TechStyle
Quality for All Professionals

Internet Week 2003
tutorial – T13

セキュアなWEBアプリケーション開発・構築
～ユーザーに不安を感じさせないサイトとは～

岡田 良太郎

株式会社テックスタイル
http://techstyle.jp/
riotaro@techstyle.jp

December 3, 2003

TechStyle
Quality for All Professionals

about Speaker

- 岡田 良太郎
  - 1989年 神戸市立神戸工業高等専門学校電気工学科卒業
  - 1999年 日本Linux協会運営委員
  - 2001年 有限会社テューンビズ代表取締役就任
    Allabout Linuxガイド担当
  - 2002年 株式会社テックスタイル代表取締役就任
    PHPカンファレンス2002プログラム委員長
    CISA
  - 2003年　PHPカンファレンス2003プログラム委員長
    http://okdt.org/

- 本資料・講演に関するお問い合わせは岡田（riotaro@techstyle.jp）まで。

TechStyle
Quality for All Professionals

Agenda

- 0. Webアプリケーションのミッション
- 1. Webアプリケーションのセキュリティ脅威
  - 脅威の整理
  - Web改ざんの実例
  - 情報漏えいの実例
- 2. Webアプリケーションをセキュアにする技術
  - ネットワーク
  - フォーム・CGI
  - クロスサイトスクリプティング
  - サーバ運用
  - コンテンツ運用体制にかかわる別の問題
- 3. Webセキュリティ保持の仕組みづくり
  - 静的コンテンツ
  - エラーメッセージ
  - ナビゲーション・デザイン
  - 情報収集

TechStyle
Quality for All Professionals

0. Webアプリケーションのミッション

TechStyle
Quality for All Professionals

ユーザ感覚を取り戻すセッション

- Webの悪いところは何ですか？
- Webの優れたところは何ですか？

TechStyle
Quality for All Professionals

Webアプリケーションの要求仕様

- Webの飽和が意味すること
  - サイト訪問＝「選ばれた」
  - 「悪いところ」があるサイトを使用する義務はない
- 対外的に「最も危険な」システムである
  - 攻撃に耐えられる保護・予防策
  - 問題発生時のフェイルオーバー

TechStyle
Quality for All Professionals

1. Webアプリケーションのセキュリティ脅威

- 実例 - Web改ざん・情報漏えい
- 脅威の整理

TechStyle
Quality for All Professionals

攻撃の対象と依存関係

- コンテンツ・アプリケーション
  - コンテンツ
  - プログラム
  - データ
- 人的依存
  - コンテンツ制作体制
  - 管理体制
  - ユーザサポート
- 物理的依存
  - 電源
  - 空調
  - ハードウエア
  - メディア
- ネットワーク的依存
  - サーバネットワーク
  - ホスティング同居ドメイン
  - ISP、ルーティング、DNS
  - バックアップ

TechStyle
Quality for All Professionals

脅威行動のモデル

- なりすまし
  - Spoofing identity
- データ改ざん
  - Tampering with data
- 否認（とぼけ）
  - Repudiation
- 情報漏えい
  - Information disclosure
- サービス停止
  - Denial of Service attack
- アクセス権の昇格
  - Elevation of Privilege

TechStyle
Quality for All Professionals

セキュリティ3要素

- 機密性：対象（人）、内容（データ）の関係を保証
  - ユーザID、管理権限、パーソナリゼーション
- 完全性：正確な伝達
  - メール、メッセージ
- 可用性：期待されるときはいつでも応じられる
  - パフォーマンス、アベイラビリティ

cf. What types of attacks exist?
There are three main types of attacks (Saltzer 1975):

1. Unauthorized release of privileged information.
2. Unauthorized modification of privileged information.
3. Denial of service.

TechStyle
Quality for All Professionals

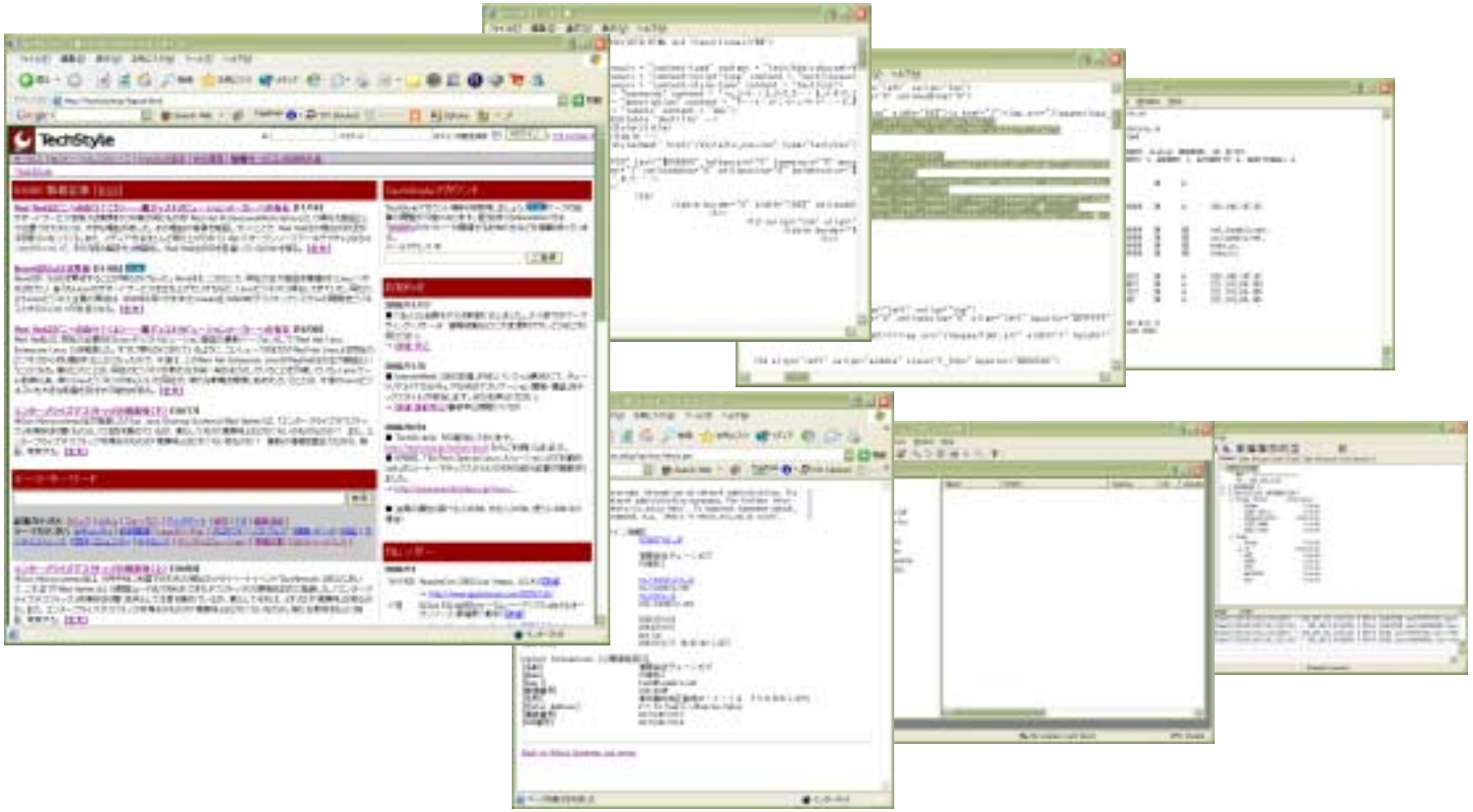
2. Webアプリケーションをセキュアにする技術

- 基本原則
- ネットワーク
- フォーム・CGI
- クロスサイトスクリプティング
- サーバ運用
- コンテンツ運用体制にかかわる別の問題

TechStyle
Quality for All Professionals

技術スキルと関心ポイントの関係

- 「技術スキルと、WEBサイト表面からの深さは比例する」



TechStyle
Quality for All Professionals

セキュリティ方策の基本原則

- まずDenyから始める
- 権限を集中させない
- ひとつの仕組みに依存しない
- シンプルである
- オープンである
- 完全な仲介にこだわる
- 同じ仕組みを用いない
- 簡単に使える

TechStyle
Quality for All Professionals

リスクは「表示」から始まる

- エラーメッセージ
  - サーバのエラーメッセージ
    - 「コンテンツがない」,「権限がない」
  - アプリケーションのエラーメッセージ
    - 「～が見当たらない」,「セッションが～」,「ログインできない」
  - プログラムのスコープはURLのスコープより広い
    - CGIのプログラムの悪用
    - プログラムの「デバッグモード」の用い方に注意

TechStyle
Quality for All Professionals

次のリスクは「入力」から来る

- ユーザが入力できるところ
  - FORMデータ
  - URL(アドレス)
  - hiddenデータ
  - メール

- 影響
  - SQLクエリ文字列
  - クロスサイトスクリプティング
  - includeするリモートURL
  - WEBコンテンツの違法な貼り付け
  - 認証のすり抜け
  - 認証情報の漏洩
  - Referer詐称
  - ファイルアップロード
  - バッファオーバーフロー
  - シェルコマンドの実行
  - CGIソースコードの漏洩

TechStyle
Quality for All Professionals

プログラミングによるリスク

- バッファオーバーフロー
  - スタック上のバッファサイズより大きなデータのコピーによって発生
  - プログラムのスコープは広い。シェルのスコープはもっと広い。

- Format String（書式指定文字列）問題
  - 可変個の引数をとる関数で、実際に渡される引数の個数がわからないことに起因

```
printf(inputbuf);
// inputbuf に書式指定文字が含まれていたら？

prinft(%s, inputbuf);
// 安全なコード
```

TechStyle
Quality for All Professionals

プログラミングによるリスク

- 汎用・フリーCGIの設定の罠
  - 汎用CGIのパラメータはどのように指定されるか
    - どこかからかもらってきたコードのデフォルトのまま？
    - 管理画面のURL
    - ユーザデータファイル名、エラーページ名をhiddenで指定
    - ソフトウエアのアップデートの問題
  - 「トロイの木馬」化してしまう可能性のある機能
    - アクセス監視
    - 広告システム
    - …

TechStyle
Quality for All Professionals

ネットワークによるリスク

- DNS、大丈夫ですか
  - DNSサーバののっとり
  - ISPのTransfer
  - 技術責任者の書き換え
  - テストサーバへのアタック
  - ルータへのアタック

TechStyle
Quality for All Professionals

XSS（クロスサイトスクリプティング）対策

- 被害
  - JavaScriptの実行に至り、Cookieデータ漏洩やサイトの不正表示・プライバシ漏洩が可能
    - Not FoundやSQL Errorなどのエラーメッセージの中に表示させることも
  - アプリ動作とHTMLを知り尽くしたexploit

- 対策

→ Cookieをdisableするユーザは増えていることを意識したサイト作り
Cookieなしで動作するよう極力努力すべき

TechStyle
Quality for All Professionals

Cookie偽証対策

- DNSクラックと連動したCookieのドメインスコープ悪用
- RFC2965（7.2 Cookie Spoofing）より超訳

1. victim.cracker.eduをアクセスした際に、デフォルトドメインvictim.cracker.eduで、クッキー session_id="1234" をセットしたとします。

2. 同じブラウザで spoof.cracker.edu をアクセスしてしまい、session-id="1111" を、Domain=".cracker.edu" と共にセットされたとします。

3. もう一度 victim.cracker.edu にアクセスすると、このプログラムがクッキーを読み出すと、下記のものが取り出せます。

```
Cookie: $Version="1"; session_id="1234",
        $Version="1"; session_id="1111"; $Domain=".cracker.edu"
```

victim.cracker.edu サーバは、二番目のものを自分のサイトのクッキーではないことを判断し、無視できるプログラムであるべきです。

- 対策：Cookieのより厳密なドメインスコープ管理をすべき

TechStyle
Quality for All Professionals

「入力」を調べる - input validation

- すべての「入力」をフィルタリング
  - input validation, data cleaning という
  - フォームデータ・URLパラメータ・環境変数を含む
- 入力値チェック・フィルタリング
  - 数字、半角英数、全角
  - 予想外のプログラムエラーを招かないようにする
  - ユーザビリティを不必要に犠牲にしない
- スペシャルキャラクタの深刻な影響
  - メタキャラクタ
  - HTMLタグ記号
  - 全角の“片割れ”が¥ 記号のようなもの

TechStyle
Quality for All Professionals

対策例：フィルタすべき「入力」文字

- %(パーセント)
  - GETメソッドによるパラメータでは、URLエンコーディングにより パラメータ文字が %HH(H:16進数)に変換される。URLの改ざん によりバイナリ化したときに問題を引き起こす文字を混入できる。

    ex.
    %00 (NULL), %0A(改行), %20(スペース), %25(パーセント)

  - URLエンコードされた文字列のvalidationでは、1度のフィルタで クリアしなければならない。

    ex.
    %2500 → %00 → NULL

cf.URLに使用できる文字はRFC2396（Uniform Resource Identifiers (URI):Generic Syntax）で規定されているので、参考にすること。

TechStyle
Quality for All Professionals

対策例：フィルタすべき「入力」文字

- ！(bang)；(semi-colon)：(colon)，(comma)－(minus)
  ￥(backslash)
  - exec, system関数、メールエージェント、シェルプログラムなどに渡される文字列にこれらが含まれると、別の任意のプログラムを動作させたり、プログラムに不必要なパラメータを与えてしまう可能性がある。

```
ex.
   !/bin/bash
   -f /etc/passwd
```

- ＊(asterisk)／(slash).. (dot/dotdot)？(question) []{} (brace)
  - ファイル名に利用される可能性のある場合に注意すべき

```
ex.
  ../../../../
  *.png
```

TechStyle
Quality for All Professionals

対策例:フィルタすべき「入力」文字

- <(lesser than) >(grater than)
  “(double quote) ‘(single quote)
  - タグの初めと終了、オプションアイテムのクローズに用いられるため、タグの埋め込み、クロスサイトスクリプティングexploitのようなスクリプトコードの埋め込みに頻繁に利用される。他サイトのコンテンツの埋め込み、不正なCookie読み出しに悪用される。
    ＜＞はシェルにおけるリダイレクト記号であるためにコンテンツファイル破壊に悪用される可能性がある。

```
ex.
    ><script>alert(window.location);</script>
    “;alert(document.cookie);
    ‘ onmouseover=‘alert(document.cookie);’ ‘
    “><script> alert(document.cookie);</script>
    “></a> <script> alert(document.cookie);</script>
ex.
    ;cat /etc/passwd >>/home/html/htdocs/index.html
```

TechStyle
Quality for All Professionals

対策例:フィルタすべき「入力」文字

- HTMLタグ
  - 必要に応じて許可せざるを得ないケース
    - 一般に、<p>,<bold>,<i>,,,<pre>,
は無害とされるが、せめてWell-formedにはしておきたいところ。
    - exploitがないわけではない
      - Javascriptは思わぬところでも動作する。

ex.
…
<img src="javascript:[code]>
<style type="text/javascript">[code]</style>

TechStyle
Quality for All Professionals

対策：データクリーニング関数例（PHP）

- ereg_replace(“[^0-9]”,””,$data)
  - 正規表現はクリーニングの基本
- addslashes($data)
  - addcslashes (string str, string charlist)
  - クオート（single,double），バックスラッシュ、NULL文字などをslashing(スラッシュ付加)する
- quotemeta($data)
  - .+?[^](*)$ などをquoteする
- escapeshellcmd($data)
  - メタ文字をescapeする
- nl2br ($str)
  - すべての改行文字の 前に ‘
’ を挿入して返す
- htmlspecialchars ("Test", ENT_QUOTES);
  - 特殊文字をHTMLエンティティに変換する

TechStyle
Quality for All Professionals

暗号・乱数に関する補足事項

- 暗号化は完全性を保証しない
  - ハッシュ、XORは見慣れればわかる
  - 暗号関数は自作すべきでない
  - 複数の方策を重ねて使う（タイムアウトなど）
- rand関数出力が類推できる問題
  - 乱数関数出力は「予測」できる
  - 乱数サーバの利用
  - 乱数関数の作成
  - 重層化
- 不可逆データで保護
  - MD5 / ハッシュ関数

TechStyle
Quality for All Professionals

権限に関するルール

- WEBサーバ
  - apache ユーザ
- DBサーバ
  - mysql / postgresql
  - DBユーザ（削除・更新権限）
- ファイルシステム
  - ユーザの読める範囲・書き込める範囲・書き込む内容
- 管理者権限
  - 運用管理者とアプリケーションへの影響
- メールユーザ
  - ログインアカウントの発行の必要性の検討
- アプリケーションユーザ
  - アプリケーション実装のユーザIDによるACL
- IPアドレス制限

TechStyle
Quality for All Professionals

データベースに関するルール

- 「保存」内容により必要性を検討
  - ID / Password
  - 不可逆なhashだと漏れても意味不明
  - 管理担当からでさえ守れる
  - hashデータでその他のプライバシーデータを暗号化
- 保存する場所・テーブル
  - 一時ファイル
    - 共通・IDごと
  - 保存ファイル
    - WEBサーバからinvisibleな場所であるべき
    - 保存ファイルの「脅威」を最低限にできるパーミッション
    - バックアップ方策
- 自動保存されるファイルのクリーニング
  - /tmp/
- 「削除」
  - 「削除」はWEBアプリではpayしない。「無効化」をひたすら用いる。
    - SQLのDELETE文、unlink関数は使用不可とする、など。

TechStyle
Quality for All Professionals

DoS攻撃と過負荷対策のルール

- DoS攻撃
  - 各層により対策は異なる

    例:
    - ネットワーク層の場合
    - アプリケーション層の場合

- CPU過負荷
  - WEBアプリケーションの過剰ロード回避
    - 不正データ時にはできるだけ早期に処理を終了させる
    - 入力処理頻度を規定する(例:slashdot/ECサイトのカゴ)
    - 高負荷を与えるクライアントサイトを記録する(NATに注意)

TechStyle
Quality for All Professionals

3. Webセキュリティ保持の仕組みづくり

- 基本原則
- テスト
- 人的問題
- 知識の共有

TechStyle
Quality for All Professionals

セキュリティ保持の仕組みづくり

- セキュリティ基本原則の周知
- 構築
  - 教訓を生かした構築
  - 目的のための機能
  - 品質・性能のテスト
- 運用
  - コンテンツマネージメントの仕組み
  - 安全を確保できる人的体制
- 監査
  - 制作者による監査
  - 第三者による監査
- 改善
  - ユーザ不在の改善をしない
  - 脅威に関する新たな知識による改善

TechStyle
Quality for All Professionals

品質と性能のためのテスト

- ホワイトボックステスト ― 品質
  - スクリプト監査
  - データフロー監査
    - 制作・開発した人間のテストは「テスト」ではなく「制作」
      - 「テストしましたか？」「はい」 Doubt!

- ブラックボックステスト ― 性能
  - 入力テスト（正常系・異常系）
  - ストレステスト
    - 同一セグメントでは問題が見えないケースが多い
    - abコマンドでできることも多い
    - ログアナライザで外部からどういうパターンで見られているか調査

TechStyle
Quality for All Professionals

コンテンツマネージメントシステム

- できるだけ静的コンテンツを用いる機構のメリット
  - サーバ・ネットワークリソースの有効化
  - バックアップ、スタンバイリカバリの平易化
  - 管理画面、ツール化による、体制の簡素化
  - WEBサイト機能の明確化
  - 構築体制と運用体制の権限分散
  - WEBサイトの活性化

TechStyle
Quality for All Professionals

人的体制の最大のリスク

- ソーシャルエンジニアリング
  - 構築段階
  - 運用段階
  - ワイヤレスネットワーク普及による別の脅威
- 人間というリソースのセキュリティ
  - 最大のセキュリティホール
  - もっとも冗長性が低い
  - もっとも可用性も低い

→ 瞬間最大風速で最適な運用をしていかなければならない
→ 複雑すぎる手順を、もう少し間便で確実な手順へと改訂

TechStyle
Quality for All Professionals

新たな脅威に関する情報源の確保

- 特性のある情報源
  - 新聞・WEBニュース
  - ML
  - Blog
  - ディストリビューション情報
- ユーザに聞け！
  - ログアナライズ（エラーログ）
  - フィードバック
  - アクセス傾向
  - ネットワークに新しいユーザのX-test
- 監査・調査による発見
  - 「制作者本人の監査は、依然、制作であって監査ではない。」

TechStyle
Quality for All Professionals

参考資料

- Saltzer, J.H., and M.D. Schroeder, "The Protection of Information in Computer Systems," Proc. IEEE, Vol. 63, No. 9, Sept. 1975, pp. 1278-1308.
- Wall, Larry and Schwartz, Randal L. "Programming Perl" : Sebastopol, California : O'Reilly And Associates, 1992.
- Al-Herbish , Thamer, Secure UNIX Programming FAQ,1999
- Wheeler, David A. Secure Programming for Linux and Unix HOWTO, 2002
- Howard, M. and LeBlanc, David, WRITING SECURE CODE, 2002
- RFC 2396, 2965
- Lim, John, Tuning Apache and PHP for Speed on Unix,2001(http://php.weblogs.com/tuning_apache_unix)
- 「情報アーキテクチャ入門 - ウェブサイトとイントラネットの情報整理術」オライリー・ジャパン
- 「大切なのはトラブル・ゼロよりトラブル後」-日経ITPro
  http://itpro.nikkeibp.co.jp/free/ITPro/OPINION/20031104/2/

ほか

Thank You

Please feel free to mail to me

riotaro@techstyle.jp

http://okdt.org/blog/