INVISIBILITY PURGE

Unmasking Dormant Events of Invisible Server Web Controls

Advanced Hacking of ASP.Net, Mono and RIA

Shay Chen

Senior Manager, Hacktics CTO

Hacktics ASC, Ernst & Young

OWASP

The Open Web Application Security Project

About Hacktics

About HACKTICS

- Formerly a boutique company that provided various information security services since 2004.
- As of 01/01/2011, Ernst & Young acquired Hacktics professional services practice, and the group joined EY as one of the firm’s advanced security centers (ASC).

Open Source Projects

HACKTICS

Current/Legacy Projects

diviner
An active information gathering platform

ria-scip
An OWASP ZAP extension for enumerating and activating events

burp-log-reviver
A solution for converting burp logs into sessions

sn-crawler
An intelligence gathering platform focused on social networks.

payload-manager
An attack payload management tool.

puzzlemall
A vulnerable web application for practicing session puzzling

Dynamic AJAX CSRF template
Dynamic AJAX CSRF - POC Code.

Session Keep Alive
A POC tool for connection pool consumption **delay of** service attack.

Ultimate Obsolete File Detection - ZAP Plugin

Introducing…

SCIP!

Server Control Invisibility Purge

A project based on a research by **Niv Sela** and **Shay Chen**,

OWASP ZAP extension implementation by **Alex Mor**.

A project used for…

EodSec

Execution of dormant server events & controls

Can be abused to…

OWASP
The Open Web Application Security Project

EodSec

Exploitation Scenarios:

- Elevate privileges by executing events of high-privileged users
- Exploit vulnerable code stored in dormant events
- Corrupt the application data
- Exceed logical restrictions
- Etc

Agenda

OWASP
The Open Web Application Security Project

- The Attack Surface of RIA Applications
- Server Controls, Events and Lifecycles
- Invisible Web Controls & Dormant Events
- Dormant Event Activation, Control Fuzzing & Event Enumeration
- Control Enumeration / Event Execution via SCIP: Diviner/OWASP ZAP Extension
- Risk Mitigation
- Q & A

The Attack Surface of RIA

Facing the Horde of Security Features

Security Features in RIA

- Event Validation
- Digital Signatures: Limit to List, Manipulation Prevention
- Security Filter (XSS)
- Sandbox
- Built-in Regular Expressions
- Secure Database Access Methods
- Etc

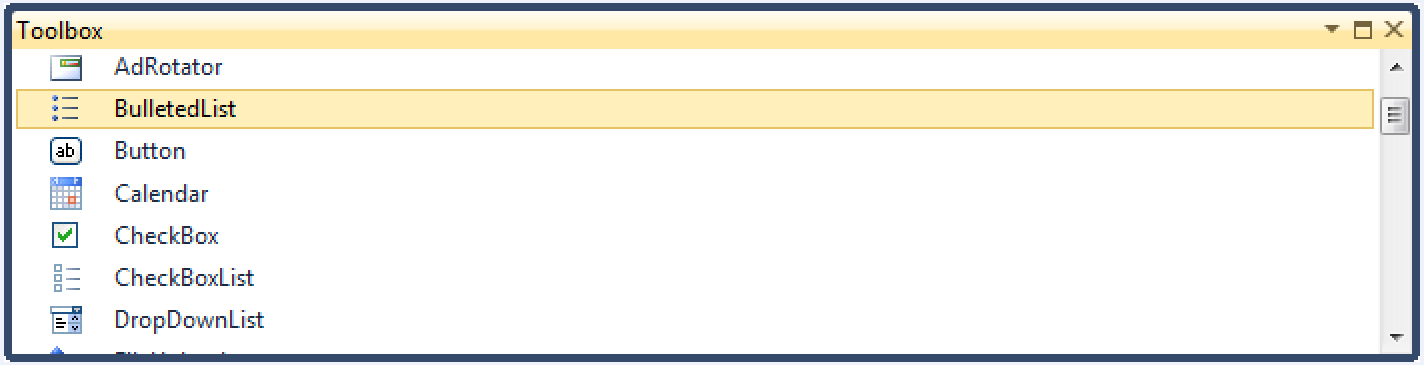
Attack Surface Analysis

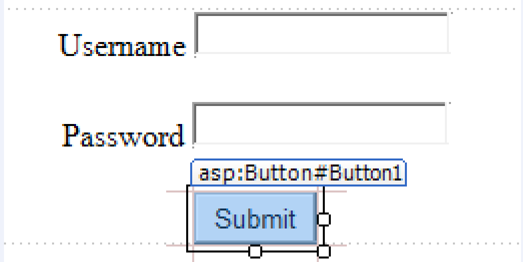
OWASP
The Open Web Application Security Project

- **Purpose:** Locating Code that can be Abused
  - Web Pages
  - Web Service Methods
  - Global Modules (Filters, Handlers, Etc)
  - …
  - *Events of Web Application Server Controls*

Server Web Controls

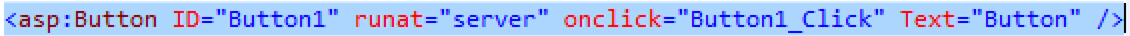
- Rendered into HTML/JS code, but include server side implementation
- Core Controls and Custom Controls (e.g. ascx)





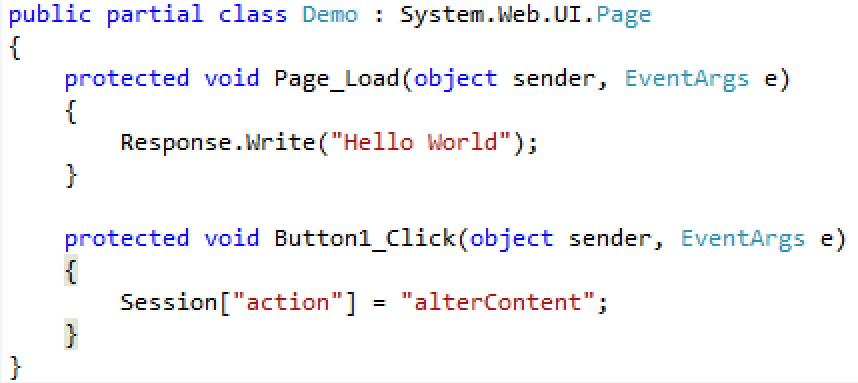
Server Control Events

- A triggered server side code segment, containing optional functionality (PostBack/CallBack in ASP.Net)
- Client triggered events rely on the EVENTTARGET, EVENTARGUMENT and VIEWSTATE mechanisms
- Sample Server Side Implementation (C#, ASP.Net):
  - aspx:



```
<asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Button" />
```

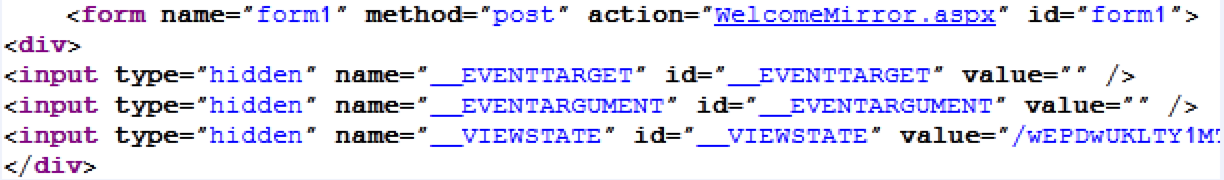
  - aspx.cs:



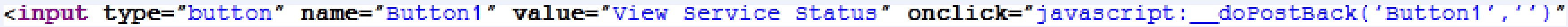
```
public partial class Demo : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Response.Write("Hello World");
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Session["action"] = "alterContent";
    }
}
```

Control Events, Cont.

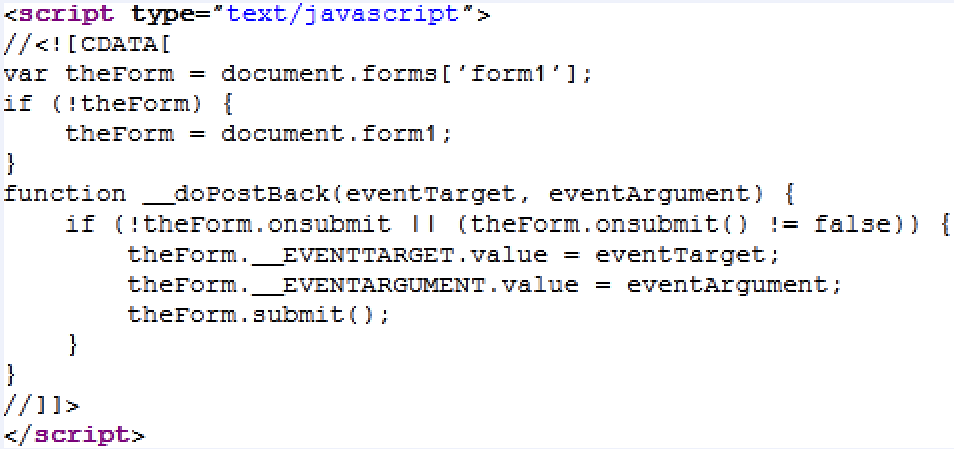
Sample client-side implementation (ASP.Net postback):



```
    <form name="form1" method="post" action="WelcomeMirror.aspx" id="form1">
<div>
<input type="hidden" name="__EVENTTARGET" id="__EVENTTARGET" value="" />
<input type="hidden" name="__EVENTARGUMENT" id="__EVENTARGUMENT" value="" />
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUKLTY1M
</div>
```



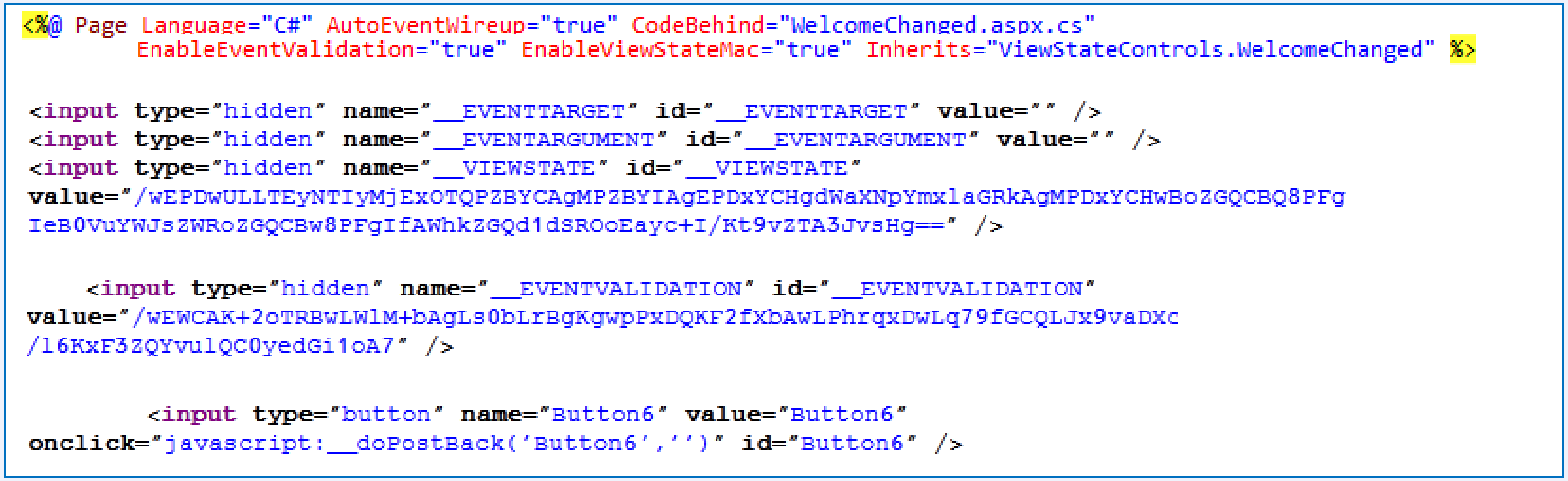
```
<input type="button" name="Button1" value="View Service Status" onclick="javascript:__doPostBack('Button1','')"
```



```
<script type="text/javascript">
//<![CDATA[
var theForm = document.forms['form1'];
if (!theForm) {
    theForm = document.form1;
}
function __doPostBack(eventTarget, eventArgument) {
    if (!theForm.onsubmit || (theForm.onsubmit() != false)) {
        theForm.__EVENTTARGET.value = eventTarget;
        theForm.__EVENTARGUMENT.value = eventArgument;
        theForm.submit();
    }
}
//]]>
</script>
```

Event Validation Drill-Down

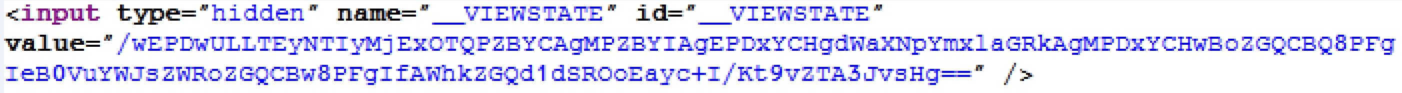
- Independent Events: buttons with usesubmitbehavior=false, checkboxes, etc
- Sample Event Lifecycle
- Programmatic vs. Declarative



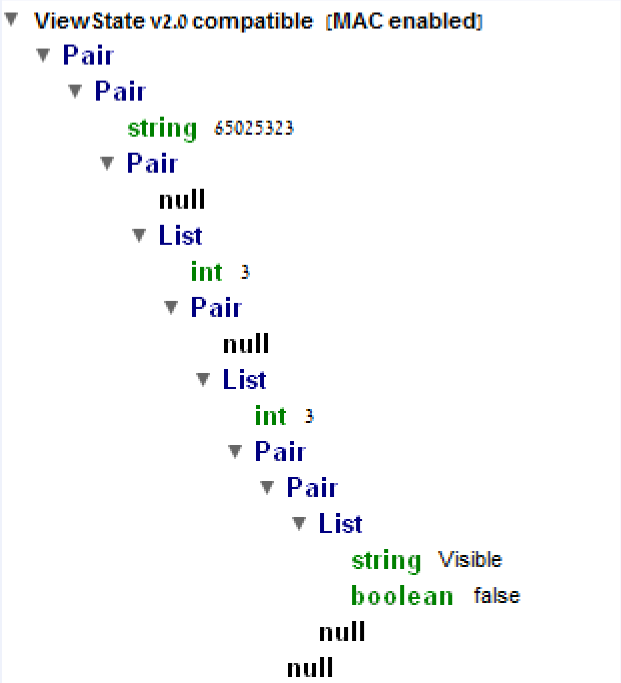
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WelcomeChanged.aspx.cs"
        EnableEventValidation="true" EnableViewStateMac="true" Inherits="ViewStateControls.WelcomeChanged" %>

<input type="hidden" name="__EVENTTARGET" id="__EVENTTARGET" value="" />
<input type="hidden" name="__EVENTARGUMENT" id="__EVENTARGUMENT" value="" />
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
value="/wEPDwULLTEyNTIyMjExOTQPZBYCAgMPZBYIAgEPDxYCHgdWaXNpYmxlaGRkAgMPDxYCHwBoZGQCBQ8PFg
IeB0VuYWJsZWRoZGQCBw8PFgIfAWhkZGQd1dSROoEayc+I/Kt9vZTA3JvsHg==" />

    <input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION"
value="/wEWCAK+2oTRBwLWlM+bAgLs0bLrBgKgwpPxDQKF2fXbAwLPhrqxDwLq79fGCQLJx9vaDXc
/16KxF3ZQYvulQC0yedGi1oA7" />

        <input type="button" name="Button6" value="Button6"
onclick="javascript:__doPostBack('Button6','')" id="Button6" />
```

Viewstate Structure

- **Viewstate Structure**



```
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
value="/wEPDwULLTEyNTIyMjExOTQPZBYCAgMPZBYIAgEPDxYCHgdWaXNpYmxlaGRkAgMPDxYCHwBoZGQCBQ8PFg
IeB0VuYWJsZWRoZGQCBw8PFgIfAWhkZGQd1dSROoEayc+I/Kt9vZTA3JvsHg==" />
```



```
▼ ViewState v2.0 compatible [MAC enabled]
  ▼ Pair
    ▼ Pair
        string 65025323
      ▼ Pair
          null
        ▼ List
            int 3
          ▼ Pair
              null
            ▼ List
                int 3
              ▼ Pair
                ▼ Pair
                  ▼ List
                      string Visible
                      boolean false
                    null
                  null
```

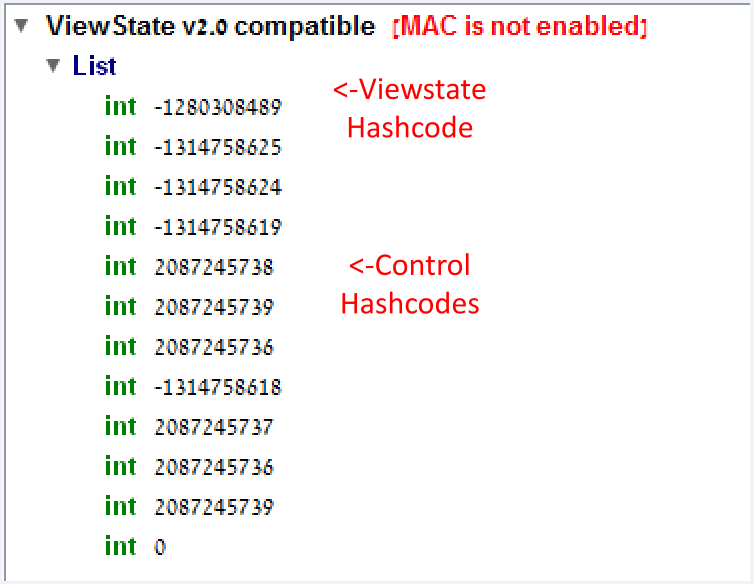
- Serialized into Base64*
- http://msdn.microsoft.com/en-us/library/ms972976.aspx
- Signed (MAC), clear-text or encrypted

Event Validation Mechanism

- **Name/Value HashCode Formula**

```
if ([ControlValue] == null)
   return GetStringHashCode([ControlName]);
else
   return GetStringHashCode([ControlName]) ^ GetStringHashCode([ControlValue]);
```

EventValidation (Viewed via Burp Viewstate Decoder):

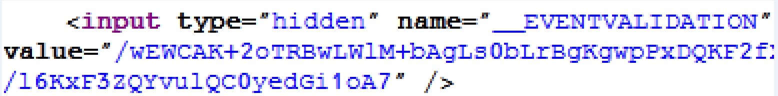


```
▼ ViewState v2.0 compatible  [MAC is not enabled]
  ▼ List
      int  -1280308489
      int  -1314758625
      int  -1314758624
      int  -1314758619
      int  2087245738
      int  2087245739
      int  2087245736
      int  -1314758618
      int  2087245737
      int  2087245736
      int  2087245739
      int  0
```

<-Viewstate Hashcode

<-Control Hashcodes

- MachineKey and MAC
- Control Name/Value Verification, Prior to Event Execution
- Include viewstate hashcode
- **Included in the HTML:**

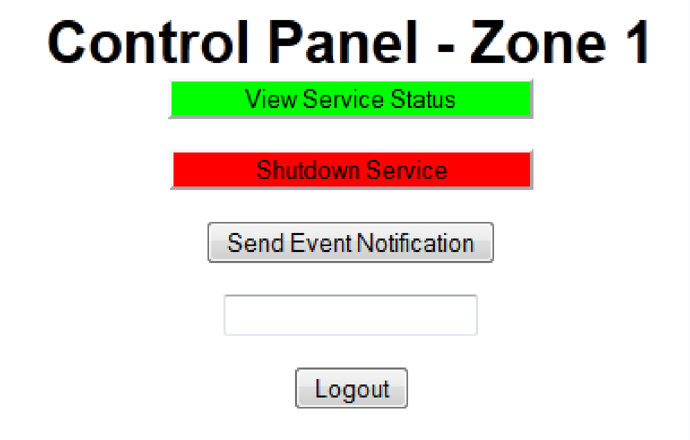


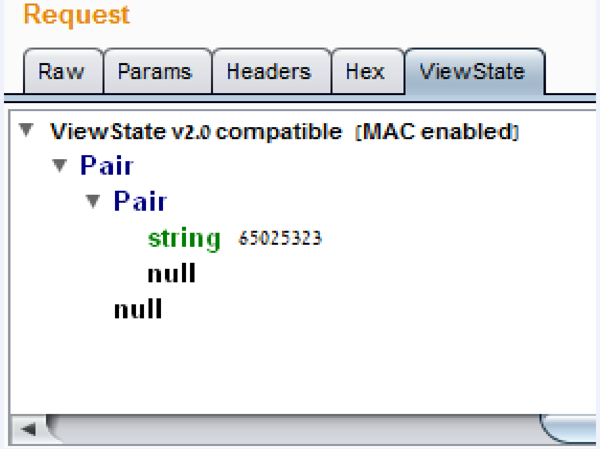
```
    <input type="hidden" name="__EVENTVALIDATION"
value="/wEWCAK+2oTRBwLWlM+bAgLs0bLrBgKgwpPxDQKF2f
/16KxF3ZQYvulQC0yedGi1oA7" />
```

Evidence of Hidden Controls

OWASP
The Open Web Application Security Project

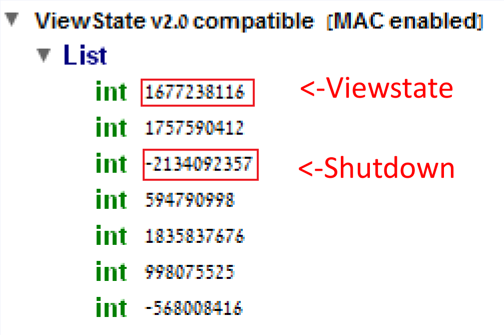
Visible / Enabled Controls:





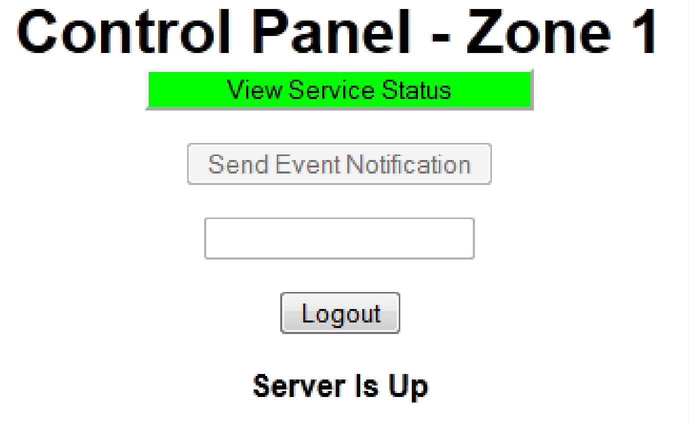
```
▼ ViewState v2.0 compatible [MAC enabled]
  ▼ Pair
    ▼ Pair
        string  65025323
        null
      null
```

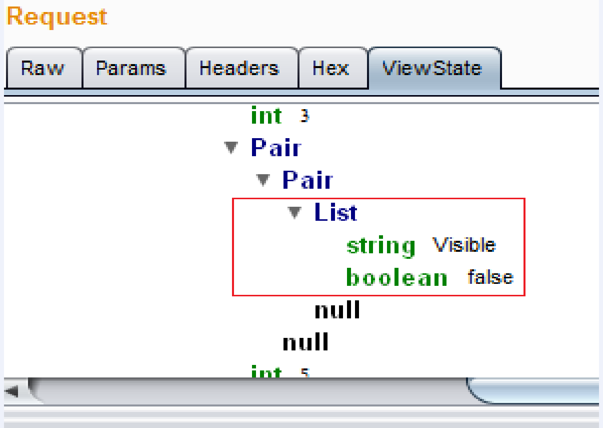
EventValidation (Viewstate Decoder):



```
▼ ViewState v2.0 compatible [MAC enabled]
  ▼ List
      int  1677238116      <-Viewstate
      int  1757590412
      int  -2134092357     <-Shutdown
      int  594790998
      int  1835837676
      int  998075525
      int  -568008416
```

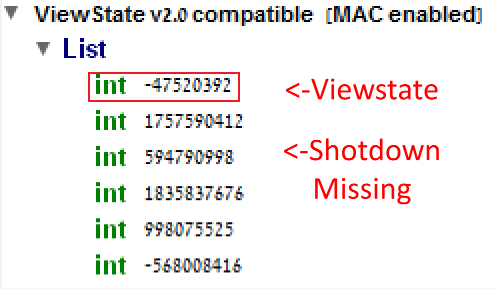
Invisible / Disabled Controls (Control Trace in Viewstate!):





```
int  3
▼ Pair
  ▼ Pair
    ▼ List
        string  Visible
        boolean  false
      null
    null
int  5
```

EventValidation (Viewstate Decoder):



```
▼ ViewState v2.0 compatible [MAC enabled]
  ▼ List
      int  -47520392       <-Viewstate
      int  1757590412
      int  594790998       <-Shotdown
      int  1835837676        Missing
      int  998075525
      int  -568008416
```

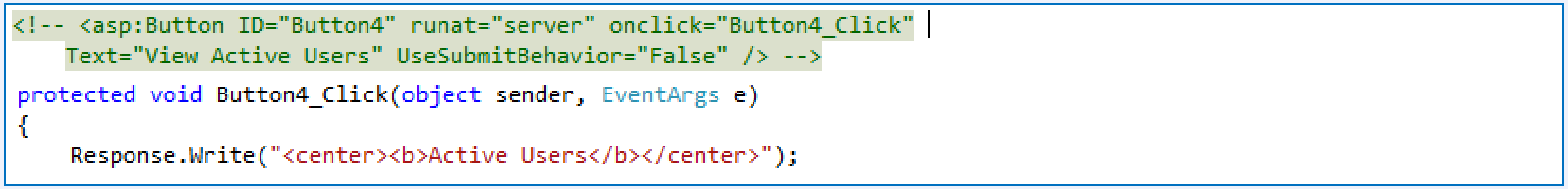
ERNST & YOUNG
Quality In Everything We Do

Invisible Web Controls: Archetypes

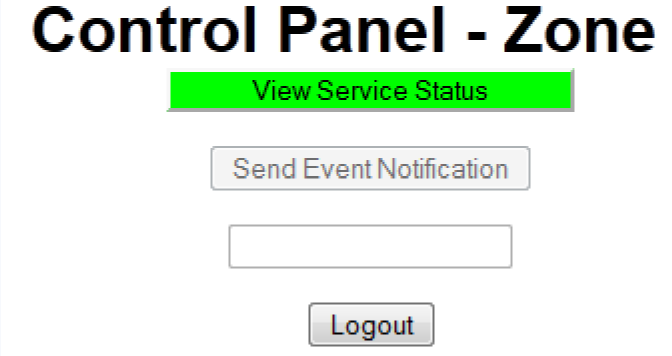
Dormant Controls/Events, 1/3

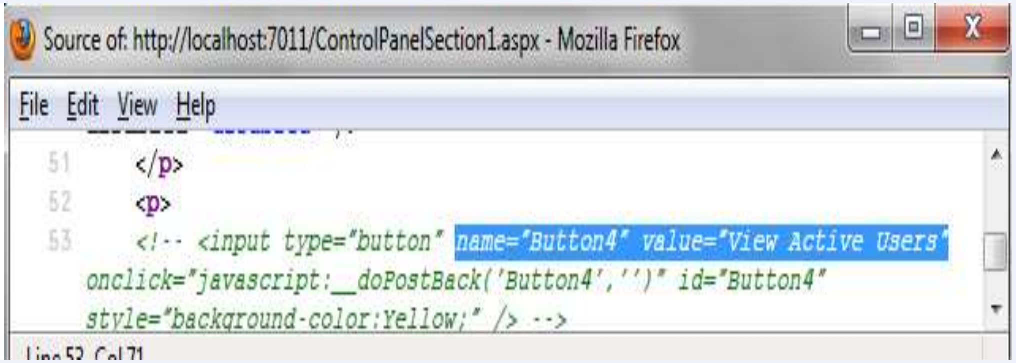
OWASP
The Open Web Application Security Project

- **Commented Out Controls**
  - The control is commented out using HTML comments
  - **Rendered inside an HTML comment,** but the server code is still active.



```
<!-- <asp:Button ID="Button4" runat="server" onclick="Button4_Click"
    Text="View Active Users" UseSubmitBehavior="False" /> -->
protected void Button4_Click(object sender, EventArgs e)
{
    Response.Write("<center><b>Active Users</b></center>");
```

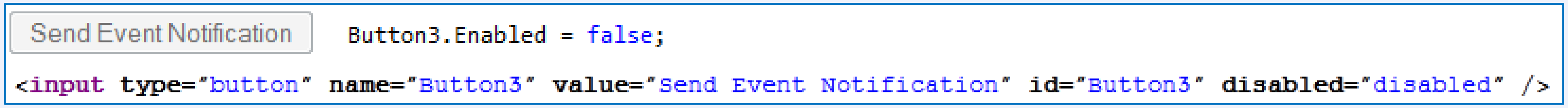



Source of: http://localhost:7011/ControlPanelSection1.aspx - Mozilla Firefox
File Edit View Help

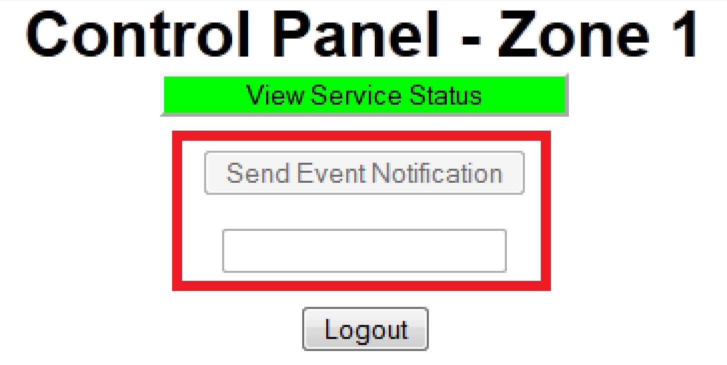
```
    </p>
    <p>
    <!-- <input type="button" name="Button4" value="View Active Users"
onclick="javascript:__doPostBack('Button4','')" id="Button4"
style="background-color:Yellow;" /> -->
```

Dormant Controls/Events, 2/3

- **Disabled Controls**
  - The control **enabled** property is set to **false**
  - Rendered with the **disabled="disabled"** HTML property
  - Rendered **without** an input **postback** method



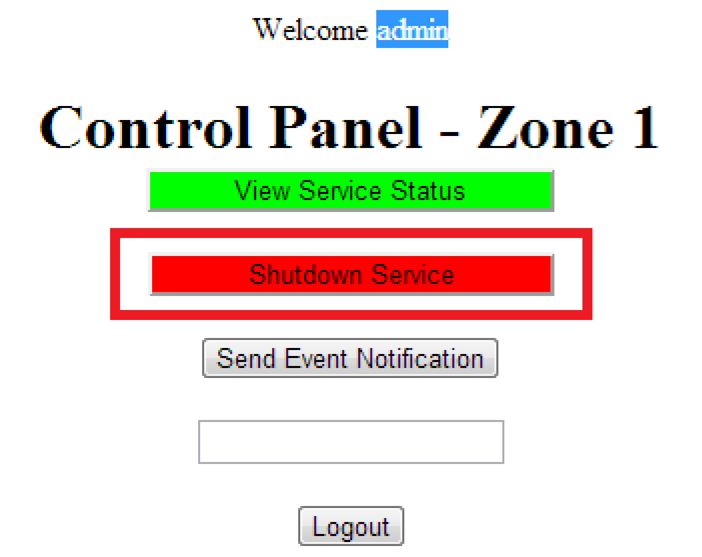
```
Button3.Enabled = false;
<input type="button" name="Button3" value="Send Event Notification" id="Button3" disabled="disabled" />
```

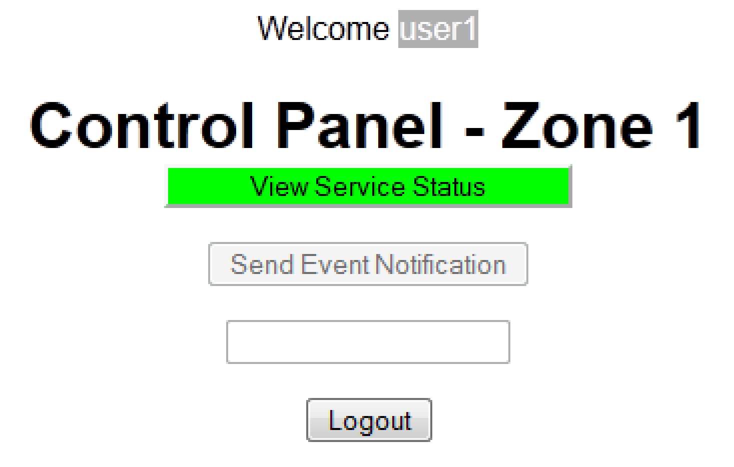

Control Panel - Zone 1

Dormant Controls/Events, 3/3

- **Invisible Controls**
  - The control **visible** property is set to **false**
  - **Not Rendered** in the presentation layer, but the code is still active

```
Button2.Visible = false;
```

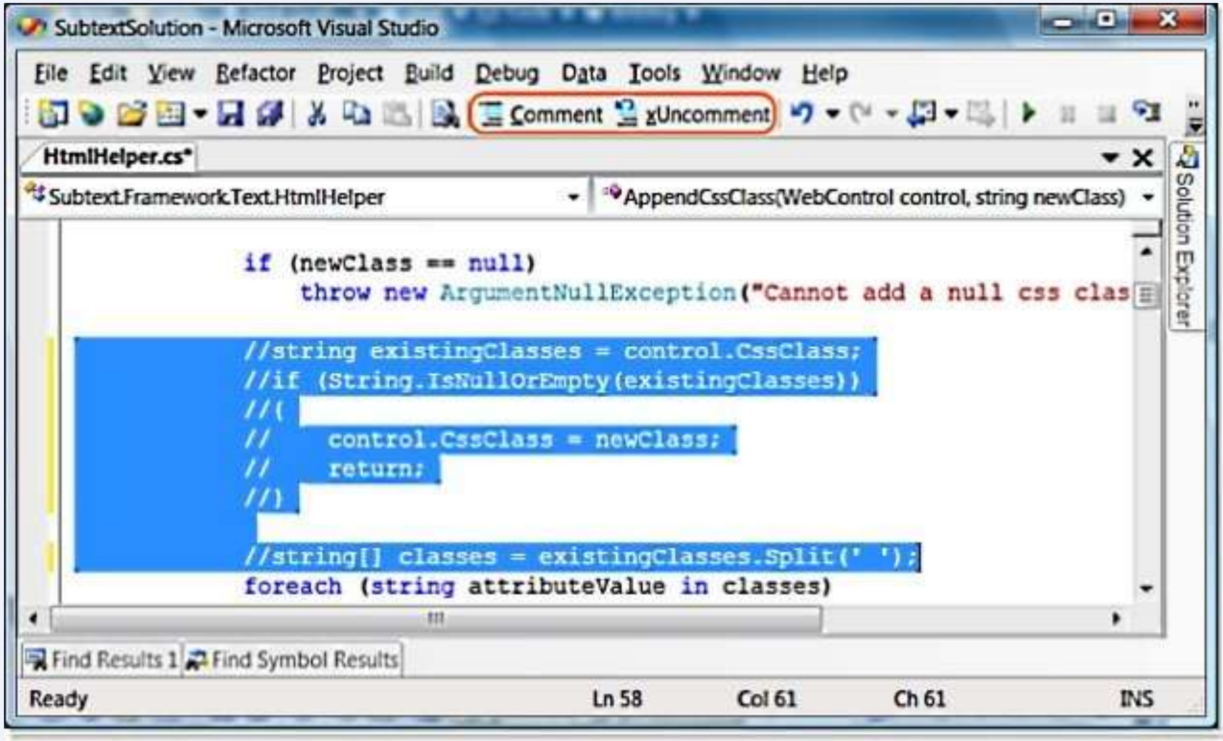




Dormant Controls/Events, Opt

- **Dormant Events of Visible Controls**
  - Optional event listeners registered in the code level, after the optional definition was deleted from a control with at least one active event.

Basic Dormant Event Activation: **Commented Controls**



SubtextSolution - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Data Tools Window Help
Comment
Uncomment
HtmlHelper.cs*
Subtext.Framework.Text.HtmlHelper
AppendCssClass(WebControl control, string newClass)
Solution Explorer
Find Results 1
Find Symbol Results
Ready
Ln 58
Col 61
Ch 61
INS

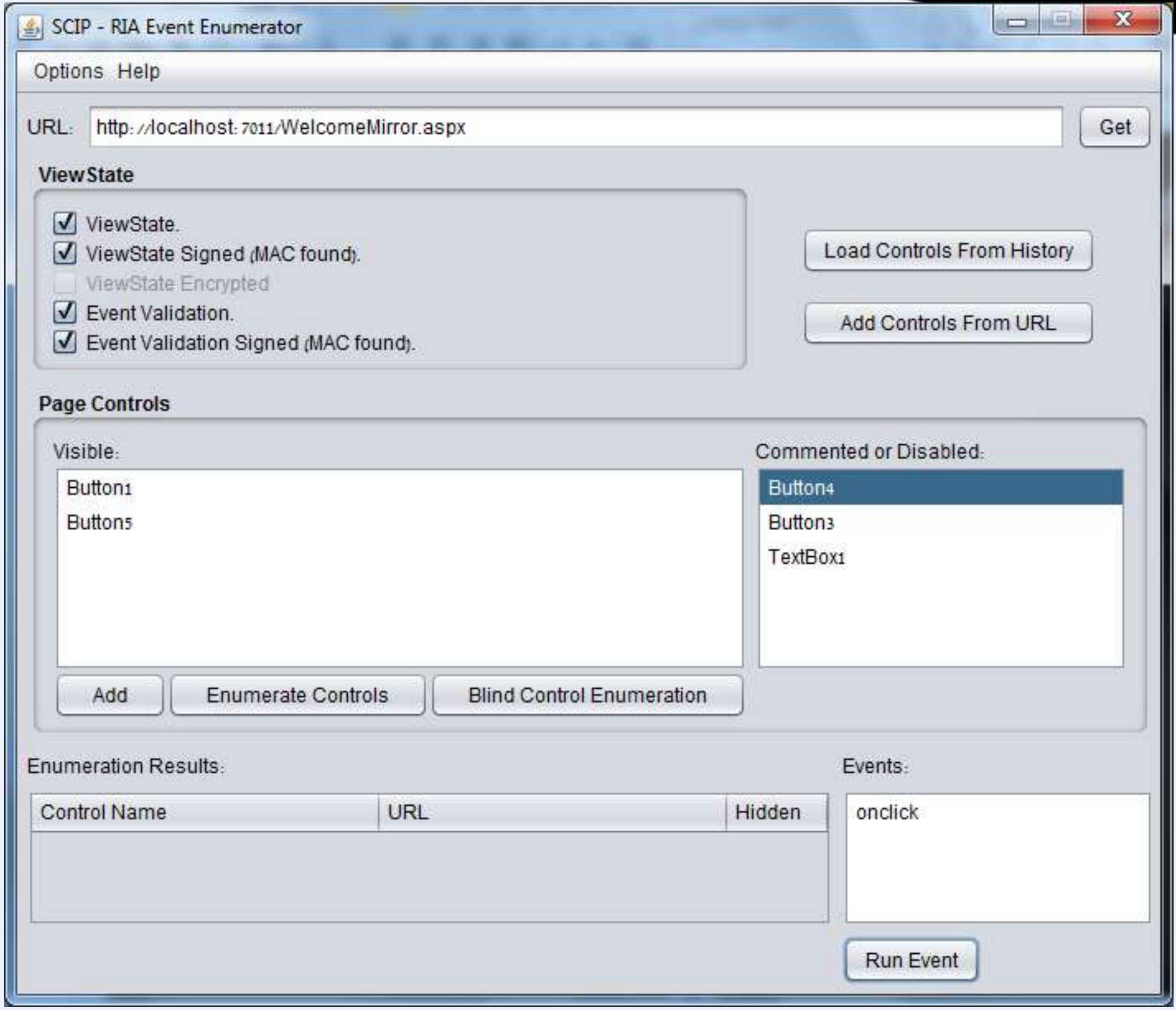
```
            if (newClass == null)
                throw new ArgumentNullException("Cannot add a null css clas

            //string existingClasses = control.CssClass;
            //if (String.IsNullOrEmpty(existingClasses))
            //{
            //    control.CssClass = newClass;
            //    return;
            //}

            //string[] classes = existingClasses.Split(' ');
            foreach (string attributeValue in classes)
```

Events of Commented Controls

- **Prerequisites (ASP.Net / Mono) - Commented Out Controls:**
  - The developer should rely solely on the fact that the control is commented.
  - The attacker can simply "uncomment" the HTML control and execute the embedded event, or send the appropriate values directly.
- **Advantages**
  - Exploit works **even** if the **Viewstate MAC** AND the **EventValidation** features are **turned ON**.

Events of Commented Controls

OWASP
The Open Web Application Security Project



SCIP - RIA Event Enumerator

URL: http://localhost:7011/WelcomeMirror.aspx

ViewState

☑ ViewState.
☑ ViewState Signed (MAC found).
□ ViewState Encrypted
☑ Event Validation.
☑ Event Validation Signed (MAC found).

Page Controls

Visible:
Button1
Button5

Commented or Disabled:
Button4
Button3
TextBox1

Enumeration Results:

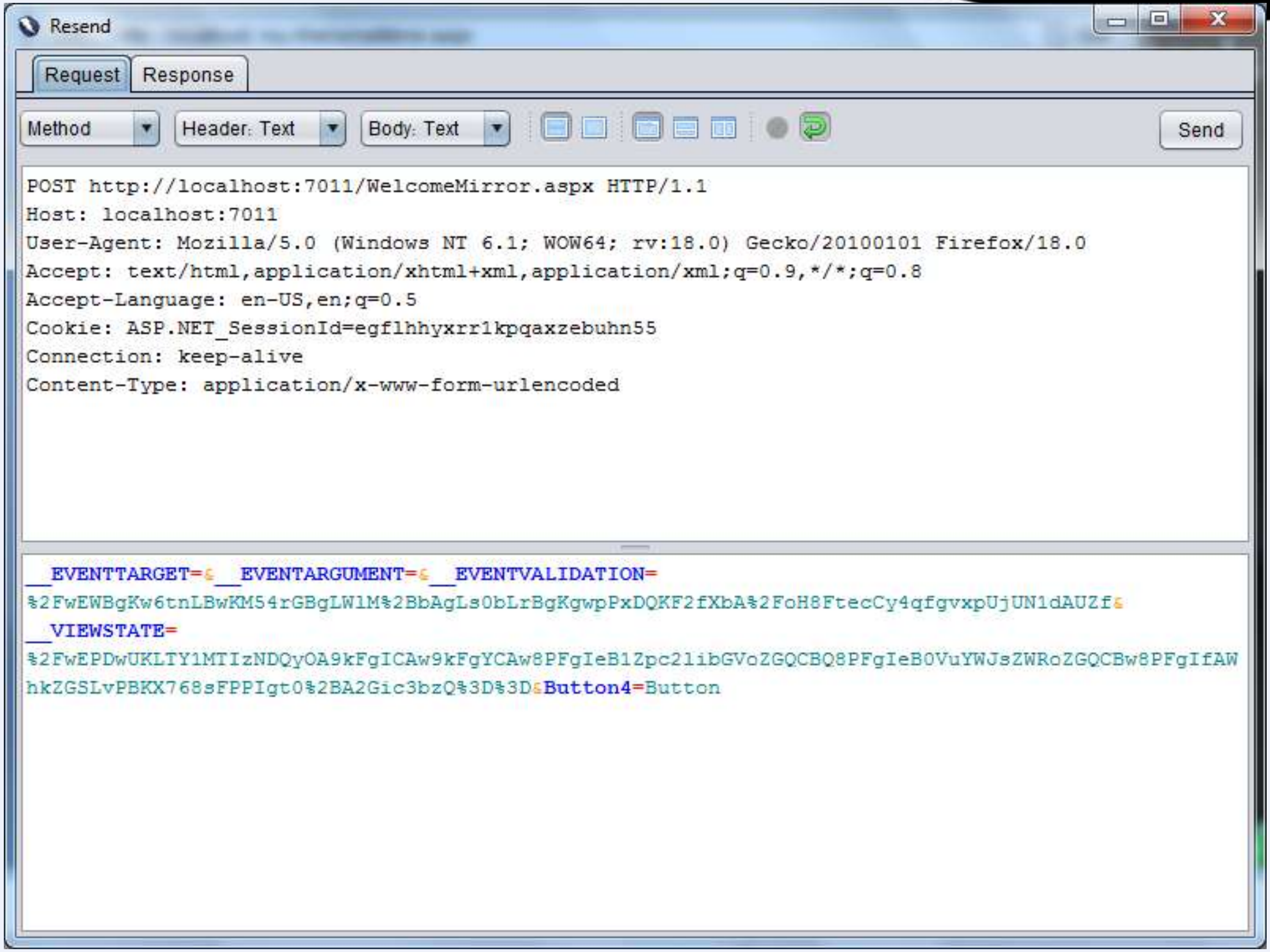
Control Name	URL	Hidden

Events:
onclick

ERNST & YOUNG
Quality In Everything We Do

Events of Commented Controls

OWASP
The Open Web Application Security Project

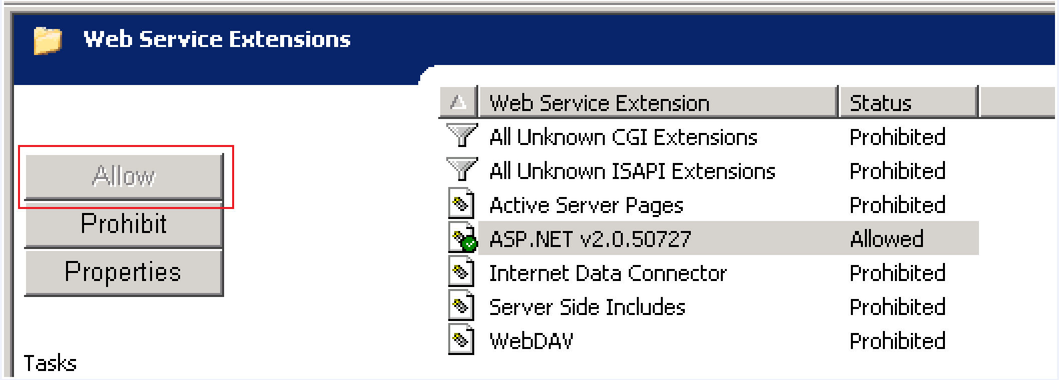


```
POST http://localhost:7011/WelcomeMirror.aspx HTTP/1.1
Host: localhost:7011
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:18.0) Gecko/20100101 Firefox/18.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Cookie: ASP.NET_SessionId=egflhhyxrr1kpqaxzebuhn55
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
```

```
__EVENTTARGET=&__EVENTARGUMENT=&__EVENTVALIDATION=
%2FwEWBgKw6tnLBwKM54rGBgLW1M%2BbAgLs0bLrBgKgwpPxDQKF2fXbA%2FoH8FtecCy4qfgvxpUjUN1dAUZf&
__VIEWSTATE=
%2FwEPDwUKLTY1MTIzNDQyOA9kFgICAw9kFgYCAw8PFgIeB1Zpc2libGVoZGQCBQ8PFgIeB0VuYWJsZWRoZGQCBw8PFgIfAW
hkZGSLvPBKX768sFPPIgt0%2BA2Gic3bzQ%3D%3D&Button4=Button
```

Intermediate Dormant Event Activation:

Disabled Controls



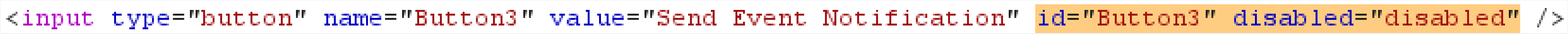
Web Service Extension	Status
All Unknown CGI Extensions	Prohibited
All Unknown ISAPI Extensions	Prohibited
Active Server Pages	Prohibited
ASP.NET v2.0.50727	Allowed
Internet Data Connector	Prohibited
Server Side Includes	Prohibited
WebDAV	Prohibited

Events of Disabled Controls

- **Prerequisites (ASP.Net / Mono) - Disabled Controls:**
  - The developer should rely solely on the control disability and the lack of JS postback/callback method for protecting the control events.
  - The attacker should forge a postback / callback method, or send the appropriate values directly.
- **Advantages**
  - Exploit works **even** if the **Viewstate MAC** AND the **EventValidation** features are **turned ON**.

Events of Disabled Controls

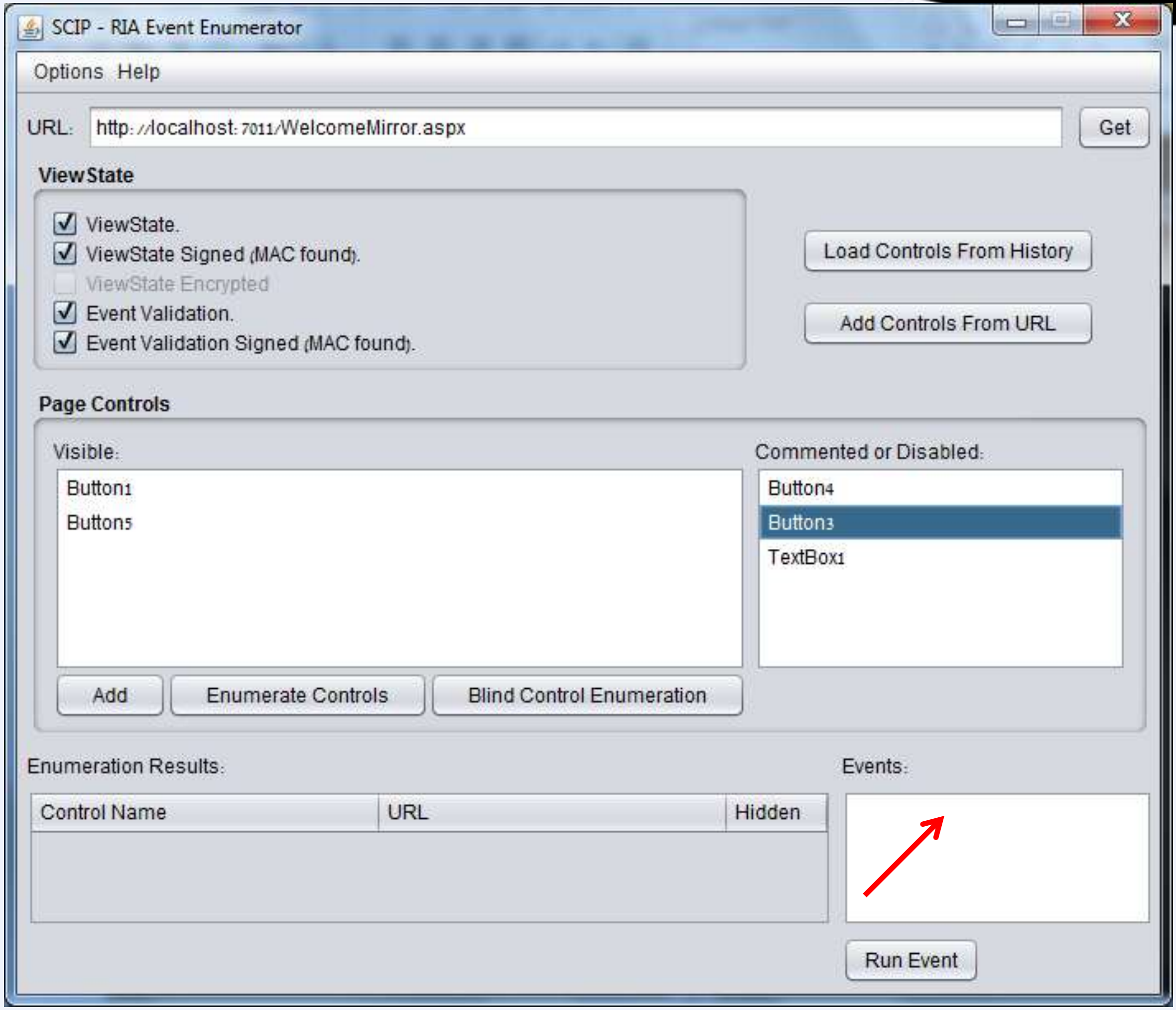
- **The Process of Forging a PostBack / CallBack Method**
  - Why does it work?
    - Using temporarily disabled controls in ASP.Net is **a feature**
    - Controls might be disabled without any relation to security, and thus, are currently not protected like invisible controls
  - How does it work?
    - The control name is exposed in the disabled control



```
<input type="button" name="Button3" value="Send Event Notification" id="Button3" disabled="disabled" />
```

-
  -
    - The attacker can use an interception proxy to “inject” postback calls into HTML control events, or craft requests manually by reusing the existing viewstate/validation fields.

Events of Disabled Controls



SCIP - RIA Event Enumerator

URL: http://localhost:7011/WelcomeMirror.aspx

ViewState

☑ ViewState.
☑ ViewState Signed (MAC found).
□ ViewState Encrypted
☑ Event Validation.
☑ Event Validation Signed (MAC found).

Page Controls

Visible:

- Button1
- Button5

Commented or Disabled:

- Button4
- Button3
- TextBox1

Enumeration Results:

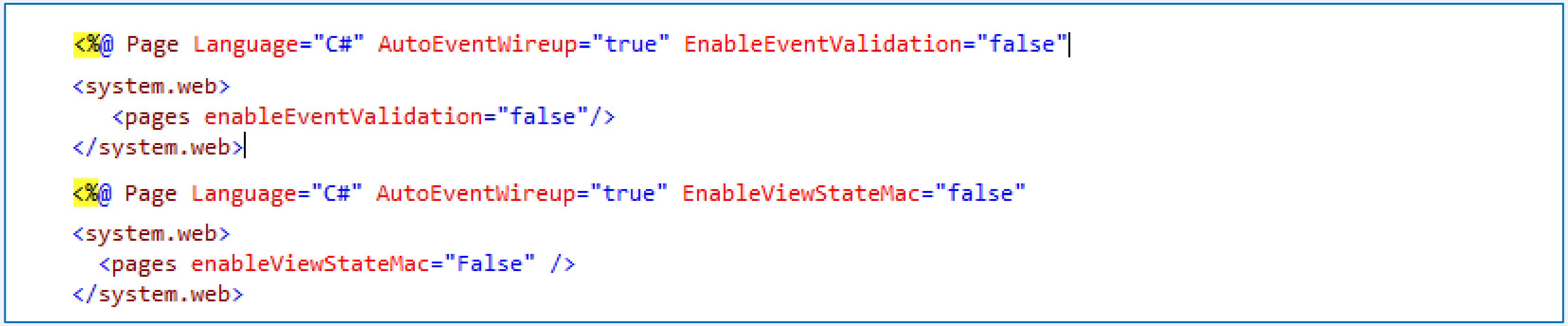
Control Name	URL	Hidden

Events:

Advanced Dormant Event Activation:
Invisible Controls!

Events of Invisible Controls

- **Prerequisites (ASP.Net / Mono) - Invisible Controls:**
  - (I) Either the Viewstate MAC **OR** the EventValidation features must be turned off.



```
<%@ Page Language="C#" AutoEventWireup="true" EnableEventValidation="false"

<system.web>
   <pages enableEventValidation="false"/>
</system.web>

<%@ Page Language="C#" AutoEventWireup="true" EnableViewStateMac="false"

<system.web>
  <pages enableViewStateMac="False" />
</system.web>
```

  - (II) The developer should rely solely on the control invisibility for protecting the invisible control events.

Events of Invisible Controls

- **EventValidation is ON but the Viewstate MAC is OFF**
  - In order for the attack to succeed, we need to forge a valid eventvalidation structure (no MAC)
    - Craft a request using SCIP or other viewstate/eventtarget editors

```
<%@ Page Language="C#" AutoEventWireup="true"
EnableEventValidation="true" EnableViewStateMac=“false“ …%>
```

Events of Invisible Controls

- **EventValidation is OFF**
  - Since there’s no event validation, any event can be executed, regardless of the viewstate value
    - Craft a request with valid EVENTTARGET value **OR**
    - Inject a custom Postback/Callback call to the response HTML, and target the event of the invisible control

```
<%@ Page Language="C#" AutoEventWireup="true"
EnableEventValidation=“false" EnableViewStateMac=“true“ …%>
```

- In all cases, we still need to obtain the control / event name…

Events of Invisible Controls

OWASP
The Open Web Application Security Project

- **The Process of Server Control Enumeration**
  - In this scenario, the control leaves no client-side traces:
    - Control Name Fuzzing
    - Core Controls vs. Custom Controls

- **Control Event Enumeration**
  - Core Events vs. Custom Events
  - Dormant Events vs. Active Events

ERNST & YOUNG
Quality In Everything We Do

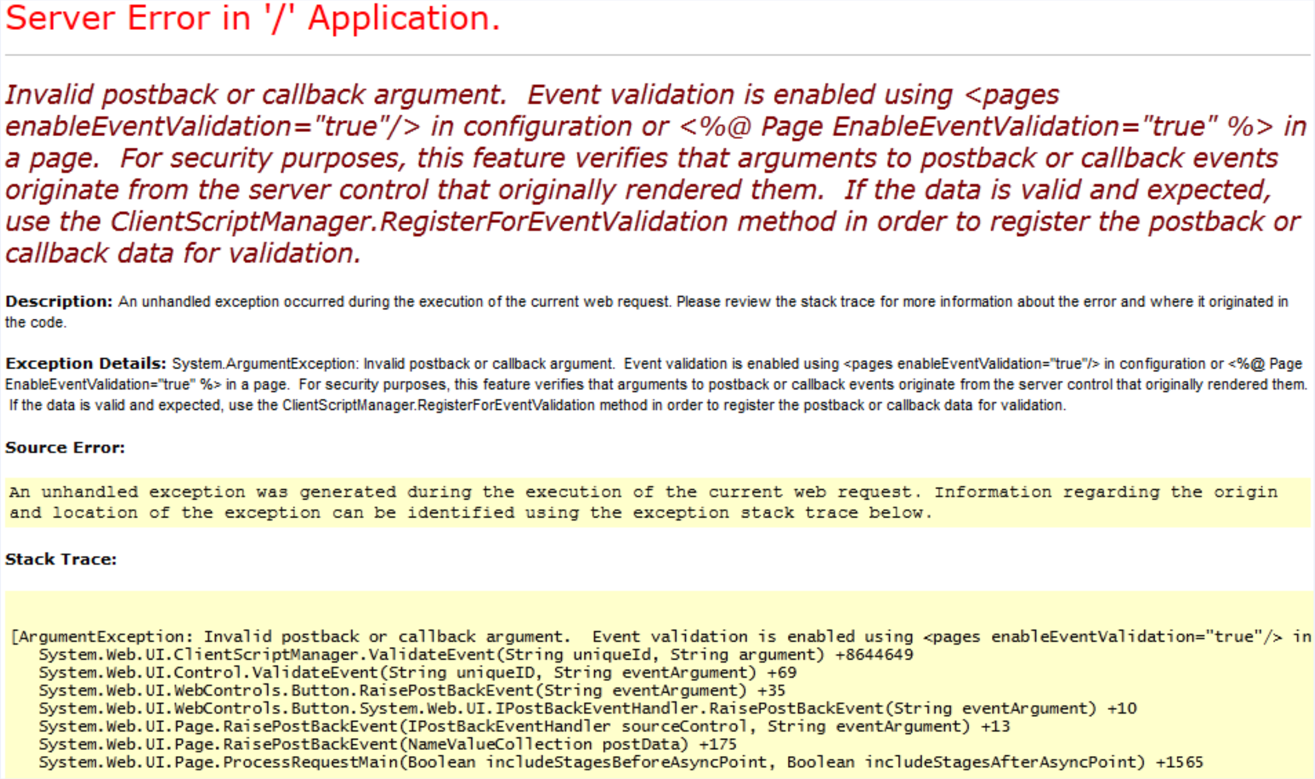
Common Naming Conventions

OWASP
The Open Web Application Security Project

- **Default:** [ControlType][Number]
  - Button1, Button2, TextBox1, TextBox2 …
- **Default II (v1.1-v3.5/Master):** ctl[ID]$[contentScope]$...
  - ctl00$MainContent$txtName, ctl00$Content$cmdSubmit
- **Legacy:** [ControlTypeShortCut][Number]
  - txt1, txt2, btn1, btn2, cmd1, cmd2, lst1, lst2 …
- **Custom Legacy:** [ControlTypeShortCut][Logic]
  - txtUsername, txtPassword, btnSubmit, cmdAddUser …
- **Plain:** [Logic]
  - user, pass, submit, delete
- **Title Match: [Title]**
  - Username, Password, Origin, Email, Update

Error-Based Enumeration

- Accessing invalid control names will NOT raise exceptions
- Accessing protected will – only works if EventValidation is **ON**



Server Error in '/' Application.

Invalid postback or callback argument. Event validation is enabled using <pages enableEventValidation="true"/> in configuration or <%@ Page EnableEventValidation="true" %> in a page. For security purposes, this feature verifies that arguments to postback or callback events originate from the server control that originally rendered them. If the data is valid and expected, use the ClientScriptManager.RegisterForEventValidation method in order to register the postback or callback data for validation.

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.ArgumentException: Invalid postback or callback argument. Event validation is enabled using <pages enableEventValidation="true"/> in configuration or <%@ Page EnableEventValidation="true" %> in a page. For security purposes, this feature verifies that arguments to postback or callback events originate from the server control that originally rendered them. If the data is valid and expected, use the ClientScriptManager.RegisterForEventValidation method in order to register the postback or callback data for validation.

Source Error:

```
An unhandled exception was generated during the execution of the current web request. Information regarding the origin
and location of the exception can be identified using the exception stack trace below.
```

Stack Trace:

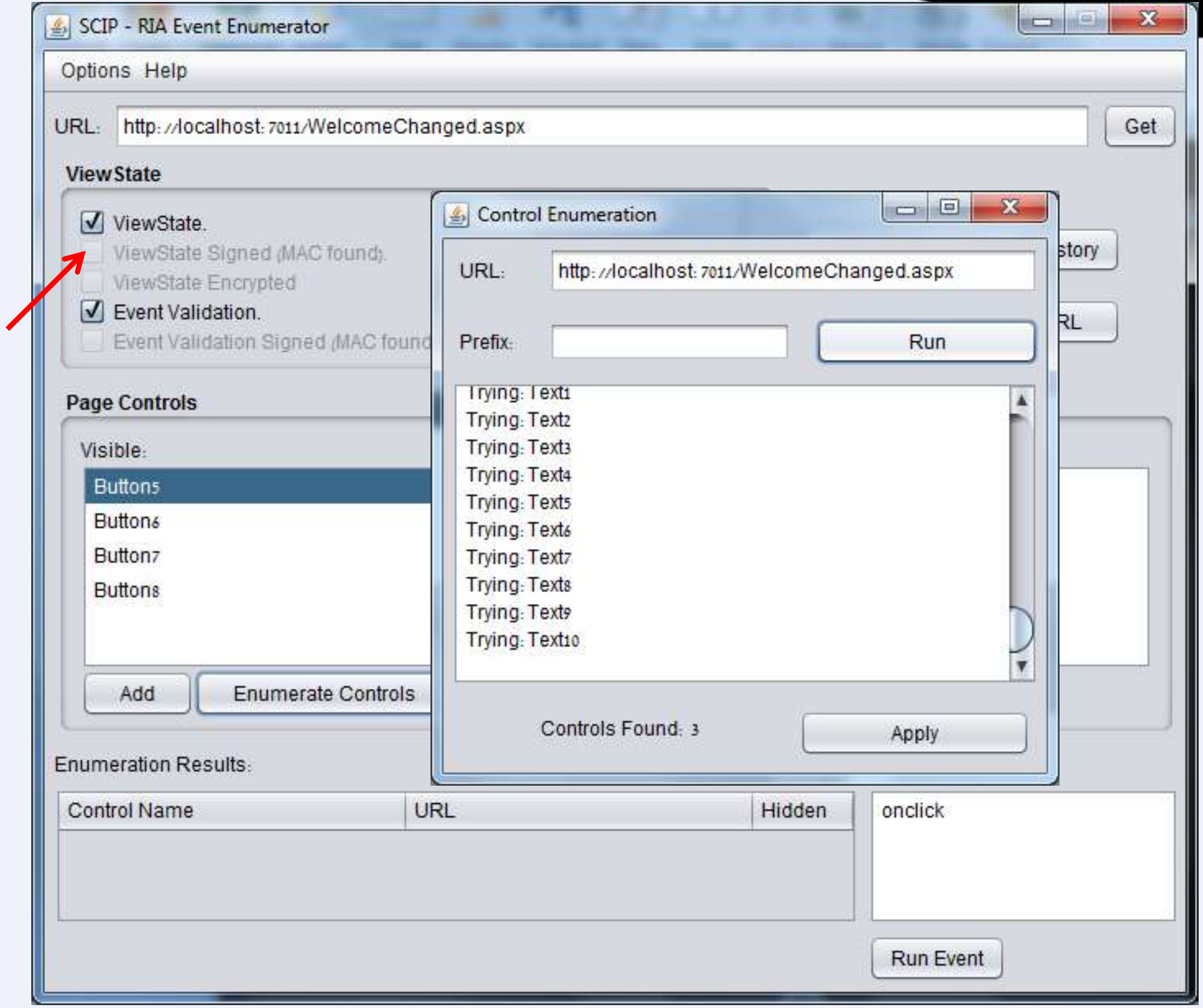
```
[ArgumentException: Invalid postback or callback argument.  Event validation is enabled using <pages enableEventValidation="true"/> in
   System.Web.UI.ClientScriptManager.ValidateEvent(String uniqueId, String argument) +8644649
   System.Web.UI.Control.ValidateEvent(String uniqueID, String eventArgument) +69
   System.Web.UI.WebControls.Button.RaisePostBackEvent(String eventArgument) +35
   System.Web.UI.WebControls.Button.System.Web.UI.IPostBackEventHandler.RaisePostBackEvent(String eventArgument) +10
   System.Web.UI.Page.RaisePostBackEvent(IPostBackEventHandler sourceControl, String eventArgument) +13
   System.Web.UI.Page.RaisePostBackEvent(NameValueCollection postData) +175
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +1565
```

Blind Enumeration

- Basic Blind Differentiation Formula:

```
ValidControlEvent = False;

OriginalResponse = getResponse(“Page1.aspx?param=value”);
VerificationResponse = getResponse(“Page1.aspx?param=value”);
ConfirmationResponse = getResponse(“Page1.aspx?param=value”);

InconsistentContent = VerificationResponse  - ReflectedValues - TimestampTokens;
ClearResponse = OriginalRespone - ReflectedValues  -
                InconsistentContent - TimestampTokens;

EventExecResponse = getResponse(“Page1.aspx?param=value&EVENTTARGET=…”);
EventExecResponse = OriginalRespone - ReflectedValues  -
                InconsistentContent - TimestampTokens;

If (Diff (ClearResponse, EventExecResponse ) > 0) ValidControlEvent = True;
```

Events of Invisible Controls



URL: http://localhost:7011/WelcomeChanged.aspx

☑ ViewState.
☐ ViewState Signed (MAC found).
☐ ViewState Encrypted
☑ Event Validation.
☐ Event Validation Signed (MAC found

Visible:
Button5
Button6
Button7
Button8

URL: http://localhost:7011/WelcomeChanged.aspx

Prefix:

```
Trying: Text1
Trying: Text2
Trying: Text3
Trying: Text4
Trying: Text5
Trying: Text6
Trying: Text7
Trying: Text8
Trying: Text9
Trying: Text10
```

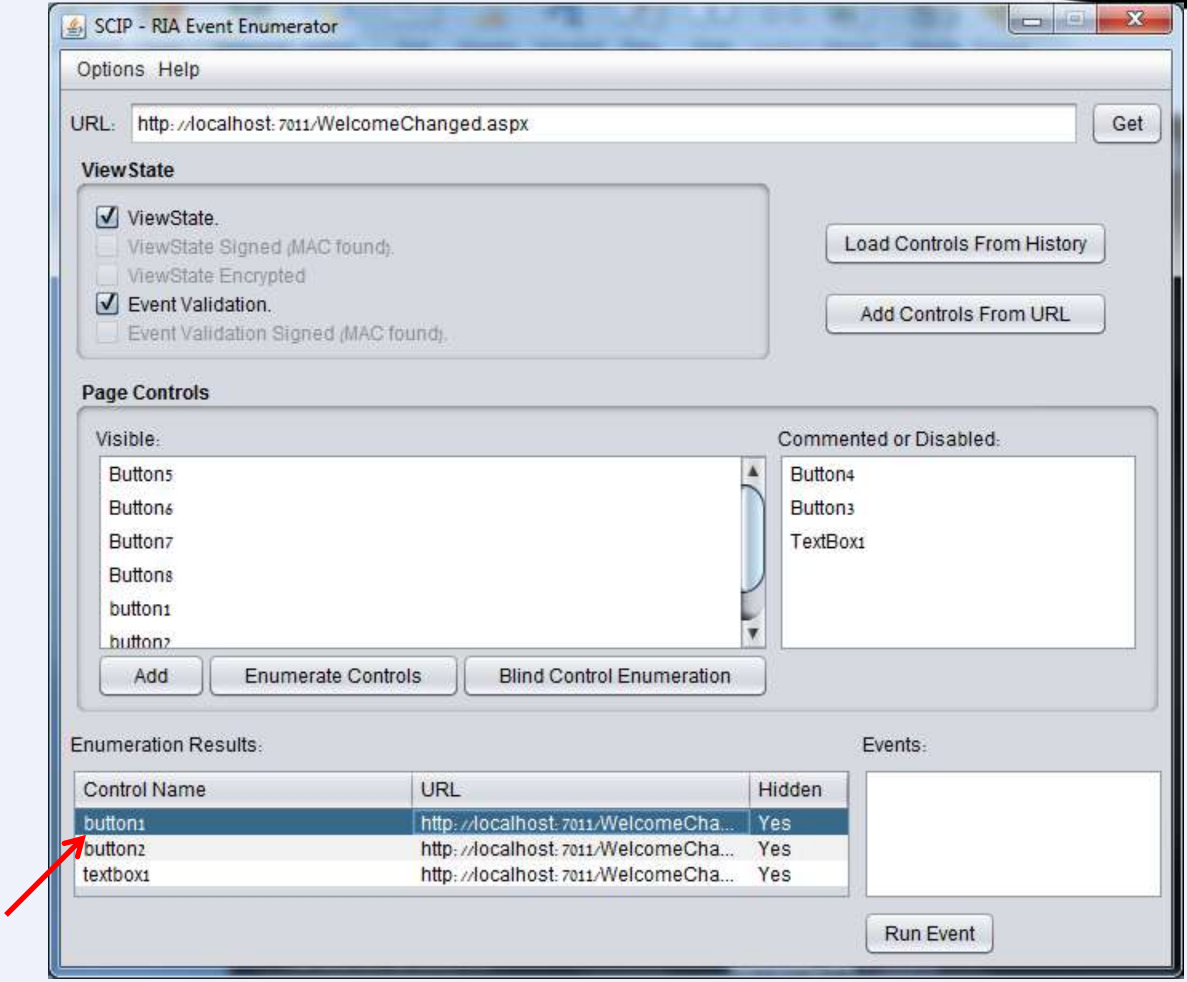
Controls Found: 3

Enumeration Results:

Control Name	URL	Hidden

onclick

Events of Invisible Controls


SCIP - RIA Event Enumerator
Options
Help
Get
Load Controls From History
Add Controls From URL
Add
Enumerate Controls
Blind Control Enumeration
Run Event

URL: http://localhost:7011/WelcomeChanged.aspx

ViewState

☑ ViewState.
□ ViewState Signed (MAC found).
□ ViewState Encrypted
☑ Event Validation.
□ Event Validation Signed (MAC found).

Page Controls

Visible:

Button5
Button6
Button7
Button8
button1
button2

Commented or Disabled:

Button4
Button3
TextBox1

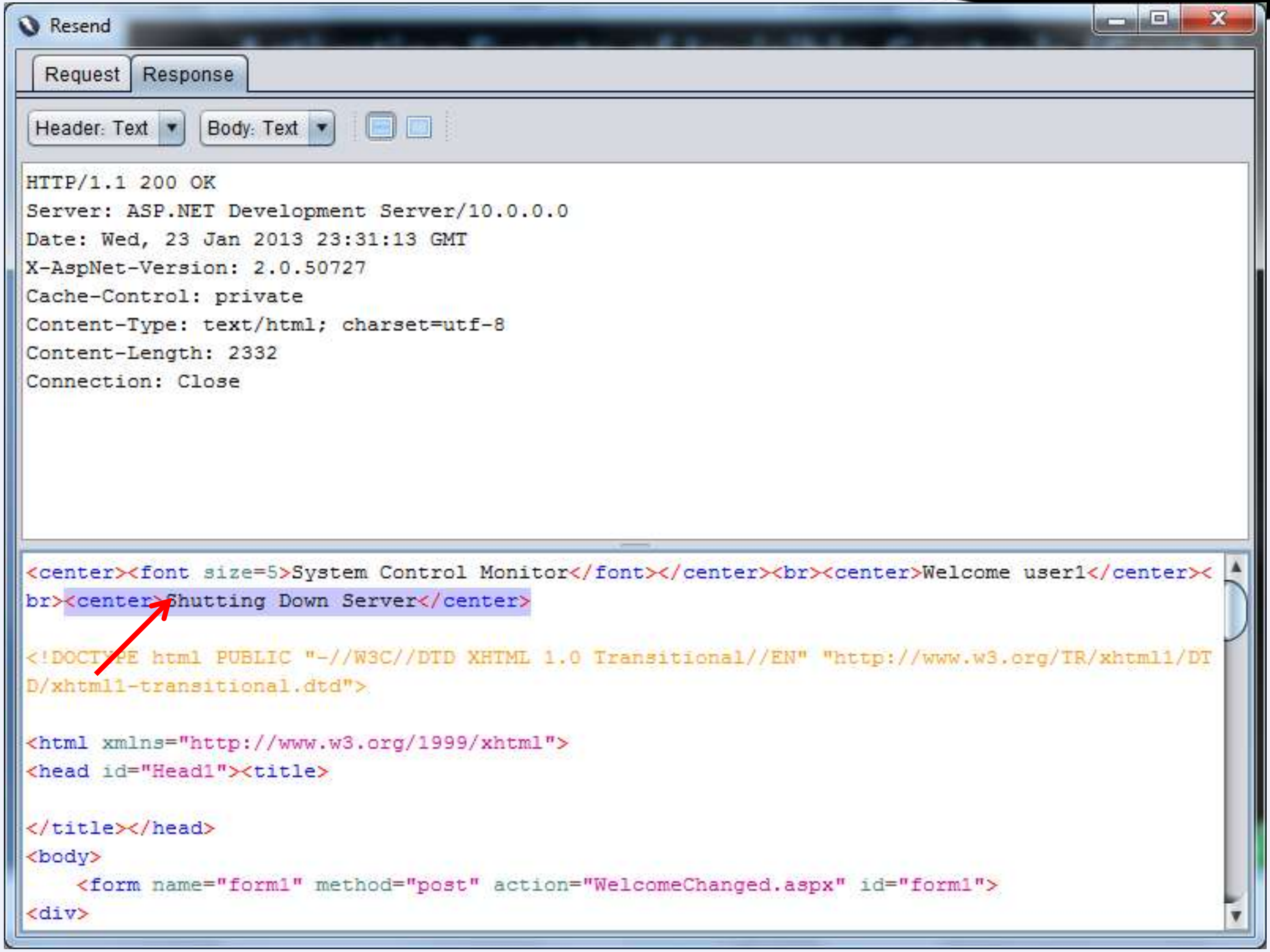
Enumeration Results:

Control Name	URL	Hidden
button1	http://localhost:7011/WelcomeCha...	Yes
button2	http://localhost:7011/WelcomeCha...	Yes
textbox1	http://localhost:7011/WelcomeCha...	Yes

Events:

Events of Invisible Controls

OWASP
The Open Web Application Security Project



```
HTTP/1.1 200 OK
Server: ASP.NET Development Server/10.0.0.0
Date: Wed, 23 Jan 2013 23:31:13 GMT
X-AspNet-Version: 2.0.50727
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Length: 2332
Connection: Close
```

```
<center><font size=5>System Control Monitor</font></center><br><center>Welcome user1</center><
br><center>Shutting Down Server</center>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DT
D/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1"><title>

</title></head>
<body>
    <form name="form1" method="post" action="WelcomeChanged.aspx" id="form1">
<div>
```

ERNST & YOUNG
Quality In Everything We Do

Master Dormant Event Activation:
Locating Hidden Optional Events

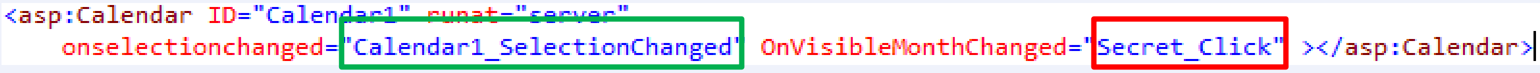
Activating Optional Events

- **Prerequisites – Multiple Dormant Events of a Single Control:**
  - By default, only a limited amount of basic controls support multiple events (not including custom controls).
  - The hidden control must be assigned with multiple valid events (example: Calendar control).
  - In addition to fuzzing a valid eventtarget, the tester can execute the "optional" events by locating/fuzzing a valid eventargument
  - Different eventargument formats can execute **different** server events (for example V[value] vs. [value])

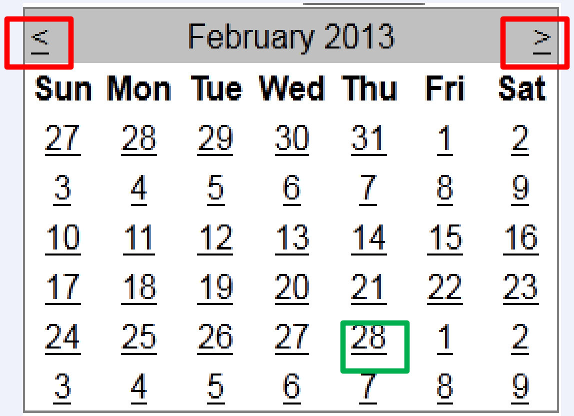
- **Advanced:** Core Events and Custom Events
  - Click, Command, onSelectionChanged, OnVisibleMonthChanged, Etc

Activating Optional Events

OWASP
The Open Web Application Security Project

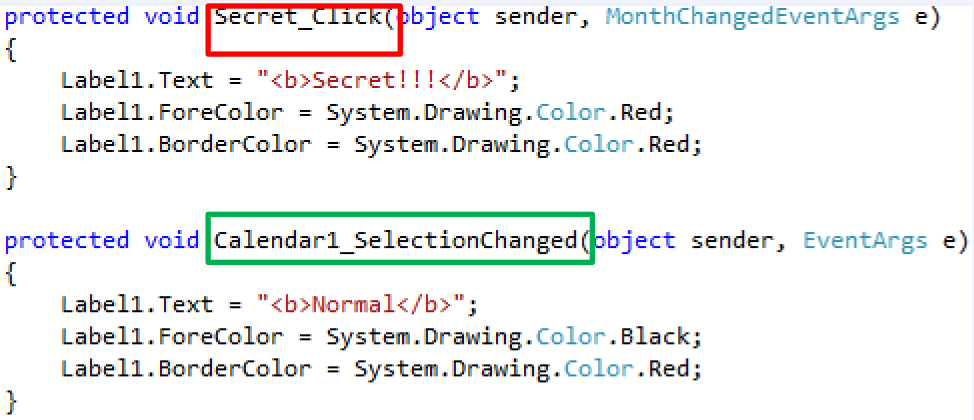


```
<asp:Calendar ID="Calendar1" runat="server"
    onselectionchanged="Calendar1_SelectionChanged" OnVisibleMonthChanged="Secret_Click" ></asp:Calendar>
```

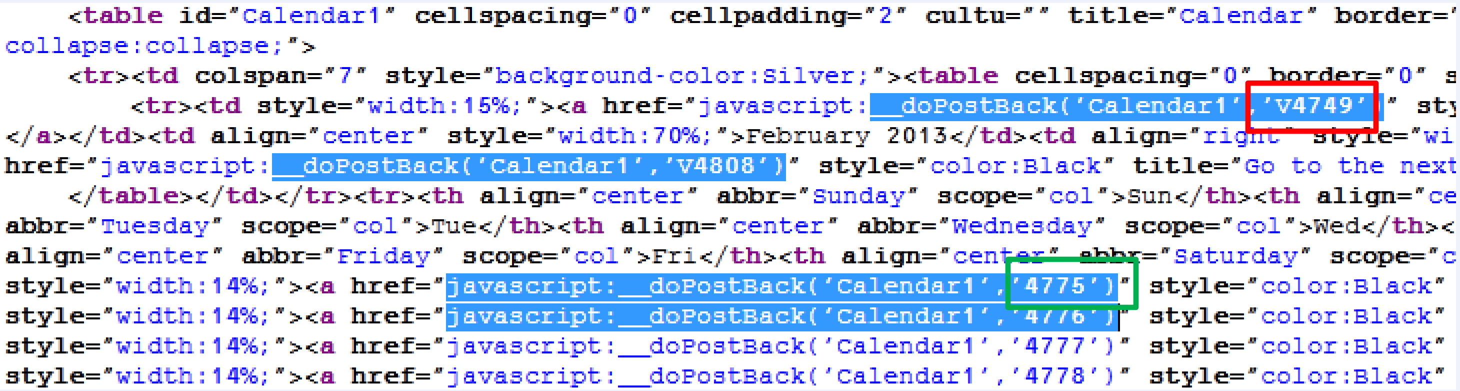


< February 2013 >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	1	2
3	4	5	6	7	8	9



```
protected void Secret_Click(object sender, MonthChangedEventArgs e)
{
    Label1.Text = "<b>Secret!!!</b>";
    Label1.ForeColor = System.Drawing.Color.Red;
    Label1.BorderColor = System.Drawing.Color.Red;
}

protected void Calendar1_SelectionChanged(object sender, EventArgs e)
{
    Label1.Text = "<b>Normal</b>";
    Label1.ForeColor = System.Drawing.Color.Black;
    Label1.BorderColor = System.Drawing.Color.Red;
}
```



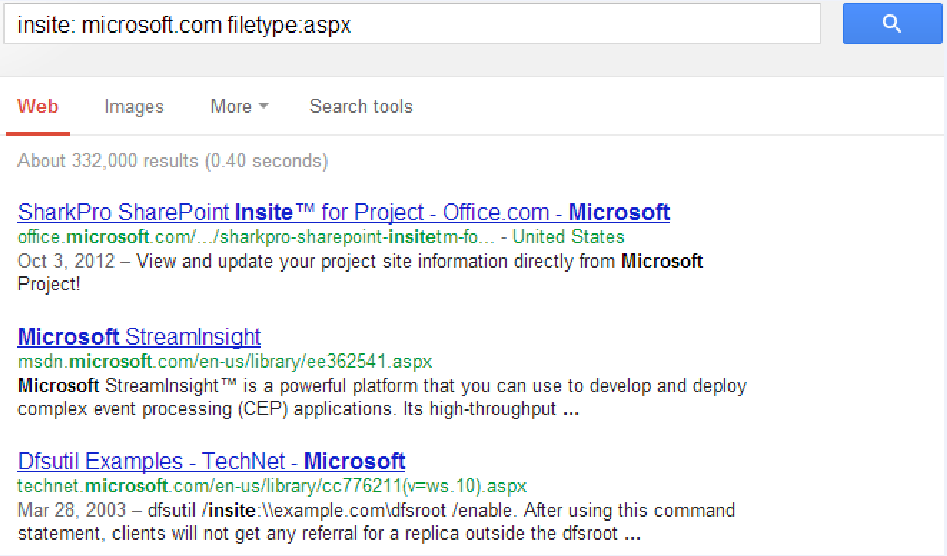
```
    <table id="Calendar1" cellspacing="0" cellpadding="2" cultu="" title="Calendar" border="
collapse:collapse;">
    <tr><td colspan="7" style="background-color:Silver;"><table cellspacing="0" border="0" s
        <tr><td style="width:15%;"><a href="javascript:__doPostBack('Calendar1','V4749')" sty
</a></td><td align="center" style="width:70%;">February 2013</td><td align="right" style="wi
href="javascript:__doPostBack('Calendar1','V4808')" style="color:Black" title="Go to the next
    </table></td></tr><tr><th align="center" abbr="Sunday" scope="col">Sun</th><th align="ce
abbr="Tuesday" scope="col">Tue</th><th align="center" abbr="Wednesday" scope="col">Wed</th><
align="center" abbr="Friday" scope="col">Fri</th><th align="center" abbr="Saturday" scope="c
style="width:14%;"><a href="javascript:__doPostBack('Calendar1','4775')" style="color:Black"
style="width:14%;"><a href="javascript:__doPostBack('Calendar1','4776')" style="color:Black"
style="width:14%;"><a href="javascript:__doPostBack('Calendar1','4777')" style="color:Black"
style="width:14%;"><a href="javascript:__doPostBack('Calendar1','4778')" style="color:Black"
```

SCIP Methods for Gurus:

Executing Events of Invisible Controls
DESPITE
Active Event Validation & Viewstate MAC

Advanced SCIP Methods

- **Prerequisites - Execute Events In Spite of Security Features:**
  - Obtain the names of server controls from cached / indexed content: (search engines, browser cache of another high privileged user, etc)
  - Reuse the **cached** VIEWSTATE, EVENTTARGET, EVENTARGUMENT and EVENTVALIDATION to executing dormant events (will work regardless of visibility or security features!)



insite: microsoft.com filetype:aspx

About 332,000 results (0.40 seconds)

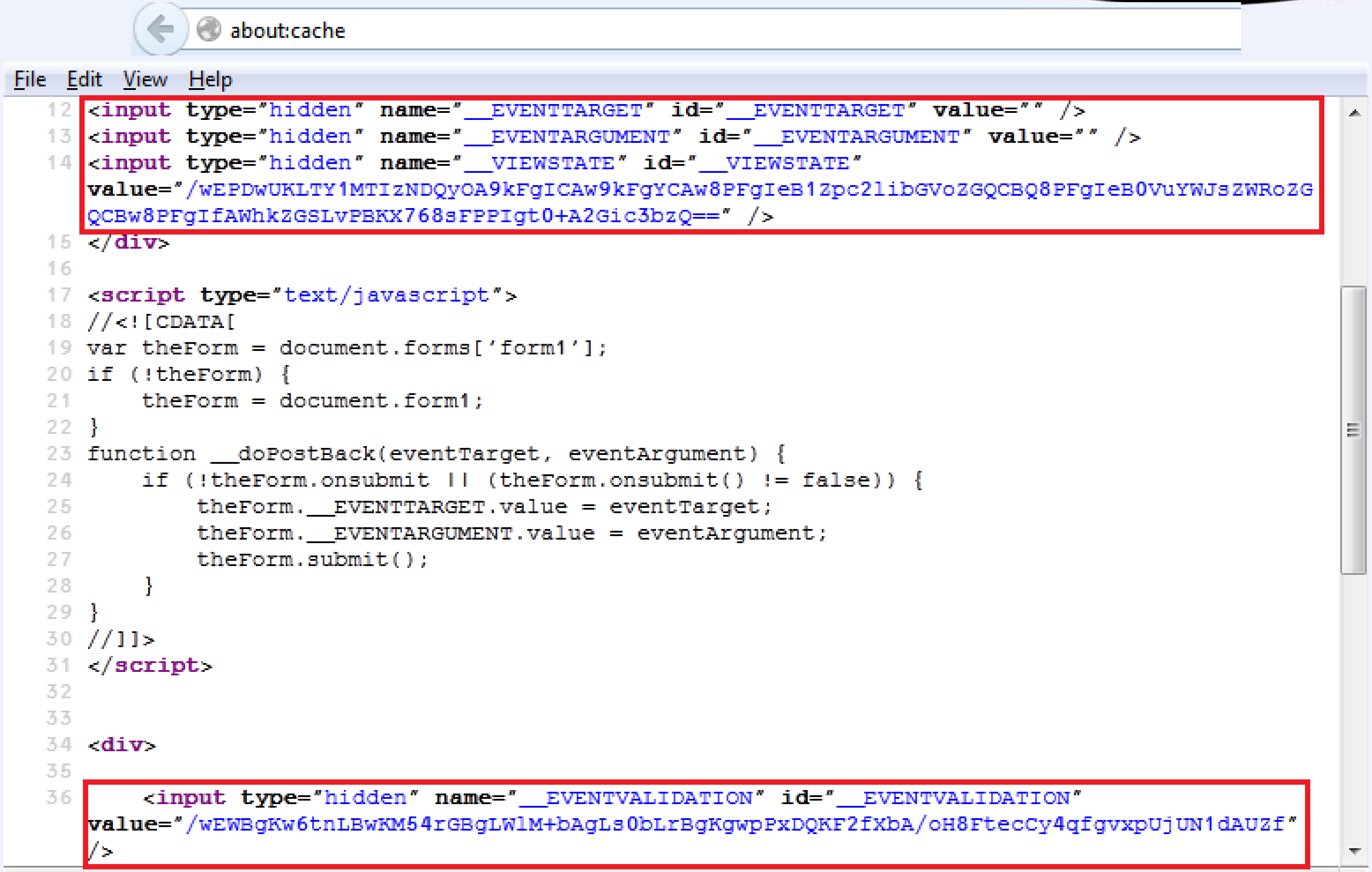
SharkPro SharePoint **Insite**™ for Project - Office.com - **Microsoft**
office.**microsoft**.com/.../sharkpro-sharepoint-**insite**tm-fo... - United States
Oct 3, 2012 – View and update your project site information directly from **Microsoft** Project!

Microsoft StreamInsight
msdn.**microsoft**.com/en-us/library/ee362541.aspx
Microsoft StreamInsight™ is a powerful platform that you can use to develop and deploy complex event processing (CEP) applications. Its high-throughput ...

Dfsutil Examples - TechNet - **Microsoft**
technet.**microsoft**.com/en-us/library/cc776211(v=ws.10).aspx
Mar 28, 2003 – dfsutil /**insite**:\\example.com\dfsroot /enable. After using this command statement, clients will not get any referral for a replica outside the dfsroot ...

Advanced SCIP Methods

OWASP
The Open Web Application Security Project



```
<input type="hidden" name="__EVENTTARGET" id="__EVENTTARGET" value="" />
<input type="hidden" name="__EVENTARGUMENT" id="__EVENTARGUMENT" value="" />
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
value="/wEPDwUKLTY1MTIzNDQyOA9kFgICAw9kFgYCAw8PFgIeB1Zpc2libGVoZGQCBQ8PFgIeB0VuYWJsZWRoZG
QCBw8PFgIfAWhkZGSLvPBKX768sFPPIgt0+A2Gic3bzQ==" />
</div>

<script type="text/javascript">
//<![CDATA[
var theForm = document.forms['form1'];
if (!theForm) {
    theForm = document.form1;
}
function __doPostBack(eventTarget, eventArgument) {
    if (!theForm.onsubmit || (theForm.onsubmit() != false)) {
        theForm.__EVENTTARGET.value = eventTarget;
        theForm.__EVENTARGUMENT.value = eventArgument;
        theForm.submit();
    }
}
//]]>
</script>


<div>

    <input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION"
value="/wEWBgKw6tnLBwKM54rGBgLWlM+bAgLs0bLrBgKgwpPxDQKF2fXbA/oH8FtecCy4qfgvxpUjUN1dAUZf"
/>
```

Advanced SCIP Methods

- Reusing Obsolete Cached / Indexed State Flags
  - Reusing the state and validation of indexed/cached versions page might work even if the control structure **changed (!)**
  - Controls, State and validation flag must origin from the same page (so the signature will be effective)
  - Controls must be included/include the controls of the page
- Signed Content Scraping Using Web Attacks
  - XSS, Clickjacking, Etc

Advanced SCIP Methods

- Shared Hosting Attack Model
  - Can bypass Viewstate MAC and EventValidation
  - Scenarios for Shared Application Pool
  - Scenarios for Isolated Application Pool

Risk Mitigation

Protecting Dormant Events

SCIP Mitigation

OWASP
The Open Web Application Security Project

- **Do NOT** use the **Disabled** property for security purposes
- **Do NOT** rely on HTML comments to hide controls
- **Remove** unnecessary dormant events from all layers: HTML, Design (e.g. aspx), CodeBehind (e.g. aspx.cs)
- **Implement** code-level privilege validation in each event
- **Enforce** digital signatures (Viewstate **MAC**)
- **Activate** event validation mechanisms (EventValidation)
- **Disable** cache / **Prevent** indexing in pages with sensitive controls!
- **Customize** the platform error messages

Privilege / Event Validation

- Explicit Privilege Validation in Event Code

```
protected void Button1_Click(object sender, EventArgs e)
{
   if (((String)Session["user"]).Equals("admin"))
   {
      ...
   }
}
```

- Enable Event Validation / MAC

```
<%@ Page Language="C#" AutoEventWireup="true"
EnableEventValidation="true" EnableViewStateMac="true“ …%>
```

Disable Cache in ASP.net

- Disable Browser/Proxy Cache (Sample Code)

```
HttpContext.Current.Response.Cache.SetExpires(DateTime.UtcNow.AddDays(-1));
HttpContext.Current.Response.Cache.SetValidUntilExpires(false);
HttpContext.Current.Response.Cache.SetRevalidation(HttpCacheRevalidation.AllCaches);
HttpContext.Current.Response.Cache.SetCacheability(HttpCacheability.NoCache);
HttpContext.Current.Response.Cache.SetNoStore();
```

- Restrict SE access in robots.txt (Sample Config)
  - http://www.robotstxt.org/robotstxt.html

```
User-agent: *
Disallow: /
```

- Restrict SE caching/crawling via meta tags
  - http://www.robotstxt.org/meta.html

The Original Theory

Research Leads and Progress

The Original Theory

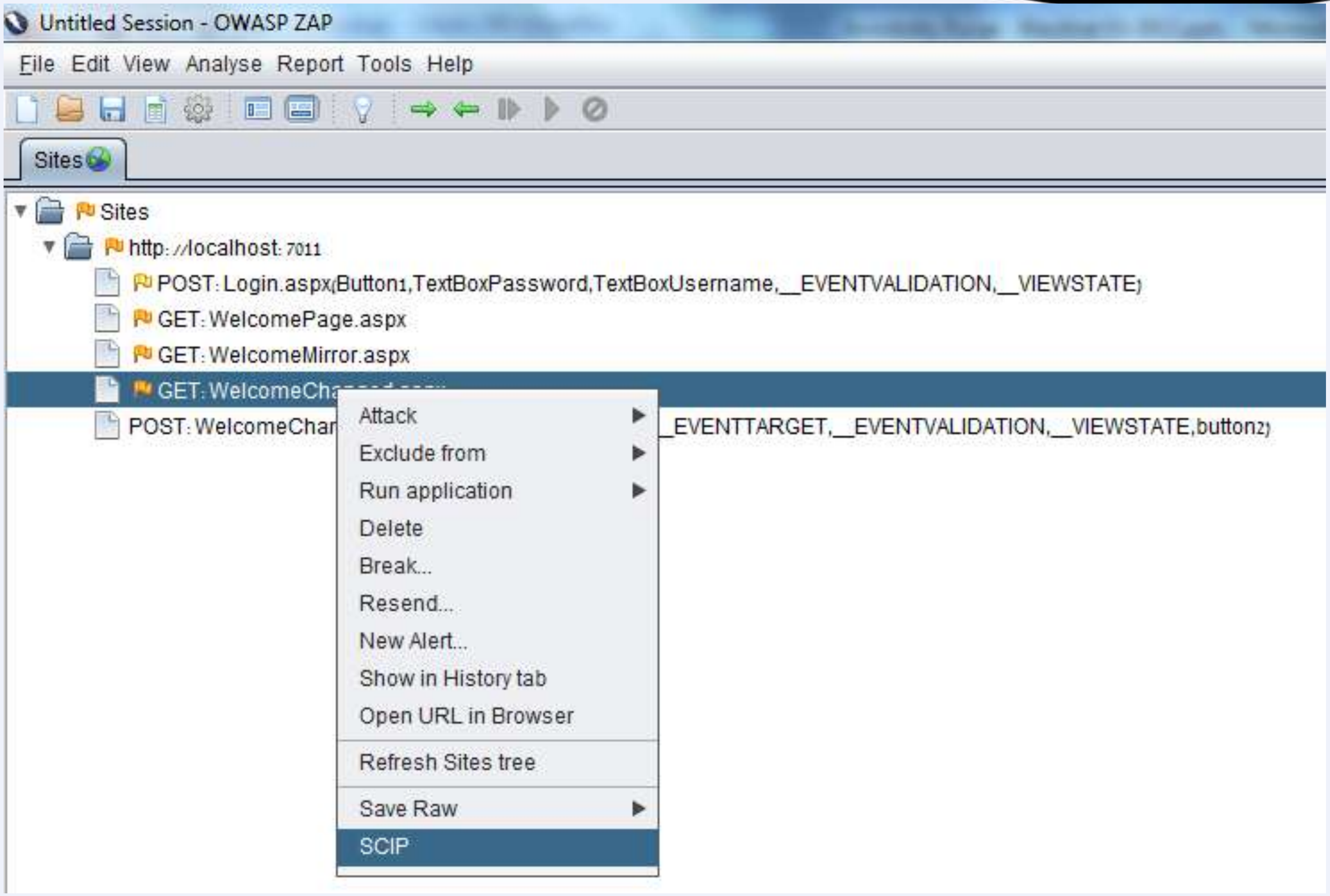
- Reuse the viewstate / eventvalidation fields of other pages
  - Pages with similar controls
  - Pages with identical controls
- EventValidation responding differently to manipulations on various control types
- Reuse a partial or included cached viewstate / eventvalidation fields
- Different behaviors for different ASP.Net versions (v1.1, v2.0,v3.5, v4.0…) and Mono versions

Summary
Enumerating Hidden Controls and Events

Diviner, SCIP and ZAP

- **Diviner**
  - OWASP ZAP extension (v1.4+/v2.0+)
  - Requires ZAP to run with Java 1.7+
  - Homepage: http://code.google.com/p/diviner/
- **SCIP**
  - OWASP ZAP extension (v2.0+), currently focused at **ASP.net**
  - **Features:** disabled/commented control event execution, error-based detection of invisible controls, manual execution of target events, parameter tampering in-spite of event validation (when MAC is off)
  - **Upcoming features:** cache scraping and analysis, reuse obsolete event-validation fields, blind event enumeration
  - Requires **Diviner** diff methods to support Blind Control Enumeration
  - Homepage: http://code.google.com/p/ria-scip/

Activating SCIP in ZAP



Summary & Conclusions

- Potential Dormant Events:
  - Events of Disabled Controls (ASP.Net: .enabled=false)
  - Events of Invisible Controls (ASP.Net: .visible=false)
  - Events of HTML Commented Controls (aspx: <!-- … -->)
  - Hidden Alternate Events of Core/Custom Controls

Summary & Conclusions

- **Prerequisites for Event Execution Methods:**
  - Events of Disabled /Commented Controls - None!
  - Events of Invisible Controls - the EventValidation OR Viewstate MAC must be turned off; can occur per machine, application, page or control
  - Hidden Alternate Events of Core/Custom Controls
- **Advanced Event Execution Methods:**
  - Execute any control event, regardless of viewstate MAC or event validation, by reusing cached values of viewstate, eventtarget, eventargument and eventvalidation fields
  - State fields must include the control’s digitally signed content

And Finally…

Additional Resources

OWASP
The Open Web Application Security Project

- SCIP Homepage (ZAP 2.0+ Extension)
  - http://code.google.com/p/ria-scip/
- Diviner Homepage (ZAP 1.4+/2.0+ Extension)
  - http://code.google.com/p/diviner/
- OWASP ZAP Proxy
  - http://code.google.com/p/zaproxy/
- **Great** posts on the subject by James Jardine
  - http://www.jardinesoftware.net/

Ernst & Young ASCs

OWASP
The Open Web Application Security Project

- Americas
  - Hacktics IL
  - Houston
  - New York
  - Buenos Aires
- EMEIA
  - Dublin
  - Barcelona
- Asia Pacific
  - Singapore
  - Melbourne

ASC
ASC
ASC
ASC
ASC
ASC
ASC

Ernst & Young
Quality In Everything We Do

About Ernst & Young

Ernst & Young

Assurance | Tax | Transactions | Advisory

About Ernst & Young
Ernst & Young is a global leader in assurance, tax, transaction and advisory services. Worldwide, our 130,000 people are united by our shared values and an unwavering commitment to quality. We make a difference by helping our people, our clients and our wider communities achieve potential.

About Ernst & Young's Technology Risk and Security Services
Information technology is one of the key enablers for modern organizations to compete. It gives the opportunity to get closer, more focused and faster in responding to customers, and can redefine both the effectiveness and efficiency of operations. But as opportunity grows, so does risk. Effective information technology risk management helps you to improve the competitive advantage of your information technology operations, to make these operations more cost efficient and to manage down the risks related to running your systems. Our 6,000 information technology risk professionals draw on extensive personal experience to give you fresh perspectives and open, objective advice – wherever you are in the world. We work with you to develop an integrated, holistic approach to your information technology risk or to deal with a specific risk and security issue. And because we understand that, to achieve your potential, you need a tailored service as much as consistent methodologies, we work to give you the benefit of our broad sector experience, our deep subject matter knowledge and the latest insights from our work worldwide. It's how Ernst & Young makes a difference.

For more information, please visit www.ey.com.

Ernst & Young refers to the global organization of member firms of Ernst & Young Global Limited, each of which is a separate legal entity. Ernst & Young Global Limited, a UK company limited by guarantee, does not provide services to clients.

Ernst & Young
Quality In Everything We Do

QUESTIONS?

Shay Chen (https://twitter.com/sectooladdict)
Niv Sela (https://twitter.com/nivselatwit)
Alex Mor (https://twitter.com/nashcontrol)