ディレクター・製作者が知っておくべきセキュリティ

～脆弱性の事例から～

株式会社ビジネス・アーキテクツ
太田 良典

Business Architects Inc.

自己紹介とお約束

Business Architects Inc.

私の所属

Business Architects Inc.

- 株式会社ビジネス・アーキテクツ (bA)
- 大規模企業サイトの構築を得意とする
  - 戦略から設計、実装までお付き合いします
  - 「ただ言われたとおりページを作るだけ」ではない

仕事でやっていること

- マークアップデザインエンジニア
  - HTML や CSS の設計、ガイドライン
    - 弊社内では変な HTML を書くと私に怒られるらしい (都市伝説)
  - Webアクセシビリティ
    - 「アクセシビリティ指針」策定
  - ユーザビリティ
    - サイト設計 / UI 設計 /情報設計諸々

こっそりやっていること

- 脆弱性の発見と届出
  - 経済産業省の「情報セキュリティ早期警戒パートナーシップ」の制度を利用していろいろ届出

届出の内容

- 主にクロスサイトスクリプティング脆弱性(XSS)
- SQLインジェクション、CSRFその他も少々
- 高度なものはあんまりない

届出の例1

- それなりに報道されたりしたケースも

届出の例2

- 複数のメールクライアントにおける mailto: URL 解釈の問題
  - mailto:にBcc:などが含まれていると……
  - 脆弱性ではないとされつつも、修正された

届出の例3

- PukiWiki他、多数のWikiエンジンに脆弱性
  - スクリプトを含むHTMLファイルを添付するだけ

届出件数

- 届出件数：たぶん120～130件くらい
  - 国内の届出は821件なので、シェア約15%
  - ウェブアプリケーションの届出に絞ると全体で564件なので、シェア20%超
- 第2回IPA賞 (情報セキュリティ部門) 受賞
  - 理由：届出件数最多のため
  - 受賞したりしている時点でもう「こっそり」ではなくなっていますが……。

本題の前に、お約束

- 情報を不正アクセスから守るためには、攻撃側の手法を知る必要があります
- そのため、この話の中にも攻撃手法の解説が出てきますが、絶対に悪用しないでください
- 不正アクセス行為は、法によって処罰されることがあります

本当にあったお話 その1

「デザインリニューアル」
のはずが……

Business Architects Inc.

ありがちなパターン

- 既存の会員制サイトのデザインリニューアル
- 情報設計、UI設計、ビジュアルデザインなどを弊社が担当
  - HTML、クライアント側のスクリプト実装は弊社が担当
- バックエンドのシステムについては、旧システムを実装した会社が引き続き担当

作業は問題なく進み……

- HTML4.01 + CSS2 で実装
  - いわゆる「フルCSS」
    - 最近は Netscape4 でのレイアウト再現が求められなくなり、tableレイアウトが必要なくなっている
- 弊社は HTML とスクリプトを納品
- システム側の実装が完了

テスト開始

- 弊社は基本的にデザイン担当なので、期待される役割は……
  - デザインが正しく反映されているか、崩れたりしていないか確認すること
  - クライアント側スクリプトの動作を確認すること

ありがちな不具合

- 表示崩れ、不自然なマージンなどが多数
- 良く見るとソースが勝手に改変されている
  - 謎のスペーサGIF
  - font要素
  - center要素 (懐かしい……)

HTMLの知識が9年前のまま

ありがちな不具合 (つづき)

- リンクがことごとく javascript: スキームになっている
  - …… など
- スクリプト無効時動かない、別ウィンドウが開けないなどさまざまな問題がある
- WCAG1.0 6.3 で名指しで禁止されているが……

WCAG読んでない

まあ、ここまでは良い

何故か……

- 大量の脆弱性が発見される
  - クロスサイトスクリプティング
  - ディレクトリトラバーサル
  - OSコマンドインジェクション
  - SQLインジェクション
  - セッション管理の不備

  etc.
- 中には洒落にならないものも……

報告してみると

- お返事：「システムの改修は想定していない」
  - 脆弱性がこんなに発見されるとは予想していない
  - 見積にもスケジュールにも盛り込まれていない
- というわけで、そのままリリース
  - しかも、良く考えるとリニューアル前のシステムも脆弱だったということであり……。
- 結局、ずいぶん経ってからひっそりと修正

私の経験からすると……

- HTML が正しく書けていないサイトはけっこう危ない感じがする
- 実装者の HTML の知識と脆弱性の多さは反比例しそうな気がする
  - そもそも、XSS の原因のほとんどは HTML を知らないことに起因する

教訓

システム会社にもいろいろありますが

「HTMLは正しく」

これができていないと

ちょっと危険かも

本当にあったお話 その2

HTMLの高度な知識が
求められたケース

Business Architects Inc.

概要

- とあるISPのサイトで発見した
  「クロスサイトスクリプティング脆弱性」事例
- 経緯
  - 2005年3月1日発見
  - 同日 IPA セキュリティセンターに届出
  - 2005年3月7日、追加情報を届出
  - 2005年10月26日、修正完了の連絡

そのサイトの仕様

- タグが書ける掲示板
  - 書けるタグはごくわずか
  - <script>……</script> などは書けない
    (書くと、<script> などとして出力される)
  - は書ける
- おそらく想定されていたであろう仕様:
  - タグは書けるが、スクリプトは動かない

いろいろ書いてみる

test

- タグ部分が削除され、test と表示される
  - リンクにもならない
- javascript: スキームの URL は無効になり、スクリプトは実行されない

もっといろいろ書いてみる

- やっぱりタグが削除される
  - URL のどこかに “script” が含まれるとダメ
- 不具合では?
  - 上記は正しい URL で、書けても問題ないはず
  - JavaScript の解説をしているようなサイトにリンクできないのですが……

あまり深く考えていない?

もっともっといろいろ書いてみる

- これは書けた
- t = t
  - しかし“script” という文字列はどこにもないため、フィルタ処理を貫通
- アンカー部分をクリックすると、スクリプトが動作

脆弱!

脆弱性は何故生まれたか?

- javascript: がまずい、というところまでは良かった
- 数値文字参照に考えが至らなかった

引用符で括られた属性値の中では
数値文字参照が展開される、
という HTML の知識が必要だった

届け出ました

- 比較的素早く対応された
  - “script” と書いてあっても、タグが削除されるようになった
  - 数値文字参照を展開してからフィルタ処理するようになった模様

これで一安心

だと良いのですが

さらにいろいろ書いてみる

- t ではなく &#116 で、末尾のセミコロンが省略されている
- これは書けた
  - そしてスクリプトが動いた

やっぱり脆弱

脆弱性は何故生まれたか?

- 数値文字参照を考慮したのは良かった
- しかし、セミコロンが省略された場合に考えが至らなかった

数値文字参照の末尾のセミコロンは
省略できることがある、
という HTML の知識が必要だった

さらにマニアックな例

- <input<script src="……"></script>
  - これは実は「正しい」記述
    - HTML では SHORTTAG YES なので短縮タグ機構が有効、タグのまとめ閉じが可能 (XHTMLでは不可)
  - そのため、最近のわりとちゃんとしたブラウザではちゃんと理解されてしまうことがある

SGMLの知識も必要

心の叫び

普通
そんなの
知らないよ!

脆弱性対応の難しさ

- 「安全なタグだけが使用できる」というウェブアプリケーションを作るためには、HTML の知識が必要
- それも聞きかじりの知識ではなく、HTML の仕様を正しく理解している必要がある
- しかも、時にはかなりマニアックな知識が求められることがある
  - 今回は紹介していませんが、ブラウザのバグの話もあり……。

教訓その1

HTMLに詳しい者だけが指摘できる脆弱性もある

教訓その2

- 「タグが使用できる」が、「危険なタグは使用できない」アプリケーションを実装するのは非常に難しい
  - 有名な Web メールサービスは脆弱性の発覚と修正を何度も何度も繰り返して現在に至る
  - そして、最近でも指摘されている
- そういうシステムを安易に提案しない
  - もしくは覚悟して提案する:-)

ほんとうにあったお話 その3

設計に起因する
HTTPSの不適切な利用

Business Architects Inc.

「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ために安全性が確保されないケース
- 「脆弱性なの?」と疑問に思うケースもあるが、ちゃんと取り扱ってもらえる
  - おそらく、「運営者が HTTPS を使用している意図が明白」かつ「HTTPS が安全に機能していない」という条件で脆弱性と判断される

概要

- とある協会のサイトで発見した「HTTPS の不適切な利用」事例
- 経緯
  - 2005年3月10日 そのサイトで XSS を発見･届出
  - 2005年03月24日 残念なことが起きる
  - 2005年03月25日 別の問題を発見したので届出
    - これがHTTPSの不適切な利用として受理される
    - 修正完了の連絡はないが、修正されている模様

セキュリティポリシー

- 「セキュリティポリシー」に以下のような記述
  - 当サイトでは SSL を使用しています。
  - 以下の方法で安全性を確認することができます。
    - ブラウザのアドレスバーを確認する
    - ブラウザのステータスバーの鍵アイコンを確認する
- そして、そのサイトではフレームが使われていた

例：とあるHTTPSなフォーム

とあることをすると……

フレームの差し替え

- フレームで構成されたサイトでは、フレームの中身を他のものに差し替えることができる
- 基本的にそういう仕様
  - しかし、ブラウザ側も対応してきていて、最近の Firefox ではできなくなっている。IE7 も対応予定

フレーム + HTTPS はダメ

- フレームの中身だけ偽物に差し替えられているかもしれない
- アドレスバーと鍵マークの確認では、外側のフレームについてしか確認することができない
  - 中身も確認するには、フレームの中で右クリックしてプロパティを出す必要がある

その他ありがちな事例

わりとよくある
「HTTPSの不適切な利用」例

「HTTPSの不適切な利用」例

1. 嘘のSSL宣言
2. 確認できないSSL宣言
3. オレオレ証明書
4. 問題ある指示
5. 信頼できないフォーム
6. 信頼できないドメイン

知っておいてほしいこと 1

- なぜ HTTPS (SSL/TLS) を使うのか?
- 答え : 安全を保証するため
  - インターネットでは、任意の通信経路を通って通信が行われる
  - 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
  - それがまずい状況の場合は、通信系路上に悪い人がいても問題ないように暗号化する
- 安全を「保証」することが重要なので、「安全かも」では HTTPS の意味がない

1. 嘘のSSL宣言

- 「SSLを使っています」と言いながら、実際には使っていない
  - セキュリティポリシーに「SSLを使って情報を保護しています」という主旨がはっきり書いてあった
  - しかしフォームの URL も送信先の URL も http: で、特に暗号化されていなかった
- 実は、SSL / TLS が使える用意はされていた
  - https: でフォームにアクセスすることは出来た
  - しかし、フォームへのリンクやフォームからの送信先の URL が http: で記述されていた

知っておいてほしいこと 2

- サイトに「SSLを使っています」と書いてあるだけでは、暗号化されたりはしない
- SSL/TLS を使えるようにしていたとしても、ユーザが https: な URL に誘導されないと意味がない
  - これはありがちなミス
  - http: でアクセスされても何も出ないようにしておくと安心
    - ただし、SSL 非対応のブラウザではアクセスできなくなる (特に携帯電話)

2. 確認できないSSL宣言

- 「SSLを使っています」と言っている
- 実際に使ってもいる
- しかしながら、ユーザにそれを確認する術がないため、偽者だとしても区別できない

とあるサイトの説明

- アドレスバーや鍵マークは表示されないが SSL は有効、と主張
- 見た目がそっくりな偽サイトを作られたとき、ユーザはどうやって見分けることができるのか?

知っておいてほしいこと 3

- 悪い人は本物そっくりな偽のフォームを作ってだまそうとするかもしれない。
- そのとき、それを偽物だと識別できるようになっていなければならない
- 通常のブラウザでは、フォームが
  - 既知の信頼できるドメインに属しており
  - HTTPSで保護され、有効なサーバ証明書があるということをアドレスバーとステータスバーで確認でき、これによって本物だと判断できる

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

利害一致、まさに思うツボ

3. オレオレ証明書

- 以下のような証明書は信用できない
  - 信頼された認証機関による署名がない
  - 発効先の名前と違うドメインで運用されている
  - 証明書の期限切れ
- ブラウザは警告を出すのだが……

とあるサイトの主張

- 警告は出るが保護される
- 「はい」を押せばOK

ってホンマかいな?

知っておいてほしいこと４

- 証明書の警告が出るのは何故?
- 答え：通信相手が偽者の可能性があるから
  - 暗号化されていれば第三者に読み取られても問題ないが、暗号化通信の相手が偽者だったら全く意味がない!
  - だから相手が偽者ではないかどうか確認する必要がある。そのための証明書
  - 相手が偽者ならサーバ証明書も偽造しているはずなので、信頼できる機関による電子署名がない証明書は信用してはならない

4. 問題ある指示

- HTTPS はちゃんと機能していて問題ない
- しかし、ユーザに対して問題のある指示が行われている
- 指示に従うとセキュリティレベルを下げられてしまったり、危険に晒されてしまったりする

とあるサイトの指示

SSL2.0が必要?

メニューから<ツール><インターネットオプション>を選択し、「詳細設定」のセキュリティ項目にある「SSL2.0を使用する」「SSL3.0を使用する」にチェックを入れてください。

このサイトでSSL3.0が使えるなら
SSL2.0は必要ないはず

SSL2.0は脆弱なのでオフ推奨

「はい」を押せ?

セキュリティの警告が表示された時に「はい」ボタンを押して、セキュリティ証明書を受け入れてからご利用いただきますようお願い致します。

悪い人がいた場合、この操作で偽の証明書を信頼してしまう

知っておいてほしいこと５

- セキュリティレベルを下げる指示はしないこと
  - どうしても必要な場合のみ、リスクを理解させたうえで指示するべき
  - そもそも理解してるの?
- ブラウザの警告を無視させないこと
  - 指示に従ったら偽の証明書をインストールさせられてしまうケース多々あり
  - 特に、ルート証明書のインストールは慎重にやらないと非常に危険

5.信頼できないフォーム

- 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」と主張されるが……。

とあるサイトの説明

知っておいてほしいこと 6

- 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

6.信用できないドメイン

- フォームが別ドメインにある
- そして、その別ドメインにあるフォームには、こんなメッセージが……

対抗して

- 悪い人は当然、フォームをこう書き換えて置くことができる

見分けがつかない

知っておいてほしいこと 7

- 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
  – 当然、悪い人も同じメッセージを書ける
- リンク元に書いてあればまだ分かるが……

おしまい

ありがとうございました

http://bakera.jp/websig247/

Business Architects Inc.