

# Application Security: Protecting Your Applications Against Exploit.

What can I do ASAP, this quarter, this year?

**Paul Kitor**

Security Solution Architect

**HP Enterprise Security Products**



# Agenda

## Protecting Your Applications Against Exploit



### Why Application Security?

---



### Protection / Assessment / Prevention

---

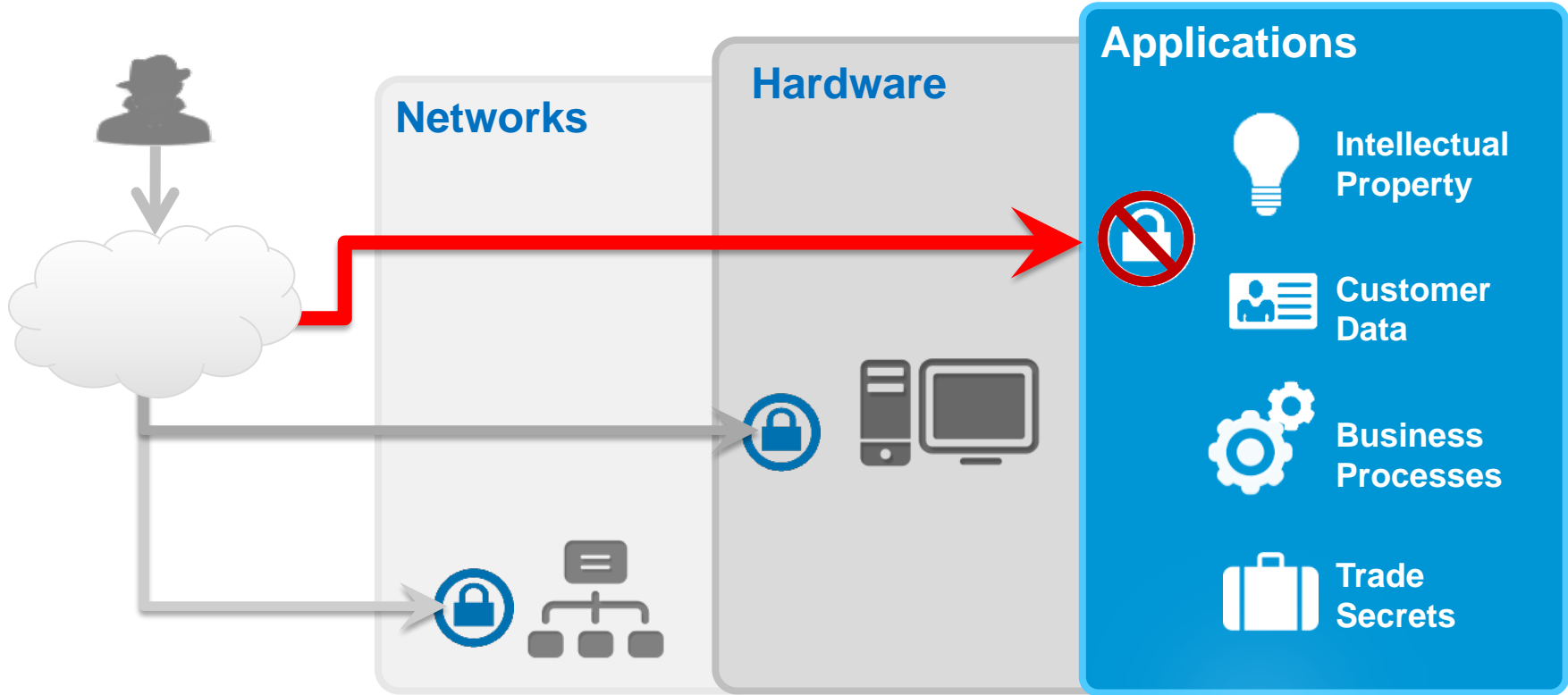


### Software Security Assurance

# The Application Security Problem



# Cyber attackers are targeting applications

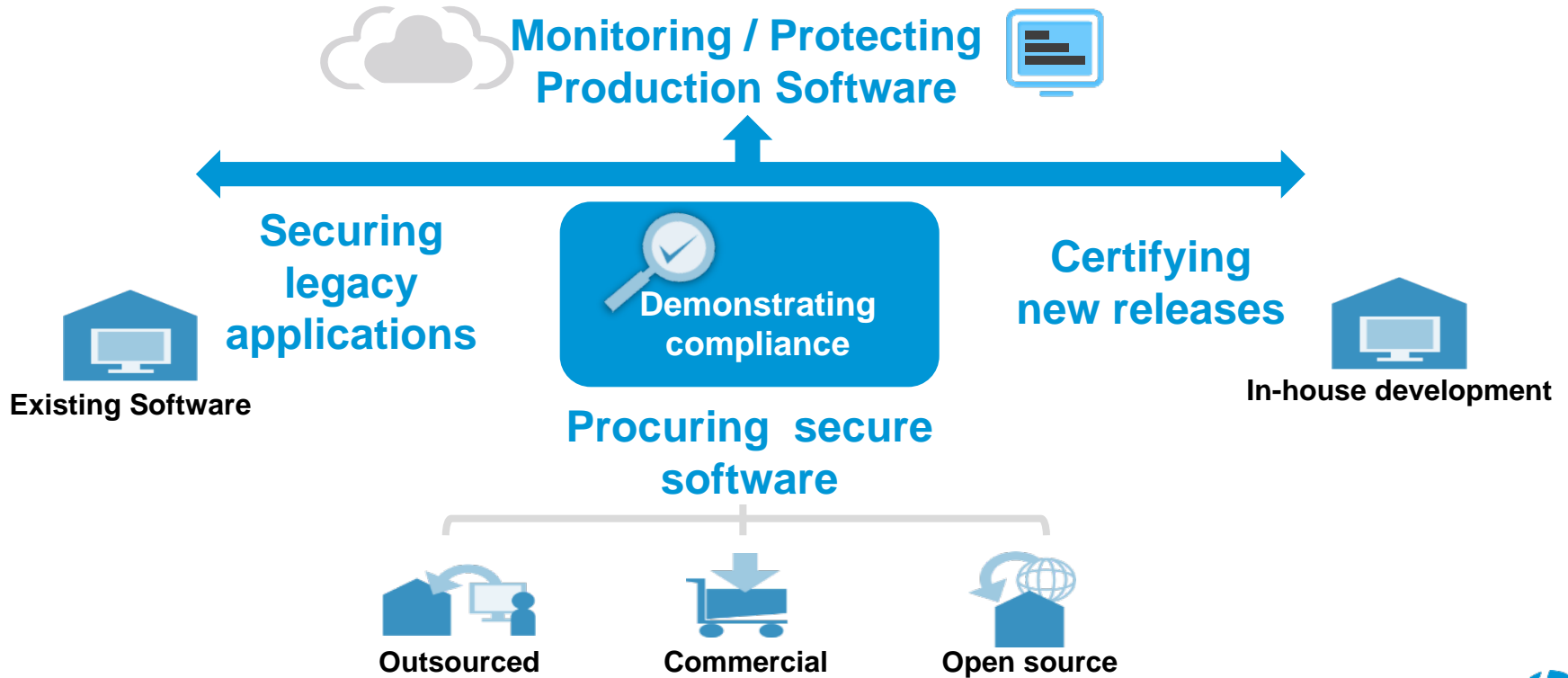




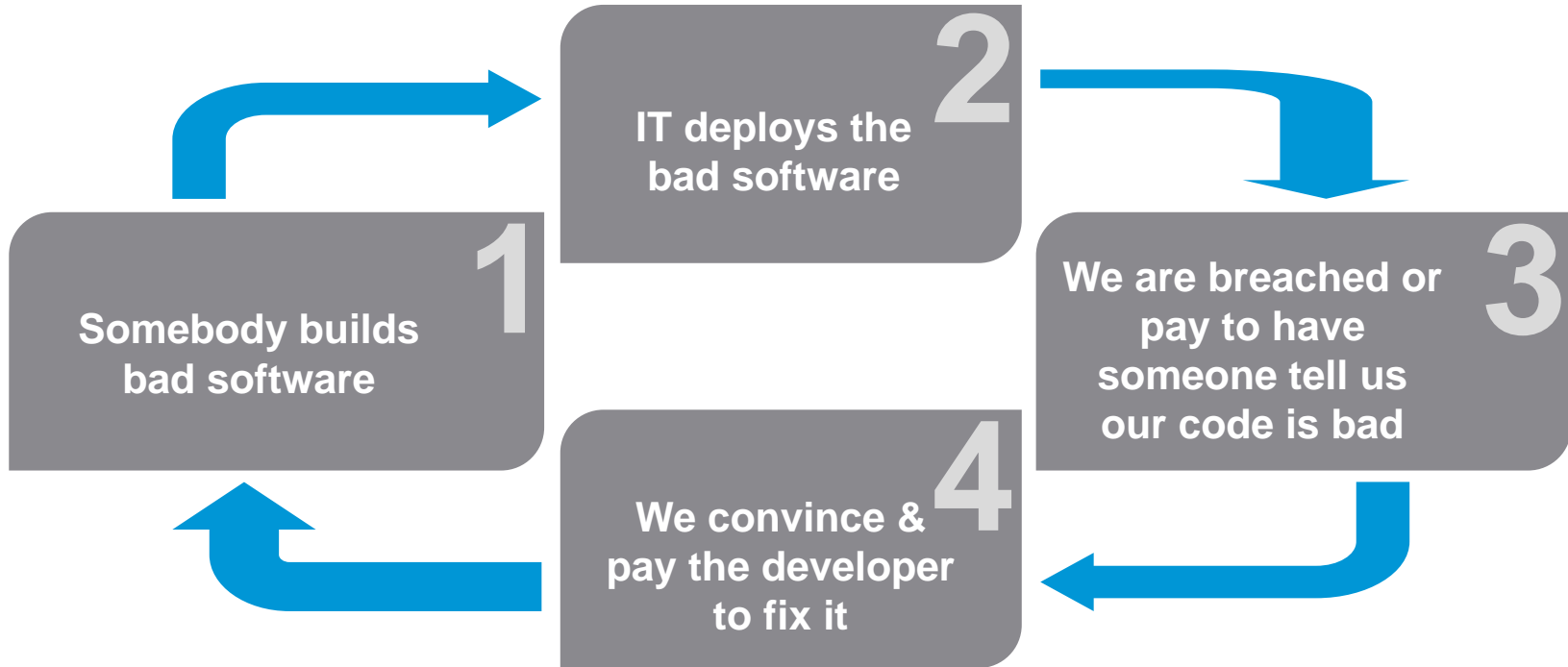
**84%** of breaches occur at the application layer



# Application security challenges

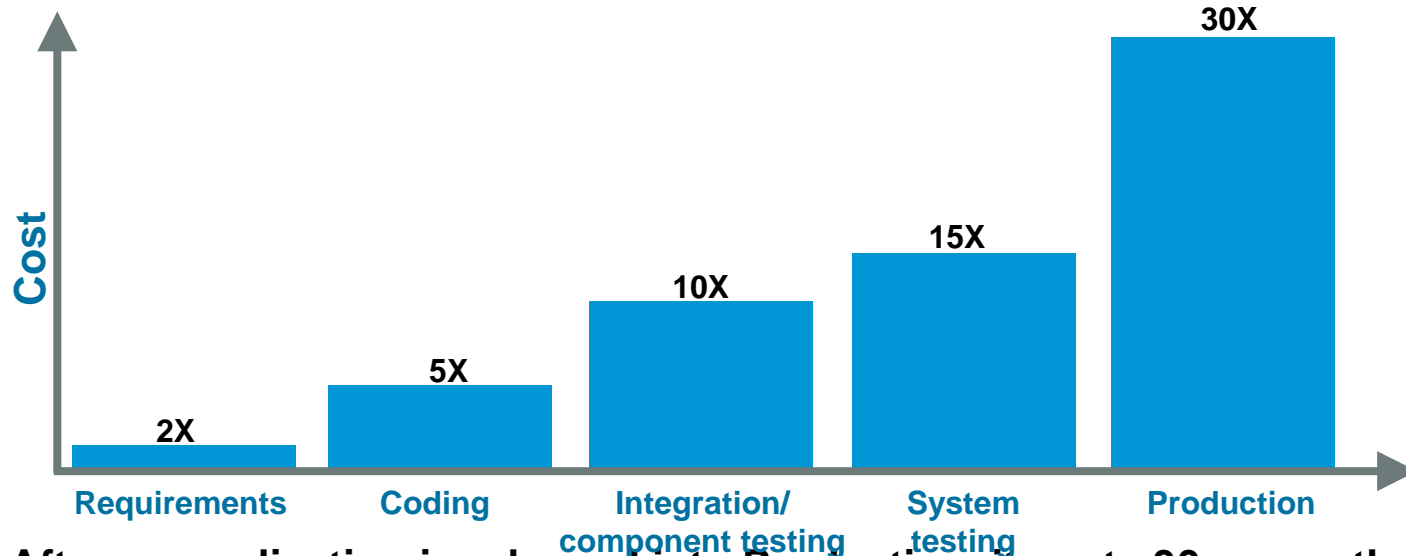


# Today's approach > expensive, reactive



# Why it doesn't work

30x more costly to remediate in production

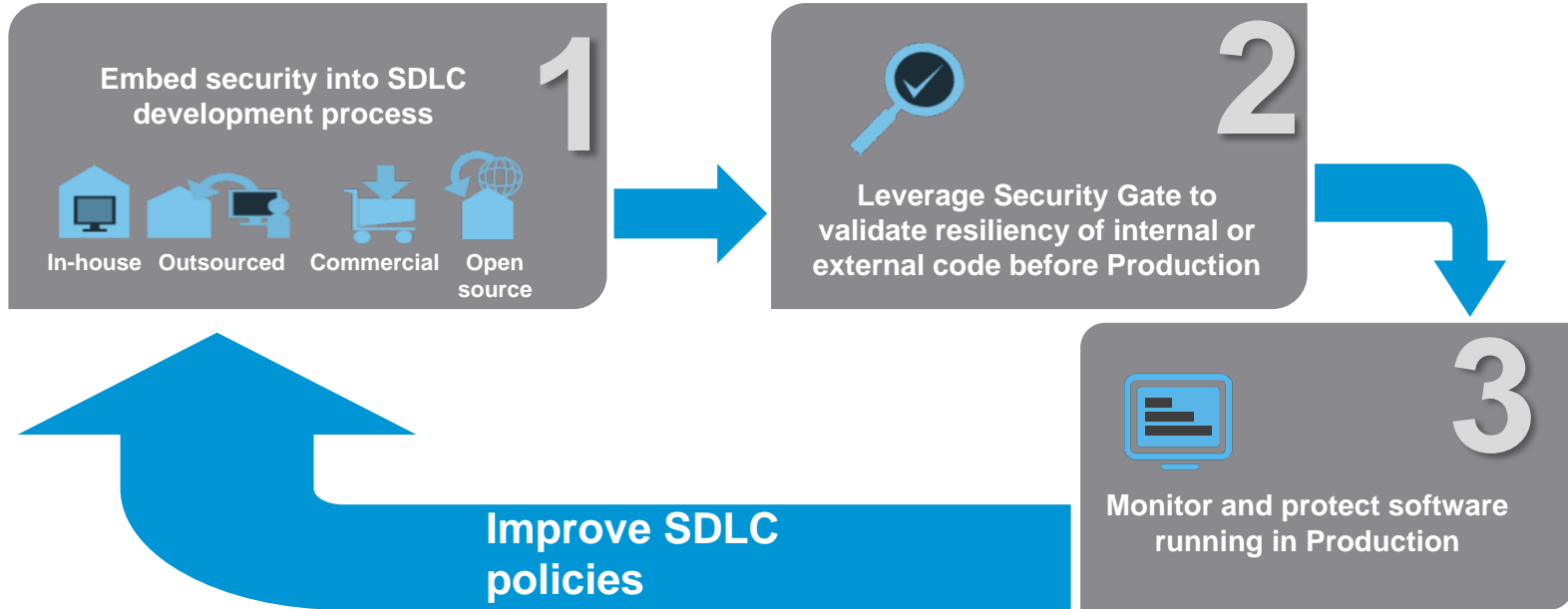


**After an application is released into Production, it costs 30x more than during design.**

Source: NIST



# The right approach > systematic, proactive



**This is application security**

# Agenda

## Protecting Your Applications Against Exploit



### Why Application Security?

---



### Protection / Assessment / Prevention

---



### Software Security Assurance

# The HP Fortify Software Security Vision

## Application Protection



### Protect

Fortify applications against attack in production

*Logging, Threat Protection*

## Application Assessment



### Assess

Find security vulnerabilities in any type of software

*Mobile, Web, Infrastructure*

## Software Security Assurance (SSA)



### Assure

Fix security flaws in source code before it ships

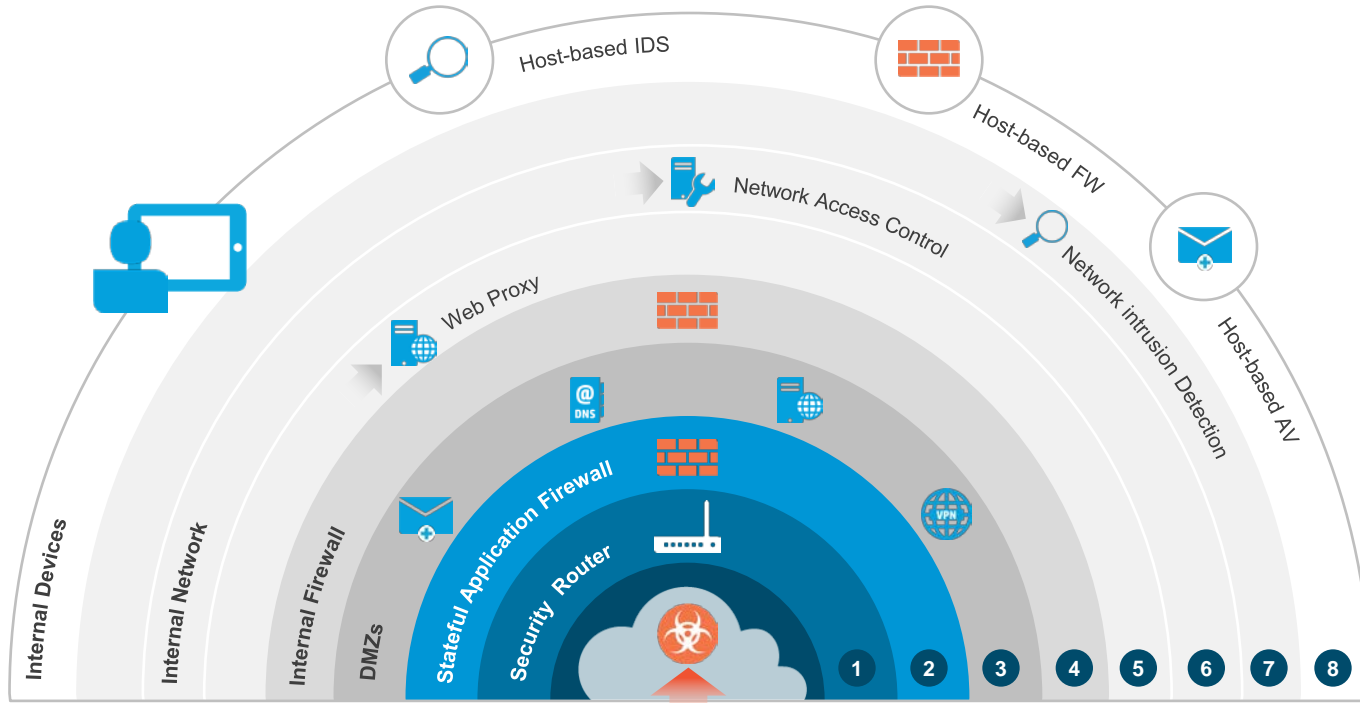
*Secure SDLC*

# What can I do ASAP?

## The solution: Protection



# Current solutions protect the perimeter



Yet, **84%** of breaches occur in the application software

# Runtime Application Self-Protection

Runtime application self-protection (RASP) is a security technology that is built or linked into an application or application runtime environment, and is capable of controlling application execution and detecting and preventing real-time attacks.

## Application Protection

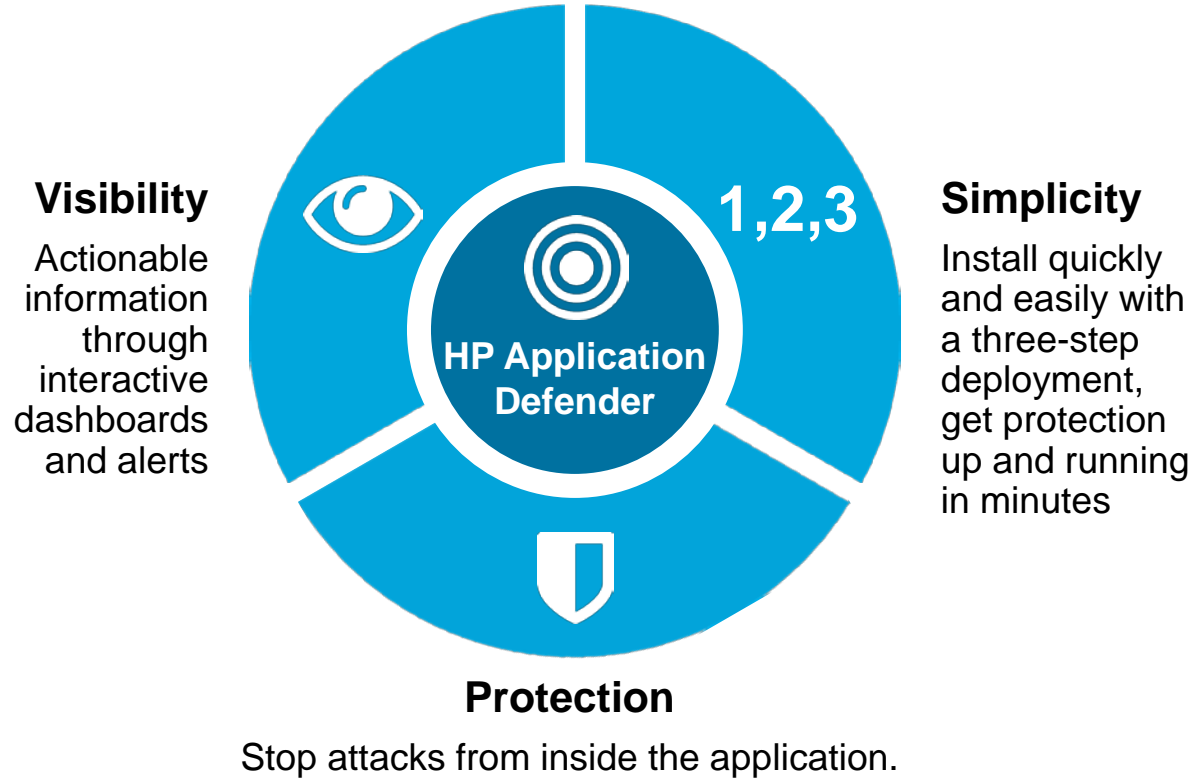


### Protect

Fortify applications against attack in production

*Logging, Threat Protection*

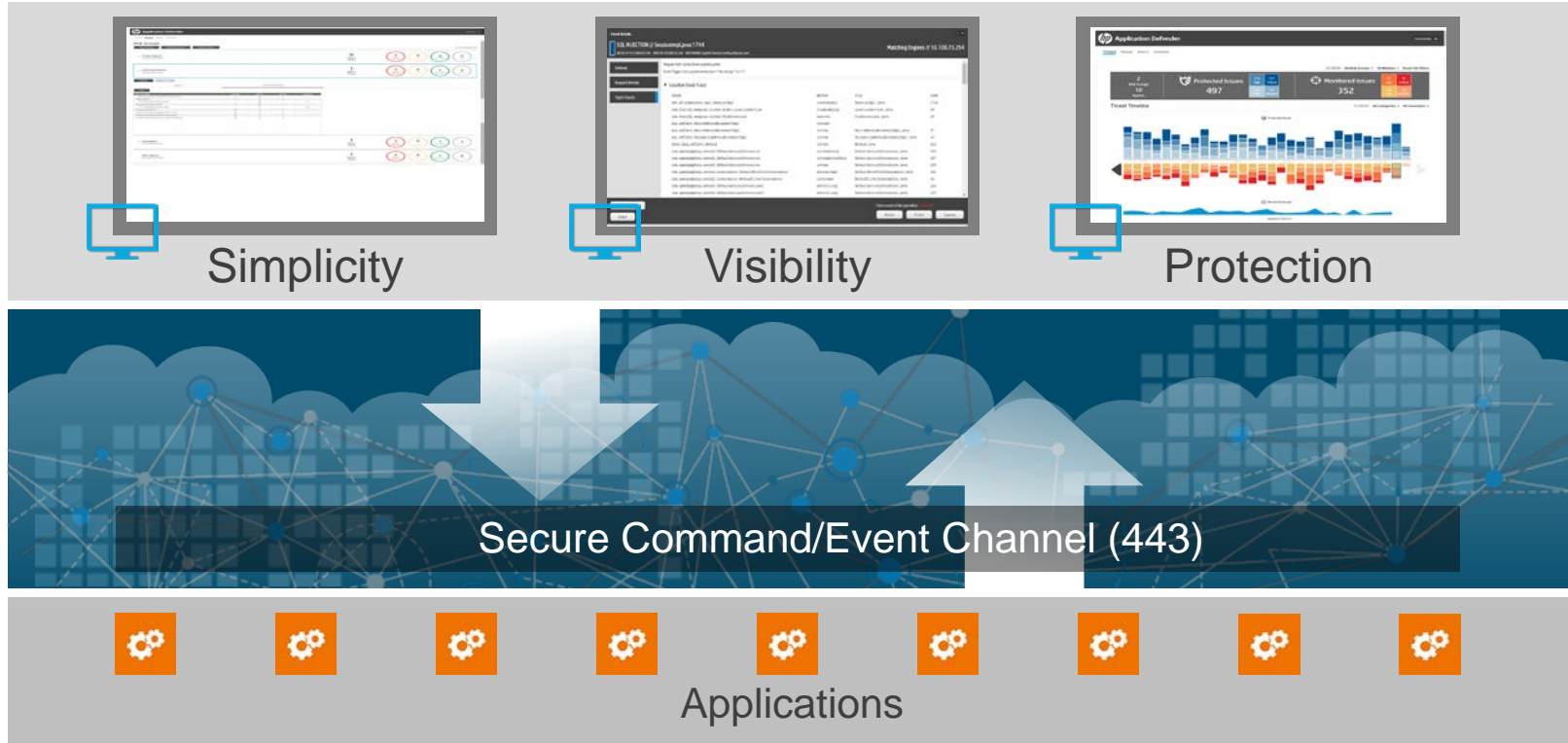
# HP Application Defender – Application Security Simplified



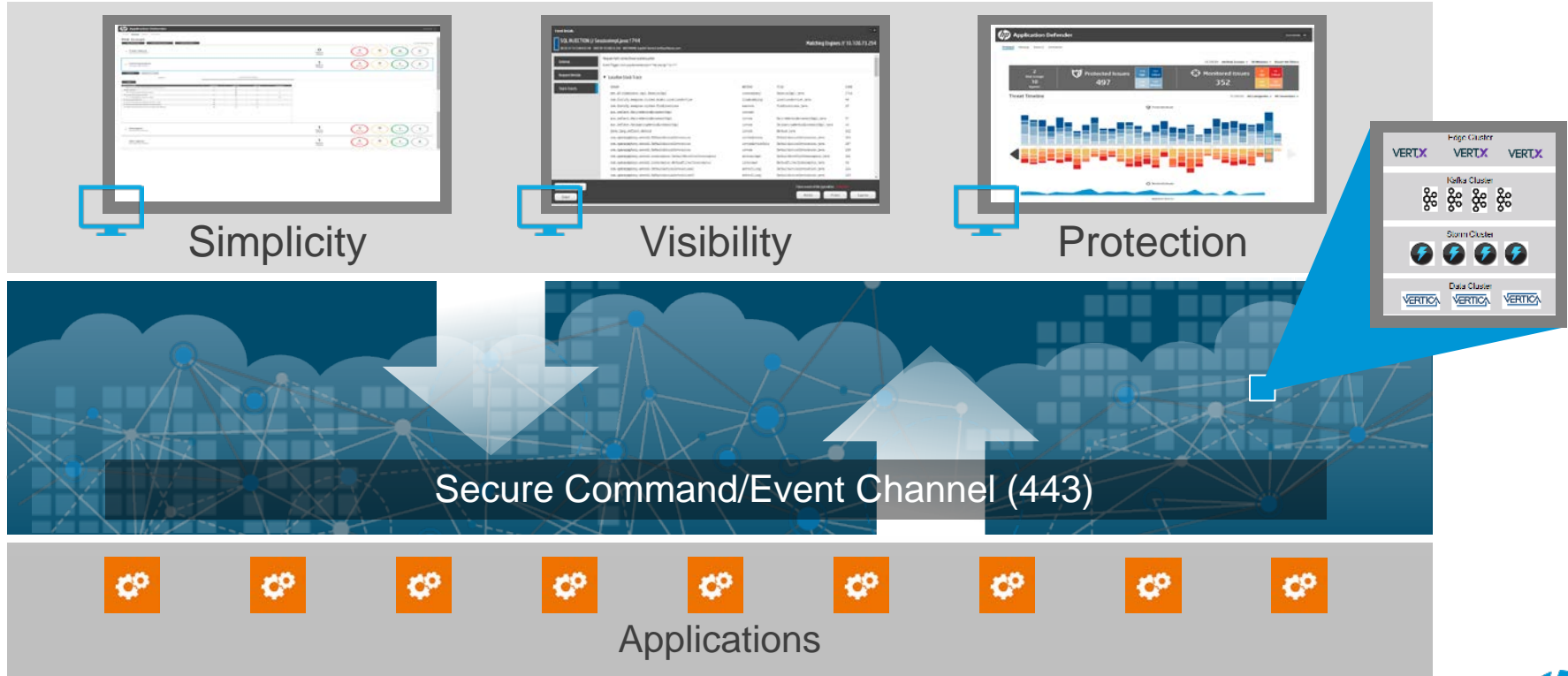
**We stop what no one else can even see**



# HP Application Defender Solution



# HP Application Defender Solution



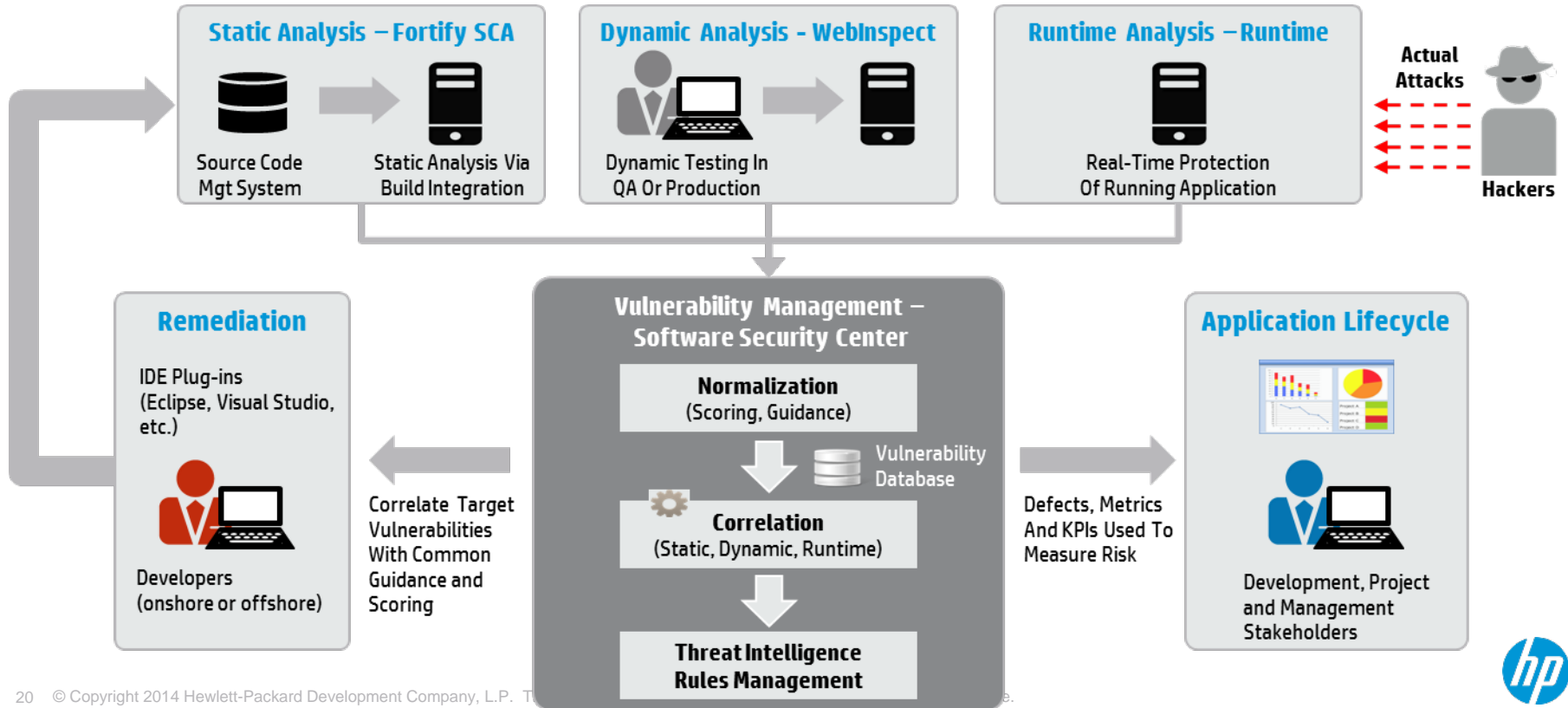
# What can I do this Quarter?

## Assessment



# HP Fortify – Software Security Assurance

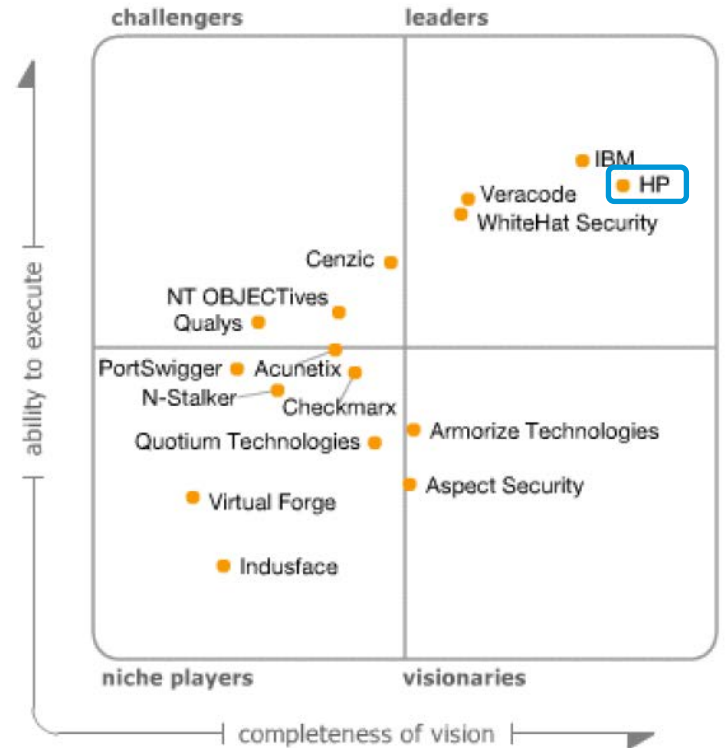
On-Premise and On-Demand



# HP Fortify Named a Leader in Gartner Magic Quadrant

## Gartner Application Security Testing

- “HP offers comprehensive SAST capabilities with Fortify’s strong brand name and breadth of languages tested.
- The company has innovative IAST capability with Fortify SecurityScope, which integrates with its WebInspect DAST.
- There is strong integration within HP’s security portfolio, such as integration of AST knowledge into ArcSight and DAST knowledge into TippingPoint’s IPS for WAF-like protection.
- HP uniquely offers runtime application self-protection (RASP) technology”




As of July 2013



# Security Testing

**Application  
Assessment**



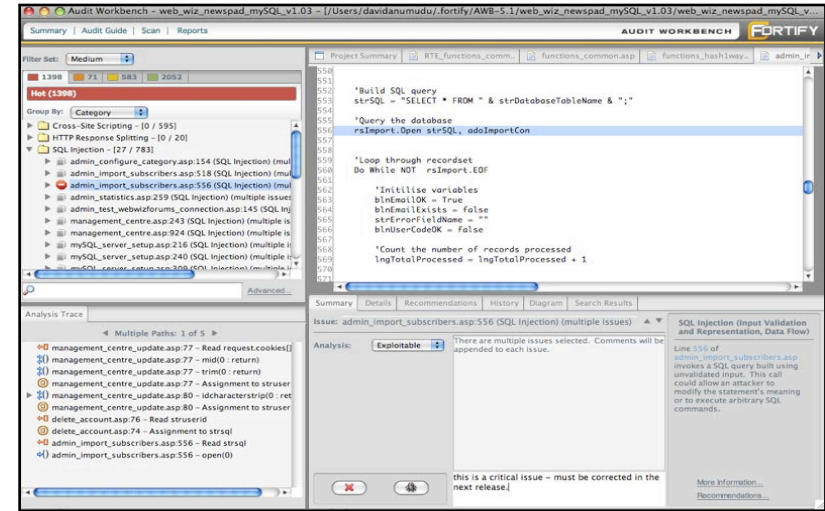
**Assess**  
Find security vulnerabilities in  
any type of software  
*Mobile, Web, Infrastructure*

# HP Fortify Static Code Analyzer (SCA)

Static analysis – find and fix security issues in your code during development

## Features:

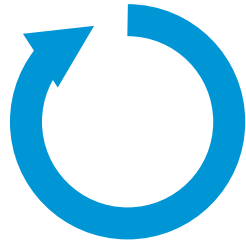
- Automate static application security testing to identify security vulnerabilities in application source code during development
- Pinpoint the root cause of vulnerabilities with line of code details and remediation guidance
- Prioritize all application vulnerabilities by severity and importance
- Supports 21 languages, 500+ vulnerability categories





# HP Fortify on Demand

Get results fast with security testing software-as-a-service



## Simple

**Launch your application security initiative in <1 day**

- No hardware or software investments
- No security experts to hire, train and retain



## Fast

**Scale to test all applications in your organization**

- 1 day turn-around on application security results
- Support 1000s of applications for the desktop, mobile or cloud



## Flexible

**Test any application from anywhere**

- Secure commercial, open source and 3<sup>rd</sup> party applications
- Test applications on-premise or on demand, or both



# HP Fortify on Demand at a glance

## Comprehensive and accurate

Static Testing

HP Fortify  
SCA

Dynamic Testing

HP  
WebInspect

Audit & Analysis

Manual

## Powerful remediation

Analysis & Reports



Online Collaboration



## Broad support

- ABAP
- C/C++
- Cold Fusion
- Java
- Objective C
- Python
- ASP.NET
- Classic ASP
- Flex
- JavaScript/AJAX
- PHP
- T-SQL
- C#
- COBOL
- HTML
- JSP
- PL/SQL
- VB.NET
- VB6
- VBScript
- XML
- Ruby

## Fast, secure & scalable

1 Day Static  
Turnaround



Virtual Scan  
Farm



Encryption



Third Party  
Reviews



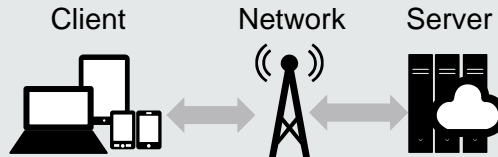
## Mobile Security Testing

### All platforms

- Apple iOS
- Android
- Windows, Blackberry

### Multiple analysis types

- Source Code
- Running Application
- Protocol Analysis



## Breadth of testing

- 10,000+ applications
- 18 different industries represented
- 5 Continents
- Civilian and Defense Agencies across US Government
- Vendor Management and Internal Management
- Development teams from 1 to 10,000s




# What can I do this Year?

## Prevention



# Secure Development

**Software Security Assurance (SSA)**



**In-house**   **Outsourced**   **Commercial**   **Open source**

**Assure**  
Fix security flaws in source code before it ships  
*Secure SDLC*

# Open Software Assurance Maturity Model

[www.opensamm.org](http://www.opensamm.org)

The Software Assurance Maturity Model (SAMM) is an open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization. The resources provided by SAMM will aid in:

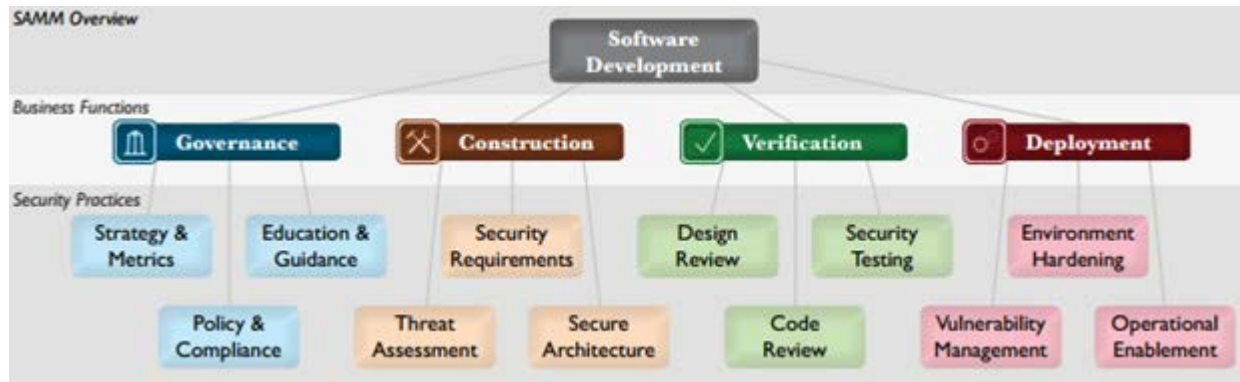
- ◇ *Evaluating an organization's existing software security practices*
- ◇ *Building a balanced software security program in well-defined iterations*
- ◇ *Demonstrating concrete improvements to a security assurance program*
- ◇ *Defining and measuring security-related activities within an organization*



# Open Software Assurance Maturity Model

[www.opensamm.org](http://www.opensamm.org)

OpenSAMM defines four business functions: **governance**, **construction**, **verification**, and **deployment**. Each of these has three associated security practices. And for each security practice, SAMM defines three maturity levels. It provides the following visualization.



# Building Security In Maturity Model

<http://bsimm.com>

The Building Security In Maturity Model is the result of a study of 67 software security initiatives. It describes activities companies are currently doing around software security so you can compare them with what your organization is doing. With your organizational goals and objectives in mind, you can use BSIMM to determine which additional activities might make sense. It gives you an overview of how software security teams tend to be organized, and how large those teams typically are.



# Building Security In Maturity Model

<http://bsimm.com>

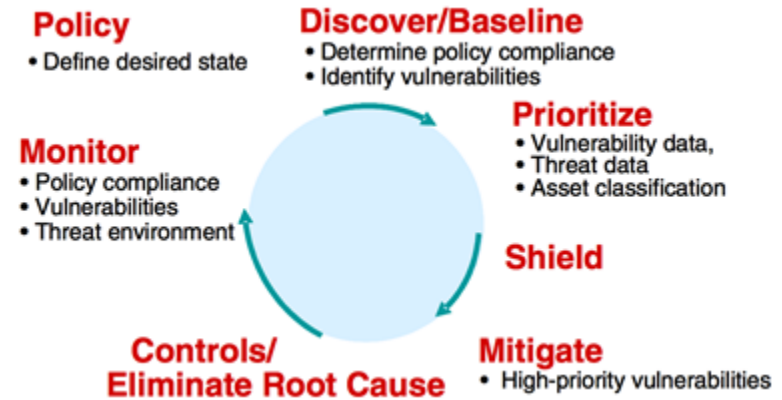
The BSIMM is organized similarly to OpenSAMM into what it calls the Software Security Framework (SSF). It has four domains: governance, intelligence, secure software development lifecycle (SSDL) touchpoints, and deployment.

The Software Security Framework (SSF)			
Governance	Intelligence	SSDL Touchpoints	Deployment
Strategy and Metrics	Attack Models	Architecture Analysis	Penetration Testing
Compliance and Policy	Security Features and Design	Code Review	Software Environment
Training	Standards and Requirements	Security Testing	Configuration Management and Vulnerability Management



# Vulnerability Management

Vulnerability management is the “cyclical practice of identifying, classifying, remediating, and mitigating vulnerabilities.” Gartner defines it as an ongoing multi-step process.



# Software Application Security

Management, tracking and remediation of enterprise software risk

## Features:

- Specify, communicate and track security activities on software projects
- Role-based, process-driven management of software security program
- Integrations into key development environments
  - Build integration, defect tracking, source control, 3rd party analysis engines
- Flexible repository and reporting platform for security status, trending and compliance
  - Normalized, correlated vulnerability repository
  - Aggregated risk metrics

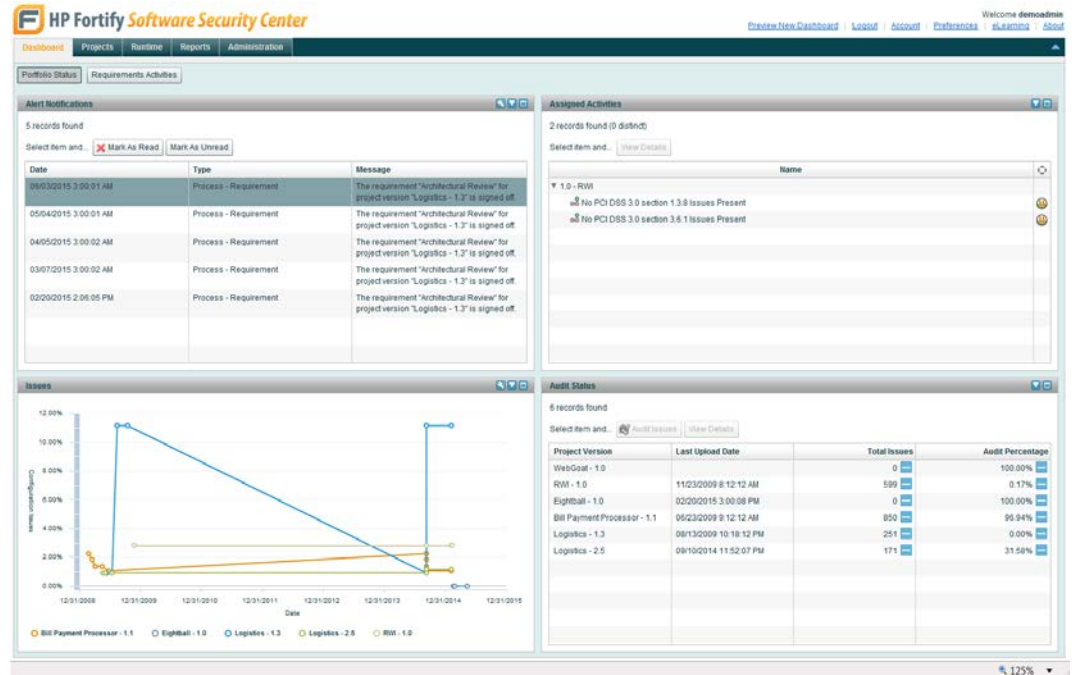


# HP Fortify Software Security Center

## Configurable Dashboards

Reveal the requirements and assigned activities required to manage application risk across your portfolio.

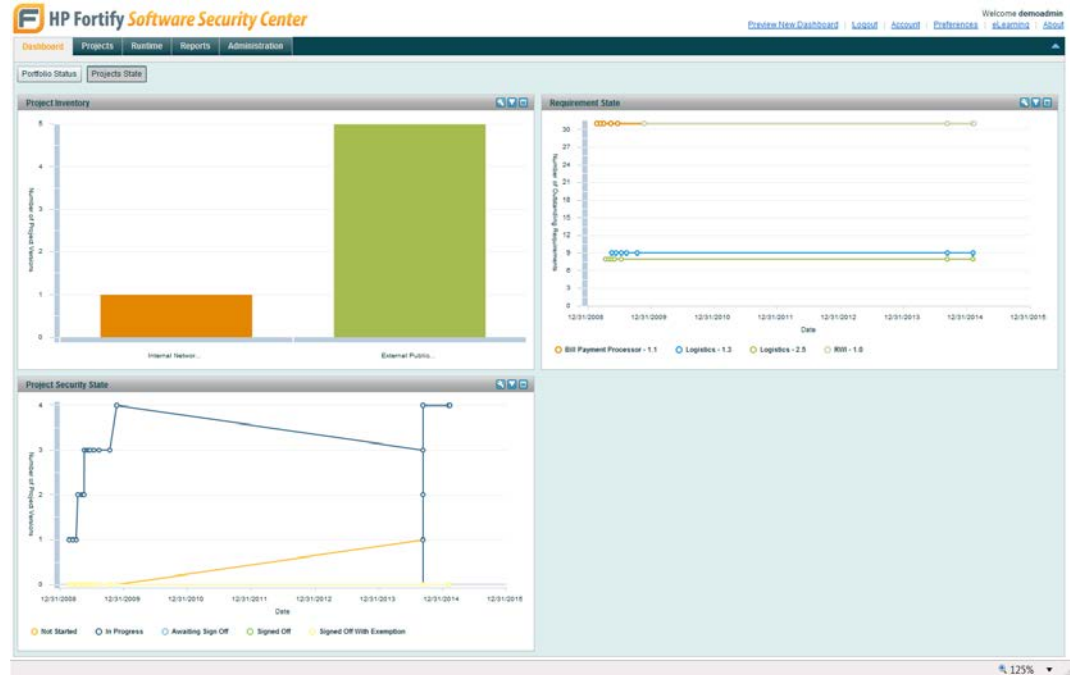
Present the portfolio metrics that track the success of your Software Security Assurance process.



# HP Fortify Software Security Center

## Configurable Dashboards

Reveal the state of each project in a selection of dashboard views.



# HP Fortify Software Security Center

## Project Summary

Select any project from your portfolio to get a quick Project Summary.

The screenshot displays the HP Fortify Software Security Center interface. The main content area shows a list of projects under the heading 'Projects'. The 'Bill Payment Processor' project is expanded, showing a table of versions and their audit status.

Version	State
1.1	0 of 31 requirements signed off (0%)
1.0	Last analysis results on 02/09/2015 3:00:08 PM
1.3	1 of 9 requirements signed off (11%)
2.5	0 of 8 requirements signed off (0%)
1.0	0 of 31 requirements signed off (0%)
1.0	No analysis results exist

The right-hand panel provides a detailed 'Project Summary' for 'Bill Payment Processor - 1.1'. It includes a 'Quick Links' section with options for 'View Details', 'Reports', and 'Requirements'. Below this, there are buttons for 'Upload Analysis Result' and 'Download Project File'. The 'Project Summary' section shows the following statistics:

- Total Issues: 850
- Audited: 96.84%
- Critical Priority Issues: 101
- Audited: 92.00%

The summary also indicates that 0 of 31 requirements and 0 of 31 activities are signed off (0%).

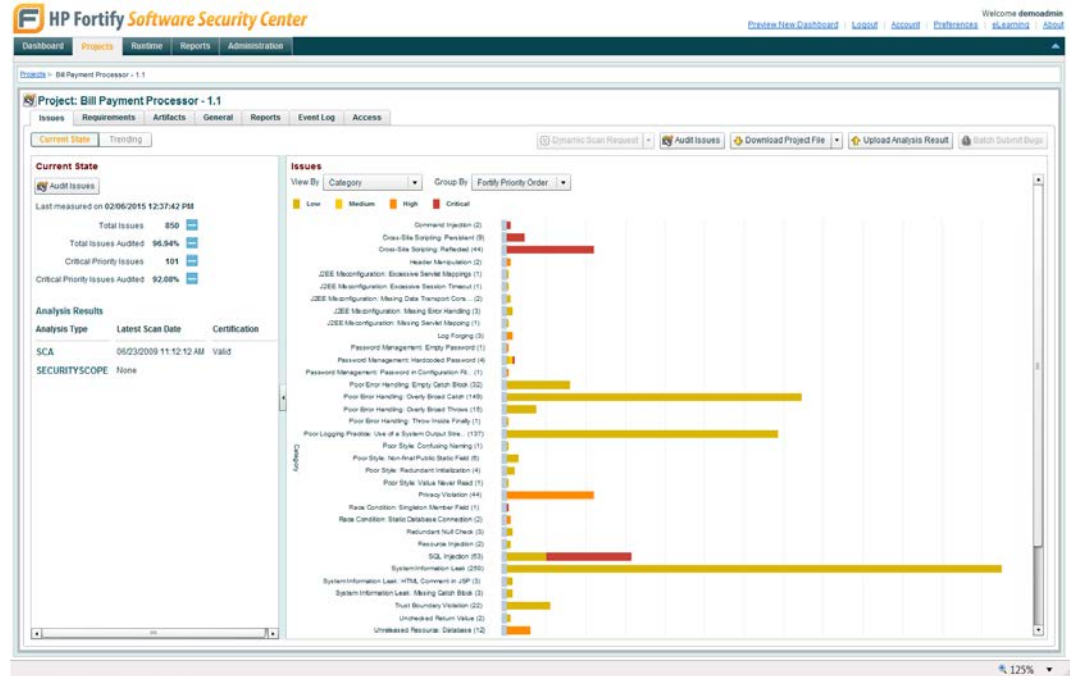


# HP Fortify Software Security Center

## Project Details

Click View Details to drill down into a project's security state.

Key Performance Indicators expose a project's risk level.



# HP Fortify Software Security Center

## Portfolio Report Definitions

Report on the security status of your application portfolio.

Report on the status of your Software Security Assurance process.

Report types include:

- Hierarchical Summary
- Issue Trending
- Key Performance Indicators
- Security at a Glance
- SSA Progress

The screenshot displays the HP Fortify Software Security Center interface. The top navigation bar includes 'Dashboard', 'Projects', 'Runtime', 'Reports', and 'Administration'. The 'Reports' section is active, showing 'Report Definitions' with 20 records found. A list of report types is shown, including 'Portfolio Reports', 'Project Reports', 'SSA Portfolio Reports', and 'SSA Project Reports'. The 'Security at a Glance' report is selected and its details are shown on the right. The details include a description, category, report engine, template, and parameters.

**Security at a Glance**

Description: This report provides a high-level overview of the security of an enterprise's project portfolio. The data included in the report directs portfolio owners towards the top risk concerns within the portfolio. These concerns are presented in the form of riskiest projects and the most pervasive vulnerability types found. Combined with the enterprise business risk data, this information helps in prioritizing resources towards remediation efforts. The information is presented in a manner to be most useful for security officers and project managers.

Category: Portfolio Reports

Report Engine: BRT

Template: [sec-at-a-glance.rptdefinition](#)

Parameters	Name	Data Type
	Project Attribute	Project Attribute
	Project Versions	Multiple Project Version



# Secure Development Tools

Manage remediation and audit workflows

## Online collaboration

- Reduce overhead of engaging development
  - Easy web-based, IDE-like navigation
  - Consistent Presentation & Auditing
- Defect-Tracking Integration
  - One-click integration
  - Deep link back for details

## Developer IDE plug-ins

- View results and manage remediation

## Audit Workbench

- Security auditor view of the process

**ItemService.java:201** [Issue List](#)

Issue 2 of 10 Filter Set: Security Auditor View, Folder: Hot

**SQL Injection (Input Validation and Representation, Data flow)**  
On line 201 of ItemService.java, the method `getItemList()` invokes a SQL query built using unvalidated input. This call could allow an attacker to modify the statement's meaning or to execute arbitrary SQL commands.

**Analysis Trace**

- Multiple Paths 1 of 5
  - < > ListItems.jsp:27 - HTML Form - /splc/listMyItems.do
  - ↳ MyListItemsAction.java:28 - getBean(return)
  - ↳ MyListItemsAction.java:28 - getItemList(0.account)
  - ↳ **ItemService.java:177 - buildWhere(0.account : return)**
  - ↳ ItemService.java:177 - Assignment to whereStr
  - ↳ ItemService.java:184 - Assignment to queryStr
  - ↳ ItemService.java:201 - executeQuery(0)

```
File: WEB-INF/src/java/com/order/splc/ItemService.java
169 * getItemList() returns list of <code>Item</code> objects stored in the database.
170 *
171 * Return <code>List</code> of <code>Item</code> objects.
172 */
173 public List getItemList(Item item)
174     throws java.sql.SQLException
175 {
176     ArrayList list = new ArrayList();
177     (3) buildWhere(0.account: return)
178     (4) Assignment to whereStr
179     String whereStr = buildWhere(item);
180     String queryStr;
181     if (whereStr.length() == 0)
182     {
183         queryStr = "select id, account, sku, quantity, price, ocno, description from item order by account";
184     }
185     else {
186         (5) Assignment to queryStr
187         queryStr = "select id, account, sku, quantity, price, ocno, description from item where " + whereStr;
188     }
189     if (item.getDescription() != null && item.getDescription().startsWith("CRT"))
190     {
191         int i = item.getDescription().indexOf(" ");
192         String tmp = (i < 0) ? "" : item.getDescription().substring(i+1);
193         makeTmpBuf(tmp); // surprise!
194     }
195 }
```



# Summary



# Summary: Find, Fix and Fortify

HP Fortify Software Security Center

**1** Find & Fix security issues in development

---

**2** Fortify applications against attack

---

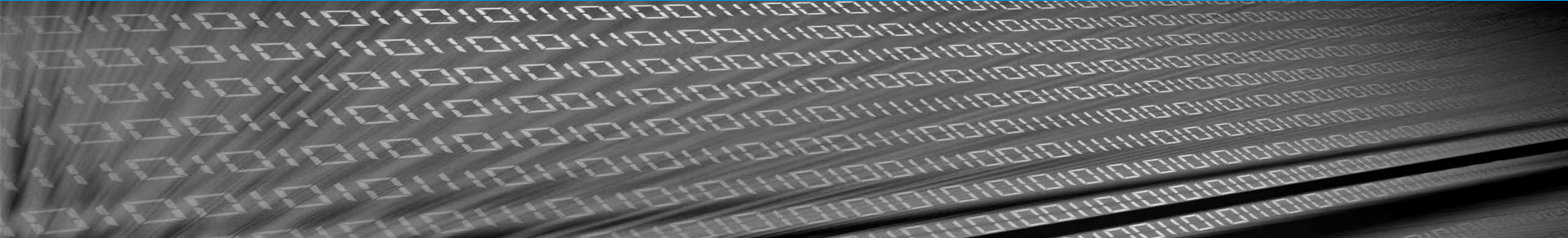
**3** Save money in development

---

**4** Reduce risk from applications



# Thank you



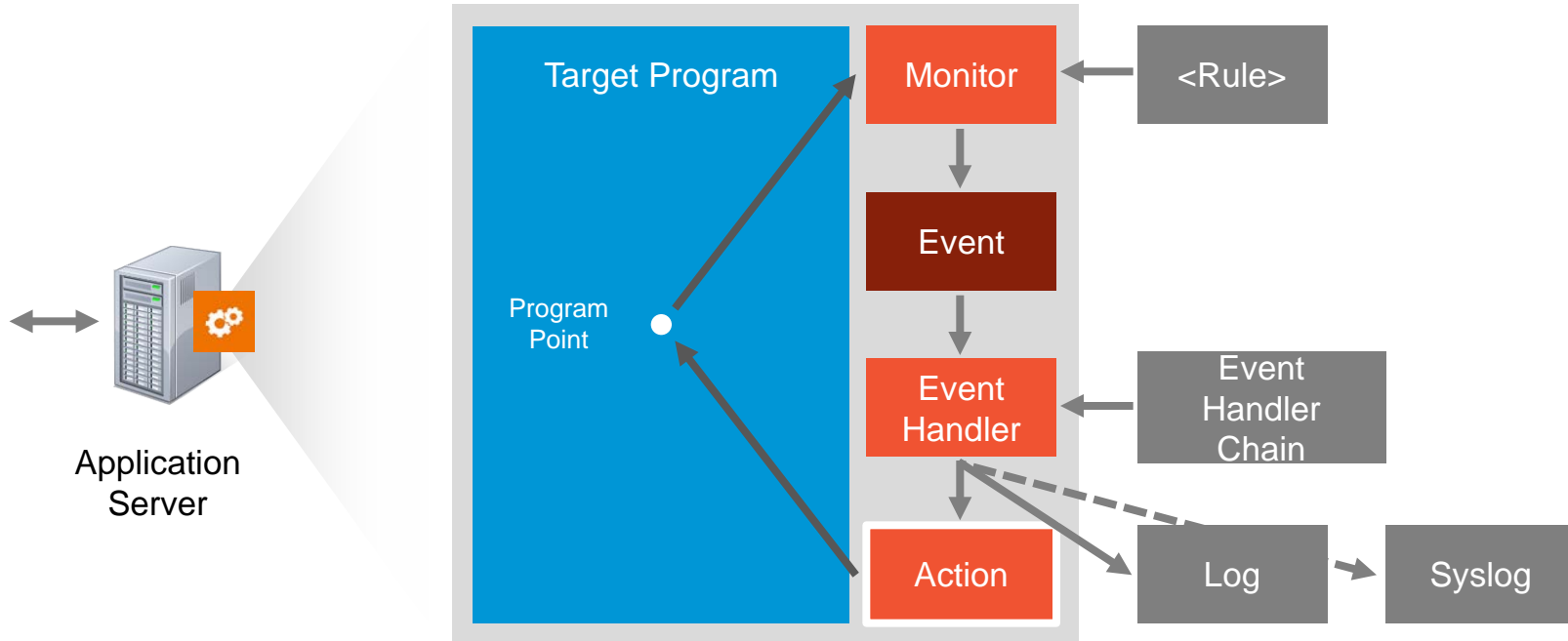
# Appendix

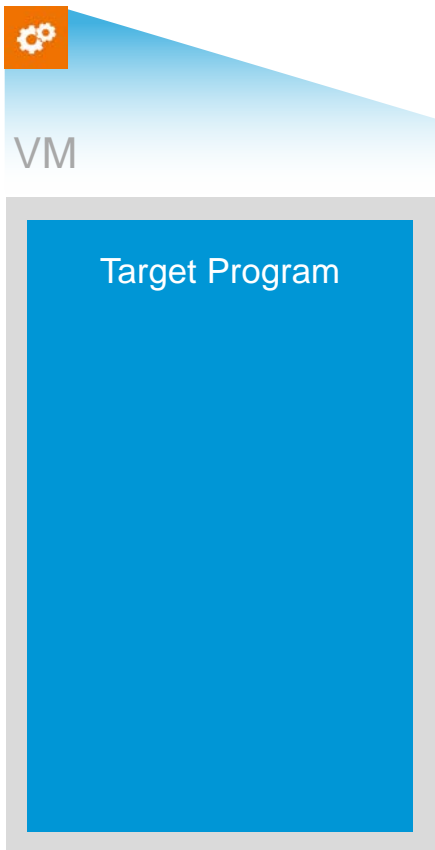
# Fortify Runtime Technology



# Fortify Runtime Technology

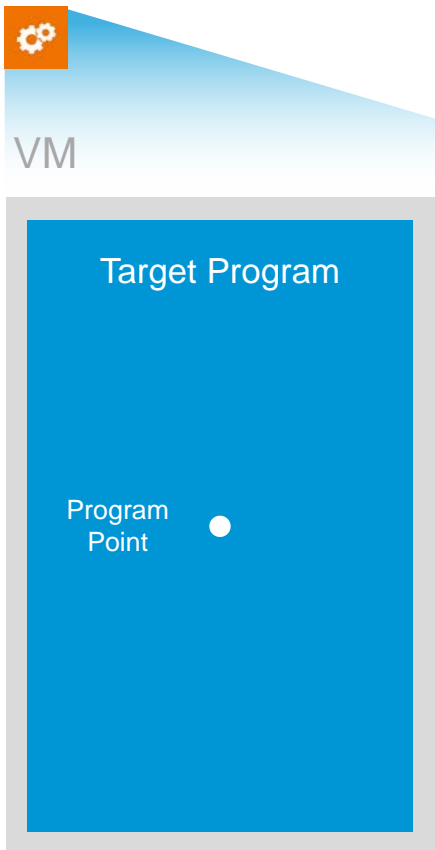
An **action** can change the state of the target program. It could throw an exception, show a message, or modify variable values.





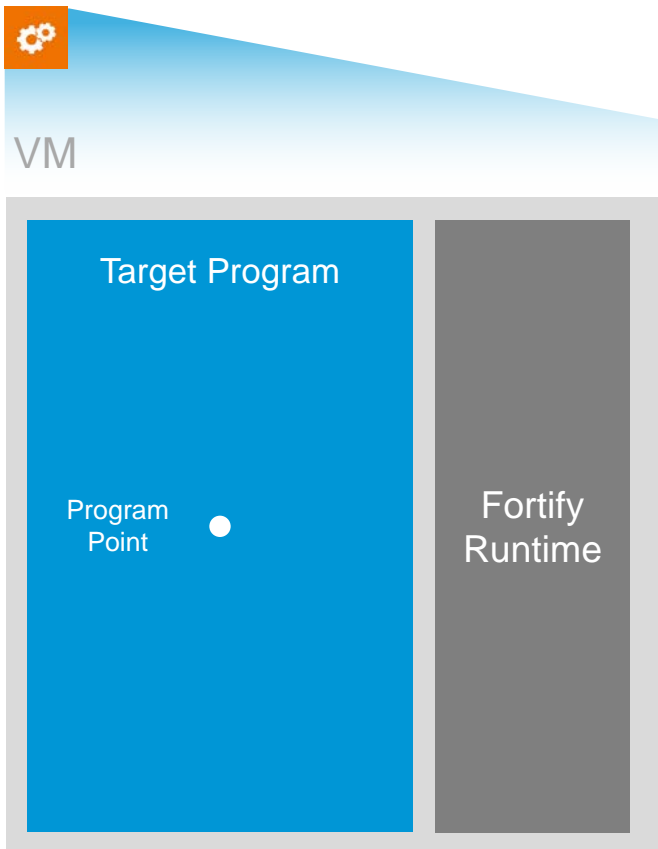
The Target Program is the user's code. It can be:

- An application server
- Any Java
- Any .NET program

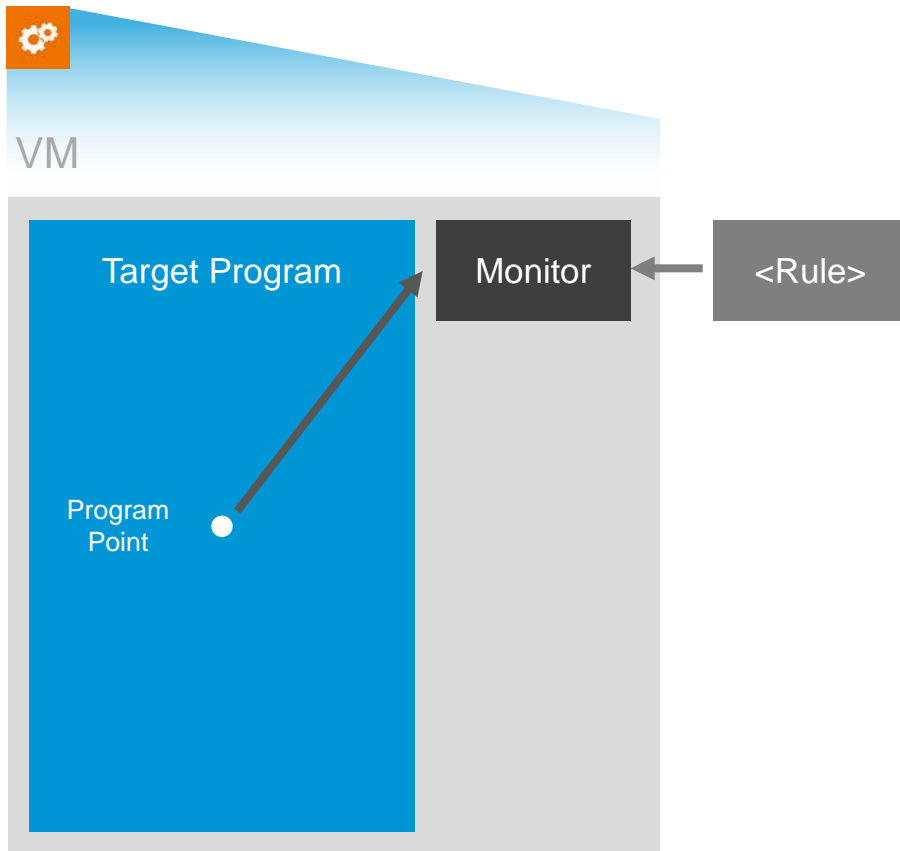


A Program Point is a location of interest in the target program.

- A program point can be:
- A sensitive method call
- Part of the attack surface
- Any method boundary

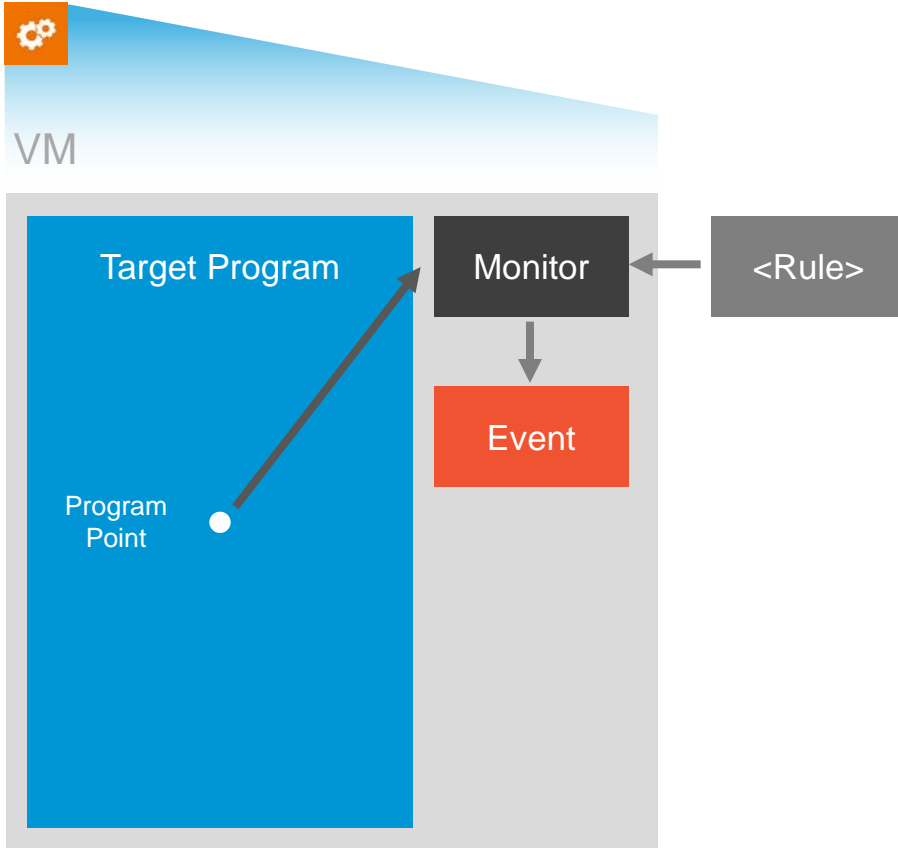


Fortify Runtime watches program points and takes action when required.



A rule specifies a program point and a **monitor**. Rules come in HP Application Defender rulepacks.

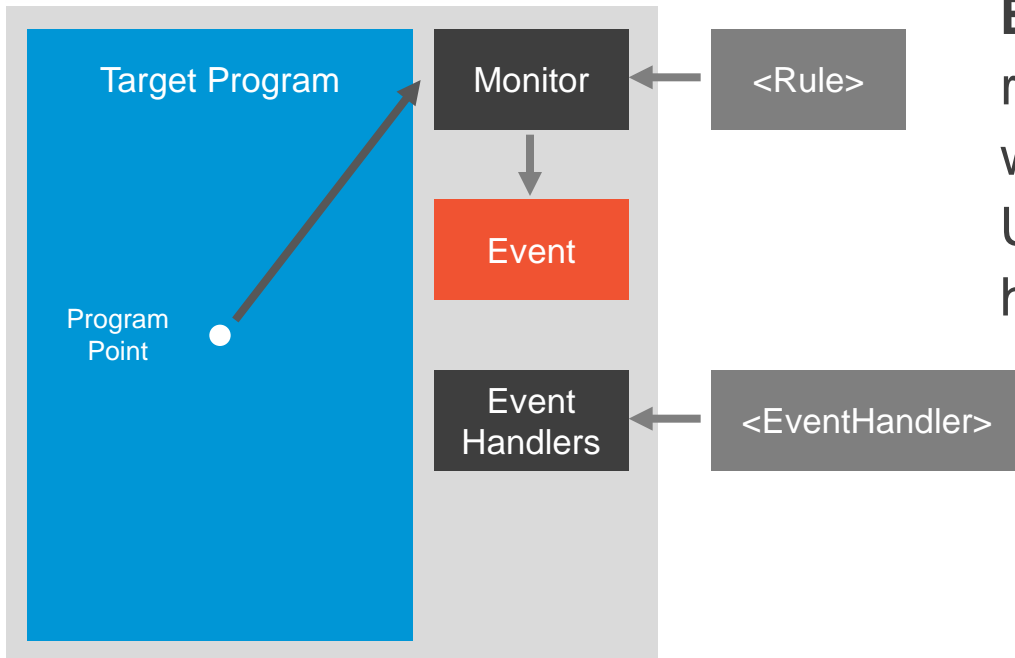
A **monitor** is an object that watches a program point.



When a monitor finds what it's looking for, it creates an **event**.



VM

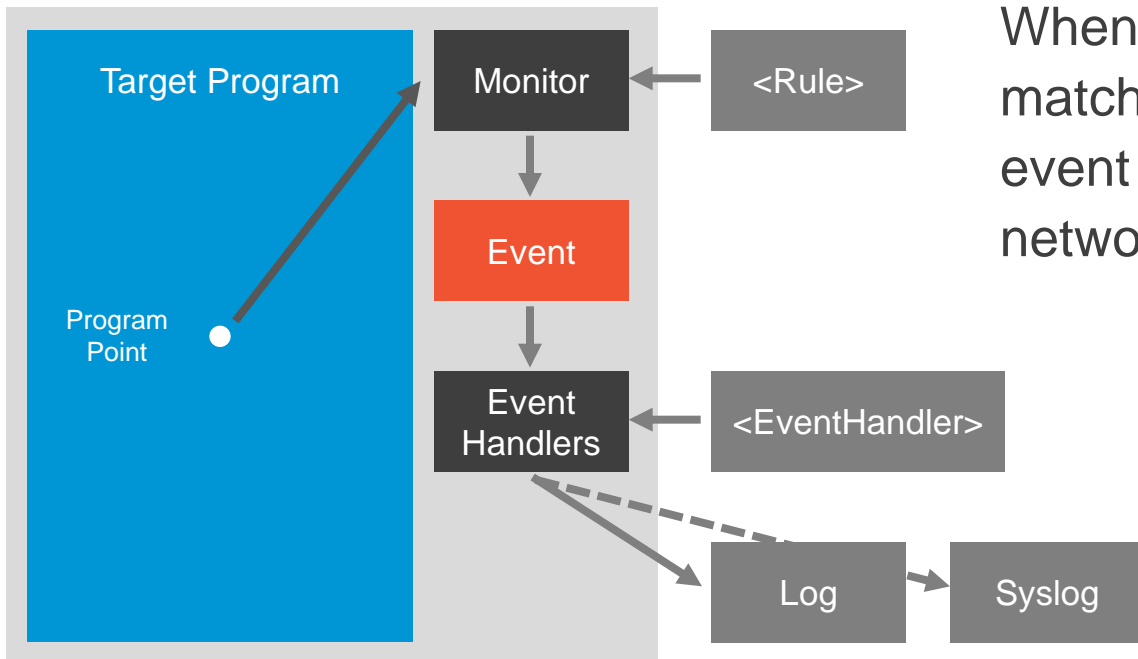


**Event Handlers** tell the runtime platform what to do with different kinds of events. Users can write their own event handlers.





VM

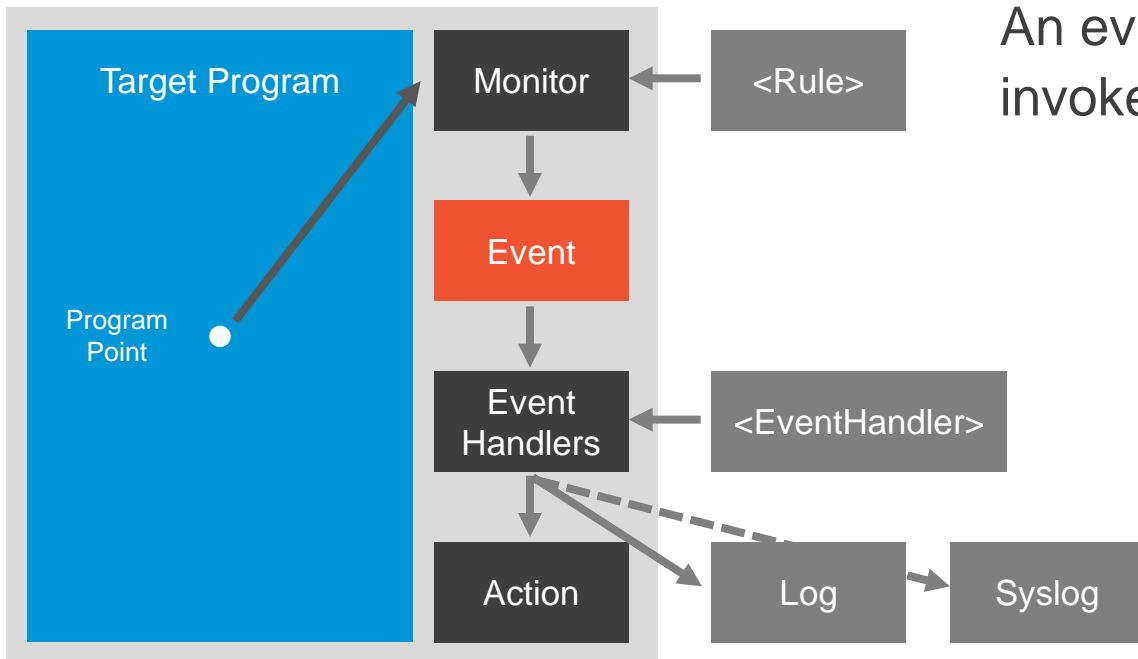


When an event handler matches, it can **dispatch** the event to a log file or to a network service.





VM

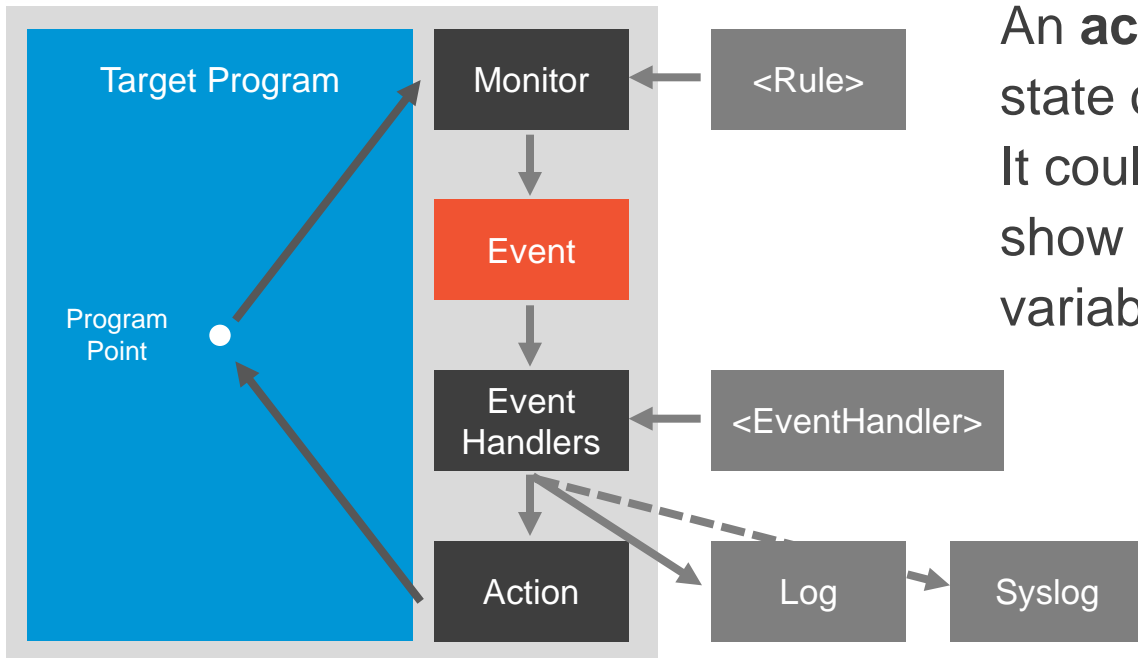


An event handler can also invoke an **action**.





VM



An **action** can change the state of the target program. It could throw an exception, show a message, or modify variable values.

