



IBM Software Group

開発工程からのセキュリティ検査

安心・安全なWebアプリケーションを実現する Rational ソリューション

2008/1

Rational software

→ Go to IBM

© 2007 IBM Corporation

Agenda

- Webアプリケーションのセキュリティ問題
- Webアプリケーションのセキュリティ対策の現状
- AppScan の紹介
- 開発工程におけるセキュリティ検査





IBM Software Group

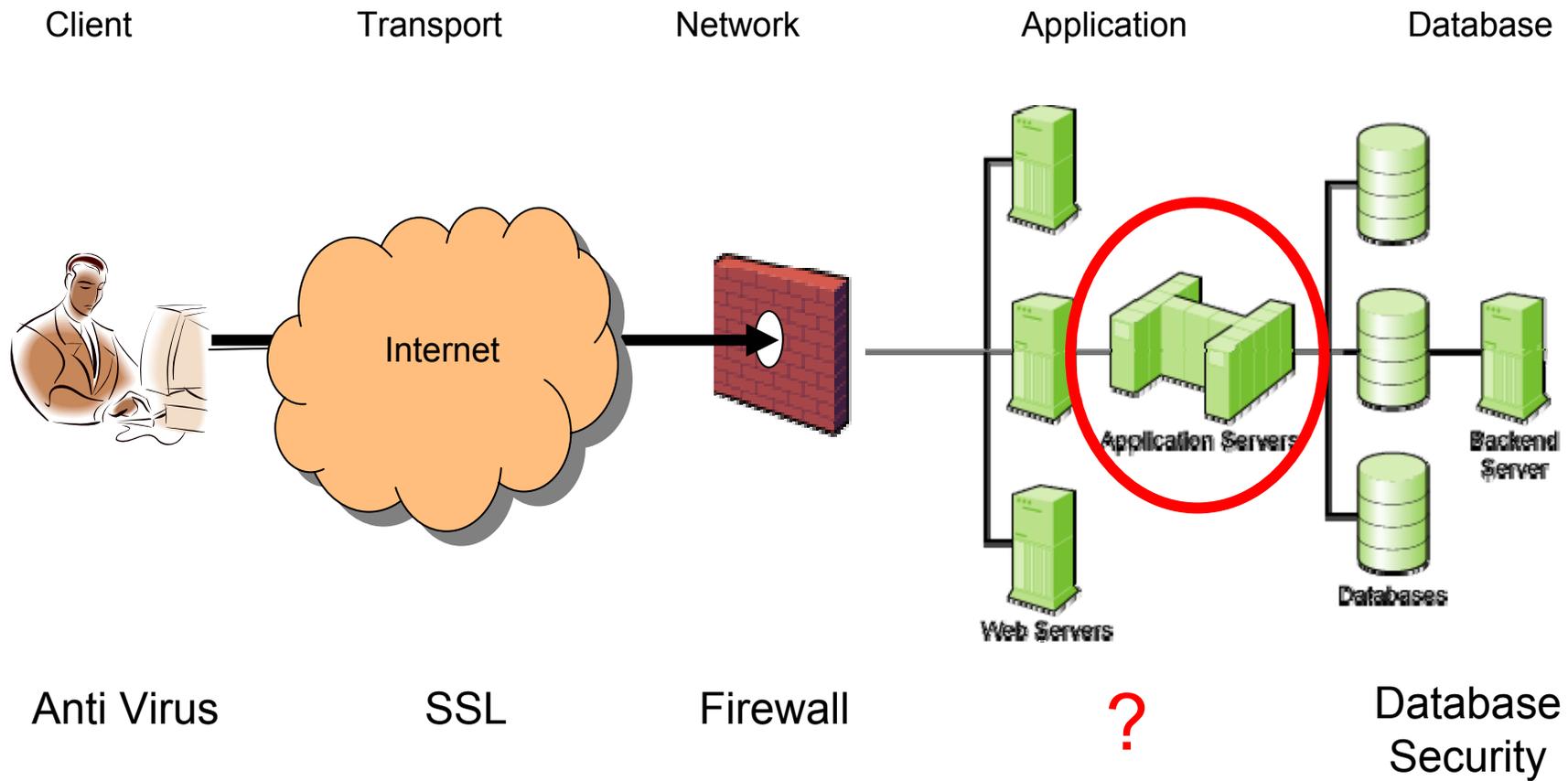
Web アプリケーションのセキュリティ問題

Rational software

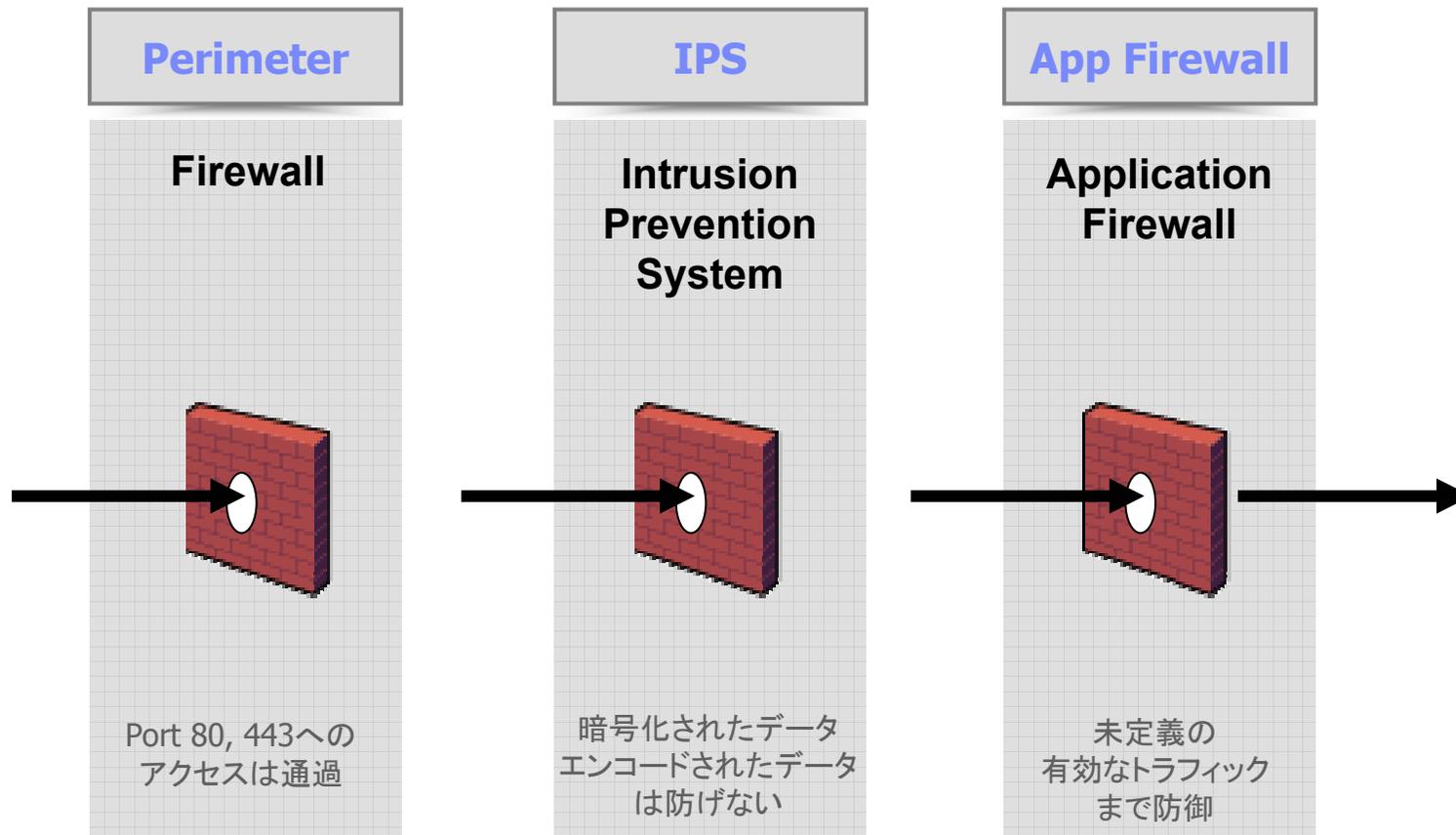
→ Go to **IBM**

© 2007 IBM Corporation

Web アプリケーション アーキテクチャ



Firewall によるプロテクション



Web アプリケーションに対する攻撃

- SQLインジェクション (SQL Injection)
- クロスサイトスクリプティング (Cross-Site Scripting – XSS)
- パラメータの改竄 (Parameter Tampering)
- Hiddenフィールドの不正操作 (Hidden Field Manipulation)
- クッキーの濫用 (Cookie Poisoning)
- バッファオーバーフロー (Buffer Overflow)
- 強制ブラウジング (Forceful Browsing)
- ステルスコマンド (Stealth Command)
- バックドア(デバッグオプションなど) (Debug Option and Backdoor)
- 設定ミスや既知の脆弱性を利用 (Know Vulnerability)



SQL インジェクション

```
SELECT * FROM users WHERE  
username='jsmith' AND password=' OR '1' = '1'
```

The screenshot shows the 'Online Banking Login' page. The page has a navigation bar with 'PERSONAL', 'SMALL BUSINESS', and 'INSIDE AL'. A sidebar on the left lists various services under 'PERSONAL' and 'SMALL BUSINESS'. The main content area is titled 'Online Banking Login' and contains a login form with the following fields:

- Username:
- Password:
- Login button

A blue arrow points from the password field to the text `' OR '1' = '1'`, indicating the injected payload.

SQL インジェクション

Altoro Mutual: Online Banking Home - Microsoft Internet Explorer

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)

戻る 戻る 戻る 戻る 戻る 戻る 戻る

アドレス(D) http://localhost/bank/main.aspx

[Sign Off](#) | [Contact Us](#) | [Feedback](#) | Search

AltoroMutual

[MY ACCOUNT](#) [PERSONAL](#) [SMALL BUSINESS](#) [INSIDE AL](#)

I WANT TO ...

- [View Account Summary](#)
- [View Recent Transactions](#)
- [Transfer Funds](#)
- [Search News Articles](#)
- [Customize Site Language](#)

Hello, John Smith

Welcome to Altoro Mutual Online.

View Account Details:

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of \$10000.00!

Click [Here](#) to apply.



SQL インジェクション テストパターン (一部)

- Cookie を汚染する SQLインジェクション
 - Cookie の値を以下のように設定します: '
 - Cookie の値を以下のように設定します: \'
 - Cookie の値を以下のように設定します: ;
 - Cookie の値を以下のように設定します: "
 - Cookie の値を以下のように設定します: \'
 - Cookie の値を以下のように設定します:)
- HTTP ヘッダーの SQLインジェクション
 - HTTP 'X-Forwarded-For' ヘッダの値を以下に変更します: '
 - HTTP 'X-Forwarded-For' ヘッダの値を以下に変更します: %27
 - HTTP 'X-Forwarded-For' ヘッダの値を以下に変更します: %2527
 - HTTP 'X-Forwarded-For' ヘッダの値を以下に変更します: "
 - HTTP 'X-Forwarded-For' ヘッダの値を以下に変更します: %c0%a7
 - HTTP 'Referer' ヘッダの値を以下に変更します: '
 - HTTP 'Referer' ヘッダの値を以下に変更します: %27
 - HTTP 'Referer' ヘッダの値を以下に変更します: %2527
 - HTTP 'Referer' ヘッダの値を以下に変更します: "
 - HTTP 'Referer' ヘッダの値を以下に変更します: %c0%a7
- Probe SQL
- SQLインジェクション
 - 元のパラメータ値に以下の文字列を追加します: ;
 - 元のパラメータ値に以下の文字列を追加します: having 1=1--
 - 元のパラメータ値に以下の文字列を追加します: 1 having 1=1--
 - 元のパラメータ値に以下の文字列を追加します: \' having 1=1--
 - 元のパラメータ値に以下の文字列を追加します:) having 1=1--
 - Append the following string to the original parameter value: %a5' having 1=1--
 - 元のパラメータ値に以下の文字列を追加します: %uFF07
 - 元のパラメータ値に以下の文字列を追加します: '
 - 元のパラメータ値に以下の文字列を追加します: ; select @@version,1,1,1--
 - 元のパラメータ値に以下の文字列を追加します: ; select * from master..sysmessages--
 - 元のパラメータ値に以下の文字列を追加します: ; select * from dbo.sysdatabases--
 - 元のパラメータ値に以下の文字列を追加します: ; select * from sys.dba_users--
 - SQLインジェクション (ポートリスナー テスト - Oracle バリエーション)
 - ポート リスナーおよび OpenRowSet (MS-SQL) を使用した SQLインジェクション
- SQLインジェクション コマンドの実行
 - 元のパラメータ値に以下の文字列を追加します: %20exec%20master..xp_cmdshell%20'vol'--
 - 元のパラメータ値に以下の文字列を追加します: '; exec master..xp_cmdshell 'echo [param name] in [path] is vulnerable >> C:\AppScanSQLTest.txt'-- (SQL Injection Probe)
 - 元のパラメータ値に以下の文字列を追加します: exec master..xp_cmdshell 'echo [param name] in [path] is vulnerable >> C:\AppScanSQLTest.txt'-- (SQL Injection Probe)
 - SQLインジェクションによるコマンドの実行 (ポートリスナー - MS-SQL バリエーション)

XSS テストパターン (一部)

```

>%22%27><img%20src%3d%22javascript:alert([VARIANT ID])%22>
>"><img%20src%3D%26%23x6a;%26%23x61;%26%23x76;%26%23x61;%26%23x73;%26%23x63;%26%23x72;%26%23x69;%26%23x7
%22%20style%3D%22background:url(javascript:alert([VARIANT ID])%22%20OA%3D%22
--><script>alert([VARIANT ID])</script>
[ORIGINAL VALUE]+'+alert([VARIANT ID])+'
[ORIGINAL VALUE]%27%2Balert%28[VARIANT ID]%29%2B%27
[ORIGINAL VALUE]" +alert([VARIANT ID])+"
[ORIGINAL VALUE]%22%2Balert%28[VARIANT ID]%29%2B%22
>'><script>alert([VARIANT ID])</script>
>"><script>alert([VARIANT ID])</script>
>'><%00script>alert([VARIANT ID])</script>
>"><%00script>alert([VARIANT ID])</script>
"><STYLE>@import"javascript:alert([VARIANT ID])";</STYLE>
"></IFRAME><script>alert([VARIANT ID])</script>
"></style><script>alert([VARIANT ID])</script>
"></title><script>alert([VARIANT ID])</script>
\u003Cscript\u003Ealert\u0028[VARIANT ID]\u0029\u003C/script\u003E
</TextArea><script>alert([VARIANT ID])</script>
>+ACJ++AD4APB-SCRIPT+AD7-alert(1234)+ADz-/SCRIPT+AD7-
>+ACJ++AD4APB-SCRIPT+AD7-alert(1234)+ADz-/SCRIPT+AD7-
onMouseOver=alert([VARIANT ID])><
%27%20onMouseOver=alert([VARIANT ID])><
%22%20onMouseOver=alert([VARIANT ID])><
[ORIGINAL VALUE]+alert([VARIANT ID])+
%A7%A2%BE%Bc%F3%E3%F2%E9%F0%F4%Be%E1%EC%E5%F2%F4%A8[VARIANT ID]%A9%Bc%Af%F3%E3%F2%E9%F0%F4%Be
[ORIGINAL VALUE]%22)%0d%0aalert([VARIANT ID])%27
<script>alert([VARIANT ID])</script>

```



参考

- Watchfire CBT
 - ▶ "Web Application Security Hacking 101"
- Watchfire ホワイトペーパー
 - ▶ 「アプリケーションレベルでの一般的な 12 種類のハッカー攻撃」
<https://www.watchfire.com/securearea/whitepapers.aspx?id=83>
- 外部情報
 - ▶ WASC 脅威の分類
<http://www.webappsec.org/projects/threat/>
 - ▶ OWASP
http://www.owasp.org/index.php/OWASP_Top_Ten_Project
 - ▶ IPA
http://www.ipa.go.jp/security/vuln/20050623_websecurity.html





IBM Software Group

Web アプリケーションセキュリティ対策の現状

Rational software

→ Go to IBM

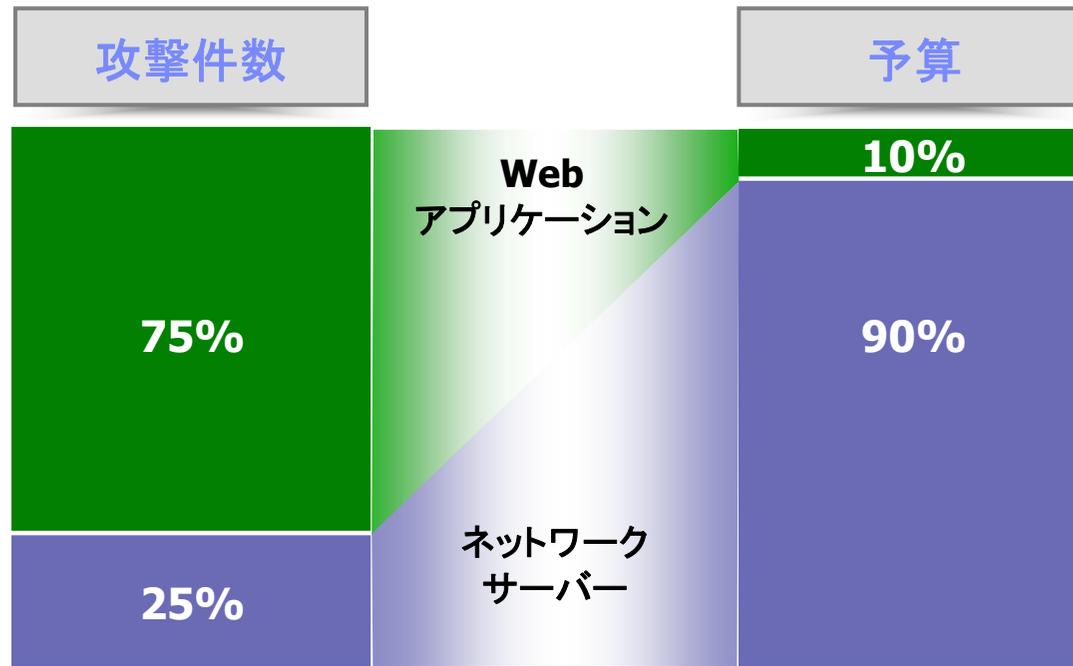
© 2007 IBM Corporation

アプリケーション セキュリティが重要な理由

- Webアプリケーションは、ハッカーの間で最も注目されている
 - ▶ 75% の攻撃はアプリケーション層で見ついている (Gartner)
 - ▶ XSSとSQLインジェクションは、報告されている脆弱性の#1と#2である (Mitre)
- ほとんどのWebサイトは脆弱である
 - ▶ 90% のWebサイトはアプリケーションの脆弱性がある (Watchfire)
 - ▶ 78% の容易に利用可能な脆弱性は、Webアプリケーションにある (Symantec)
 - ▶ 80% の組織は、2010年までにアプリケーションセキュリティ事件に遭遇する (Gartner)
- Webアプリケーションはハッカーにとって価値のあるターゲット
 - ▶ 顧客データ、クレジットカード、IDの盗難、詐欺、サイトの改竄、など
- コンプライアンスの要求
 - ▶ GLBA、HIPAA、FISMA、PCI データセキュリティ標準、SOX、個人情報保護法



Web アプリケーションに対する攻撃の増加と対策予算



75% の情報セキュリティ攻撃は、アプリケーション層で起こっている

2/3 のWebアプリケーションは、脆弱性がある

Sources: Gartner



Web アプリケーション開発におけるセキュリティ

64% の開発者は、セキュアなアプリケーションを書く自信が無い

Microsoft Developer Research

70% の企業は、SDLC にセキュアなアプリケーションを書くための技術を導入していない

Aberdeen Group, May 2007

90% のアプリケーションには脆弱性がある

Watchfire



セキュリティ対策の手法

手法	利点	欠点	
検査サービス	<ul style="list-style-type: none"> ■ 専門家による詳細な調査 	<ul style="list-style-type: none"> ■ コストが高いため、検査頻度、検査範囲が限られる ■ 運用開始後のテストではサービスに影響も 	
Web アプリケーション ファイアウォール	<ul style="list-style-type: none"> ■ 既存のアプリケーションに対してしても対応可能 	<ul style="list-style-type: none"> ■ アプリケーションの更新に伴う設定変更が煩雑 ■ ネットワークの経路上に置かれる 	
脆弱性検査ツール	ホワイトボックス	<ul style="list-style-type: none"> ■ 問題の修正位置が的確に把握できる 	<ul style="list-style-type: none"> ■ ソースコードが必要 ■ 実際には脆弱性でないものも指摘 ■ プラットフォーム依存の問題は指摘できない
	ブラックボックス	<ul style="list-style-type: none"> ■ 現実に即した検査 ■ アプリケーションに関する詳細な知識は不要 	<ul style="list-style-type: none"> ■ アプリケーションが稼動する必要がある ■ 実環境では副作用も



IBM Software Group

AppScan

Rational software

→ Go to **IBM**

AppScan

- Webアプリケーション セキュリティ テスト ツール
 - ▶ Webアプリケーションの脆弱性と、インフラ(OS、Webサーバー等)の設定ミス、既知の問題を検知
 - ▶ テストを自動化し、手作業に比べて圧倒的な時間とコストの削減が可能
 - ▶ 脆弱性の指摘、修正方法の提示、レポートの作成
 - ▶ ブラックボックス テスト
 - ハッカーの視点でセキュリティ問題を検査 - **"Hacker in the box"**
 - "現実的な" 問題点の指摘 - **"Low hanging fruits"**





Demo



AppScan 7.7 - ウィザードによる容易な設定



AppScan 7.7 - スキャン エキスパート(New)

The screenshot displays the IBM Rational AppScan 7.7 interface. The main window is titled "タイトルなし - IBM Rational". The menu bar includes "ファイル (F)", "編集 (E)", "表示 (V)", "スキャン (S)", "ツール (T)", and "ヘルプ (H)". The toolbar contains icons for "スキャン", "一時停止", "マニュアル探索", "スキャン設定", "スキャン エキスパート (P)", "スキャン ログ", "レポート", and "更新".

The left sidebar shows a "表示" (View) section with icons for "セキュリティ問題", "修復作業", and "アプリケーション データ". The main content area is titled "スキャン エキスパート評価スキャン..." and shows a progress bar at 5% completion. Below the progress bar, it displays the URL "http://localhost/altoro/default.aspx?content=personal_other.htm" and a timer "00:23".

The "スキャン エキスパート" section contains the following text:

スキャン エキスパートを使用して、スキャン設定を最適化し、スキャン効率を向上させることができます。

スキャン エキスパートは、アプリケーションを探索して情報およびネットワーク動作パターンを収集しています。このフェーズの後、結果を分析し、スキャン効率を高めるために現在のスキャン設定に対する変更を推奨します。

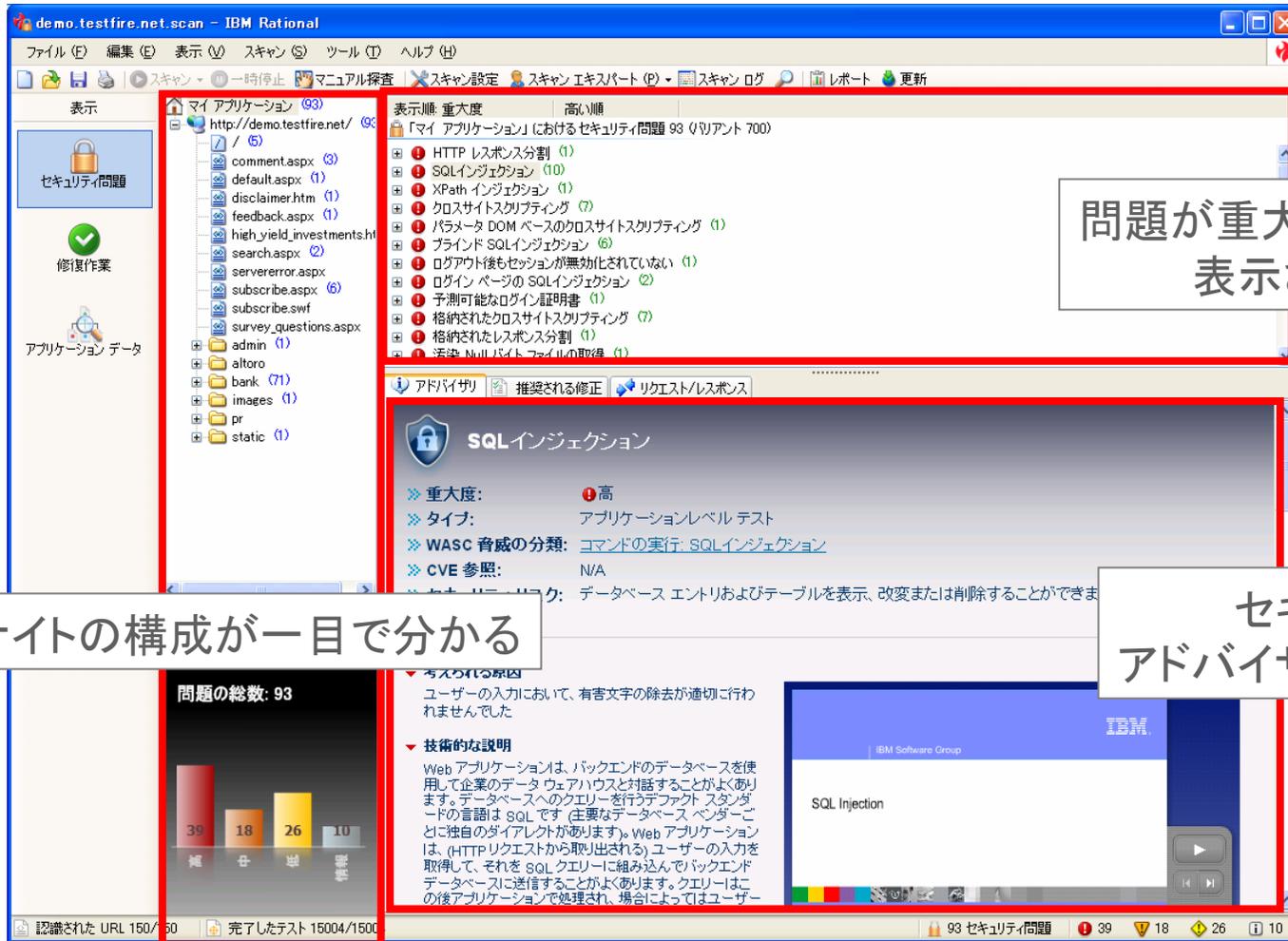
これらの推奨を確認し、適用する推奨を決定することができます。その後、AppScan はメイン スキャンに進みます。

Below the text, there is a "表示順" (Sort) section with "重大度" (Severity) and "高い順" (High) options. The "アドバイザー" (Advisor) section is active, showing "推奨される修正" (Recommended fixes) and "リクエスト/レスポンス" (Request/Response).

At the bottom left, a "重大度ゲージを発行" (Issue severity gauge) section shows "問題の総数: 0" (Total number of issues: 0).

A white box with a black border is overlaid on the bottom right of the screenshot, containing the text: "スキャン効率を高める設定を推奨します" (Recommend settings to improve scan efficiency).

AppScan 7.7 : 見易い GUI

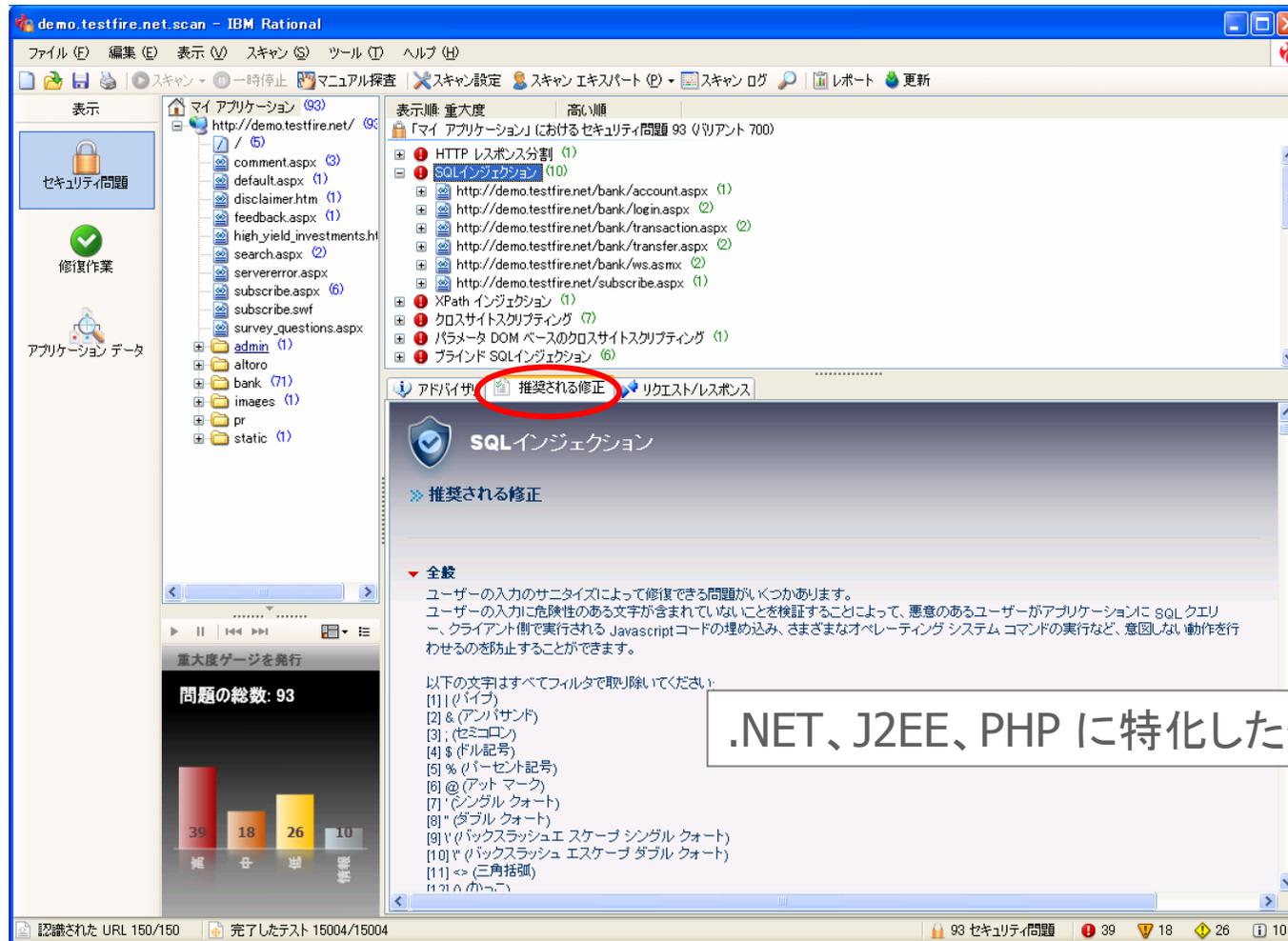


問題が重大度分けして表示される

サイトの構成が一目で分かる

セキュリティアドバイザリが日本語で

AppScan 7.7 - 推奨される修正の提示



.NET、J2EE、PHP に特化した修正方法も提示

AppScan 7.7 - リクエスト/レスポンスの表示

The screenshot displays the IBM AppScan 7.7 interface. The main window shows a list of security issues on the left, with 'HTTP レスポンス分割' (HTTP Response Splitting) selected. The central pane shows the details of this issue, including the request and response. The 'リクエスト/レスポンス' (Request/Response) tab is highlighted with a red circle. Below the main window, there are two callout boxes:

- 変更点と、問題点の強調表示** (Highlighting changes and problem points): This callout points to a bar chart in the response pane showing the number of occurrences for different issues. The chart has four bars with values 39, 18, 26, and 10.
- スクリーンショットの添付** (Attachment of screenshots): This callout points to a screenshot of the 'Altoro Mutual Server Error' message within the response details.

The response pane shows the following details:

```

POST /bank/account.aspx HTTP/1.0
Cookie: lang=international; amCreditOffer=CardType; amUserInfo=UserName; ASP.NET_SessionId=
=f1ple5va01pmgm550d2xhcej; amSessionId=244520224; amUserId=100116014
Content-Length: 41
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

listAccounts=1001160141%27+having+1%3D1--
HTTP/1.1 500 Internal Server Error
Content-Length: 5057
Connection: close
Date: Mon, 26 Nov 2007 08:15:45 GMT
Server: Microsoft-IIS/6.0
Content-Type: text/html; charset=utf-8

<!DOCTYPE html PUBLIC "-//W3C/DTD XHTML 1.0 Transitional/EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1
-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head id="ctl0_head"><title>
Altoro Mutual Server Error
</title><meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"><link href="._style.css" rel="
stylesheet" type="text/css"></head>
<body style="margin-top:5px;">
<div id="header" style="margin-bottom:5px; width: 99%;">

```

AppScan 7.7 - 修復作業

The screenshot shows the IBM Rational AppScan 7.7 interface. The main window is titled 'タイトルなし - IBM Rational'. The left sidebar contains navigation options: '表示' (Display), 'セキュリティ問題' (Security Issues), '修復作業' (Remediation), and 'アプリケーション データ' (Application Data). The '修復作業' tab is selected, showing a list of remediation tasks for the application 'http://demo.testfire.net/'.

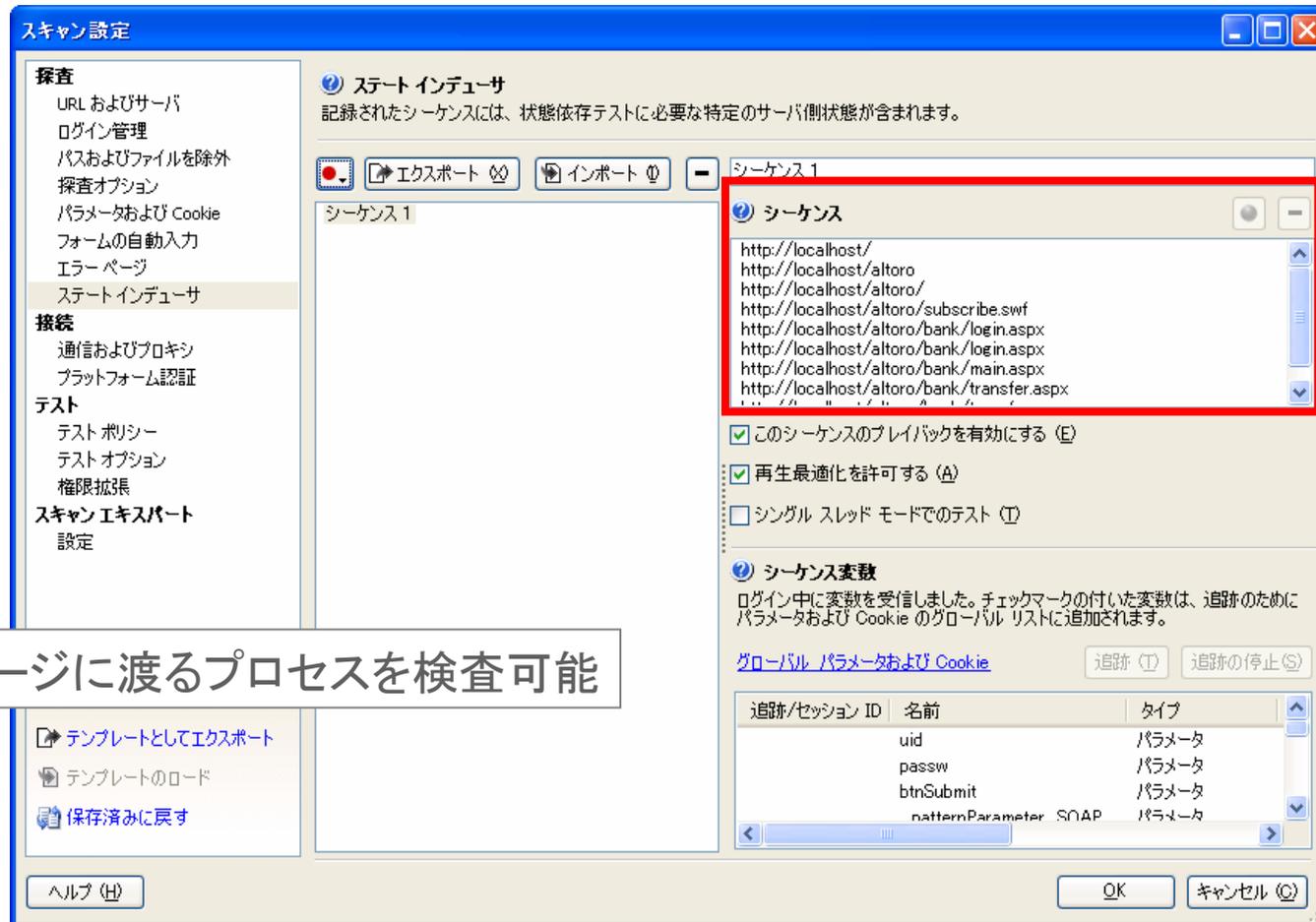
The remediation list includes tasks such as:

- アクセスされるファイルが仮想パスにあり、特定の拡張子を持つようにします。
- クライアント側のコードを分析して、入力ソースをサニタイズします。
- ユーザーがサインアウトしたときに、対応するセッション識別子を無効化します。
- ユーザーの入力から有害な文字をフィルタで取り除きます。
- ログイン証明書をより強力な組み合わせに変更します。
- すべてのログインリクエストを暗号化します。
- ディレクトリの一覧表示を拒否するようにサーバの設定を変更し、使用できる最新のセキュリティパッチをインストールします。
- 仮想ディレクトリから不要なファイルを削除します。
- Atutor の最新バージョンにアップグレードします。
- HTML コメントから秘密情報を削除します。
- Microsoft ASP.NET 上でのデバッグを無効にします。
- Web.Config ファイルを修正して、VIEWSTATE パラメータを暗号化します。
- サーバからテストスクリプトを削除します。
- パラメータ値が期待される範囲とタイプであることを確認します。
- ログインの試行に複数回失敗したアカウントをロックアウトします。
- 書く ASP.NET ページのプロパティを修正して、VIEWSTATE パラメータを署名します。
- 禁止されているリソースについて 404 - Not Found レスポンス ステータスコードを送出するか、完全に削除します。
- 秘密のセッション情報をパーマネント Cookie に保存しないようにします。
- 秘密情報を送信するときには、必ず HTTP POST メソッドを使用します。

A callout box with a green checkmark icon and the text 'ユーザーの入力から有害な文字をフィルタで取り除きます' (Filter out harmful characters from user input) points to the corresponding item in the list. Below the list, a summary section states: 'この修復作業は、以下のようなセキュリティ問題を解決することを想定して設計されています:' (This remediation is designed to solve the following security issues:). It lists 10 types of issues, including cross-site scripting, database errors, SQL injection, and XSS. A bar chart at the bottom shows the total number of issues: 93 total, with 39 high, 18 medium, 26 low, and 10 critical severity.

A callout box on the right side of the screenshot contains the text: '修正する人の観点に立った表示' (Display from the perspective of the person making corrections).

AppScan 7.7 -ステート インデューサー(New)



複数ページに渡るプロセスを検査可能

AppScan 7.7 - Word テンプレート(New)

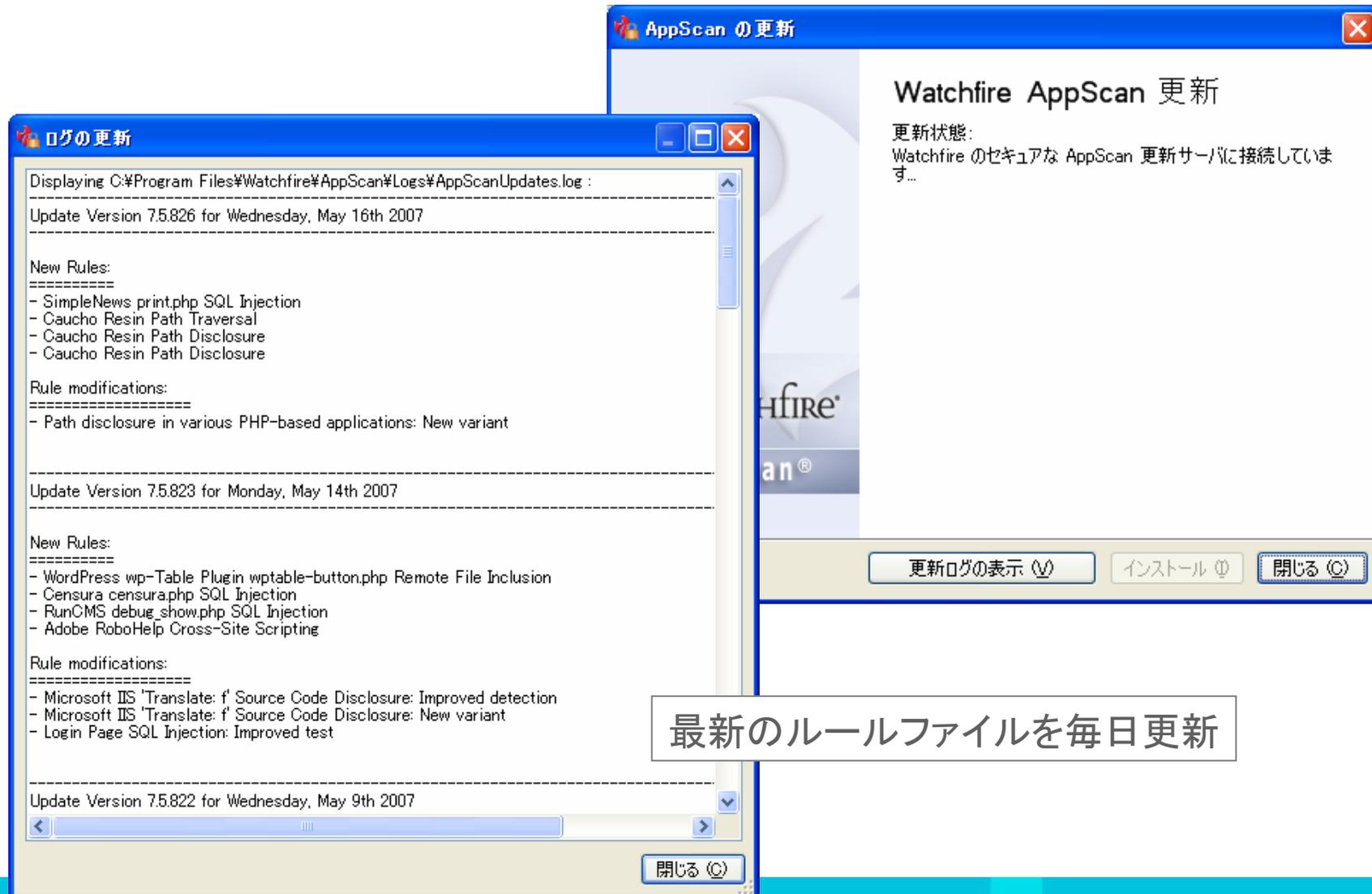
Word テンプレートを使って自由にレイアウト

IBM Rational AppScan Custom Word Report Fields - Microsoft Internet Explorer

フィールド名: C:\Documents and Settings\Amemiya\Local Settings\Temp\WordCustomReportFields.html

Field Name	AppScan Data
Broken URLs	
AS:BrokenURLRepeaterStart	Start BrokenURL repeater
AS:BrokenURLRepeaterIndex	The index of the broken URL
AS:BrokenURLReason	The reason for the broken URL
AS:BrokenURLName	The broken URL name
AS:BrokenURLRequest	The request associated with the URL
AS:BrokenURLCount	The total number of broken URLs
AS:BrokenURLRepeaterEnd	End BrokenURL repeater
Comments	
AS:CommentRepeaterStart	Start Comment repeater
AS:CommentRepeaterIndex	The index of the comment
AS:CommentCount	The total number of comments

AppScan - デイリーアップデート



AppScan 7.7 - レポート一覧

業界標準

- OWASP Top 10 2007
- OWASP Top 10 2004
- SANS Top 20 V5
- SANS Top 20 V6
- WASC Threat Classification
- The Payment Card Industry Data Security Standard (PCI)
- NERC CIPC Electricity Sector Security Guidelines
- International Standard – ISO 17799
- International Standard – ISO 27001
- Visa's Payment Application Best Practices

コンプライアンス

- PIPED Act
- European Directive 1995/46/EC
- European Directive 2005/58/EC
- Japan's Personal Information Protection Act
- Data Protection Act
- California Assembly Bill No. 1950 and Senate Bill 1386
- Children Online Privacy Protection Act (COPPA)
- Electronic Funds and Transfer Act (EFTA)
- Freedom of Information and Protection of Privacy Act (FIPPA)
- Federal Information Security Mgmt. Act (FISMA)
- Financial Services (GLBA)
- Healthcare Services (HIPAA)
- Management of Information Security Technology (MITS)
- NERC Cyber Security Standards
- Privacy Act of 1974
- Safe Harbor
- Sarbanes-Oxley Act (SOX)
- The Securities Act
- Title 21 Code of Federal Regulations
- DCID 6/3 Availability Basic
- DCID 6/3 Availability Medium
- DCID 6/3 Availability High
- DCID 6/3 Confidentiality Reqs Protection Level 1
- DCID 6/3 Confidentiality Reqs Protection Level 2
- DCID 6/3 Confidentiality Reqs Protection Level 3
- DCID 6/3 Confidentiality Reqs Protection Level 4
- DCID 6/3 Confidentiality Reqs Protection Level 5
- DCID 6/3 Integrity Basic
- DCID 6/3 Integrity Medium
- DCID 6/3 Integrity High
- DCID 6/3 Securing Advanced Technology IS
- MasterCard SDP
- Visa CISP
- Basel II
- Family Education Rights and Privacy Act (FERPA)
- NIST Special Publication 800-53



AppScan - Pyscan エクステンション

The screenshot shows the Watchfire AppScan interface with a scan of localhost. The main window displays a list of security issues, including HTTP response splitting, Microsoft ASP.NET Cross-Site Scripting, SQL injection, and others. A Pyscan terminal window is open, showing the output of a `printIssues()` command. A callout box highlights that script languages can access AppScan objects.

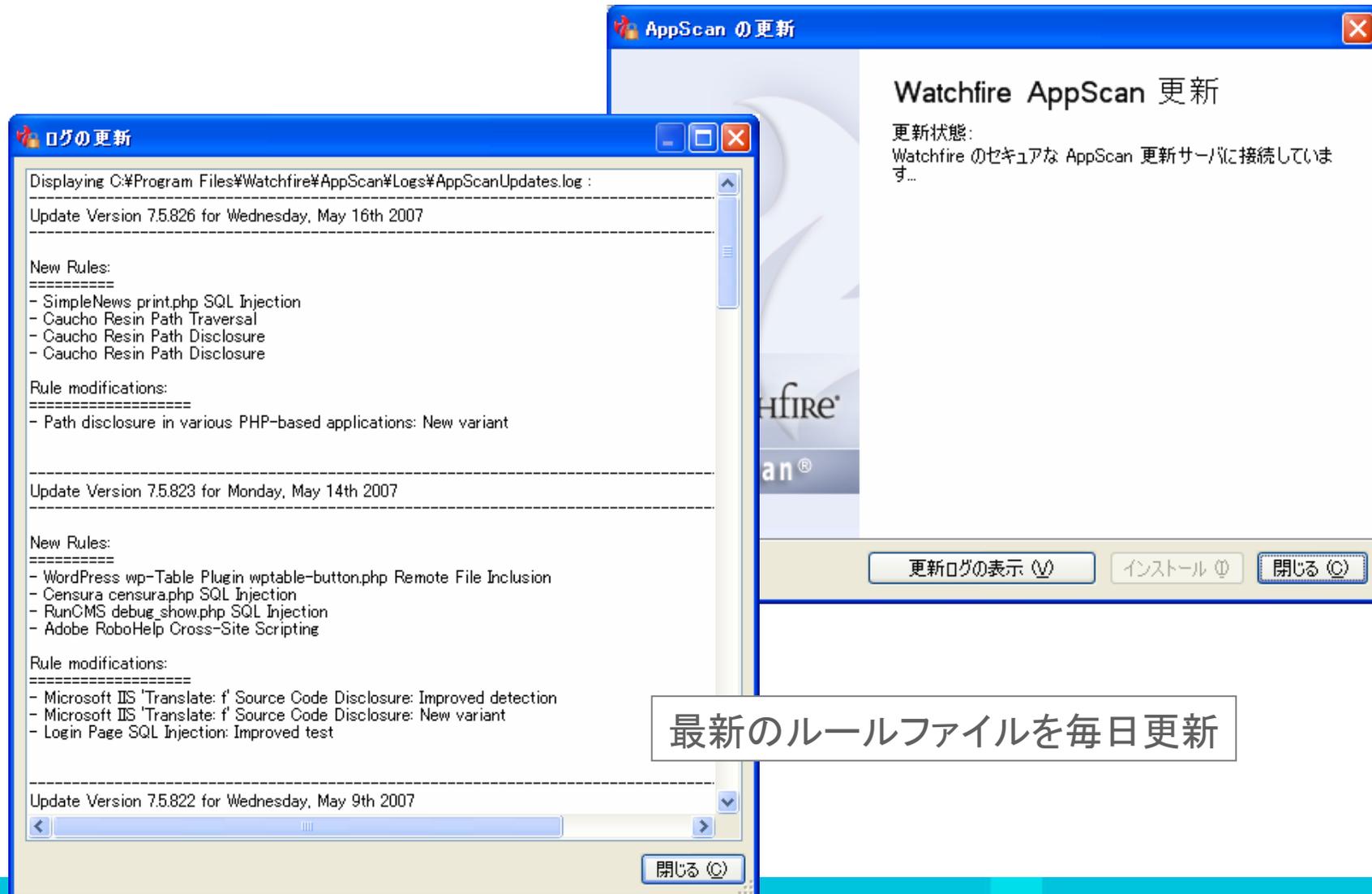
スクリプト言語から AppScan オブジェクトにアクセス可能 (認が必要)

の外觀を破壊する
...ハッカーが正規の

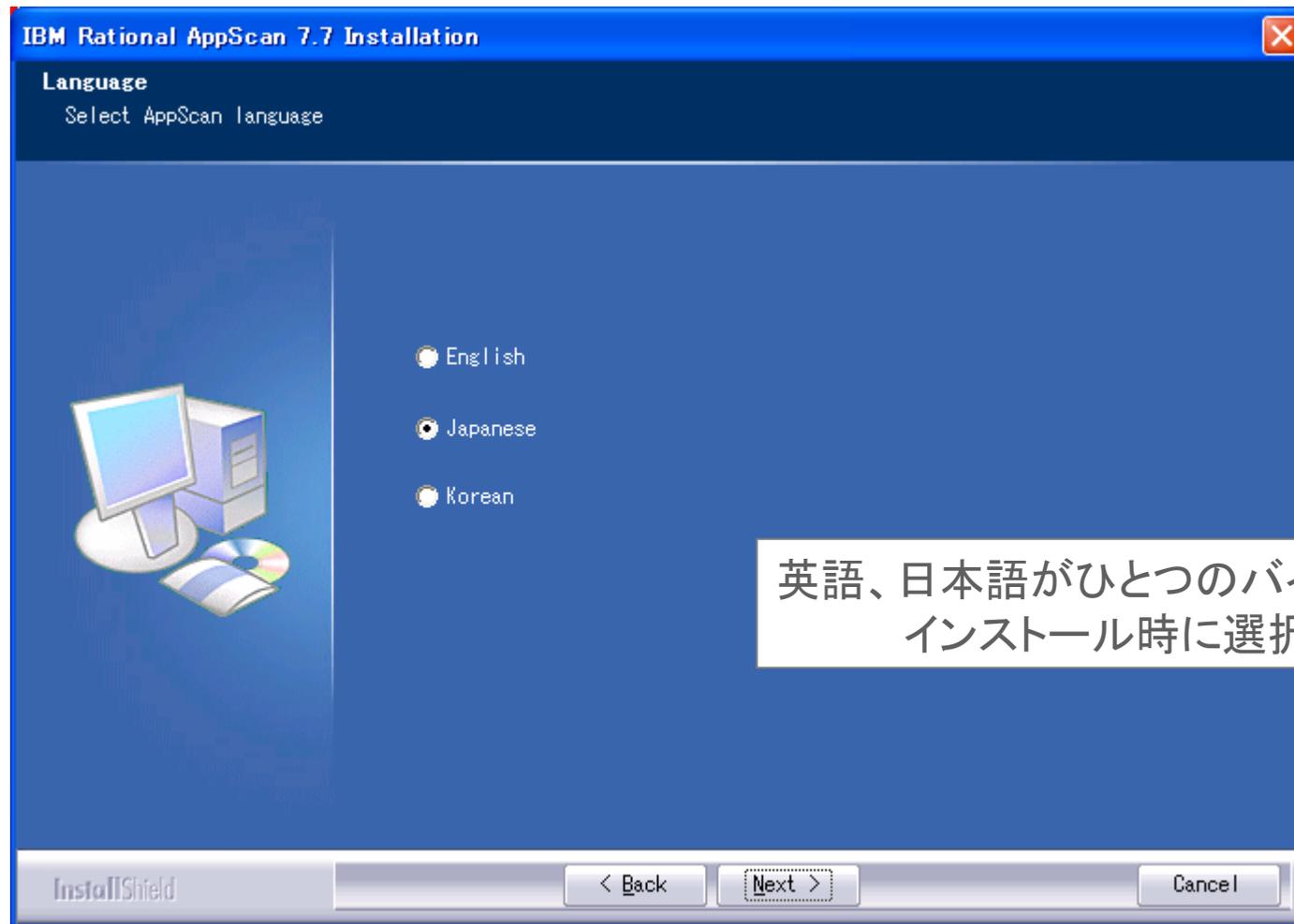
```
>>> printIssues()
HTTP レスポンス分割
Microsoft ASP.NET Cross-Site Scripting
SQLインジェクション
WS_FTP.log のファイル分析ソース コード開示
XPath インジェクション
クロスサイトスクリプティング
ブラインド SQLインジェクション
ログイン ページの SQLインジェクション
予測可能なログイン証明書
書式文字列リモート コマンドの実行
汚染 Null バイト ファイルの取得
Link Injection
TRACE および TRACK HTTP メソッドが有効
ディレクトリの一覧作成
暗号化されていないログイン要求
秘密ファイルの検出
IIS localstart.asp への総当たり攻撃
Microsoft ASP.NET のデバッグが有効
WS_FTP.LOG の情報漏洩
アカウントの不適切なロックアウト
アプリケーション エラー
サーバーのパス開示の可能性のあるパターンを発見
セッション識別子が未更新
暗号化されていないパスワード パラメータ
符号のない __VIEWSTATE パラメータ
HTTP レスポンスによる秘密情報の開示
セキュリティで保護されていない HTTP メソッドが有効
暗号化されていない __VIEWSTATE パラメータ
非表示のディレクトリを検出
>>> |
```

Ln: 129 | Col: 4

AppScan - デイリーアップデート



日本語版



英語、日本語がひとつのバイナリに
インストール時に選択

AppScan のマーケット

- 侵入検査サービス
 - ▶ サービスのツールとして
- Sler、アプリケーション開発
 - ▶ 開発段階での検査
 - ▶ 納品時検査
- エンドユーザ
 - ▶ 検収時検査
 - ▶ 自社 Web の定期的な検査



AppScan の利点 - Sler、アプリケーション開発にとって

- セキュリティ検査にかかっていたコストの削減
- 高品質な製品の提供
- 人やプロジェクトごとに異なるセキュリティレベルの標準化

- 開発者から見た視点での問題の指摘
 - ▶ 修復作業ビュー
 - ▶ スクリーンショット付きの詳細なレポート
- セキュリティ専門家でなくても利用可能





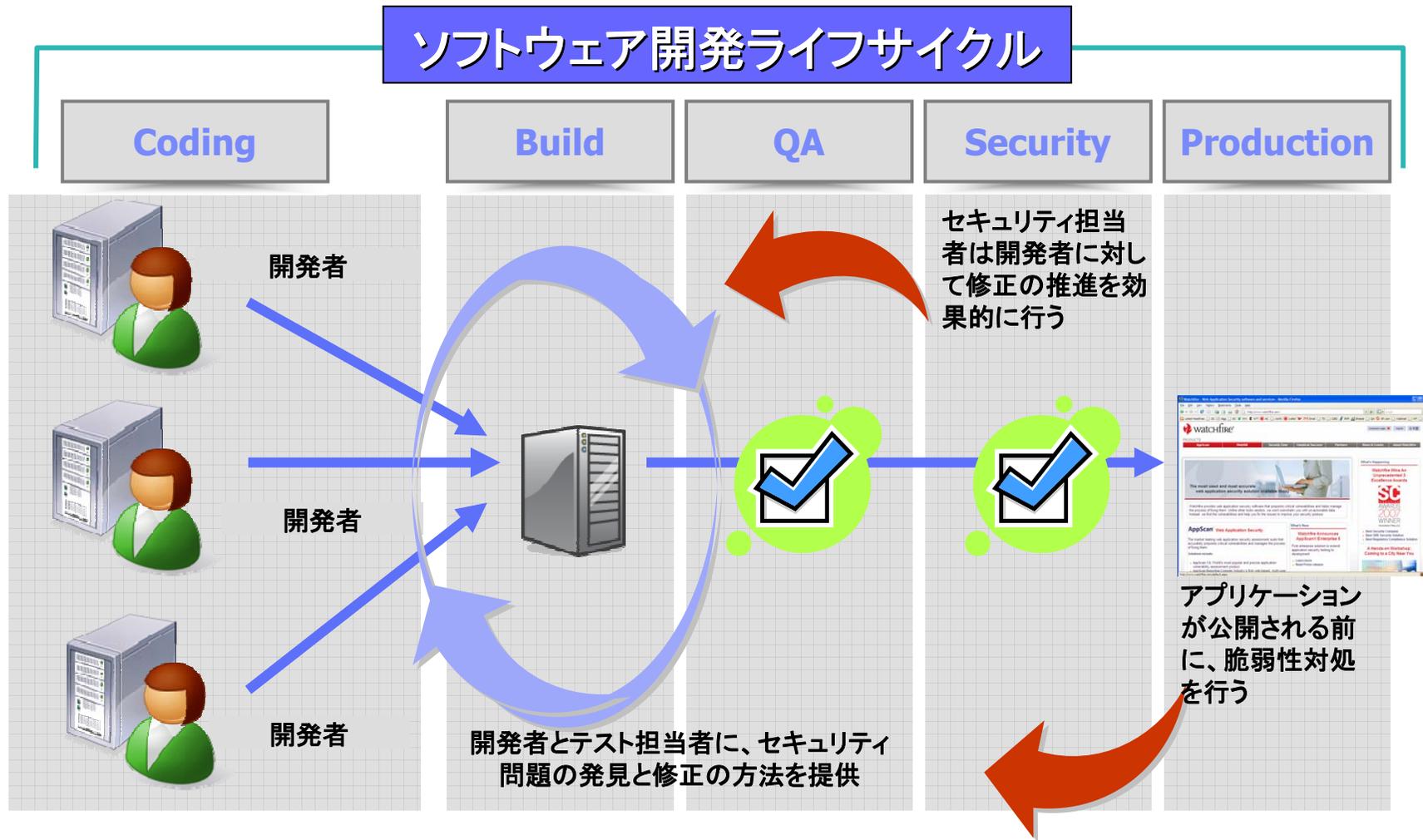
IBM Software Group

開発工程におけるセキュリティテスト

Rational software

→ Go to **IBM**

SDLCにおける Web アプリケーション セキュリティ検査



アプリケーション セキュリティ確立のポイント

1. 人

- 教育
- コミュニケーション
- 責任の所在

2. プロセス

- プロセスの確立
- ゴールの設定

3. 技術

- 全社的に展開可能な技術を選定
- 導入のしやすさ、使いやすさ



ユーザーの声 - 開発者の意識改革

- 「導入当初は、テストの結果により大幅な修正を要求することもあったため、開発者からは嫌がられることもありましたが。しかし現在では、開発者自らがセキュアなコーディングを心がけるようになってきているのを感じます」(ヤマハ様)
- 「スキャンして出てきた問題はレポートでチェックし、システムを修正します。次のシステムの開発時には注意してコーディングするようになり、同じ間違いをすることが減りました」
「我々は多くのお客様の大切な情報を預かっています。システムのどこにも脆弱性があるはいけません。安全性の高いシステムを構築することが我々の使命であり、仕事のモチベーションです。AppScan導入はセキュリティ品質の向上と同時に、開発者の意識向上のきっかけにもなりました」(東京海上日動システムズ様)

セミナーの御案内

- **2/8(金) : Webアプリケーション脆弱性対策セミナー**
 - ▶ @IBM SWCOC 渋谷
 - ▶ 各種の Web アプリケーション脆弱性対策を解説

- **2/19(火) : Webアプリケーションをハッカーからの攻撃から守る**
 - ▶ @IBM 箱崎事業所
 - ▶ AppScan を使った事例、効果を解説

<http://www.ibm.com/jp/software/rational/events/>





Thank You

