Prep Work Exercises

- Open Your VM c:\VM Images\2017\windows 10 x64 (2).vmx
  - **UID:** Admin
  - **PWD:** P@ssword1

- Check Access to http://6.94.185.35.bc.googleusercontent.com:8080/ssc/#/
  - **UID**: Admin
  - **PWD**: Fortify@01

- Check Access to http://zero.webappsecurity.com/
  - **UID**: username
  - **PWD**: password

MICRO FOCUS

Government Solutions

Application Security

Fortify-On-Demand | Application Defender | Fortify | WebInspect

Haleh Nematollahy
Senior Security Solutions Architect

Jeffrey Hsiao
Security Solutions Architect
Jeffrey.Hsiao@microfocusgov.com

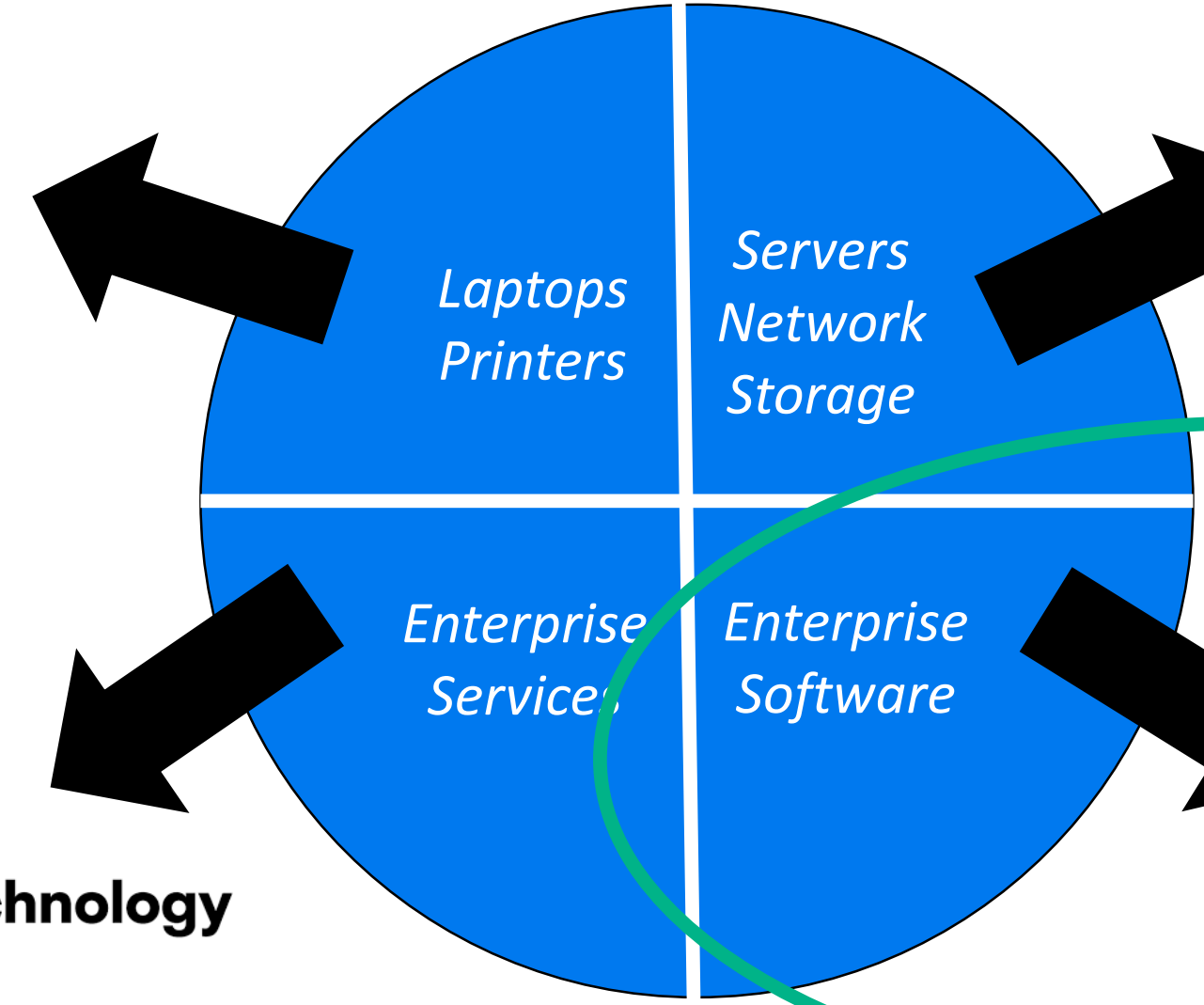
Agenda

- Prep Work Exercises
- Introductions
- Application Security Challenges
- HPE Fortify Solution
- HPE WebInspect Overview and Exercises
- Lunch
- HPE Softrware Security Center Overview and Exercises
- HPE WebInspect Enterprise Overview
- Wrap-Up

Introductions

- Name and organization
- Role and duties
- Secure coding background

HPE Software is now Micro Focus!



Hewlett Packard Enterprise

DXC.technology

MICRO FOCUS
Government Solutions

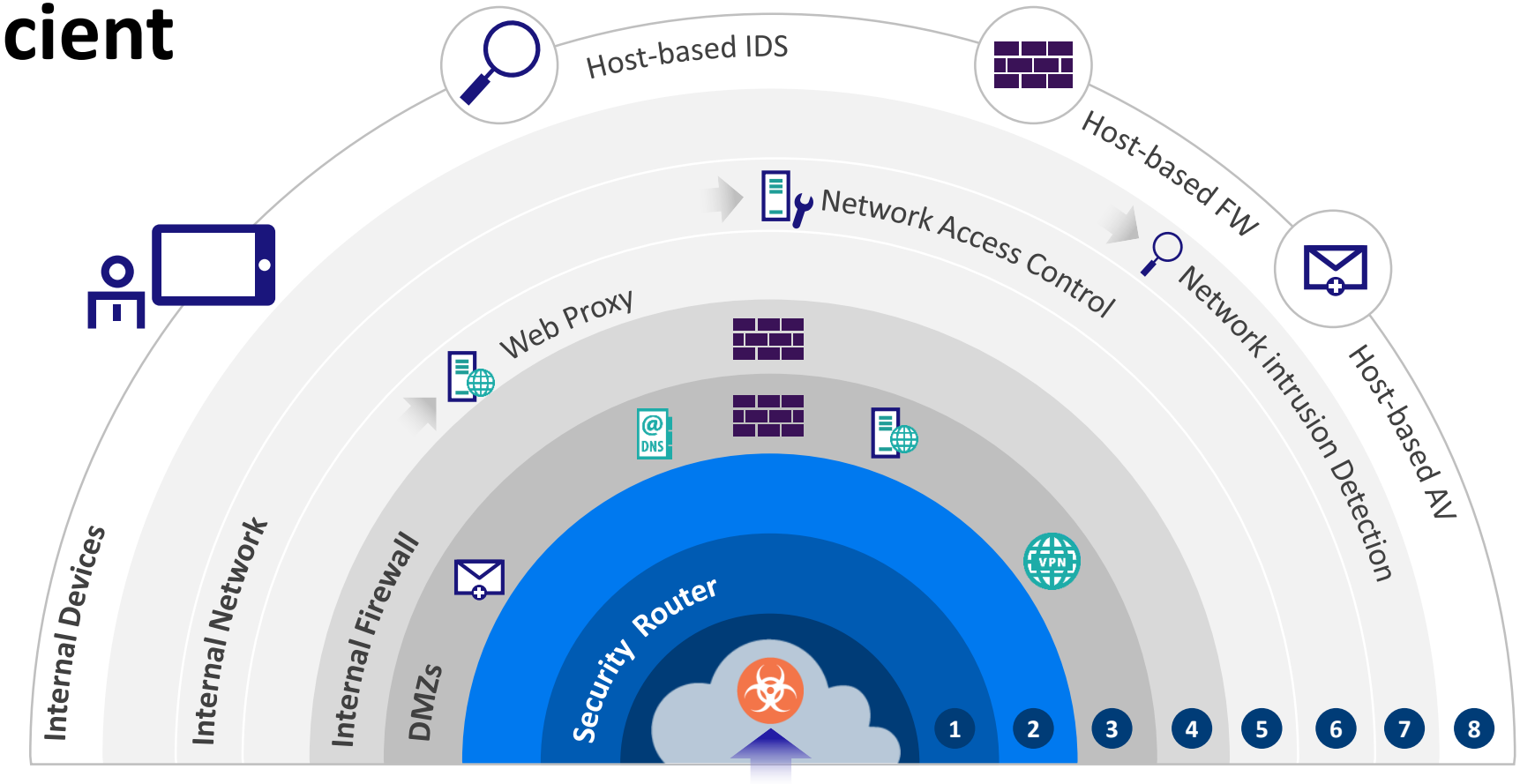
MICRO FOCUS
Government Solutions

MICRO FOCUS

Government Solutions

The Software Security Problem

Existing network and perimeter based security is insufficient



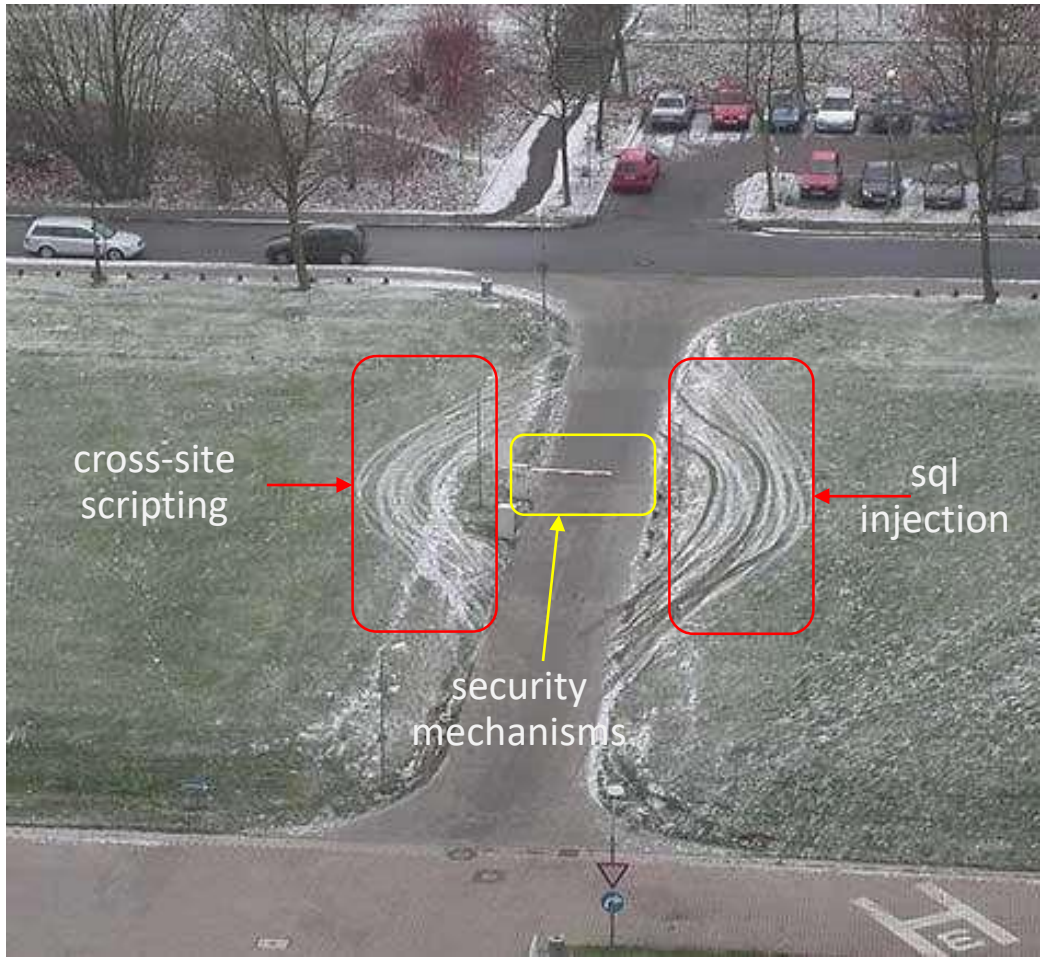
84% of breaches exploit vulnerabilities in the application layer

Yet the ratio of spending between perimeter security and application security is **23-to-1**

- Gartner Maverick Research: Stop Protecting Your Apps; It's Time for Apps to Protect Themselves (2014)

Vulnerability Defined

Is this a vulnerability?



Vulnerabilities in Software

What is a software or application vulnerability?

A **vulnerability** is a hole or a weakness in the application, which can be a design flaw or an implementation bug, that allows an attacker to cause harm to the stakeholders of an application.

MICRO FOCUS

Government Solutions

So How Bad Can It Be?

Vulnerabilities in Software

OWASP

- The **O**pen **W**eb **A**pplication **S**ecurity **P**roject is a worldwide free and open community focused on improving the security of application software.

  www.owasp.org

- This community routinely publishes a list of the top-10 application security vulnerabilities.
  - New list published in 2017.
  - Previous list was published in 2013.

OWASP Top 10

As of 2017, OWASP lists the following top-10 categories:

1) Injection
2) Broken Authentication
3) Sensitive Data Exposure
4) XML External Entities (XXE)
5) Broken Access Control
6) Security Misconfiguration
7) Cross-Site Scripting (XSS)
8) Insecure Deserialization
9) Using Components with Known Vulnerabilities
10) Insufficient Logging & Monitoring

OWASP A1 – Injection Flaws

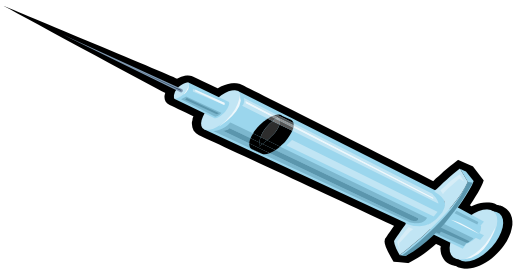
- Multiple Types of Vulnerabilities
  - SQL Injection
  - LDAP Injection
  - XPath Injection
  - XSLT Injection
  - HTML Injection
  - OS Command Injection
- Basic Definition: Interpreters execute unintended commands on a server

OWASP A1 – Injection Flaws

- SQL Injection – The Worst of the Worst
  - Has been on the top of the OWASP Top 10 since the beginning
  - The favorite vulnerability of Anonymous, LulzSec, and Black Hats
  - Pretty easy to detect and exploit
  - Easy access to your data
- What is it?
  - Data from an untrusted source is mixed into a SQL query allowing an attacker to change the query.

Injection Attack

General Algorithm



1. Generate dynamic string to be used by an interpreter
   - Append raw data to string
   - Raw data is unexpected
2. Pass string to interpreter to be executed
3. Interpreter performs some other operation as a result of unexpected data

SQL Injection Attack

Conceptual Example and Flow

1) Perform a dynamic query against a SQL database such as:
    1) Select * from USERS where name = ‘+userName+’

2) User sets userName = x'; drop table members; --

3) SQL database query has now changed meaning:
    1) Select * from USERS where name = ‘x'; drop table members; --’

4) Database will now delete the *members* table instead of querying user table

Preventing SQL Injections

Another Exploit - incorrect filtered escape character

1) SQL Statement intended to retrieve the user's account transactions:

```
String query = "SELECT * FROM acct_trans WHERE acct_num = '" +
  request.getParameter("AcctNum") + "'";

…
```

2) Exploit: `AcctNum = '12345' or '1' = '1' --`

3) Result:

```
SELECT * FROM acct_trans
  WHERE acct_num = '12345' or '1' = '1' --
```

How Do I Fix SQL Injection?

- Input Validation – Yes
  - Detect unauthorized input before processed by application
- Parameterized Queries (prepared stmt) – Even Better
  - Define SQL code and then pass in each parameter to the query later
  - Parameters are placeholders for values

Parameterized Query

```
String selectStatement = "SELECT * FROM User WHERE userId = ? ";
PreparedStatement prepStmt = con.prepareStatement(selectStatement);
prepStmt.setString(1, userId);
ResultSet rs = prepStmt.executeQuery();
```

NOT

```
String strUserName = request.getParameter("Txt_UserName");
PreparedStatement prepStmt = con.prepareStatement("SELECT * FROM user WHERE userId = '+strUserName+'");
```

HETZNER

November 2, 2017
15:40 GMT

Hetzner hack: Top South African web host hit with mega-breach, every client may be exposed

Firm provides internet hosting services to more than 40,000 customers.

By Jason Murdock
November 2, 2017 15:40 GMT

(Hetzner)

Hackers have compromised a major database maintained by Hetzner Ltd, one of the largest data centre and web hosting services in South Africa.

The Johannesburg-based company said on Wednesday (1 November) that a key client portal called "konsoleH" had been accessed by unknown cybercriminals. "We should assume that all our customer data has been exposed," it said in a lengthy statement 24 hours later.

The company claimed that hackers exploited an SQL injection vulnerability its database.

"There is no way for us to ascertain how the exposed data will be used," it warned Thursday.

The company claimed that hackers exploited an SQLinjection vulnerability its database.

Static Analysis – SQL Injection

```
117     {
118         Connection conn=null;
119         Statement statement = null;
120         ArrayList locations = new ArrayList();
121 
122         try{
123             conn = ConnFactory.getInstance().getConnection();
124 
125             String queryStr = "SELECT * FROM location WHERE branch = 'Yes' AND state = '" + state + "' AND city = '" + city
126             statement = conn.createStatement();
127             ResultSet rs = statement.executeQuery(queryStr);
128             while (rs.next())
129             {
130                 locations.add(new Location(rs.getString("address"), rs.getString("city"), rs.getString("state"), rs.getStrin
131             }
132         }
133         finally{
134             safeCloseStatement(statement);
135             safeCloseConnection(conn);
136         }
137 
138         return locations;
139     }
140 
141     public static List findBranchByAddress(String address, String city, String state) throws Exception
142     {
143         List branches=null;
144         Session session=null;
145 
```

Dynamic Analysis – SQL Injection

```
GET /riches/ShowLocations.action?errorMsg=Invalid%2bcredentials%2bfor%2bnull&zip=30346%27%09OR&location
Referer: http://riches:8080/riches/login/Login.action?errorMsg=Invalid%20credentials%20for%20null
Accept: */*
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)
Host: riches:8080
Connection: Keep-Alive
X-WIPP: AscVersion=10.0.483.0; Capture=sql_command, vulnerability
X-Scan-Memo: Category="Audit.Attack"; SID="27EBF4CBC77C106EA710257781E6A09D"; PSID="C92BDA7C1E1F3271E51
AttackType="QueryPar
; AttackParamDesc="z
```

```
HTTP/1.1 500 Internal Server Error
Server: Apache-Coyote/1.1
X-WIPP-Update: Application
X-WIPP-RequestID: cd427fec-8623-45b1-aa45-10f6c81d266e
X-WIPP-Version: java / 1.3 / ren_4045
Content-Type: text/html;charset=utf-8
Date: Mon, 25 Mar 2013 16:26:57 GMT
Connection: close
Content-Length: 11502

<html><head><title>Apache Tomcat/6.0.35 - Error report</title><style><!--H1 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-
;} H2 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:16px;} H3 {font-family:Tahoma,Arial,sans-serif;color:white;backgr
-color:#525D76;font-size:14px;} BODY {font-family:Tahoma,Arial,sans-serif;color:black;background-color:white;} B {font-family:Tahoma,Arial,sans-serif;color
background-color:#525D76;} P {font-family:Tahoma,Arial,sans-serif;background:white;color:black;font-size:12px;}A {color : black;}A.name {color : black;}HR
#525D76;}--></style> </head><body><h1>HTTP Status 500 - </h1><HR size="1" noshade="noshade"><p><b>type</b> Exception report</p><p><b>message</b> <u></u></p
description</b> <u>The server encountered an internal error () that prevented it from fulfilling this request.</u></p><p><b>exception</b> <pre>javax.servle
ServletException: java.sql.SQLException: Unexpected end of command in statement [SELECT * FROM location WHERE zip = '30346' OR']
    org.apache.struts2.dispatcher.Dispatcher.serviceAction(Dispatcher.java:515)
    org.apache.struts2.dispatcher.FilterDispatcher.doFilter(FilterDispatcher.java:419)
</pre></p><p><b>root cause</b> <pre>java.sql.SQLException: Unexpected end of command in statement [SELECT * FROM location WHERE zip = '30346'   OR']
    org.hsqldb.jdbc.Util.throwError(Unknown Source)
    org.hsqldb.jdbc.jdbcPreparedStatement.&lt;init&gt;(Unknown Source)
    org.hsqldb.jdbc.jdbcConnection.prepareStatement(Unknown Source)
    org.apache.tomcat.dbcp.dbcp.DelegatingConnection.prepareStatement(DelegatingConnection.java:281)
```

Hybrid Analysis – SQL Injection

SQL Injection (confirmed)

This stack trace is from the running application and was returned by SecurityScope. It can be used to determine root cause.

SecurityScope Trigger:

SELECT * FROM location WHERE zip = '30346"\tOR'

SecurityScope Stack Trace:

```
   at org.apache.tomcat.dbcp.dbcp.PoolingDataSource$PoolGuardConnectionWrapper.prepareStatement(PoolingDataSource.java:312)
----- start of user application code ----------------------------------------
   at com.fortify.samples.riches.model.LocationService.findByZip(LocationService.java:110)
   at com.fortify.samples.riches.FindLocations.execute(FindLocations.java:50)
   at sun.reflect.GeneratedMethodAccessor69.invoke(Unknown Source)
   at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
   at java.lang.reflect.Method.invoke(Method.java:616)
   at com.opensymphony.xwork2.DefaultActionInvocation.invokeAction(DefaultActionInvocation.java:404)
   at com.opensymphony.xwork2.DefaultActionInvocation.invokeActionOnly(DefaultActionInvocation.java:267)
   at com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:229)
   at com.opensymphony.xwork2.interceptor.DefaultWorkflowInterceptor.doIntercept(DefaultWorkflowInterceptor.java:221)
   at com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:86)
   at com.opensymphony.xwork2.DefaultActionInvocation$2.doProfiling(DefaultActionInvocation.java:224)
   at com.opensymphony.xwork2.DefaultActionInvocation$2.doProfiling(DefaultActionInvocation.java:223)
   at com.opensymphony.xwork2.util.profiling.UtilTimerStack.profile(UtilTimerStack.java:455)
   at com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:221)
   at com.opensymphony.xwork2.validator.ValidationInterceptor.doIntercept(ValidationInterceptor.java:150)
   at org.apache.struts2.interceptor.validation.AnnotationValidationInterceptor.doIntercept(AnnotationValidationInterceptor.java:48)
   at com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:86)
   at com.opensymphony.xwork2.DefaultActionInvocation$2.doProfiling(DefaultActionInvocation.java:224)
```

Exercise 1: Start the Fortify Demo

Environment Setup

- Start the Fortify Demo Server
- There’s a “Launch the Riches Demo App” Shortcut on your desktop

- **It should already be started. You Should see some Command Prompt Windows.

Demo: SQL Injection

- Open Internet Explorer and browse to http://localhost:8080/riches (there should also be a shortcut)
- Click the Locations Button at the top.
- There is SQL Injection in this form. See if you can find it!
- Valid Zip Codes (94404, 10005, 94123)

Demo: SQL Injection

- Open Internet Explorer and browse to http://localhost:8080/riches (there should also be a shortcut)
- Click the Locations Button at the top.
- There is SQL Injection in this form. See if you can find it!
- Valid Zip Codes (94404, 10005, 94123)

Try entering

' or '1'='1

in the **Find ATMs/Locations** field

OWASP A7 – Cross Site Scripting (XSS)

- The Most Prevalent Vulnerability on the OWASP Top 10
  - Was at the top until the ranking methodology changed
  - Very easy to detect and exploit
- What is it?
  - Your application is used as a platform to attack your users.
  - Allows an attacker to insert malicious code into the user's session
  - Used to deface web sites, hijack sessions, steal credentials, and install malware

OWASP A7 – Cross Site Scripting (XSS)

- Reflected XSS
  - Requires a user to execute an action that contains the attack payload. (Such as clicking a link in a phishing email)
  - The attack only affects the user that executes the action
- Persistent XSS
  - Attack payload is injected into a data store, such as a database.
  - The attack affects every user that uses the application.
  - The most impactful variant of XSS

Preventing XSS – Input/Output Validation

- Blacklisting – Developing a naughty list of characters/tags
  - Nearly impossible to write black lists that cover all attack vectors
  - Many ways to obfuscate attack payloads
    - Using case, null characters, and flaws in browser rendering
    - Using alternate tags, such as IMG, IFRAME, links, body, etc
    - Using alternate encodings and languages
    - Check out: https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
- Whitelisting – Using regular expressions
  - [A-Za-z0-9]{5,25} – Possible regex for a username

Preventing XSS – Output Encoding

- Encoding – Making Malicious Code Unexecutable
  - <script> becomes <script>
  - HTML Encoding for all data rendered in plain HTML
  - Special care should be taken for data inserted into JavaScript and as tag attributes
- Many Standard Encoding Libraries are not sufficient
  - Use the AntiXSS Library from Microsoft for .NET (now included in 4.5)
  - OWASP Enterprise Security API (ESAPI)

Cross Site Scripting Famous Example

PayPal, circa 2004 - 2006

- Steal credit card numbers
    1. Users access URL on genuine PayPal site
    2. Page modified via XSS attack to silently redirect user to external server
    3. Fake PayPal Member log-in page
    4. User supplies login credentials to fake site
- Exploitable for two years

Exercise 2: XSS Injection

- Click the submit button on the login form.
- Open Internet Explorer and browse to http://localhost:8080/riches (there should also be a shortcut)
- There is Cross Site Scripting in the login page. See it?
- Valid Login (eddie/eddie)

Exercise 2: XSS Injection

- Click the submit button on the login form.
- Open Internet Explorer and browse to http://localhost:8080/riches (there should also be a shortcut)
- There is Cross Site Scripting in the login page. See it?
- Valid Login (eddie/eddie)

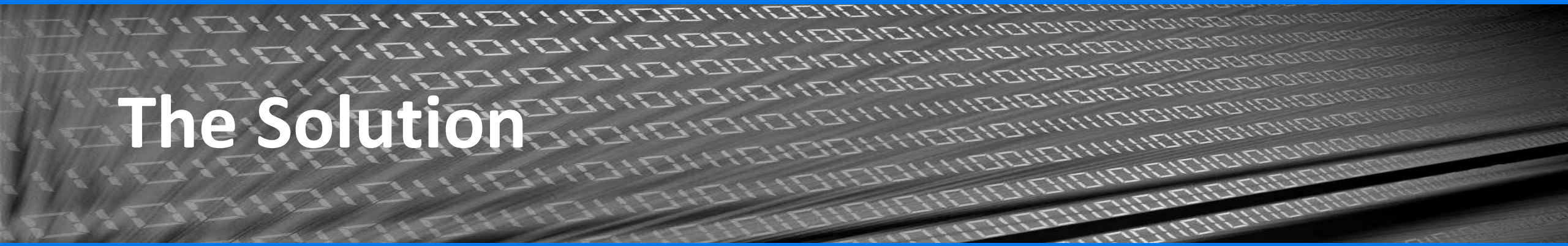
Try entering

';</script><script>alert('XSS');</script>

In the login field

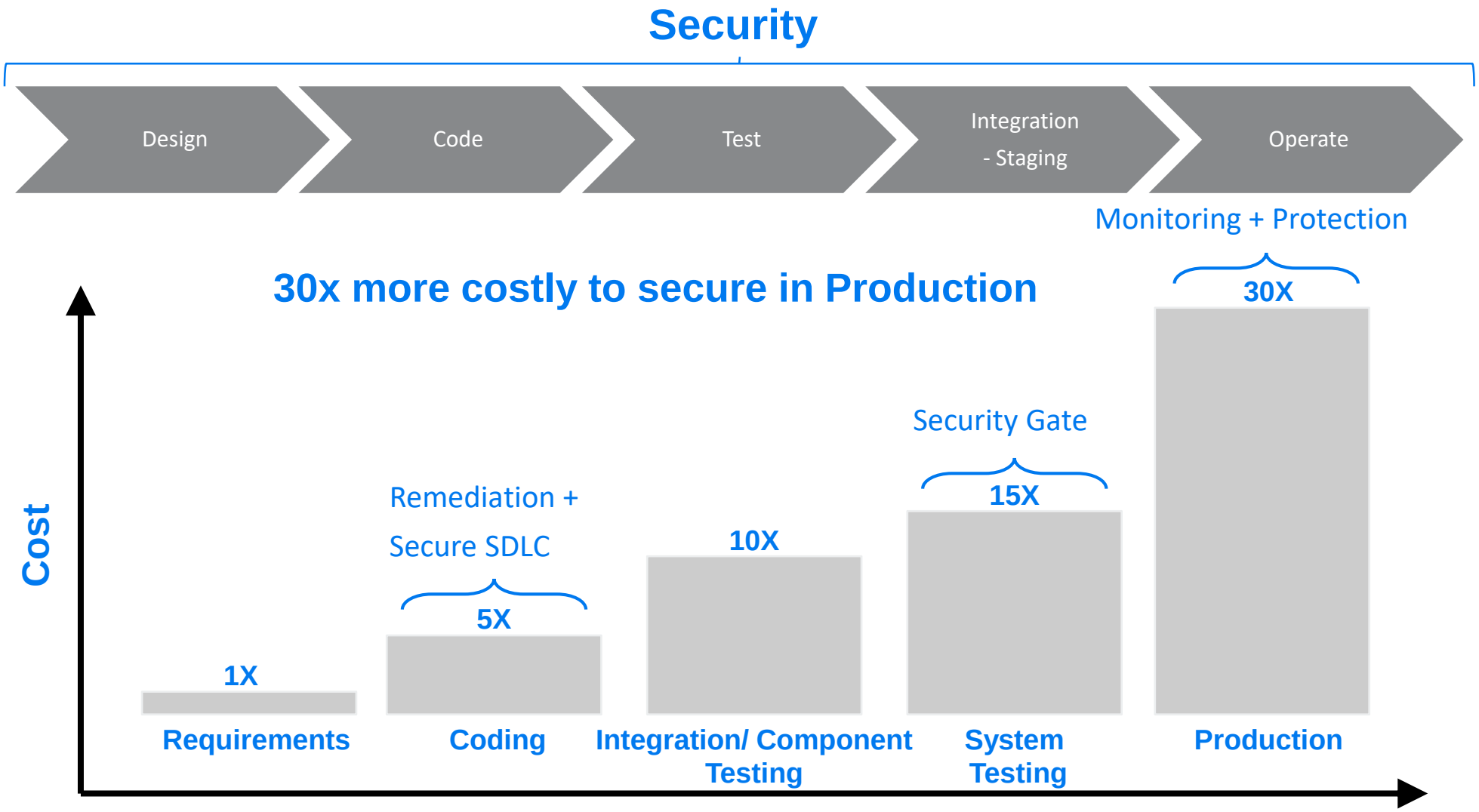
MICRO FOCUS

Government Solutions



The Solution

Software Security Assurance (SSA & SDLC)



Source: NIST

Software Security Assurance

Return On Investment

Does Application Security Pay?
Measuring the Business Impact of Software Security Assurance Solutions

Findings:

- Significant Cost Savings: The combination of test and remediation cost savings and development productivity improvements are generating benefits estimated at **$8M** per year.
- Expanded Revenue Potential: Companies in some industries can capture an estimated **$8M** in additional revenue and save **$15M** in development costs.
- Greater Overall Economic Value Potential: Mainstay calculated that software security programs can generate as much as **$50M** in annual benefits.

http://h30528.www3.hp.com/Security/Fortify_Mainstay_ROI_Study.pdf

Software Security Assurance

Return On Investment

Does Application Security Pay?

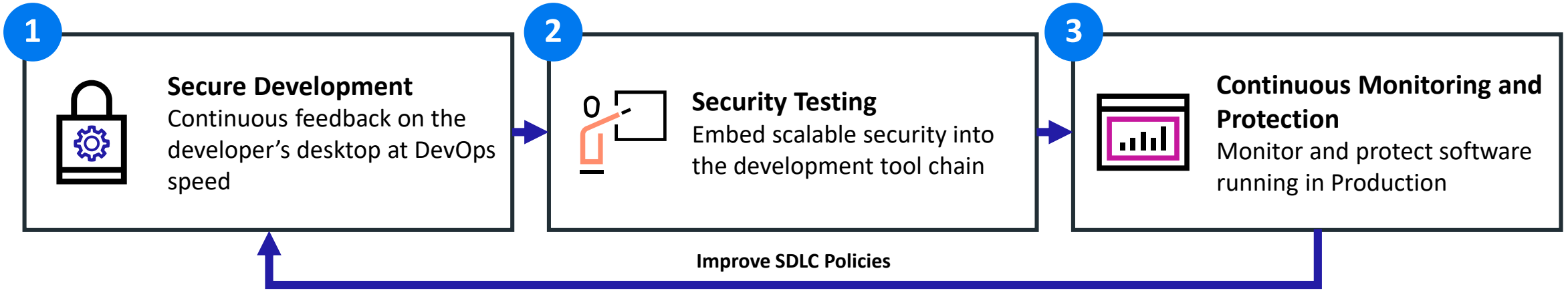
Measuring the Business Impact of Software Security Assurance Solutions

After implementing an SSA program…

Performance Metric	Improvement
Vulnerabilities per application	From 100s to 10s
Average time to fix a vulnerability	From 1 to 2 weeks to 1 to 2 hours
Percentage of repeat vulnerabilities	From 80% to 0%
Compliance and penetration testing effort	From ~$500k to ~$250k
Time-to-market delays due to vulnerabilities	From 4+ incidents (30 days each) per year to none

http://h30528.www3.hp.com/Security/Fortify_Mainstay_ROI_Study.pdf

The right approach for the new SDLC – Build it in



This is application security for the new SDLC

Software Security Analysis and Defense

- **Static Analysis (Fortify)**: analysis of computer software that is performed without actually executing programs (i.e. run on source code)
- **Dynamic Analysis (WebInspect)**: analysis of computer software that is performed by executing programs on a real or virtual processor (Pentesting against a running web application)
- **RASP (Application Defender):** software agent protects and monitors production systems.

Static	Dynamic	RASP
Requires source code	Only Web Applications and services	Only Web Applications and services
Finds most issues	Finds fewer issues	Protects and monitors applications
Provides line of code level detail	Provides request and response	Requires a software agent to be installed
Preferred by developers	Preferred by security and SOC	Can protects against zero days

SAST -vs- **DAST**

WebInspect

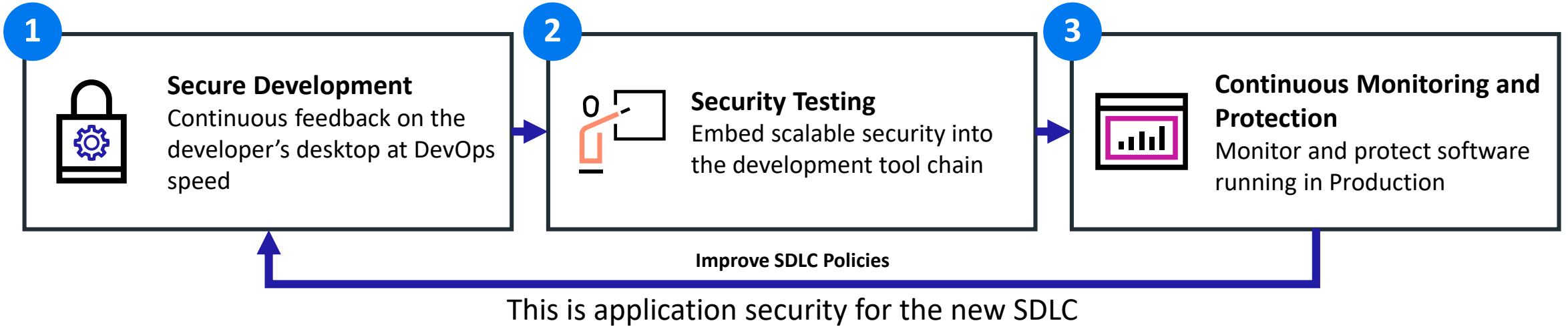
Outside In Testing

Dynamic **A**pplication **S**ecurity **T**esting

- Mimics actual human testing – open a browser, input value, look at results
- Works completely remotely against the interface of an application
- Requires no source code!
- Can be used on COTS applications.
- Limited visibility – can only find vulnerabilities discoverable via the interface
- High credibly results – since it runs remotely, everything it finds is remotely exploitable

The right approach for the new SDLC – Build it in

Fortify SCA SAST	WI Agent IAST	WebInspect DAST	App Defender RASP



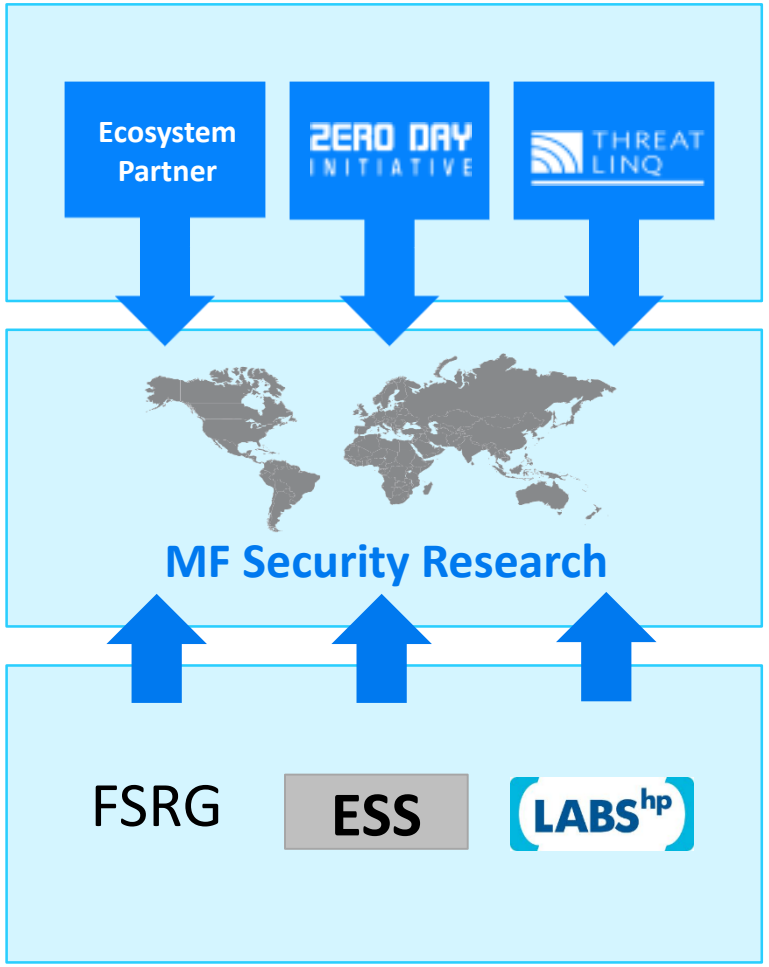
This is application security for the new SDLC

MICRO FOCUS

Government Solutions

MF Fortify Solutions

Security solutions backed by MF Security Research

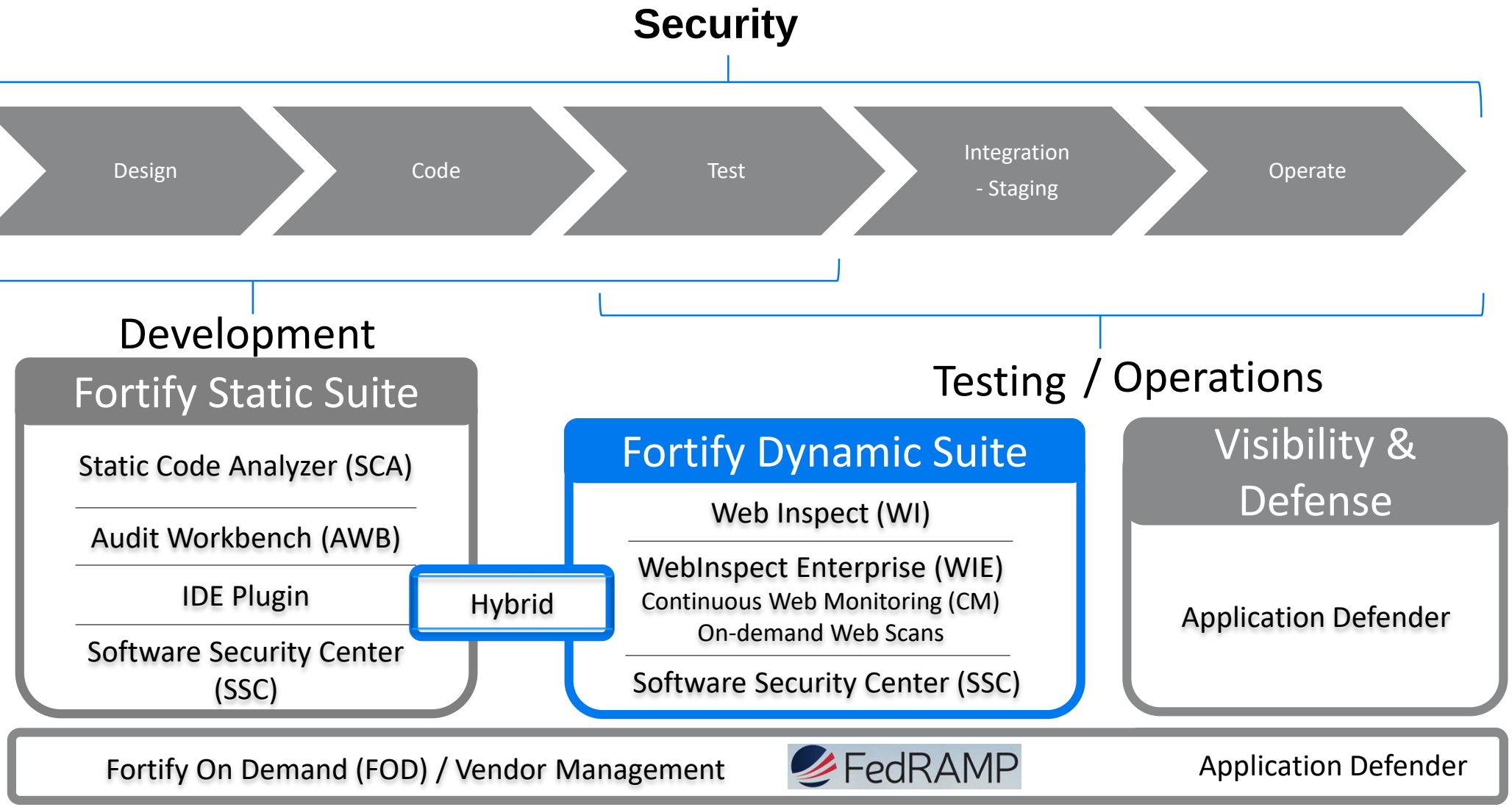


- SANS, CERT, NIST, OSVDB, software & reputation vendors
- 3000+ Researchers
- 2000+ Customers sharing data

- Largest commercial IT security research group
- Continuously finds more vulnerabilities than the rest of the market combined (50-75% of publicly reported critical vulnerabilities)
- Frost & Sullivan winner for vulnerability research last four years

- Collaborative effort of market leading teams: ArcSight, Fortify, MF Labs, Application Security Center
- Collect network and security data from around the globe

Software Security Assurance (SSA & SDLC)



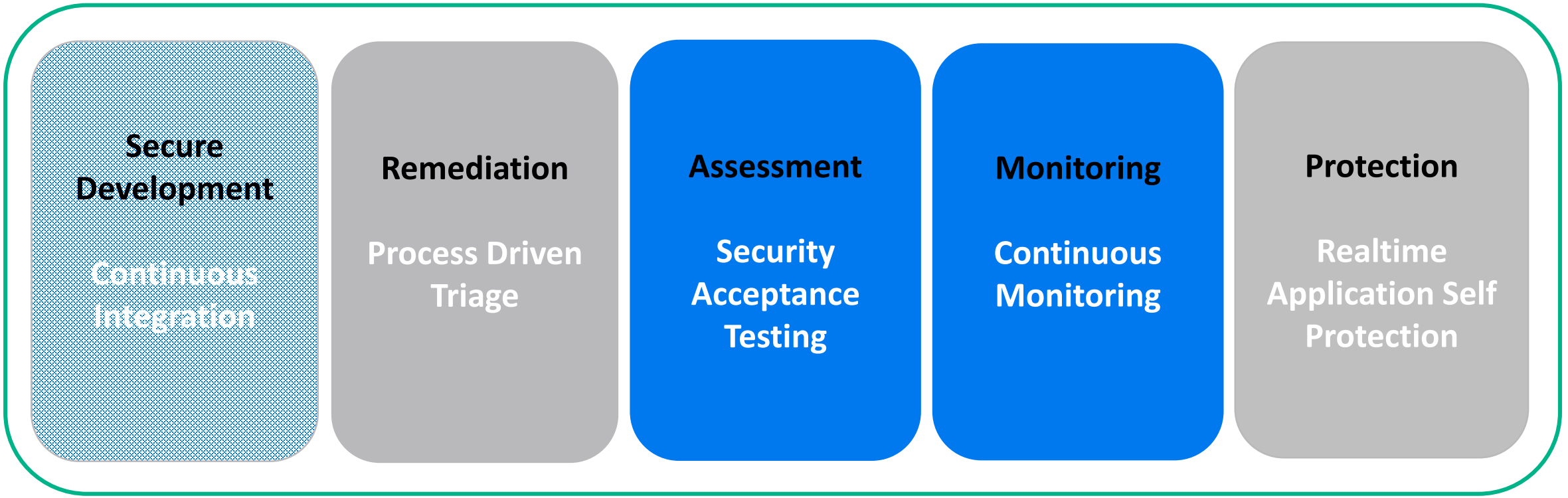
MICRO FOCUS
Government Solutions

MICRO FOCUS

Government Solutions

HPE WebInspect Overview and Exercises

Software Security Assurance – Top 2 for WebInspect

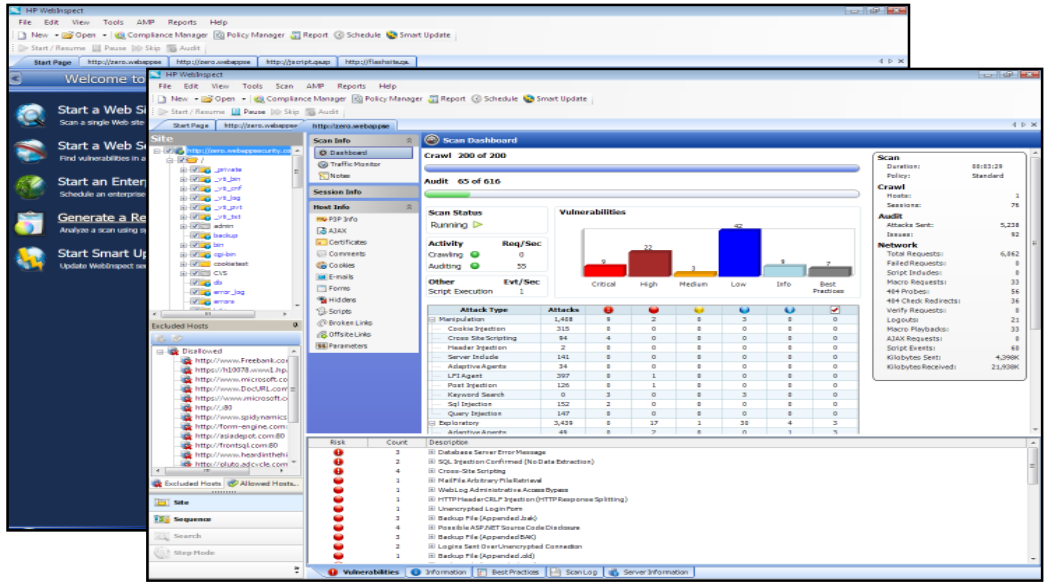


HPE WebInspect

Dynamics analysis – find critical security issues in running applications

- Features:

- Quickly identify risk in existing applications
- Automate dynamic application security testing of any technology, from development through production
- Validate vulnerabilities in running applications, prioritizing the most critical issues for root-cause analysis



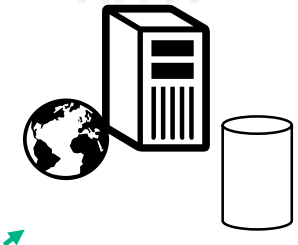
MICRO FOCUS
Government Solutions

WI/WIE Conceptual Architecture - Types

Flexible modeling for Web App Continuous Monitoring and Centralized Vulnerability Management

WebInspect

WIE



Enterprise Manager

On-Demand

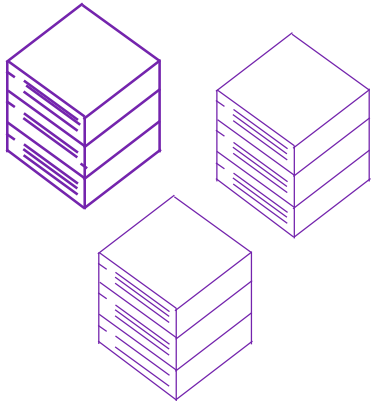
Continuous Monitoring

Customers get to kick off and manage their own scans without relying on security availability

LIM Server

Testers share licenses via check-in/check-out

Sensors



Software Security Center

Custom Reporting

Management gets enterprise metrics, dashboards and reporting

MICRO FOCUS Government Solutions

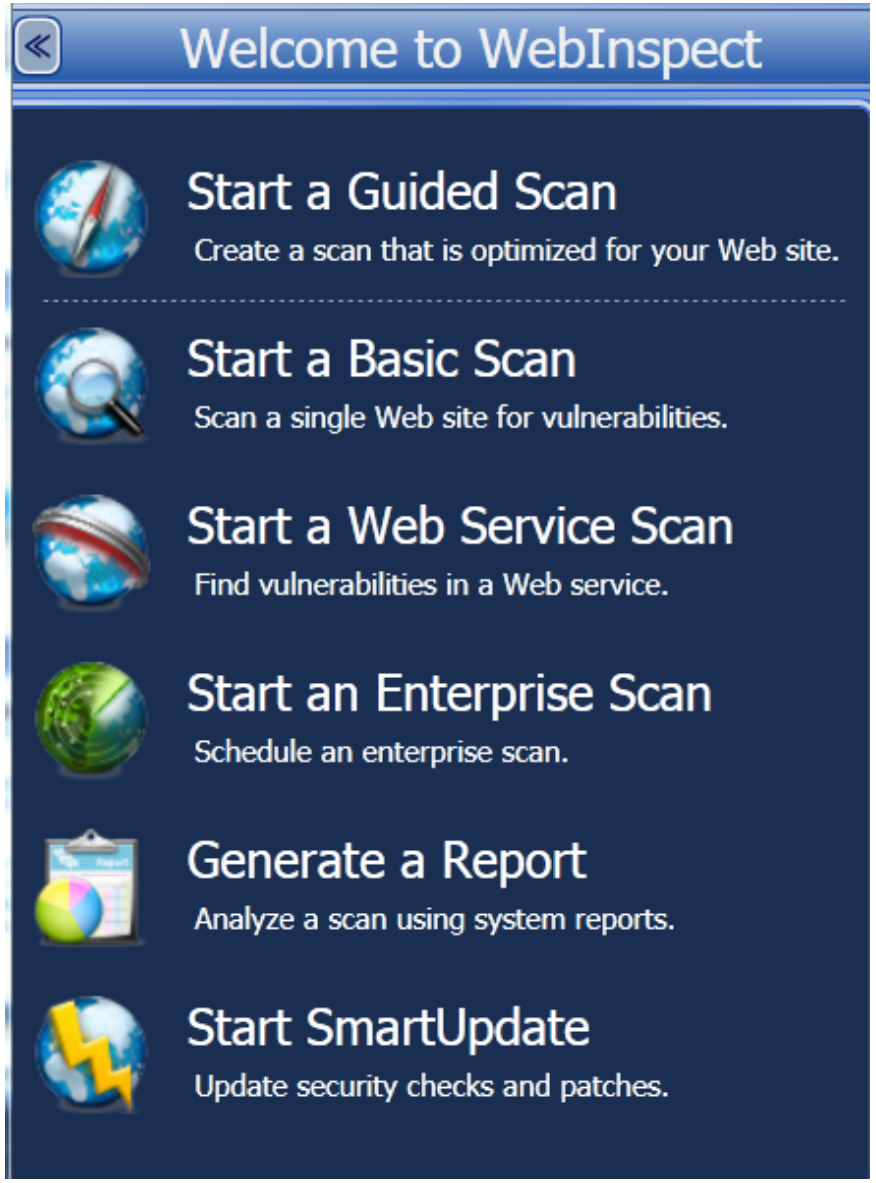
Included In Every WebInspect License

- **SmartCard** / CAC Authentication
- FISMA / 800-53 **Compliance Reporting**
- Scan **Web Applications, SOAP and RESTful** Services, URL Rewriting
- Scan Mobile Web sites, plus Mobile Native Scan
- Integration into **ArcSight, WAFs, Software Security Center, WebInspect Enterprise**
- Hybrid scanning with the **WebInspect Agent**
- WebInspect API plus **BURP** Testing **Integration**
- SmartUpdate automatic frequent security content updates from the largest dedicated Software Security Research group.
- OFFLINE activations and updates

Dynamic application security testing

Quickly find and validate exploitable vulnerabilities in running applications

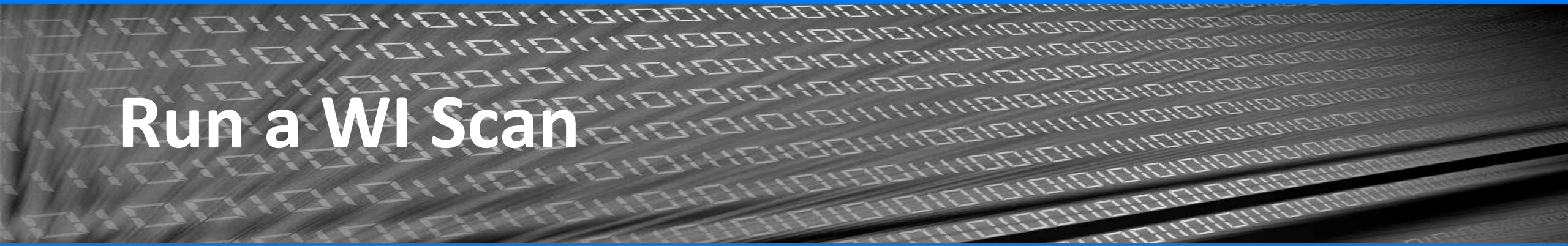
WebInspect Components



MICRO FOCUS
Government Solutions

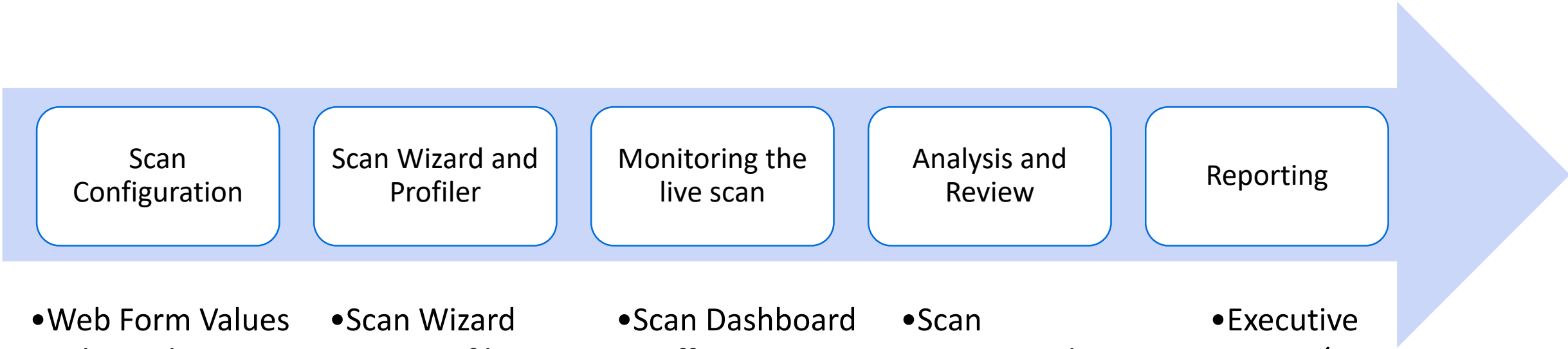
MICRO FOCUS

Government Solutions



Run a WI Scan

High Level Workflow



- Web Form Values
- Policy Selection
- Policy Inputs
- Scan Settings
- Macros

- Scan Wizard
- Scan Profiler

- Scan Dashboard
- Traffic Monitor
- Status Line
- Scan Log
- Sequential View

- Scan Recommendations
- Session Details, Site Search
- Vulnerability Review
- Manual Verification

- Executive
- FISMA/STIG

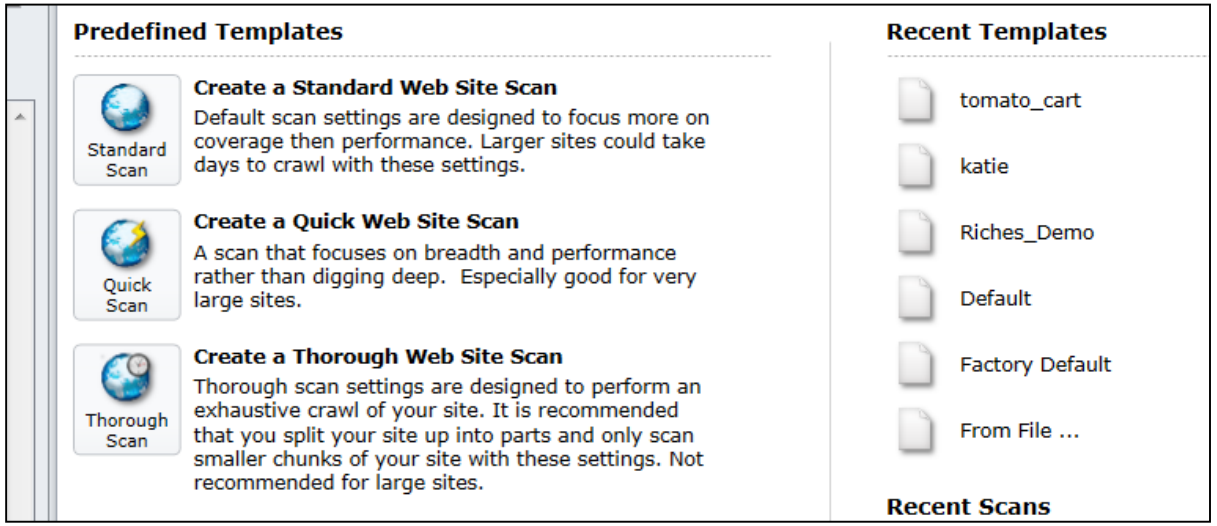
Components of a Scan Configuration

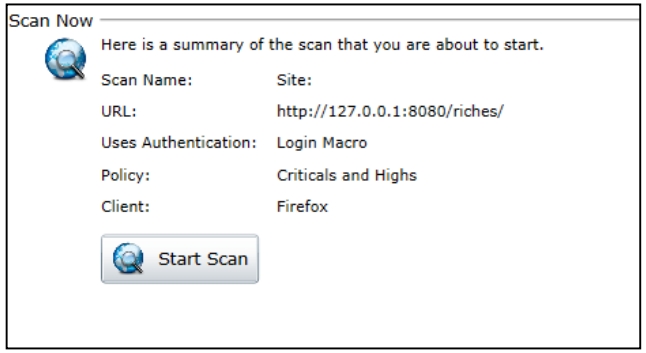
Macros	• Login, starting, workflow • for web form authentication or entry navigation
Web Form Values	• control values supplied to form, which affects crawling
Policy	• collection of tests enabled for use in the scan • Policy inputs via Input Editor
Scan Settings	• detailed configuration for the website

Exercise 3

Scan Riches – Guided Unauthenticated Scan

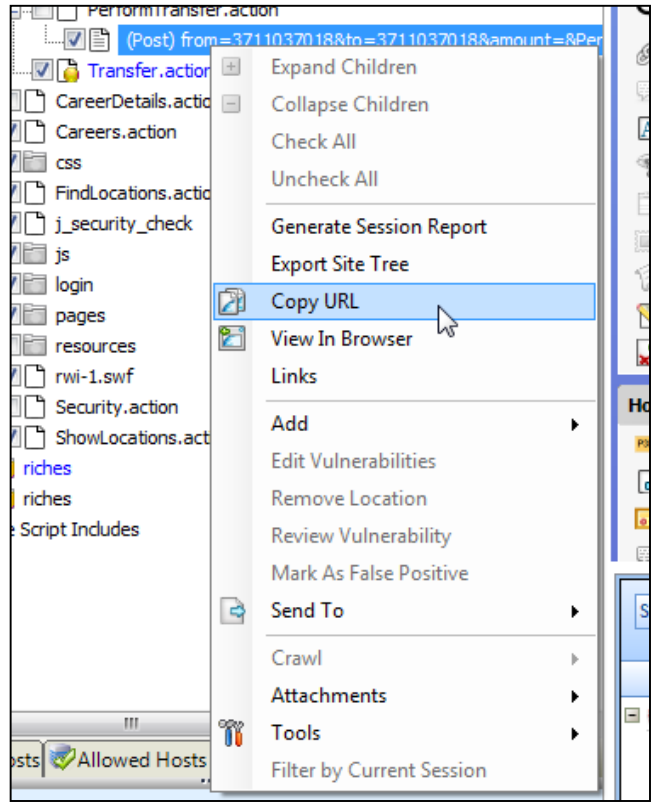
- Select your Template
- Verify http:127.0.0.1:8080/riches/
- Restrict it to this directory and its subs
- Keep Defaults
- Use Riches Optimization
- Enhance Coverage If you wish.
- There are no False Positives to import.
- Turn on Traffic Monitor.
- Save as scan template for later use
- Start Scan





While This Scan is Running

- Notice that vulnerabilities are reported even while still crawling
- Review the growing Site Tree
- Pause, review Sequence View, Search View, and Step Mode
- Explore the Context Menu in the Site Tree
- Traffic Monitor
- Explore HTTP Packets
- Explore Host Info

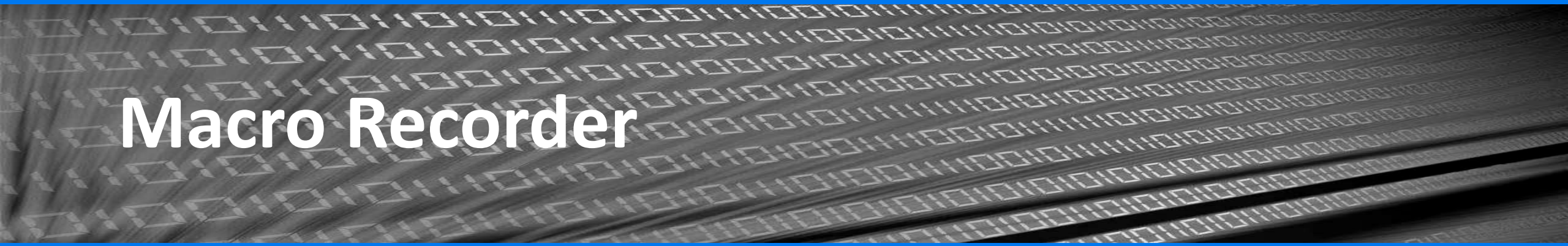


We can leave this scan running, and open a completed scan in a new tab

You can have 2 scans running at a time

MICRO FOCUS

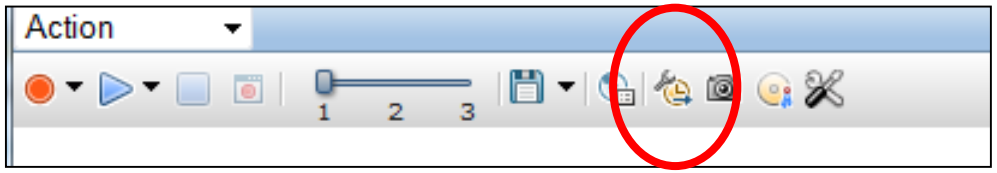
Government Solutions

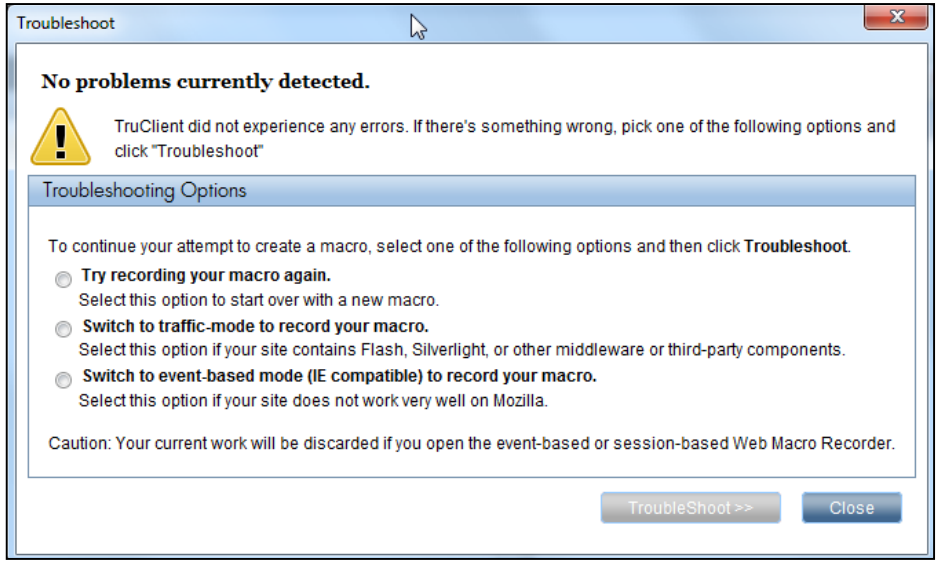


Macro Recorder

Different Macro Recorders

- TruClient (default)
  - advanced client recording developed within HP
  - Used in multiple HP products
  - Records actual browser *actions*
  - Provides advanced configuration and troubleshooting capabilities





Replaces Legacy Recorders

- Traffic-Mode : records and replays basic packets
- Event-based: records events, still more primitive

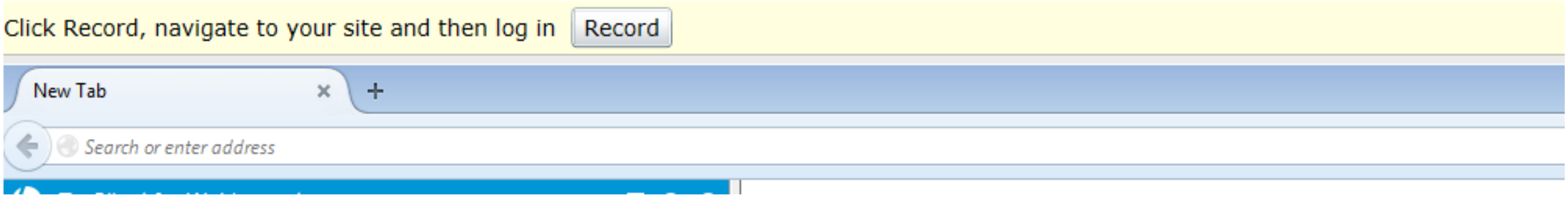
TruClient Features

- IDE-Like environment
- Parameterizing Values (passwords, usernames etc)
- Javascript based events
- Editable Success/Failure Conditions
- Switchable Rendering: IE, FF, variety of Mobile

Exercise 4

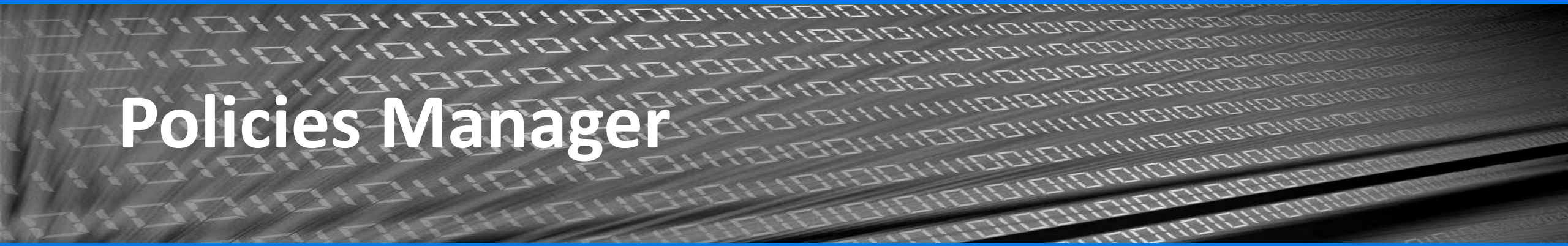
Riches Login Macro

- Go To Tools and open Login Macro Recorder
- Follow the Yellow-Bar Prompts to create a login to the Riches bank site using:
  - **Navigate To URL:** http://127.0.0.1:8080/riches
  - **User Name/Password:** Eddie / Eddie
- Go ahead and test your Macro
- Save the macro as **MYLOGIN**



MICRO FOCUS

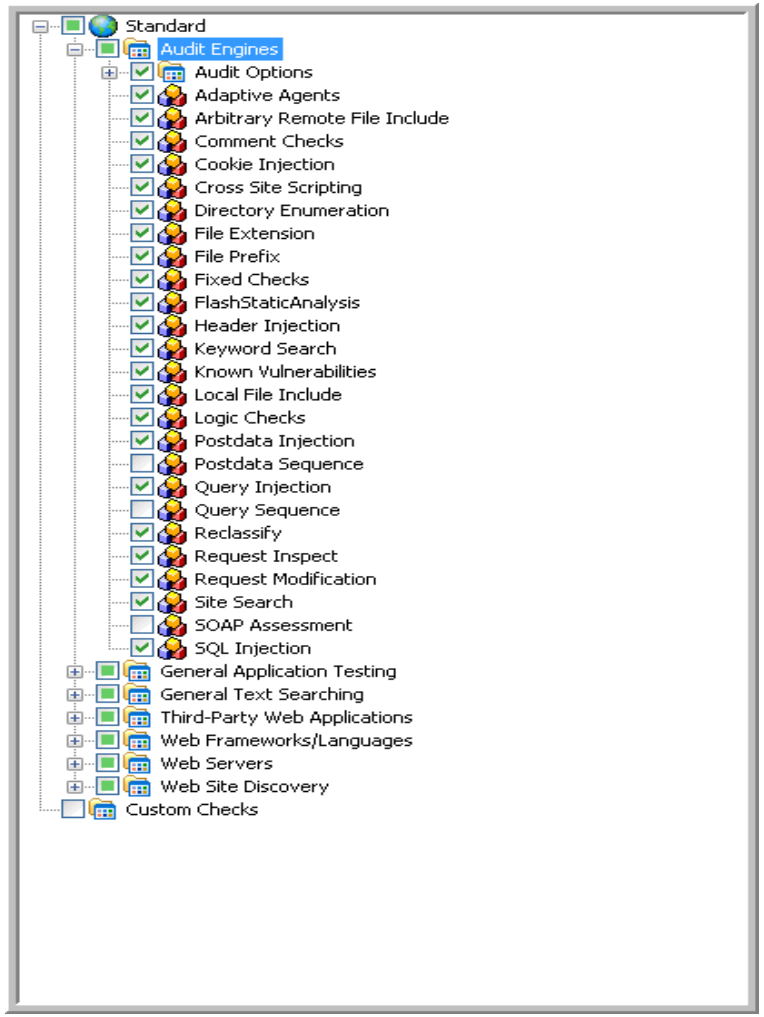
Government Solutions



Policies Manager

Policy Manager Walkthrough

- A Policy is a collection of tests in the SecureBase to be executed.
- Several Default policies exist – they cannot be modified but can be inherited
- You can select and deselect at any level.
- Checks require corresponding Engine
- Custom Checks – create custom checks in different engines.
- Default policy is “Standard”



Exercise 6

How to Create a Policy

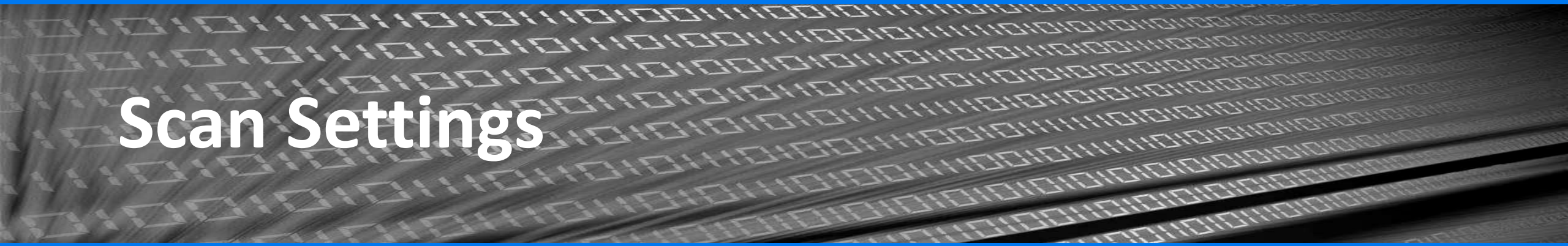
- Click **Create** and The Policy Manager tool opens.
- Select **New** from the **File** menu (or click the New Policy icon).
- Select the policy on which you will model a new one from:
  - SQL Injection Policy → Select Command Injection → SQL Injection → Site Database Disclosure and SQLI
- Save
- Under Custom → My Policy
- When you configure your settings, pick your new My Policy for scan

Exercise 6 (Cont')

- Do a New Critical and Highs Policy
  - Default Settings → Policy → Create → File → New → Criticals and Highs Policy
  - Search for and deselect all checks mapped to CWE 521
    - Search View → Criteria → Vulnerability ID = CWE ID Contains 521
    - Deselect All
    - Save the new Custom Policy **MYCWEPOLICY**

MICRO FOCUS

Government Solutions



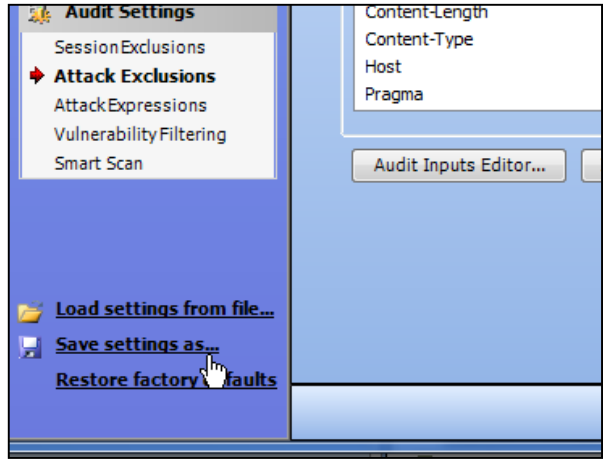
Scan Settings

Settings

- Auto fill web forms during crawl
- Prompt for web form values during scan
- Only prompt for tagged input
- Enable Traffic Monitor
- Perform redundant page detection Limit Maximum Web Form Submission To
- Enable Site-Wide Event Reduction

Exercise 7

- Go to Edit and open the Default Scan Settings
- Change the following Settings:
  - Method: Auto fill web forms during crawl – **MFFORMS**
  - Method: Prompt for web form values during scan**: True**
  - Method: Only prompt for tagged input: **True**
  - General: Enable Traffic Monitor:**True**
  - General: Perform redundant page detection: **True**
- Save the Settings as **MYSETTINGS**



You now have your entire configuration stored in a reusable format.

MICRO FOCUS

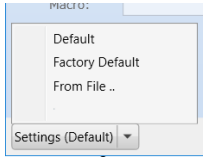
Government Solutions

Putting it all together

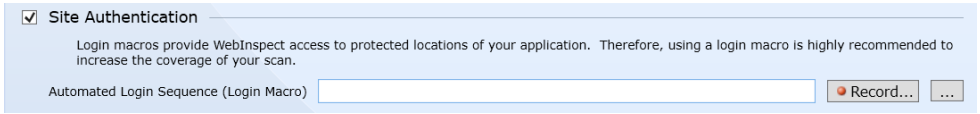
Exercise 8

- Start a Basic Scan

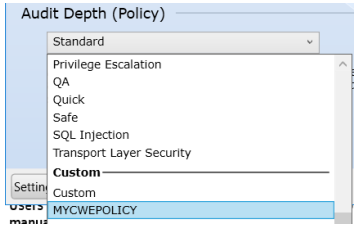
- Load **MYSETTINGS**



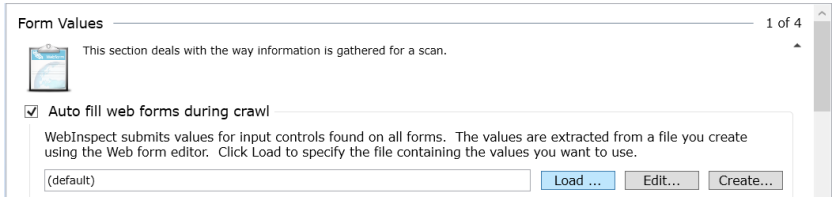
- Check Site Authentication: **MYLOGIN**



- Choose your Audit Depth: **MYCWEPOLICY**

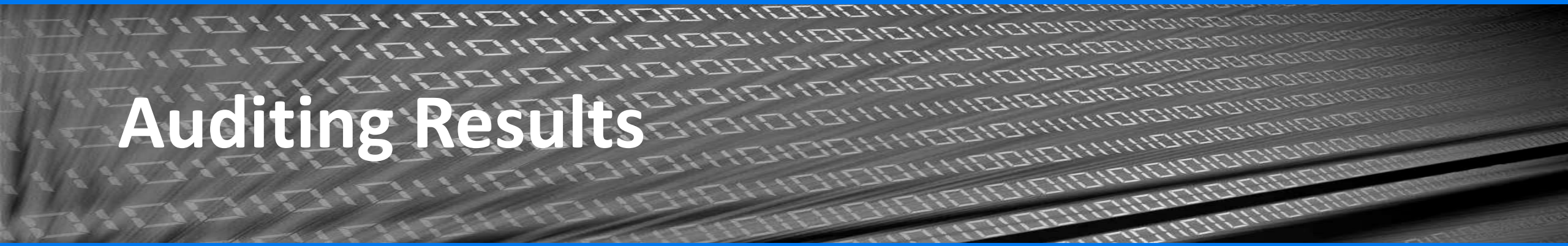


- Choose your web forms: **MYFORMS**



MICRO FOCUS

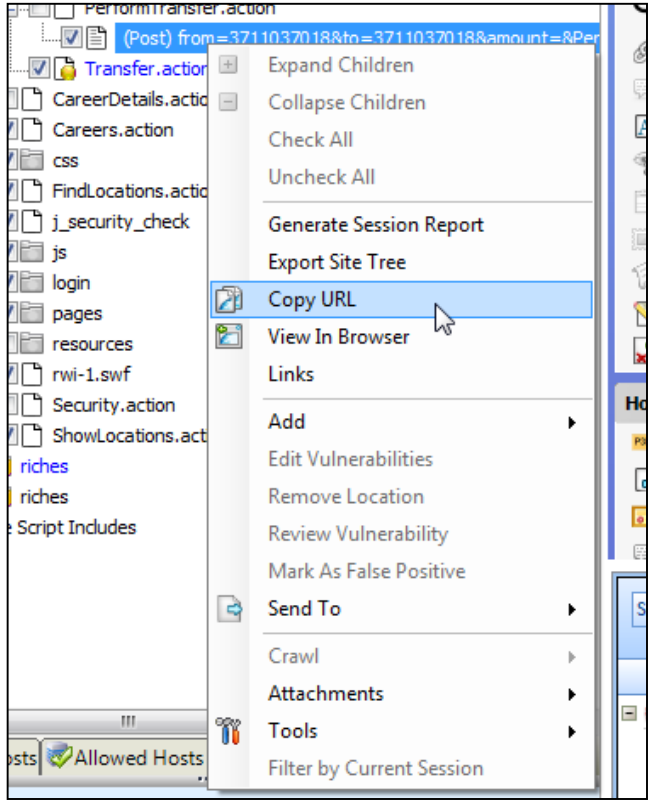
Government Solutions



Auditing Results

While This Scan is Running

- Notice that vulnerabilities are reported even while still crawling
- Review the growing Site Tree
  - Pause, review Sequence View, Search View, and Step Mode
  - Explore the Context Menu in the Site Tree
- Traffic Monitor
- Add the column “Location” and move up
- Explore HTTP Packets
- Explore Host Info



We can leave this scan running, and open a completed scan in a new tab

You can have 2 scans running at a time

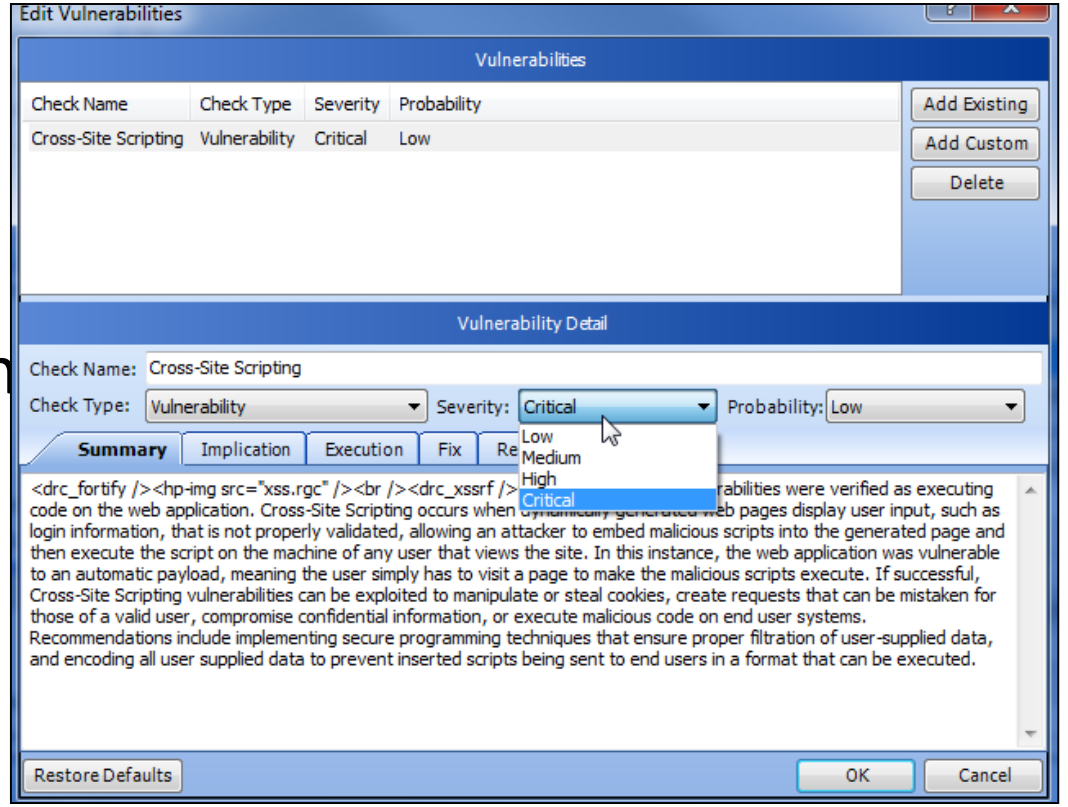
Exercise 9

Validate a finding

- Look at request, response, vulnerability description and stack trace
- Retest the vulnerability
- Look at one finding
- Select “Edit Vulnerability” and change severity for one
- Add your own information to the reporting.
- Find one false positive and mark it as such

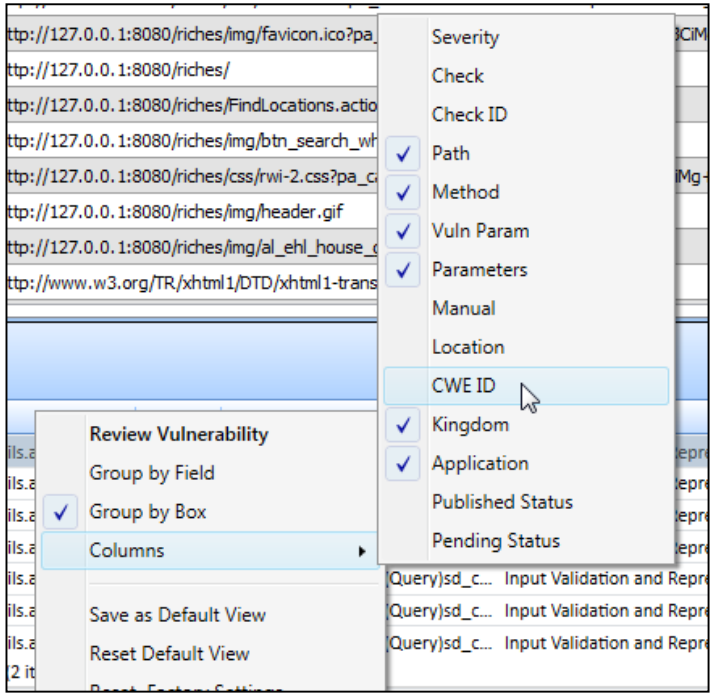
Edit a Vulnerability

- Right Click a Vulnerability in the Site Tree
- Select “Edit Vulnerability”
- Change the Severity
- Add your own information to the reportin
- This is all saved to the scan file
- Flows upstream with scan:
  - Reports,
  - FPR’s into Software Security Center,
  - Scan uploads into WebInspect Enterprise



Review a Completed Scan

- Review the Site Tree
- Vulnerability Pane:
  - Select column “CWE ID”
  - Review packets for a vulnerability
  - Change severity for a vulnerability
  - Add notes to a vulnerability
  - Review Steps
- Retest a single vulnerability / All Vulnerabilities/Repeat Scan
- Add your own manual finding (Edit Vulnerabilities)



MICRO FOCUS

Government Solutions

Reports and Data Export

Exercise 10

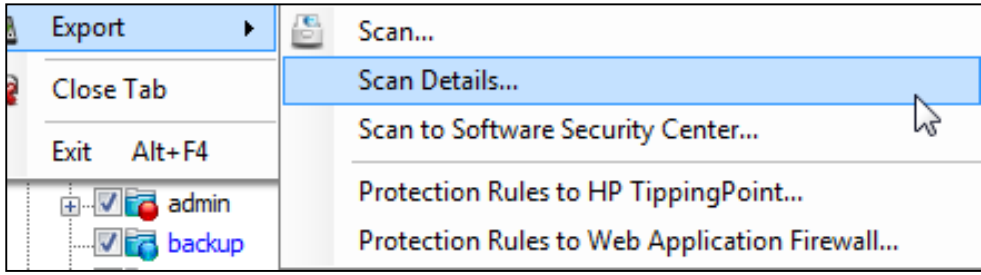
Generate a Report

- Use your existing scan
- Select Executive Summary, Compliance and Vulnerability
- For Compliance: Select DoD Application Security And Development STIG V3 R9 or FISMA.
- For Executive Summary: No change
- For Vulnerability: Clear ALL Severities except Critical.

Exercise 11

Export Data

- Scan
  - exports to proprietary binary .scan format
  - saves entire scan and be re-loaded into WebInspect
- Scan Details
  - Exports selectable sections or full scan
  - XML based - can be used for integration (STIG Viewer)
- Scan to Software Security Center
- Saves results in FPR format for uploading into Software Security Center to be managed alongside static scans



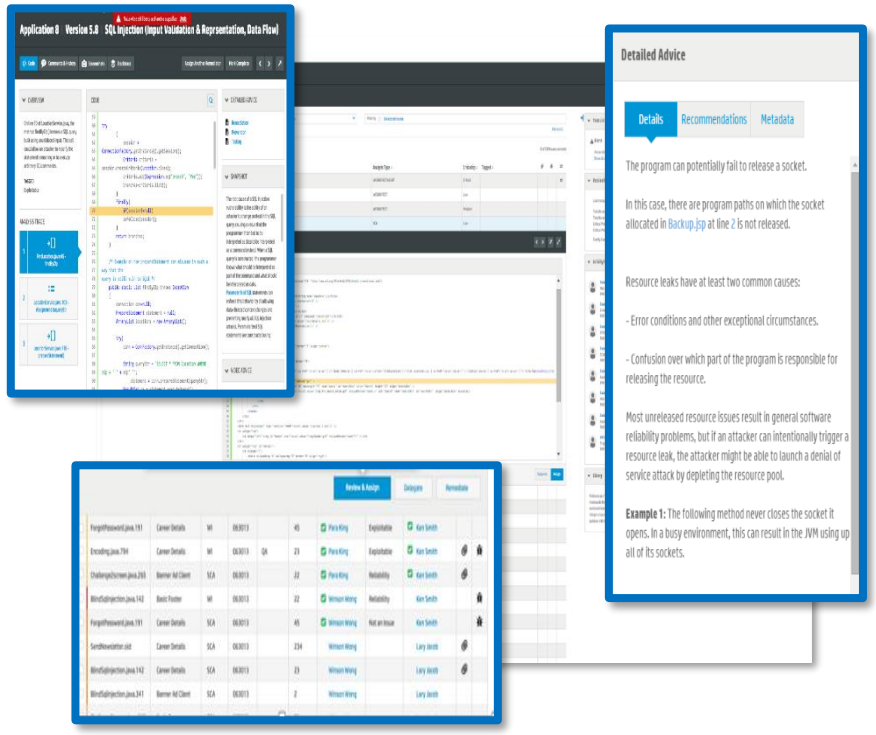
MICRO FOCUS

Government Solutions

HPE Softrware Security Center Overview and Exercises

HPE Fortify Software Security Center

Management, tracking and remediation of enterprise software risk



Problem it solves:

Provides visibility into security activities within development

Features:

- Specify, communicate and track security activities performed on projects
- Role-based, process-driven management of software security program
- Flexible repository and exporting platform for security status, trending and compliance

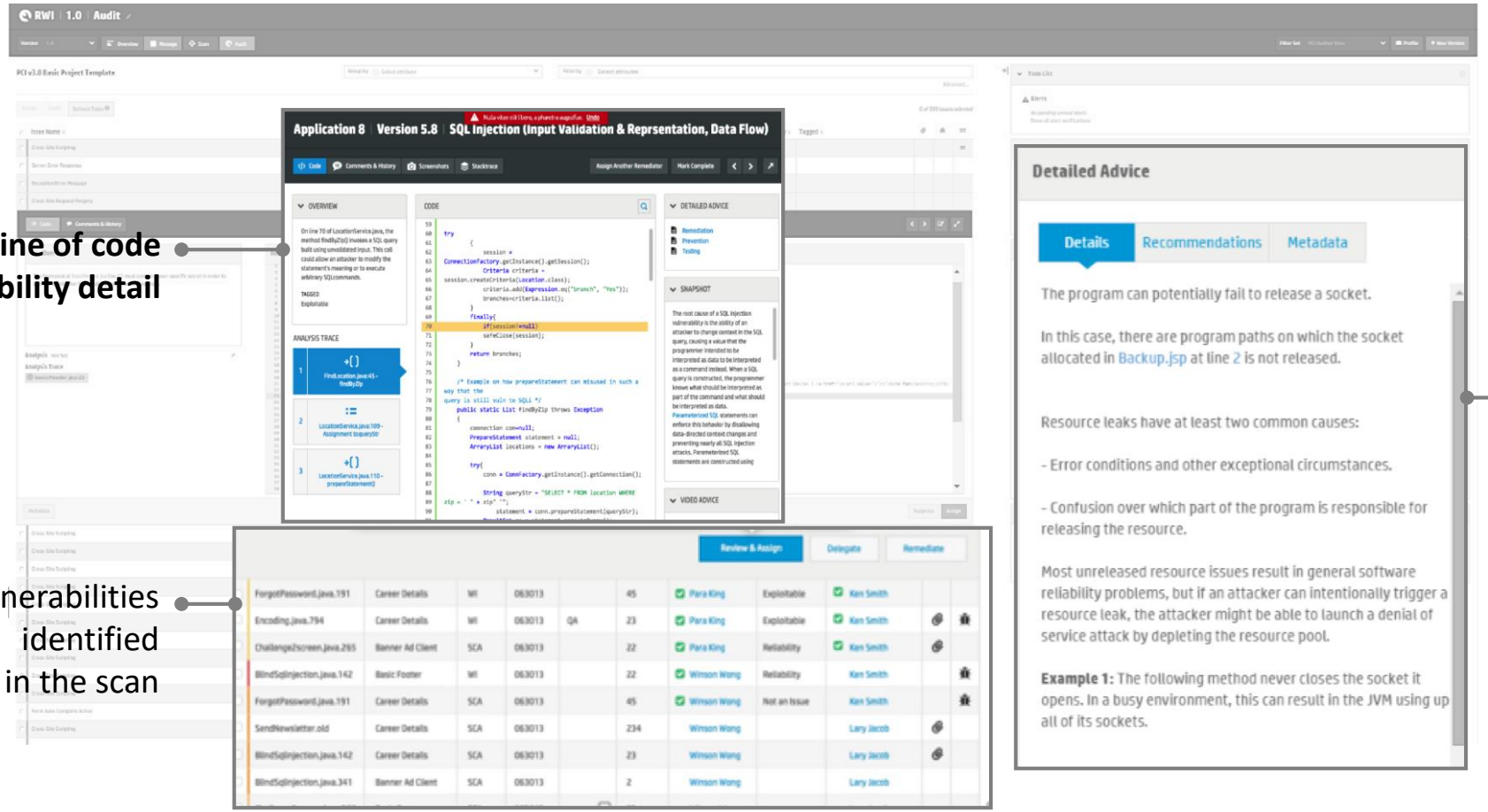
Benefits:

- Provides a clear, accurate picture of software risk across the enterprise
- Lowers cost of resolving vulnerabilities
- Identify areas of improvement for accelerated reduction of risk and costs

MICRO FOCUS
Government Solutions

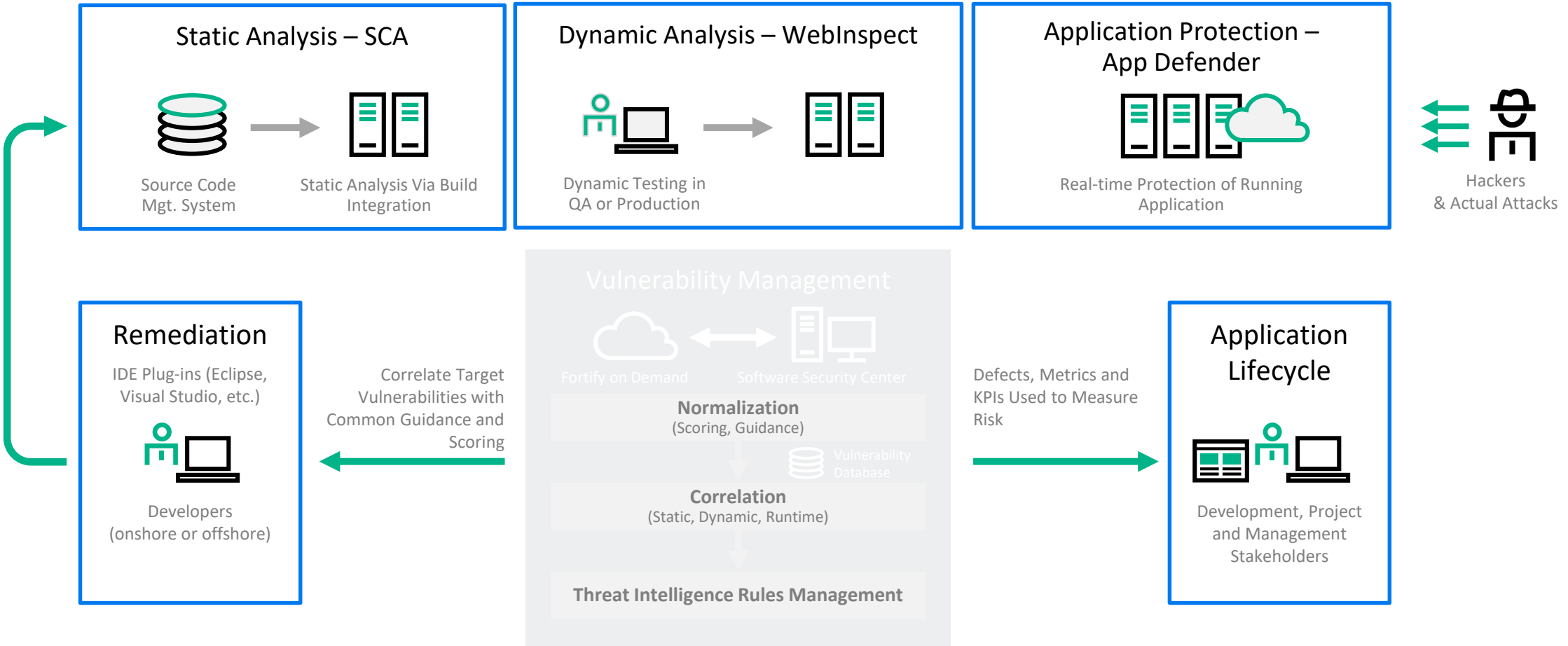
HPE Security Fortify Software Security Center

Vulnerability detail



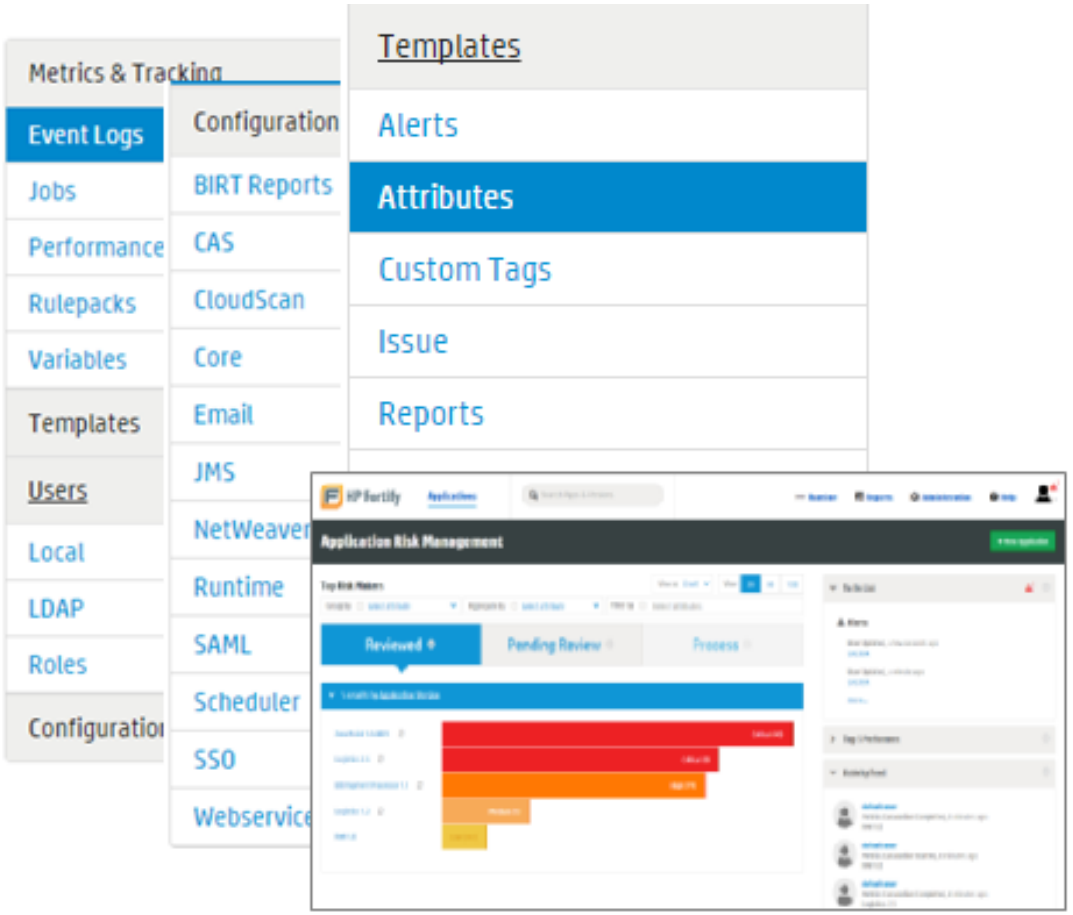
HPE Security Fortify Application Security Solutions

On premise and on demand



SSC Functional Areas

- **SSC Administration**
  - **User Access**
  - **System Configuration**
  - **Issue Template (Project Template)**
- **Application (Project) Administration**
  - **Application Management**
  - **Attributes Assignment**
- **Program Management**
  - **Audit Page (Collaboration Module)**
  - **Reporting**

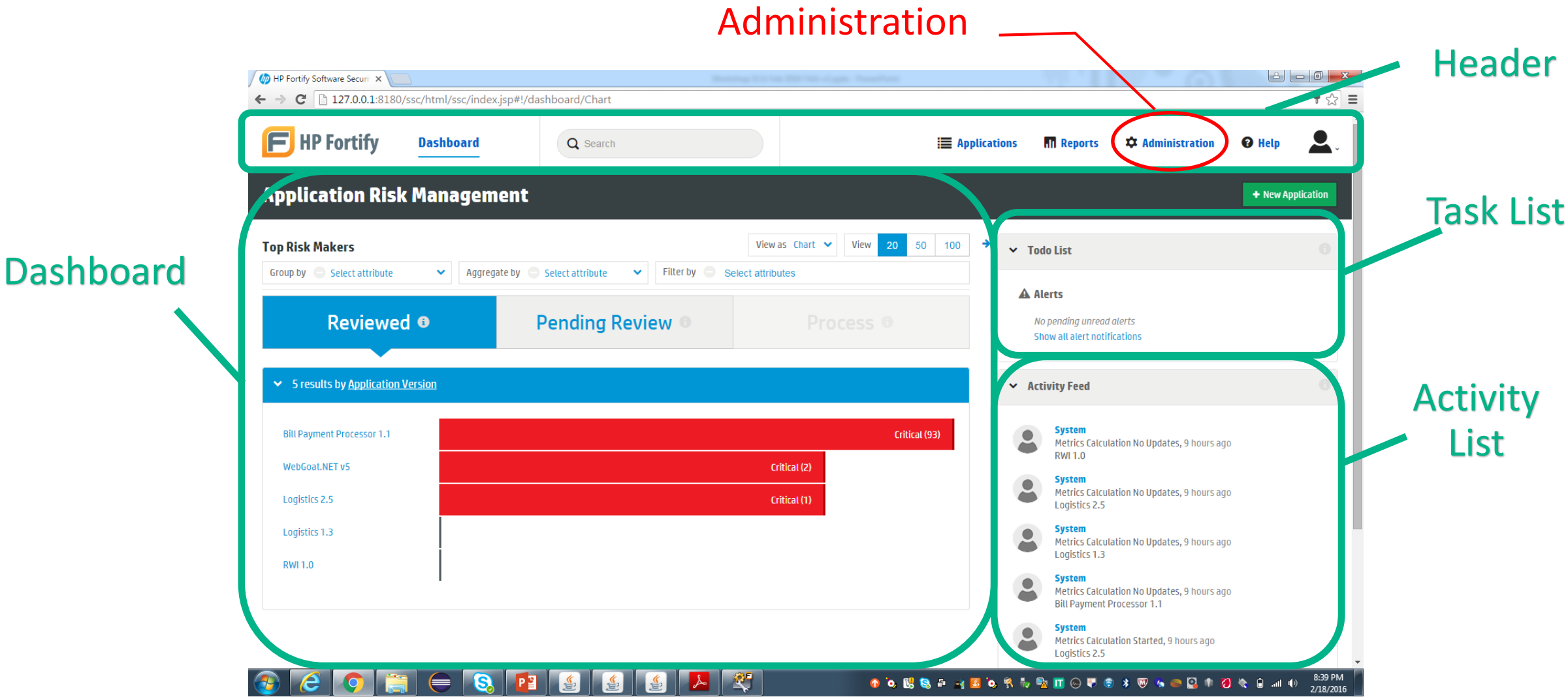


Exercise 12:

Software Security Center Walk Through

1. Click on “Launch the Fortify SSC Server”
2. Open a web browser
3. Navigate to http://localhost:8180/ssc
4. Login information is in student_logins.txt on your Desktop. Log in as admin.
5. Password is Workshop2017!

SSC Interface



MICRO FOCUS
Government Solutions

Exercise 13

Create a New Project

1. Click on “Launch the Fortify SSC Server”
2. Open a web browser
3. Navigate to http://localhost:8180/ssc
4. Login information is in student_logins.txt on your Desktop. Log in as admin.
5. Password is Workshop2017!

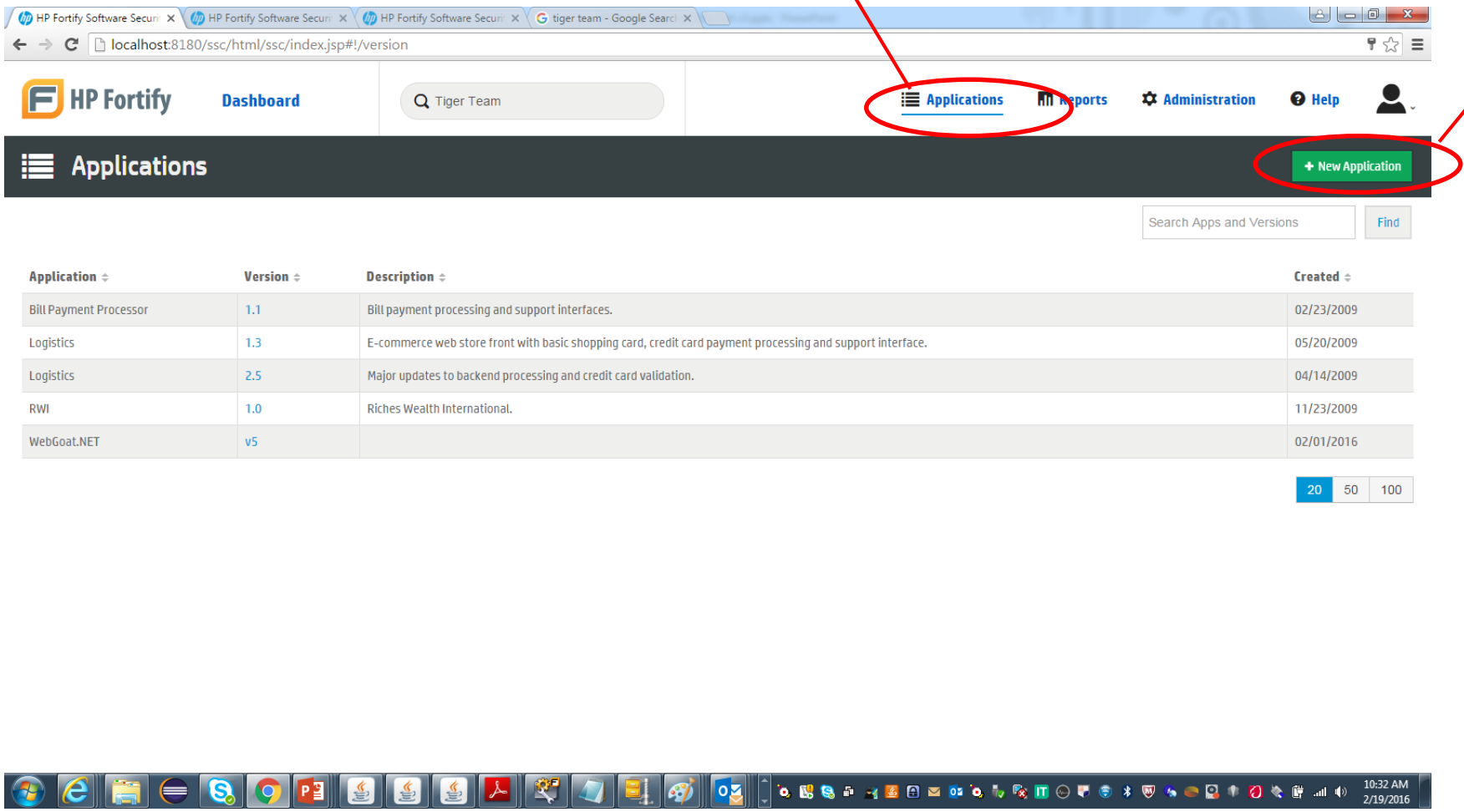
New Application

Name: Riches2

Version: v9

Development Phase: New

Create A New Application

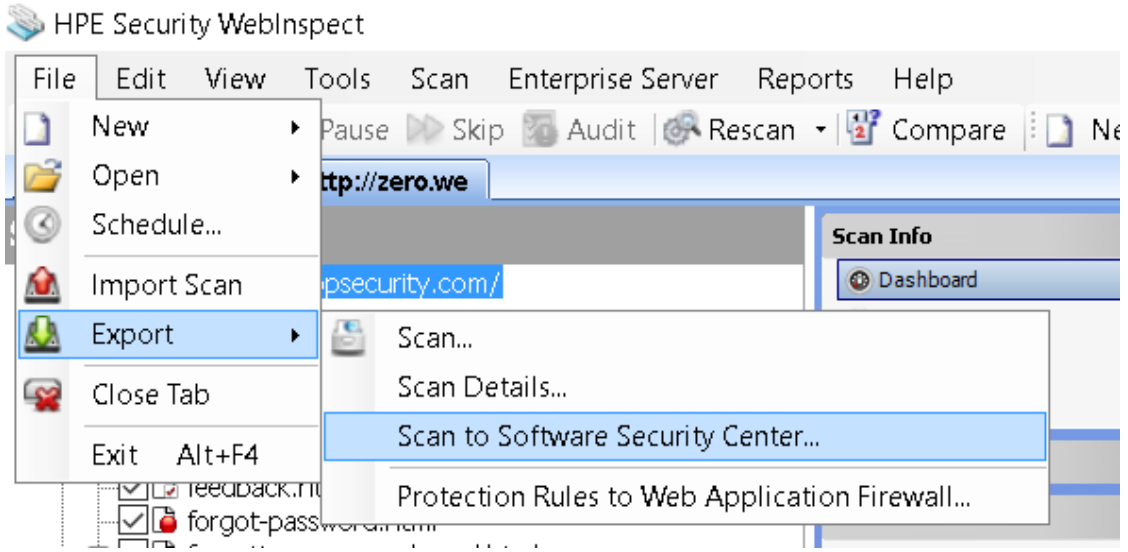


Exercise 14

Upload Results to SSC from WI

- From WebInspect open a completed WebInspect scan
- Export the scan

  File->Export->Scan to Software Security Center
- Save the file as **MYSCAN.fpr**

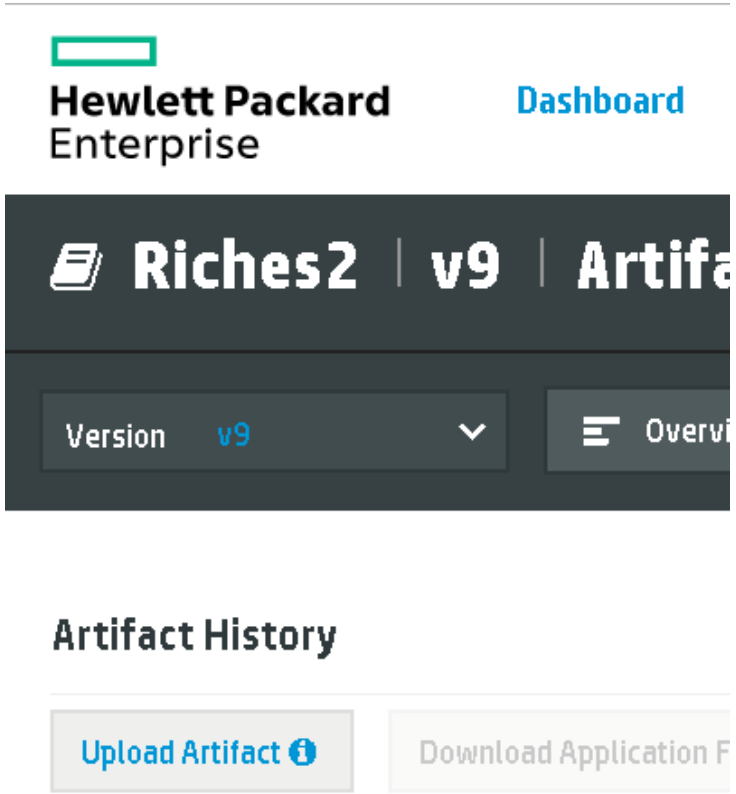


Exercise 14 Continued

Upload Results to SSC from WI

- From SSC navigate to the **Riches2** project

  Click **Applications- Riches v9**
- Navigate to Artifacts
- Upload the **MYSCAN.fpr**



The scan should be available for viewing after SSC finishes processing the file

MICRO FOCUS

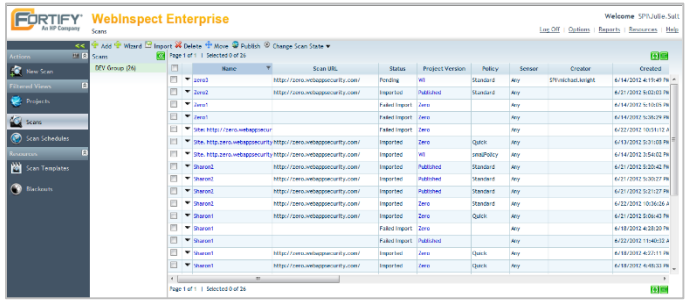
Government Solutions

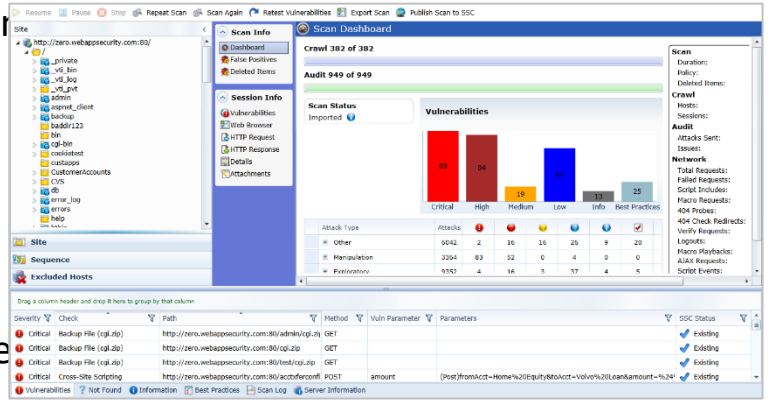
WebInspect Enterprise

HPE WebInspect Enterprise

Extending effective application security testing across the entire enterprise

- **Problem it solves**:
  - Manages large-scale, distributed security testing programs across thousands of applications
    - Continuous application monitoring
    - On-demand application scanning
- **Features:**
  - Monitor critical metrics, progress and trends across large-scale application security testing programs
  - Provide an ongoing enterprise-wide view of production and pre-production applicatio security assurance
  - Control your application security program through role-based scanning and reporting administration
- **Benefits:**
  - Eliminate inefficient and inconsistent assessment and vulnerability management proce
  - Increase visibility and control of security testing efforts and reporting
  - Prove compliance with regulations, standards and policies





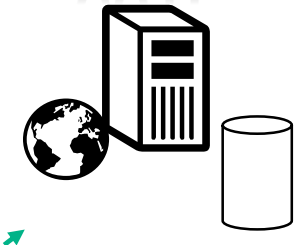
MICRO FOCUS
Government Solutions

WI/WIE Conceptual Architecture - Types

Flexible modeling for Web App Continuous Monitoring and Centralized Vulnerability Management

WebInspect

WIE



Enterprise Manager

On-Demand

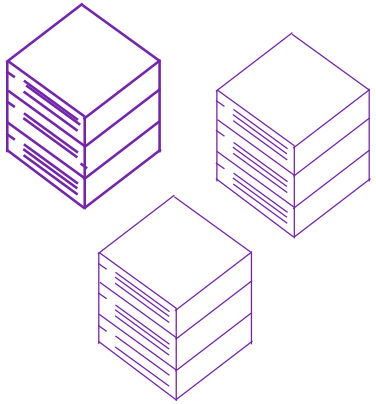
Continuous Monitoring

Customers get to kick off and manage their own scans without relying on security availability

LIM Server

Testers share licenses via check-in/check-out

Sensors



Software Security Center

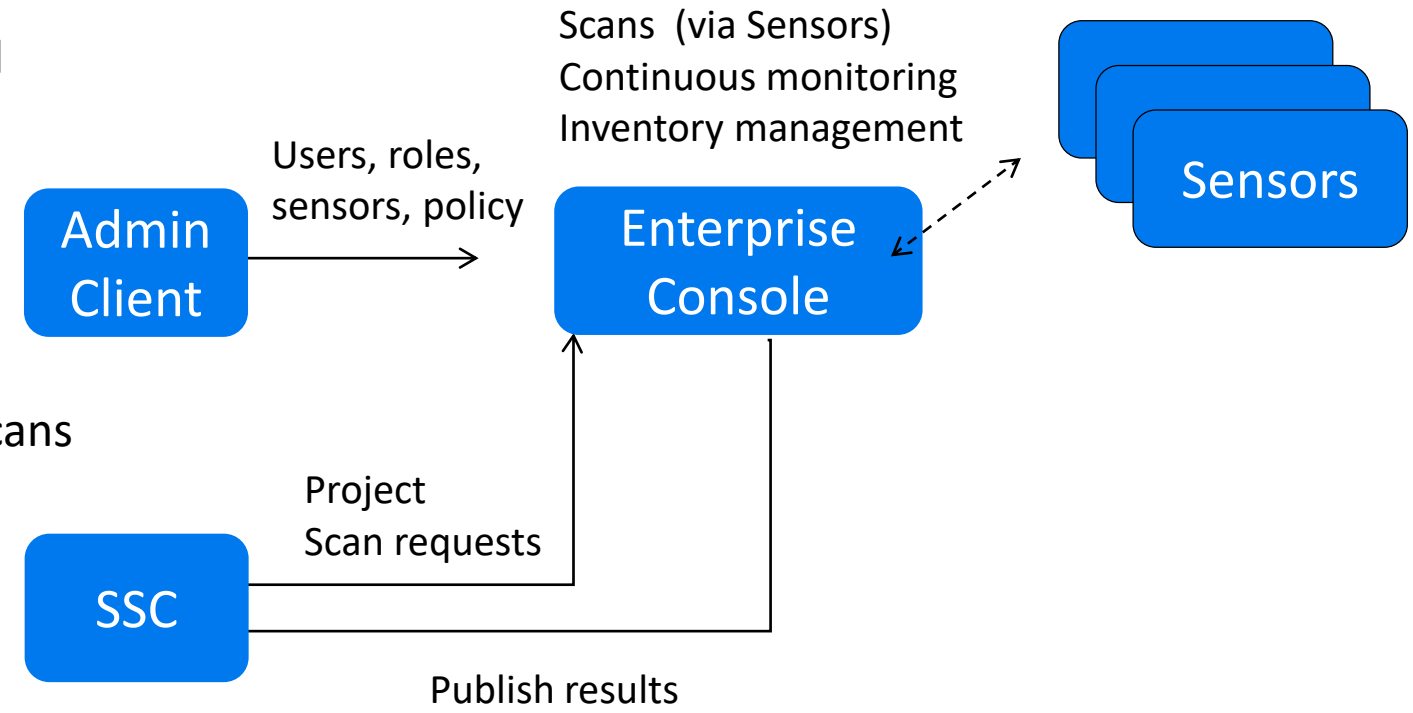
Custom Reporting

Management gets enterprise metrics, dashboards and reporting

MICRO FOCUS Government Solutions

WebInspect Enterprise Components

- **Software Security Center (SSC)** – Start and end of the process
  - Create new projects, request scans, review results and run reports
- **Administration Client** – Manage WIE users and components
  - create and manage groups, users, roles, and policy
  - Administer sensors
  - Update rulepacks
- **Enterprise Console** – Web portal to manage scans
  - run, schedule and publish scans
  - View list completed scans
  - View list of scan requests
- **WebInspect Sensor**
  - Headless WebInspect to conduct scans



WI/WIE Manpower Efficiencies

Advantage	WI	WIE
Reduce Reporting and Compliance Costs	X	X
Achieve Greater Accuracy with Less Effort	X	X
Handle Complex Applications Out-of-the-Box	X	X
Scale Using Technology Not People		X
Pool Security Expertise		X
Streamline Enterprise Security Integration		X

Additional Topics

MICRO FOCUS

Government Solutions

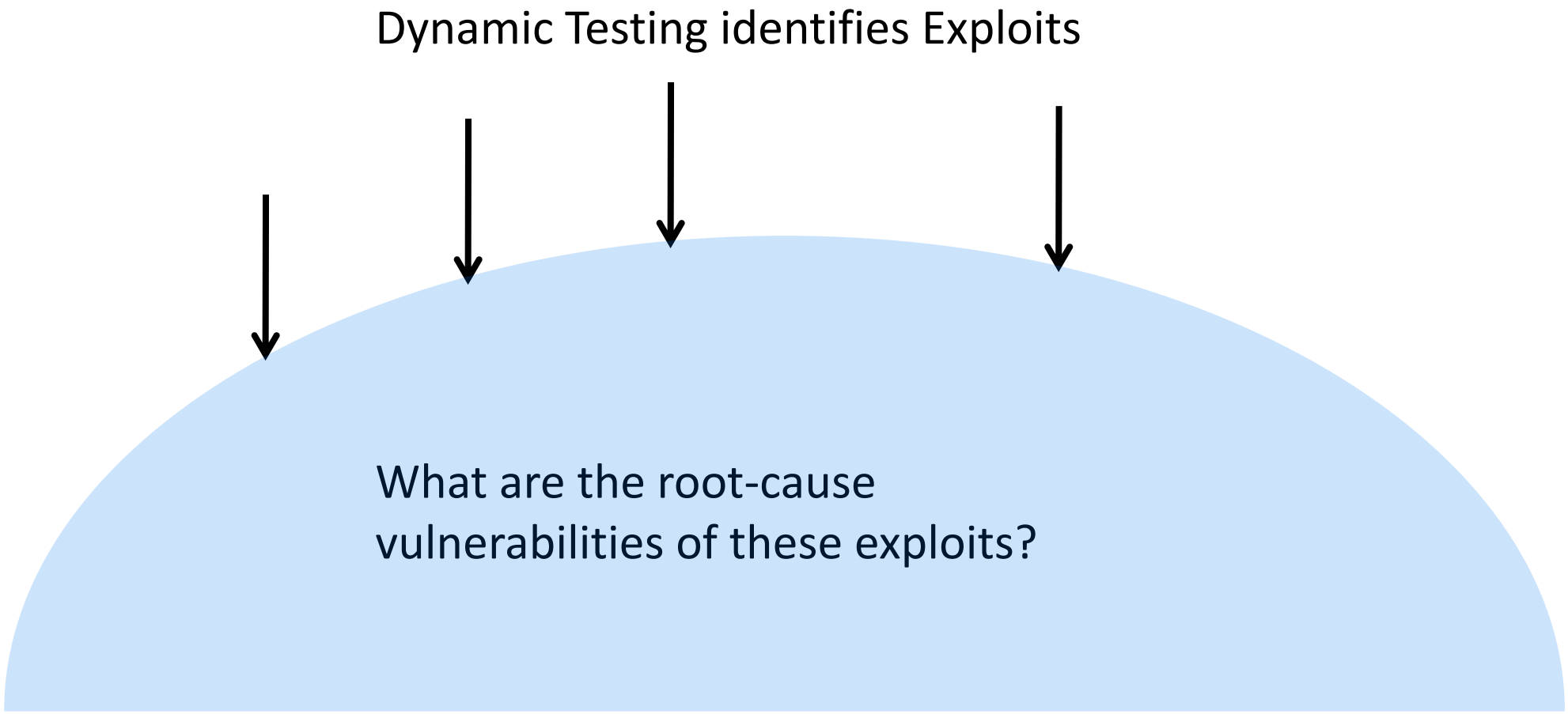
MF Fortify Hybrid Analysis

WebInspect Hybrid Dynamic and Static Analysis

HP WebInspect Agent combines the power of WebInspect and Fortify SCA to perform hybrid dynamic and static testing for Java and .NET applications.

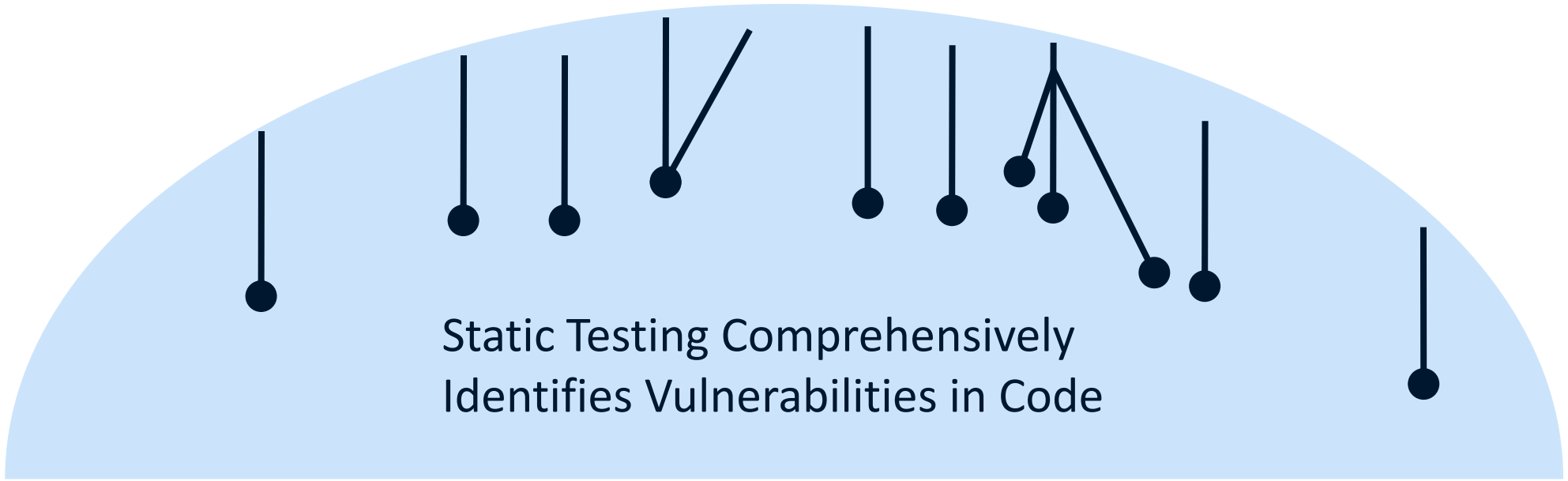
- Analyze more of the application via automatic attack surface identification from inside the application.
- Detect new types of vulnerabilities beyond just dynamic testing, e.g., Privacy Violation, Log Forging.
- Confirm vulnerabilities inside the code.
- Provide actionable details, such as the stack trace and the exact line of code.
- Collapse duplicate issues by identifying the root cause.

Dynamic Testing



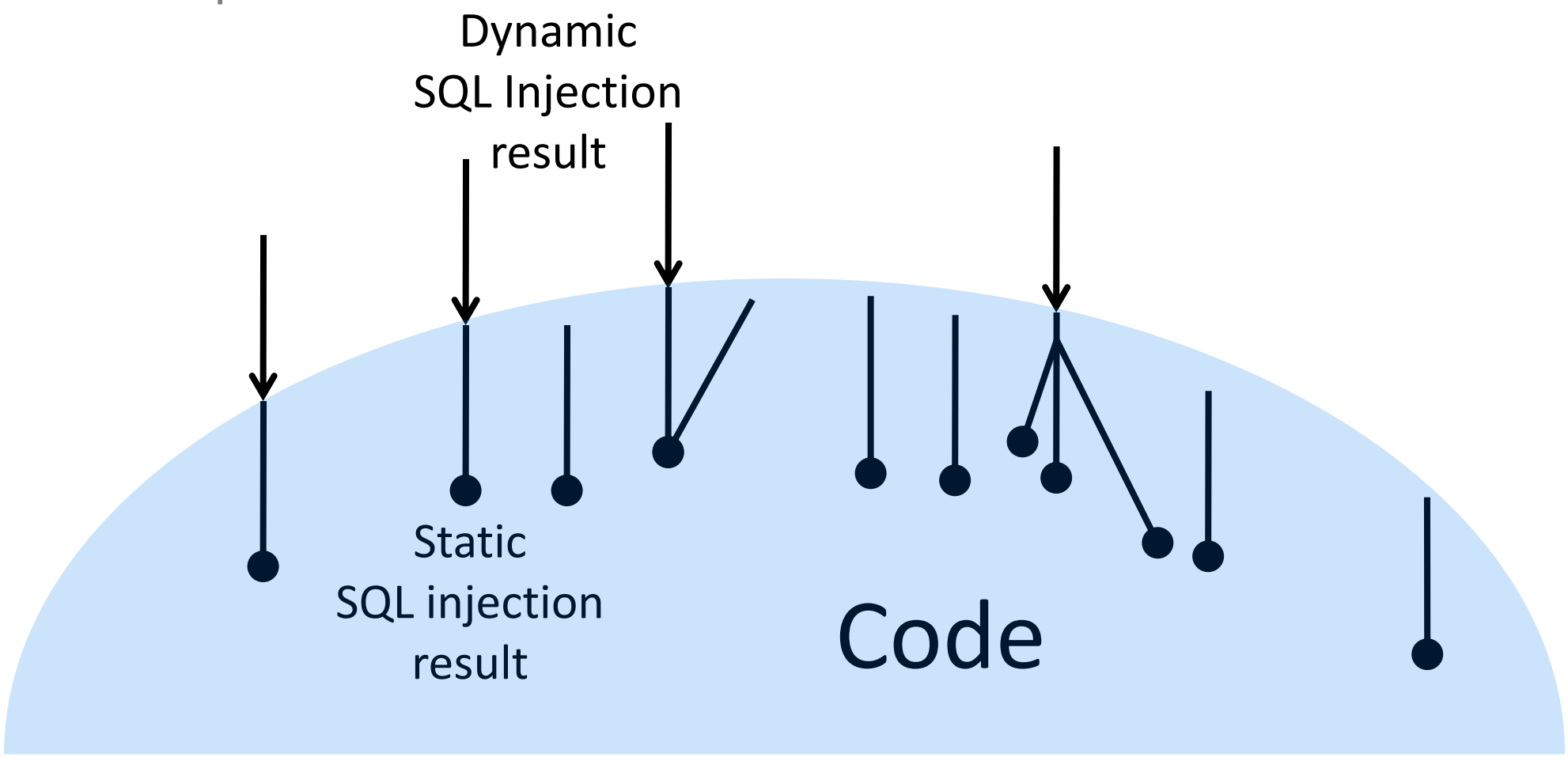
Static Testing

Which vulnerabilities are most important to fix?

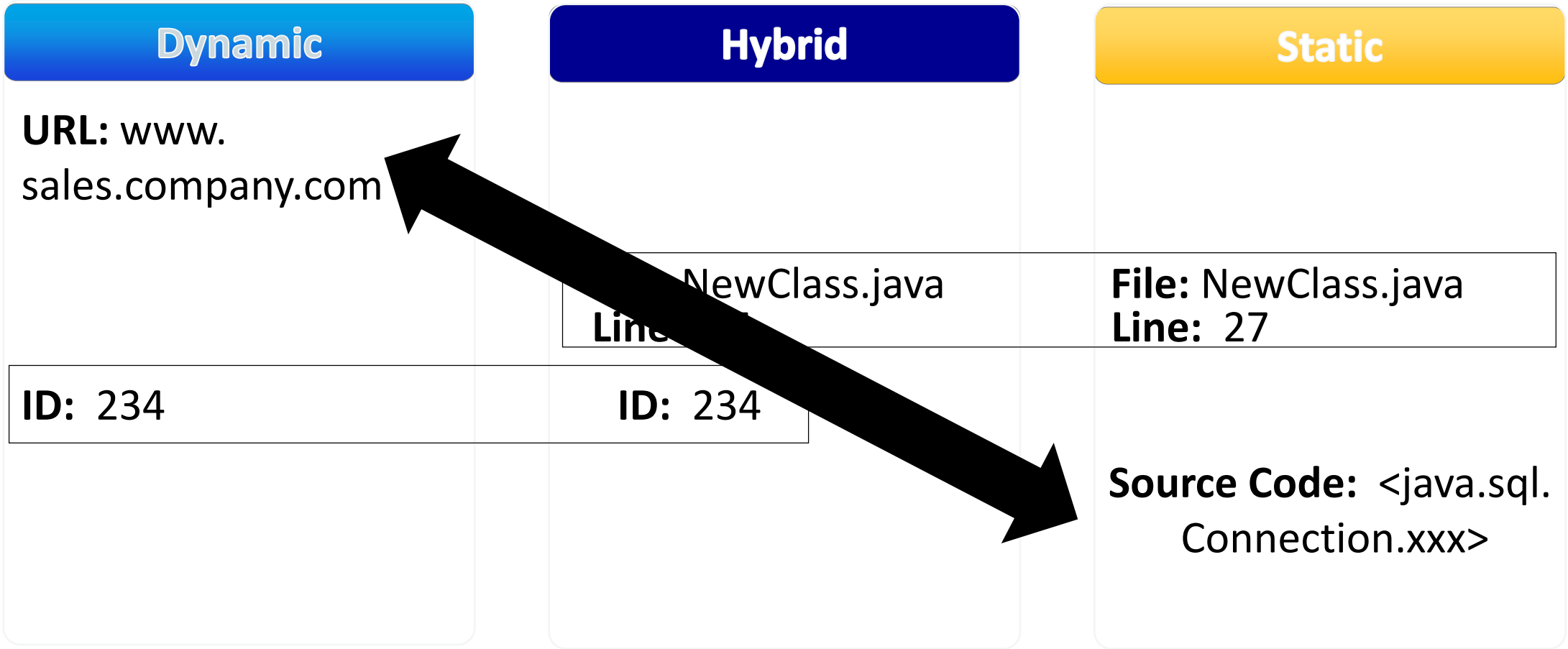


Hybrid Technology

Correlates Exploits with Vulnerabilities

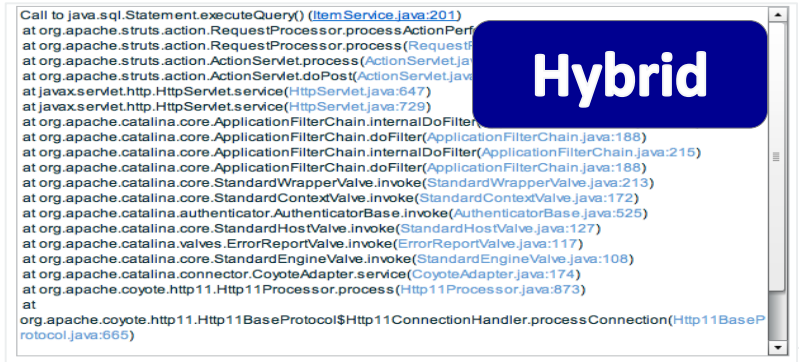


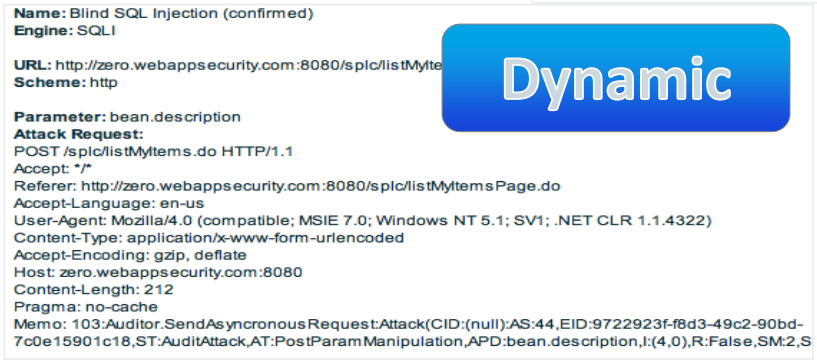
Hybrid Technology

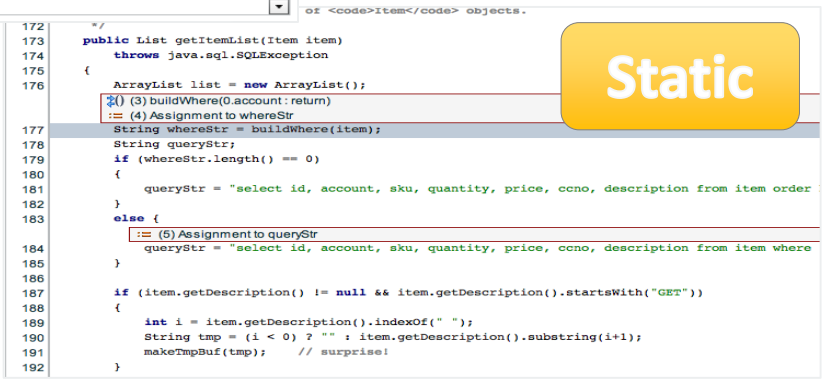


Fortify Real-Time Hybrid Analysis

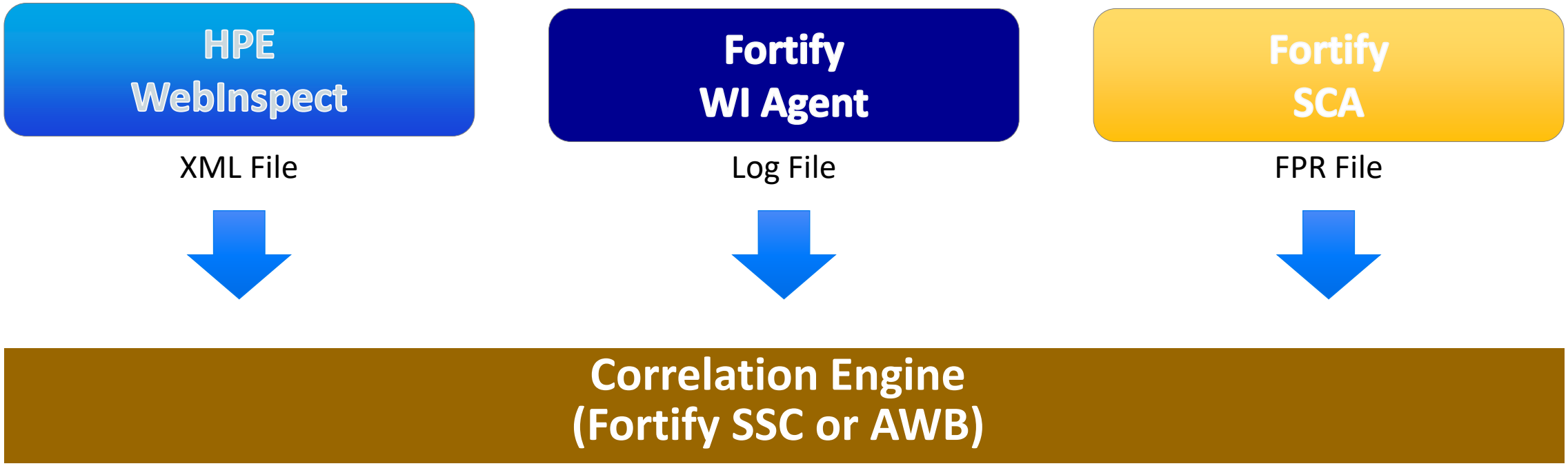
Real-Time Hybrid Analysis = **HPE WebInspect** + **Fortify WI Agent** + **Fortify SCA**







Fortify Real-Time Hybrid Analysis



Real-Time Hybrid Analysis: Benefits

Relevance	• Understand the context of vulnerabilities • Find the root cause
Importance	• Prioritize your resources and time • Fix the most critical vulnerabilities
Speed	• Fix security issues fast • Release secure applications to market quickly

MICRO FOCUS

Government Solutions

Thank you

Sameer Kamani
(Sameer.Ashwin.Kamani@microfocusgov.com)