



# Staying Ahead of Cybercrime: The Importance of Web Application Security

With your **real** host:  
**Adrian John Lim**  
Quality Management

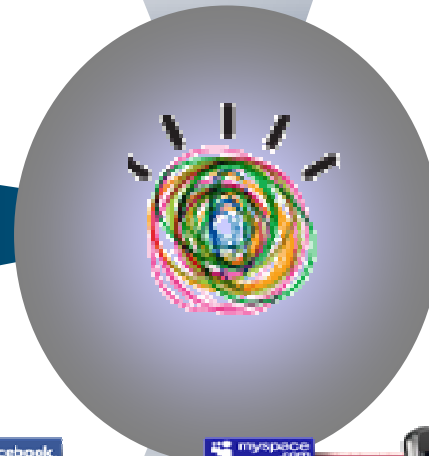
**Rational.** software

ASC02

# Welcome to the smarter planet

Globalization and Globally Available Resources

Billions of mobile devices  
accessing the Web



Access to streams of  
information in the Real Time

**New possibilities.**  
**New complexities.**  
**New risks.**



New Forms of  
Collaboration

# Smarter planet opportunities driven by Web-enabled applications

## The Opportunity – smarter planet



## The Security Equation Has Changed

- How businesses look at security has changed
  - ▶ Security is now business driven not technology driven
  - ▶ Security is now defined through risk management and compliance disciplines instead of threat and technology disciplines
  
- The threat landscape has changed
  - ▶ Traditional operating system and native client application security risks have become somewhat passé
  - ▶ Client threats are now all about the browser environment
  - ▶ Server threats are now all about web applications

## The Security Landscape of Old

- **Traditional Infrastructure was easier to protect . . .**
- Concrete entities that were easy to understand
- Attack surface and vectors were very well-defined
- Application footprint very static
- Perimeter defense was king



## Changing Security Landscape of Today

### “Webification” has changed everything ...

- Infrastructure is more abstract and less defined
- Everything needs a web interface
- Agents and heavy clients are no longer acceptable
- Traditional defenses no longer apply

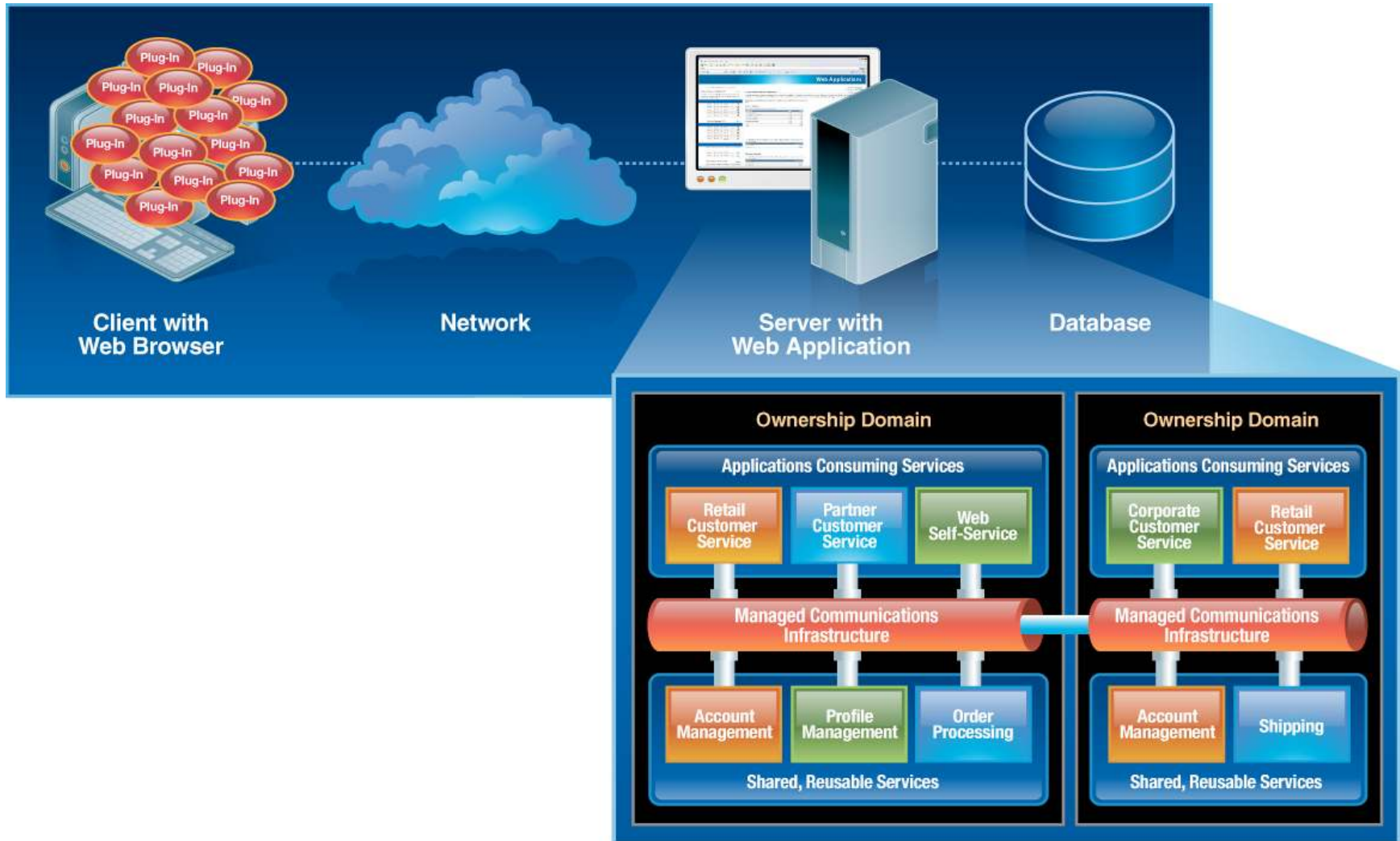


## The Web Ecosystem (simple view)



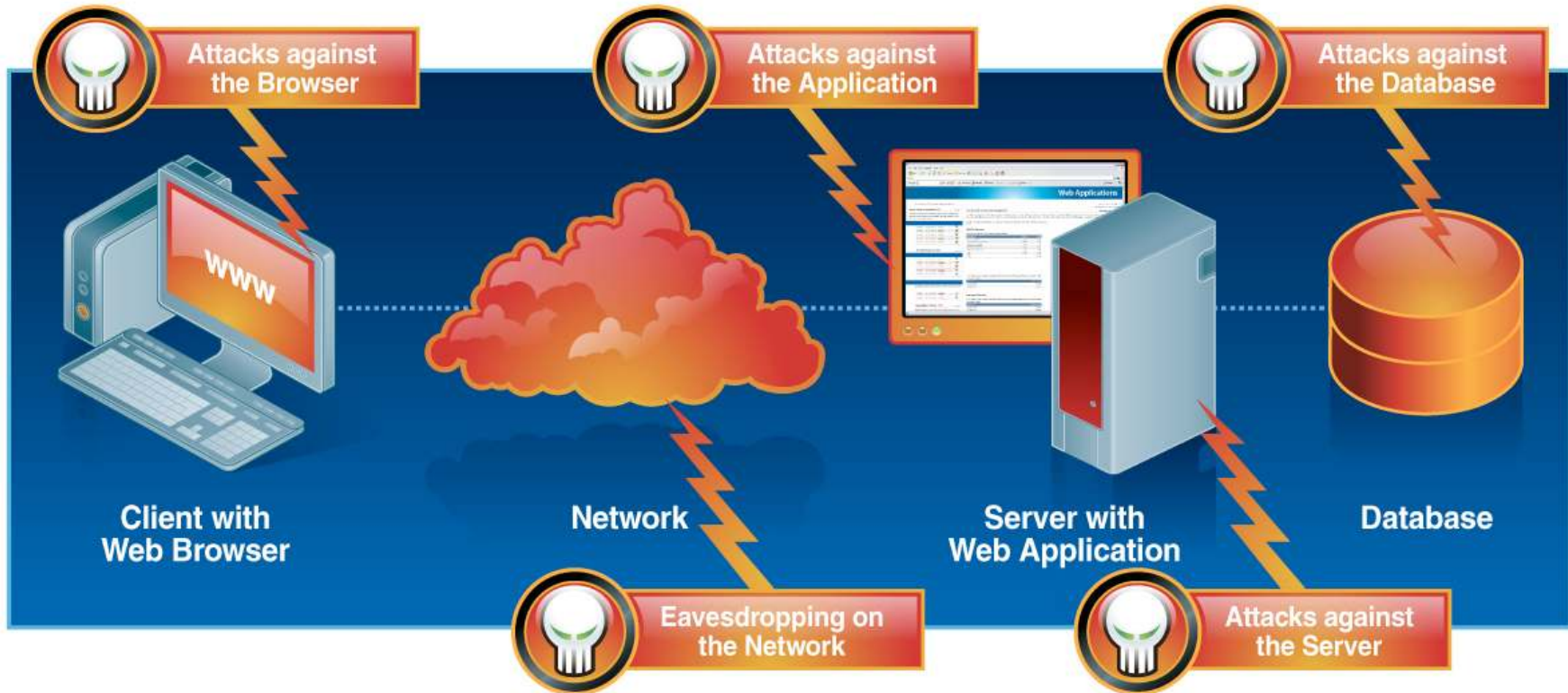
- Client with a web browser renders the content for a user
- Network transports content between the server and the client
- Server with the web application performs the required action
- Database stores information

# The Web Ecosystem (complex view)





# Attack Vectors



## 2008 Web Threats Take Center Stage

### ■ Web application vulnerabilities

- ▶ Represent largest category in vuln disclosures (55% in 2008)
- ▶ 74% of Web application vulnerabilities disclosed in 2008 have no patch to fix them

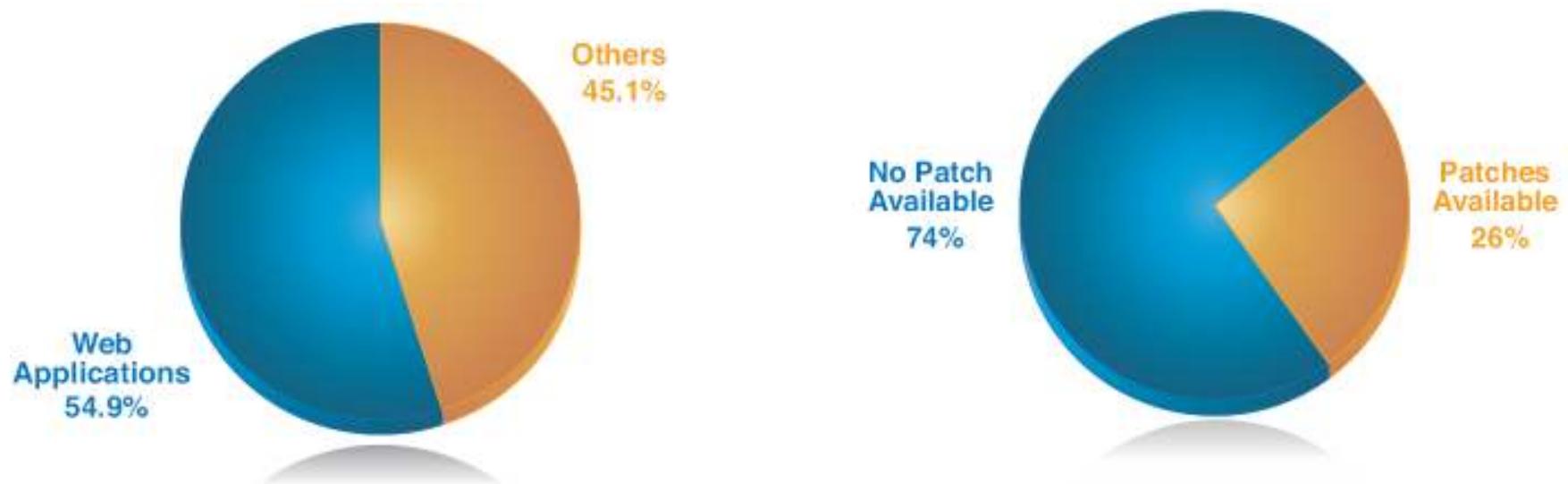


Figure 22: Percent of 2008 Web Application Vulnerabilities with No Vendor-Supplied Patch Available at the End of 2008

## Attack Techniques are Plentiful and Trivial

- SQL injection and cross-site scripting are the two largest categories of Web application vulnerabilities
- SQL injection is fastest growing category (up 134% in 2008)

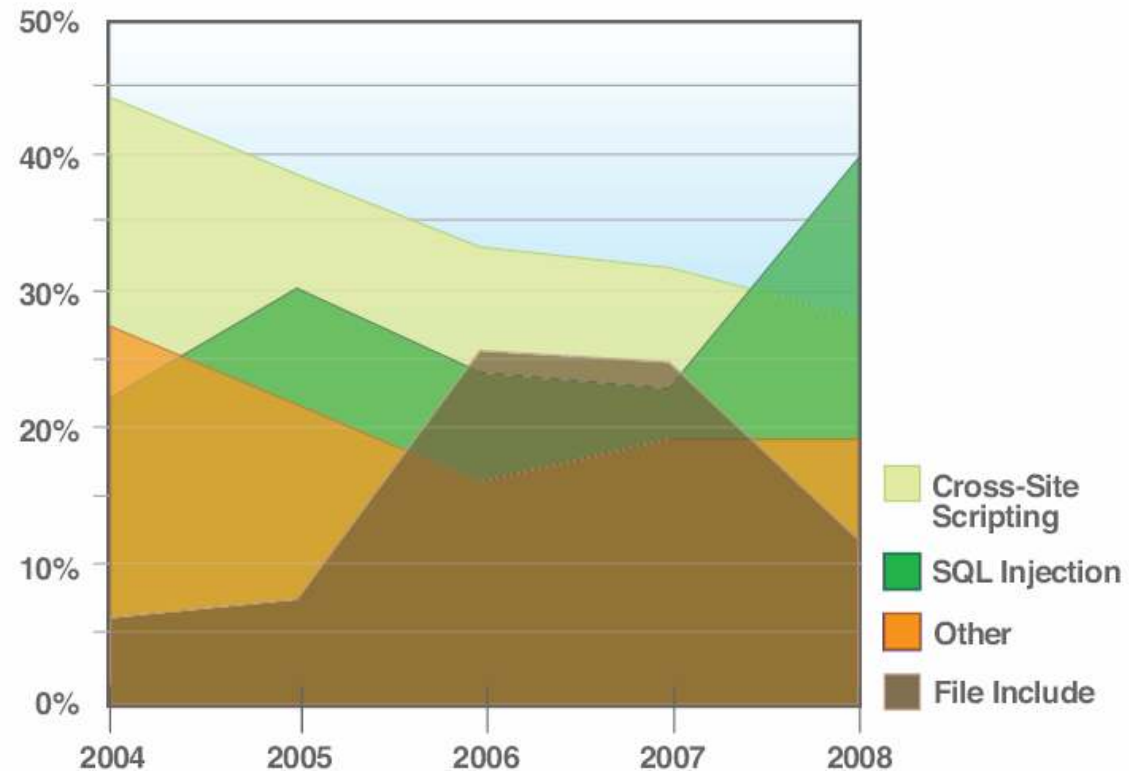


Figure 19: Web Application Vulnerabilities by Attack Technique, 2004 – 2008

## Exploitation is Rampant

- Exploitation of SQL injection skyrocketed in 2008
  - ▶ Increased by 30x from the midyear to the end of 2008

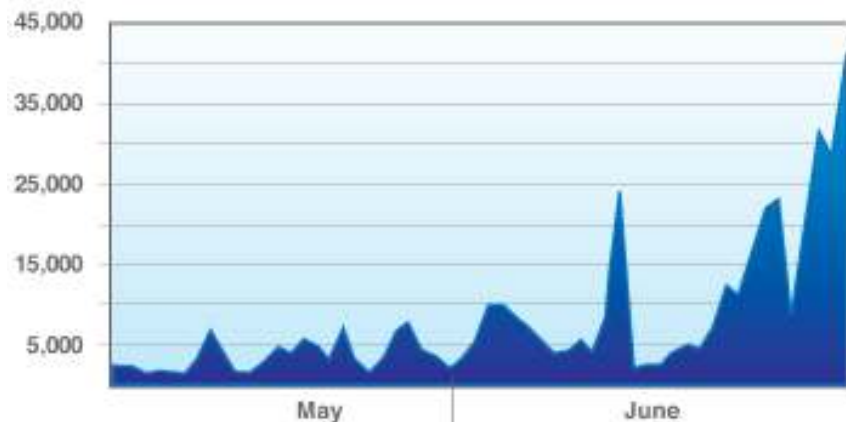


Figure 20: Initial SQL Injection Attacks Monitored by IBM ISS Managed Security Services, May – June 2008

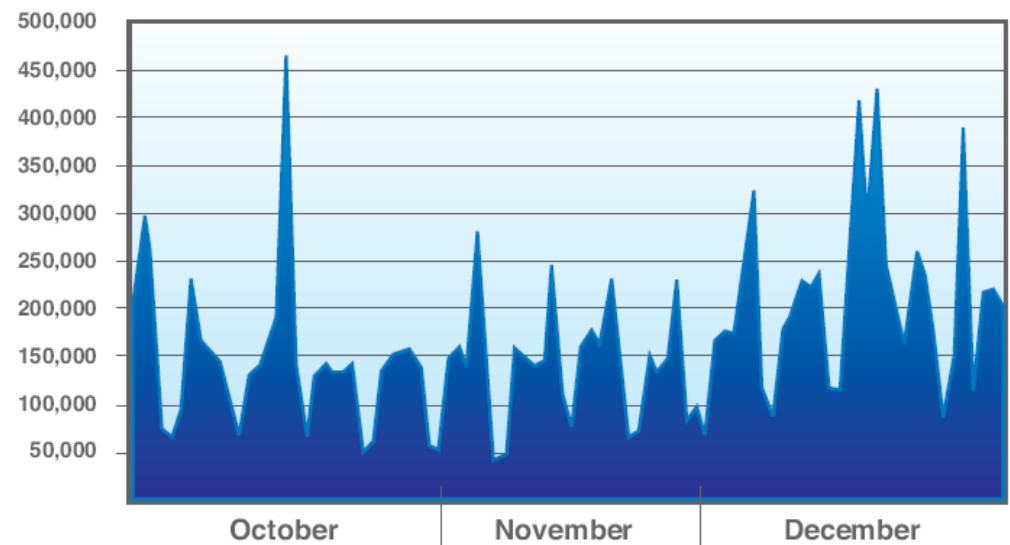
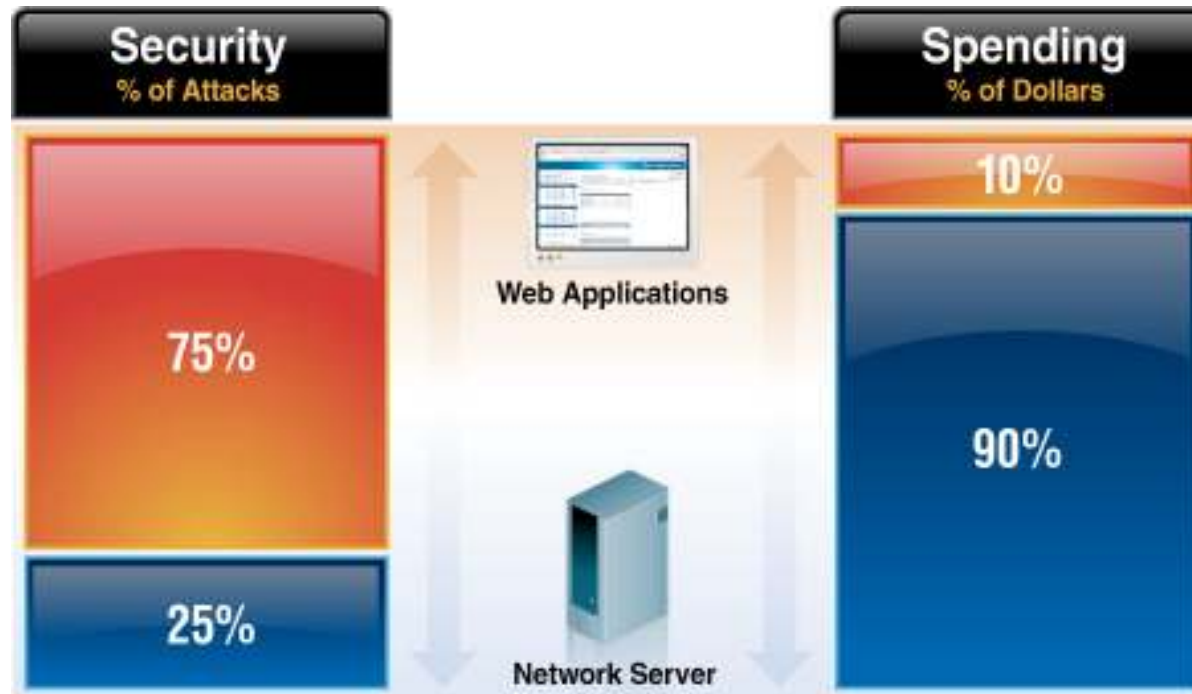


Figure 21: SQL Injection Attacks Monitored by IBM ISS Managed Security Services, Q4 2008

## Reality: Security and Spending Are Unbalanced



**75%** of All Attacks on Information Security are Directed to the Web Application Layer

**2/3** of All Web Applications are Vulnerable \*\*Gartner

## The Conundrum

- How security professionals and businesses prioritize risk and threats haven't changed with the overall landscape
- Businesses and professionals still tend to prioritize risk against an outdated traditional infrastructure viewpoint
- Businesses and professionals still tend to implement security solutions that focus on traditional threats and vectors
- Big blind spots
  - ▶ Browsers and web applications are still largely ignored or prioritized below other infrastructure from a security perspective

## Web Threats Will Become Increasingly Complex

- Web becoming main application delivery interface and ecosystem
- Popularization of new web technologies (Web 2.0) growing attack surface
- New techniques and scenarios for targeting web infrastructure

**Web Protection Does  
Not Have To Be ...**



# 1. Cross-Site Scripting (XSS)

## ■ What is it?

- ▶ Malicious script echoed back into HTML returned from a trusted site, and runs under trusted context

## ■ What are the implications?

- ▶ Steal your cookies for the domain you're browsing
- ▶ Completely modify the content of any page you see on this domain
- ▶ Track every action you do in that browser from now on
- ▶ Redirect you to a Phishing site
- ▶ Exploit browser vulnerabilities to take over machine



## 2. Injection Flaws

- What is it?
  - ▶ User-supplied data is sent to an interpreter as part of a command, query or data.
  
- Many kinds of injection flaws
  - ▶ LDAP, XPath, SSI, MX (Mail)...
  - ▶ HTML Injection (Cross Site Scripting)
  - ▶ HTTP Injection (HTTP Response Splitting)
  
- What are the implications?
  - ▶ SQL Injection – Access/modify data in DB
  - ▶ SSI Injection – Execute commands / access sensitive data
  - ▶ LDAP Injection – Bypass authentication

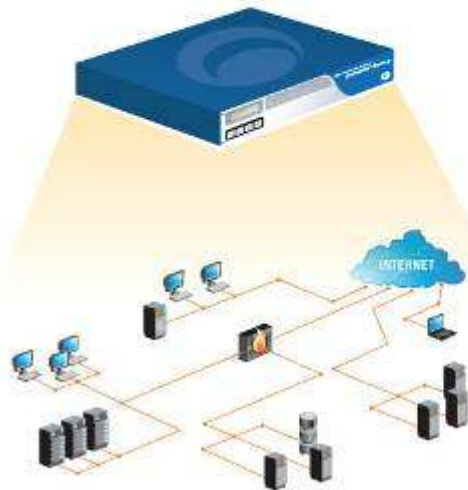
## The Strategy

- **Reduce Cost and Time to Market**
  - ▶ Find the issues earlier in the SDLC
  - ▶ Automate the process
  - ▶ Use less security savvy employees by leveraging tools
- **Mitigate Risk and increase quality**
  - ▶ Increase coverage
  - ▶ Involve more people in the process: Developers / QA
- **Increase Visibility**
  - ▶ Distribute reports to different levels
  - ▶ Dashboards
- **Increase Productivity**
  - ▶ Build the knowledge
  - ▶ Prevent of doing the same mistake

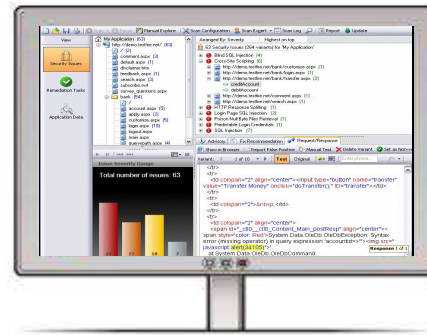
# How does Application Security Testing work?



Explore source code and/or web site to detect structure



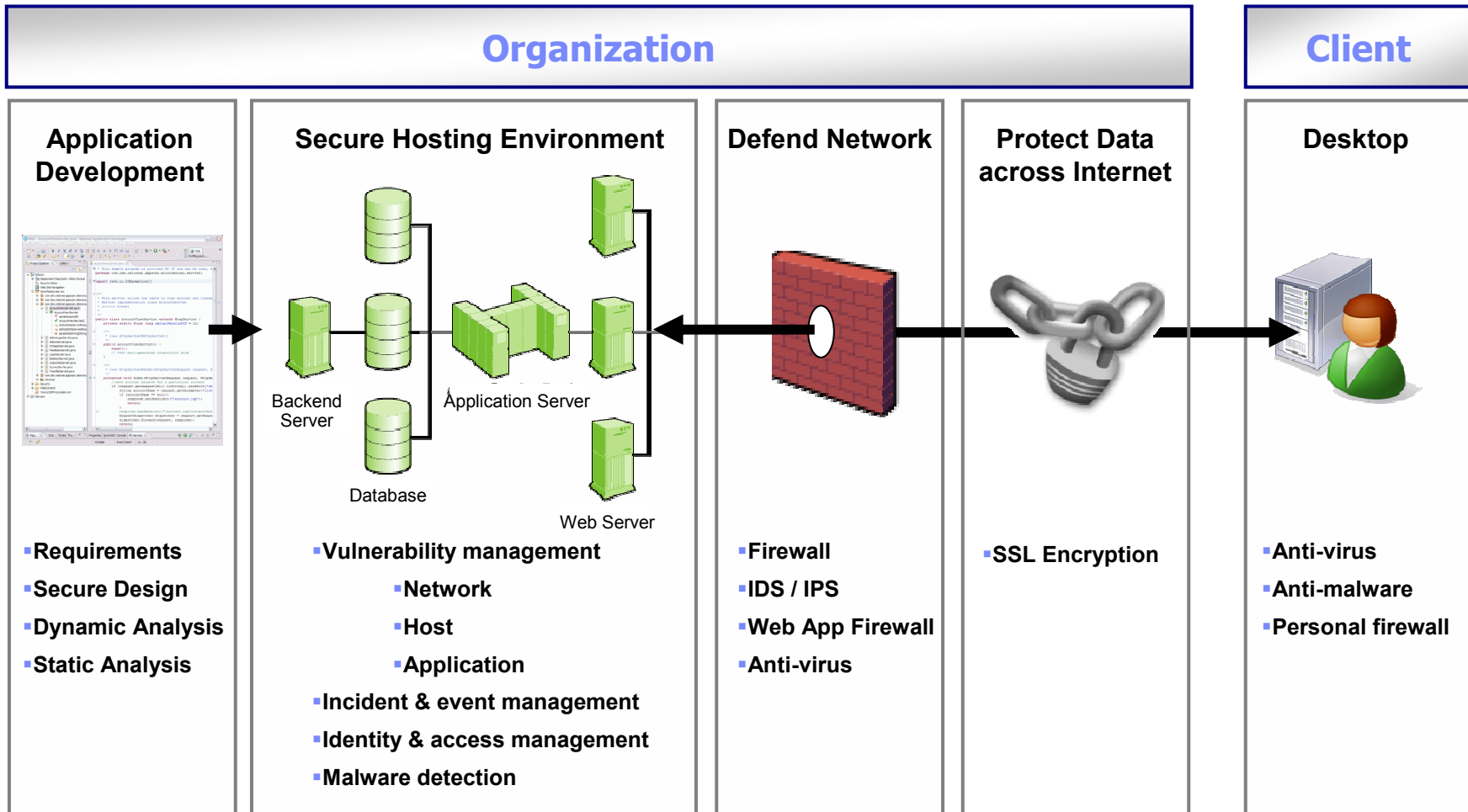
Identify Vulnerabilities ranked after severity and show how it was identified



Advanced remediation, fix recommendations and security enablement



# Secure Web Applications: Who is responsible?



# Secure Application Development

## ■ Challenge

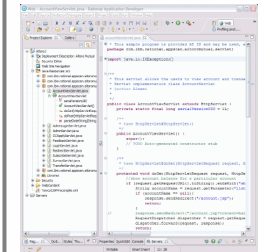
- ▶ Ensure the creation of high quality, secure and compliant software
- ▶ Ensure effective management of secure requirements, design and testing
- ▶ Lifecycle management of vulnerabilities
- ▶ Application Lifecycle Management (ALM)

## ■ IBM Solutions

- ▶ IBM Rational AppScan
  - Dynamic Analysis
  - Static Analysis
  - Runtime Analysis

Rational. AppScan.

### Application Development

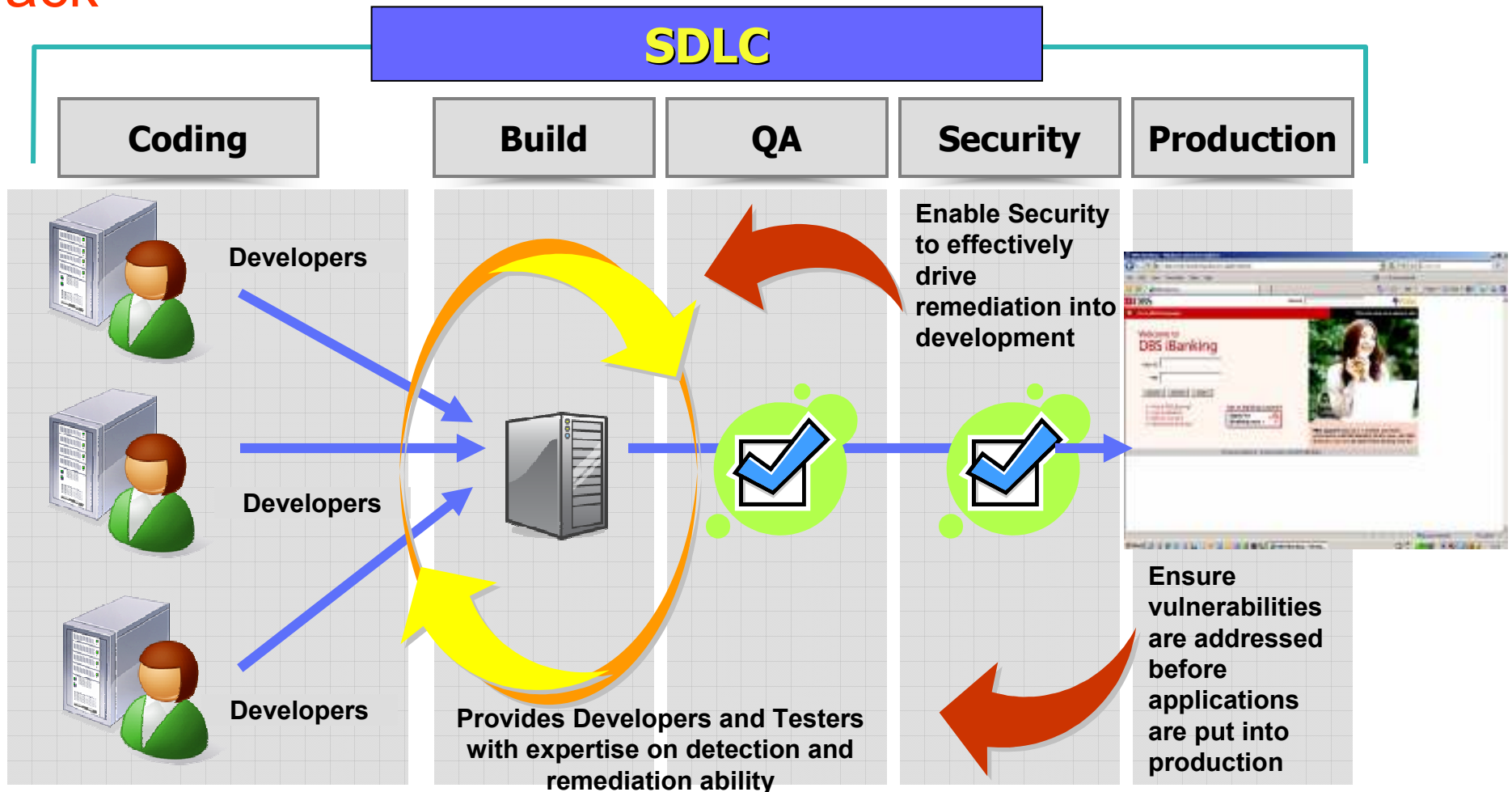


- Requirements
- Secure Design
- Dynamic Analysis
- Static Analysis

## Required Technologies for Securing the SDLC

- Tier 1: Source Control & Change Request Management
  - ▶ Rational ClearQuest / Change
  - ▶ Rational ClearCase / Synergy
- Tier 2: Requirements & Test Management
  - ▶ Rational RequisitePro / Doors
  - ▶ Rational AppScan
  - ▶ Rational Quality Manager
- Tier 3: Build Management
  - ▶ Rational Build Forge
- Tier 4: Architectural & Asset Management
  - ▶ Rational Asset Manager
  - ▶ Rational Software Architect

# Building security & compliance into the SDLC – further back



# Identify Vulnerabilities

The screenshot displays the IBM AppScan 7.5 interface. The main window title is "AppScan 7.5 Demo Scan 1.scan - Watchfire AppScan". The interface is divided into several sections:

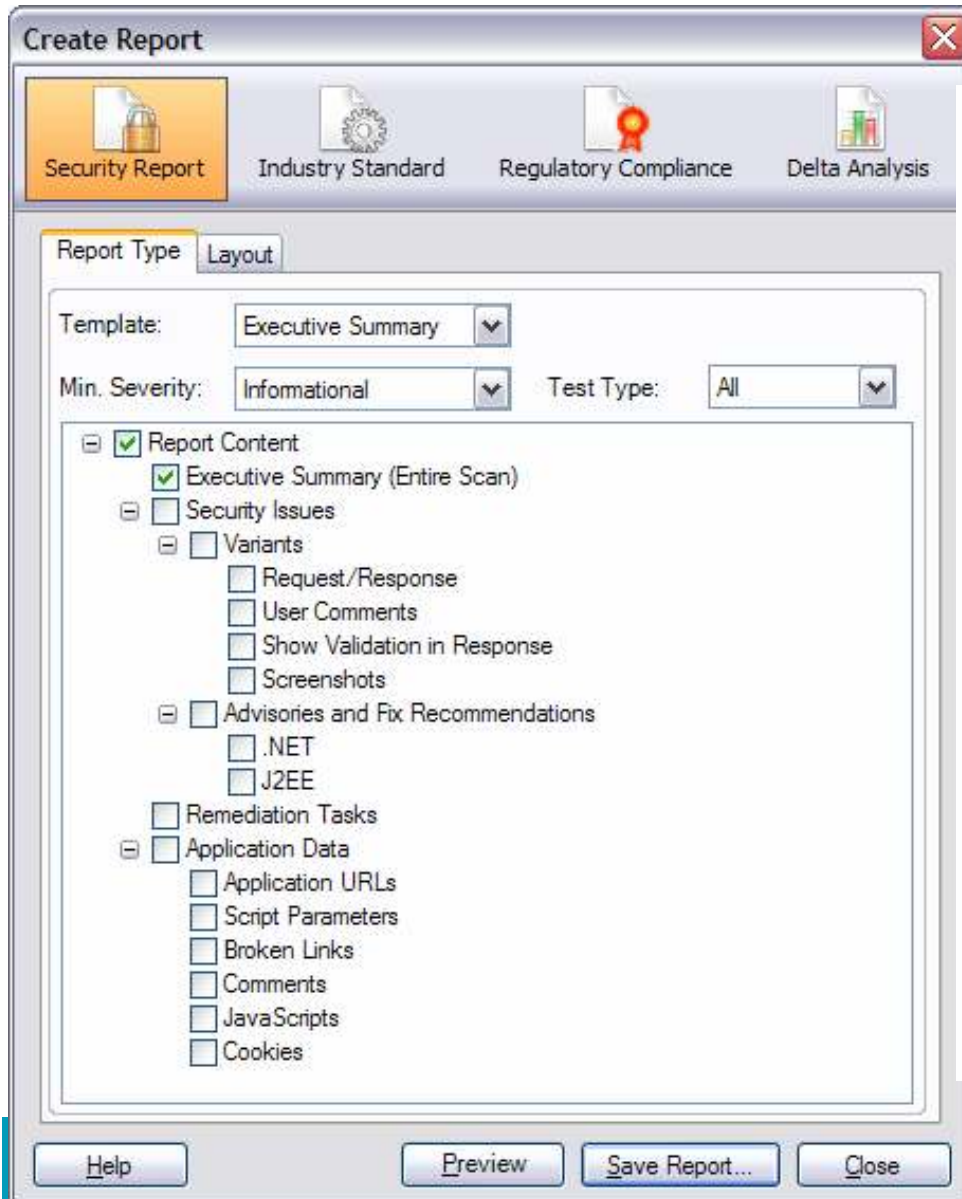
- Left Panel:** Contains navigation options: "Security Issues" (with a lock icon), "Remediation Tasks" (with a green checkmark icon), and "Application Data" (with a magnifying glass icon).
- Tree View:** Shows the scanned application structure under "My Application (53)":
  - http://demo.testfire.net/ (53)
    - / (3)
      - cgi.exe (1)
      - comment.aspx (2)
      - default.aspx
      - disclaimer.htm
      - feedback.aspx (1)
      - search.aspx (1)
      - servererror.aspx
      - subscribe.aspx (3)
      - subscribe.swf
      - survey\_questions.aspx
    - admin (1)
    - bank (40)
    - images (1)

- Main Content Area:**
- Header: "Scan is Incomplete" with a "More Information" link.
- Summary: "53 Security Issues (368 variants) for 'My Application'".
- Issues List (Aranged By: Severity, Highest on top):
  - Blind SQL Injection (4)
    - http://demo.testfire.net/bank/account.aspx (1)
    - http://demo.testfire.net/bank/login.aspx (2)
    - http://demo.testfire.net/bank/transaction.aspx (1)
  - Cross-Site Scripting (5)
  - Format String Remote Command Execution (1)
  - HTTP Response Splitting (1)
  - SQL Injection (6)
  - XPath Injection (1)
  - Cookie Poisoning SQL Injection (1)
- Variant: 1 of 2. Buttons: "Test", "Original", "Properties".
- Actions: "Show in Browser", "Report False Positive", "Manual Test", "Delete Variant", "Set as Non-vulnerable".
- Request/Response view:
  - Request: POST /bank/account.aspx HTTP/1.0, Cookie: amCreditOffer=CardType=Gold&Limit=10000&Inter, Content-Length: 35, Accept: \*/\*, Accept-Language: en-us, User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Win32), Host: demo.testfire.net, Content-Type: application/x-www-form-urlencoded, Referer: http://demo.testfire.net/bank/main.aspx.
  - Variant: listAccounts=0%2B0%2B1001160141%2B0
  - Response: HTTP/1.1 200 OK, Content-Length: 11744, Connection: close, Date: Thu, 05 Apr 2007 15:03:34 GMT, Server: Microsoft-IIS/6.0, X-Powered-By: ASP.NET, X-AspNet-Version: 2.0.50727, Cache-Control: no-cache, Pragma: no-cache, Expires: -1.
- Variant Details (ID: 9294):
  - Difference:** The following changes were applied to the original request:
    - Set parameter listAccounts's value to '0%2B0%2B1001160141%2B0'
  - Reasoning:** This test uses several different HTTP requests in order to verify the existence of a Blind SQL Injection vulnerability. The resulting
- Status Bar:** Visited URLs 108/108, Completed Tests 14194/14194, 53 Security Issues, 18 Critical, 4 High, 22 Medium, 9 Low.



# With Rich Report Options

44 Regulatory Compliance Standards, for Executive, Security, Developers.



## Detailed Findings

**Vulnerable URL:** <http://fake/fake.aspx>

Total of 2 findings in this URL

### [1 of 2] [Cross site scripting](#)

Severity: **High**

Advisory & Fix Recommendation: [See Appendix 1](#)

**Vulnerable URL:** <http://fake/fake.aspx> (parameter = fake)

**Remediation:**

**Sanitize user input**

#### Variant 1 of 4 [ID=2416]

This test variant was constructed from the original request by applying the following change(s):

- Set parameter 'uid's value to '>><script>alert('Appscan%20-%20CSS%20attack%20may%20be%20used')</script>'
- Set parameter 'uid's value to '>><script>alert('Appscan%20-%20CSS%20attack%20may%20be%20used')</script>'

**Request:**

```
GET /bank/login.aspx?uid=>><script>alert('Appscan%20-%20CSS%20attack%20may%20be%20used')</script>&passw=Demo1234&x=&y= HTTP/1.0
Cookie: ASP.NET_SessionId=3bg3jsupvfrjf0i3bph10rq1
Host: bern
Accept: */*
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)
Referer: http://bern/bank/login.aspx
```

#### Variant 2 of 4 [ID=2418]

This test variant was constructed from the original request by applying the following change(s):

- Set parameter 'uid's value to '>><script>alert('Appscan%20-%20CSS%20attack%20may%20be%20used')</script>'
- Set parameter 'uid's value to '>><script>alert('Appscan%20-%20CSS%20attack%20may%20be%20used')</script>'

**Request:**

```
GET /bank/login.aspx?uid=>><script>alert('Appscan%20-%20CSS%20attack%20may%20be%20used')</script>&passw=Demo1234&x=&y= HTTP/1.0
Cookie: ASP.NET_SessionId=3bg3jsupvfrjf0i3bph10rq1
Host: bern
Accept: */*
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)
Referer: http://bern/bank/login.aspx
```

# Actionable Fix Recommendations



The screenshot displays the AppScan 7.5 interface. The left sidebar shows navigation options: Security Issues (locked), Remediation Tasks (checked), and Application Data. The main area shows a tree view of the scanned application structure, including folders like 'admin', 'bank', and 'images', and various files like 'cgi.exe', 'comment.aspx', etc.

A notification bar at the top indicates "Scan is Incomplete". Below it, a summary shows "53 Security Issues (368 variants) for 'My Application'". The issues are listed as follows:

- Blind SQL Injection (4)
  - http://demo.testfire.net/bank/account.aspx (1)
  - http://demo.testfire.net/bank/login.aspx (2)
  - http://demo.testfire.net/bank/transaction.aspx (1)
- Cross-Site Scripting (5)
- Format String Remote Command Execution (1)
- HTTP Response Splitting (1)
- SQL Injection (6)
- XPath Injection (1)
- Cookie Poisoning SQL Injection (1)

The "Fix Recommendation" tab is selected, showing details for "Blind SQL Injection".

### Blind SQL Injection

#### Fix Recommendation

**General**

There are several issues whose remediation lies in sanitizing user input. By verifying that user input does not contain hazardous characters, it is possible to prevent malicious users from causing your application to execute unintended operations, such as launch arbitrary SQL queries, embed Javascript code to be executed on the client side, run various operating system commands etc.

It is advised to filter out all the following characters:

- [1] | (pipe sign)
- [2] & (ampersand sign)
- [3] ; (semicolon sign)

The status bar at the bottom shows: Visited URLs 108/108, Completed Tests 14194/14194, 53 Security Issues, 18 Critical, 4 High, 22 Medium, and 9 Informational.

# AppScan - CQTM & RQM Integration *Protect Your Investment*

The screenshot displays the Eclipse IDE interface for CQTM integration with AppScan. The main window is titled 'localhost\_rev2.scan\_1180471626035\_CQTM'. The 'Test Log' panel shows a list of events, with the last one being a 'fail' event labeled 'Watchfire AppScan Event'. To the right, the 'Watchfire AppScan Regression Results' panel includes links for 'Show in AppScan', 'Update Baseline', and 'View Delta Analysis Report'. Below this is the 'Extended Properties' section, which contains a table of properties and an 'Attachments' section.

The 'Test Results' panel at the bottom shows a table with the following data:

Result	Test Type	Verdict	Descri...	ID	Headline	Test Script File	Lo...
Uncommitted ...							
Configure...	AppScan	fail		SAMPL00000075	AppScanTest	\\cconboy-xpl2\...	\\c...
Recently Com...							

Other panels visible include 'ClearQuest Navigator' showing a tree view of 'Asset1' and 'Test Plans', 'Console' with execution logs, and 'Progress' showing 'Test Script Execution' at 0%.

## Executive Summary - IBM acquires Ounce

- Today's smart products are increasingly interconnected, instrumented and intelligent. Software makes products smarter but has the potential to **introduce risk from cyber-attack and regulatory non-compliance**
  - Application security is one of the top security issues
  - The cost and lack of coverage of reactive security is driving companies towards proactive measures – **building security into the application development process**
  - Traditional approaches make it unlikely that development will support security testing due to schedule risks and potential project failure
- 
- **IBM is acquiring Ounce Labs to expand its market leading web application security portfolio.**
  - **By combining Ounce Labs source code testing with IBM Rational AppScan's application security testing we are the **only provider of a true end-to-end application security solution for managing security compliance across all stages of the development process** – from coding to production.**



*Delivering greater value  
from your investments in software*

# Who is Ounce Labs?

## ▪ Ounce Labs is a proven leader in Static Code Analysis Security

- Named “leader” in Gartner’s 1<sup>st</sup> Magic Quadrant for SAST (Static Application Security Testing)
- Software developer founded 2002
- Headquarters: Waltham, MA
- Source code security experts

### Products

- **Identifies** security vulnerabilities in application source code
- **Automates and integrates** security analysis in the SDLC
- **Connects** source code analysis to enterprise security and GRC platforms

### Technology

- 4 **granted** source code security patents, 3 pending
- **Superior** architecture for scalable enterprise deployments
- **Community leadership:** Multiple solutions released into open source, senior-level contributions to PCI, SANS, CWE, OWASP, and more

### Customers

- 150+ customers; 98% renewal rate
- **Marquee** financial, government, retail and e-commerce customers
- Solution of choice for security SIs and consultant community

## Why has IBM acquired Ounce Labs?

*Ounce Labs provides application source code testing tools to help enterprises reduce risk and cost associated with online security and compliance breaches.*

- Application security is the largest category of vulnerability disclosures (55% in 2008)
- Rational acquired Watchfire in 2007 to address customers' Application Security Testing needs
  - ▶ AppScan continues to be recognized as the leader in Dynamic Analysis Security Testing (DAST)
  - ▶ Avanti is a recognized leader in Static Analysis Security Testing (SAST)
- Application security markets are converging
  - ▶ The combination of these two industry leading technologies provides the most accurate solution in the market
  - ▶ Avanti's technology enables Rational to fulfill our vision of moving testing earlier in the development process
- Continues to add to our competitive advantage in the application security testing segment
  - ▶ Only vendor to offer complete solutions for both SAST & DAST
  - ▶ Only vendor to offer complete integration across the software delivery lifecycle
  - ▶ Only vendor to offer a complete IT security solution across all major domains (IBM Security Framework)
- Ounce Labs is a good fit
  - ▶ Mature technology that supports all key development technologies and languages (Java, .NET, C/C++)
  - ▶ Excellent integrations with Rational SDLC products and for the developer – Rational's traditional user base

# Ounce Labs & IBM Customer Value

*The addition of Ounce Labs technology and expertise will build on IBM's broad security and compliance product and services offerings to help enterprises reduce risk and cost associated with application security breaches*



## Key to Business Integrity is Application Security - a top of mind issue for businesses and governments Worldwide

- ✓ *75% of the cyber attacks today are at the application level<sup>(1)</sup>*, where personal, customer, credit card and other high value data resides.



## This is a complex problem that requires comprehensive technology

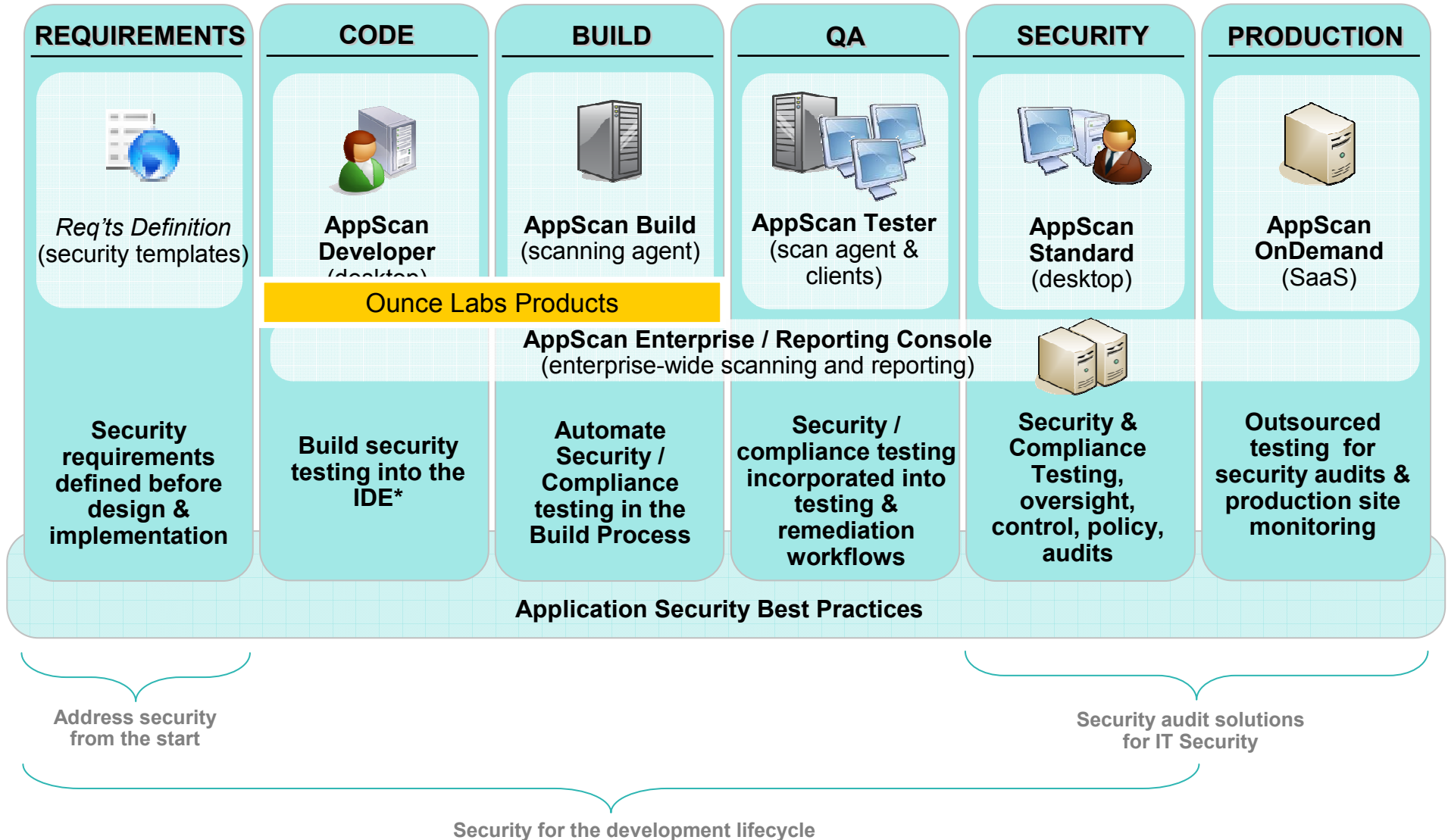
- ✓ The complexity of today's systems, coupled with the sophistication of attackers *requires equally sophisticated technology* to combat these challenges without the expense of developing or employing specialized expertise.



## IBM Provides the complete solution to Application Security

- ✓ By *integrating Ounce labs Static Code Analysis with IBM Rational AppScan's Dynamic analysis*, IBM is the *only provider of a true end-to-end application security solution for managing security compliance across all stages of the development process* – from coding to production.
- ✓ *Remediation of vulnerabilities before applications go live significantly improves business integrity and dramatically reduces cost.*

# Rational Security & Compliance—with Ounce Labs







## Why Choose IBM Rational Security & Compliance

- Broadest suite of offerings to support security testing across the development lifecycle
- **IBM is the only company to offer a complete IT security framework across all major security domains**
- Only web application security testing solution to provide combined code, dynamic, runtime and string analysis
  - Only vendor to offer complete solutions for both SAST & DAST
  - Only vendor to offer complete integration across the software delivery lifecycle
- Broadest set of security compliance reporting
- R&D backed by IBM's \$1.5B annual investment in security
- Best Application Security Analysis - Includes multiple analysis techniques to leverage strengths of all techniques & overcomes weaknesses
- Naturally fits into the SDLC process
  - Minimize disruption
  - Scale to large number of users
  - Support collaboration within development
  - Integrate with development tools



© Copyright IBM Corporation 2009. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.