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

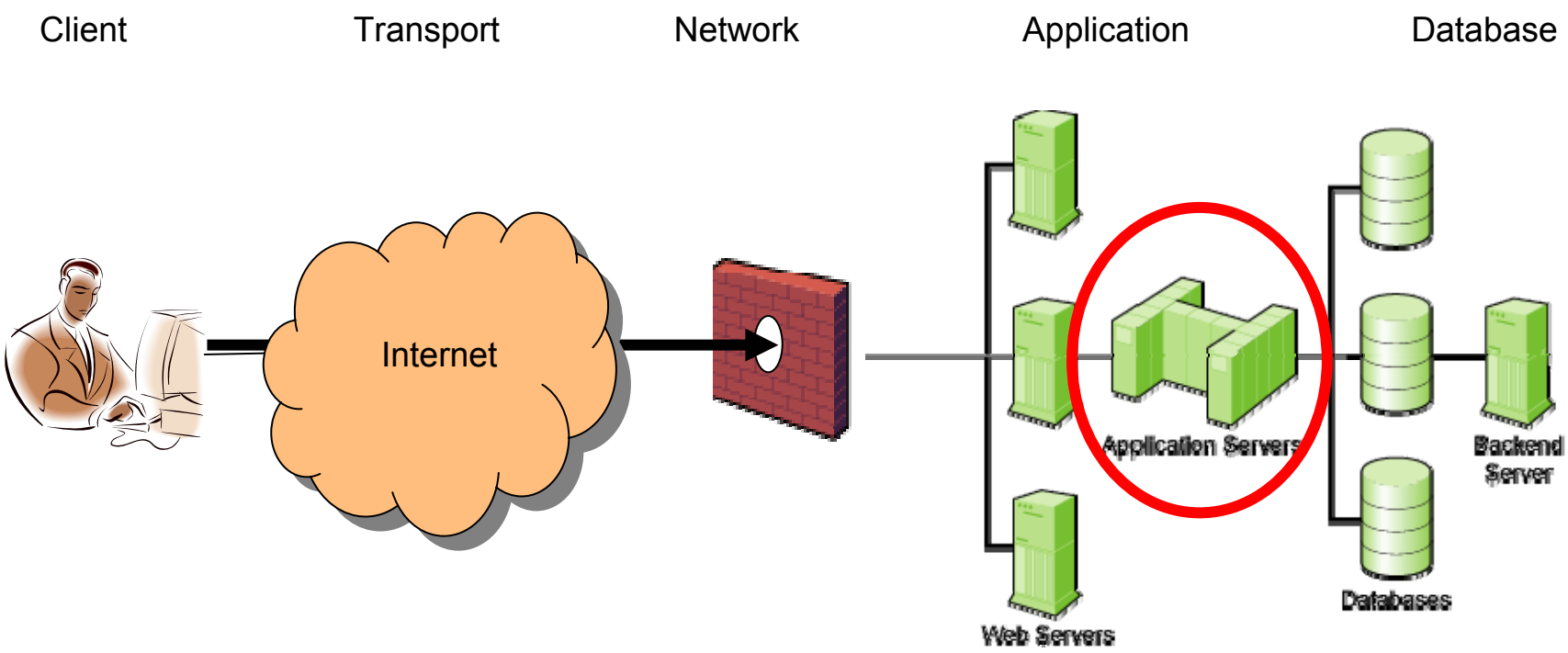
Agenda

- 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

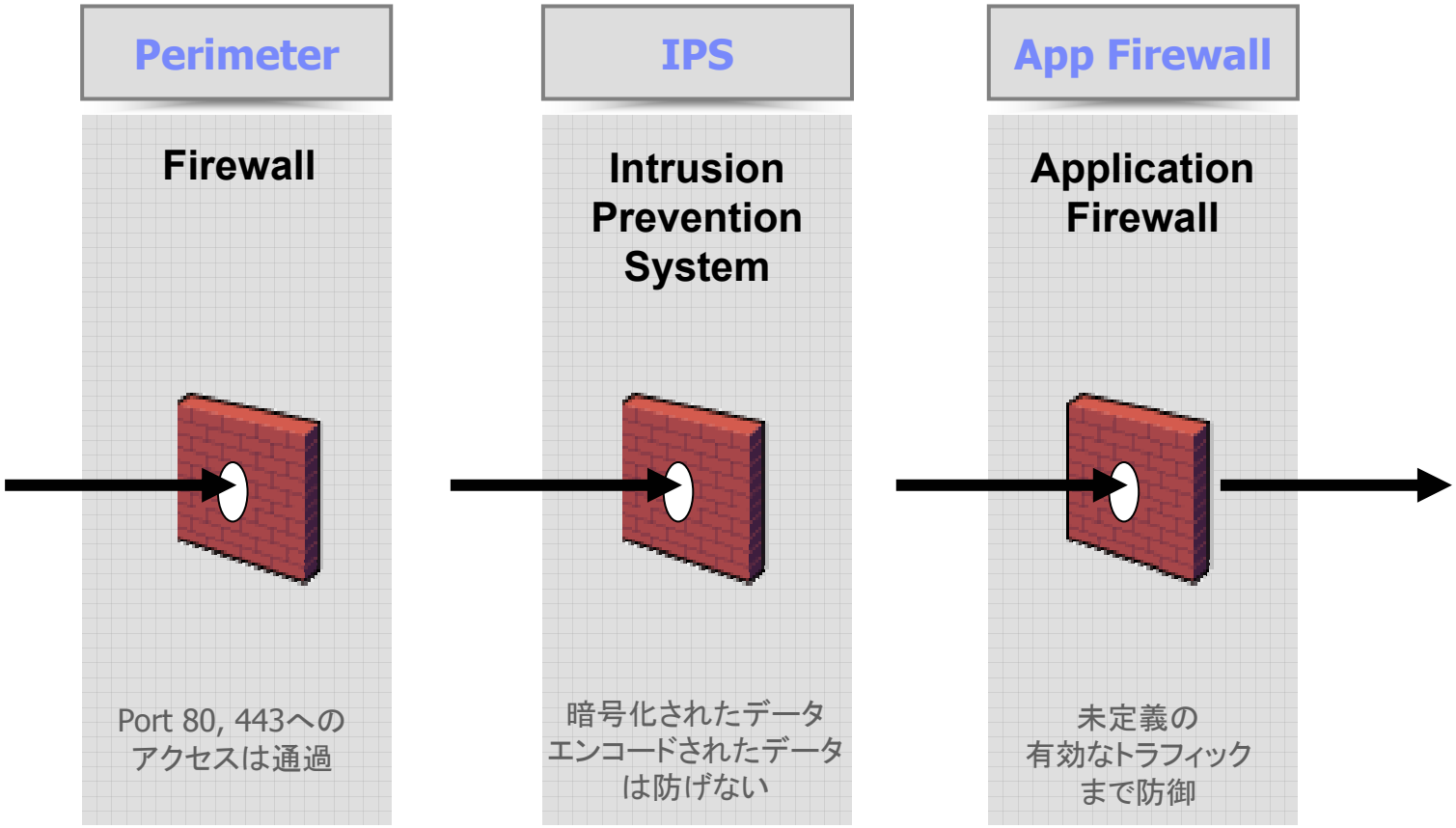
IBM Software Group

Web アプリケーションのセキュリティ問題

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



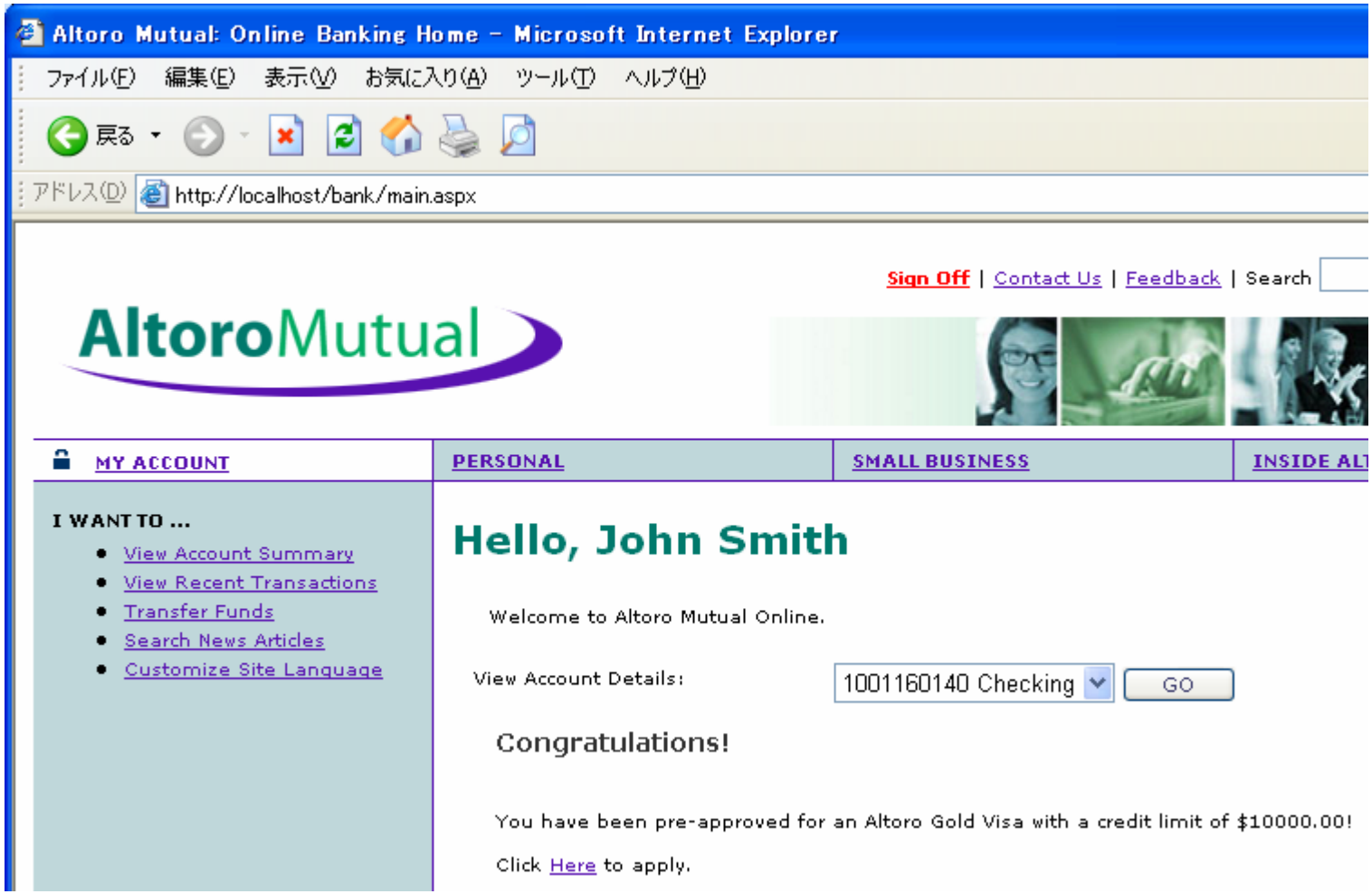
Web アプリケーションに対する攻撃

- 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'
```

SQL インジェクション



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-Fowarded-For' ヘッダの値を以下に変更します: '
  - HTTP 'X-Fowarded-For' ヘッダの値を以下に変更します: %27
  - HTTP 'X-Fowarded-For' ヘッダの値を以下に変更します: %2527
  - HTTP 'X-Fowarded-For' ヘッダの値を以下に変更します: "
  - HTTP 'X-Fowarded-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 バリアント)
  - ポート リスナおよび 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 バリアント)

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

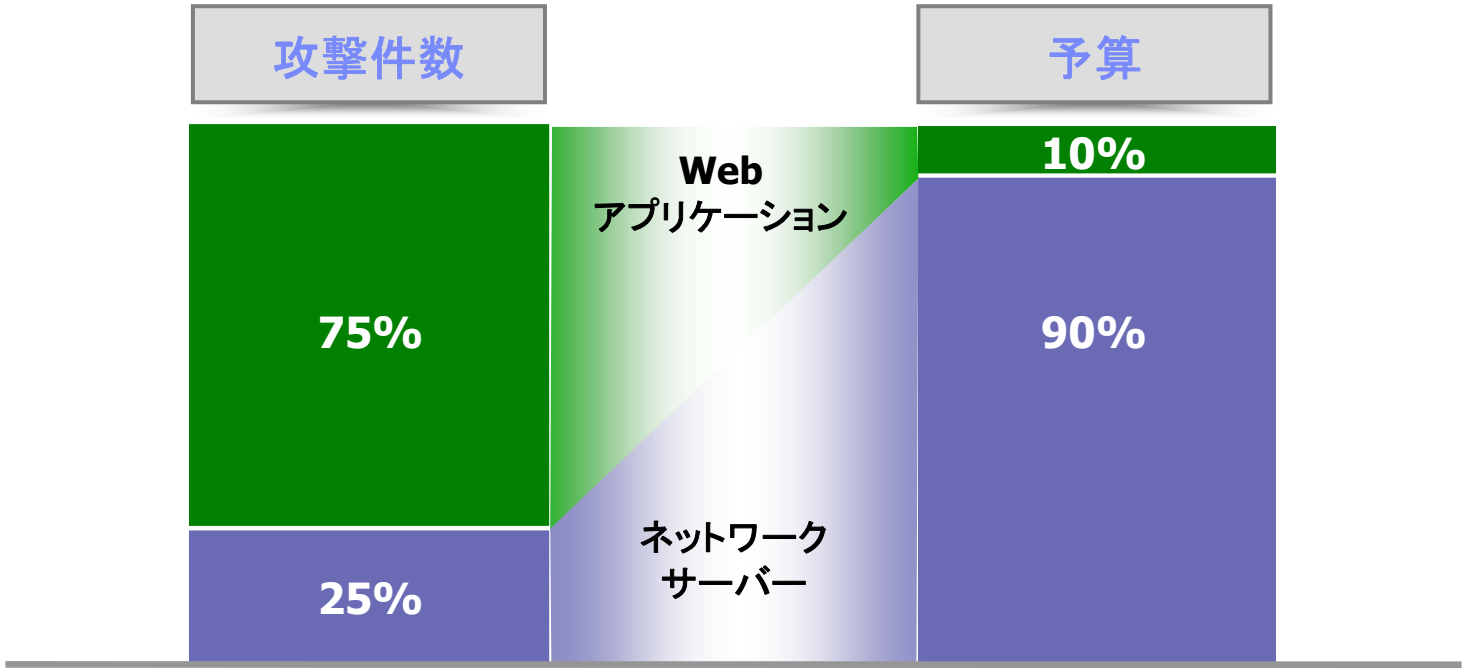
IBM Software Group

Web アプリケーションセキュリティ対策の現状

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かっている (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		▪専門家による詳細な調査	▪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		▪既存のアプリケーションに対してしても対応可能	▪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	▪問題の修正位置が的確に把握できる	▪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	▪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	▪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

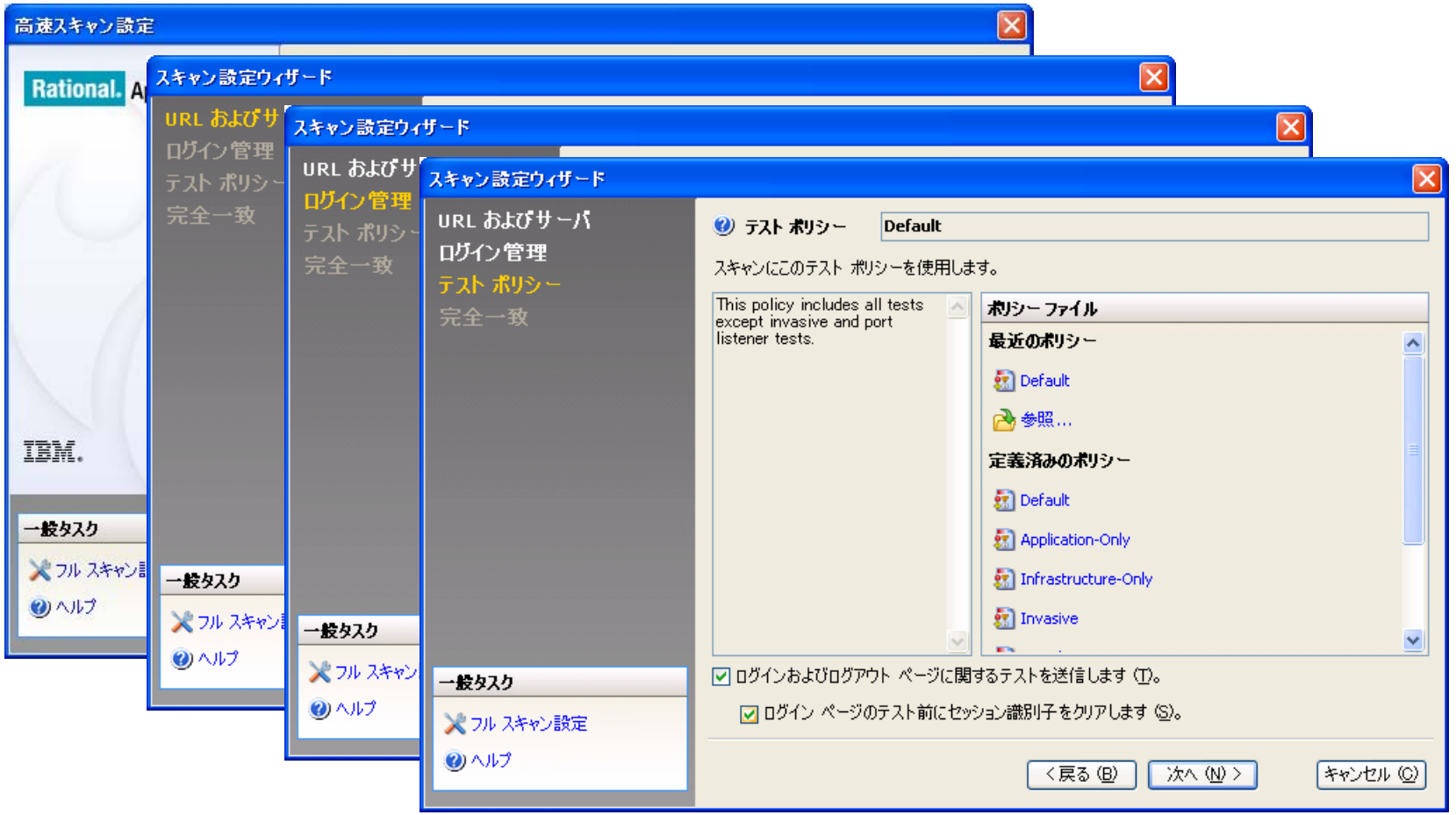
AppScan

AppScan

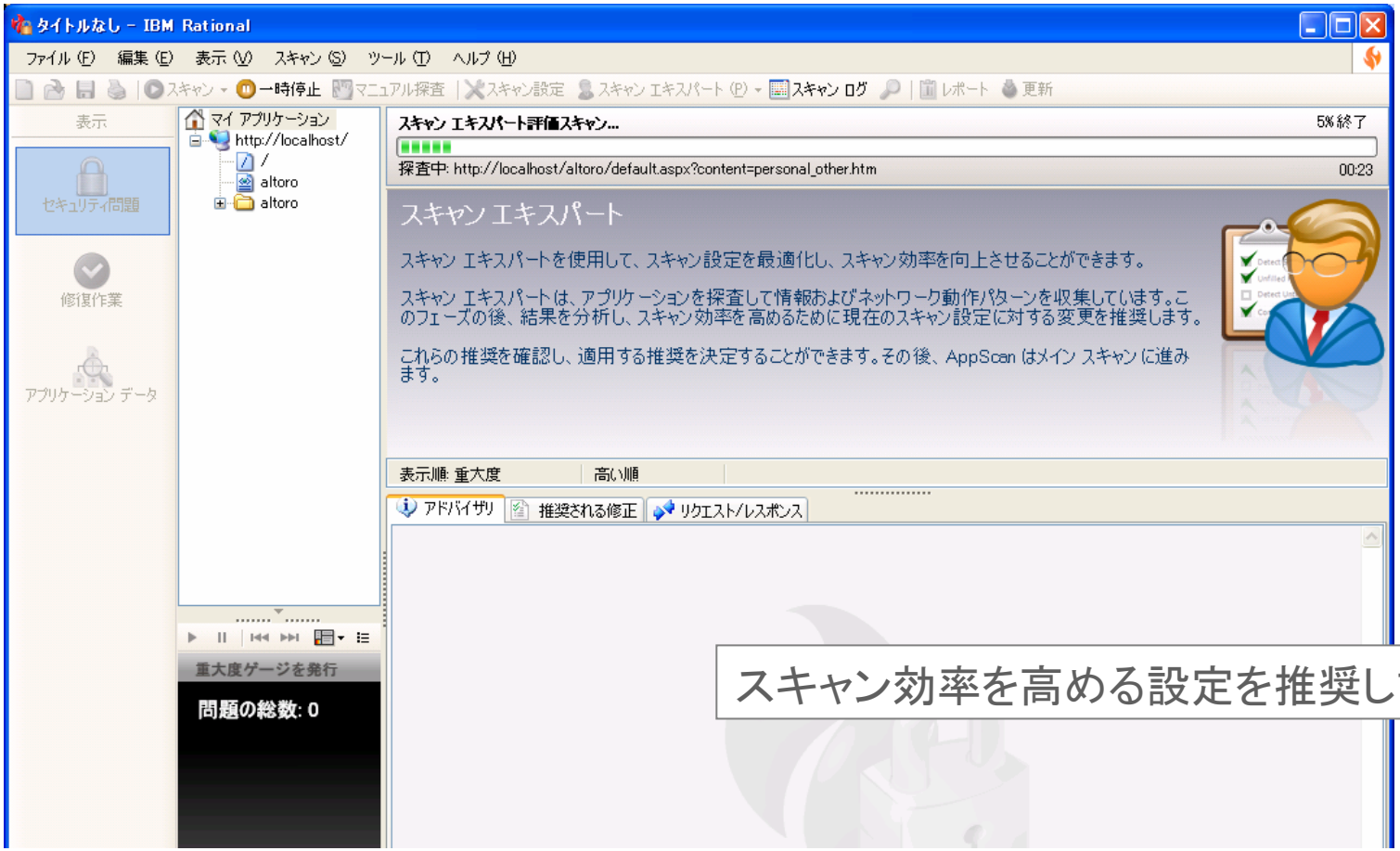
- Webアプリケーション セキュリティ テスト ツール
  - Webアプリケーションの脆弱性と、インフラ(OS、Webサーバー等)の設定ミス、既知の問題を検知
  - テストを自動化し、手作業に比べて圧倒的な時間とコストの削減が可能
  - 脆弱性の指摘、修正方法の提示、レポートの作成
  - ブラックボックス テスト
    - ハッカーの視点でセキュリティ問題を検査 - **"Hacker in the box"**
    - "現実的な" 問題点の指摘 - **"Low hanging fruits"**

Demo

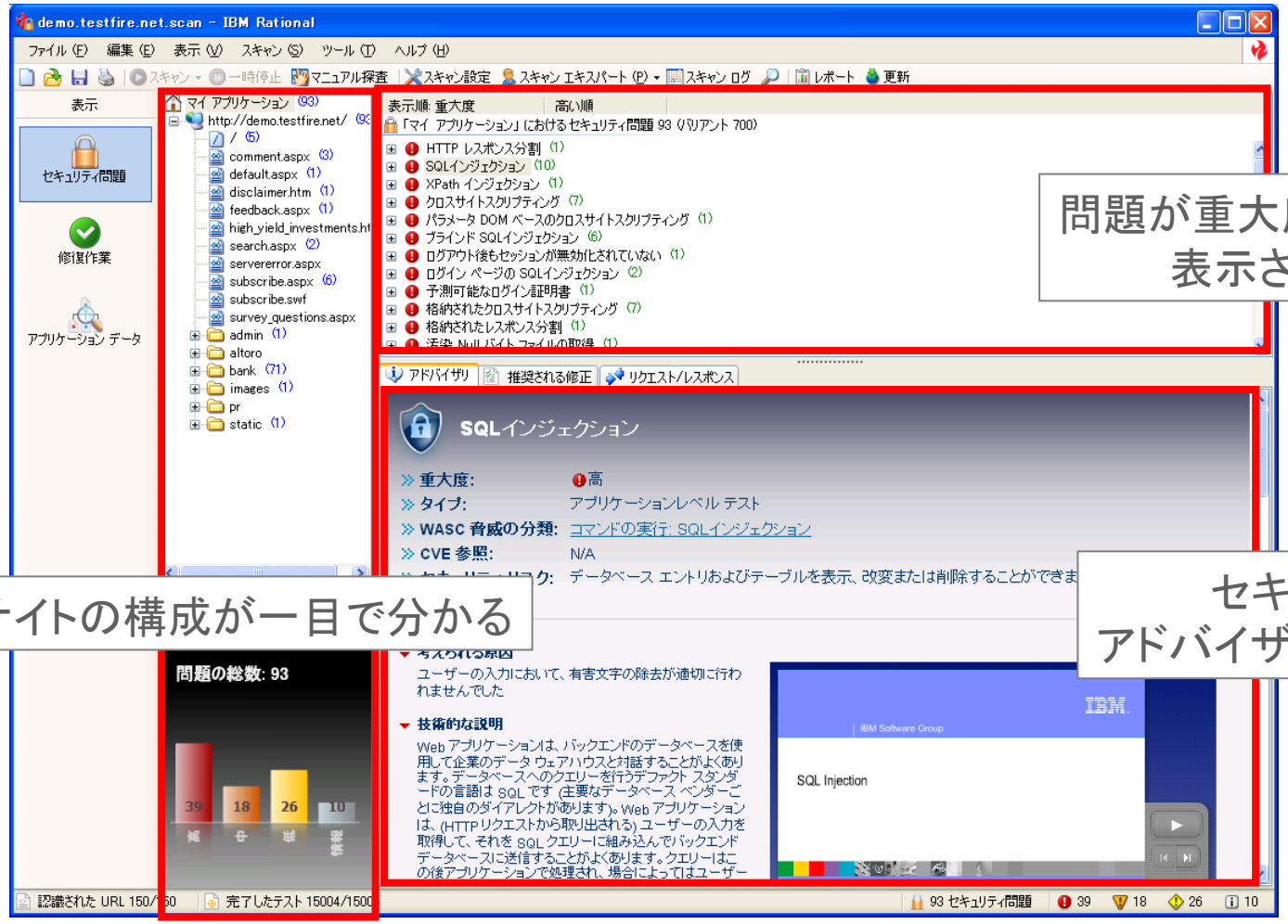
AppScan 7.7 - ウィザードによる容易な設定



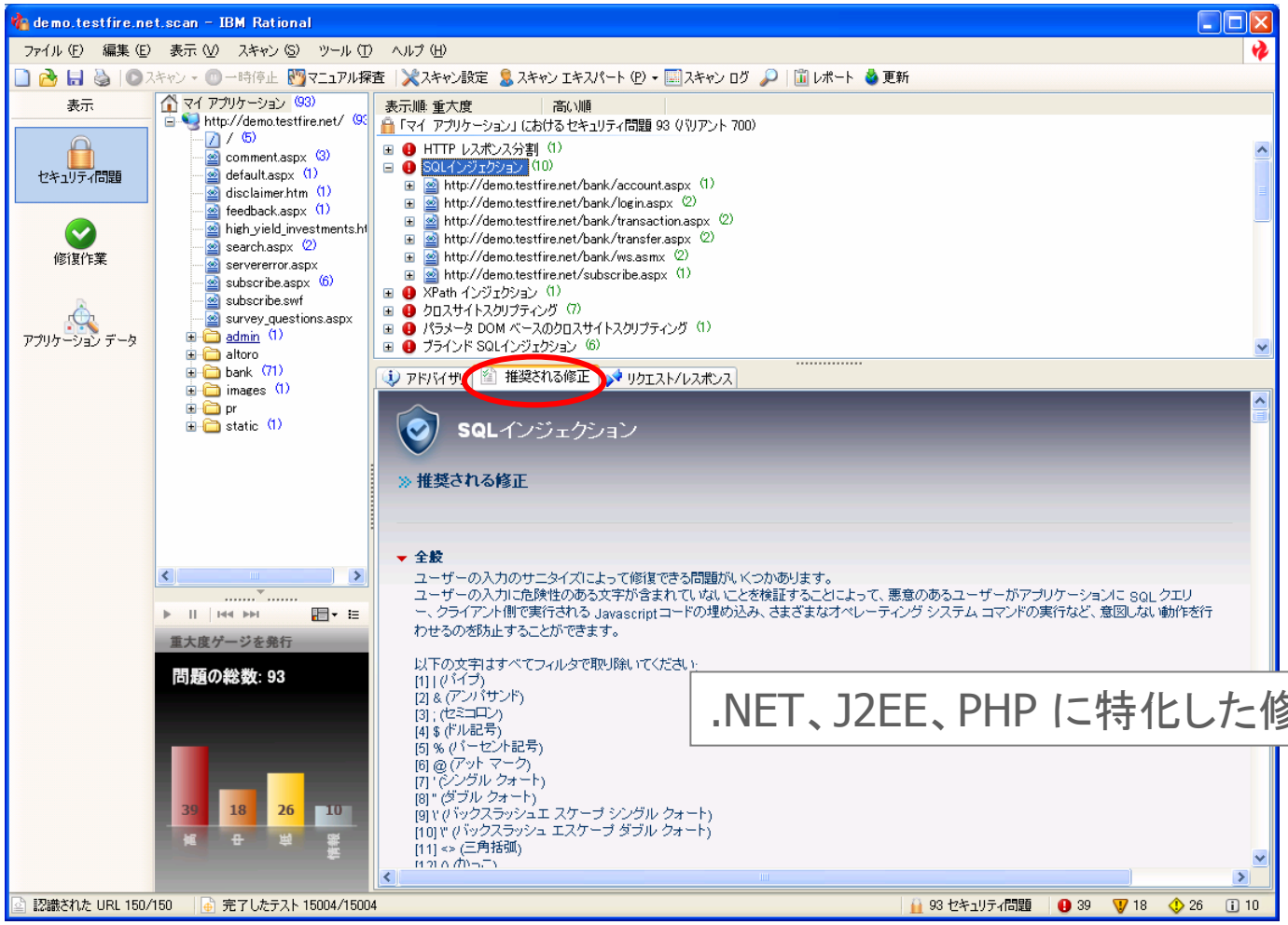
AppScan 7.7 - スキャン エキスパート(New)



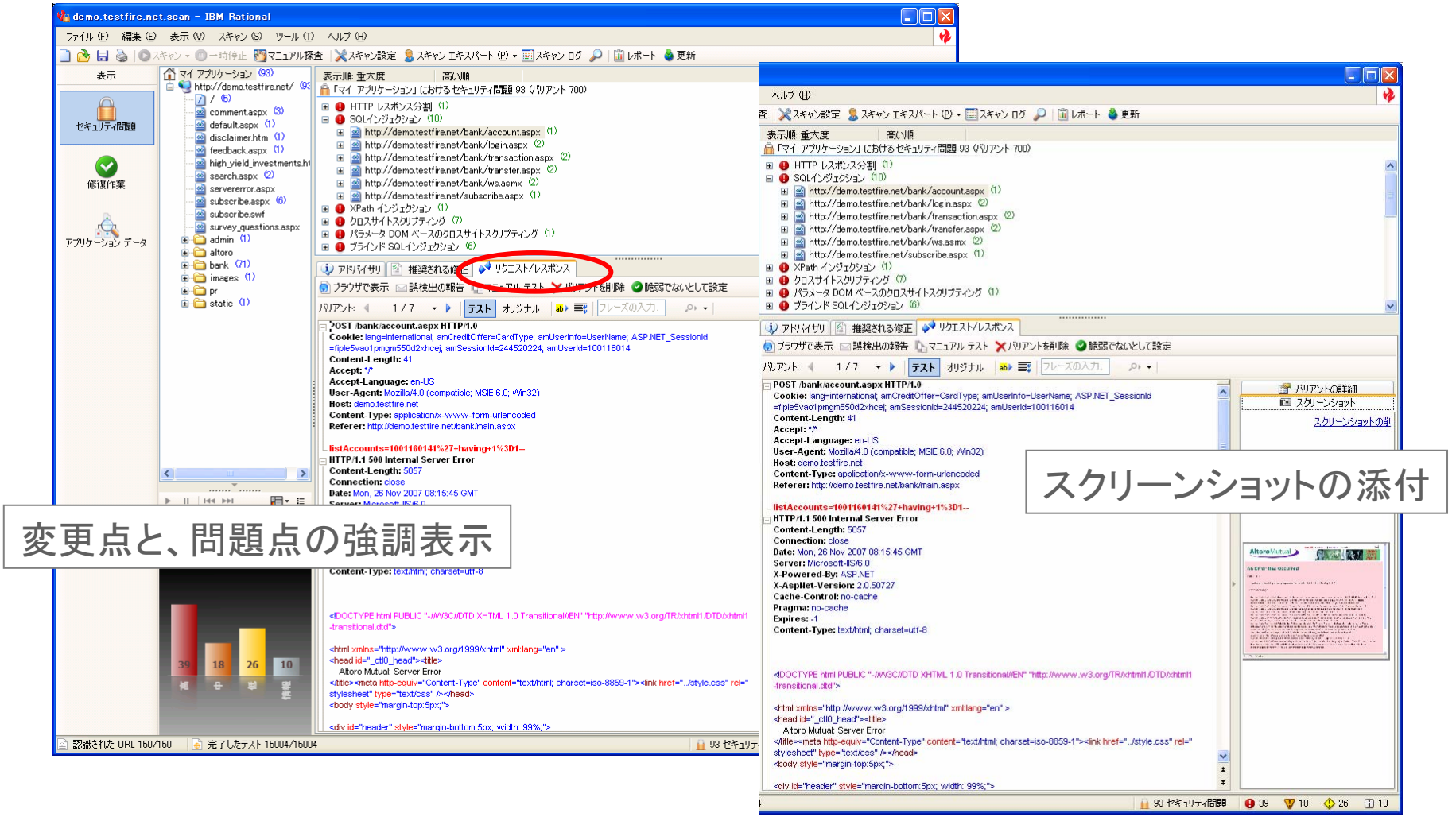
AppScan 7.7：見易い GUI



AppScan 7.7 - 推奨される修正の提示



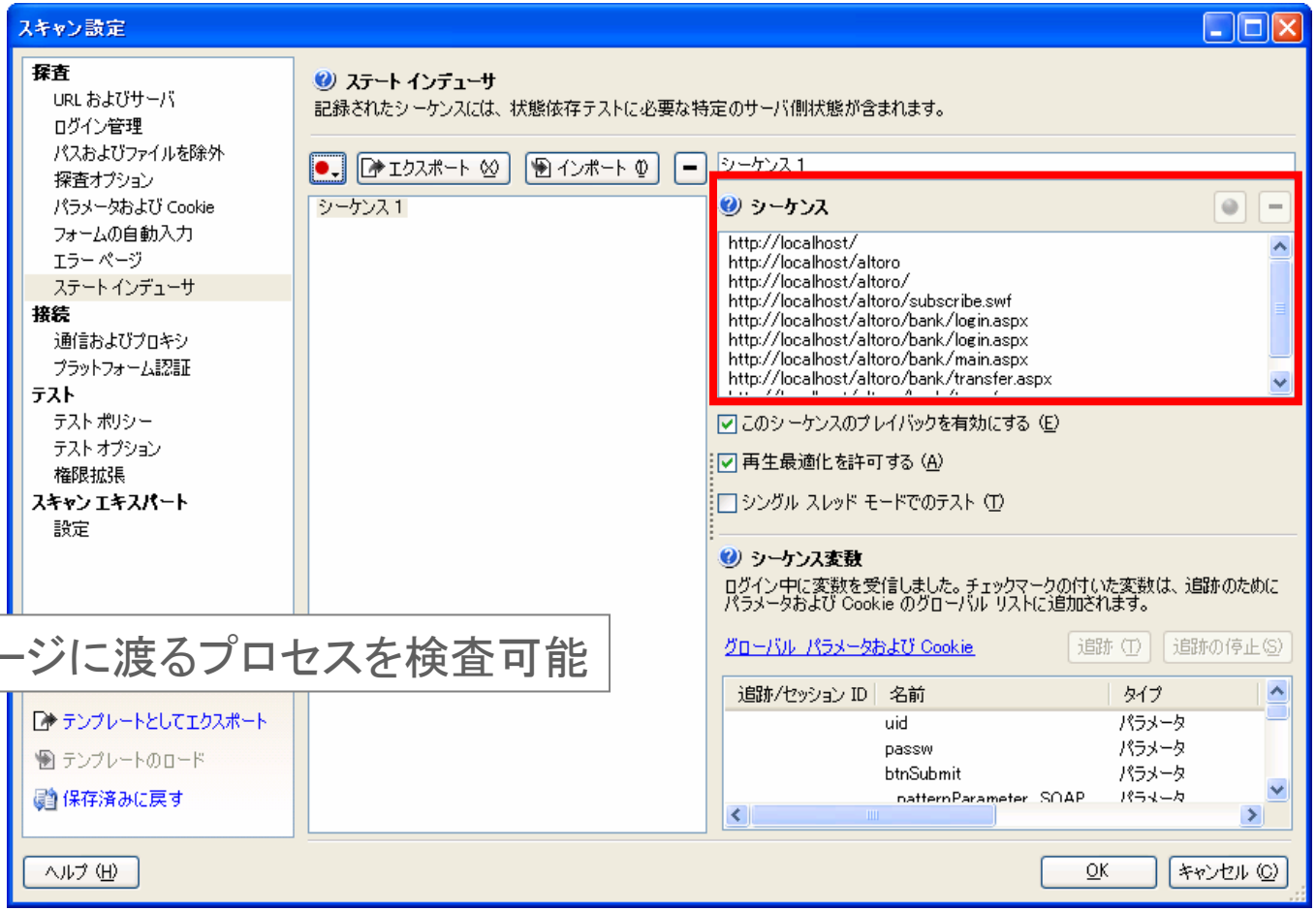
AppScan 7.7 - リクエスト/レスポンスの表示



AppScan 7.7 - 修復作業

修正する人の観点に立った表示

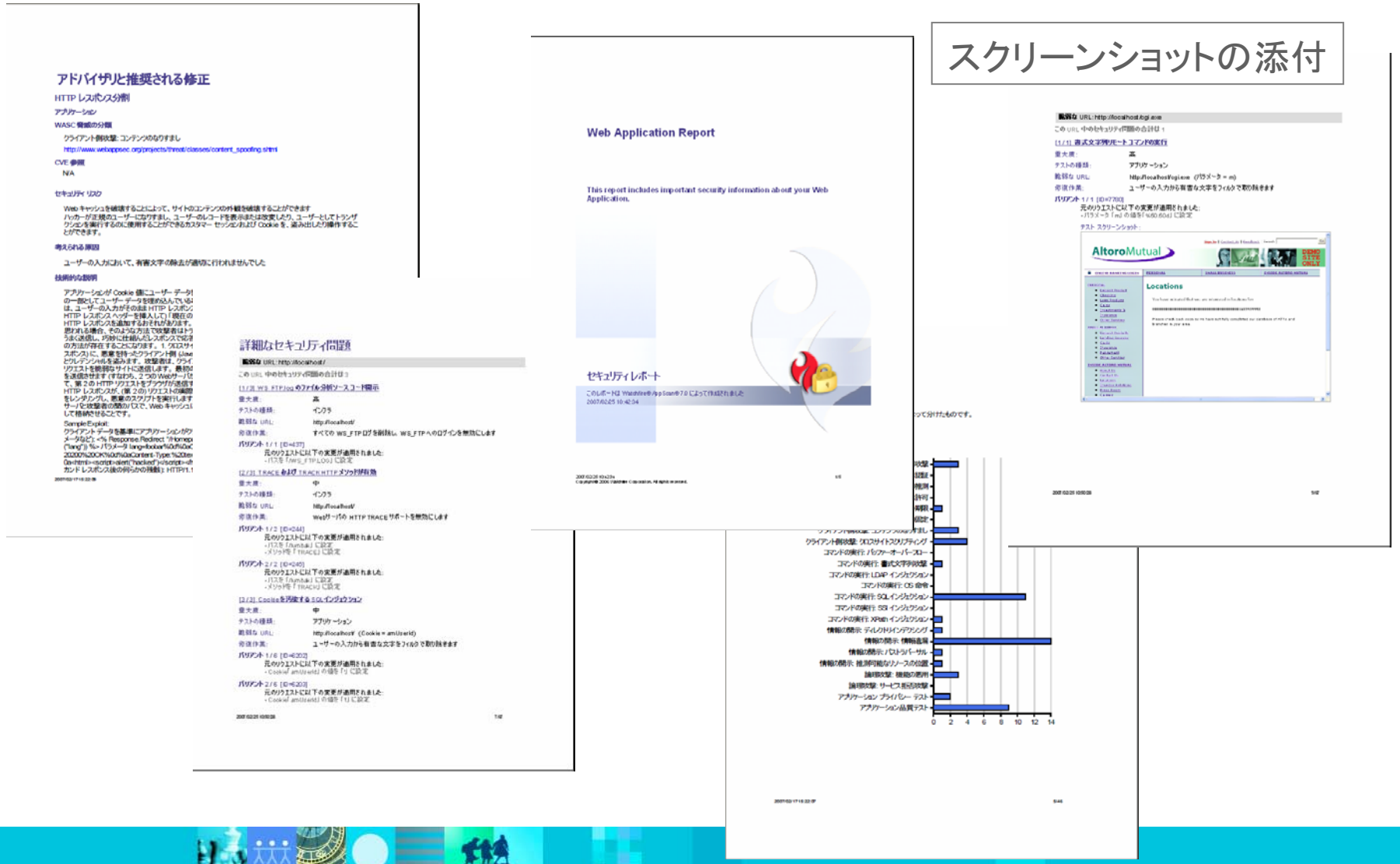
AppScan 7.7 -ステート インデューサー(New)



複数ページに渡るプロセスを検査可能

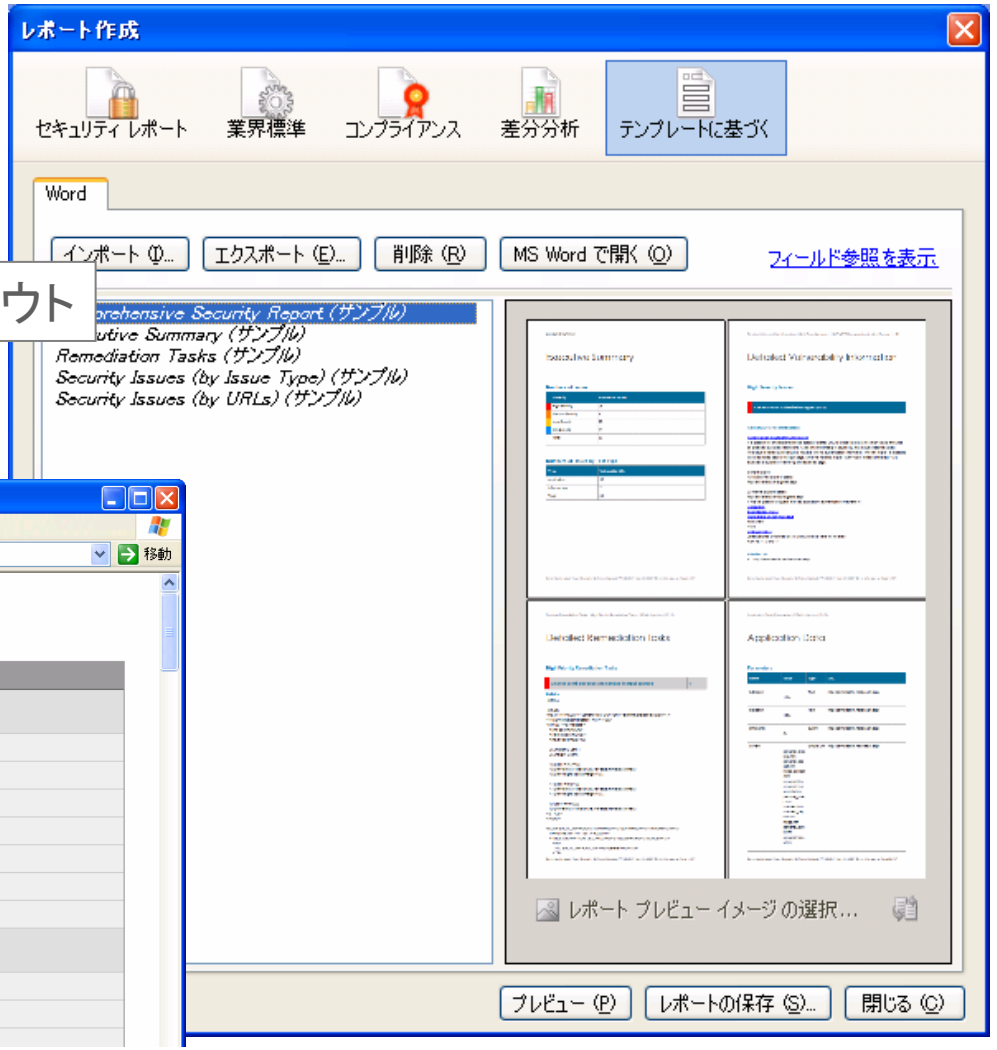
AppScan 7.7 - 詳細なレポート

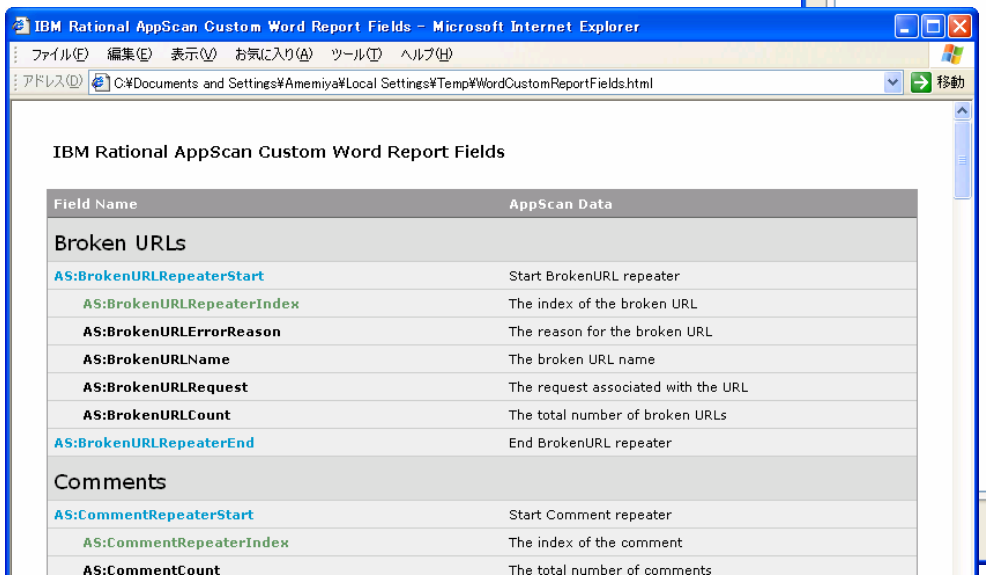
スクリーンショットの添付



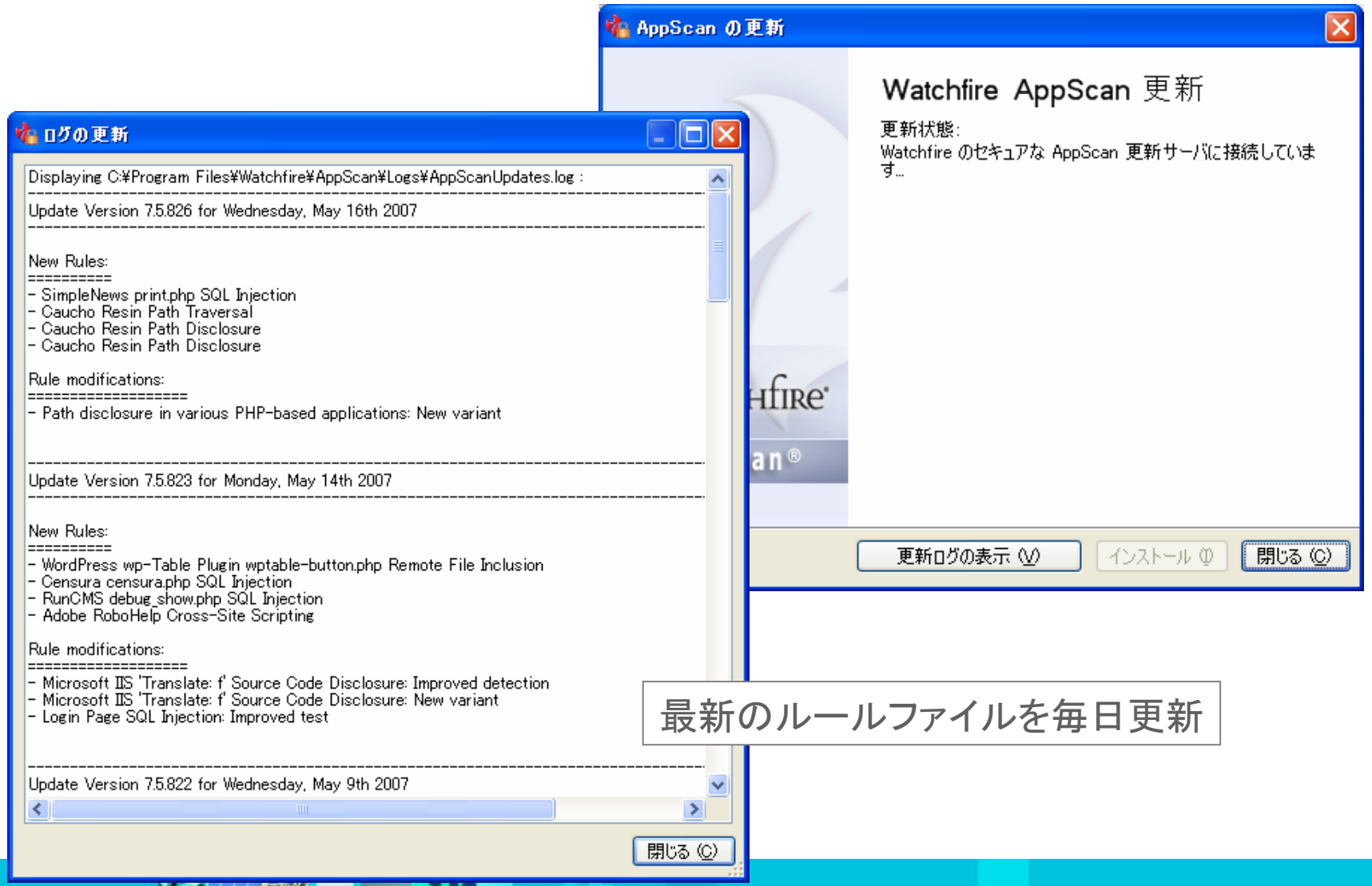
AppScan 7.7 - Word テンプレート(New)

Word テンプレートを使って自由にレイアウト





AppScan - デイリーアップデート



最新のルールファイルを毎日更新

AppScan 7.7 - レポート一覧

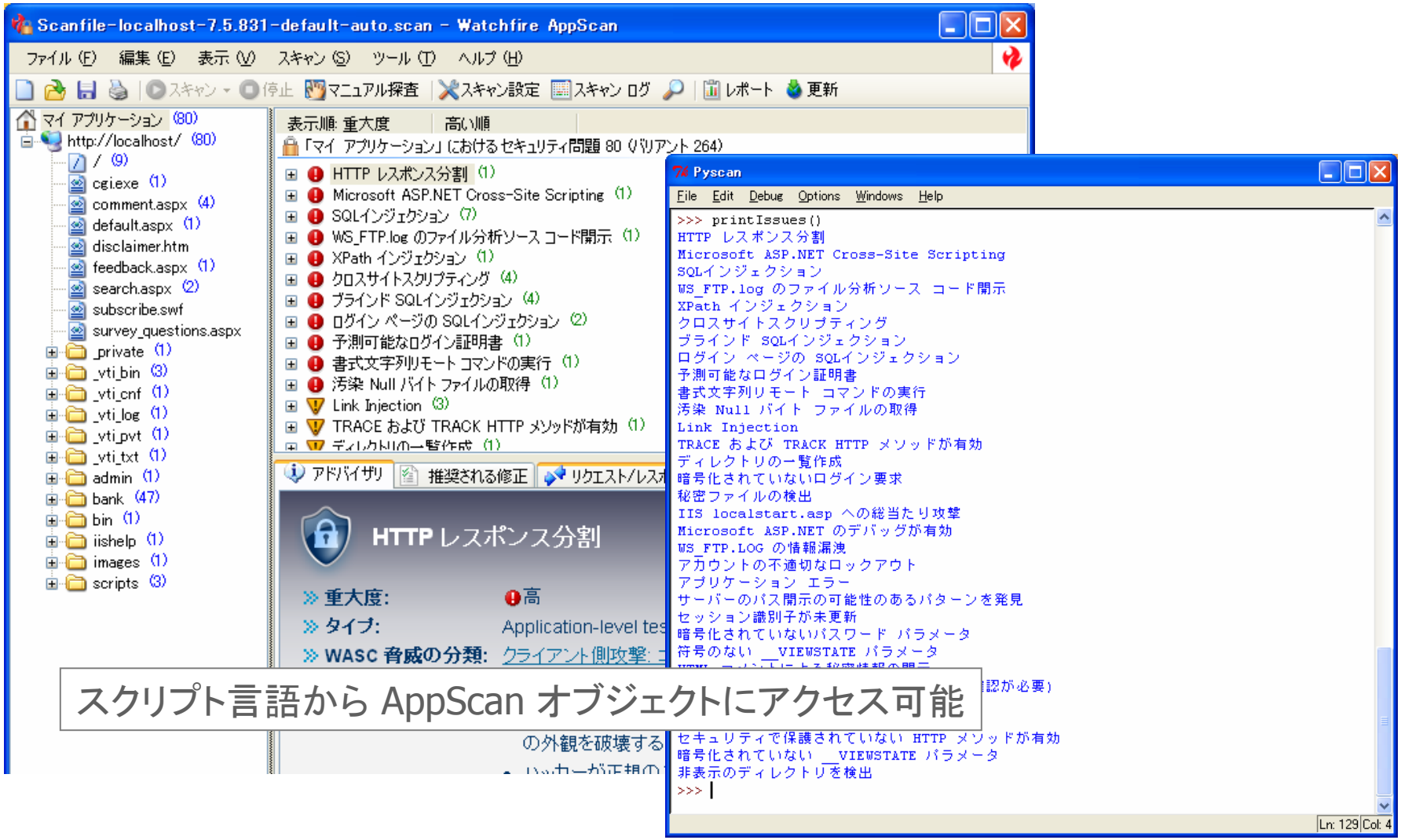
業界標準

- 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

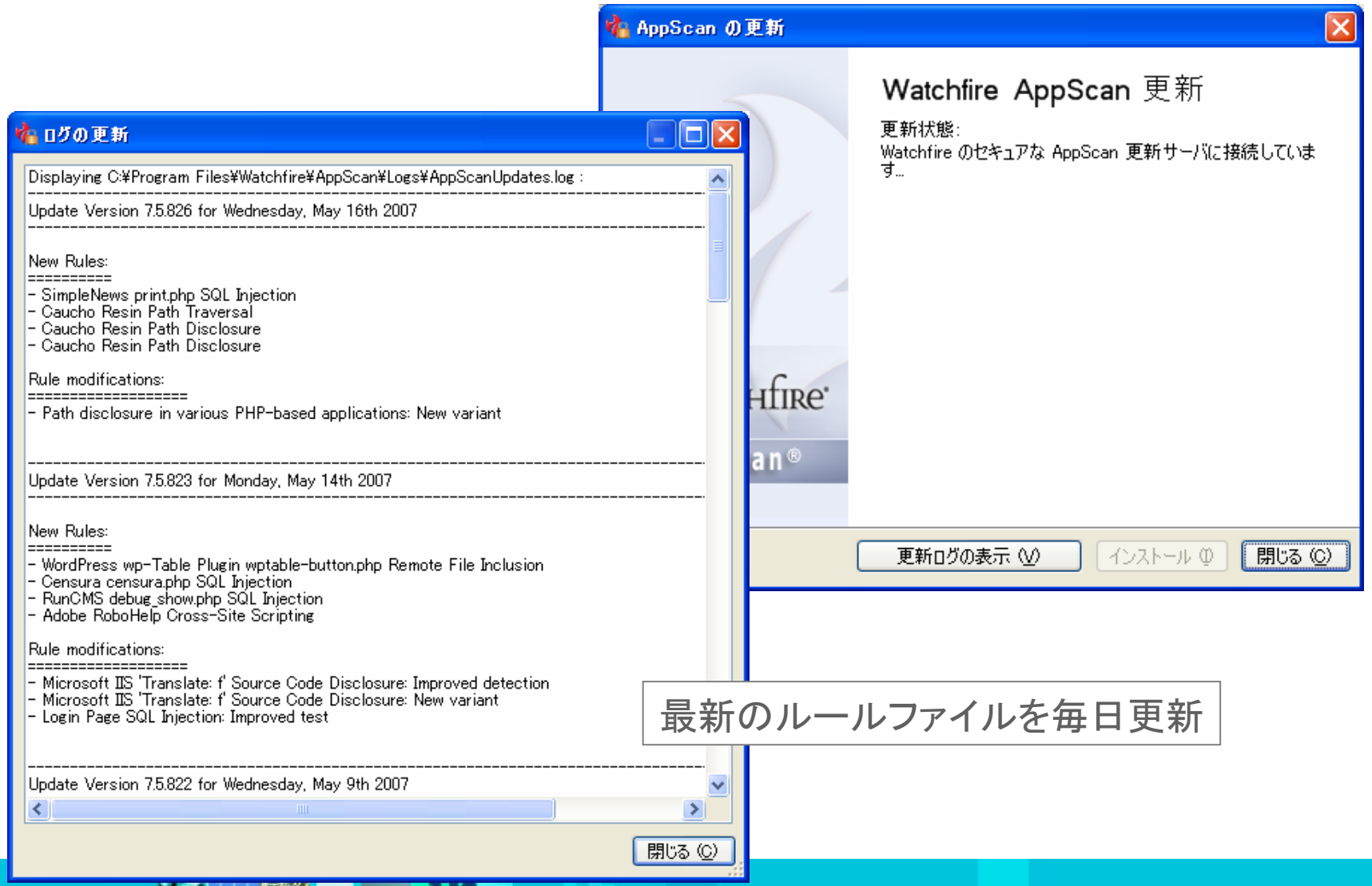
- PIPED Act
- European Directive 1995/46/EC
- European Directive 2005/58/EC
- Japan's Personal Information Protection Act
- Data Protection Act
- California Assembly Bill No. 1950 and Senate Bill 1386
- Children Online Privacy Protection Act (COPPA)
- Electronic Funds and Transfer Act (EFTA)
- Freedom 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



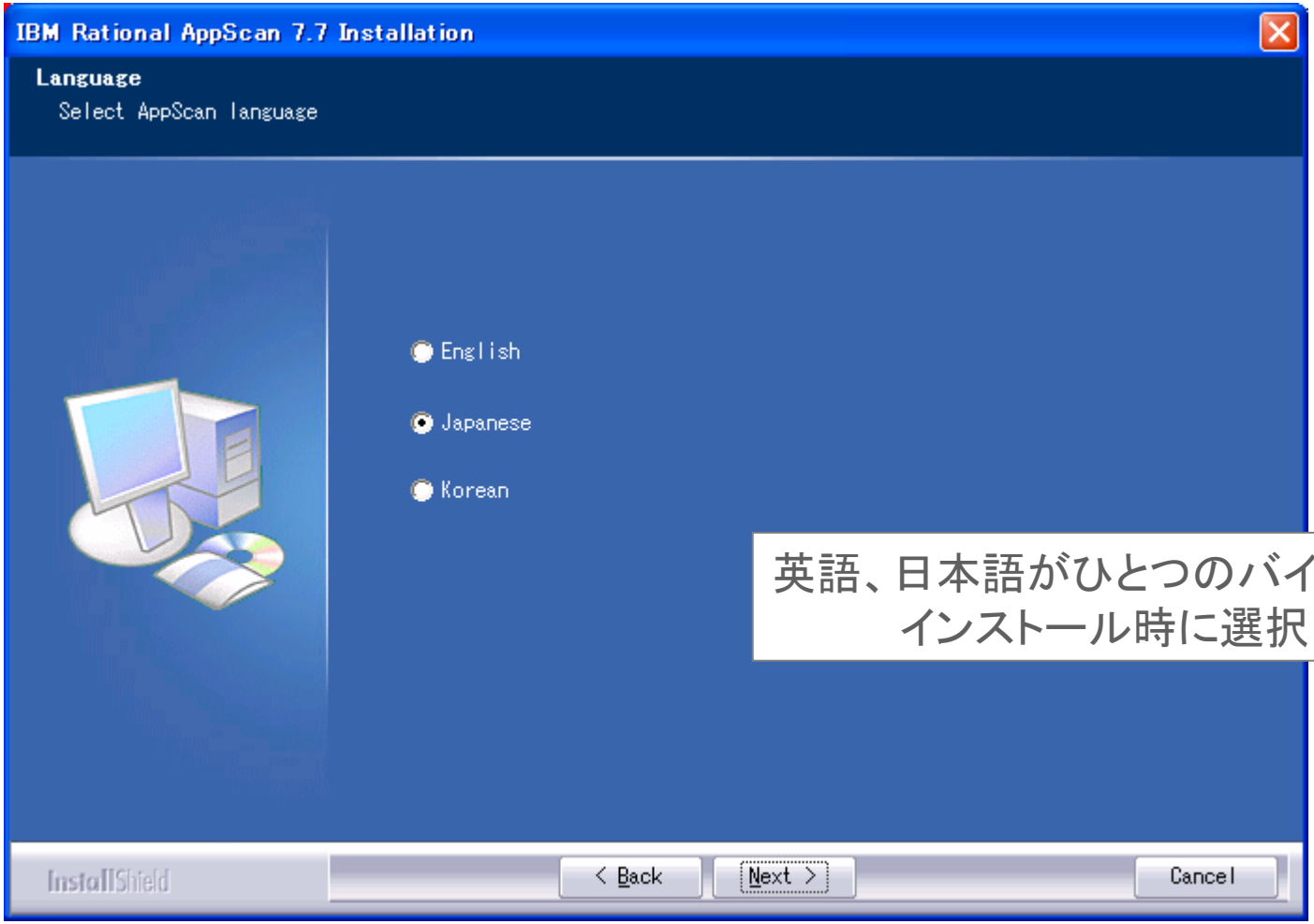
スクリプト言語から AppScan オブジェクトにアクセス可能

AppScan - デイリーアップデート



最新のルールファイルを毎日更新

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
- SIer、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 - SIer、アプリケーション開発にとって

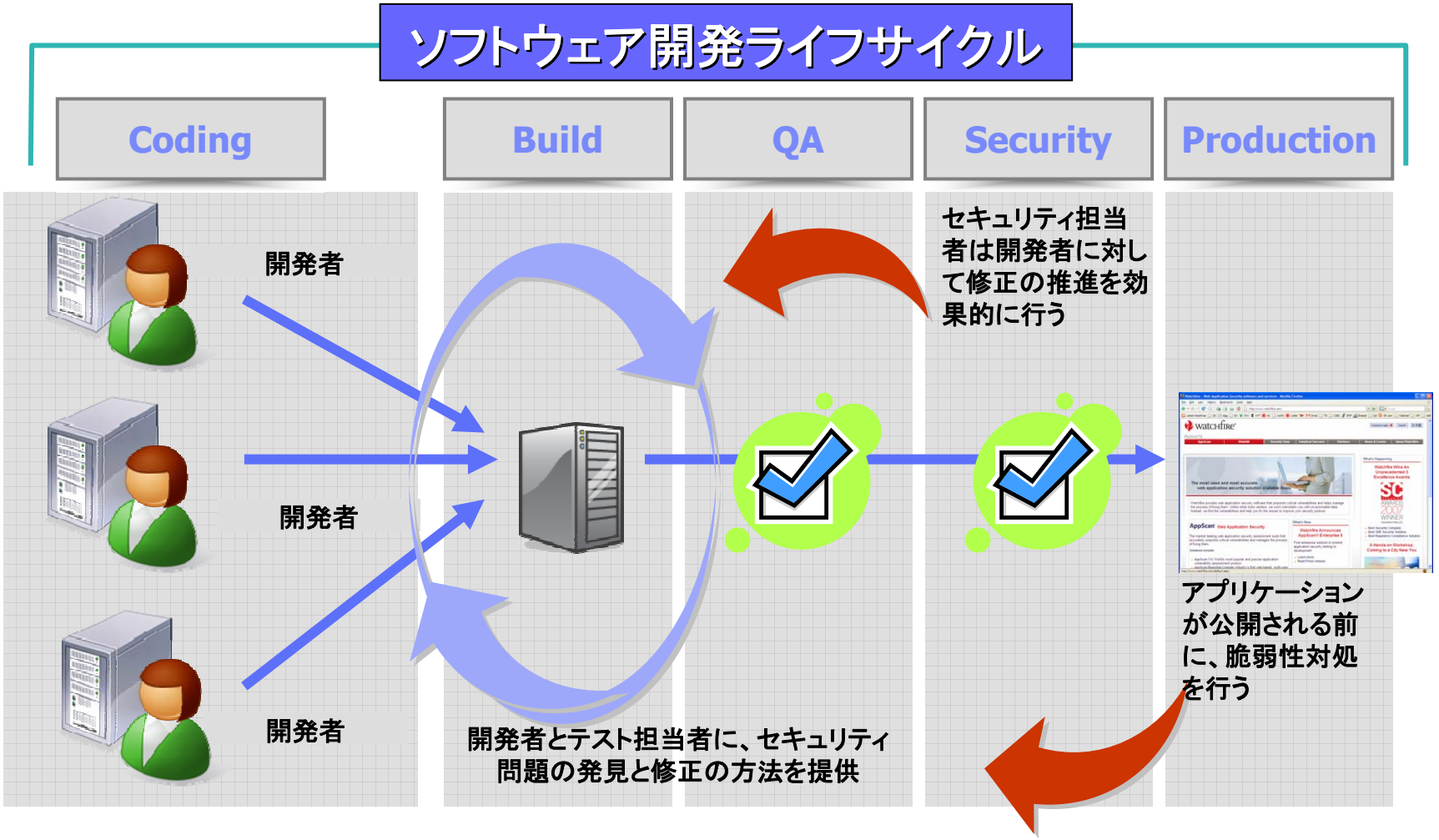
- セキュリティ検査にかかっていたコストの削減
- 高品質な製品の提供
- 人やプロジェクトごとに異なるセキュリティ レベルの標準化

- 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

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。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変わ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ってはいけません。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