## XMLコンソーシアム セキュリティ部会報告

# 情報セキュリティの動向と生産システムへの教訓

2009年06月05日

製造業XMLフォーラム2009

XMLコンソーシアム セキュリティ部会
松永 豊（TELデバイス）








# 情報セキュリティの動向と生産システムへの教訓

本日の内容

- **SQL**インジェクションが生産システムにも脅威になる日が来るか?
  - 情報セキュリティの動向と生産システムへの影響
  - 制御システム向けガイドライン
- 生産工場のセキュリティ検討
  - **SCF2007, MOF2008**
    - リスク、セキュリティ対策、実践









# XMLコンソーシアムの概要

■設立：２００１年6月１８日

■活動目的：　社会基盤＆ビジネス基盤としてのXMLの発展と普及に貢献し、XMLの利活用促進を目指す

■会員：155社、70%：IT関連、30%：ユーザー企業

■会長：鶴保　征城 (IＰＡ　ソフトエンジニアリング・センター所長)

■理事会社：23社　　監事：2名　　顧問：4名

■エバンジェリスト：23名

■特徴：(1)　中立性

(2)　自主的、自発的な活動

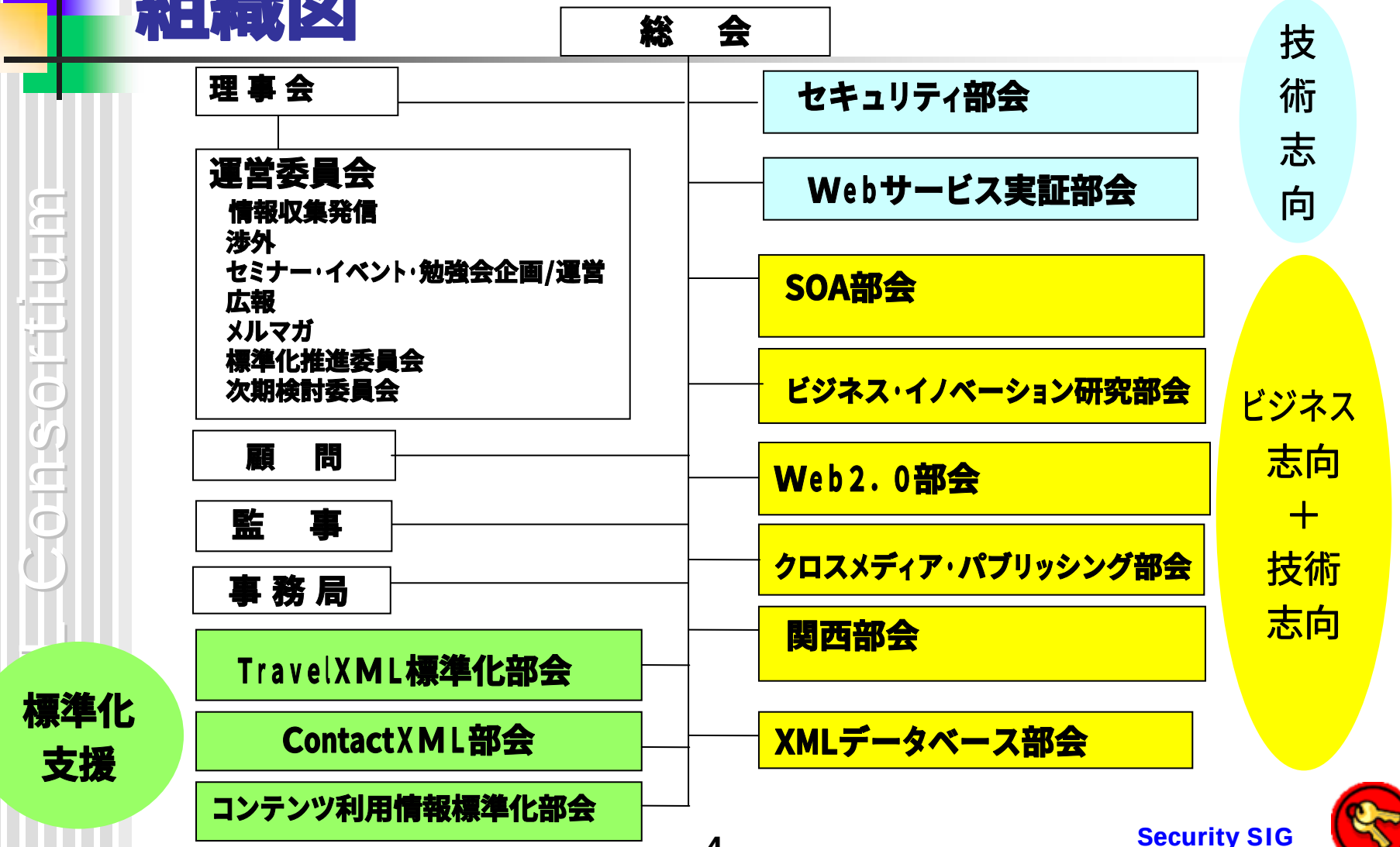
(3)　一社ではできない活動

(4)　ヒューマン・ネットワーク、ビジネス・ネットワークの確立

(5)　成果物の公開

http://www.xmlconsortium.org/

# XMLコンソーシアム組織図



# 生産工場のセキュリティに対する取り組み

- アライアンス・パートナーとの協調活動として、
  生産工場のセキュリティに関する取り組みを行ってきました

製造業XML推進協議会　　PSLXフォーラム

- **2007年 システム コントロール フェア (SCF) 2007**
  - セキュリティ検討報告書
  - 合同デモシステムについてセキュリティ上のリスクを分析し対策を検討
- **2008年 Manufacturing Open Forum (MOF) 2008**
  - 合同デモシステム向けセキュリティ報告書
  - 暗号化と電子署名について具体的な手段を検討して報告

# 情報セキュリティにおける課題の例
# …SQLインジェクション

# SQLインジェクション
## (Webサイトへの不正アクセス)

IPA
INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN
JPCERT CC®

プレスリリース
2009 年 1 月 26 日
独立行政法人情報処理推進機構
有限責任中間法人 JPCERT コーディネーションセンター

ソフトウェア等の脆弱性関連情報に関する届出状況 [2008 年第 4 四半期(10 月～12 月)]

IPA（独立行政法人情報処理推進機構、理事長：西垣 浩司）および JPCERT/CC（有限責任中間法人 JPCERT コーディネーションセンター、代表理事：歌代 和正）は、2008 年第 4 四半期（10 月～12 月）の脆弱性関連情報の届出状況[1]をまとめました。

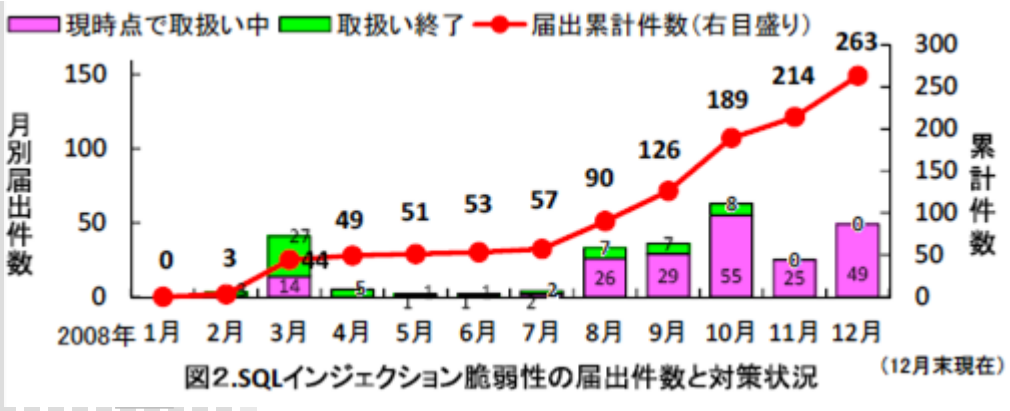


図2.SQLインジェクション脆弱性の届出件数と対策状況

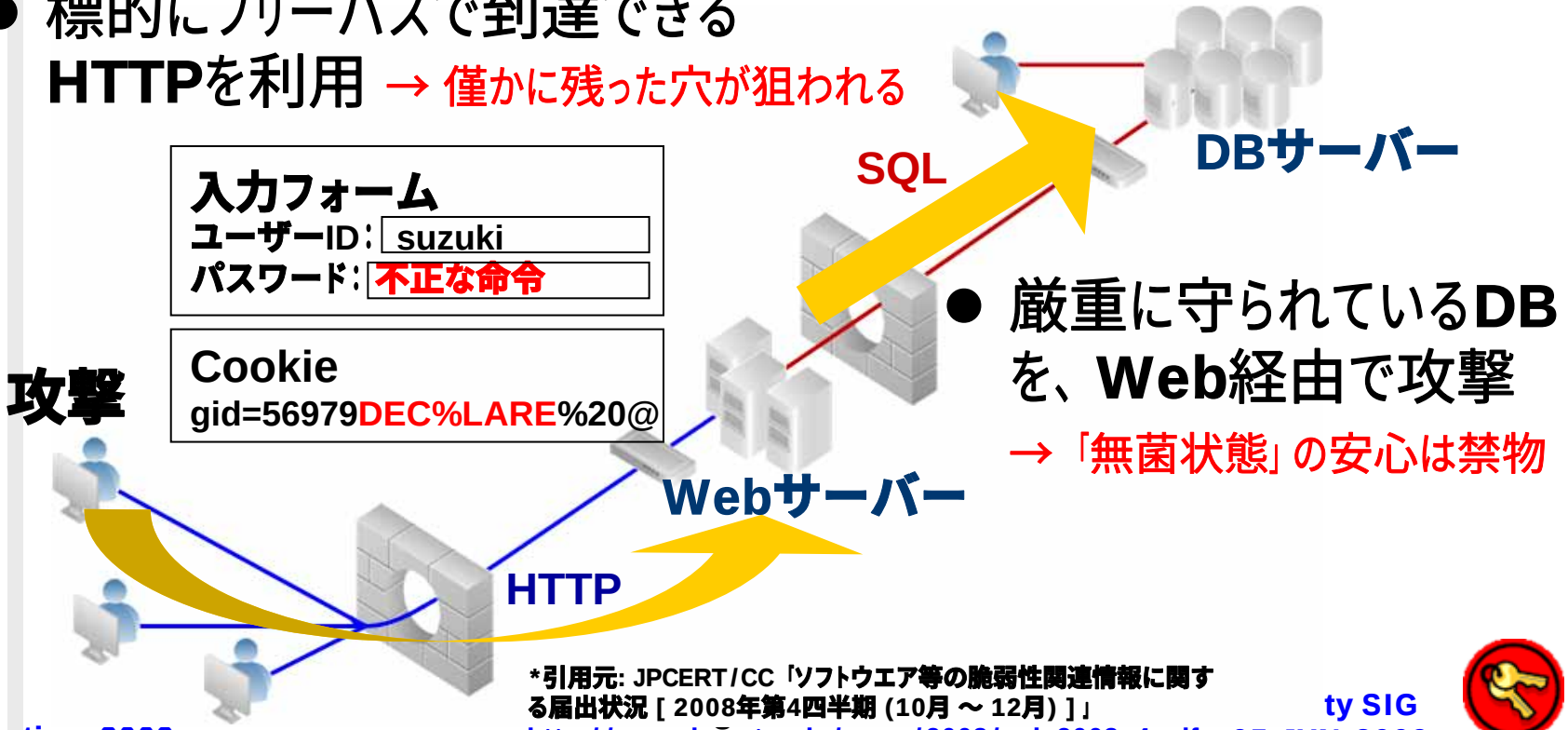
*引用元: JPCERT/CC「ソフトウエア等の脆弱性関連情報に関する届出状況 [ 2008年第4四半期 (10月 ～ 12月) ] 」
http://www.jpcert.or.jp/press/2008/vuln2008q4.pdf

- 「東京都障害者サービス情報」への不正アクセスについて（続報） 2008/12/26 東京都福祉保健局
- セミナー情報ページが改ざん、閲覧でウイルス感染の可能性 - 情報処理サービス会社 Security NEXT - 2008/12/12
- (駒ヶ根市)防犯防災メールに不正リンク　サーバー攻撃が原因 2008-12-11 長野日報
- JR北海道、不正アクセスを受けホームページの運用を一時停止 2008年12月02日 ITmedia
- JA全農のサイトが不正アクセスで改ざん- Security NEXT - 2008/11/17
- 「今までにないタイプのSQLインジェクション」――ゴルフダイジェストへの不正アクセス手口が判明 2008/10/06 ITpro
- CookieにSQLインジェクションを埋め込む新手の手口、ラックが緊急注意喚起　2008/10/03 Enterprise Watch
- 「サウンドハウス」名指しの攻撃マニュアルが中国で公開されていた 2008/04/18 INTERNET Watch
- JPCERT/CC、SQLインジェクションによるWebサイト改ざんを警告 2008年3月14日 RBB Today
- SQLインジェクション攻撃を行なうワームが出現、米SANS Instituteが警告 2008年5月7日 INTERNET Watch
- IPA、SQLインジェクション攻撃の急増をWeb管理者に注意喚起 20080515 INTERNET Watch

# SQLインジェクションの特徴

- 標的はアプリケーション
  → OSのパッチでは対処できない
- 標的にフリーパスで到達できる
  HTTPを利用 → 僅かに残った穴が狙われる

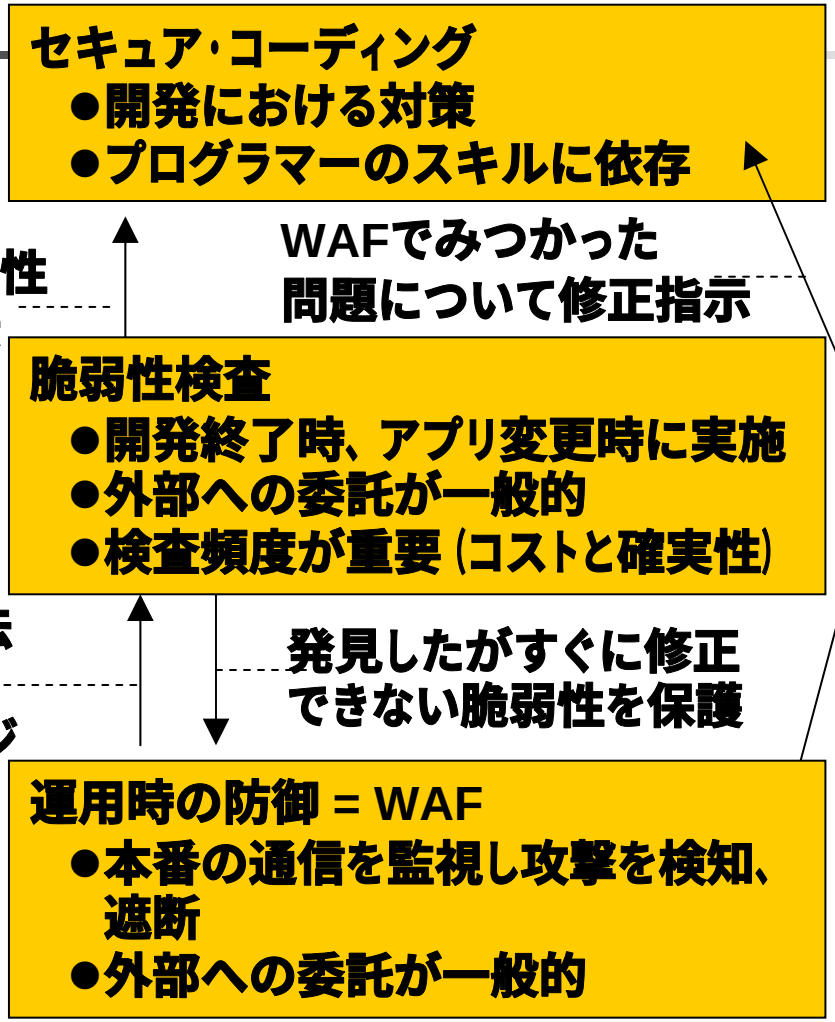
2008 年の届出は263 件で、現時点で取扱い中 (対策中) のものが計202 件 (*)




- 厳重に守られているDBを、Web経由で攻撃
  → 「無菌状態」の安心は禁物

*引用元: JPCERT/CC「ソフトウエア等の脆弱性関連情報に関する届出状況 [ 2008年第4四半期 (10月 ～ 12月) ]」
http://www.jpcert.or.jp/press/2008/vuln2008q4.pdf

# Webアプリケーションの安全対策




- 「どれを実施すればいいのですか?」
  ↓
- いずれも重要です
- 補完し合って堅牢性を高める
- どこまでやるか、は保護対象の性質 / リスクに依存

継続的にセキュリティを維持するために検査頻度を上げると開発日程を束縛する恐れがあり、効率の良い補完手段を求めていた。
その目的を満たすのは、脆弱性を修正する前でも安全性が保てるWAFの導入だった。 (外為どっとコム様)

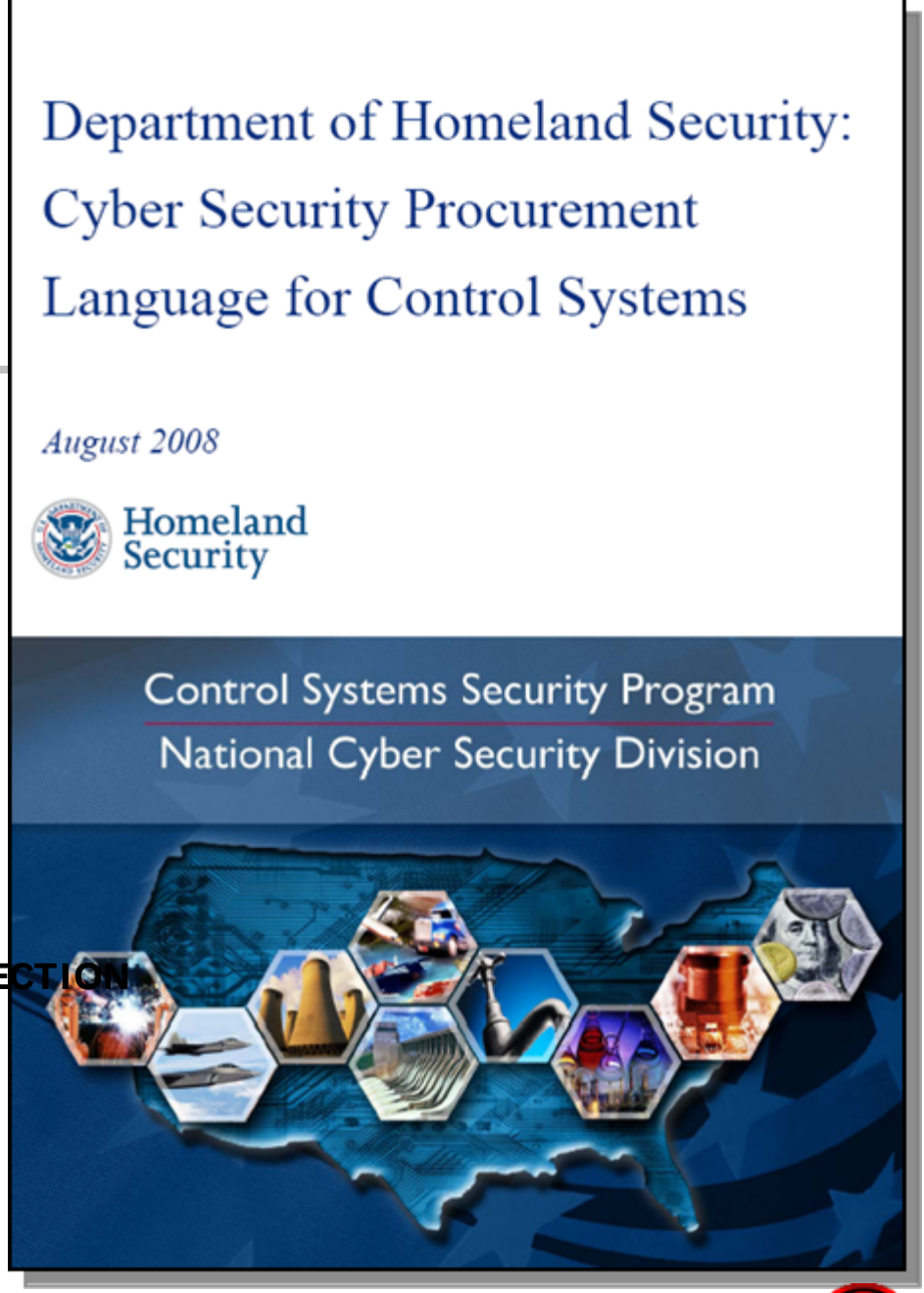
# 生産システムへの教訓

- 注意すべきポイント
  - パッチの適用漏れ
  - アプリケーション
  - 正規の通信、正規の入力
- 対策は、
  計画・開発→運用
  →検査→改善
  (PDCA)

# 制御システムにおけるSQLインジェクション

- 米国土安全保障省による制御システムの為のサイバーセキュリティ調達仕様

2. SYSTEM HARDENING
3. PERIMETER PROTECTION
4. ACCOUNT MANAGEMENT .
5. CODING PRACTICES
6. FLAW REMEDIATION
7. MALWARE DETECTION AND PROTECTION
8. HOST NAME RESOLUTION
9. END DEVICES
10. REMOTE ACCESS
10.4 Web-based Interfaces
11. PHYSICAL SECURITY
12. NETWORK PARTITIONING




http://www.us-cert.gov/control_systems/

## 10.4 Web-based Interfaces

Many control systems have Web-based interfaces for performing some tasks.

### 10.4.1 Basis

Web-based interfaces to control systems are gaining popularity and are often poorly designed and configured making these interfaces vulnerable to exploits.

### 10.4.2 Language Guidance

Web applications are often vulnerable to injection attacks of several varieties including command injection, Remote File Include (RFI) and Cross-Site Scripting (XSS). Web applications with a database back-end commonly mishandle Structured Query Language (SQL) statements as well, allowing SQL injection. Additionally, the HTTP servers on which these applications are hosted can be vulnerable to buffer overflows or other memory corruption attacks. Another common mistake in Web applications is directory traversal, which allows attackers access to more files than the programmer intended. Web applications in embedded devices are often written in a low-level language like C and are potentially vulnerable to buffer overflows.

Other non-HTTP services are also commonly
combination of these services can lead to greater

**Authentication.** Web interfaces typically co
information, therefore authentication is essential
about the system. Poorly implemented interfaces
security provided by authentication. Authenticati
flaws, allowing an attacker to gain database acce
compromise of the host or device.

**RFI.** Remote File Include (RFI) vulnerabiliti
applications written in the PHP (hypertext prepro
successful, it results in the attacker running arbitr
equivalent to full-host compromise.

**Input Validation.** String input validation is needed to prevent command injection, which can lead to complete host compromise. Like SQL injection, command injection can be accomplished by inputting characters that the application treats specially. The specific characters used will depend on the target

62

system, but commonly include those in the following (non-exhaustive) list: $ % ! ` ; ' " \. Flaws of this nature are usually easy to find, are relatively simple, and provide access to an attacker as the user running the HTTP server. When combined, these factors make command injection a dangerous vulnerability that must be addressed.

**Cross-Site Scripting (XSS).** There are two basic types of XSS: reflected and persistent. In a reflected XSS vulnerability, the attacker must convince a user to visit a malicious Web site or click on a malicious link. The persistent variety, in which the exploit is stored on the target server itself, is less common but more likely to succeed in a control system environment because using the Web application is sufficient to trigger the exploit. Regardless of how XSS is launched, it works by running JavaScript on the user's browser in the context of the target Web page. This allows an attacker to steal the user's cookies, thereby gaining access as that user.

Like other types of software, Web applications need to be designed and developed with security in mind.

■背後にデータベースを持つWebアプリケーションではSQL文の取り扱いを誤ることもよくあり、SQLインジェクションを許すこととなる。

### 10.4.3 Procurement Language

The Vendor shall provide physical and cyber security features, including but not limited to authentication, encryption, access control, event and communication logging, monitoring, and alarming to protect the system from unauthorized modification or use.

The Vendor shall clearly identify the physical and cyber security features and provide the methodology(ies) for maintaining the features including the methods to change settings from the Vendor-configured or manufacturer default conditions.

The Vendor shall verify that the addition of security features does not adversely affect connectivity, latency, bandwidth, response time, and throughput, including during the SAT when connected to existing equipment.

The Vendor shall remove or disable all software components and services that are not required for the operation and maintenance of the devices that run an HTTP server prior to the FAT. The Vendor shall provide documentation on what is removed and/or disabled.

The Vendor shall provide, within a prenegotiated period, appropriate software and service updates and/or workarounds to mitigate all vulnerabilities associated with the product and to maintain the established level of system security.

The Vendor shall verify and provide documentation that the SIS is certified after incorporating the security devices.

The Vendor shall provide documentation of input sanitization for all Web-form inputs, including, but not limited to, measures for prevention of command injection, SQL injection, directory traversal, RFI, XSS, and buffer overflow.

The Vendor shall follow secure coding practices and reporting for all Web-based interface software (see Section 5.1). This requirement includes both Web applications and Web servers.

The Vendor shall provide user configurable and managed passwords (see Section 4.3).

The Vendor shall provide an independent third-party security code validation of all Web-based interface software (see Section 5.1).

63

■納入者は全ての**Web**フォーム入力欄について入力サニタイジング**(**消毒、無害化**)**の記録を提出し、これは最低でもコマンド・インジェクションや**SQL**インジェクション、ディレクトリ・トラバーサル、**RFI**、**XSS**、およびバッファ・オーバーフローの防止手段を含んでいなければならない。

# SQLインジェクション参考資料


- プロなら知っておきたいWebサイトの強化策 (アスキームック)
  - Webセキュリティ完全防御マニュアル、WAFの存在意義と製品の種類、WAFの防御メカニズム、WAFの導入とその課題
  - その他Webの性能チューニングなど

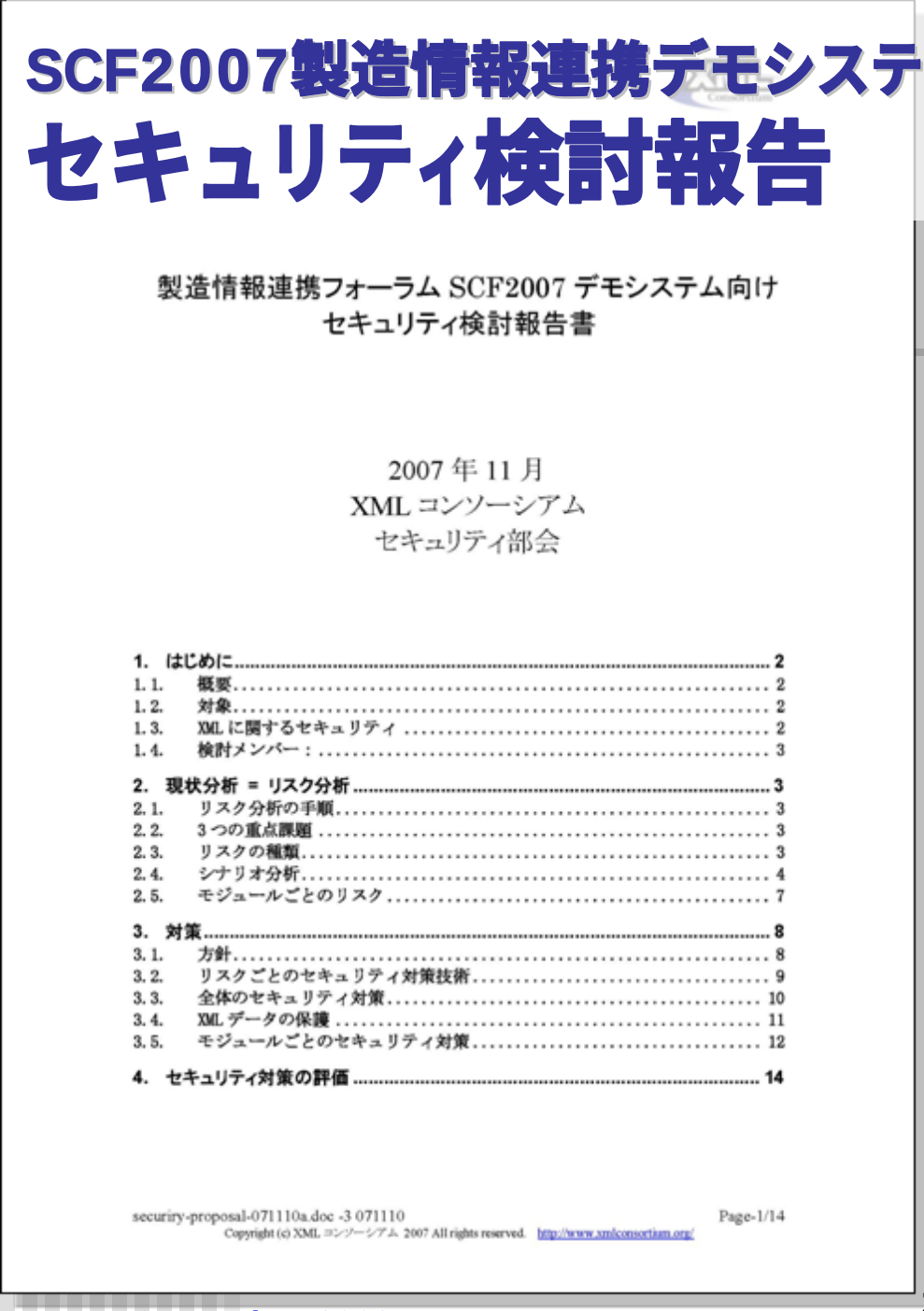
- まずできること：IPA SQLインジェクション検出ツール iLogScanner V2.0
- ウェブサイト運営者のための脆弱性対応ガイド 2008年 4月 4日 第2版、情報システム等の脆弱性情報の取扱いに関する研究会編
- NPO 日本ネットワークセキュリティ協会、2007年度 情報セキュリティインシデント調査報告書
- 安全な ウェブサイトの 作り方 改訂第3版、独立行政法人 情報処理推進機構
- 株式会社ラック【2008 侵入傾向分析レポートvol.11】

# 生産工場のセキュリティ

**SCF2007**

**MOF2008**

# SCF2007製造情報連携デモシステム
# セキュリティ検討報告



製造情報連携フォーラム SCF2007 デモシステム向け
セキュリティ検討報告書

2007 年 11 月
XML コンソーシアム
セキュリティ部会





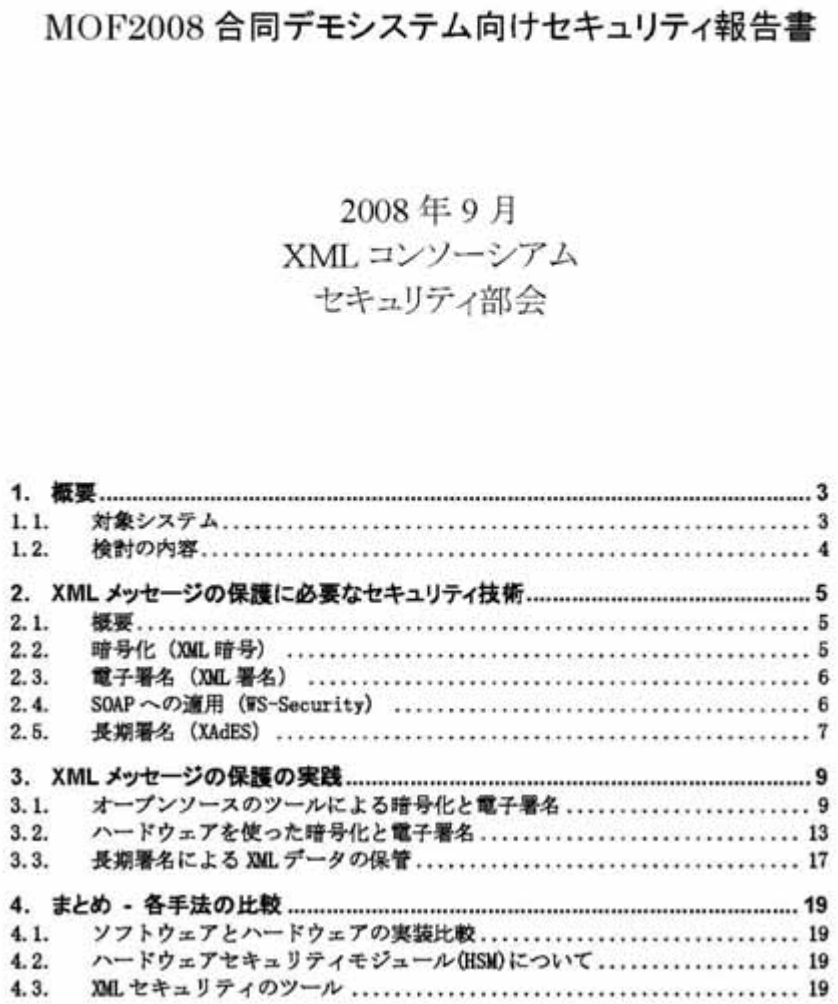

■システム コントロール フェア 2007
- 2007/11/13(火)～16(金)
  東京ビッグサイト

■XMLコンソーシアムのセキュリティ部会が製造情報連携フォーラムの合同デモシステムに対してセキュリティ面のリスクと対策を検討→報告

■主なセキュリティ対策
- 認証の一元化
- アクセス制御
- ログ管理、システム監視
- XMLデータの保護、WS-Security

**報告書**:
http://www.xmlconsortium.org

# MOF2008合同デモシステム向けセキュリティ報告書



MOF2008 合同デモシステム向けセキュリティ報告書

2008 年 9 月
XML コンソーシアム
セキュリティ部会



■**MOF2008**

- http://www.mstc.or.jp/iaf/mof2008/
- 名称：マニュファクチャリング　オープン　フォーラム２００８　（ＭＯＦ２００８）
- **2008/9/10~12** 東京ビックサイト
- 主催：ＩＡ懇談会

■前年の**SCF2007**に引き続き、合同デモシステムのセキュリティを検討

**報告書：**
http://www.xmlconsortium.org/public_doc/mof2008_security/mof2008.html

# MOF2008合同デモシステム向けセキュリティ報告書

1. 概要
1.1. 対象システム
1.2. 検討の内容
2. XMLメッセージの保護に必要なセキュリティ技術
2.1. 概要
2.2. 暗号化 (XML暗号)
2.3. 電子署名 (XML署名)
2.4. SOAPへの適用 (WS-Security)
2.5. 長期署名 (XAdES)
3. XMLメッセージの保護の実践
3.1. オープンソースのツールによる暗号化と電子署名
3.2. ハードウェアを使った暗号化と電子署名
3.3. 長期署名によるXMLデータの保管
4. まとめ - 各手法の比較
4.1. ソフトウェアとハードウェアの実装比較
4.2. ハードウェアセキュリティモジュール(HSM)について
4.3. XMLセキュリティのツール

**SCF2007では、セキュリティ要件を網羅的に検討**

↓

**今回は、実装面も含めたより具体的な情報提供を目的**

- **機密性 – 生産計画などの漏洩防止**
  **→ 暗号化**
- **完全性 – 生産記録の改竄防止**
  **→ 電子署名**

**それぞれについて、**
- **技術解説**
- **具体的な手段**
- **手段ごとの特徴や課題**

# オープンソースのツールによる暗号実践

```
// AESによる私有鍵の生成
    KeyGenerator keyGenerator =  KeyGenerator.getInstance("AES");
    keyGenerator.init(128);
    Key symmetricKey = keyGenerator.generateKey();
// 私有鍵を暗号化するDESによる共通鍵の生成
    keyGenerator = KeyGenerator.getInstance("DESede");
    SecretKey kek = keyGenerator.generateKey();
    String algorithmURI = XMLCipher.TRIPLEDES_KeyWrap;
    XMLCipher keyCipher = XMLCipher.getInstance(algorithmURI);
    keyCipher.init(XMLCipher.WRAP_MODE, kek);
//共通鍵の暗号化
    EncryptedKey encryptedKey = keyCipher.encryptKey(document, symmetricKey);
// 暗号化方法の設定
    XMLCipher xmlCipher = XMLCipher.getInstance(algorithmURI);
    xmlCipher.init(XMLCipher.ENCRYPT_MODE, symmetricKey);
// 暗号化データに組み込む<KeyInfo>要素の設定
    EncryptedData encryptedData = xmlCipher.getEncryptedData();
    KeyInfo keyInfo = new KeyInfo(document);
    keyInfo.add(encryptedKey);
    encryptedData.setKeyInfo(keyInfo);
// 暗号化する要素名
    Element enElement = (Element) document.getElementsByTagName("要素名").item(int 要素位置);
 // 暗号化すべきデータをEncrypteData要素により置換
     xmlCipher.doFinal(document, enElement, true);
```

# XMLコンソーシアム セキュリティ部会の 今後の計画

- **XMLセキュリティのツール調査、ツール検証**
  - ツール調査「どんなツールがあるのか？」
    - アプリケーションサーバー、ライブラリ、ゲートウェイ/**ESB**、テストツール
    - 機能分野：**XML**暗号化、電子署名、**XML**/**SOAP**ファイアウォール
    - 現在調査中
  - ツール検証「どうやって使うのか？」
    - **MOF2008**合同デモシステム向けセキュリティ報告書での「**XML**メッセージの保護の実践」の継続検証
    - プラットフォーム（**Windows**，**Linux**，・・）
    - **Java6**対応状況
    - **Web**サービス実証部会との協業

# XML暗号化検証報告まとめ

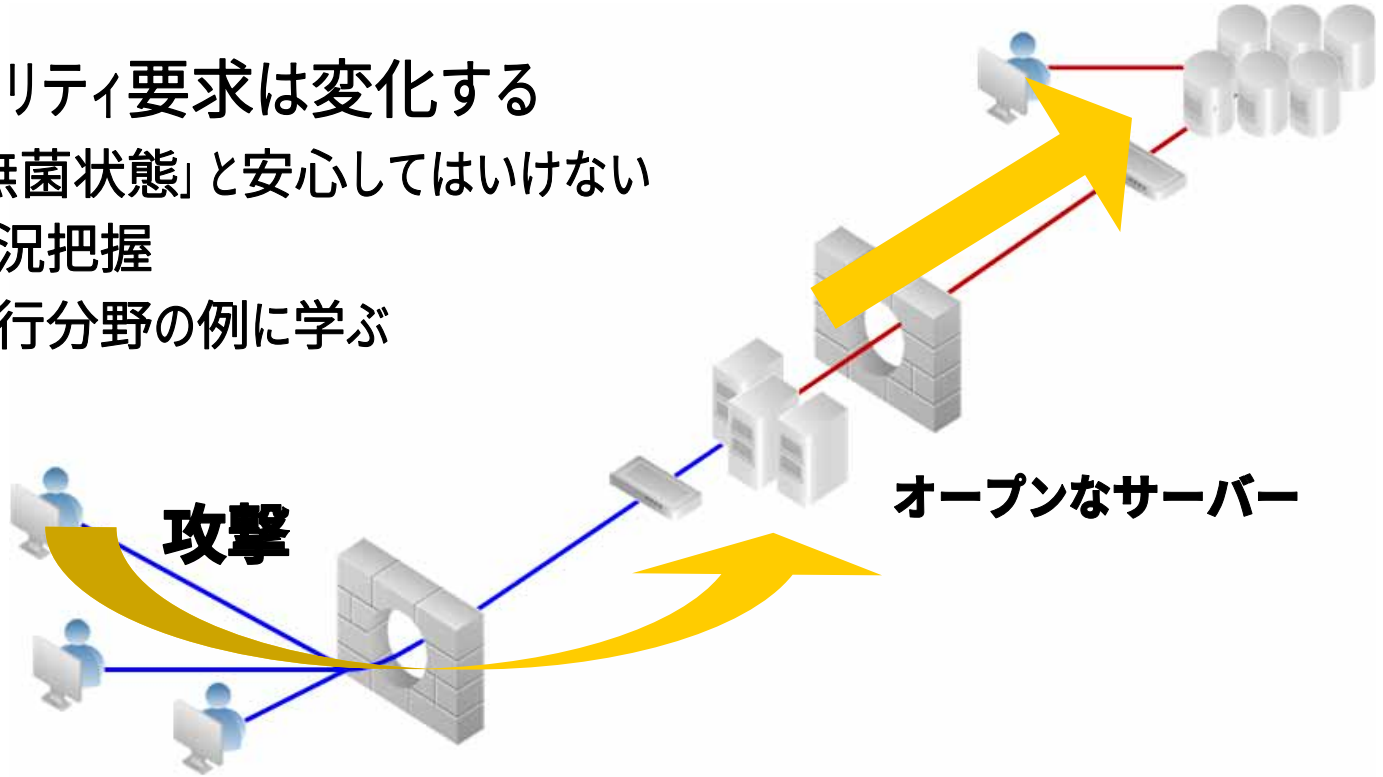
## (XMLコンソーシアムWeek 2009/05/12より、資料公開中)

| 復号 \ 暗号化 | Apache | JSR-106 | .net |
|---|---|---|---|
| Apache | ◎ | ◎ | ○*1 |
| JSR-106 | ○ *3 | ◎ | ○ *2 *3 |
| .net | ○ *4 | ◎ | ◎ |

◎：サンプルプログラムのままで復号できた
○：対応策を入れることで復号できた
*1：JCE (Java Cryptography Extension) を入れてAES256に対応
*2：ポリシーファイルを入れ替えて、AES256に対応
*3：暗号化時に ISO10126Padding ではなく PKCS5Padding を指定
*4：暗号化時に /EncryptedData/KeyInfo/EncryptedKey/KeyInfo/KeyName を追加

# まとめ

- セキュリティ要求は変化する
  - 「無菌状態」と安心してはいけない
  - 状況把握
  - 先行分野の例に学ぶ




www.xmlconsortium.org

XMLコンソーシアム セキュリティ部会 リーダー 松永 豊
wg-sec-ldr@xmlconsortium.org