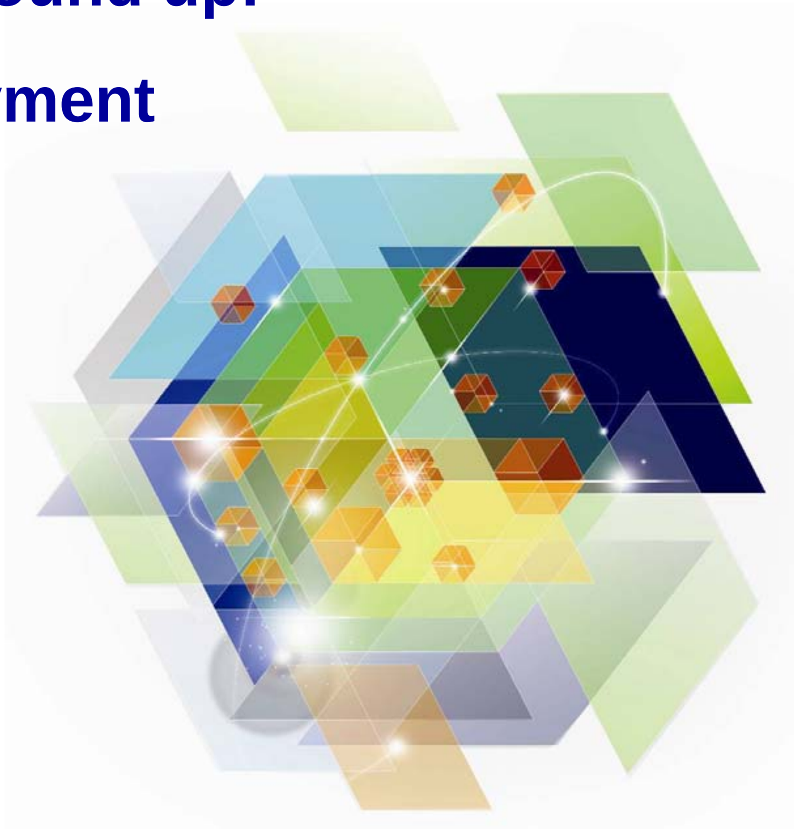
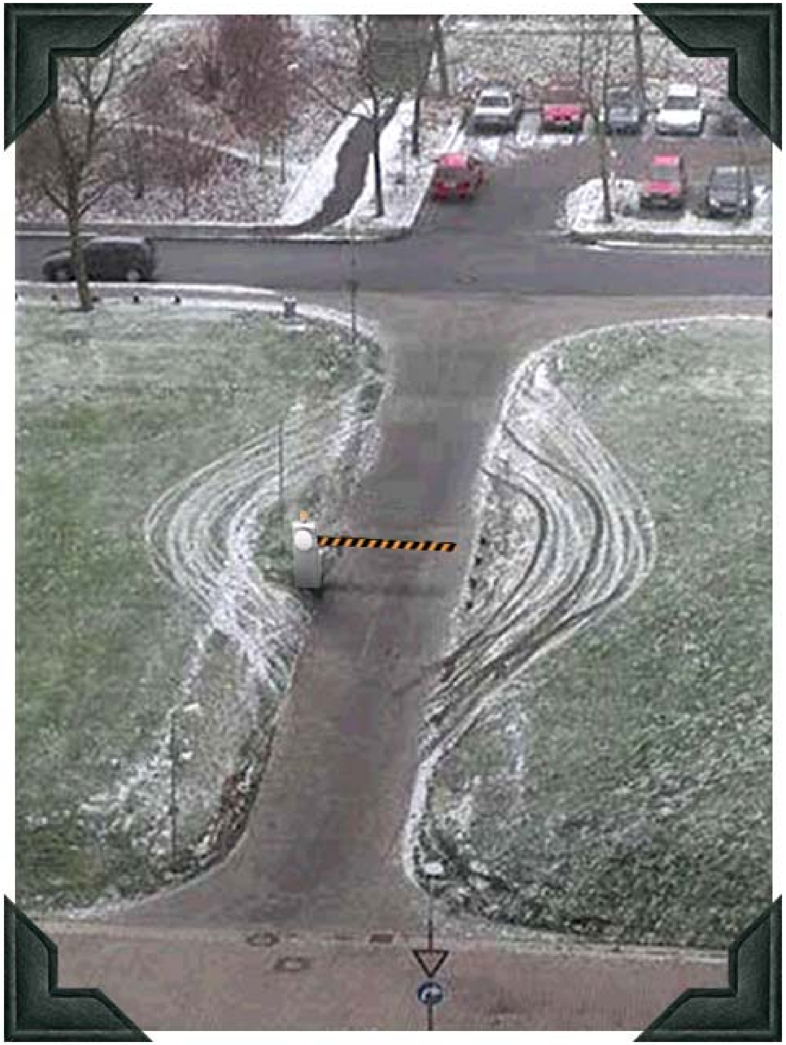
Securing applications from the ground up:

Development, delivery and deployment



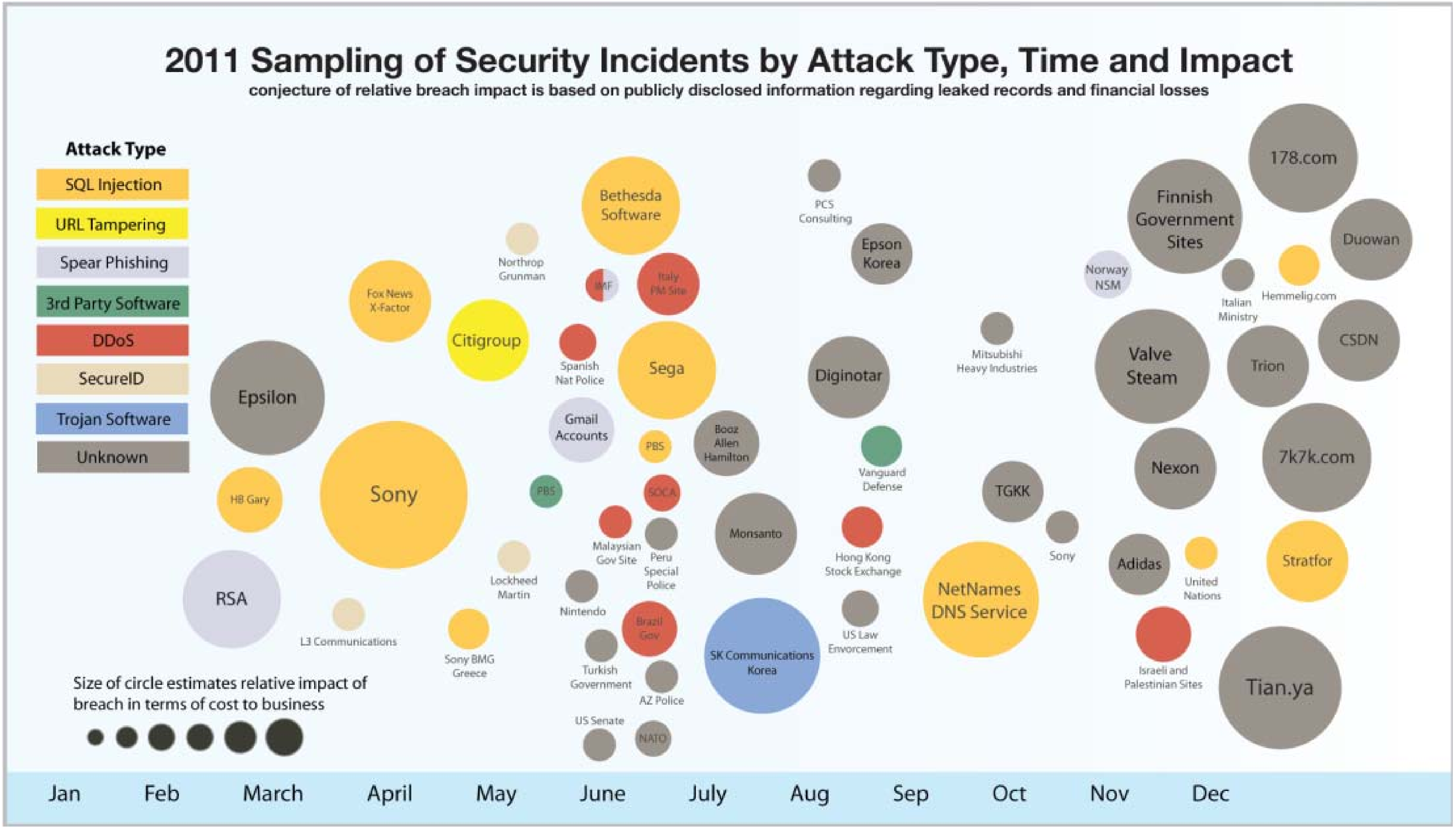
Can you Circumvent this Security Control?



Agenda

- Enterprise application security landscape
- Finding application security vulnerabilities
- Building products that are secure by design
- Bridging the Security/Development gap
- IBM Security AppScan Solutions

Security Incidents in 2011

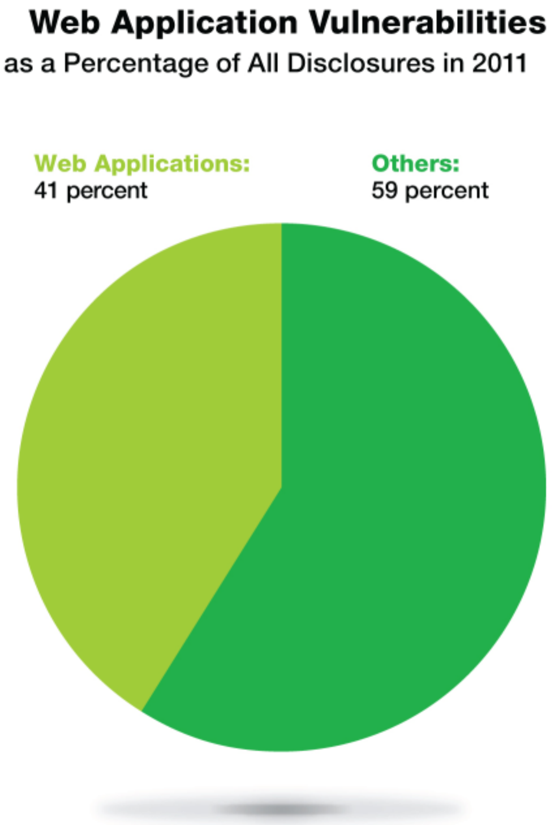


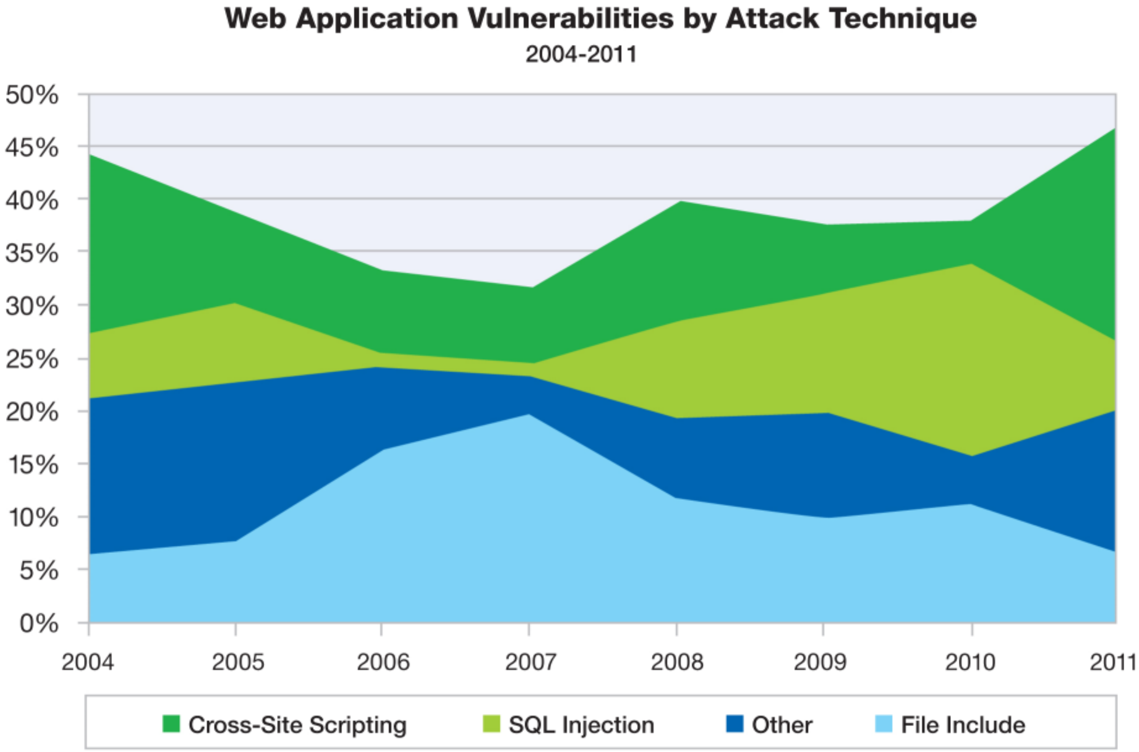
2011 Sampling of Security Incidents by Attack Type, Time and Impact

conjecture of relative breach impact is based on publicly disclosed information regarding leaked records and financial losses

Source: IBM X-Force 2011 Trend & Risk Report

Application Security Landscape





Source: IBM X-Force 2011 Trend & Risk Report

Enterprise Security Applies to Applications

- **New business models require new interfaces to existing business functions**
- **Enterprise Systems are now accessed in new ways**
  - Web applications and web services hosted on z/OS
  - Web services linked directly to transactions on z/OS
  - Business-to-business communications

- **And from more than console-based user interfaces**
  - Mobile applications
  - Web browser-based user interfaces
  - Rich client applications using RESTful services
- **User authentication/authorization and resource protection are required but not sufficient**
  - Need to also protect data accessed by applications and services by ensuring they are free from security vulnerabilities

➔Application Security testing and analysis applies across all Enterprise systems!

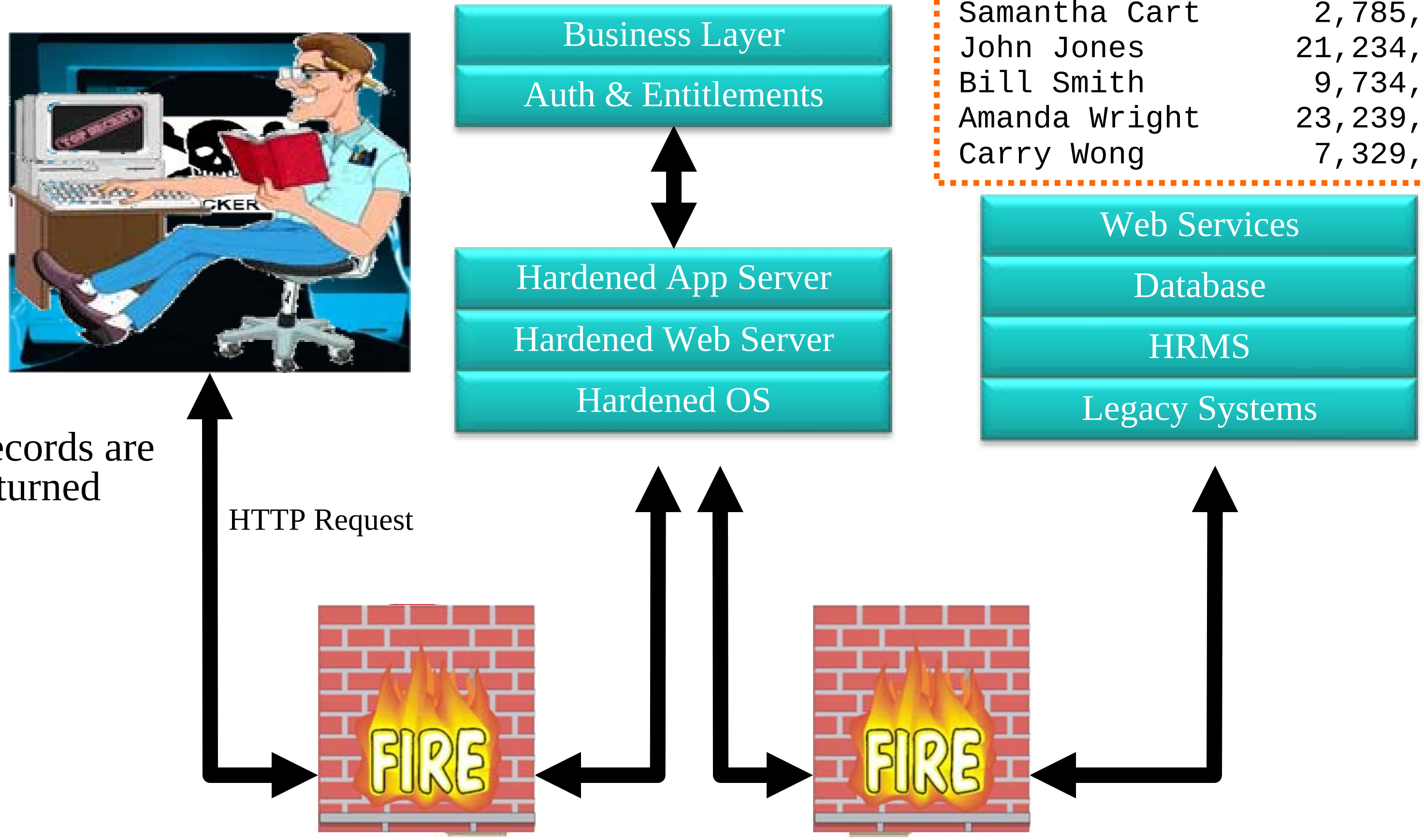
SQL Injection Illustrated

```
Select * from Account where acct = ' + acctNum + '
```

```
Select * from Account where acct = '876398' or '1' = '1'
```

Account: 876398' or '1' = '1

```
Name                 Balance
-------------        -------------
Samantha Cart          2,785,294
John Jones            21,234,345
Bill Smith             9,734,123
Amanda Wright         23,239,329
Carry Wong             7,329,011
```



Business Layer

Auth & Entitlements

Hardened App Server

Hardened Web Server

Hardened OS

Web Services

Database

HRMS

Legacy Systems

All records are returned

HTTP Request

Intrusion detection, firewalls, and hardened infrastructure won't detect or prevent most application level attacks

Solving Customer Challenges

Find the vulnerabilities

Leverage advanced and comprehensive testing methodologies

Build products that are secure by design

Reduce costs by integrating security testing early in the development lifecycle

Bridge the Security/Development gap

Engage Security and Development organizations to collaboratively address application vulnerabilities

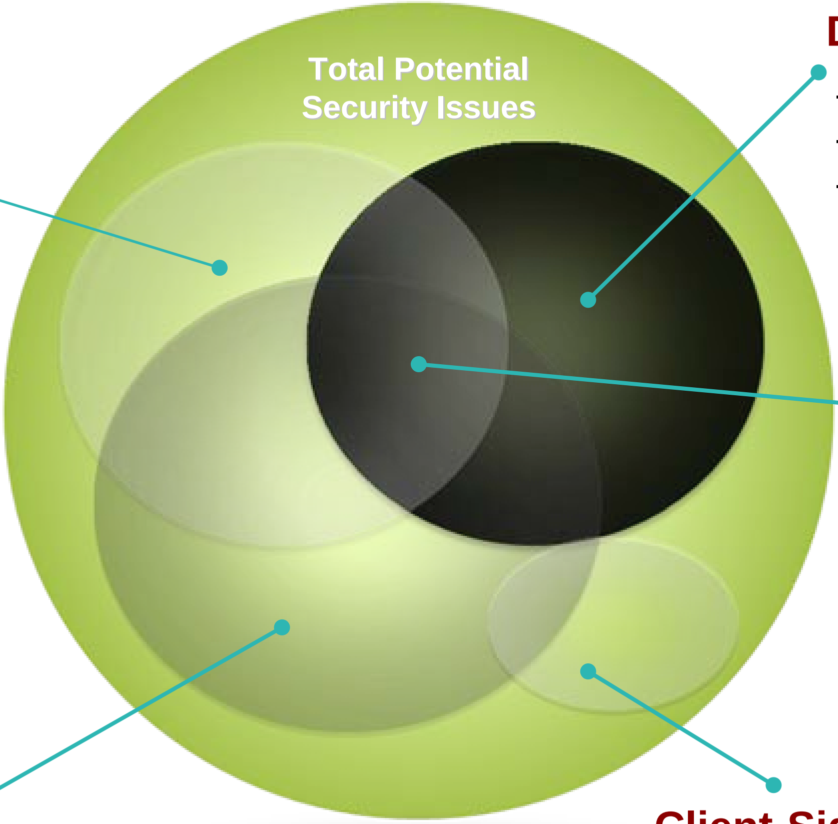
Agenda

- Enterprise application security landscape
- Finding application security vulnerabilities
- Building products that are secure by design
- Bridging the Security/Development gap
- IBM Security AppScan Solutions

Primary App Security Testing Techniques

	Static Analysis	Dynamic Analysis
Scan input	Source code	Live web application
Assessment Techniques	Taint analysis & pattern matching	Tampering with HTTP messages
Where does it fit in the SDLC	Application development	Anywhere in the SDLC where you have a live app (dev, QA, deployment)
Results and output	Results are presented by line of code	Results are presented as HTTP messages (exploit requests)

Advanced App Security Testing Techniques



Static Analysis

- Analyze Source Code
- Use during development
- Uses Taint Analysis / Pattern Matching

Dynamic Analysis

- Analyze Live Web Application
- Use during testing
- Uses HTTP tampering

Hybrid Analysis

- Correlate Dynamic and Static results
- Assists remediation by identification of line of code

Run-Time Analysis

- Combines Dynamic Analysis with run-time agent
- More results, better accuracy

New!

Client-Side Analysis

- Analyze downloaded JavaScript code which runs in client
- Unique in the industry

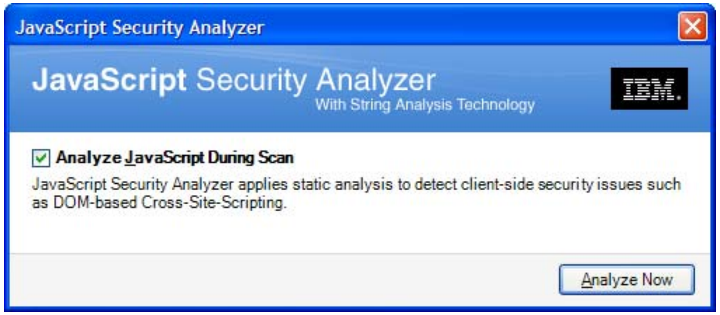
New!

Client-Side Analysis: JavaScript Security Analyzer

What is JSA?

- JavaScript Security Analyzer – An extension of AppScan Standard, developed in collaboration with IBM Research, that does **static taint analysis of JavaScript**, detecting a range of client-side security issues:

•DOM Based Cross-Site Scripting	*•Web Worker Script URL Manipulation*
•Phishing Through URL Redirection	*•Notification Phishing*
•Email Attribute Spoofing	*•Client-Side Stored Cross-Site Scripting*

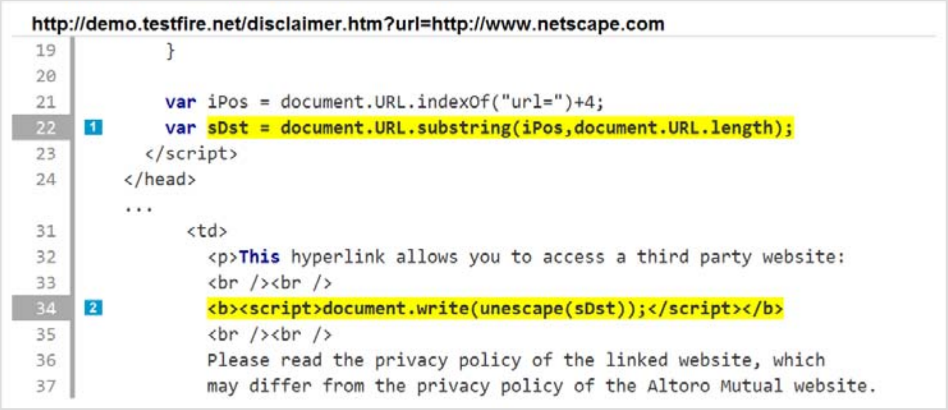

JavaScript Security Analyzer

JavaScript Security Analyzer
With String Analysis Technology

☑ **Analyze JavaScript During Scan**
JavaScript Security Analyzer applies static analysis to detect client-side security issues such as DOM-based Cross-Site-Scripting.

Why is this significant?

- The role of JavaScript in modern web applications becomes greater as technologies such as AJAX, Dojo and HTML5 become more prolific.
- JSA completes a missing piece in scanning modern web applications. JSX provides an answer for crawling, JSA provides an answer for testing. In the future we see great potential for **synergy between JSX and JSA**.

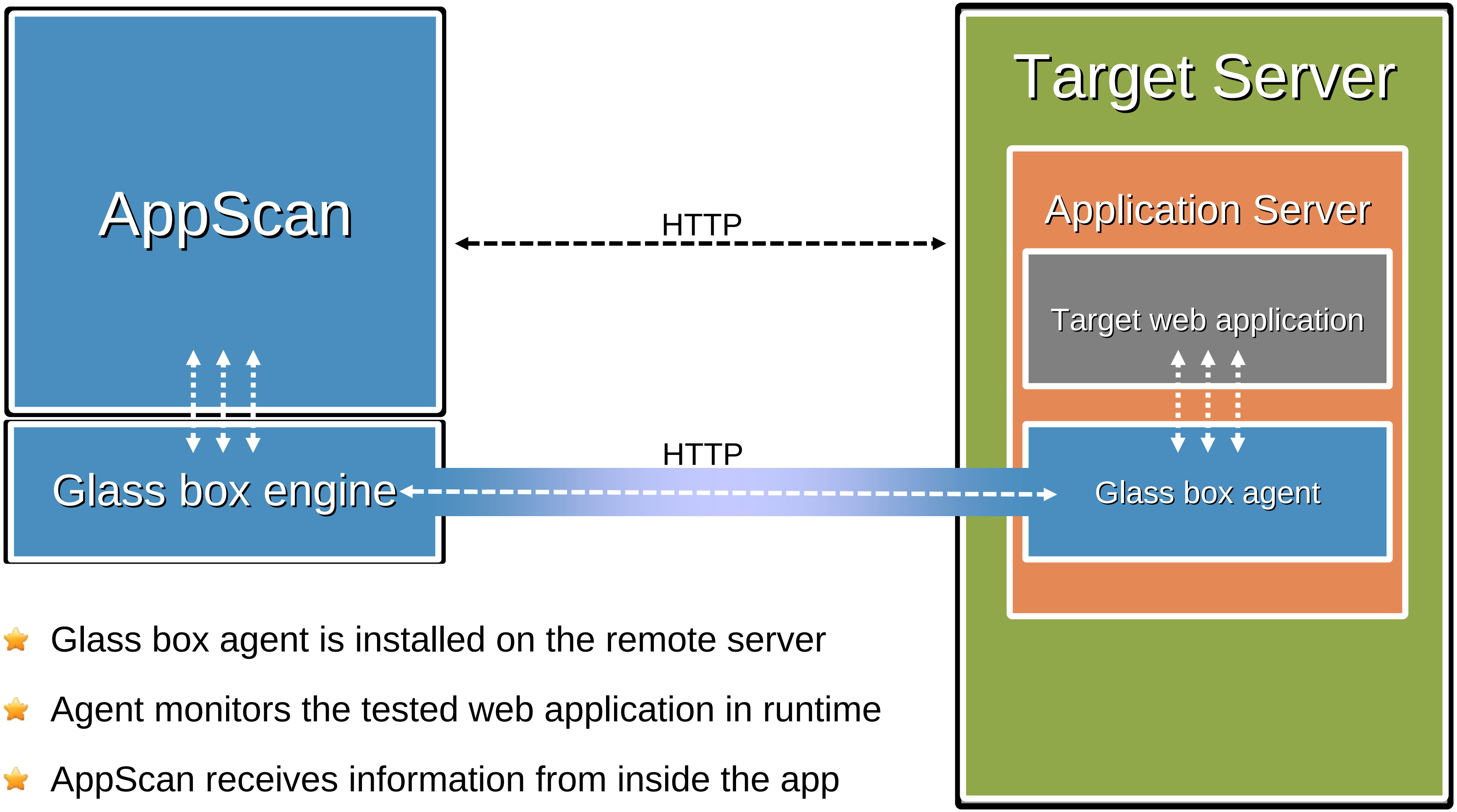

http://demo.testfire.net/disclaimer.htm?url=http://www.netscape.com

```
          }

          var iPos = document.URL.indexOf("url=")+4;
1         var sDst = document.URL.substring(iPos,document.URL.length);
        </script>
      </head>
      ...
            <td>
              <p>This hyperlink allows you to access a third party website:
              <br /><br />
2             <b><script>document.write(unescape(sDst));</script></b>
              <br /><br />
              Please read the privacy policy of the linked website, which
              may differ from the privacy policy of the Altoro Mutual website.
```

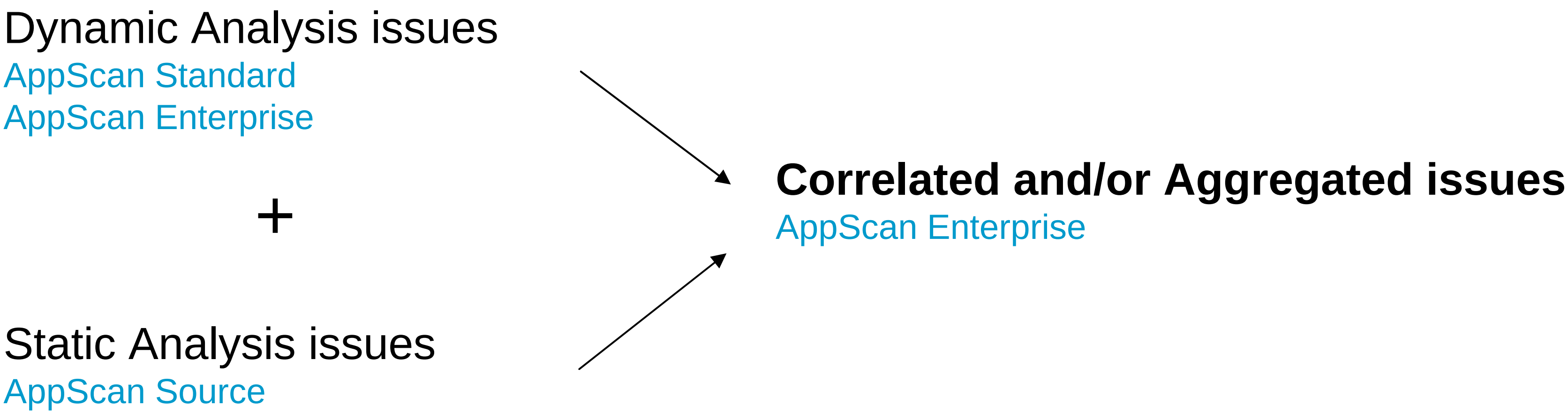
Example of trace provided by JSA

Run-Time Analysis: Glass Box Scanning



- Glass box agent is installed on the remote server
- Agent monitors the tested web application in runtime
- AppScan receives information from inside the app

Correlation of Static & Dynamic Analysis Issues

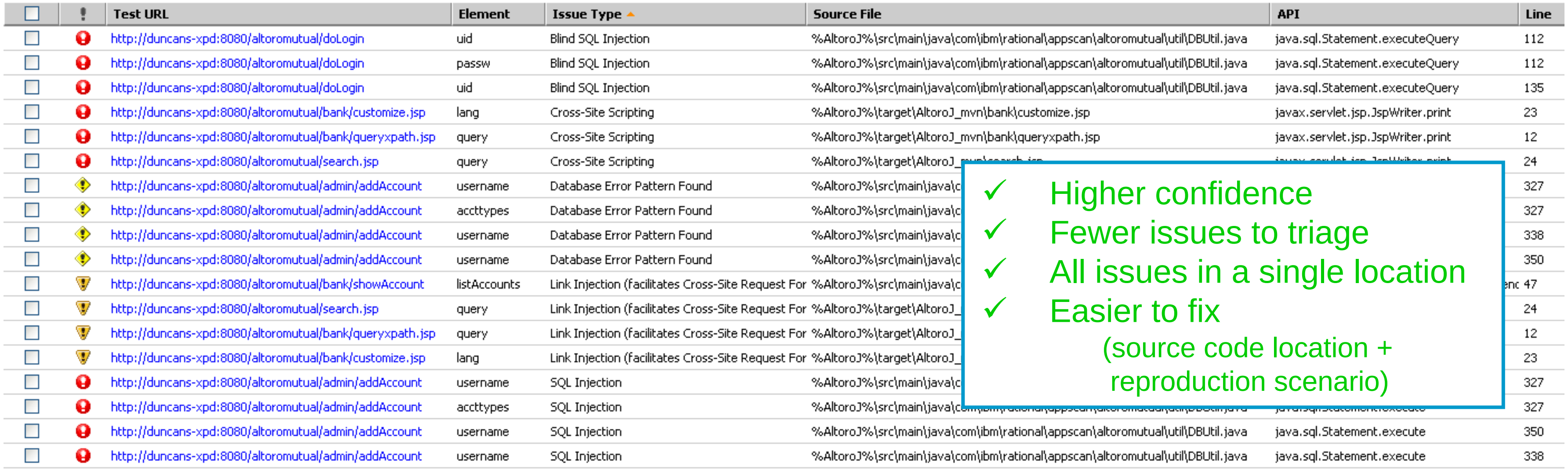


Dynamic Analysis issues
AppScan Standard
AppScan Enterprise

+

Static Analysis issues
AppScan Source

Correlated and/or Aggregated issues
AppScan Enterprise



□	!	Test URL	Element	Issue Type ▲	Source File	API	Line
□	●	http://duncans-xpd:8080/altoromutual/doLogin	uid	Blind SQL Injection	%AltoroJ%\src\main\java\com\ibm\rational\appscan\altoromutual\util\DBUtil.java	java.sql.Statement.executeQuery	112
□	●	http://duncans-xpd:8080/altoromutual/doLogin	passw	Blind SQL Injection	%AltoroJ%\src\main\java\com\ibm\rational\appscan\altoromutual\util\DBUtil.java	java.sql.Statement.executeQuery	112
□	●	http://duncans-xpd:8080/altoromutual/doLogin	uid	Blind SQL Injection	%AltoroJ%\src\main\java\com\ibm\rational\appscan\altoromutual\util\DBUtil.java	java.sql.Statement.executeQuery	135
□	●	http://duncans-xpd:8080/altoromutual/bank/customize.jsp	lang	Cross-Site Scripting	%AltoroJ%\target\AltoroJ_mvn\bank\customize.jsp	javax.servlet.jsp.JspWriter.print	23
□	●	http://duncans-xpd:8080/altoromutual/bank/queryxpath.jsp	query	Cross-Site Scripting	%AltoroJ%\target\AltoroJ_mvn\bank\queryxpath.jsp	javax.servlet.jsp.JspWriter.print	12
□	●	http://duncans-xpd:8080/altoromutual/search.jsp	query	Cross-Site Scripting	%AltoroJ%\target\AltoroJ_mvn\search.jsp	javax.servlet.jsp.JspWriter.print	24
□	◆	http://duncans-xpd:8080/altoromutual/admin/addAccount	username	Database Error Pattern Found	%AltoroJ%\src\main\java\c		327
□	◆	http://duncans-xpd:8080/altoromutual/admin/addAccount	accttypes	Database Error Pattern Found	%AltoroJ%\src\main\java\c		327
□	◆	http://duncans-xpd:8080/altoromutual/admin/addAccount	username	Database Error Pattern Found	%AltoroJ%\src\main\java\c		338
□	◆	http://duncans-xpd:8080/altoromutual/admin/addAccount	username	Database Error Pattern Found	%AltoroJ%\src\main\java\c		350
□	▼	http://duncans-xpd:8080/altoromutual/bank/showAccount	listAccounts	Link Injection (facilitates Cross-Site Request For	%AltoroJ%\src\main\java\c	enc	47
□	▼	http://duncans-xpd:8080/altoromutual/search.jsp	query	Link Injection (facilitates Cross-Site Request For	%AltoroJ%\target\AltoroJ_		24
□	▼	http://duncans-xpd:8080/altoromutual/bank/queryxpath.jsp	query	Link Injection (facilitates Cross-Site Request For	%AltoroJ%\target\AltoroJ_		12
□	▼	http://duncans-xpd:8080/altoromutual/bank/customize.jsp	lang	Link Injection (facilitates Cross-Site Request For	%AltoroJ%\target\AltoroJ_		23
□	●	http://duncans-xpd:8080/altoromutual/admin/addAccount	username	SQL Injection	%AltoroJ%\src\main\java\c		327
□	●	http://duncans-xpd:8080/altoromutual/admin/addAccount	accttypes	SQL Injection	%AltoroJ%\src\main\java\com\ibm\rational\appscan\altoromutual\util\DBUtil.java	java.sql.Statement.execute	327
□	●	http://duncans-xpd:8080/altoromutual/admin/addAccount	username	SQL Injection	%AltoroJ%\src\main\java\com\ibm\rational\appscan\altoromutual\util\DBUtil.java	java.sql.Statement.execute	350
□	●	http://duncans-xpd:8080/altoromutual/admin/addAccount	username	SQL Injection	%AltoroJ%\src\main\java\com\ibm\rational\appscan\altoromutual\util\DBUtil.java	java.sql.Statement.execute	338

- ✓ Higher confidence
- ✓ Fewer issues to triage
- ✓ All issues in a single location
- ✓ Easier to fix
  (source code location +
  reproduction scenario)

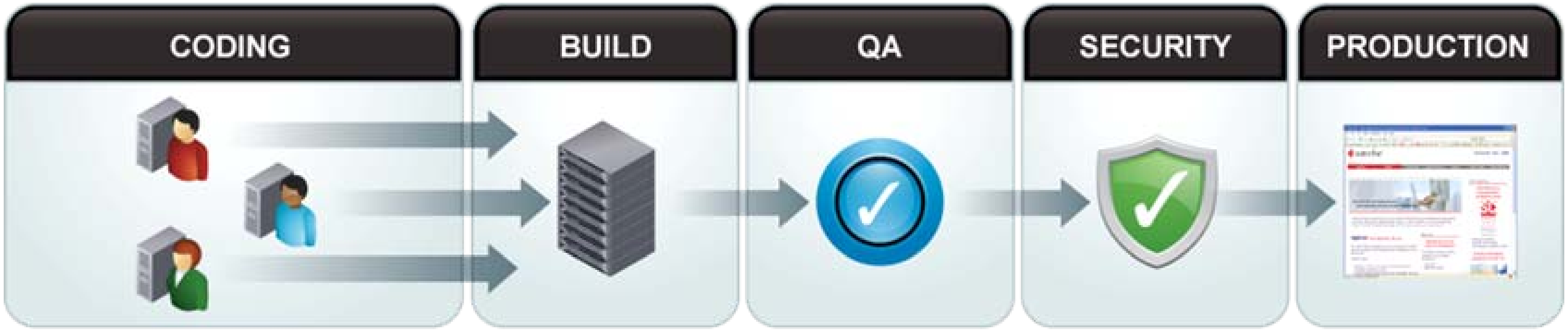
Agenda

- Enterprise application security landscape
- Finding application security vulnerabilities
- Building products that are secure by design
- Bridging the Security/Development gap
- IBM Security AppScan Solutions

Reduce Costs Through a Secure by Design Approach

*80% of development costs are spent identifying and correcting defects!**

Average Cost of a Data Breach
$7.2M** from law suits, loss of customer trust, damage to brand



Find during Development
$80/defect

Find during Build
$240/defect

Find during QA/Test
$960/defect

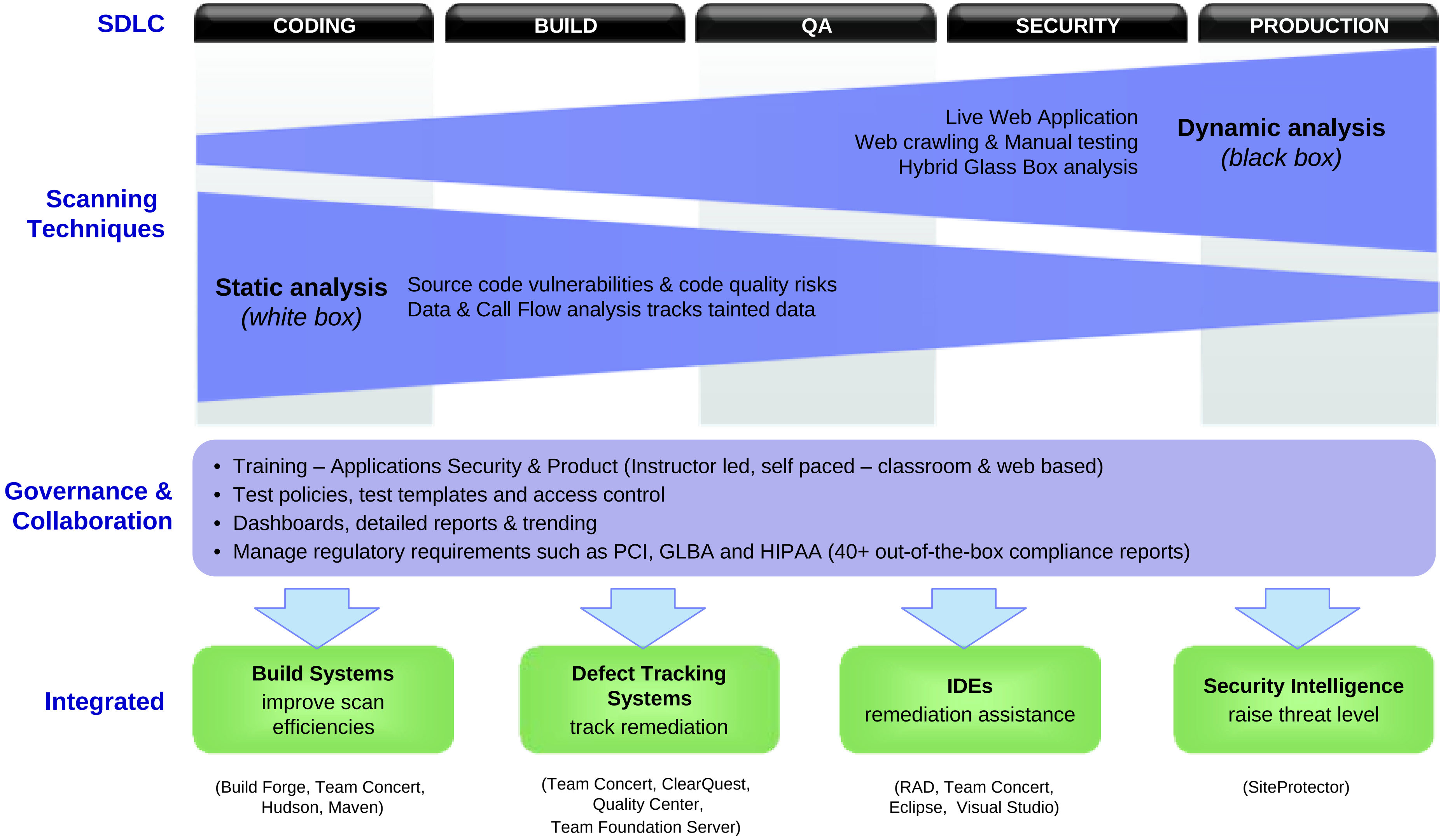
Find in Production
$7,600 / defect

> *"As financially-motivated attackers have shifted their focus to applications, Web application security has become a top priority. However, the responsibility for web application security cannot rest solely with information security. Enterprises should evaluate how to identify vulnerabilities in Web applications earlier in the development process as transparently as possible using web application security testing products or services."*
>
> *Neil MacDonald, Gartner, 12-6-11*

* Source: National Institute of Standards and Technology

** Source: Ponemon Institute 2009-10

Introduce Security Testing Early in the SDLC



SDLC

CODING | BUILD | QA | SECURITY | PRODUCTION

Scanning Techniques

Live Web Application
Web crawling & Manual testing
Hybrid Glass Box analysis

Dynamic analysis
(black box)

Static analysis
(white box)

Source code vulnerabilities & code quality risks
Data & Call Flow analysis tracks tainted data

Governance & Collaboration

- Training – Applications Security & Product (Instructor led, self paced – classroom & web based)
- Test policies, test templates and access control
- Dashboards, detailed reports & trending
- Manage regulatory requirements such as PCI, GLBA and HIPAA (40+ out-of-the-box compliance reports)

Integrated

Build Systems
improve scan efficiencies

(Build Forge, Team Concert, Hudson, Maven)

Defect Tracking Systems
track remediation

(Team Concert, ClearQuest, Quality Center, Team Foundation Server)

IDEs
remediation assistance

(RAD, Team Concert, Eclipse, Visual Studio)

Security Intelligence
raise threat level

(SiteProtector)

Make Applications Secure By Design

Cycle of secure application development

Requirements & Design

- Consider security requirements of the application & apply threat models
- Issues such as required controls and best practices are documented on par with functional requirements
- Secure code libraries maintained for reusable secure code

Development

- Create work items that map to security requirements
- Use secure code libraries
- Software is checked during coding for:
  - Implementation error vulnerabilities
  - Compliance with security requirements

Build & Test

- Map test plan to security requirements
- Testing begins for errors and compliance with security requirements across the entire application
- Applications are also tested for exploitability in deployment scenario

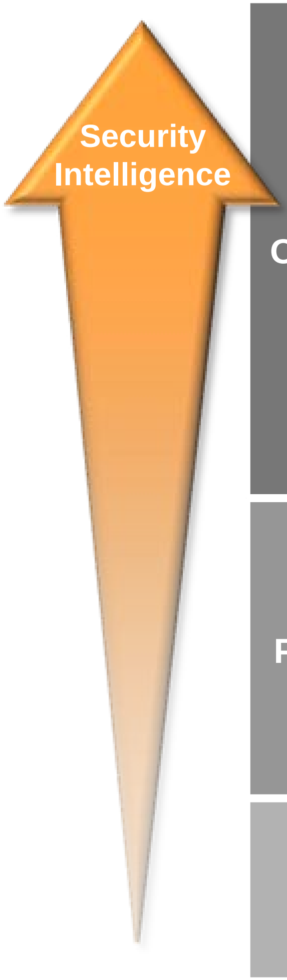
Deployment

- Configure infrastructure for application policies
- Deploy applications into production

Operational

- Continuously monitor applications for appropriate application usage, vulnerabilities and defend against attacks

Helping Organizations Progress in Their Security Maturity



	People	Data	Applications	Infrastructure
	Security Intelligence: Information and event management Advanced correlation and deep analytics External threat research			
Optimized	Role based analytics Identity governance Privileged user controls	Data flow analytics Data governance	Secure app engineering processes Fraud detection	Advanced network monitoring Forensics / data mining Secure systems
Proficient	User provisioning Access mgmt Strong authentication	Access monitoring Data loss prevention	Application firewall Source code scanning	Virtualization security Asset mgmt Endpoint / network security management
Basic	Centralized directory	Encryption Access control	Application scanning	Perimeter security Anti-virus

Agenda

- Enterprise application security landscape
- Finding application security vulnerabilities
- Building products that are secure by design
- Bridging the Security/Development gap
- IBM Security AppScan Solutions

Application security challenge: Security-Development disconnect fails to prevent vulnerabilities in production applications

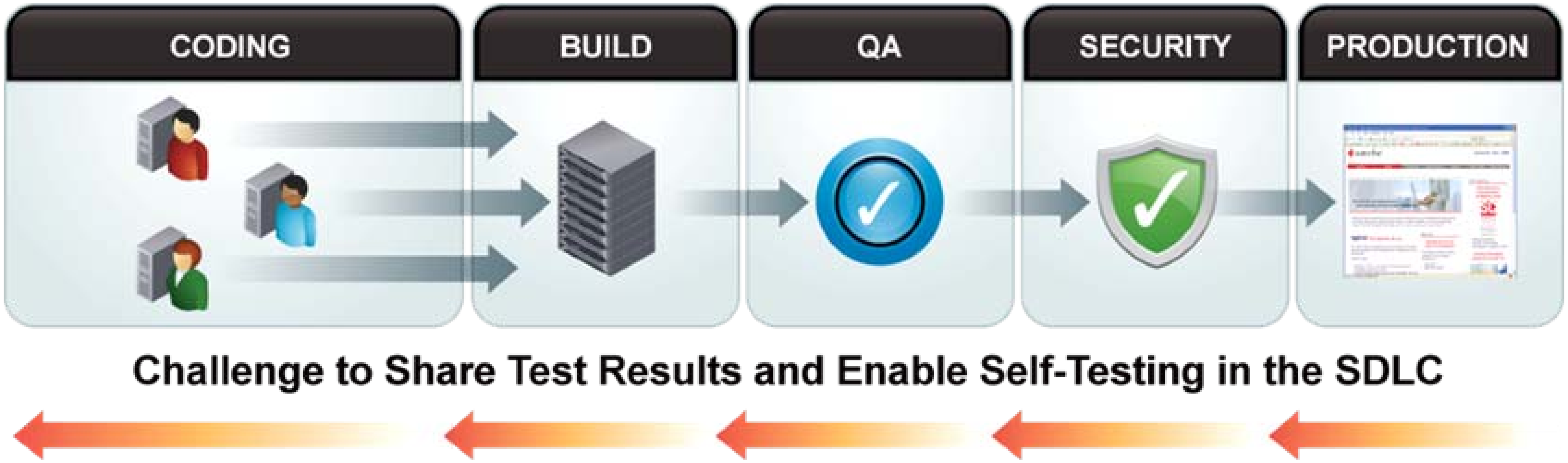
Developers Lack Security Insights

(or Incentives to Address Security)

- Mandate to deliver functionality on-time and on-budget – but not to develop secure applications
- Developers rarely educated in secure code practices
- Product innovation drives development of increasingly complicated applications

Security Team = SDLC Bottleneck

- Security tests executed just before launch
  - Adds time and cost to fix vulnerabilities late in the process
- Growing number of web applications but small security staff
  - Most enterprises scan ~10% of all applications
- Continuous monitoring of production apps limited or non-existent
  - Unidentified vulnerabilities & risk

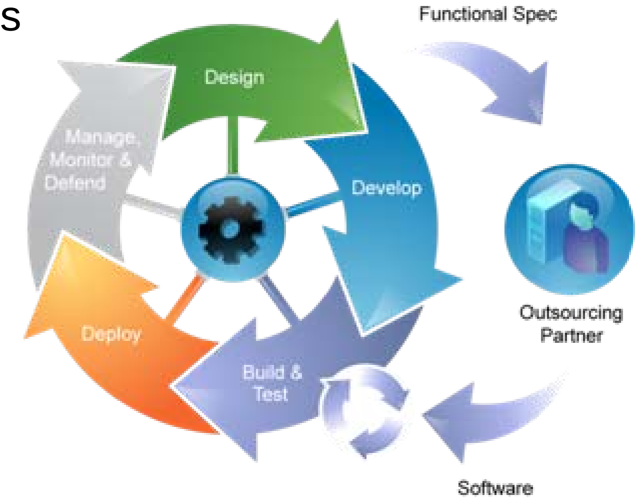


Challenge to Share Test Results and Enable Self-Testing in the SDLC

Bridge the Security/Development gap

Break down organizational silos

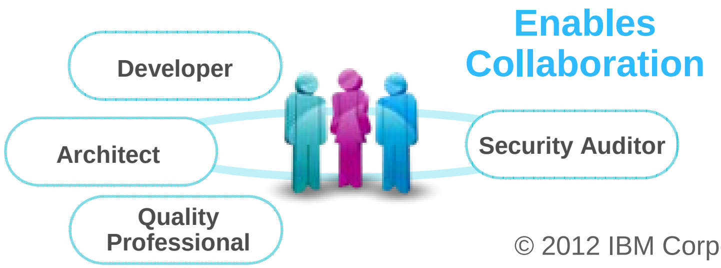
- Security experts establish security testing policies
- Development teams test early in the cycle
- Treat vulnerabilities as development defects



Provide Management Visibility

- Dashboard of application risk
- Enable compliance with regulation-specific reporting

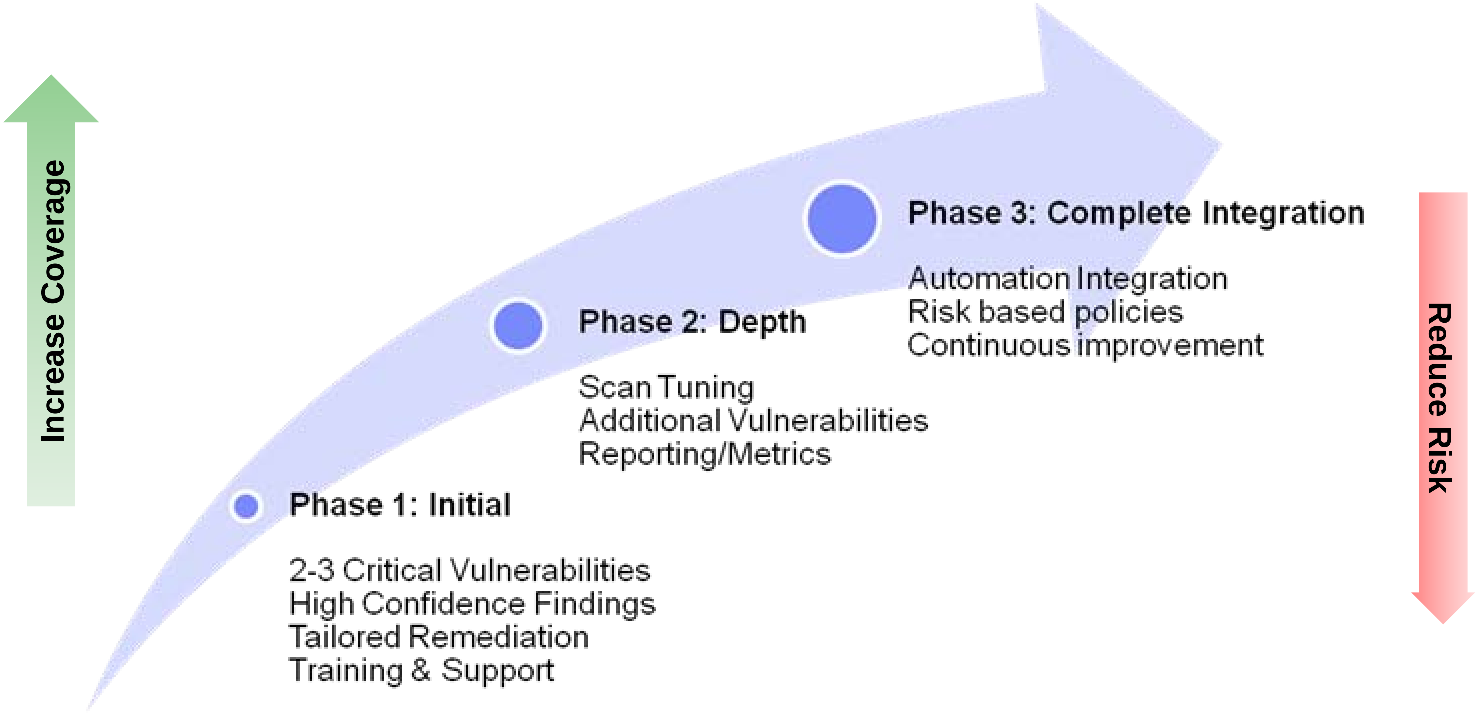
> *"… we wanted to go to a multiuser web-based solution that enabled us to do concurrent scans and provide our customers with a web-based portal for accessing and sharing information on identified issues."*
>
> *Alex Jalso, Asst Dir, Office of InfoSecurity, WVU*



Enables Collaboration

Security Policy and Scanning Rollout

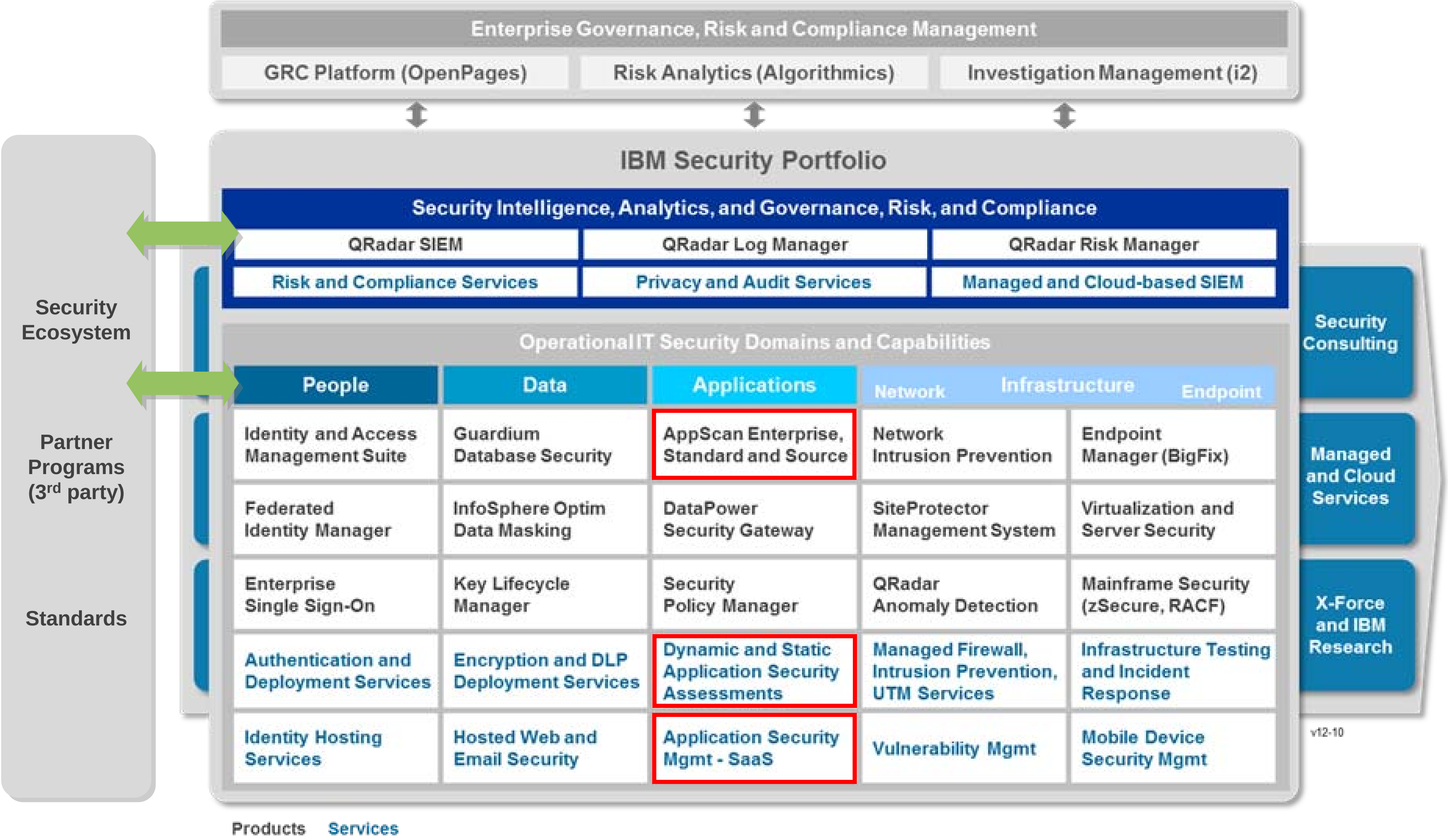
- Defines vulnerabilities that are critical to the business
- Consider application security as a whole: organization, tools, resources, support, and training
- Limit the scope of the initial implementation, but plan for the complete portfolio
- Build capabilities through a sequence of steps that build upon each other
- Communicate realistic expectations about how and when business benefits accrue



Agenda

- Enterprise application security landscape
- Finding application security vulnerabilities
- Building products that are secure by design
- Bridging the Security/Development gap
- IBM Security AppScan Solutions

IBM’s security product and service portfolio



Enterprise Governance, Risk and Compliance Management

GRC Platform (OpenPages)	Risk Analytics (Algorithmics)	Investigation Management (i2)

Security Ecosystem

Partner Programs (3rd party)

Standards

IBM Security Portfolio

Security Intelligence, Analytics, and Governance, Risk, and Compliance

QRadar SIEM	QRadar Log Manager	QRadar Risk Manager
Risk and Compliance Services	Privacy and Audit Services	Managed and Cloud-based SIEM

Operational IT Security Domains and Capabilities

People	Data	Applications	Infrastructure	
			Network	Endpoint
Identity and Access Management Suite	Guardium Database Security	AppScan Enterprise, Standard and Source	Network Intrusion Prevention	Endpoint Manager (BigFix)
Federated Identity Manager	InfoSphere Optim Data Masking	DataPower Security Gateway	SiteProtector Management System	Virtualization and Server Security
Enterprise Single Sign-On	Key Lifecycle Manager	Security Policy Manager	QRadar Anomaly Detection	Mainframe Security (zSecure, RACF)
Authentication and Deployment Services	Encryption and DLP Deployment Services	Dynamic and Static Application Security Assessments	Managed Firewall, Intrusion Prevention, UTM Services	Infrastructure Testing and Incident Response
Identity Hosting Services	Hosted Web and Email Security	Application Security Mgmt - SaaS	Vulnerability Mgmt	Mobile Device Security Mgmt

Security Consulting

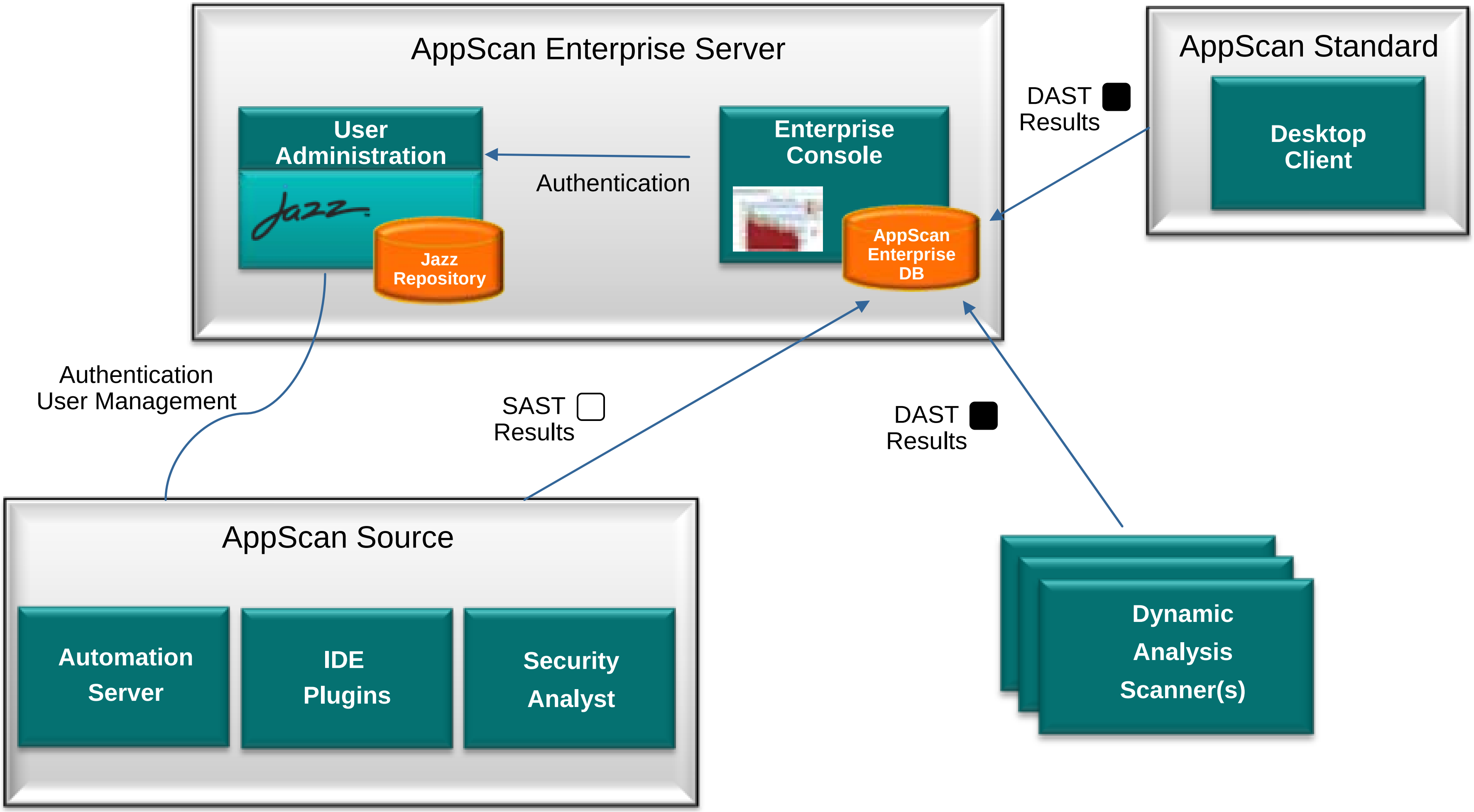
Managed and Cloud Services

X-Force and IBM Research

v12-10

Products Services

AppScan Portfolio



AppScan Enterprise

Scalability & Control

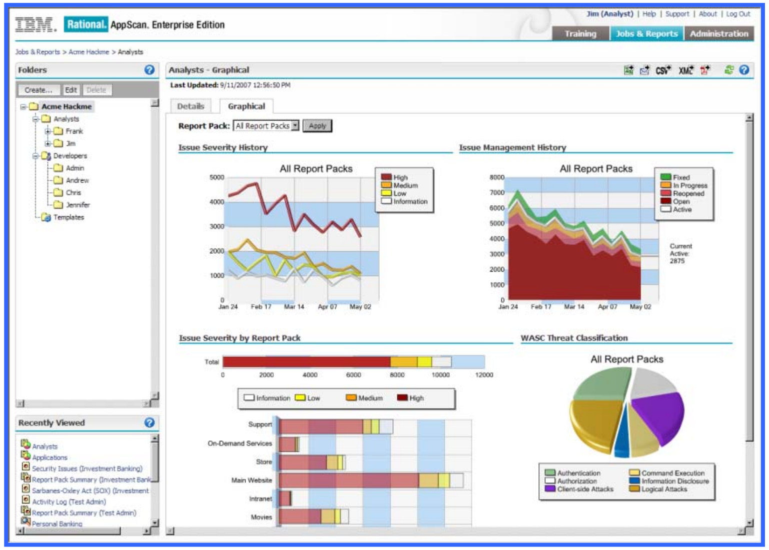
- Schedule and execute 1000s of application assessments
- Empower non-security experts to implement security best practices
- Manage user roles and access to reports
- Define scan permissions, policies and templates

Visibility & Compliance

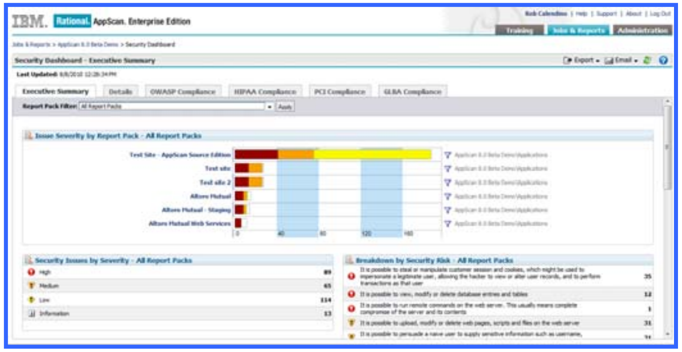
- Dashboards of application security risk
- Non-compliance risk (40+ compliance reports)

Measure & Improve

- Number of issues by severity
- Top security issues and risks
- Trending of issues over time



Last Updated: 9/11/2007 12:56:50 PM


IBM. Rational. AppScan. Enterprise Edition
Bob Calendino | Help | Support | About | Log Out
Training
Jobs & Reports
Administration
Jobs & Reports > AppScan 8.0 Beta Demo > Security Dashboard
Security Dashboard - Executive Summary
Export
Email
Executive Summary
Details
OWASP Compliance
HIPAA Compliance
PCI Compliance
GLBA Compliance
Report Pack Filter: All Report Packs
Apply
Issue Severity by Report Pack - All Report Packs
Test Site - AppScan Source Edition
Test site
Test site 2
Altoro Mutual
Altoro Mutual - Staging
Altoro Mutual Web Services
AppScan 8.0 Beta Demo\Applications
0
40
80
120
160

Last Updated: 8/8/2010 12:28:34 PM

Security Issues by Severity - All Report Packs

High	89
Medium	65
Low	114
Information	13

Breakdown by Security Risk - All Report Packs

It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user	35
It is possible to view, modify or delete database entries and tables	12
It is possible to run remote commands on the web server. This usually means complete compromise of the server and its contents	1
It is possible to upload, modify or delete web pages, scripts and files on the web server	31
It is possible to persuade a naive user to supply sensitive information such as username,	31

AppScan Source: Static analysis (white box) security & quality testing in the collaborative application lifecycle

Source Code Analysis for Security Testing in Development & Build Automation

Static Analysis (White box)

- **Covers OWASP, SANS, & WASC threat classes**
  - SQL Injection
  - Cross-Site Scripting
  - Exposed Credentials
  - OS Command Injection
  - LDAP Injection
  - XPath Injection
  - Buffer Overflows
  - URL Redirect
  - ***Many more***

- **Language Support**
  - Java, JSP
  - C, C++
  - Classic ASP (VB6)
  - C#, VB.NET, ASP.NET
  - COBOL
  - SAP ABAP*
  - PHP
  - PERL
  - ColdFusion
  - Client-Side JavaScript
  - Server-Side JavaScript
  - VBScript
  - PL/SQL
  - T-SQL

- **Framework Support**
  - Struts
  - Spring MVC
  - EJB
  - .NET
  - New additions in future releases
  - Extensible

Code Quality

- Identify code-level quality defects within IDE
- Automate code quality analysis as part of the build process for centralized software code scanning
- Key Performance Indicators (KPIs) to help developers learn best practices
- Languages: Java, C, C++

Application Lifecycle Integrations

Develop

- IDE plug-ins to remediate identified issues (*Source for Remediation*)
- Options to scan code locally from IDE (*Source for Developer*)

Build

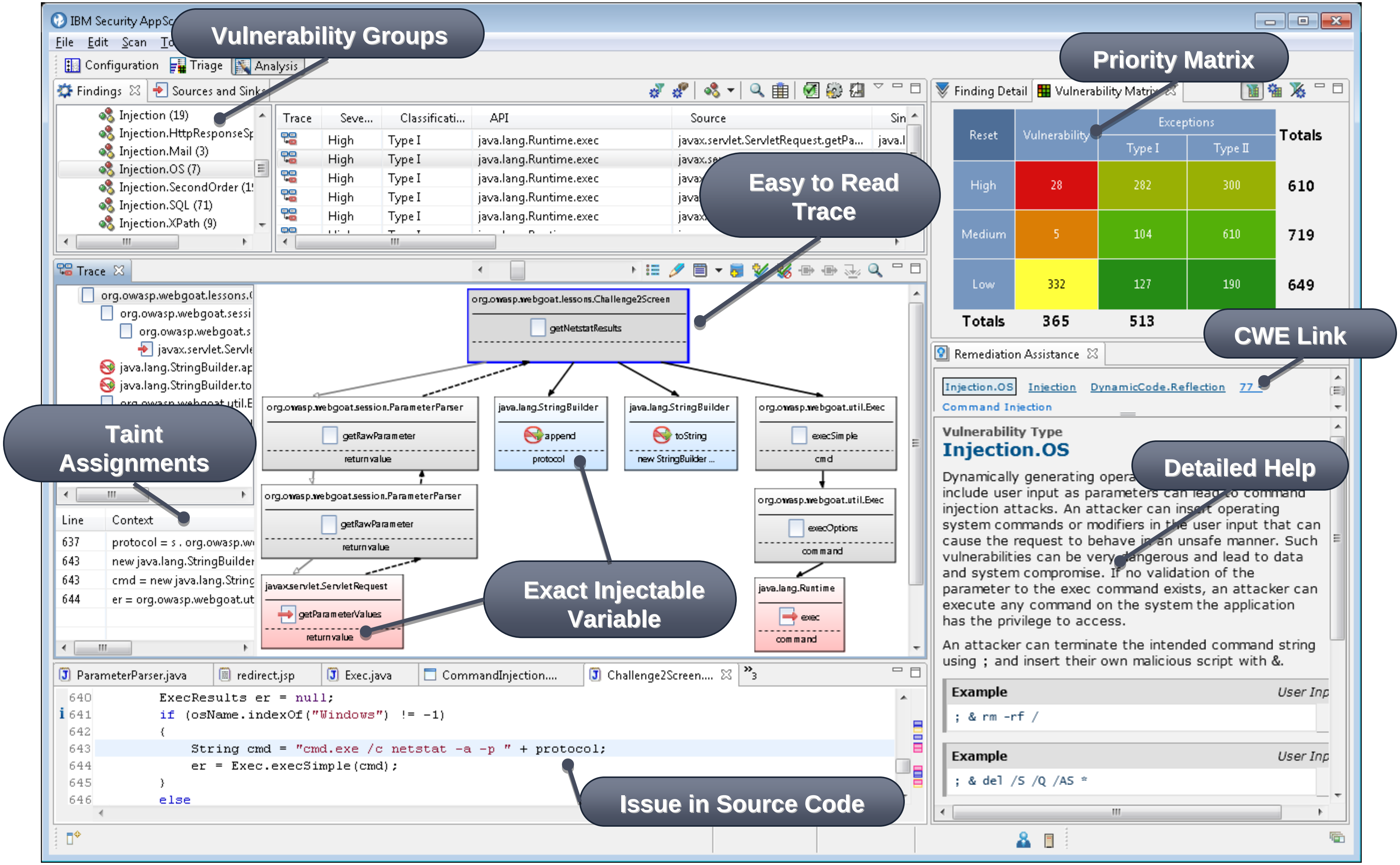
- Automatically trigger security scans with each build (*Source for Automation*)
- Review results from IDE or Security user & create work items for remediation

Security

- Power user creates SAST scans executed from IDE or in build automation
- Executes advanced scans in pre-production security audits

* Requires Virtual Forge CodeProfiler for AppScan Source Edition

AppScan Source Edition Finding View



Vulnerability Groups

Priority Matrix

Easy to Read Trace

CWE Link

Taint Assignments

Detailed Help

Exact Injectable Variable

Issue in Source Code

- Injection (19)
- Injection.HttpResponseSp
- Injection.Mail (3)
- Injection.OS (7)
- Injection.SecondOrder (1!
- Injection.SQL (71)
- Injection.XPath (9)

Trace	Seve...	Classificati...	API	Source	Sin
	High	Type I	java.lang.Runtime.exec	javax.servlet.ServletRequest.getPa...	java.l
	High	Type I	java.lang.Runtime.exec	javax.se	
	High	Type I	java.lang.Runtime.exec	javax	
	High	Type I	java.lang.Runtime.exec	java	
	High	Type I	java.lang.Runtime.exec	javax	

Reset	Vulnerability	Exceptions		Totals
		Type I	Type II	
High	28	282	300	610
Medium	5	104	610	719
Low	332	127	190	649
Totals	365	513		

- org.owasp.webgoat.lessons.(
  - org.owasp.webgoat.sessi
    - org.owasp.webgoat.s
      - javax.servlet.Servle
  - java.lang.StringBuilder.ap
  - java.lang.StringBuilder.to
  - org.owasp.webgoat.util.E

Line	Context
637	protocol = s . org.owasp.we
643	new java.lang.StringBuilder
643	cmd = new java.lang.String
644	er = org.owasp.webgoat.ut

```
        ExecResults er = null;
        if (osName.indexOf("Windows") != -1)
        {
            String cmd = "cmd.exe /c netstat -a -p " + protocol;
            er = Exec.execSimple(cmd);
        }
        else
```

Injection.OS   Injection   DynamicCode.Reflection   77
Command Injection

Vulnerability Type

Injection.OS

Dynamically generating opera include user input as parameters can lead to command injection attacks. An attacker can insert operating system commands or modifiers in the user input that can cause the request to behave in an unsafe manner. Such vulnerabilities can be very dangerous and lead to data and system compromise. If no validation of the parameter to the exec command exists, an attacker can execute any command on the system the application has the privilege to access.

An attacker can terminate the intended command string using ; and insert their own malicious script with &.

Example *User Inp*

```
; & rm -rf /
```

Example *User Inp*

```
; & del /S /Q /AS *
```

AppScan Standard: Desktop solution combines advanced security testing, broad technology coverage and ease of use

Web Application Assessments for Pen-Testers and Security Practitioners

Dynamic Analysis (black box)

- **Covers all relevant OWASP & WASC TCv2 threat classes**
  - SQL Injection
  - Cross-Site Scripting
  - HTTP Response Splitting
  - OS Commanding
  - LDAP Injection
  - XPath Injection
  - Buffer Overflows
  - ***1000s more***

- **Web 2.0 and Rich Internet Applications**
  - JavaScript & Ajax
  - Adobe Flash & Flex
- **Malware analysis**
  - Scan site with malware analysis from IBM X-Force Security Research

- **Web Services/ SOA**
  - SOAP/XML parser issues (External entities, XML blowup, etc.)
  - Application-layer issues
  - Infrastructure issues

Ease of Use

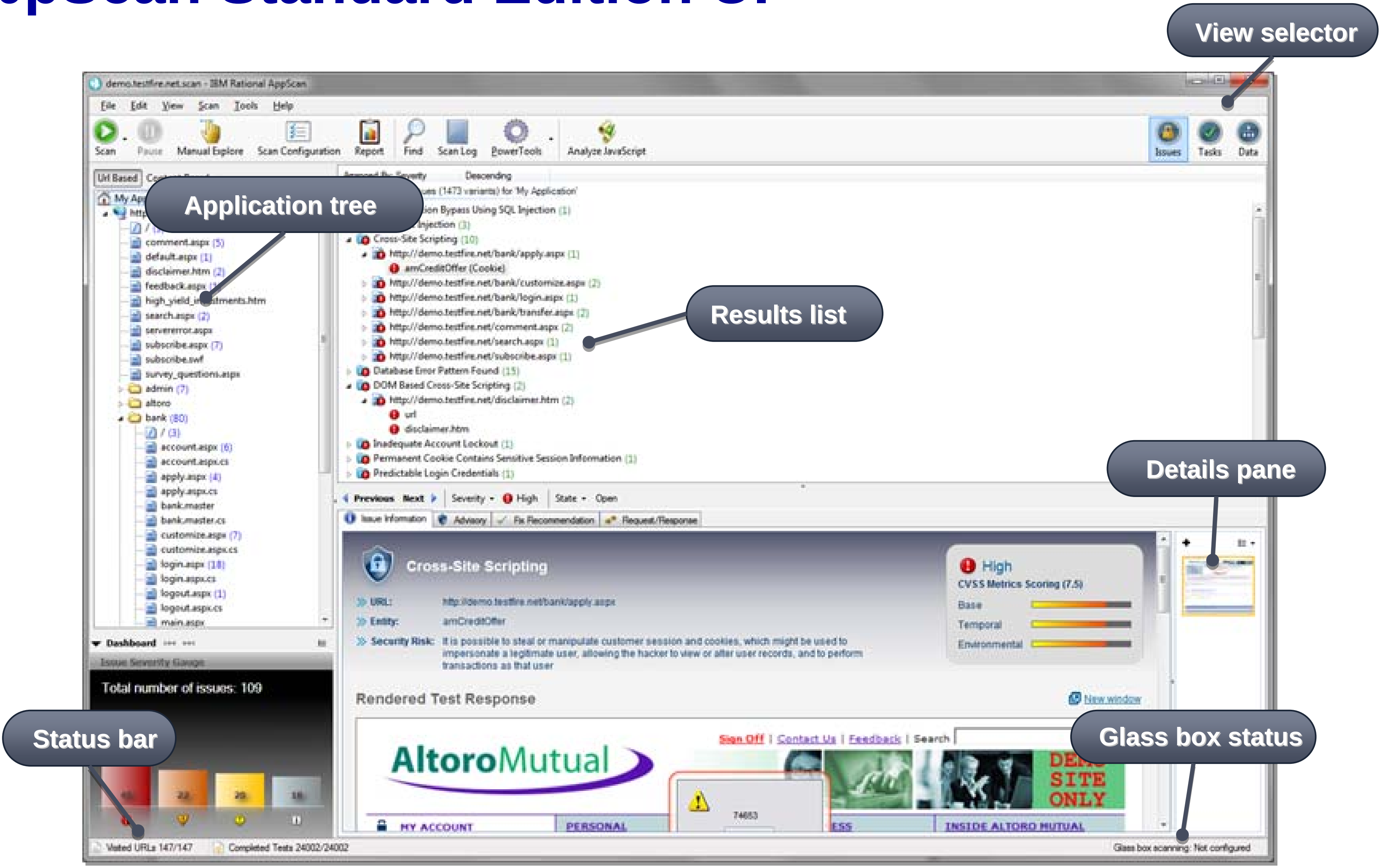
- **Configure & test**
  - Scan Expert provides recommended settings based on your apps
- **Details & guidance to correct the vulnerability**
  - Explanation of threat and recommended fix

- **Integrate with Defect Tracking Systems**
  - Rational® ClearQuest
  - HP Quality Center
- **Compliance & Reporting**
  - 40+ compliance reports
  - Executive-level summaries
  - Guidance for development

Hybrid Technology

- **Runtime Analysis (glass box testing)**
  - Expanded threat coverage with less configuration
  - Precise results (line of code) assist remediation
- **JavaScript Security Analyzer**
  - Static taint analysis of client-side JavaScript

AppScan Standard Edition UI



View selector

Application tree

Results list

Details pane

Status bar

Glass box status

issues (1473 variants) for 'My Application'

- Authentication Bypass Using SQL Injection (1)
- Injection (3)
- Cross-Site Scripting (10)
  - http://demo.testfire.net/bank/apply.aspx (1)
    - amCreditOffer (Cookie)
  - http://demo.testfire.net/bank/customize.aspx (2)
  - http://demo.testfire.net/bank/login.aspx (1)
  - http://demo.testfire.net/bank/transfer.aspx (2)
  - http://demo.testfire.net/comment.aspx (2)
  - http://demo.testfire.net/search.aspx (1)
  - http://demo.testfire.net/subscribe.aspx (1)
- Database Error Pattern Found (15)
- DOM Based Cross-Site Scripting (2)
  - http://demo.testfire.net/disclaimer.htm (2)
    - url
    - disclaimer.htm
- Inadequate Account Lockout (1)
- Permanent Cookie Contains Sensitive Session Information (1)
- Predictable Login Credentials (1)

Cross-Site Scripting

URL: http://demo.testfire.net/bank/apply.aspx

Entity: amCreditOffer

Security Risk: It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

High

CVSS Metrics Scoring (7.5)

Base

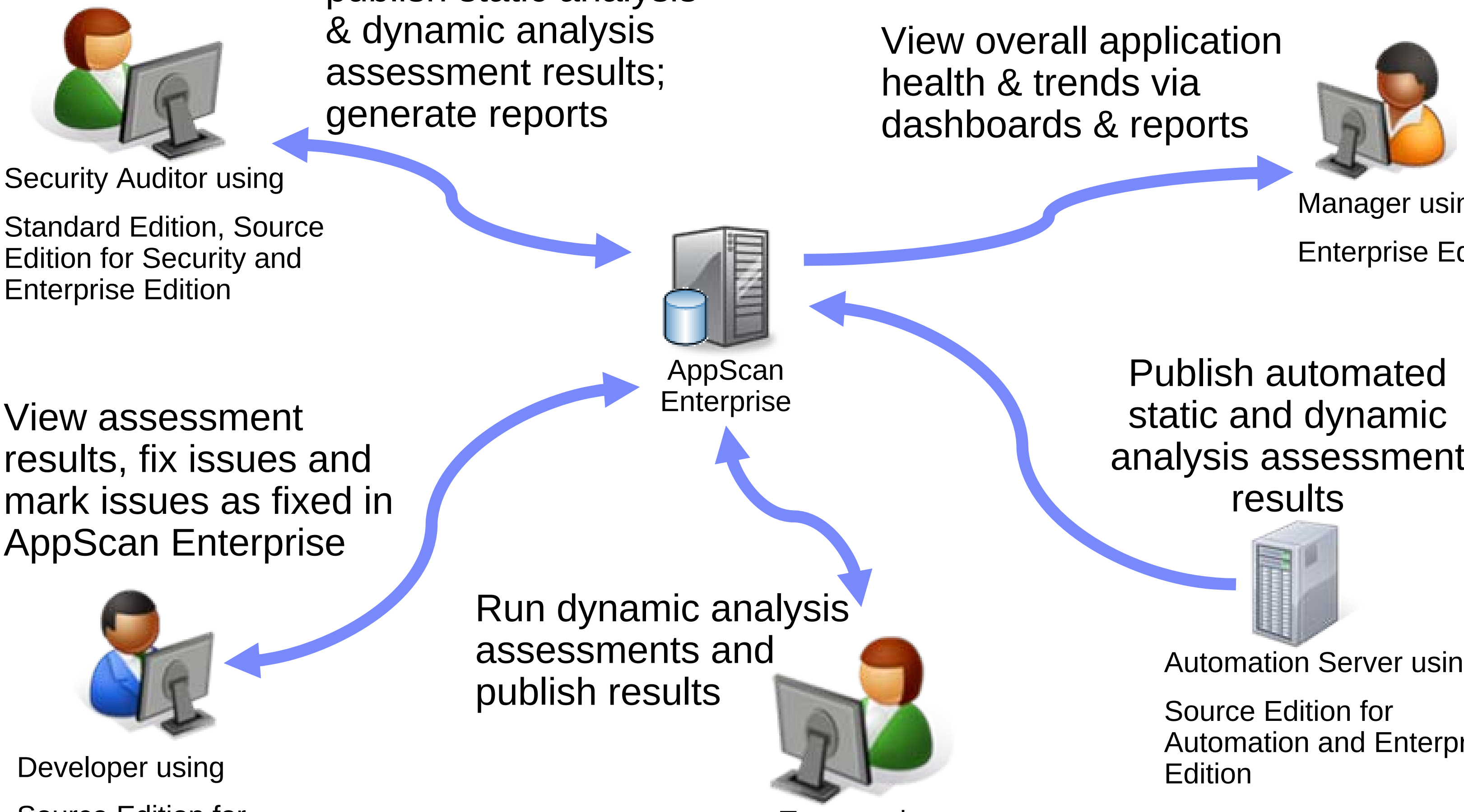
Temporal

Environmental

Rendered Test Response

Visited URLs 147/147 Completed Tests 24002/24002 Glass box scanning: Not configured

Collaboration with AppScan Suite



Run, triage and publish static analysis & dynamic analysis assessment results; generate reports

Security Auditor using

Standard Edition, Source Edition for Security and Enterprise Edition

View overall application health & trends via dashboards & reports

Manager using

Enterprise Edition

AppScan Enterprise

View assessment results, fix issues and mark issues as fixed in AppScan Enterprise

Developer using

Source Edition for Security/Developer and Enterprise Edition

Run dynamic analysis assessments and publish results

Tester using

Source Edition for Security and Enterprise Edition

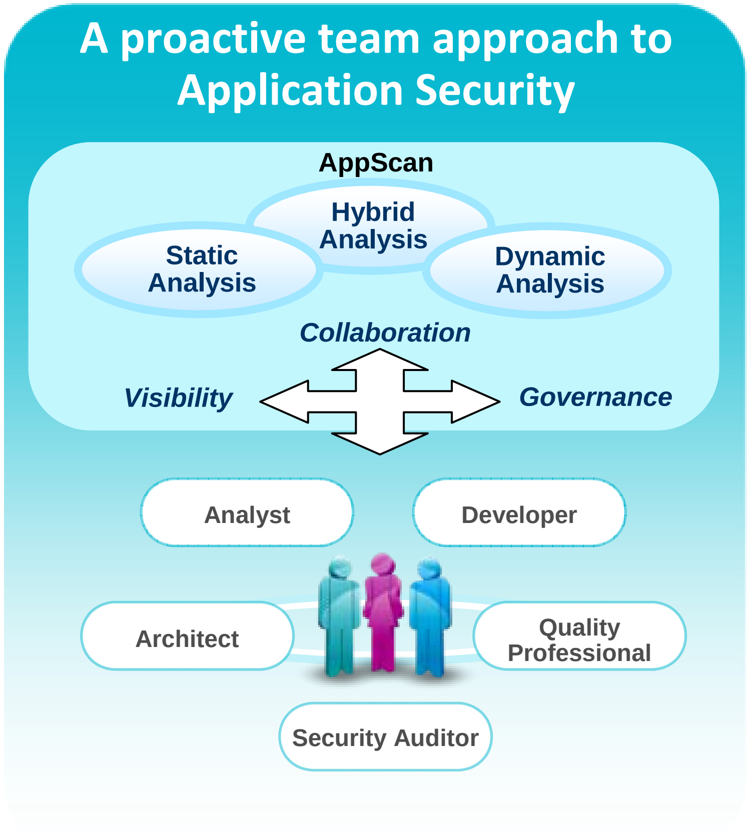
Publish automated static and dynamic analysis assessment results

Automation Server using

Source Edition for Automation and Enterprise Edition

Organizations need to take a *proactive approach* to Application Security

- Integrate **secure engineering practices** in the development lifecycle to support agile delivery demands
- Bridge the gap between "Security" and "Development" through **joint collaboration and visibility**, enabling regulatory compliance
- Use security testing tools that leverage **advanced security testing techniques**



Questions

Thank
You