



# Contents

- XSS
- Clickjacking
- CSRF
- SQLi
- Session Fixation
- Improper Session Invalidation
- Cookie Settings
- User Enumeration
- Information Disclosure
  - Error Messages
  - Server Headers

# Testing for XSS

- Place input in all the variables you can
  - Use something sufficiently unique (farm animals)
- See what is reflected back onto the page
- See what characters you can get through
- Figure out what you can do with the space you have and the characters you can get through

# Testing for XSS

- If you can inject into the tag but can't escape it (< and > are unavailable) then try and use:
  - ‘ onmouseover=alert(0) [may need to use “ ”]
  - ‘ onload=alert(0) [may need to use “ ”]
- If you can escape the tag (< and > are available) try and use:
  - “”><script>alert(0)</script>
  - “”><img/src/onerror=alert(0)>

# Injecting into HTML Tag Parameters

- Sometimes user input ends up inside tags like:
  - ``
- In this case, if you want to script to execute you need to break out of the tag so you can start a new one
  - Payload: `"><script>alert(0)</script>`
  - Injected tag:
    - `<script>alert(0)</script>”">`
    - ` <script>alert(0)</script>">`
- What if you cant use the “<” or “>” symbols?
  - You wont be able to break out of the tag, but you may be able to break out of the parameter
  - Payload: `“ onmouseover="alert(0)" blah="`
  - Injected tag:
    - ``
    - ``

# Injecting into Existing JavaScript

- Sometimes user input ends up inside existing JavaScript:

```
<script>
  $(document).ready( function() {
    if(window.location.hash) {
      var dogs = '[user input]';
      document.getElementById("dogs").innerHTML = dogs;
    }
  });
</script>
```

- In this case, if you want your script to execute you need to insert valid JavaScript so the existing script executes without errors
  - Payload: `'; alert(0); var a = `
  - Injected line: var dogs = `'; alert(0); var a = `';
- How else could you insert alert(0);?

# Testing for Clickjacking

- Does the server return the X-Origins header?
- Does the HTML source contain frame-busting JavaScript code?
- Can you embed the page in an iFrame hosted on another domain? (it should not be embeddable)

# Testing for CSRF

- Watch the traffic between the browser and server when submitting a form
  - Is there a unique token sent with the form? (should be present)
  - Is the token in the header or the body of the request? (should at least be in the body)
  - Does the request still go through when the token is removed? (request should be blocked)
  - Does the request still go through when the token is altered? (request should be blocked)
- When in doubt, write a proof of concept and see if it works!

# Testing for SQLi

- Supply unexpected user input such as ‘ “ ) -- #
- Identify any error messages or changes in response/behavior
- Determine if your input is being executed as code
- Types of searching:
  - Regular – see if extra data is returned
  - Equivalency – see if statements are executed differently
  - Blind – see if you can cause a backend delay or out-of-band response

# Testing for SQLi (text data)

- Does the DB send an error back when it receives a ‘ or “ or ) or –
- If you get an error, read it
- Does sending ” (two single ticks) alleviate the error?
- Test to see if the DB does the same thing when you input FOO as it does when you input:
  - ‘||’FOO (Oracle)
  - ‘+’FOO (MS-SQL)
  - ‘ ‘FOO (space between the single ticks) (MySQL)

# Testing for SQLi (numerical data)

- Supply a simple mathematical expression
  - If testing for two supply  $1+1$  or  $3-1$
- User a more complicated expression such as:
  - $67 - \text{ASCII}('A')$   $67 - 65 = 2$
  - $51 - \text{ASCII}(1)$   $51 - 49 = 2$

# Testing for Session Fixation

- Login
- Change the session token (keep the same format)
- Login again
- See if a new token is issued

# Testing for Improper Session Invalidation

- Login
- Copy the session token
- Logout
- Put the old session token back
- See if you are logged in

# Testing Cookie Settings

- **Expiration** - when the cookie expires
  - Don't set! Otherwise it is written to disk
  - If expiration or Max-Age aren't set it will be deleted when the browser closes
- **HttpOnly** - whether the cookie is accessible to JS
  - Set to True
- **Secure** - whether the cookie is sent encrypted
  - Set to True if the site uses SSL/TLS

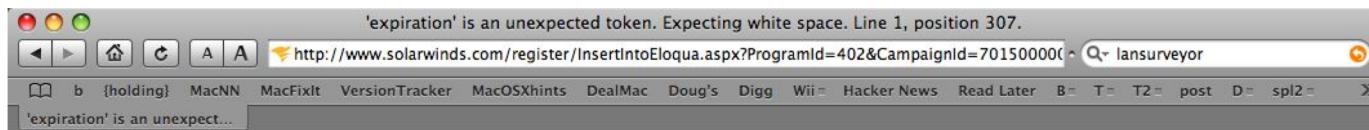
# Testing for User Enumeration

- Error messages
  - Try logging in with several usernames (some valid and some invalid) several times
    - Use invalid passwords
  - See if the error messages change
  - Do error messages indicate whether or not the username used is valid? (it shouldn't)

# Information Disclosure – Server Response Headers

- Server response headers are leaky!
- [-] x-powered-by: ASP.NET
- [-] server: Microsoft-IIS/7.0
- [-] x-aspnet-version: 4.0.30319
- [-] x-powered-by: ASP.NET
- [-] server: Microsoft-IIS/7.5
- [-] x-aspnetmvc-version: 3.0
- [-] x-powered-by: PHP/5.3.27
- [-] server: nginx/1.2.9
- [-] x-powered-by: PHP/5.3.6
- [-] server: Apache/2.2.3 (CentOS)
- [-] x-powered-by: PHP/5.3.3
- [-] server: Apache/2.2.15 (Red Hat) DAV/2 PHP/5.3.3 mod\_ssl/2.2.15 OpenSSL/1.0.0-fips mod\_perl/2.0.4 Perl/v5.10.1

# Information Disclosure – Error Messages



## Server Error in '/' Application.

*'expiration' is an unexpected token. Expecting white space. Line 1, position 307.*

**Description:** An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

**Exception Details:** System.Xml.XmlException: 'expiration' is an unexpected token. Expecting white space. Line 1, position 307.

### **Source Errors**

```
Line 90: XmlDocument oXML = new XmlDocument();
Line 91: 
Line 92: oXML.LoadXml(Request.Cookies["RegistrationDetails"].Value);
Line 93: XmlNode oName = oXML.DocumentElement.SelectSingleNode("Name");
Line 94: string sCountry;
```

**Source File:** e:\InetPub\wwwsync\register\InsertIntoEloqua.aspx.cs    **Line:** 92

### Stack Trace:

```
[XmlException: 'expiration' is an unexpected token. Expecting white space. Line 1, position 307.]  
System.Xml.XmlTextReaderImpl.Throw(Exception e) +90  
System.Xml.XmlTextReaderImpl.Throw(String res, String arg) +127  
System.Xml.XmlTextReaderImpl.ParseAttributes() +1949846  
System.Xml.XmlTextReaderImpl.ParseElement() +545  
System.Xml.XmlTextReaderImpl.ParseElementContent() +461  
System.Xml.XmlTextReaderImpl.Read() +29  
System.Xml.XmlLoader.LoadNode(Boolean skipOverWhitespace) +557  
System.Xml.XmlLoader.LoadDocSequence(XmlDocument parentDoc) +50  
System.Xml.XmlLoader.Load(XmlDocument doc, XmlReader reader, Boolean preserveWhitespace) +162  
System.Xml.XmlDocument.Load(XmlReader reader) +96  
System.Xml.XmlDocument.LoadXml(String xml) +197  
InsertIntoEloqua.Page_Load(Object sender, EventArgs e) in e:\InetPub\wwwsync\register\InsertIntoEloqua.aspx.cs:92  
System.Web.Util.CalliHelper.EventArgFunctionCaller(IntPtr fp, Object o, Object t, EventArgs e) +15  
System.Web.Util.CalliEventHandlerDelegateProxy.Callback(Object sender, EventArgs e) +33  
System.Web.UI.Control.OnLoad(EventArgs e) +99  
System.Web.UI.Control.LoadRecursive() +47  
System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +1436
```

**Version Information:** Microsoft .NET Framework Version:2.0.50727.1433; ASP.NET Version:2.0.50727.1433

# Information Disclosure – Error Messages

The screenshot shows a Google Chrome browser window with the title "Apple - Rebates - Rebate Status - Google Chrome". The address bar displays "www.apple.com/promo/rebate/status.html". The page content is titled "Apple Rebates" and includes links for "Current Rebates", "Rebate Status", and "FAQs". A prominent red error message reads "Server Error in '/APP' Application." Below this, a section titled "Configuration Error" contains the following details:

**Description:** An error occurred during the processing of a configuration file required to service this request. Please review the specific error details below and modify your configuration file appropriately.

**Parser Error Message:** Could not load file or assembly 'Microsoft.ReportViewer.WebForms, Version=9.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a' or one of its dependencies. There is not enough space on the disk. (Exception from HRESULT: 0x80070070)

**Source Error:**

```
Line 1790:      <add verb="*" path="_AppService.axd" validate="false" type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Exten
Line 1791:      <add verb="GET,HEAD" path="ScriptResource.axd" type="System.Web.Handlers.ScriptResourceHandler, System.Web.Extensions,
Line 1792:      <add verb="*" path="Reserved.ReportViewerWebControl.axd" type="Microsoft.Reporting.WebForms.HttpHandler, Microsoft.Repor
Line 1793:      </httpHandlers>
Line 1794:      </httpModules>
```

**Source File:** D:\Apps\IIS\WEB\WEBAPP\web.config    **Line:** 1792

**Assembly Load Trace:** The following information can be helpful to determine why the assembly 'Microsoft.ReportViewer.WebForms, Version=9.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a' could not be loaded.

WRN: Assembly binding logging is turned OFF.  
To enable assembly bind failure logging, set the registry value [HKLM\Software\Microsoft\Fusion!EnableLog] (DWORD) to 1.  
Note: There is some performance penalty associated with assembly bind failure logging.  
To turn this feature off, remove the registry value [HKLM\Software\Microsoft\Fusion!EnableLog].

**Version Information:** Microsoft .NET Framework Version:2.0.50727.3625; ASP.NET Version:2.0.50727.3634