Overcoming your web insecurity

Kirk Jackson
Xero
www.xero.com
pageofwords.com
kirkj@paradise.net.nz
DON’T TRUST USERS!
This talk

- Introduce some common threats
  - Don’t trust user input
  - Phishing
  - **XSS** – Cross-site scripting
  - CDRF
  - Click-jacking

- Quick wins

- Not a comprehensive treatment
Demo

- Classifieds Web Site Starter Kit
- Rebranded:
  
  ![Covertly Classified Logo]

- A couple of small tweaks for demo purposes 😊
The latest activity relating to your ads and offers will appear on the My Ads & Profile page. Go to My Ads & Profile

Placing a new ad is fast and easy. Place a New Ad

Browse Categories

Antiques & Collectibles (1)       Garden (0)
Arts & Crafts (1)                Home (0)
Auto (0)                         Music (0)
Electronics (0)
Step 1: Select Category

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Step 2: Specify Ad

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Step 3: Add Photos

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Post an Ad: Details

Your Selected Category:
All Categories > Antiques & Collectibles | Change

Type:
- For Sale
- Wanted

Title: (50 characters max)

Description: (500 characters max)

URL: (optional)

Price: $
Go to...

- Current Ads
- Inactive Ads
- Saved Bookmarks
- My Profile

Search

---

My Current Ads

<table>
<thead>
<tr>
<th>Title</th>
<th>Posted</th>
<th>Expires</th>
<th># Views</th>
<th># Resp.</th>
<th>Category</th>
<th>Edit Photos</th>
<th>Unlist</th>
</tr>
</thead>
<tbody>
<tr>
<td>test 2</td>
<td>30/07/08</td>
<td>06 November</td>
<td>1</td>
<td>0</td>
<td>Antiques &amp; Collectibles</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Site Statistics

Last generated on:
31/10/2008 9:30:57 p.m.

Search:

Site Administration

Ads
# Active Ads: 2
# Total Ads: 4

Manage Ads & Features

# Responses for currently active Ads: 0
# Total Responses: 0

Activations & Deletions
# Ads pending activation: 0
# Ads marked as deleted: 0

Manage Activations & Deletions

Members
# Total Members: 2
# New Registrations in last 7 days: 2

Categories:
# Top-Level Categories: 7

Manage Categories
Demo Users

Kirk
Site Admin
Chrome

Marge
Regular User
IE

Bob
Naughty User
Firefox
Threats

- Employees / students / other authenticated users
- Malicious third parties
- Malware
- Phishing
- Anonymous users
Things we shouldn’t trust

- GET’s
- POST’s
- Headers
- Cookies
- Any user content
GET url

It’s very easy to edit a url
- Nothing sensitive
- Nothing destructive
- No updates or deletes (idempotent)

http://google.com/?q=codecamp

GET /search?q=codecamp
GET - DEMO
POST data

- Send data in the body of the request, rather than the url
- Still just as easy to fake

POST /path/foo.aspx HTTP/1.0
User-Agent: EvilHacker/2.0
Content-Type: application/x-www-form-urlencoded

id=7&title=Hello
POST - DEMO
Redirects

- Don’t redirect to untrusted urls
REDIRECTS - DEMO
Cookie data

- Use ASP.NET forms auth for authentication
  - Secure, non-predictable
- Use Session for state
- Only store unimportant stuff in cookies (theme choice vs username)
Session

- Not as bad as you might think!
- Encrypting, transmitting, round-tripping probably worse than server memory usage
- ASP.NET Session ID is hard to guess
Storing and using data

- Authorisation / Role: Server side only
- Navigation data: In the url is fine (check access on server side)
- Presentation flags / theme / lang: fine in cookies
- Hidden fields: preferably change to server-side
- Validate every request
- Application secrets: Never visible to client
SQL Injection

- Use parameterised SQL so that arguments are escaped
Phishing

- A fake version of your website
- Really hard to protect against!
- User education is our best protection
  - Train to look at url bar (yeah, right!)
  - Never send links via email
  - Always address emails with their name
PHISHING – DEMO
Framebusting

- Many attacks wrap your page inside a frameset to trick the user
- Bust out of your frame

```javascript
if (parent.frames.length > 0) {
    parent.location.replace(self.document.location);
}
```
FRAMEBUSTING - DEMO
Encoding

- Encoding is "the process of transforming information from one format into another" [Wikipedia]
- Taking some input text and making it appropriate to use in a given context
- Untrusted input → Safe to output
- User enters: Kirk <script>...
- We output: Kirk &lt;script&gt;....
XSS – Cross site scripting

Displaying untrusted user input
- Often used to steal a cookie, phish, etc.

Input arrives: `<script>alert('Hello!')</script>`

Invalid input!  Encode into DB  Store verbatim

```
&lt;script&gt;alert('Hello!')&lt;/script&gt;
```

```
&lt;script&gt;alert('Hello!')&lt;/script&gt;
```
Display on web page

Encoded in DB

```html
&lt;script
 &gt;alert('H
```

Display directly

```html
&lt;script
 &gt;alert('H
```

Store verbatim

```html
&lt;script>alert('Hello!'
```

Encode on display

```html
&lt;script
 &gt;alert('H
```
Display contexts

What if we want to display it in a non HTML context?

\[
\text{var text = 'XXXX'}
\]

URL context

HTML attribute context

Javascrip context
XSS – Cross site scripting

Displaying untrusted user input

- Sanitise all input
- Encode all output
- HTTP Headers – don’t insert untrusted content
- Some ASP.NET controls don’t encode output
- Use Anti-XSS Library rather than HttpUtility
Defense in Depth

- Validate in, encode out
- White List character sets
- Principle of inclusions
  - a-z, A-Z, 0-9, space, period, comma, underscore, hyphen
  - Latin, Greek, Bengali, Balinese, Japanese
AntiXSS library

- Encode text for a variety of contexts
- AntiXSS module for automatically encoding controls
- Produced by Microsoft ACE Team (Security, Performance and Privacy)
- Recently open-sourced
Steps to use AntiXSS

1. Review existing code
2. Look for untrusted input
3. Determine encoding method
4. Encode output
1. Review existing code

Review the ASP.NET code that generates output

- Response.Write
- ASP.NET / ASP <%= %>
- <%# %>
- Label.Text = “blah”
2. Look for untrusted input

Determine whether the output includes untrusted input parameters

- Application variables
- Cookies
- Databases
- Form fields
- Query String variables
- Session variables

(err on the side of caution and to encode the output anyways)
3. Determine encoding method

Determine the encoding method to use

- **HtmlEncode** - html output, except when an attribute
- **HtmlAttributeEncode** - html attribute
- **JavascriptEncode** - used within javascript, puts inside quotes
- **UrlEncode** - used in a url (e.g. query param)
- ...and **VisualBasicScriptEncode**, **XmlEncode**, **XmlAttributeEncode**
4. Encode output

- Only encode output once
- Encode just before the output is written
Controls to encode

- ASP.NET controls were made in a simpler time
- All Response.Write, databinding etc
SRE - Security Runtime Engine

- Runs over entire page on pre-render
- Looks at all controls, and all fields that need encoding
- Doesn’t double-encode
- Add httpHandler in web.config
- Deploy in bin directory
MarkAntiXssOutput

- Add query string parameter to show data that has been encoded by SRE
XSS – DEMO
XSS – Cross site scripting

Displaying untrusted user input

- Sanitise all input
- Encode all output
- HTTP Headers – don’t insert untrusted content
- Some ASP.NET controls don’t encode output
- Use Anti-XSS Library rather than HttpUtility
CDRF – Cross domain request forgery

Posting a form from one website to another
CDRF – Cross domain request forgery

Posting a form from one website to another

- Cookie checks aren’t enough
- ViewState isn’t enough
- Check referer
- Use a token in the POST data / ViewState to validate
Clickjacking

- E.g. Content hidden behind a flash animation
- Flash, CSS, PDF
- Newly disclosed
Clickjacking demo
Clickjacking

- E.g. Content hidden behind a flash animation
- Flash, CSS, PDF
- Newly disclosed
Take-aways

- Don’t trust anything given to you by the user’s browser
- Be careful where you store sensitive stuff
- XSS, CDRF, Clickjacking
References

- Clickjacking: [http://ha.ckers.org/blog/20081007/clickjacking-details/](http://ha.ckers.org/blog/20081007/clickjacking-details/)
- Framebusting: [http://pageofwords.com/blog/2008/10/06/FrameBustingInJavascript.aspx](http://pageofwords.com/blog/2008/10/06/FrameBustingInJavascript.aspx)
- AntiXSS library
- Security Runtime Engine (coming)
- [http://pageofwords.com](http://pageofwords.com) – my blog