

# HP Fortify Software Security Center产品介绍

北京晓通宏志科技有限公司

**ENTERPRISE SECURITY**

# HP Fortify ——软件安全的领导者

## ● 领先的市场份额

- 全世界最大的10大银行的9家、大型IT基建供应商、大型独立软件公司
- 支持市场上最流行、最多样化的编程语言

## ● 领先解决方案

- 获奖产品支持整个开发周期
- 超过150项专利
- 最庞大的安全编码规则库

## ● 专业的安装部署

- 与Fortune100 企业及大型 ISVs 开发出来的最佳做法
- 由世界顶尖安全专家鉴定

“Fortify is the clear winner for many reasons, including their superior analysis and reporting capabilities, and their understanding and support of how security fits into the software development lifecycle.”

— Mary Ann Davidson  
CSO, Oracle

**ORACLE**



# HP Fortify的主要客户

## 银行及金融



## 电子商务



## 基础软件厂商



## 政府机构



## 医疗及保健



## 电信



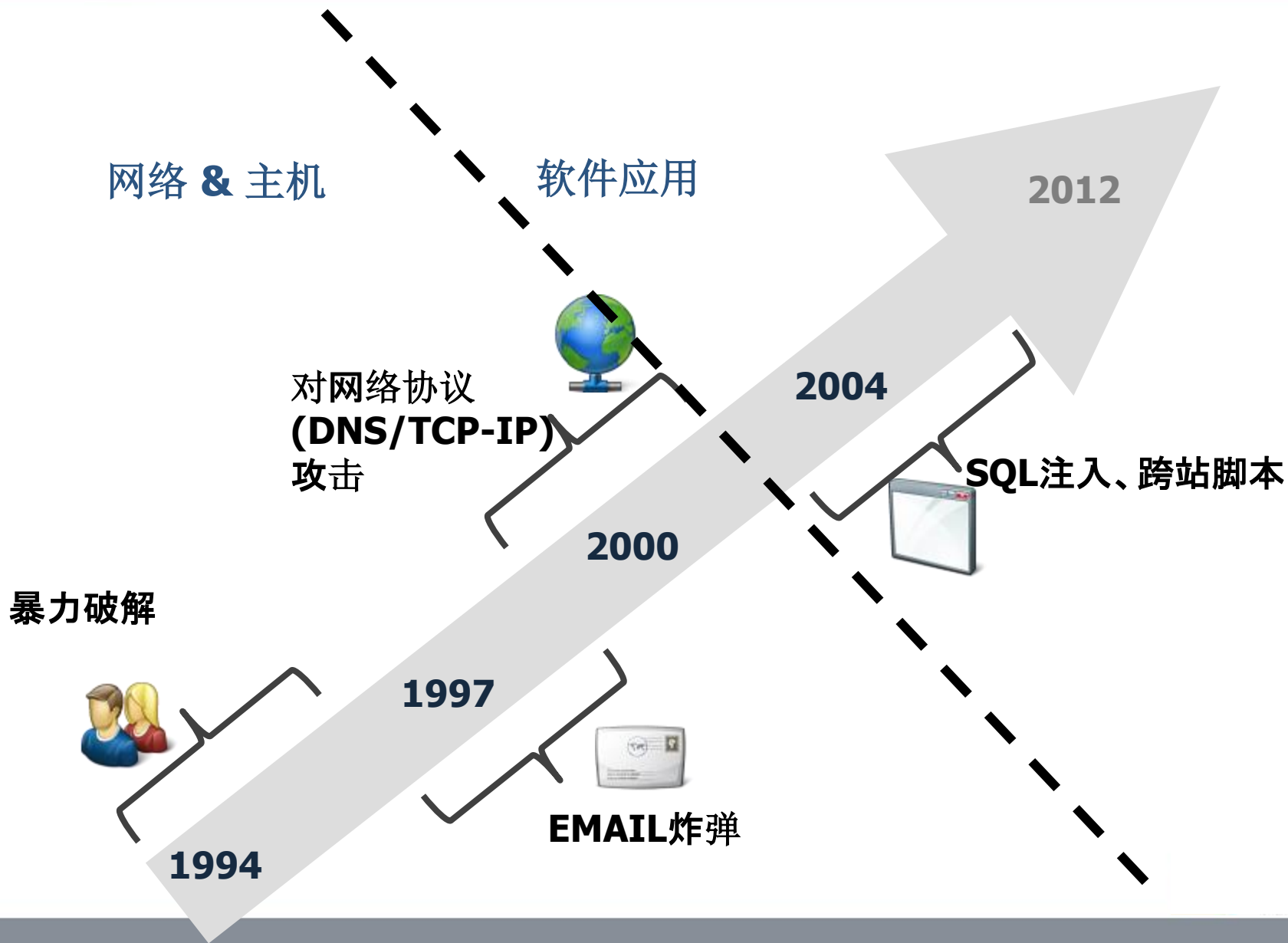
## 其它



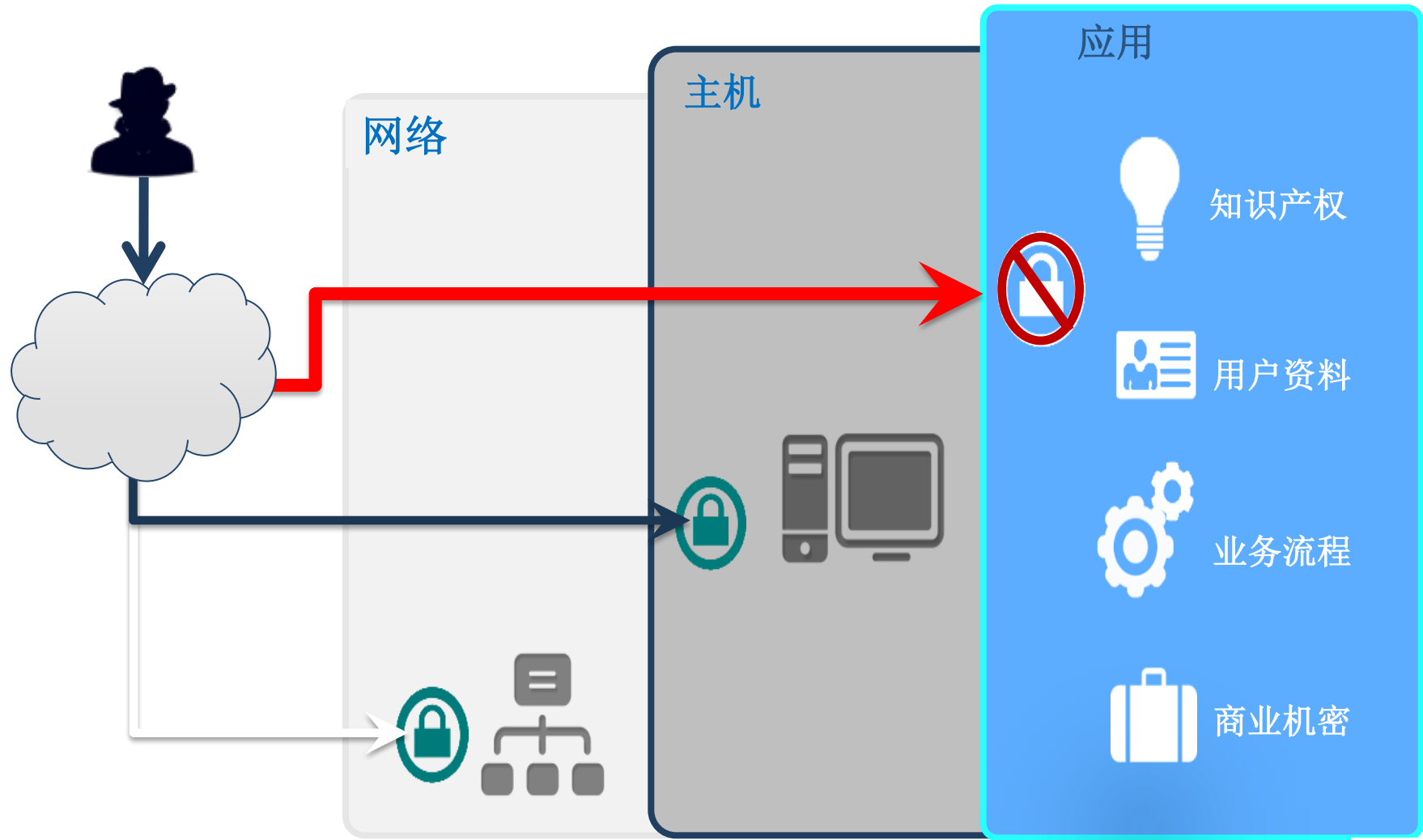
# 国内客户



# 黑客攻击的演化方式



# 新的软件架构——软件的应用因为业务和功能的需要必打破传统的保护层,直接与外面的系统交互



# OWASP 2010 TOP 10 漏洞

- **A1. 注入漏洞 Injection Flaw**
- **A2. 跨站脚本攻击 Cross Site Scripting 简称XSS**
- **A3. 错误的认证和会话管理 Malicious File Execu Broken Authentication and Session Management tion**
- **A4. 不安全的直接对象引用 Insecure Direct Object Reference**
- **A5. 跨网站请求伪造 Cross-Site Request Forgery 简称CSRF**
- **A6. 安全性误配置 Security Misconfiguration**
- **A7. 不安全的加密存储 Insecure Cryptographic Storage**
- **A8. 限制远程访问失败 Failure to Restrict URL Access**
- **A9. 不足的传输层保护 Insufficient Transport Layer Protection**
- **A10. 未验证的重定向和传递 Unvalidated Redirects and Forwards**

# Fortify 软件安全漏洞研究



[http://www.hpenterprisesecurity.com/vulncat/zh\\_CN/vulncat/index.html](http://www.hpenterprisesecurity.com/vulncat/zh_CN/vulncat/index.html)

Expand All | Close All

F A Taxonomy of Coding Errors that Affect Security

- [-] ColdFusion
- [-] C/C++
- [-] C#/VB.NET/ASP.NET
- [-] HTML
- [-] Java/JSP
  - [-] API Abuse
  - [-] Code Quality
  - [-] Encapsulation
  - [-] Environment
  - [-] Errors
  - [-] Input Validation and Representation
  - [-] Security Features
  - [-] Time and State
    - Code Correctness: Double-Checked Locking
    - J2EE Bad Practices: Non-Serializable Object Stor
    - J2EE Bad Practices: System.exit
    - J2EE Bad Practices: Threads
    - Race Condition: Singleton Member Field
    - Race Condition: Static Database Connection(dbc
    - Session Fixation
- [-] Javascript
- [-] PHP
- [-] PLSQL/TSQL
- [-] VisualBasic/VBScript/ASP
- [-] XML

## Cross-Site Scripting

### ABSTRACT

向 Web 浏览器发送非法数据会导致浏览器执行恶意代码。

### EXPLANATION

Cross-Site Scripting (XSS) 漏洞在以下情况下发生:

1. 数据通过一个不可信赖的资源进入 Web 应用程序, 通常是一个网页请求或者数据库。
2. 在未检验包含数据的动态内容是否存在恶意代码的情况下, 便将其传递给了 Web 用户。

传送到 Web 浏览器的恶意内容通常采用 JavaScript 代码片段的形式, 但也可能会包含一些 HTML、Flash 或者其它任何一种可以被浏览器执行的代码。基于 XSS 的攻击手段花样百出, 几乎是无穷无尽的, 但通常它们都会包含传输给攻击者的私人数据 (如 Cookie 或者其它会话信息)。在攻击者的控制下, 指引受害者进入恶意的网络内容; 或者利用易受攻击的站点, 对用户的机器进行其它恶意操作。

**例 1:** 下面的 JSP 代码片段可从 HTTP 请求中读取雇员的 ID, eid, 并将其显示给用户。

```
<% String eid = request.getParameter("eid"); %>
...
Employee ID:<%= eid %>
```

如果 eid 只包含标准的字母或数字文本, 这个例子中的代码就能正确运行。如果 eid 里有包含元字符或源代码中的值, 那么 Web 浏览器就会像显示 HTTP 响应那样执行代码。

起初, 这个例子似乎是不会轻易遭受攻击的。毕竟, 有谁会输入导致恶意代码的 URL, 并且还在自己的电脑上运行呢? 真正的危险在于攻击者会创建恶意的 URL, 然后采用电子邮件或者社会工程学的欺骗手段诱使受害者访问到此 URL 的连接。当受害者单击这个链接时, 他们不知不觉地通过易受攻击的网络应用程序, 将恶意内容带到了自己的电脑中。这种对易受攻击的 Web 应用程序进行盗取的机制通常被称为反射式 XSS。



# CWE : 美国安全漏洞辞典采用Fortify漏洞分类

CWE - CWE List (Draft 6) - Microsoft Internet Explorer

地址 http://cwe.mitre.org/data/index.html#Definition



## Common Weakness Enumeration

A community-developed dictionary of common software weaknesses

Home > CWE List (Draft 6)

[View the CWE List](#)

### CWE List

[Full Dictionary View](#)  
[Classification Tree](#)  
[Leaf Nodes](#)  
[Other Views](#)

### About

[Sources](#)  
[Process](#)  
[Documents](#)

### Community

[Related Activities](#)  
[Discussion List](#)

### News

[Calendar](#)  
[Free Newsletter](#)

### Compatibility

[Program](#)  
[Requirements](#)  
[Declarations](#)  
[Make a Declaration](#)

### Contact Us

[Search the Site](#)

## CWE List (Draft 6)

The Common Weakness Enumeration (CWE™), currently in a very preliminary form, is a list of software weaknesses. Creating the list is a [community initiative](#). Together, these organizations and any others that wish to join the effort, are creating specific and succinct definitions for each of the elements in the CWE List. By leveraging the widest possible group of interests and talents we hope to ensure that the CWE elements are adequately described and differentiated. The next steps are to adequately capture the specific effects, behaviors, exploit mechanisms, and implementation details in the CWE dictionary as well as to review and revise the presentation approaches that will best suit this information.

## CWE Classification Tree (Draft 6)

[Expand All](#) [Collapse All](#)

### Common Weakness Enumeration and Classification

- [-] Location - (1)
  - [-] Environment - (2)
    - [-] Errant Files or Directories Accessible - (552)
      - [-] Insecure Compiler Optimization - (14)
      - [-] Setting Manipulation - (15)
    - [-] Technology-specific Environment Issues - (3)
  - [-] Configuration - (16)
- [-] Code - (17)
  - [-] Source Code - (18)
    - [-] Data Handling - (19)
    - [-] API Abuse - (227)
    - [-] Security Features - (254)
    - [-] Time and State - (361)
    - [-] Error Handling - (388)
    - [-] Code Quality - (398)
    - [-] Encapsulation - (485)
    - [-] Byte/Object Code - (503)
  - [-] Motivation/Intent - (504)
    - [-] Intentional - (505)

Search by ID

### Section Contents

#### CWE List

[Full Dictionary View](#)  
[Classification Tree](#)  
[Leaf Nodes](#)  
[Other Views](#)

#### Other Items of Interest

[Sources](#)

网页上有错误。

Internet

开始

Skype? wanghong ...

fortify software ...

SCA install

CWE - Google 搜 ...

CWE - CWE List ( ...

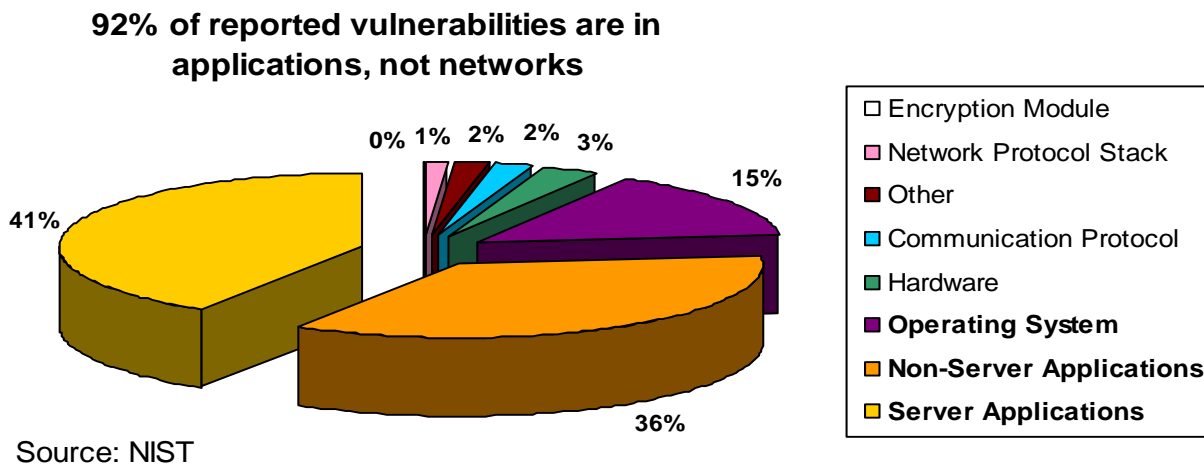
EN

10:15

# 软件安全新的防御方式

● 软件安全问题的产生的根源：

如今的黑客攻击主要利用软件本身的安全漏洞，这些漏洞是由不良的软件架构和不安全的编码产生的。

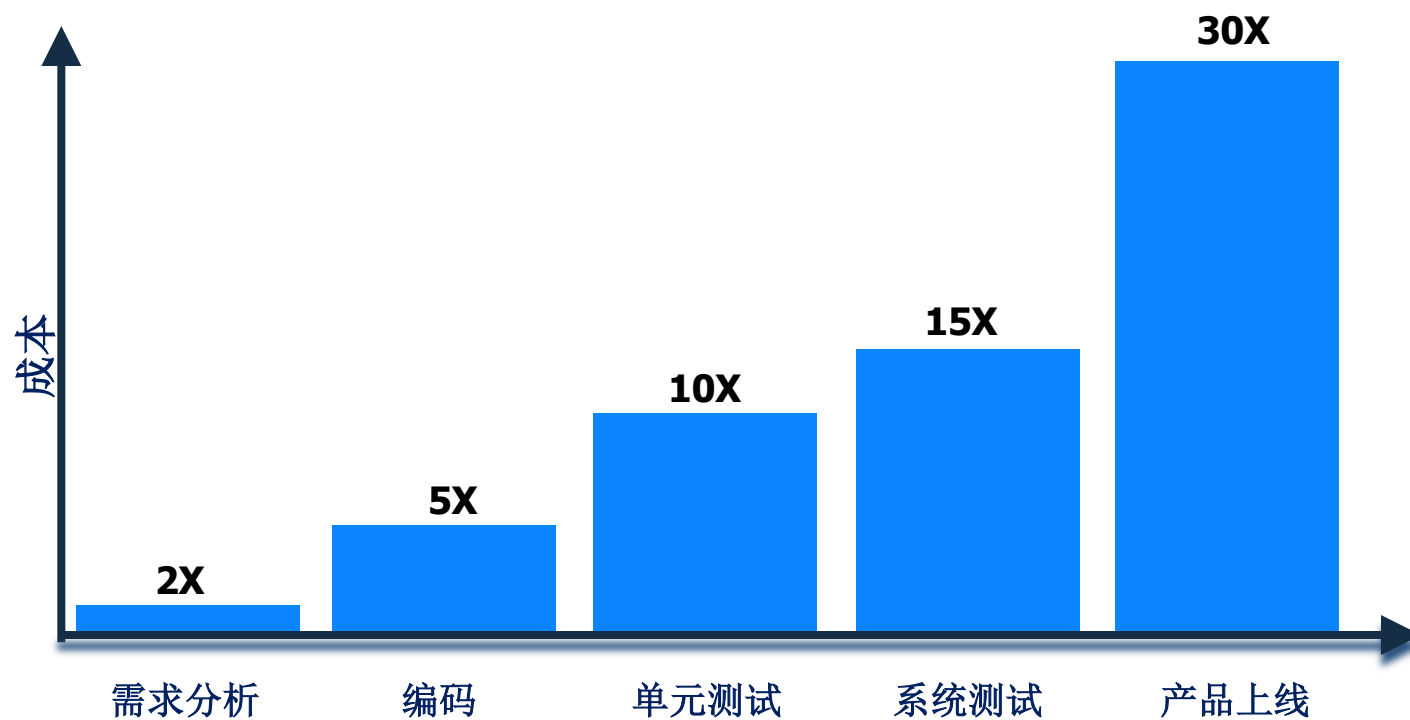


● 新的防御方案：

**Building Security In**——构建安全的代码安全的软件

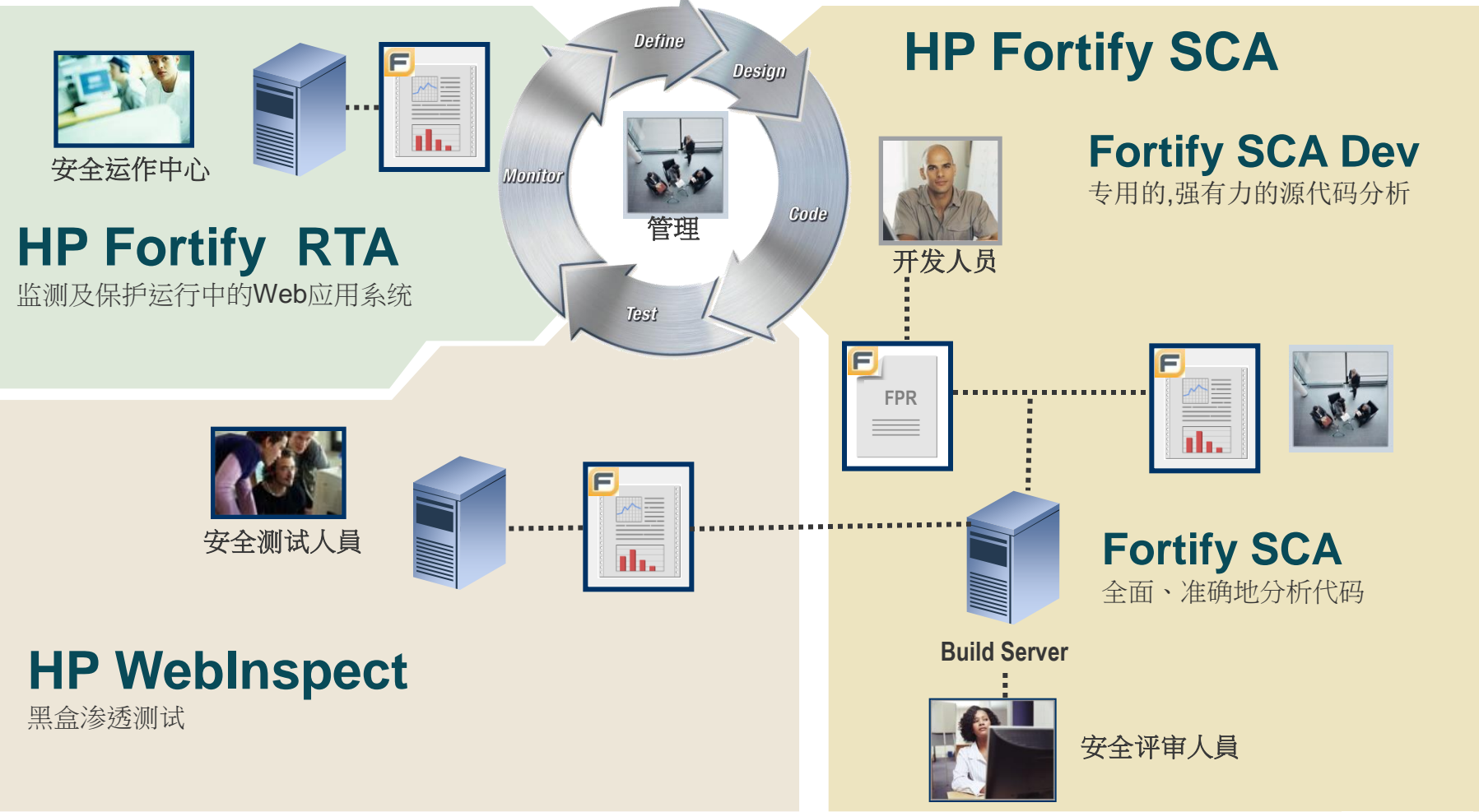
# 修复漏洞的成本

- 在产品上线阶段修复漏洞的成本多花费超过**30**倍



来源: *NIST*

# HP Fortify 的安全方案—HP Software Security Center



# HP Fortify SCA –静态应用安全测试工具

The screenshot displays the HP Fortify SCA Audit Workbench interface. The top navigation bar includes 'Summary', 'Audit Guide', 'Scan', and 'Reports'. The main window is divided into several panes:

- Summary:** Shows a filter set of 'Medium' and a total of 1398 issues, with 71 'Hot' issues highlighted in red.
- Project Summary:** Lists various project files like 'RTE\_functions\_comm...', 'functions\_common.asp', etc.
- Issue List:** A tree view showing categories like 'Cross-Site Scripting', 'HTTP Response Splitting', and 'SQL Injection'. A specific issue is selected: 'admin\_import\_subscribers.asp:556 (SQL Injection) (multiple issues)'. The analysis is marked as 'Exploitable'.
- Code View:** Displays the source code for the selected issue, showing a SQL query construction and a loop through records.
- Analysis Trace:** Shows a list of execution paths, including 'management\_centre\_update.asp:77 - Read request.cookies[]' and 'admin\_import\_subscribers.asp:556 - open(0)'.

At the bottom right, a critical issue message states: 'this is a critical issue - must be corrected in next release.'

1. 源代码白盒安全测试
2. 完全自动化地完成测试——省时省力
3. 最广泛的安全漏洞规则，多维度分析源代码安全问题

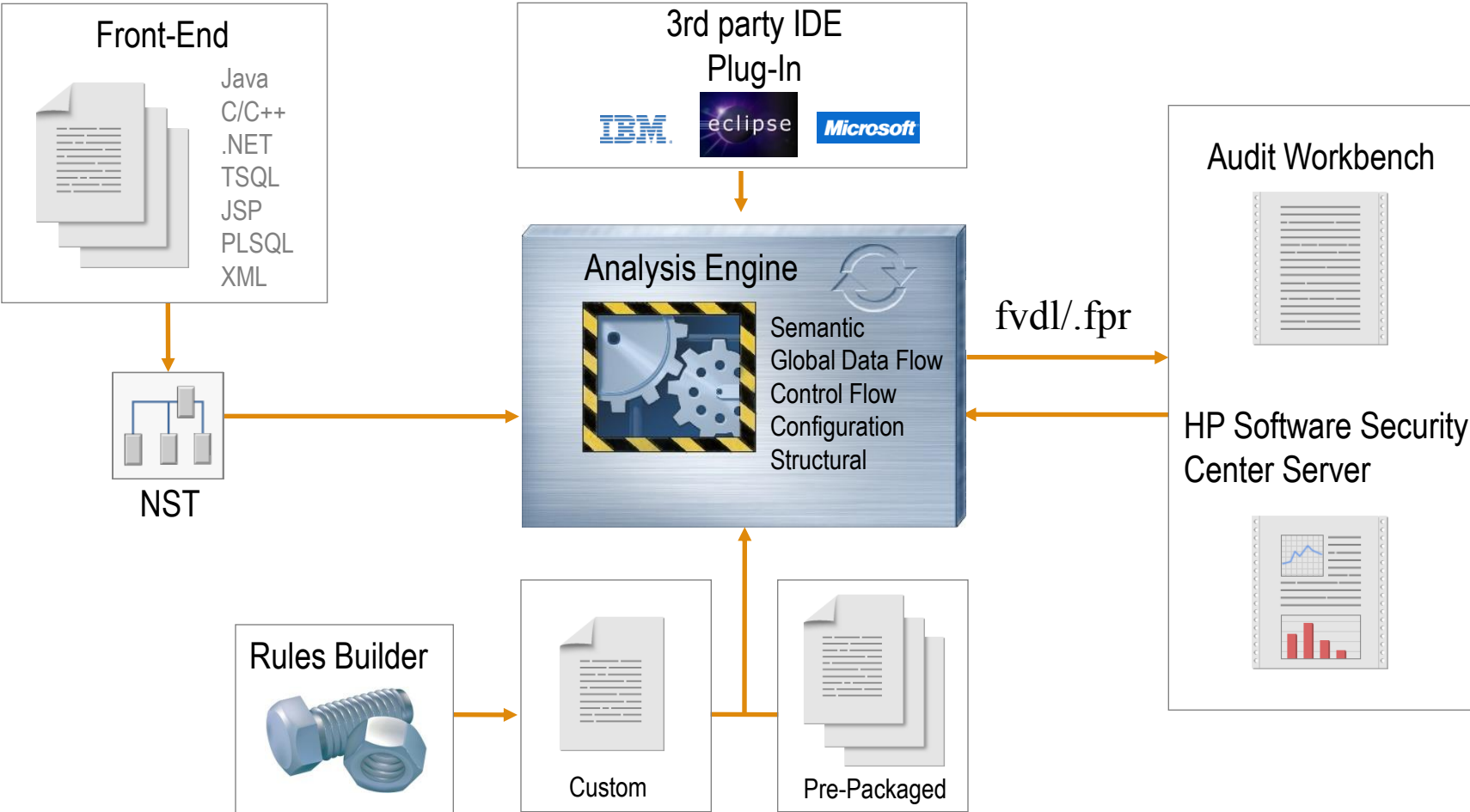
# Fortify SCA 产品组件及功能

- Source Code Analysis Engine (源代码分析引擎)
  - 数据流分析引擎-----跟踪,记录并分析程序中的数据传递过程的安全问题
  - 语义分析引擎-----分析程序中不安全的函数,方法的使用的安全问题
  - 结构分析引擎-----分析程序上下文环境,结构中的安全问题
  - 控制流分析引擎-----分析程序特定时间,状态下执行操作指令的安全问题
  - 配置分析引擎 -----分析项目配置文件中的敏感信息和配置缺失的安全问题
  - 特有的**X-Tier™**跟踪器-----跨越项目的上下层次,贯穿程序来综合分析问题
- Secure Coding Rulepacks™ (安全编码规则包)
- Audit Workbench (审查工作台)
- Custom Rule Editor & Custom Rule Wizard(规则自定义编辑器和向导)
- Developer Desktop (IDE 插件)

# Fortify SCA的关键特点:

- 最庞大的安全编码规则包
- 跨层、跨语言地分析代码的漏洞的产生  
**ABAP、ASP.NET、C,C++、C#、Classic ASP、COBOL、Cold Fusion、Flex/ActionScript、Java、JavaScript/AJAX、JSP、Objective C、PL/SQL、PHP、Python、T-SQL、VB.NET、VBScript、VB6、XML/HTML等语言**
- 精确地定位漏洞的产生的全路径
- 支持不同的软件开发平台  
**Platform: Windows, Solaris, Red Hat Linux, Mac OS X, HP-UX, IBM AIX**  
**IDEs : Visual Studio, Eclipse, JDeveloper , JBuilder , RAD, RSA**

# Fortify SCA 工作原理





# Fortify 漏洞审计---Audit Workbench

File Edit Tools Options Help

**分级报告漏洞的信息**

**项目的源代码**

**漏洞推荐修复的方法**

**漏洞产生的全路径的跟踪信息**

**漏洞的详细说明**

**Recommendations:**  
造成 SQL injection 攻击的根本原因  
在于攻击者可以改变 SQL 查询的  
上下文, 使程序员原本要作为数据解  
析的数值, 被算改为命令了。当构造  
一个 SQL 查询时, 程序员应当清  
楚, 哪些输入的数据将会成为命令的  
一部分, 而哪些仅仅是作为数据。参  
数化 SQL 指令可以防止直接更改上  
下文, 避免几乎所有的 SQL  
injection 攻击。参数化 SQL 指令  
是用常规的 SQL 字符串构造的, 但  
是当需要加入用户输入的数据时, 它  
们就需要使用绑定参数, 这些绑定参  
数是一些占位符, 用来存放随后插入  
的数据。换言之, 绑定参数可以使程  
序员清楚地分辨数据库中的数据, 即  
其中有哪些输入可以看作命令的一  
部分, 哪些输入可以看作数据。这  
样, 当程序准备执行某个指令时, 它  
可以详细地告知数据库, 每一个绑定  
参数所使用的运行时的值, 而不会被  
解析成对该命令的修改。  
  
前面的例子可以改成使用参数化 SQL  
指令的攻击方式(替代用户输入连续  
的字符串), 如下所示:

```
try
{
Statement statement = connection.createStatement
ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
ResultSet results = statement.executeQuery(query);

if ((results != null) && (results.first() == true)
{
ResultSetMetaData resultsMetaData = results.get
ResultSetMetaData(DatabaseUtilities.writeTable(re
resultsMetaData));
results.last();

// If they get back more than one row they s
if (results.getRow() > 1)
{
ma
ge
s
Stage(2);
```

**Abstract:**  
在 `SqlNumericInjection.java` 的第 124 行, `injectableQuery()` 方法调用 SQL 查询, 该 SQL 查询是由未经验证的输入创建的。通过  
这种调用, 攻击者能够修改指令的含义或执行任意的 SQL 命令。

**Explanation:**  
SQL injection 错误在以下情况下出现:  
1. 数据从一个不可信赖的数据源进入程序。  
  
在这种情况下, 数据经由 `ParameterParser.java` 的第 632 行进入 `getParameterValues()`。

RuleID:  
9B5F0161-88EC-4104-B70B-0182FEB53BF2  
Taint Flags: HTTPS, WEB, XSS  
Direct Function Call:  
java.sql.Statement.executeQuery()

# Audit Workbench---Audit

The screenshot displays the Fortify Audit Workbench interface. At the top, the title bar reads "Audit Workbench - WebGoatDeveloper5.1Scan - [D:\JAVA Project\WebGoatDeveloper5.1\webgoatDev5.1.fpr]". The menu bar includes "File", "Edit", "Tools", "Options", and "Help". The main window is divided into several sections:

- Left Panel:** Shows a "Filter Set" dropdown set to "Broad". Below it, a summary of findings: 118 Hot, 478, 650, 0, and 1246. A "Group By" dropdown is set to "Category". A tree view lists various vulnerability categories and specific instances, such as "SQL Injection - [6 / 18]" and "SqlNumericInjection.java:124 (SQL I)".
- Code Editor:** Displays a snippet of Java code from "SqlNumericInjection.java":

```
119     try
120     {
121         Statement statement = connection.createStatement(
122             ResultSet.TYPE_SCROLL_INSENSITIVE
```
- Filters Panel:** A "Filters" section is visible, with a red circle around the "Filters" label. A red arrow points to the filter configuration. The configuration shows "If" conditions: "confidence is in range [4.0,5.0]" and "severity is in range (3.0,5.0)". The "Then" section has "Set Folder to:" set to "Hot". A red circle highlights the "Set Folder to:" dropdown menu, which is open, showing options like "Hot", "Warning", "Info", "wanghong", and "Other Folder...".
- Control Flow Graph (CFG):** A diagram showing the flow of execution. It includes nodes for "SqlNumericInjection.injectableQuery", "ParameterParser.getRawParameter", and "SqlNumericInjection.executeQuery(0)". Red arrows highlight the flow from the "Return" node (613) back to the "executeQuery(0)" node (124), which is labeled as a "sink".
- Analysis Trace:** A list of events from the analysis, including "getParameterVal", "Assignment to v", "Return values[C", "getRawParameter", "Return", "getRawParam", "Assignment", "Assignment", and "executeQuer".

Red annotations in Chinese provide additional context:

- A red circle around the "Filters" label is accompanied by the text: "通过过滤器的confidence/severity可以将漏洞放到不同的级别中，用户也可以自定义漏洞级别。" (Through the confidence/severity of the filter, vulnerabilities can be placed in different levels, and users can also customize the vulnerability level.)
- A red circle around the "Set Folder to:" dropdown is accompanied by the text: "将漏洞产生的过程图形化表现出来" (Visualize the process of vulnerability generation).

# Audit Workbench---Report

**Generate Report...**

Report: Fortify Security Report

**Executive Summary**

- Executive Summary
- Project Summary
- Results Outline
- Detailed Project Summary
- Issue Count by Category
- Issue Breakdown by Analysis
- Issue Breakdown by OWASP T
- New Issues

Issues Overview

This section provides an overview of the issues uncovered during analysis. The report covers a summary of vulnerability categories discovered by the tool. The auditor should augment this section with higher-level conclusions derived from human review of the application (including architecture reviews, black-box testing, compliance issues, etc.)

► Edit Text

**Save Report ...**

Title: Fortify Security Report

Author: hong

Footnote: Copyright 2007 Fortify Software Inc.

Location: C:\DOCUME~1\hong\LOCALS~1\Temp\HacmeBooks2 Browse ...

Format: PDF Report

- PDF Report
- XML Report
- RTF Report

Save Cancel

Save Settings as Default

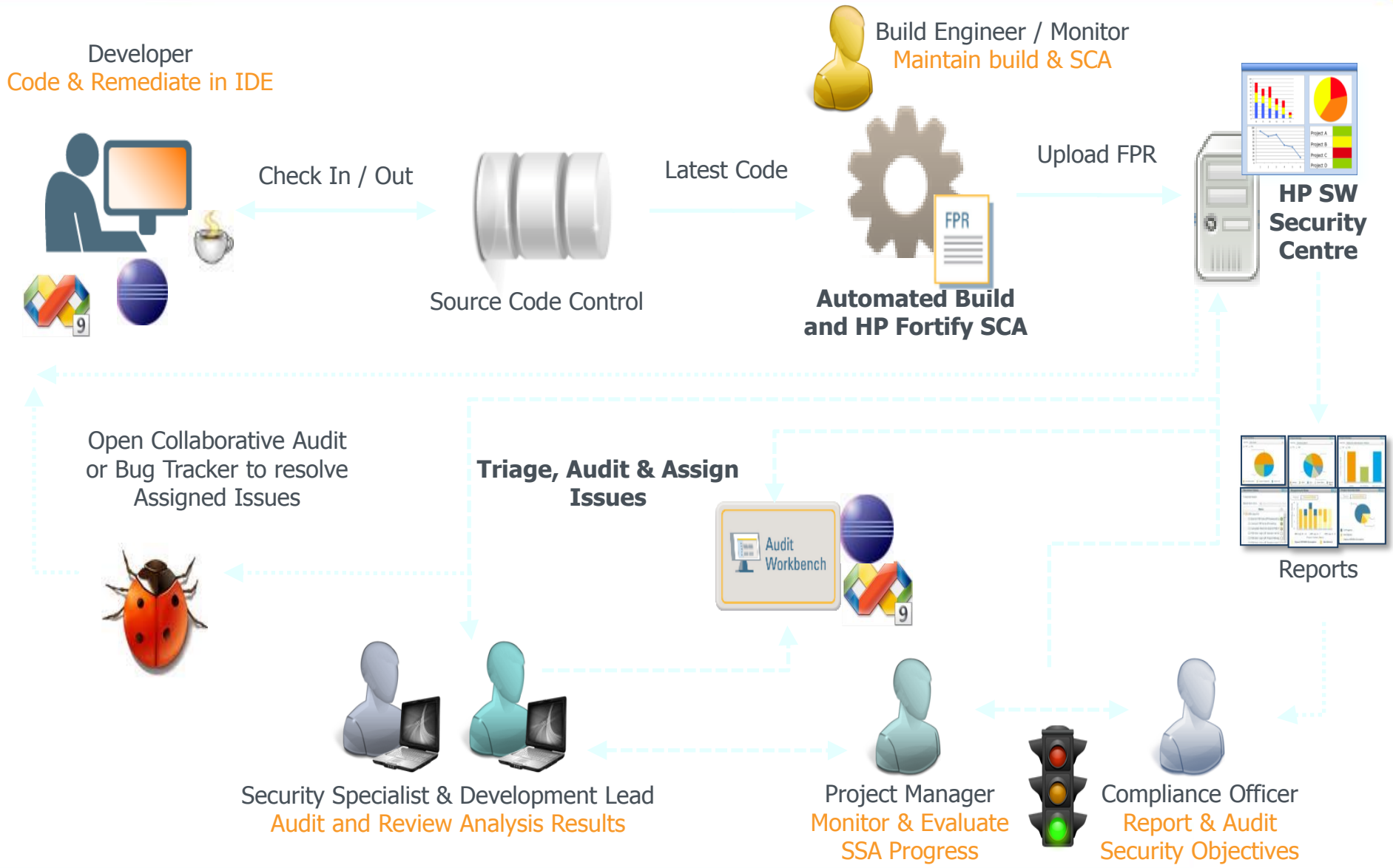
Save as New Template...

Save Report... Cancel

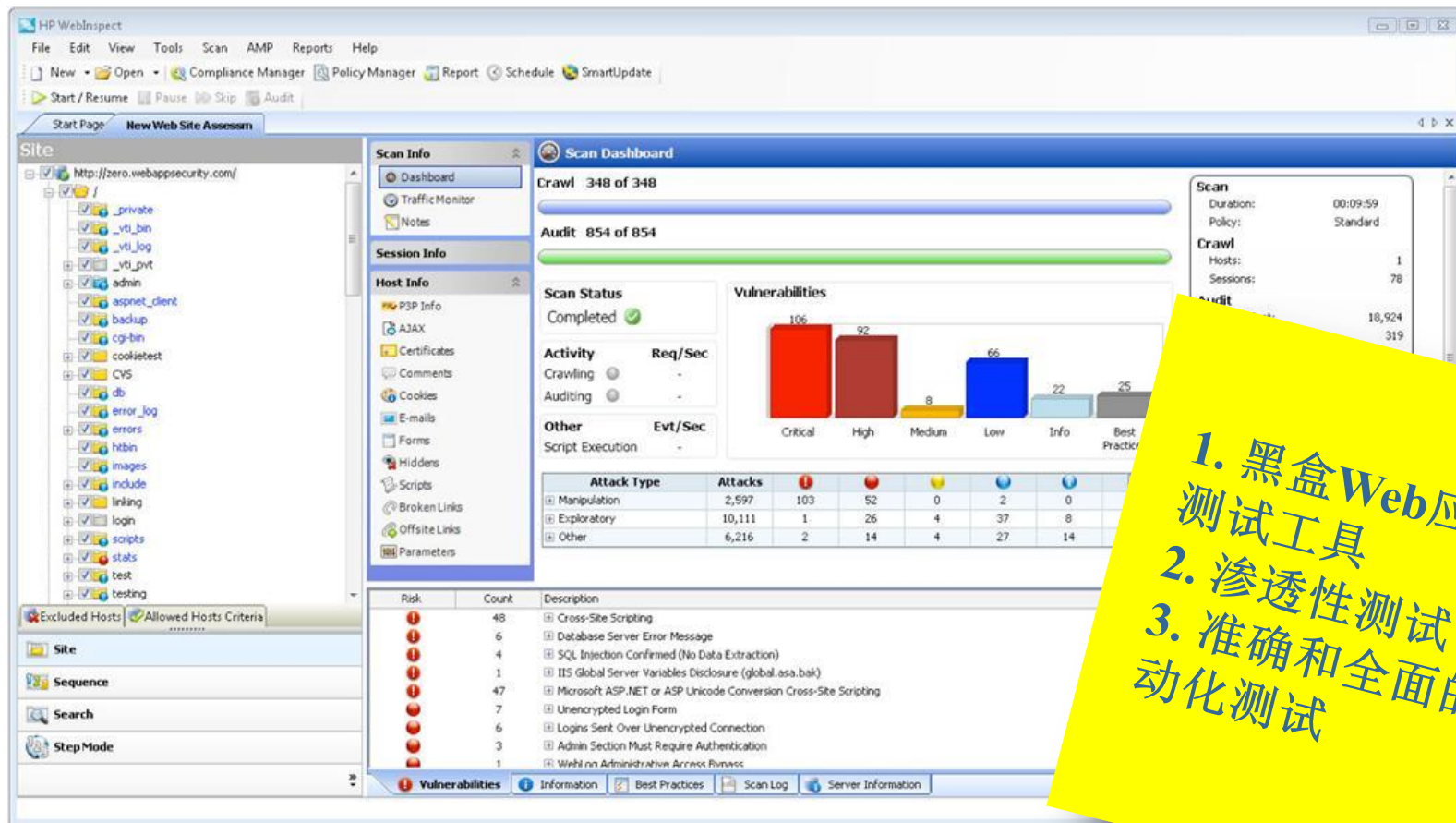
Selections in the image:

- 选取报告内容: Points to the 'Report:' dropdown.
- 两种报告模板: Points to the 'Executive Summary' and 'Issues Overview' sections.
- 选择报告格式: Points to the 'Format:' dropdown in the 'Save Report ...' dialog.

# HP Fortify SCA Remediation Process



# HP WebInspect - 动态应用安全测试工具



1. 黑盒Web应用安全测试工具
2. 渗透性测试
3. 准确和全面的自动化测试

# HP WebInspect

- 业界领先的自动化应用安全测试解决方案
- 对Web、WebServices应用进行安全检测
- 抓取 ( crawl ) 与审计 ( audit )
- 自动更新安全规则
- 自动产生缺陷报告和详细修复建议

The screenshot displays the HP WebInspect application interface. The top window shows a list of scan results with columns for Scan Name, Start URL, Vulnerabilities, Duration, Crawl Status, Audit Status, and Scan Date. Below this, a 'Scan Dashboard' provides a detailed overview of the current scan, including 'Crawl 200 of 200', 'Audit 65 of 616', and 'Scan Status Running'. A 'Vulnerabilities' bar chart shows counts for Critical, High, Medium, Low, Info, and Best Practices. A table lists various attack types such as Manipulation, Cookie Injection, and SQL Injection, along with their counts. A 'Risk' table at the bottom lists specific vulnerabilities with their counts and descriptions.

Attack Type	Attacks	Critical	High	Medium	Low	Info	Best Practices
Manipulation	1,438	1	2	0	0	0	0
Cookie Injection	315	0	0	0	0	0	0
Cross Site Scripting	94	4	0	0	0	0	0
Header Injection	2	0	0	0	0	0	0
Server Include	141	0	0	0	0	0	0
Adaptive Agents	34	0	0	0	0	0	0
LFI Agent	297	0	1	0	0	0	0
Port Injection	126	0	1	0	0	0	0
Keyword Search	0	3	0	0	3	0	0
SQL Injection	152	2	0	0	0	0	0
Query Injection	147	0	0	0	0	0	0
Exploitative	2,429	0	27	1	39	4	3
Adversarial Agents	44	0	7	0	0	1	1

Risk	Count	Description
3	1	Database Server Error Message
2	2	SQL Injection Confirmed (No Data Extraction)
4	1	Cross-Site Scripting
1	1	Header Arbitrary File Retrieval
1	1	WebLog Administrative Access Bypass
1	1	HTTP Header CR/LF Injection (HTTP Response Splitting)
1	1	Unencrypted Login Form
3	1	Backup File (Appended .bak)
4	1	Possible ASP.NET Source Code Disclosure
3	1	Backup File (Appended.BAK)
2	1	Logon Sess Over/Unencrypted Connection
1	1	Backup File (Appended.cab)

# HP Web Inspect

- 对**Web**应用技术的广泛支持
  - AJAX、JavaScript、Flash、Silverlight、Web Services等
- 创建宏以记录检测步骤，实现重复性检测的自动化
- 同步扫描与审计和智能引擎
- 同时启动和管理多个扫描进程，从而增加检测量
- 自带多种渗透检测工具，从而对所发现的应用安全漏洞进行再验证
- **WebInspect smart update**安全策略的支持力度与更新频率很高。
- 快速分析应用与**web** 服务
- **WebInspect**较其他产品，具备更全面的安全高级工具集成以及更深度的安全评估报告。以各种标准格式（**HTML**、**PDF**、**RTF**、**XML**、**TXT**以及**XLS**等格式）导出结果报告



# HP Software Security Center Server

**FORTIFY 360 Server** Welcome admin  
[Logout](#) | [Account](#) | [Preferences](#) | [About](#)

Dashboard | Projects | Reports | Administration

### Alerts

[Manage](#) | [Preferences](#)

Show Read Alerts

4 records found

Select item and...

Date	Project Ver	Alert Trigg	Current Value
01-20-200	SCA - Sofi	Allocate TI	Awaiting Sign C
01-15-200	SCA - Sofi	Allocate ti	Awaiting Sign C
01-15-200	SCA - Sofi	Allocate TI	Awaiting Sign C
01-15-200	SCA - Sofi	Allocate TI	Document Reje

### Document Status

Select item and...

Name
Webgoat-5
SPLC-1
Asterisk-1.2.10
SCA-Sofia
Riches Bank-1.0

### Issues

Trend  Current Issues

Group By  Fortify Priority Order  Analyzer

Number of Issues

Legend: High (Red), Low (Yellow), Medium (Orange)

### Requirement State

Trend  Current State

Number of Requirements

### Audit Status

6 records found

Select item and...

Project Version	Last Upload	Total Issu	Audit Perc
Asterisk - 1.2.10	01-22-2009...	968	0.21%
Riches Bank - 1	01-22-2009...	14	0.00%
SCA - Sofia	01-15-2009...	67	0.00%
SPLC - 1	01-26-2009...	204	0.00%
Sample - 1	01-16-2009...	22	0.00%
Webgoat - 5	01-16-2009...	277	0.00%

### Project Meta Data

View By

Legend: Unix (Orange), Platform Neutral (Green)

### Project Security State

Trend  Current State

Legend: In Progress (Blue), Not Started (Orange)

1. 软件安全工作管理平台
2. 及时了解所有部门的安全工作状况及趋势
3. 与其它Fortify产品结合，制定安全策略，保障软件安全



# HP Software Security Center Server: 软件安全管理平台

- 软件安全管理器是软件安全分析、管理的综合平台。

帮助软件开发的管理人员统计和分析软件安全的风险、趋势，跟踪和定位软件安全漏洞，提供足够多的软件安全质量方面的真实的状态信息以便于管理人员制定安全管理决策及编码规则。

- 其主要特点：

1. 集中管理
2. 化分优先级
3. 标记趋势
4. 协同工作平台
5. 产生多种报表

# Fortify RTA – Application Deployment Protection

### Security Events

Search Export Search Filter: None

Chart View List View

Application All Date Range All All Dates

1178 records found < < 1 - 50 of 1178 > >

Select item and... View Details Add Event Handler

		Application	Host	Category	Action	Request IP Address	Date
!		Riches DotNet	localhost	SQL Injection	display (default)	10.100.100.137	01/10/2012 8:52:30 PM
		Riches DotNet	localhost	SQL Injection	display (default)	10.100.100.137	01/10/2012 8:52:29 PM
		Riches DotNet	localhost	SQL Injection	display (default)	10.100.100.137	01/10/2012 8:52:13 PM
		Riches DotNet	localhost	SQL Injection	display (default)	10.100.100.137	01/10/2012 8:52:10 PM
		Riches Java	localhost	Probing: Command Inj	<none>	10.100.100.137	01/10/2012 8:52:09 PM
		Riches Java	localhost	Probing: Command Inj	<none>	10.100.100.137	01/10/2012 8:52:09 PM
		Riches Java	localhost	Probing: Command Inj	<none>	10.100.100.137	01/10/2012 8:52:09 PM
		Riches DotNet	localhost	Poor Error Handling: U	display (default)	10.100.100.137	01/10/2012 8:51:58 PM
		Riches Java	localhost	Cross-Site Scripting: R	display (default)	10.100.100.137	01/10/2012 8:51:57 PM
		Riches Java	localhost	Cross-Site Scripting: R	display (default)	10.100.100.137	01/10/2012 8:51:49 PM
		Riches Java	localhost	Cross-Site Scripting: R	display (default)	10.100.100.137	01/10/2012 8:51:49 PM
		Riches Java	localhost	Cross-Site Scripting: R	display (default)	10.100.100.137	01/10/2012 8:51:47 PM
		Riches Java	localhost	Brute Force Login Atter	<none>	10.100.100.137	01/10/2012 8:51:29 PM
		Riches Java	localhost	Brute Force Login Atter	<none>	10.100.100.137	01/10/2012 8:51:29 PM
		Riches Java	localhost	Brute Force Login Atter	<none>	10.100.100.137	01/10/2012 8:51:29 PM
		Riches Java	localhost	Brute Force Login Atter	<none>	10.100.100.137	01/10/2012 8:51:29 PM
		Riches Java	localhost	Brute Force Login Atter	<none>	10.100.100.137	01/10/2012 8:51:29 PM
		Riches Java	localhost	Brute Force Login Atter	<none>	10.100.100.137	01/10/2012 8:51:29 PM
		Riches Java	localhost	Brute Force Login Atter	<none>	10.100.100.137	01/10/2012 8:51:29 PM
		Riches Java	localhost	Brute Force Login Atter	<none>	10.100.100.137	01/10/2012 8:51:29 PM

**SQL Injection**  
Critical

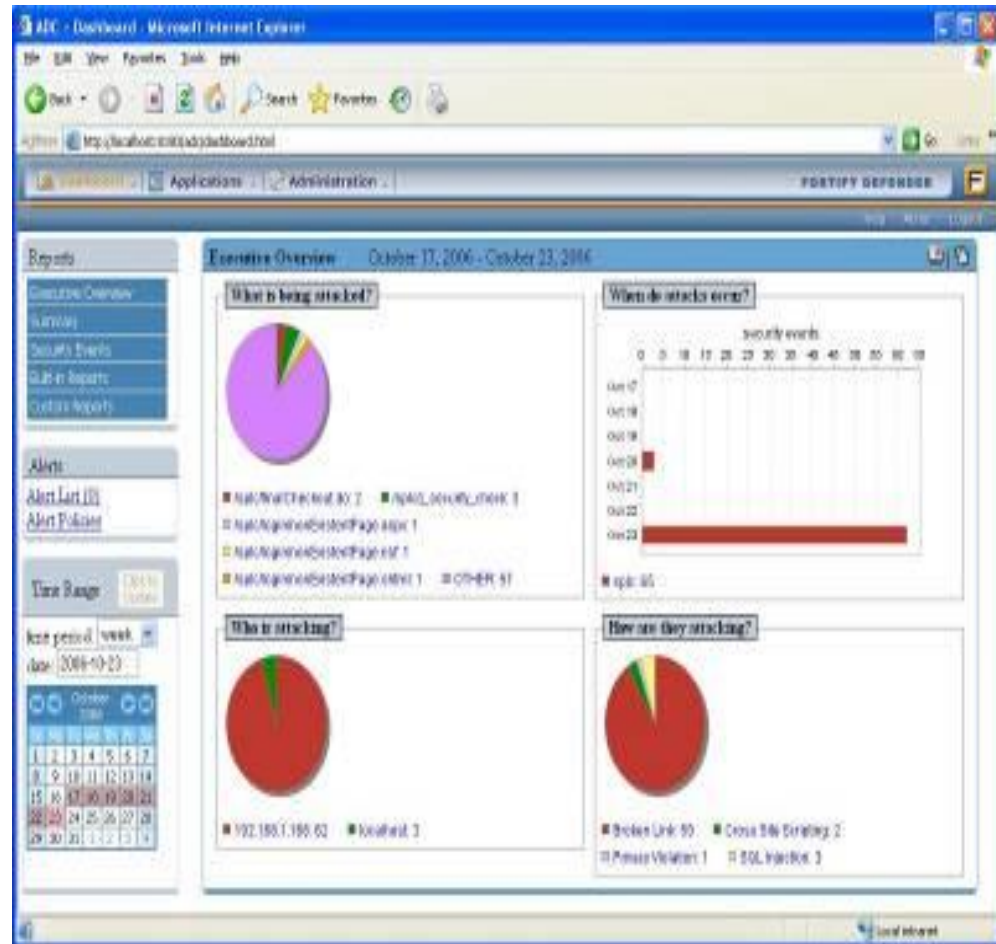
[Event Details](#) [Category Description](#)

Action  
Request IP Address  
Referer  
Request Path

1. 运行时刻保护并  
监控  
2. 无需源代码，部署  
简单  
3. 拦截恶意数据，回  
应恶意挑战，主动  
防御

# Fortify RTA – 保护运行时的 Web 系统

- 在不可以修改源代码的情况下，实为最理想的解决办法
- 直接安装在应用系统二进制码上 —— 完全不需要源代码
- 快、易于安装及部署
- 为大部份已知的漏洞提供防御



# Fortify OnDemand – Security Testing Service

**Fortify on Demand**

Welcome danumudu-emp - BigBankESMP  
[Logout](#) [Account](#) [Preferences](#) [Terms Of Use](#) [Contact](#)

**Projects** | Reports | Administration

**Projects**

Search Filter: None  
 1 record found

**Executive Summary**

**Company:** BigBankESMP  
**Project:** SPLC  
**Version:** 1.0  
**Static Analysis Date:** July 15, 2009  
**Dynamic Analysis Date:** N/A

**Fortify Security Rating**  
 ★★★★★  
 64 issues  
 Based on impact and likelihood of issues (see Appendix A).  
**Static:** ✓ **Dynamic:** ✗

**Application Type:** E-Commerce  
**Technology Stack:** Java/J2EE  
**Interfaces:** Web Services (SOA), Web Access

**Project Type:** Application  
**Data Classification:** Customer personally identifiable

**Top 5 Prevalent Categories**

- Cross-Site Scripting: 35
- Cross-Site Request Forgery: 13
- SQL Injection: 5
- SQL Injection: Hibernate: 3
- Unreleased Resource: Streams: 3
- Other: 5

**Issues by Impact and Likelihood**

High Impact	44	Critical
Low Impact	13	Medium
		7

**Issues by Attack Vector**

Attack Vector	Issues
Database	2
Network	0
Web	41
Web Service	0
Other	21
<b>Total</b>	<b>64</b>

**Remediation Roadmap**

	To Achieve	Major Fixes	Minor Fixes
Database	★★★★★	0	0
Network	★★★★★	0	0
Web	★★★★★	0	44
Web Service	★★★★★	0	0
Other	★★★★★	0	0
<b>Total</b>	<b>Total</b>	<b>0</b>	<b>44</b>

**Fortify** 软件安全中心  
 产品的基础上利用云  
 技术  
 开发的一种“安全即服  
 务”解决方案

**Thanks!**

