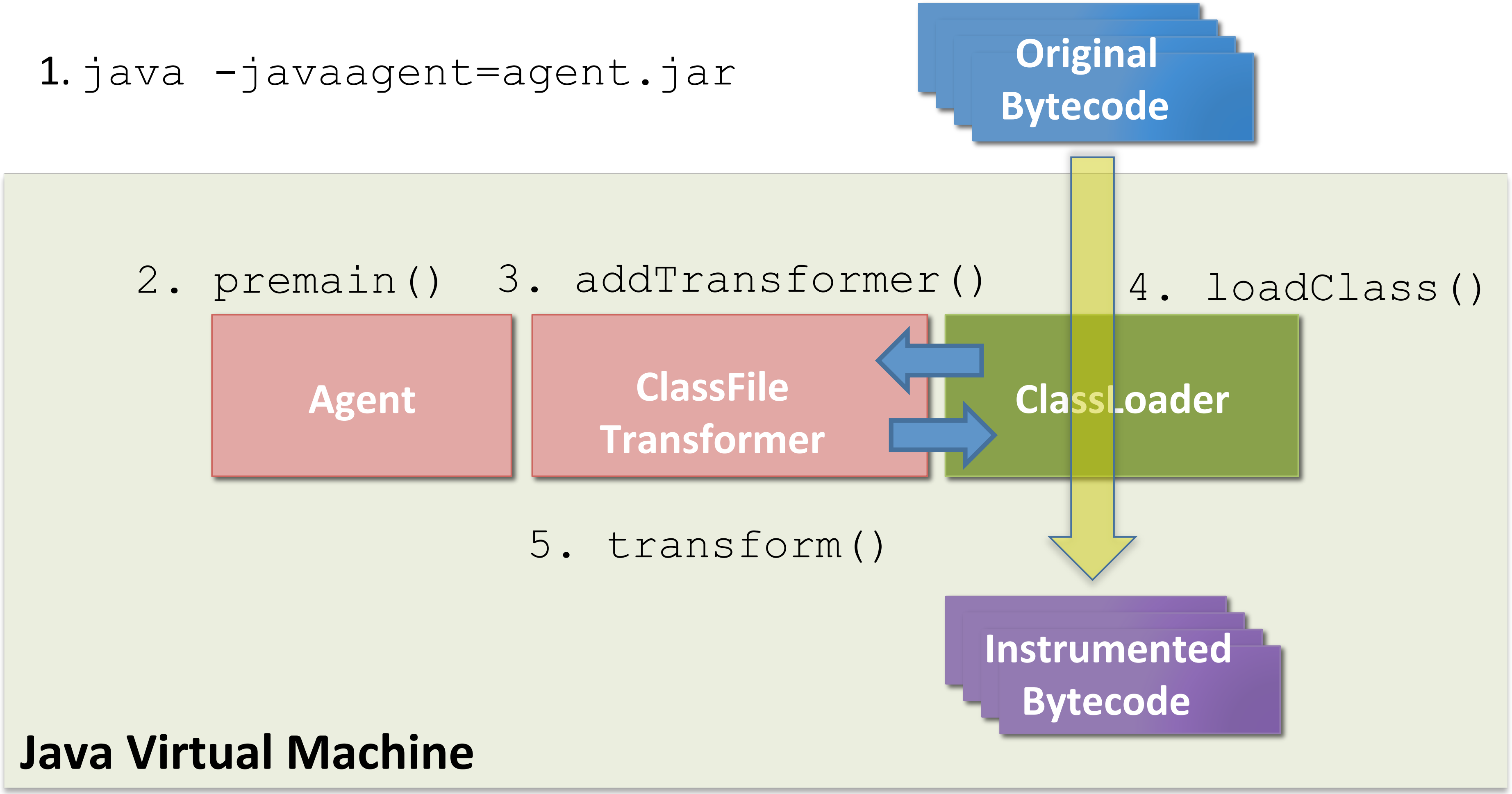
In this paper, we examine the techniques that malicious programmers can use to insert and hide these attacks in an enterprise Java application. We examine techniques for bootstrapping external attacks, avoiding code review, avoiding static analysis, trojaning libraries, and trojaning an enterprise build server. The point here is not to show how complex these attacks are, but rather how many opportunities there are and how simple and obvious they are to most developers.

The paper also presents several strategies for minimizing the risk to organizations from malicious Java developers. We evaluate the benefits and limitations of procedural controls, technical controls, and detection. In particular we focus on techniques for limiting the trust you grant developers through restricting APIs, establishing a trustworthy build process, and limiting trust in operation with the Java SecurityManager. We’ll also discuss improving detection techniques such as code review and static analysis tools.

Businesses should be aware of these risks so that they can make informed decisions about their software supply chain, and even whether to automate certain business functions at all.

https://www.blackhat.com/presentations/bh-usa-09/WILLIAMS/BHUSA09-Williams-EnterpriseJavaRootkits-PAPER.pdf

Instrumentation



1. `java -javaagent=agent.jar`

2. `premain()`

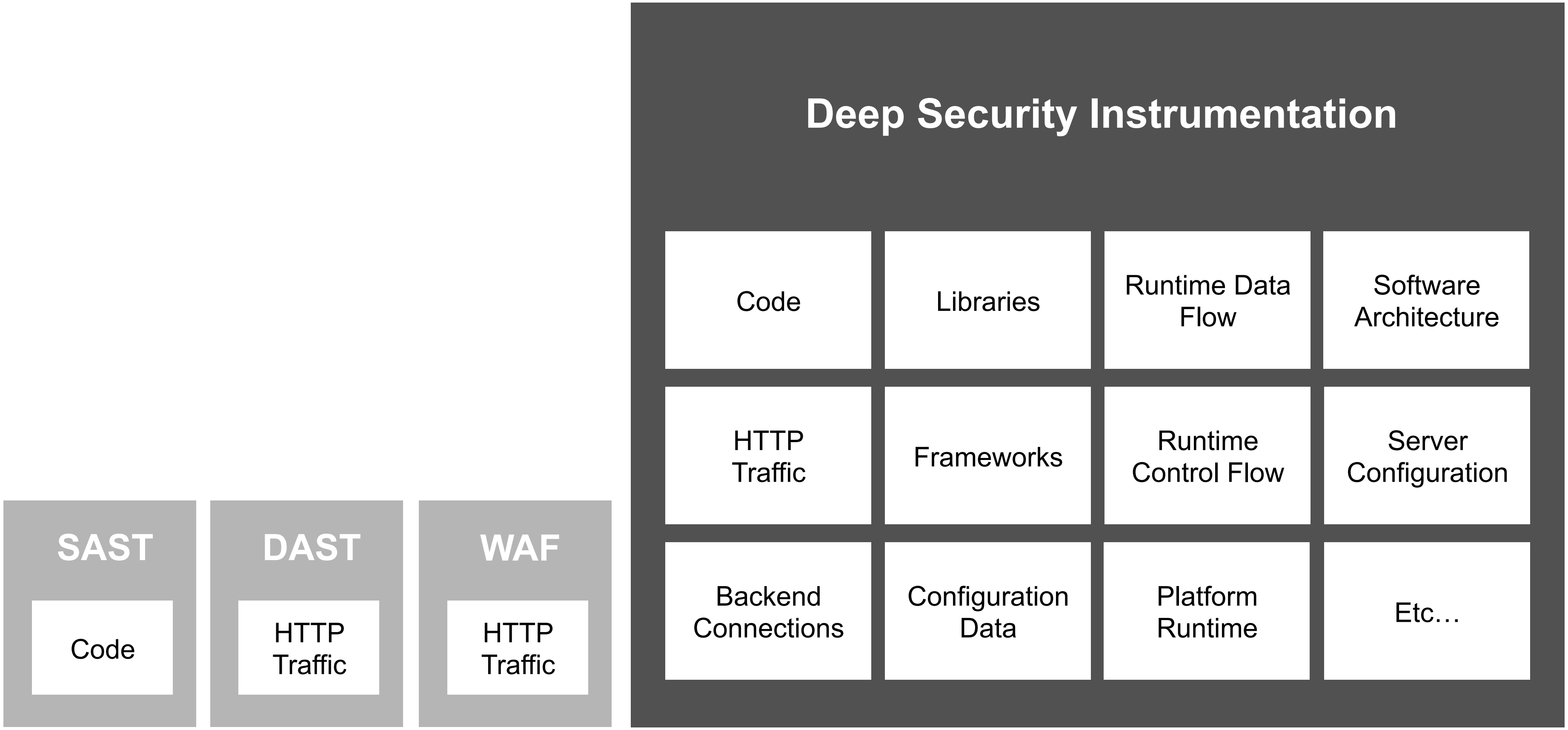
3. `addTransformer()`

4. `loadClass()`

5. `transform()`

Add sensors…. bytecode is instrumented for security!

DEEP SECURITY INSTRUMENTATION HAS UNFAIR ADVANTAGES…



SAST

Code

DAST

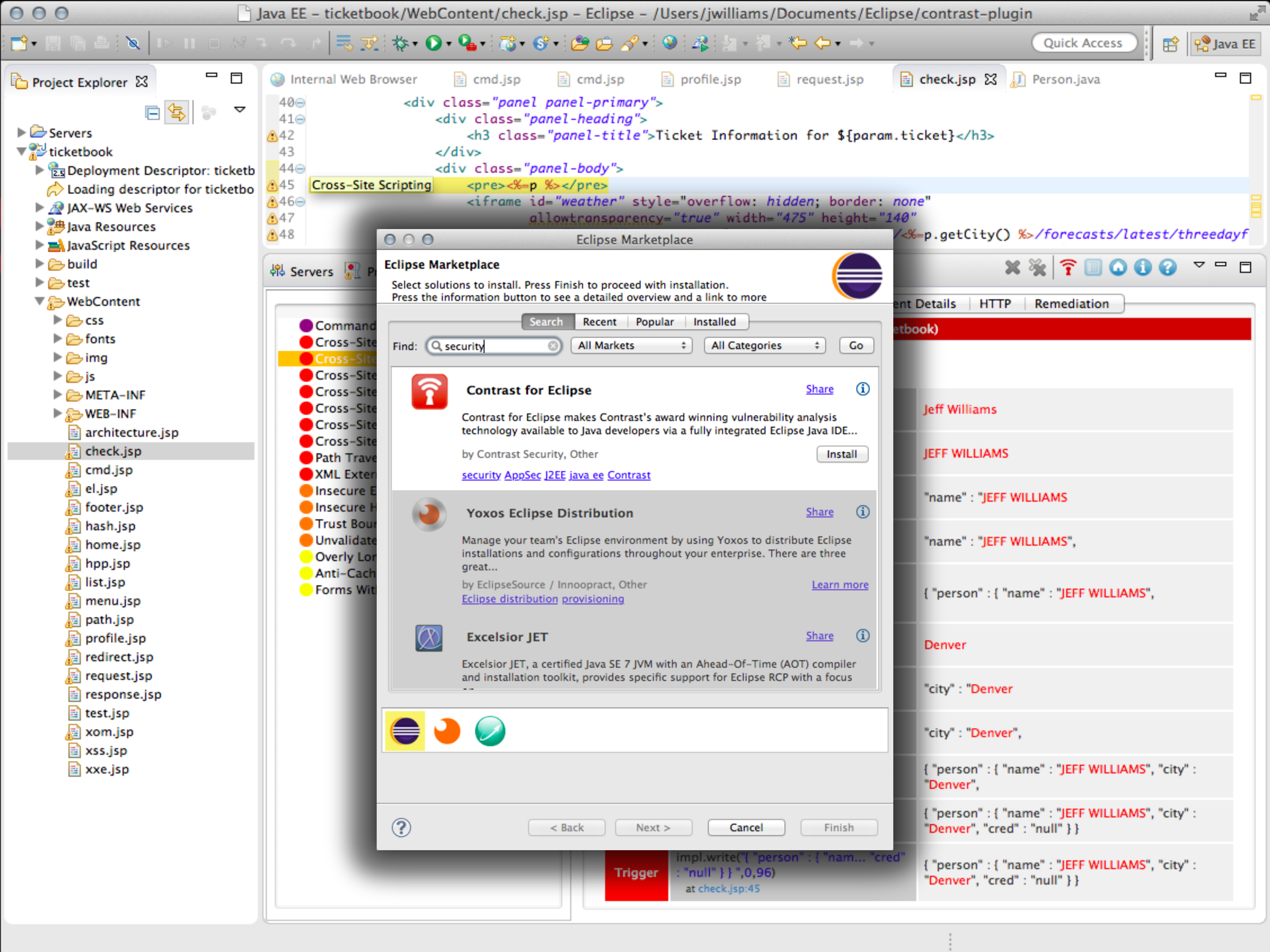
HTTP Traffic

WAF

HTTP Traffic

Deep Security Instrumentation

Code	Libraries	Runtime Data Flow	Software Architecture
HTTP Traffic	Frameworks	Runtime Control Flow	Server Configuration
Backend Connections	Configuration Data	Platform Runtime	Etc…



Java EE – ticketbook/WebContent/check.jsp – Eclipse – /Users/jwilliams/Documents/Eclipse/contrast-plugin

- Servers
- ticketbook
  - Deployment Descriptor: ticketb
  - Loading descriptor for ticketbo
  - JAX-WS Web Services
  - Java Resources
  - JavaScript Resources
  - build
  - test
  - WebContent
    - css
    - fonts
    - img
    - js
    - META-INF
    - WEB-INF
    - architecture.jsp
    - check.jsp
    - cmd.jsp
    - el.jsp
    - footer.jsp
    - hash.jsp
    - home.jsp
    - hpp.jsp
    - list.jsp
    - menu.jsp
    - path.jsp
    - profile.jsp
    - redirect.jsp
    - request.jsp
    - response.jsp
    - test.jsp
    - xom.jsp
    - xss.jsp
    - xxe.jsp

```
            <div class="panel panel-primary">
                <div class="panel-heading">
                    <h3 class="panel-title">Ticket Information for ${param.ticket}</h3>
                </div>
                <div class="panel-body">
Cross-Site Scripting <pre><%=p %></pre>
                    <iframe id="weather" style="overflow: hidden; border: none"
                        allowtransparency="true" width="475" height="140"
                                                    /<%=p.getCity() %>/forecasts/latest/threedayf
```

- Command
- Cross-Site
- Cross-Site
- Cross-Site
- Cross-Site
- Cross-Site
- Cross-Site
- Cross-Site
- Path Trave
- XML Exter
- Insecure E
- Insecure H
- Trust Boun
- Unvalidate
- Overly Lor
- Anti-Cach
- Forms Wit

Eclipse Marketplace

Eclipse Marketplace

Select solutions to install. Press Finish to proceed with installation.
Press the information button to see a detailed overview and a link to more

Find: security All Markets All Categories Go

Contrast for Eclipse Share

Contrast for Eclipse makes Contrast's award winning vulnerability analysis technology available to Java developers via a fully integrated Eclipse Java IDE...

by Contrast Security, Other Install

security AppSec J2EE java ee Contrast

Yoxos Eclipse Distribution Share

Manage your team's Eclipse environment by using Yoxos to distribute Eclipse installations and configurations throughout your enterprise. There are three great...

by EclipseSource / Innopract, Other Learn more

Eclipse distribution provisioning

Excelsior JET Share

Excelsior JET, a certified Java SE 7 JVM with an Ahead-Of-Time (AOT) compiler and installation toolkit, provides specific support for Eclipse RCP with a focus

etbook)

Jeff Williams

JEFF WILLIAMS

"name" : "JEFF WILLIAMS

"name" : "JEFF WILLIAMS",

{ "person" : { "name" : "JEFF WILLIAMS",

Denver

"city" : "Denver

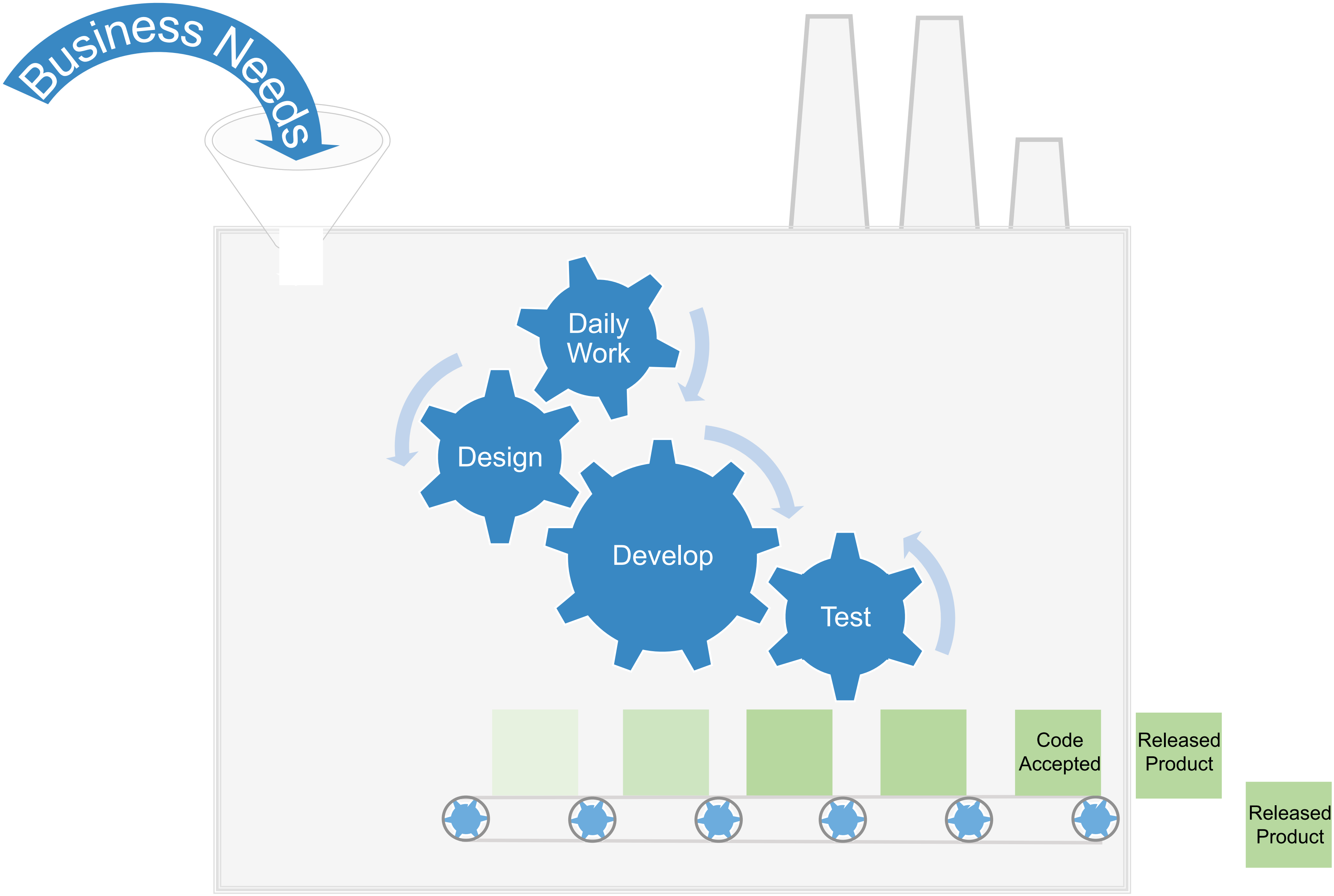
"city" : "Denver",

{ "person" : { "name" : "JEFF WILLIAMS", "city" : "Denver",

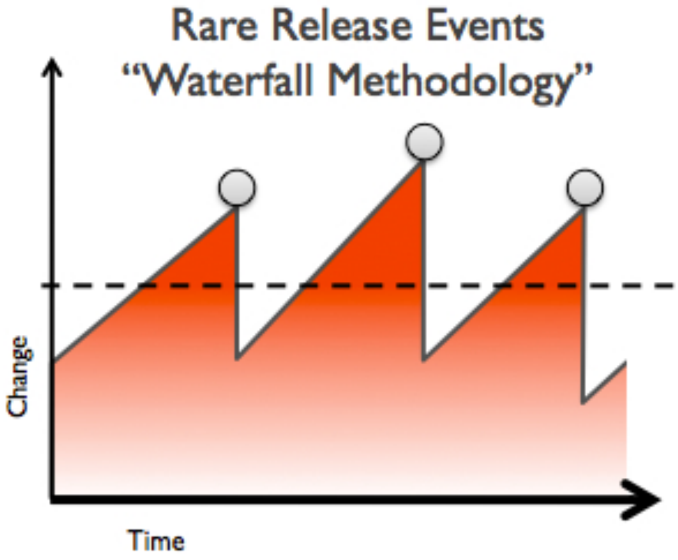
{ "person" : { "name" : "JEFF WILLIAMS", "city" : "Denver", "cred" : "null" } }

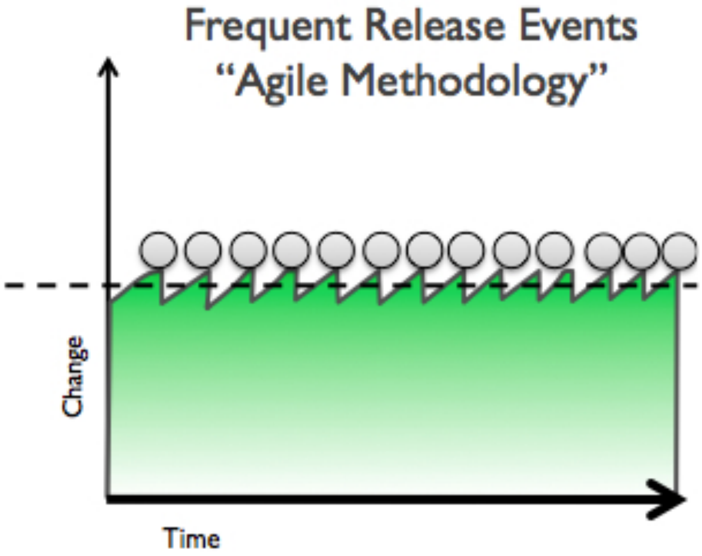
Trigger impl.write("{ "person" : { "nam... "cred" : "null" } } ",0,96)
at check.jsp:45

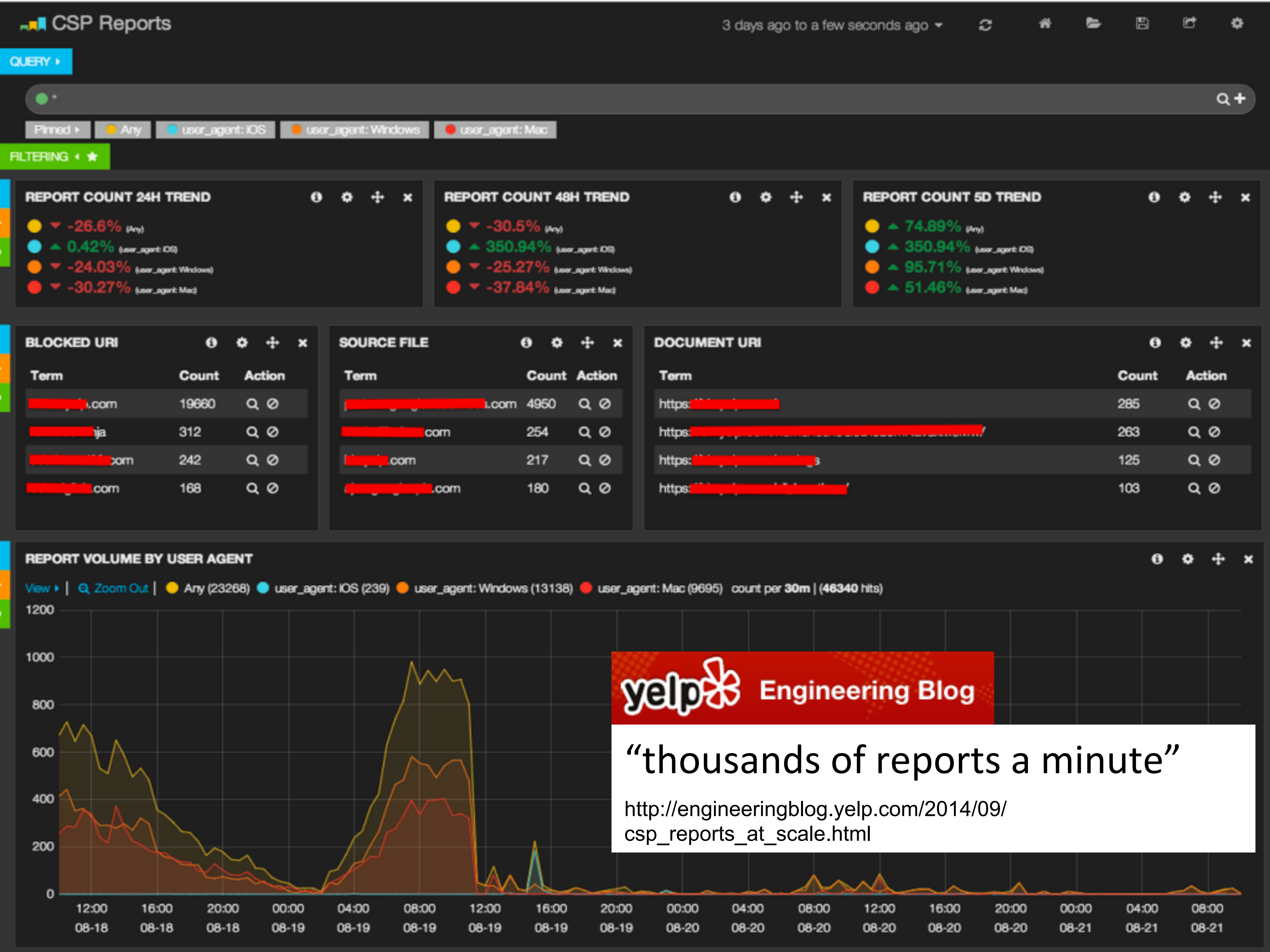
{ "person" : { "name" : "JEFF WILLIAMS", "city" : "Denver", "cred" : "null" } }


Business Needs
Daily Work
Design
Develop
Test
Code Accepted
Released Product
Released Product

Continuous AppSec


Rare Release Events
“Waterfall Methodology”


Frequent Release Events
“Agile Methodology”



CSP Reports

3 days ago to a few seconds ago

QUERY ▸

Pinned ▸ | ● Any | ● user_agent: iOS | ● user_agent: Windows | ● user_agent: Mac

FILTERING ◂ ★

REPORT COUNT 24H TREND

- ● ▼ -26.6% (Any)
- ● ▲ 0.42% (user_agent: iOS)
- ● ▼ -24.03% (user_agent: Windows)
- ● ▼ -30.27% (user_agent: Mac)

REPORT COUNT 48H TREND

- ● ▼ -30.5% (Any)
- ● ▲ 350.94% (user_agent: iOS)
- ● ▼ -25.27% (user_agent: Windows)
- ● ▼ -37.84% (user_agent: Mac)

REPORT COUNT 5D TREND

- ● ▲ 74.89% (Any)
- ● ▲ 350.94% (user_agent: iOS)
- ● ▲ 95.71% (user_agent: Windows)
- ● ▲ 51.46% (user_agent: Mac)

BLOCKED URI

Term	Count	Action
[redacted].com	19660	🔍 ⊘
[redacted]ja	312	🔍 ⊘
[redacted]com	242	🔍 ⊘
[redacted].com	168	🔍 ⊘

SOURCE FILE

Term	Count	Action
[redacted].com	4950	🔍 ⊘
[redacted]com	254	🔍 ⊘
[redacted].com	217	🔍 ⊘
[redacted].com	180	🔍 ⊘

DOCUMENT URI

Term	Count	Action
https:[redacted]	285	🔍 ⊘
https:[redacted]/	263	🔍 ⊘
https:[redacted]s	125	🔍 ⊘
https:[redacted]/	103	🔍 ⊘

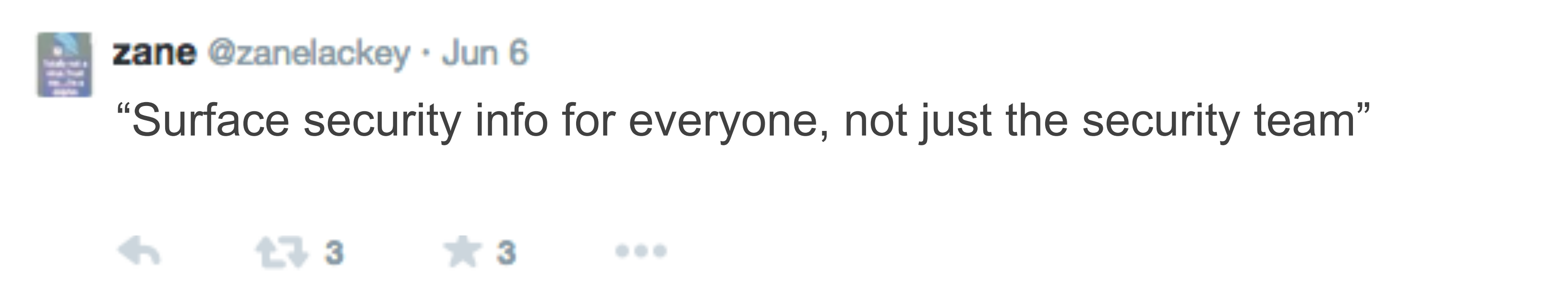
REPORT VOLUME BY USER AGENT

View ▸ | 🔍 Zoom Out | ● Any (23268) ● user_agent: iOS (239) ● user_agent: Windows (13138) ● user_agent: Mac (9695) count per 30m | (46340 hits)

yelp Engineering Blog

“thousands of reports a minute”

http://engineeringblog.yelp.com/2014/09/csp_reports_at_scale.html


zane @zanelackey · Jun 6

“Surface security info for everyone, not just the security team”

APPLICATION SECURITY – TWO SEPARATE WORLDS

Development

Tools to detect vulnerabilities

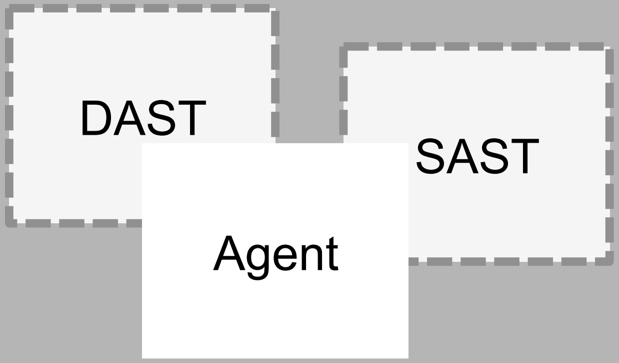
Operations

Tools to stop attacks

TODAY – EVOLUTIONARY ADVANCES

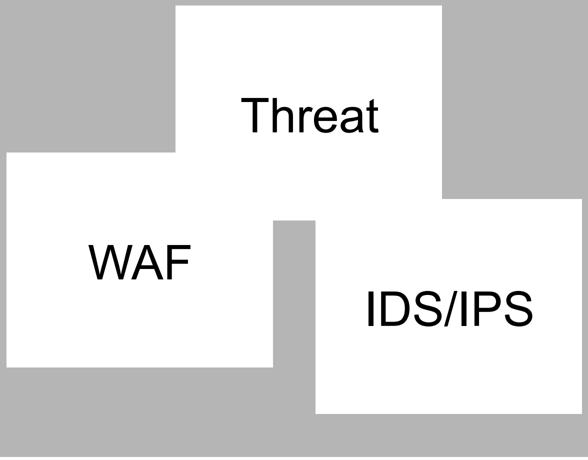
Development

IAST
Interactive Application Security Testing


DAST
SAST
Agent

Operations

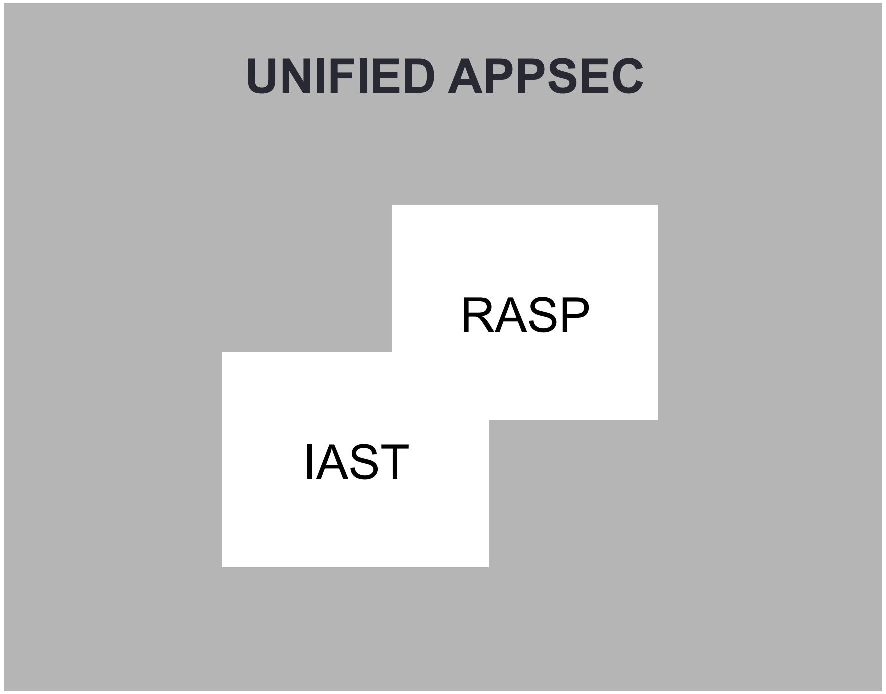
RASP
Runtime Application Self-Protection


Threat
WAF
IDS/IPS

“Must have technologies” – Joseph Feiman (Gartner)

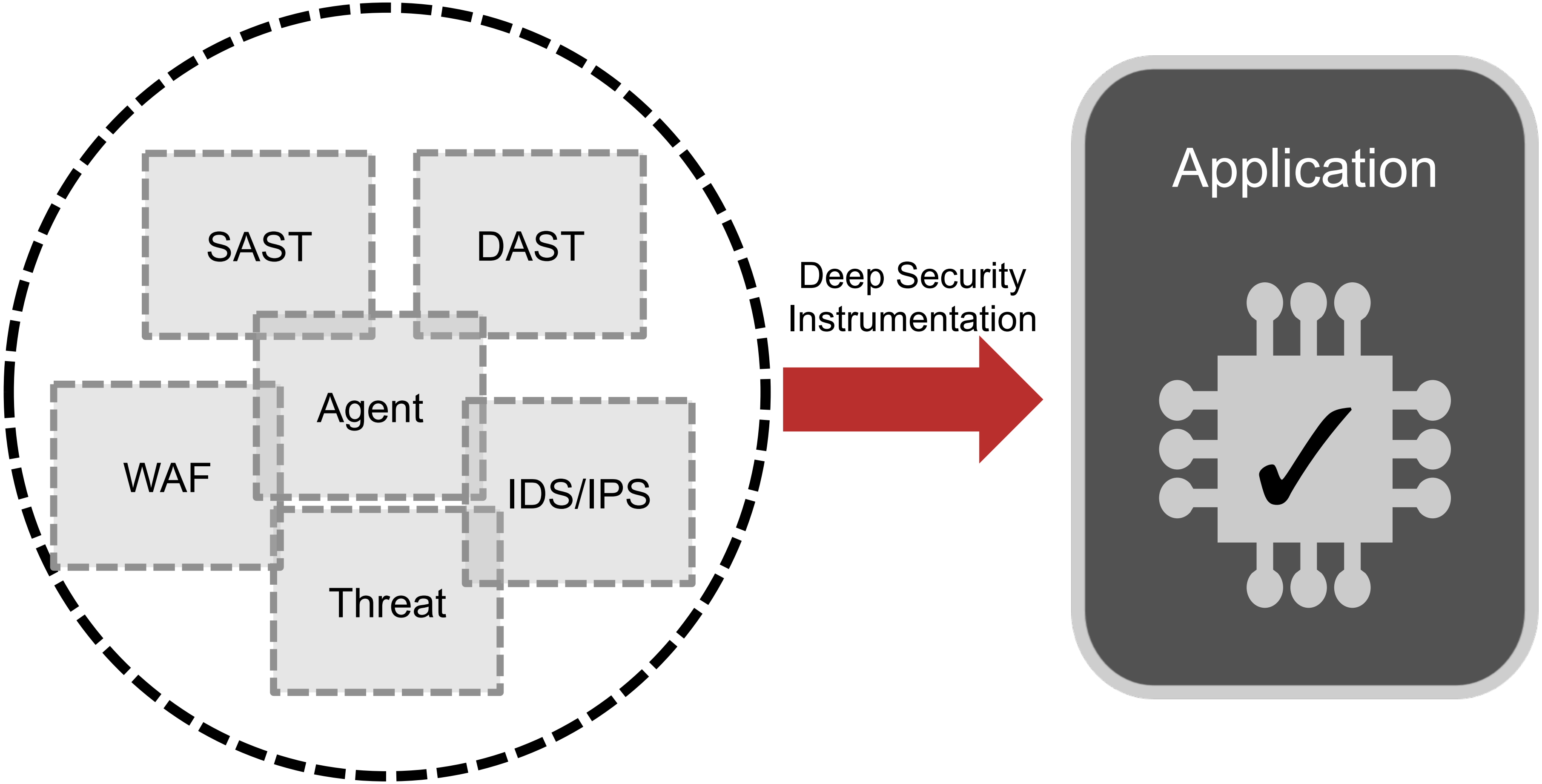
TOMORROW – UNIFIED APPSEC

Development **AND** Operations

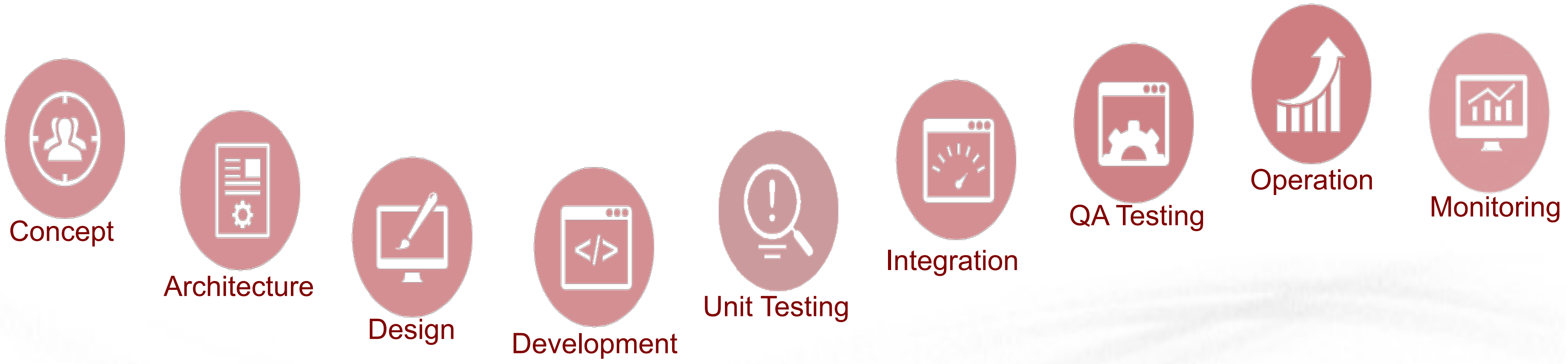


A single appsec technology across the entire lifecycle

INFUSE SECURITY TECHNOLOGIES INTO APPLICATIONS



EXAMPLE: CODE VULNERABILITIES (SQL INJECTION)



Concept

Architecture

Design

Development

Unit Testing

Integration

QA Testing

Operation

Monitoring

Generate accurate security architecture diagrams

Provide instant vulnerability feedback

Add security testing to existing test efforts

Identify and block attacks and exploits

EXAMPLE: THIRD PARTY LIBRARIES (SPRING EL INJECTION)

Employee Apps

Third Party Apps

Public Apps

Cloud Apps

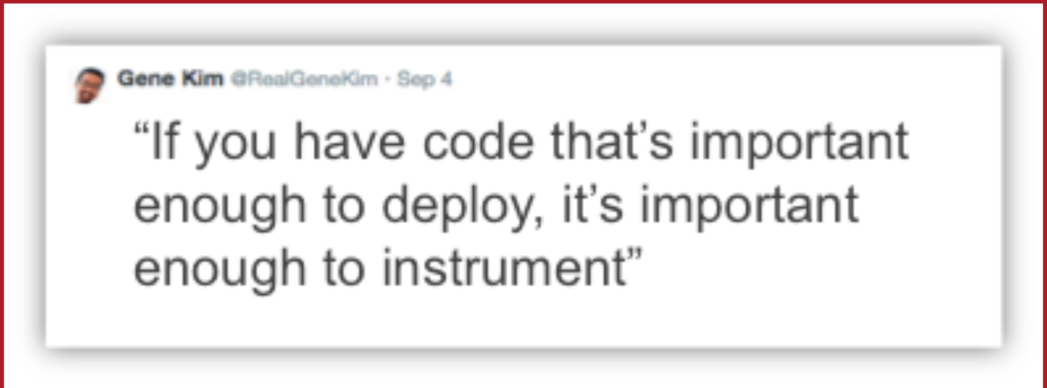
“Rogue” Apps

Automatically collect library inventory

Notify projects of library vulnerabilities

Ensure that developers use libraries safely

Shield applications from attacks on known vulnerabilities



Gene Kim @RealGeneKim · Sep 4

"If you have code that's important enough to deploy, it's important enough to instrument"

The world's fastest application security.

CONTRAST
SECURITY