危険なコード

SourceForge.netのPHPアプリケーションから

大垣 靖男
yohgaki@ohgaki.net

このファイルはPHP Conference 2006のプレゼンテーションファイル
から一般に公開するには不適切な部分を削除したバージョンです

セキュリティの話は退屈?

- 同じ話ばっかりで聞き飽きた!
- 本当に問題なところはどこ?
  - 今日の題材：sourceforge.netのPHPで作成されたアプリケーションTOP50の危険なコードを検索&解説
- セキュリティ対策に魔法の杖は無い!
  - アーキテクト・プログラマ、が安全な設計・コーディングを行うしかない

Webアプリのセキュリティは難しい?

- プログラミングの基本は簡単
  - セキュリティリスクはサブシステムとのバウンダリで発生
  - バウンダリで「入力時」に確実にバリデーション、「出力時」に確実にエスケープ処理を行えばセキュリティ上の問題は発生しない
- 基本的な攻撃手法
  - インジェクション
    - JavaScript,SQL,LDAP,XML
    - 概ねコーディング上の問題が原因
  - 情報漏えい
    - 盗聴(Tapping, AP Phishing),キャッシュ,秘密情報の公開化
    - 概ね設計上の問題が原因

Webアプリのセキュリティは難しい?(続き)

- 設計レベルの問題
  - 不適切なセッション管理
  - 不適切な認証管理
  - 不適切な権限管理
  - 不適切なページデザイン
  - 不適切な情報管理
- 運用レベルの問題
  - 不適切なバックアップ
  - 不適切な設定
  - 不適切なデータアクセス端末管理
  - 不適切なシステム管理者ID管理
- その他多数...

注意事項

- ソースは全て8/13時点でリリースされていたソース
- sf.netのトップ50からPHPアプリを「適当」に選択
  - 概ねトップ30～50のアプリケーション
- 実際にアプリは使用（動作）していない
- 正式な検証作業ではない
  - ソースコードを全て解読していない
  - 全ての脆弱性の可能性を検証していない
  - ホワイトボックステストのみ

注意事項(続き)

- 「危険なコード」!=「セキュリティホール」
  - 局所的には脆弱でも攻撃可能な経路がない限りセキュリティホールにはならない
- 例えば、C言語の場合
  - 「strcpy(dst, src)」の利用は薦められない。安全なコードはstrncpyを使用すべき。
    - 他にもstrcat, memcpy, などバッファオーバーフローセキュリティホールを作る関数は多数ある。しかし、これらの関数を利用する危険コードでも安全なアプリケーションは作成可能。
- PHPの場合、register_globals=onを前提とするコード(安易なregister_globals=off対応を含む)は「危険なコード」
  - ただし、「危険なコード」でも「安全なアプリ」は構築可能

注意事項(続き)

- 本プレゼンテーションの目的は揚げ足取りではありません!
- 危険なコーディングを知る
- より安全なコーディングを知る
- 比較的人気のある(?)アプリケーションのセキュリティレベルを知る
- できれば危険なコードを読み取れるようになる
- できればより安全なコードを書けるようになる

Open Computers and Software Inventory

- バージョン：1.0RC3-1
- サイズ：約4MB
- URL: http://ocsinventory.sourceforge.net/

- OCS Inventory NG, Open Computers and Software Inventory Next Generation is an application designed to help administrator keep track of the computers configuration and installed softwares. Low network traffic HTTP communications between agents and server
- インベントリ管理のソフトウェア（らしい）

- この手のシステム管理用アプリケーションはセキュリティ対策がずさんなアプリケーションが比較的多い
  - 通常システム管理者はシステム上のデータへのアクセスができる場合が多いので、あえてセキュアなコードにする必要がない

PHPWebsite 1.0.0RC2

- バージョン：1.0.0RC2
- サイズ：約23MB
- URL: http://phpwebsite.appstate.edu/

- Developed by the Web Technology Group at Appalachian State University, phpWebSite provides a complete web site content management system (CMS). All client output is XHTML 1.0 and meets the W3C's Web Accessibility Initiative requirements.
- コンテンツマネジメントシステム（らしい）

Uber Uploader

- バージョン：3.2
- サイズ：約56KB
- URL： http://sourceforge.net/projects/uber-uploader

- Uber Uploader is a web based file upload mechanism that displays the status of an upload in the form of a progress bar using AJAX.
- ファイルアップローダ（らしい）

Web Calendar

- バージョン：1.0.4
- URL： http://www.k5n.us/webcalendar.php

- **WebCalendar** is a **PHP-based calendar application** that can be configured as a single-user calendar, a **multi-user calendar** for groups of users, or as an **event calendar** viewable by visitors. MySQL, PostgreSQL, Oracle, DB2, Interbase, MS SQL Server, or ODBC is required.
- Web用のスケジュール管理ソフト

Sitesys

- バージョン：0.0.3
- サイズ：約280KB
- URL： http://sourceforge.net/projects/sitesys/

- SiteSys is a website management system in PHP which allows a web designer to design a site and then hand over creation of all new content, including pages, news items, and articles, to the client, reducing their work load.
- たぶんCMS

Covide

- バージョン：0.6.1
- サイズ：約49MB
- URL: http://www.covide.nl/

- **Covide offers you OpenSource and webbased Groupware-CRM with possibilities for integrated usage as firewall, mail-, web-, and fileserver AND VoIP PBX!**
- グループウェア+CRMらしい

RoundCubeMail

- バージョン：0.1 beta2
- URL：http://www.roundcube.net/

- RoundCube Webmail is a browser-based multilingual IMAP client with anapplication-like user interface. It provides full functionality you expect froman e-mail client, including MIME support, address book, folder manipulation andmessage filters. RoundCube Webmail is written in PHP and requires the MySQLdatabase. The user interface is fully skinnable using XHTML and CSS 2.
- Webメールアプリケーション

まとめ

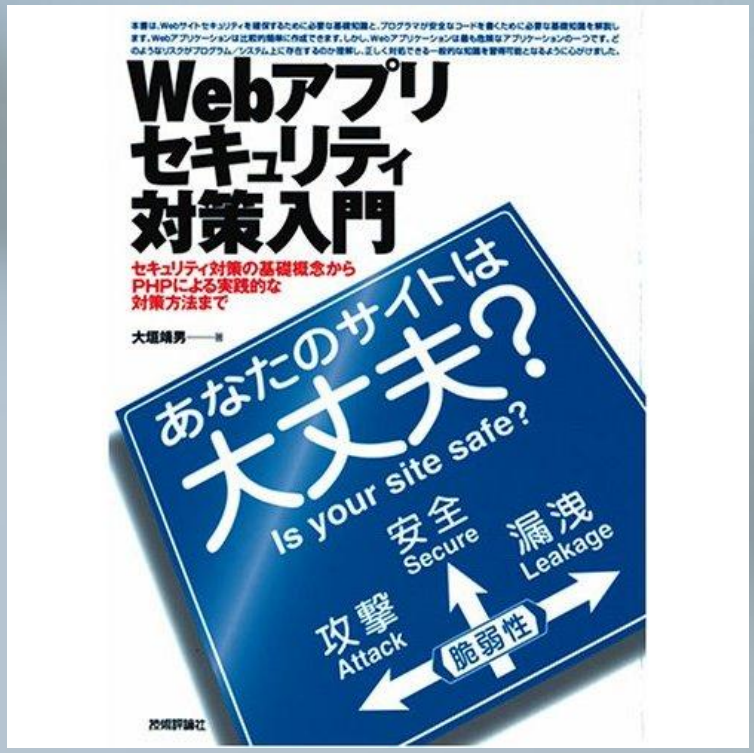
- 全てのアプリケーションが不適切なコードを含む
  - つまり参考にしてはならないコードを含む
- 一部アプリには明らかなセキュリティホールもある…
  - 本当に直ぐに攻撃可能な物も…
- コードを精査していません
  - 他にもある（はず）

安全なコードの書き方

- 脆弱性が発生する原因とリスクを正しく理解
- 入力時の完全な検証
  - ホワイトリスト
- 出力時の完全なフィルタ処理
  - 全て処理

宣伝

- 詳しく安全なWebアプリの書き方を知りたい方は
  - 「Webアプリセキュリティ対策入門」技術評論社



終わり

ご清聴ありがとうございました