

Webアプリケーションに対する 攻撃と防御

ネットエージェント株式会社
愛甲健二

自己紹介



- 愛甲健二 <http://d.hatena.ne.jp/kenjaiko/>
- 現在、ネットエージェント株式会社にて、取締役兼 セキュリティエンジニアをやっています
- 主にリバースエンジニアリングが中心ですが、OSカーネル関連の開発にも関わっており、若干プログラマよりのセキュリティ技術者です
- 出身は熊本です

目次

- XSS(クロスサイトスクリプティング)
- Webセッションハイジャック
- ディレクトリトラバーサル
- SQLインジェクション
- まとめ

目次

- XSS(クロスサイトスクリプティング)
- Webセッションハイジャック
- ディレクトリトラバーサル
- SQLインジェクション
- まとめ

XSS (Cross Site Scripting) とは

クロスサイトスクリプティング (Cross Site Scripting) とは、動的に Web ページを生成するアプリケーションのセキュリティ上の不備を意図的に利用し、狭義にはサイト間を横断して悪意のあるスクリプトを混入させること。また、それを許す脆弱性のこと。広義にはスクリプトを混入させずとも、任意の要素を混入させられうる脆弱性を含む。
(Wikipediaより)

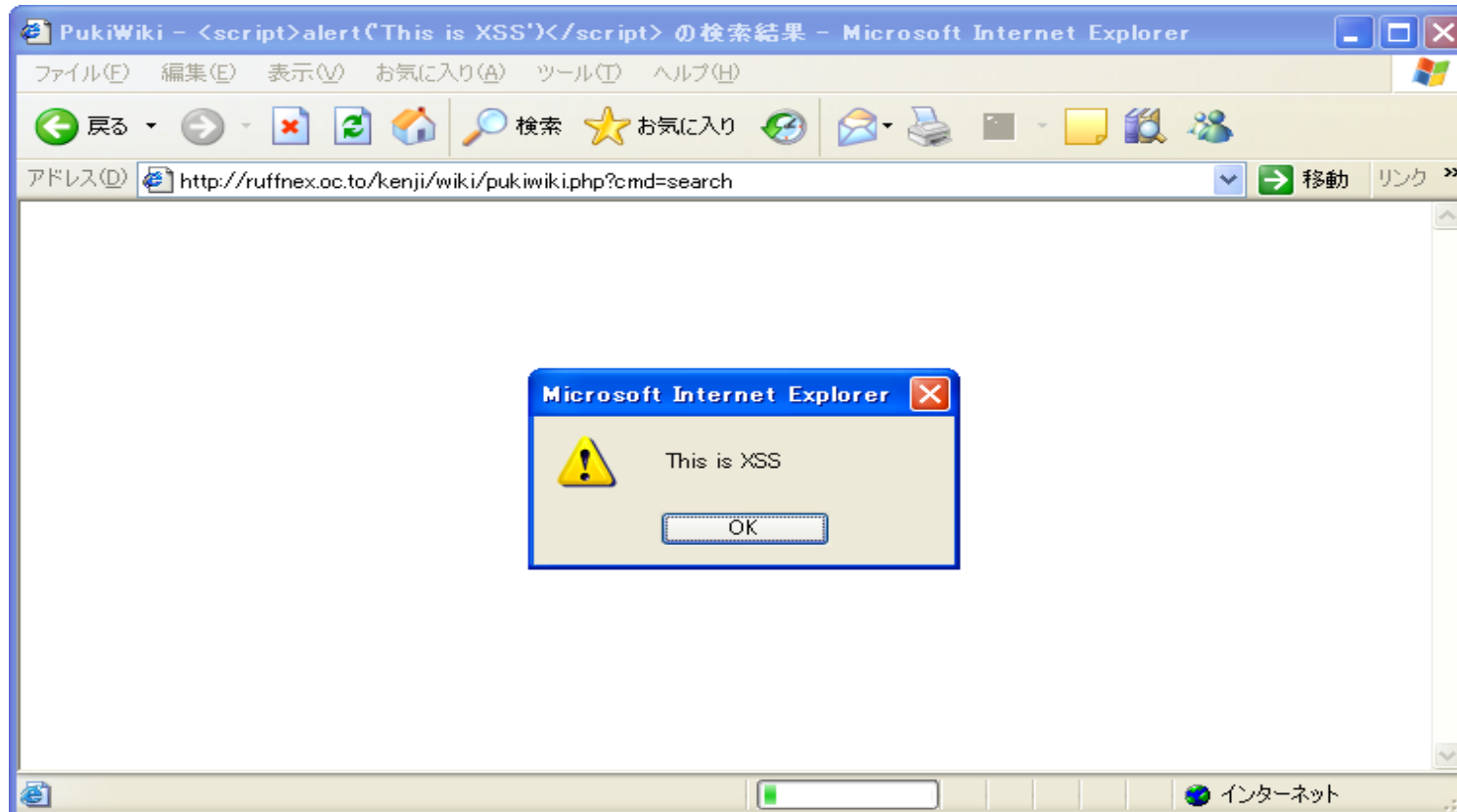
XSS (Cross Site Scripting) とは

クロスサイトスクリプティング (Cross Site Scripting) とは、動的に Web ページを生成するアプリケーションのセキュリティ上の不備を意図的に利用し、狭義にはサイト間を横断して悪意のあるスクリプトを混入させること。また、それを許す脆弱性のこと。広義にはスクリプトを混入させずとも、任意の要素を混入させられうる脆弱性を含む。
(Wikipediaより)

誤解を恐れずに言うと、

他人の Web ページでスクリプトが実行できるってこと？

XSS脆弱性の例



DEMO

<http://ruffnex.oc.to/kenji/wiki/pukiwiki.php>

スクリプトを実行するとは？

- HTMLタグやjavascriptコードをブラウザに認識(実行)させる
- ただ実行させるだけじゃなく、他のユーザが開くブラウザ内で任意のHTMLタグやjavascriptコードを実行させる
- XSSのみでは、それほど致命的な被害はないが、セッションハイジャックなどの起点になりうる
- では、その対策方法は？

特殊な文字を置換する

- タグを意味する<や>といった文字をHTML用に変換
 - < を < へ変換
 - > を > へ変換
 -
- さらに、& や " や ' もHTML用文字へ変換
 - & を & へ変換
 - " を " へ変換
 - ' を ' へ変換

htmlspecialchars関数 by PHP

- HTMLに関連する特殊な意味を持つ文字を適切に変換する関数であり、PHPに実装されている
 - `&` を `&` アンパサンドを変換
 - `<` を `<` 小なりを変換
 - `>` を `>` 大なりを変換
 - `"` を `"` ダブルクォートを変換
 - `'` を `'` シングルクォートを変換
- (ENT_QUOTES)

HTMLとして出力する場合やformから受け取ったデータに、htmlspecialchars関数を通すことで、XSSを防ぐコードが書ける

特殊なXSS

- 最上位bit無視によるXSS (us-ascii)
- 文字エンコーディングの差異によるXSS
- MIMEタイプ無視によるXSS

ブラウザの実装に依存する脆弱性であるが、可能な限りWebアプリケーション側で対処する必要がある

目次

- XSS(クロスサイトスクリプティング)
- Webセッションハイジャック
- ディレクトリトラバーサル
- SQLインジェクション
- まとめ

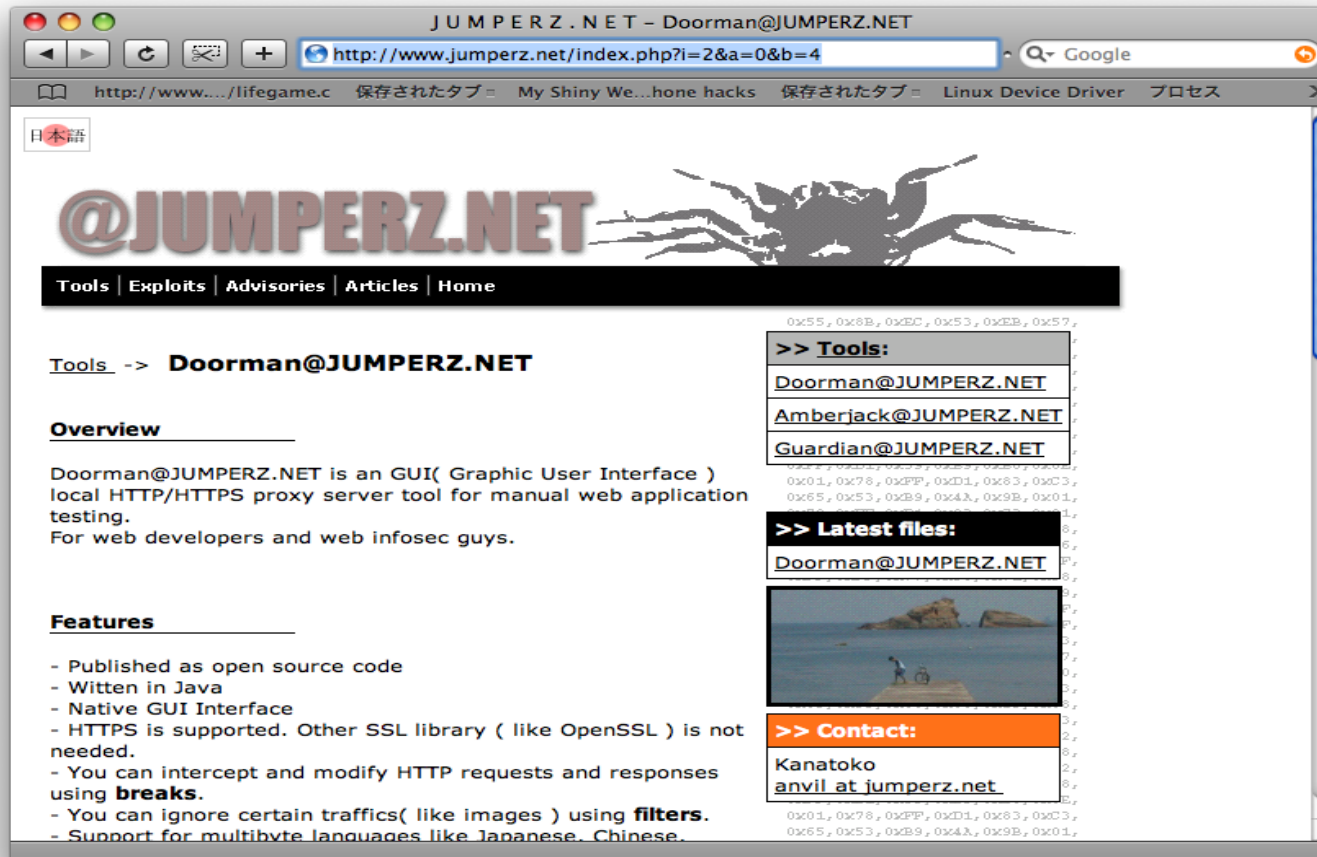
Webセッションハイジャック

- HTTPは、基本的に「リクエスト」、「レスポンス」、「ブラウザに反映」という処理単位で行われるため、連続的な時間での「状態」を管理できない
- よって、WEBアプリケーション側で各セッションを管理するための識別情報(セッションID)を作り、**cookie**にセットしてクライアント(ブラウザ)との状態管理を行う

つまり、XSSによって**管理者のcookie情報を盗めば**、パスワードを知らずとも、管理者に成り代わることができる

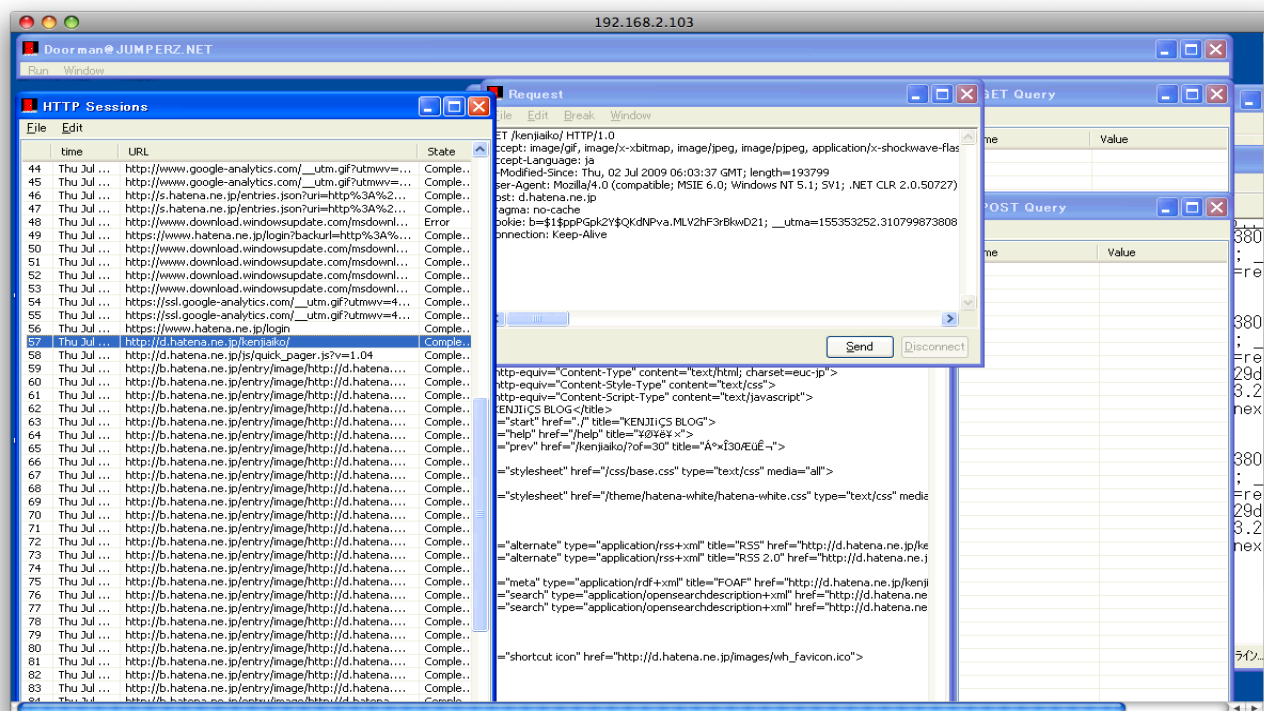
HTTP通信キャプチャツール

- doorman by jumperz net
 - <http://www.jumperz.net/index.php?i=2&a=0&b=4>



セッション管理状況を確認

- はてなブログのセッション管理用cookie



DEMO

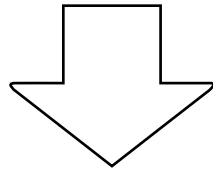
<http://d.hatena.ne.jp/kenjiaiko/>

管理者のcookieを盗めたら

- Web2.0と言われ始めて、ほとんどのサービスがブラウザのみで完結するようになった
- はてなやmixiもブラウザからログインするシステム
- しかし、サーバが、通信相手を「管理者」と判断できる理由は、パスワード入力によるログイン後、**cookieでセッションIDを管理している**から
- つまり、管理者ログイン時のcookieを第三者が盗めたら、パスワードを知らずとも管理者に成りすますことができる

cookieが盗まれる手口と対策

- XSSなどのWebアプリケーションの脆弱性を利用
- セッションIDの生成アルゴリズムを解析、推測される



- セッションIDの生成アルゴリズムは、推測されない高度なものを使う
- 他の脆弱性と連携して致命的な効果になることを意識してWebアプリ開発を行う

目次

- XSS(クロスサイトスクリプティング)
- Webセッションハイジャック
- ディレクトリトラバーサル
- SQLインジェクション
- まとめ

どこに問題がある？

```
<?php
    $fh = @fopen($_SERVER['QUERY_STRING'], "r") or exit;
    $data = @fread($fh, 256);
    fclose($fh);
    print $data;
?>
```

どこに問題がある？

```
<?php
    $fh = @fopen($_SERVER['QUERY_STRING'], "r") or exit;
    $data = @fread($fh, 256);
    fclose($fh);
    print $data;
?>
```

- QUERY_STRINGに **index.php** を入れてみたら？

どこに問題がある？

```
<?php
    $fh = @fopen($_SERVER['QUERY_STRING'], "r") or exit;
    $data = @fread($fh, 256);
    fclose($fh);
    print $data;
?>
```

- QUERY_STRINGに **index.php** を入れてみたら？
- QUERY_STRINGに **../../etc/passwd** を入れてみたら？

どこに問題がある？

```
<?php
    $fh = @fopen($_SERVER['QUERY_STRING'], "r") or exit;
    $data = @fread($fh, 256);
    fclose($fh);
    print $data;
?>
```

- QUERY_STRINGに **index.php** を入れてみたら？
- QUERY_STRINGに **../../../../etc/passwd** を入れてみたら？
- QUERY_STRINGに **ls |** を入れてみたら？

ひとつの脆弱性で様々なことが可能

- index.phpのソースコードが閲覧できる
- サーバ内の/etc/passwdの中が見える
 - ディレクトリトラバーサル
- 任意のOSコマンドが実行できる
 - コマンドインジェクション

ディレクトリトラバーサル の例

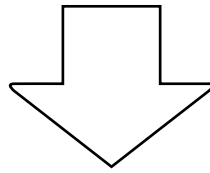


DEMO

<http://192.168.2.102/?hello.txt>

XSSと違い、ダメージが致命的

- ディレクトリトラバーサルや、コマンドインジェクション
- は、サーバへの直接的な攻撃となり得るため、致命的な
- ダメージとなりやすい



- ユーザーからの入力データはすべてフィルタ(置換)する
 - スラッシュ **'/'** を置換するなど
- なるべくホワイトリスト方式で実装する

目次

- XSS(クロスサイトスクリプティング)
- Webセッションハイジャック
- ディレクトリトラバーサル
- SQLインジェクション
- まとめ

SQLインジェクションとは

SQLインジェクション(SQL Injection)とは、アプリケーションのセキュリティ上の不備を意図的に利用し、アプリケーションが想定しないSQL文を実行させることにより、データベースシステムを不正に操作する攻撃方法のこと。また、その攻撃を可能とする脆弱性のこと。
(Wikipediaより)

SQLインジェクションとは

SQLインジェクション(SQL Injection)とは、アプリケーションのセキュリティ上の不備を意図的に利用し、アプリケーションが想定しないSQL文を実行させることにより、データベースシステムを不正に操作する攻撃方法のこと。また、その攻撃を可能とする脆弱性のこと。
(Wikipediaより)

誤解を恐れずに言うと、

ブラウザではなくデータベースに対するXSSということ？

認証の回避

例えば、かの有名なSNSサイトであるmixiは、のべ1000万を超えるユーザーアカウントを管理しているが、大量のデータを一括して管理するシステムとして、DBはもはや必需品

mixiにログインする際、メールアドレスとパスワードが必要だが、ユーザーがこれらを入力し、サーバへ送信するとサーバ側では、以下のようなSQL文が実行される(おそらく)

```
SELECT * FROM user WHERE uid='$mailaddress' AND pwd='$password'
```

登録していれば、メールアドレスとパスワードが一致するユーザーは必ずひとつはあるため、ヒットしたらそのユーザーでログインさせる

認証の回避2

例えば、メールアドレスを **kenji@test.com** とし、パスワードを **testcode** とすると

```
SELECT * FROM user WHERE uid='kenji@test.com' AND pwd='testcode'
```

ここで、仮にメールアドレスを '**or 1=1 --**' と入力し、パスワードを未入力とすると

```
SELECT * FROM user WHERE uid="' or 1=1 -- ' AND pwd=""
```

--は以降をコメントアウトするという意味なので結果的に

```
SELECT * FROM user WHERE uid="" or 1=1 --
```

となり、必ず認証が成功(全件ヒットします)

DBが不正に操作される

- Webアプリケーションのチェックをすり抜けてブラウザが実行できるコードを挿入するのがXSS
- Webアプリケーションのチェックをすり抜けてSQL文を作り実行するのがSQLインジェクション
- 攻撃者にDBの操作を可能にさせるため、XSSよりも致命的なダメージを受ける場合が多い(情報漏洩など)
- SQLインジェクションは、技術的に奥が深く、SQL文の知識が不可欠

目次

- XSS(クロスサイトスクリプティング)
- Webセッションハイジャック
- ディレクトリトラバーサル
- SQLインジェクション
- まとめ

まとめ

- gmailやgoogle docsなど、コンピュータは今後ますますWebアプリケーションが中心となっていく
- また、Webアプリケーションセキュリティは、ブラウザの問題であったり、Webアプリの問題であったりと、様々な技術が組み合わさっているため、根本的な解決も難しい
- しかし、今後のコンピュータネットワークの進化の方向性として、おそらくWebベースに変わっていくと思われるため、知識として覚えていても損はないはず

ご清聴、有難うございました
m(_ _)m

Any Questions?