INVISIBILITY PURGE

UNMASKING DORMANT EVENTS OF INVISIBLE SERVER-SIDE WEB CONTROLS

ADVANCED HACKING OF ASP.NET, MONO AND RIA

Shay Chen
Senior Manager, Hacktics CTO
Hacktics ASC, Ernst & Young
March 2013

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).

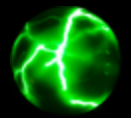
Introducing

SCIP!

Server Control Invisibility Purge

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

Diviner/ZAP Extension Implementation by **Alex Mor**.



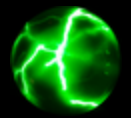
A project used for...

EodSec

Execution of Dormant Server Events & Controls

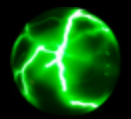
ERNST & YOUNG
Quality In Everything We Do

black hat EU 2013



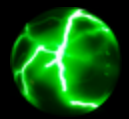
EodSec Exploitation Scenarios

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



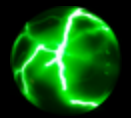
Agenda

- 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



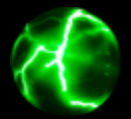
The Attack Surface of RIA

Facing the Horde of Security Features



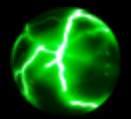
Security Features in ASP.net/Mono/RIA

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



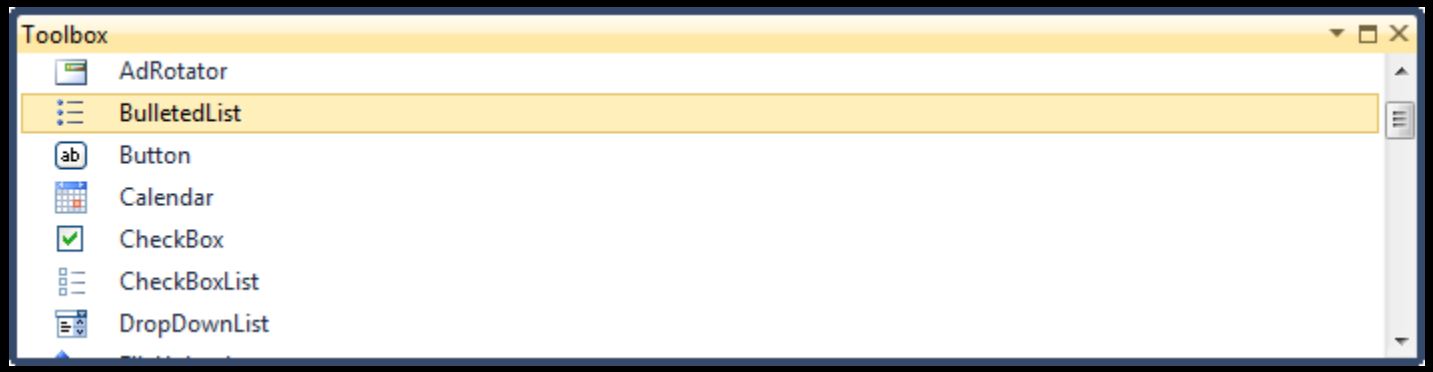
Identifying the Attack Surface

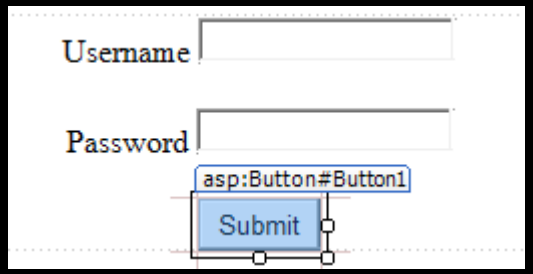
- **Purpose:** Locating Code that can be Abused
  - Web Pages
  - Web Service Methods
  - Global Modules (Filters, Handlers, Etc)
  - …
  - *Events of Web Application Server Controls*



What are Web Application Server Controls?

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





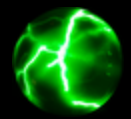
What is a Server Control Event?

- 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";
    }
}
```

What is a Server Control Event? (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>
```

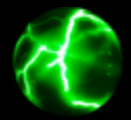
```
<input type="button" name="Button1" value="View Service Status" onclick="javascript:__doPostBack('Button1','')"
```

```
<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

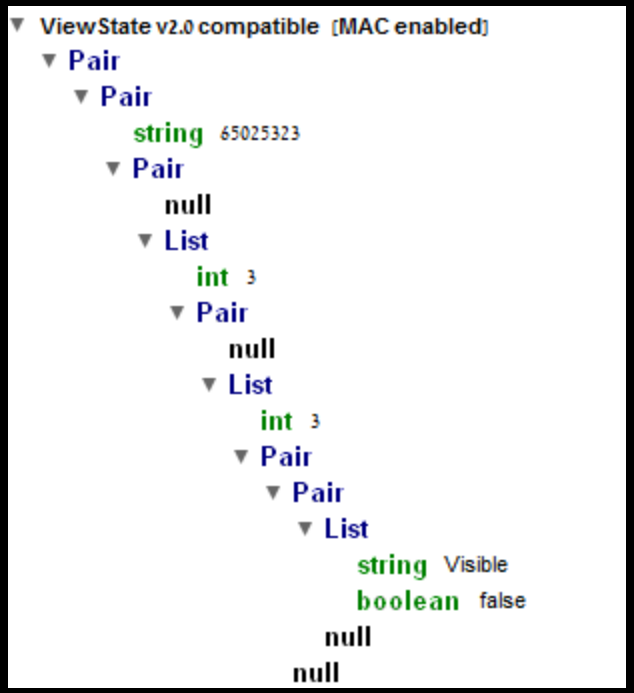
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WelcomeChanged.aspx.cs"
        EnableEventValidation="true" EnableViewStateMac="true" Inherits="ViewStateControls.WelcomeChanged" %>

<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
/l6KxF3ZQYvulQC0yedGi1oA7" />

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

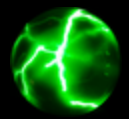
Viewstate Structure

- **Viewstate Structure**

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



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

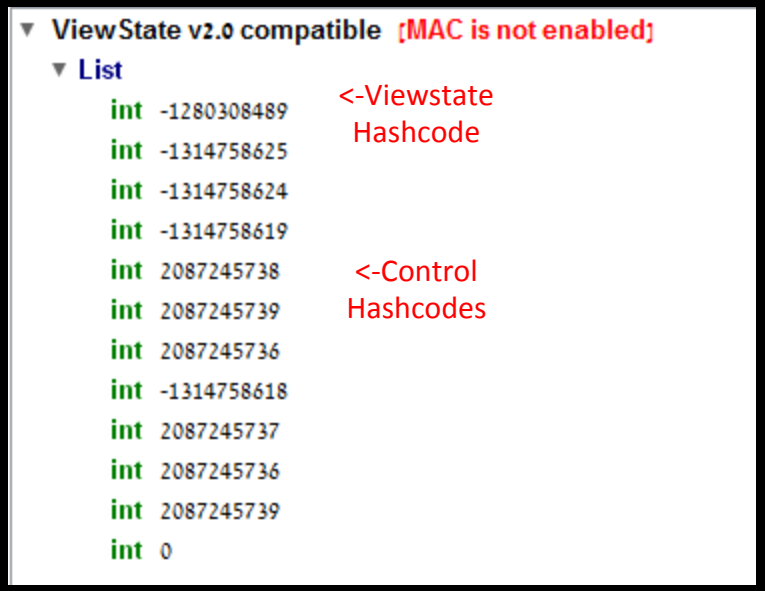


Event Validation Mechanism

- **Name/Value HashCode Formula**

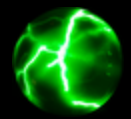
```
if ([ControlValue] == null)
   return GetStringHashCode([ControlName]);
else
   return GetStringHashCode([ControlName]) ^ GetStringHashCode([ControlValue]);
```

EventValidation (Viewed via Burp Viewstate Decoder):



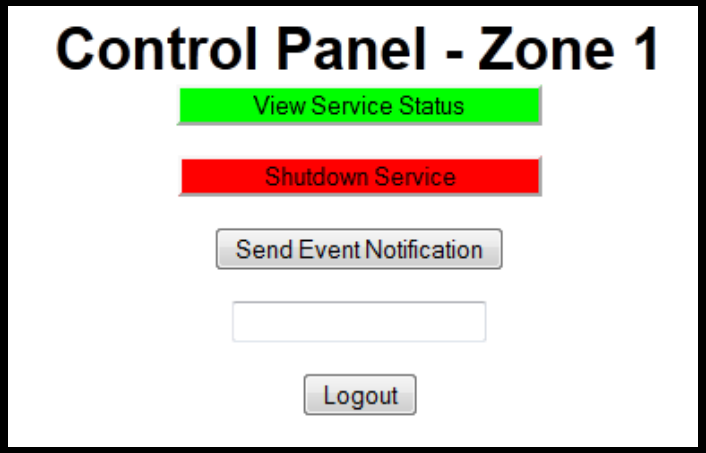
- MachineKey and MAC
- Control Name/Value Verification, Prior to Event Execution
- Include viewstate hashcode
- **Included in the HTML:**

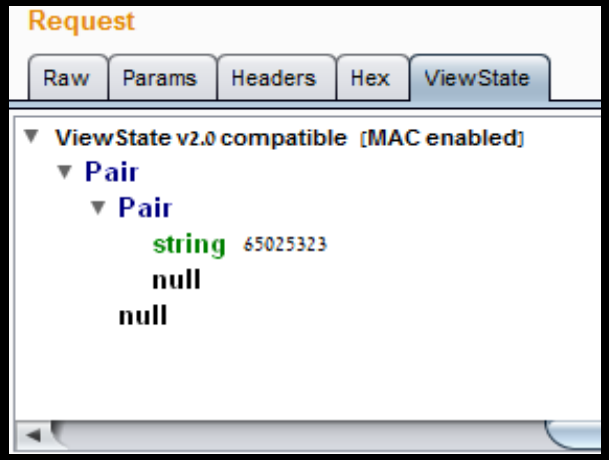
```
<input type="hidden" name="__EVENTVALIDATION"
value="/wEWCAK+2oTRBwLWlM+bAgLs0bLrBgKgwpPxDQKF2f
/16KxF3zQYvulQC0yedGi1oA7" />
```

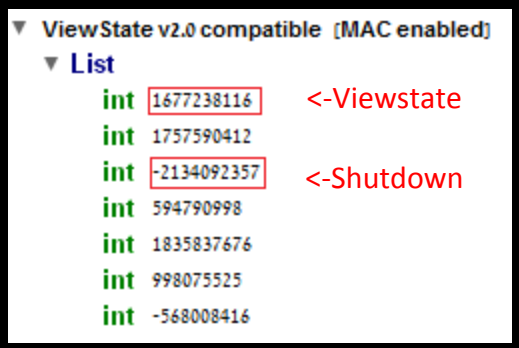
Evidence of Hidden Controls

Visible / Enabled Controls:

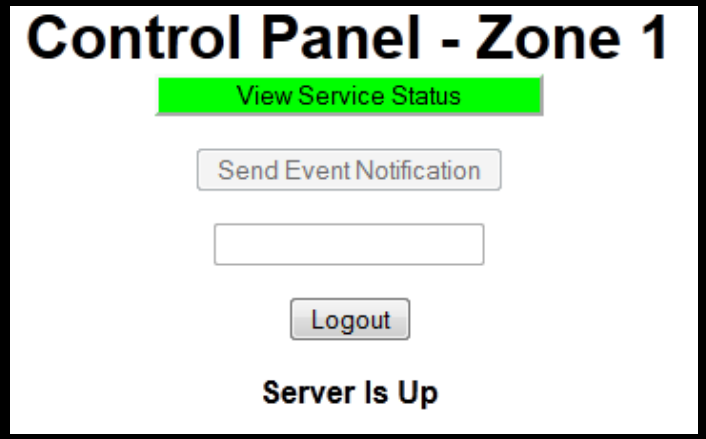


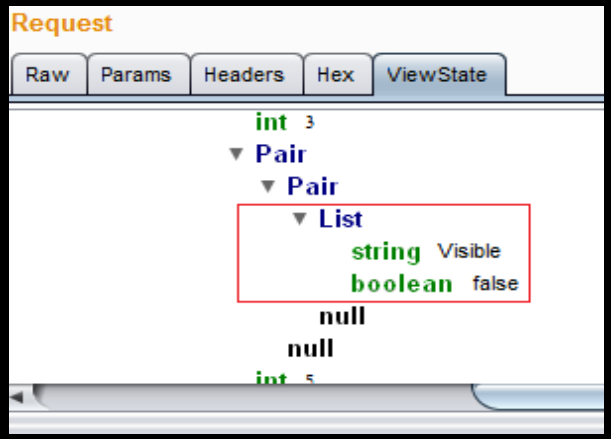


EventValidation (Viewstate Decoder):

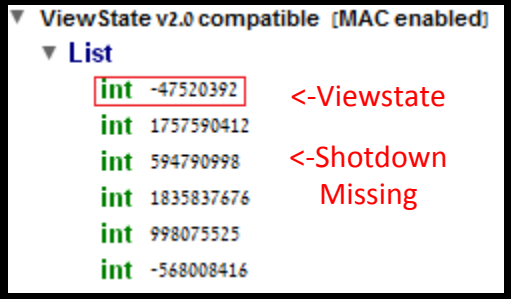


Invisible / Disabled Controls (Control Trace in Viewstate!):

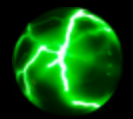




EventValidation (Viewstate Decoder):



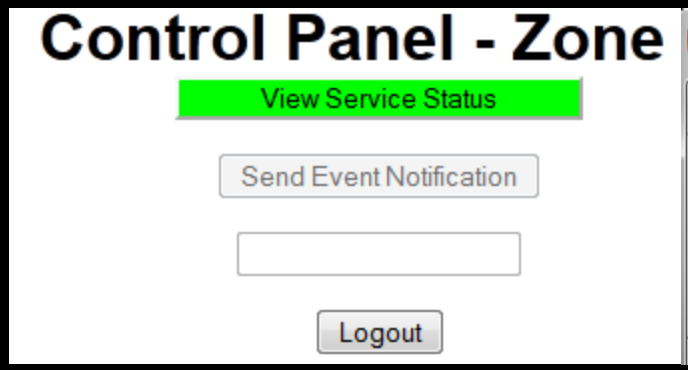
Invisible Web Controls: Archetypes

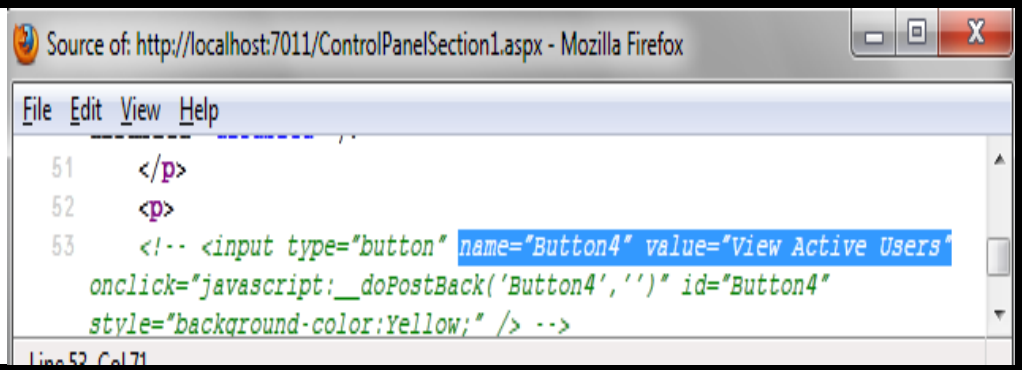


Dormant Events of Web Controls, 1 of 3

- **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>");
```

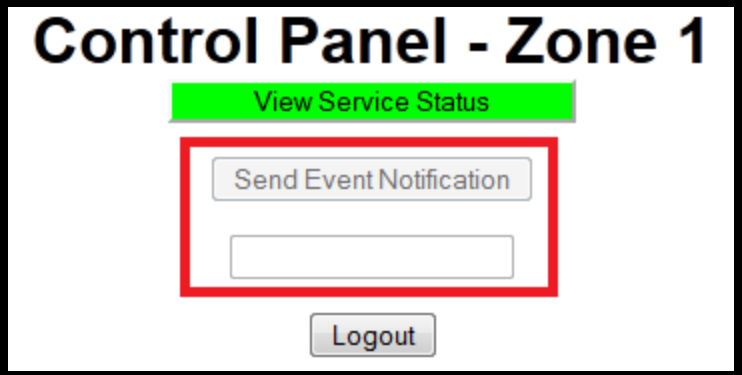




Dormant Events of Web Controls, 2 of 3

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

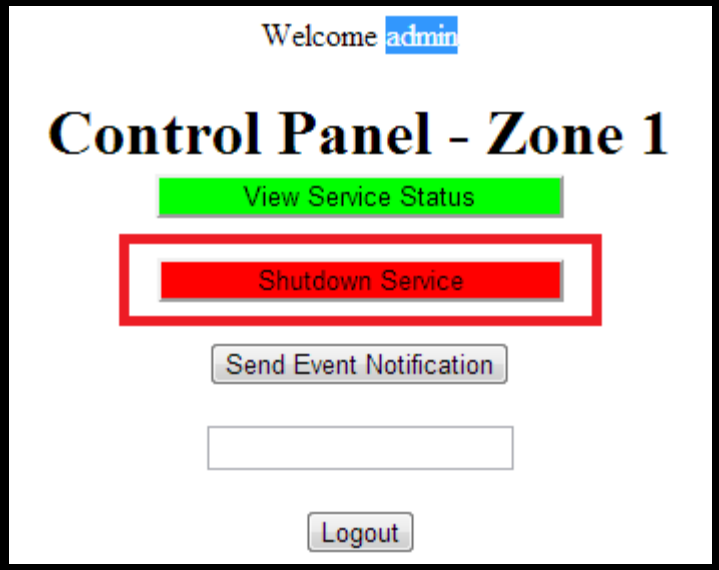
```
Send Event Notification    Button3.Enabled = false;
<input type="button" name="Button3" value="Send Event Notification" id="Button3" disabled="disabled" />
```

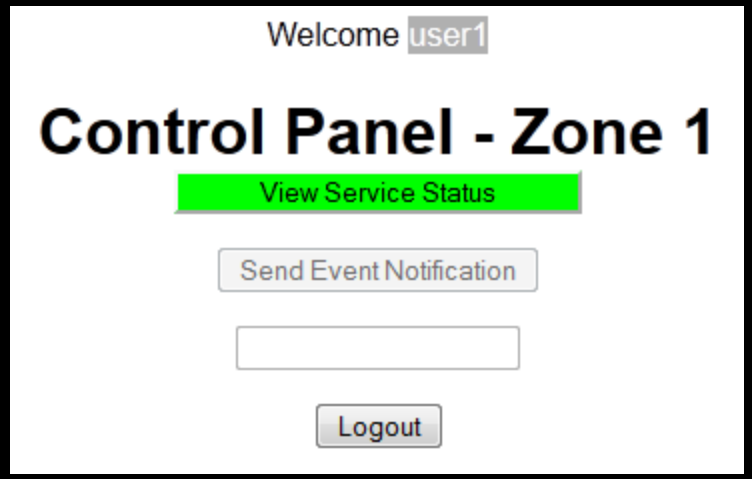


Dormant Events of Web Controls, 3 of 3

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

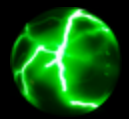
```
Button2.Visible = false;
```





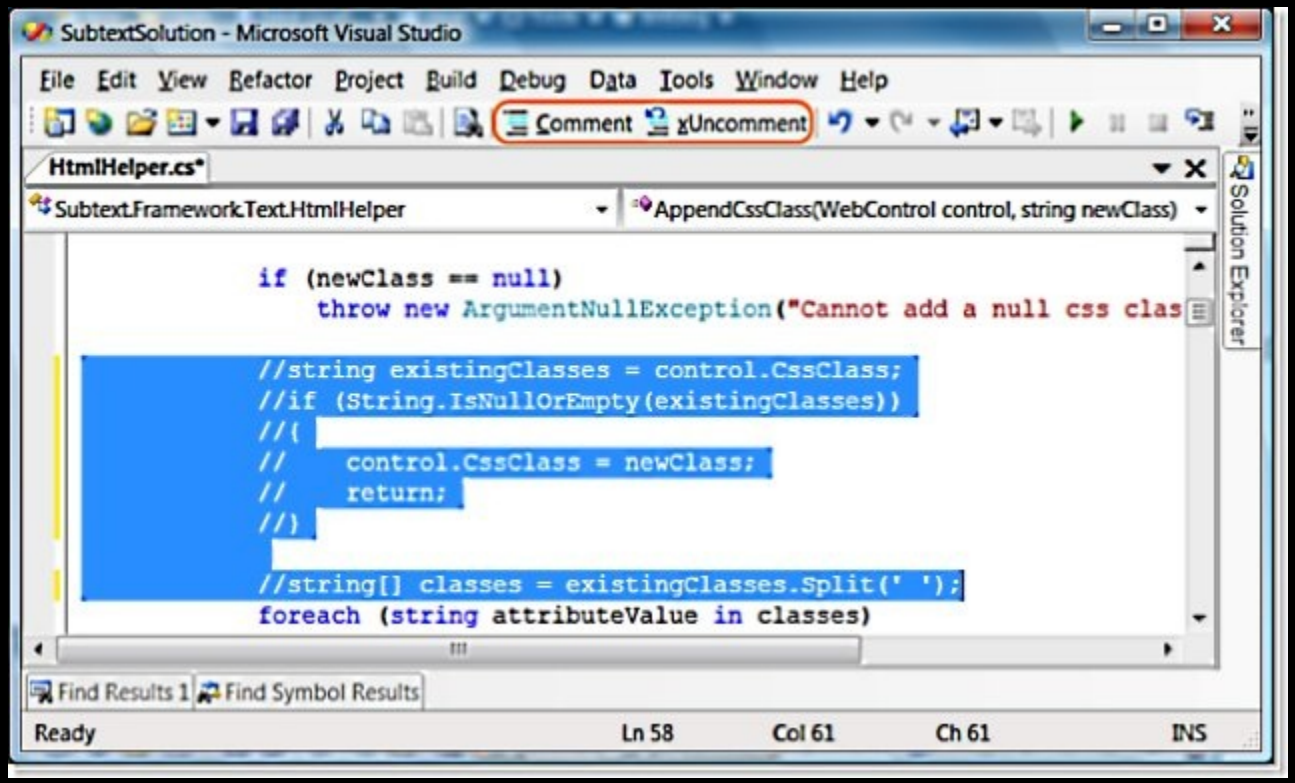
Dormant Events of Web Controls, Opt.

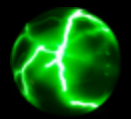
- **Dormant Events of Visible Controls**
  - Optional event listeners registered in the code level, after the optional definition was added to a control with at least one active event.



“Uncomment” Controls

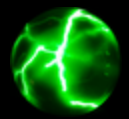
SKILL LEVEL: I'M TOO YOUNG TO DIE



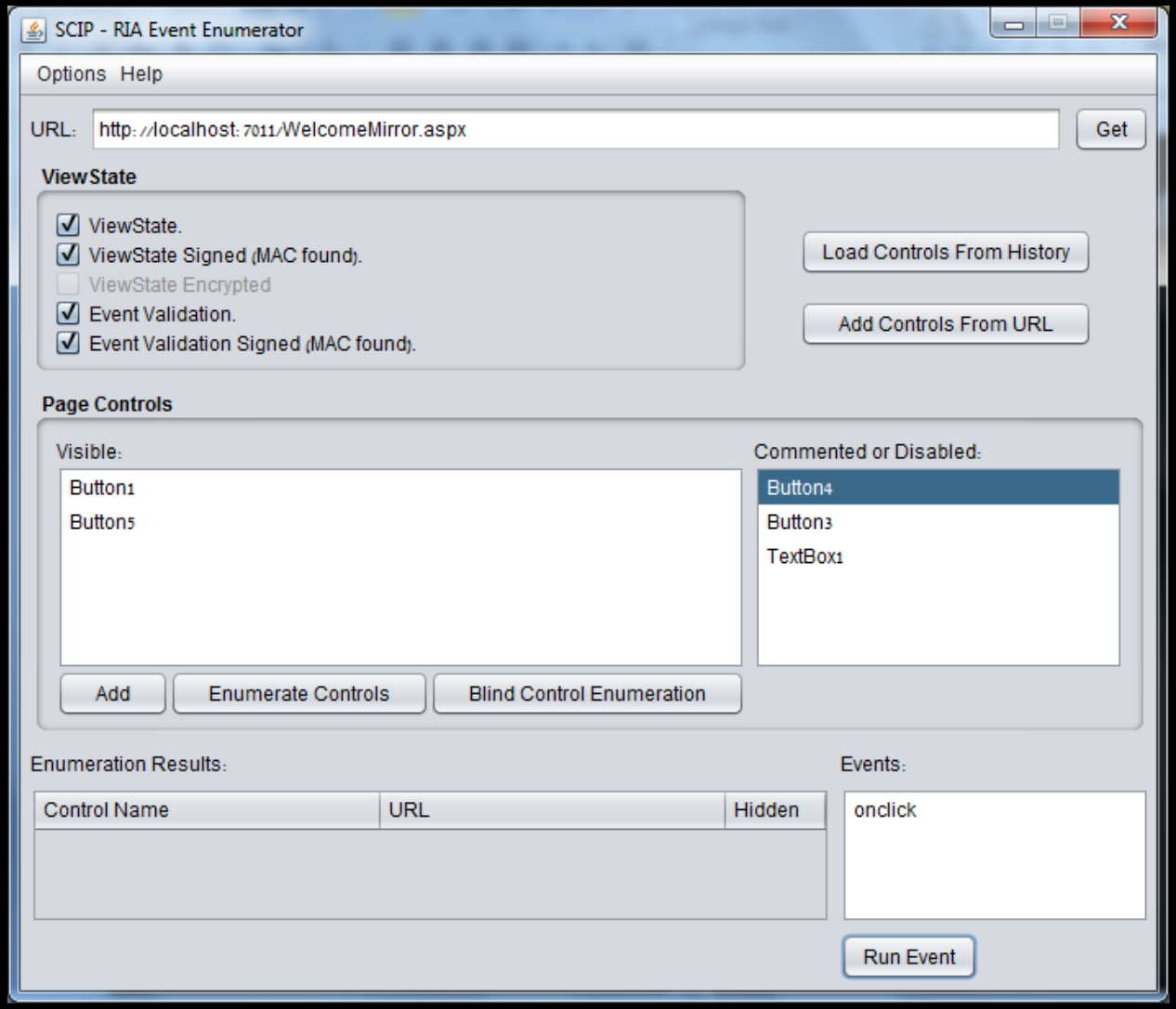


Activating Events of Commented Controls

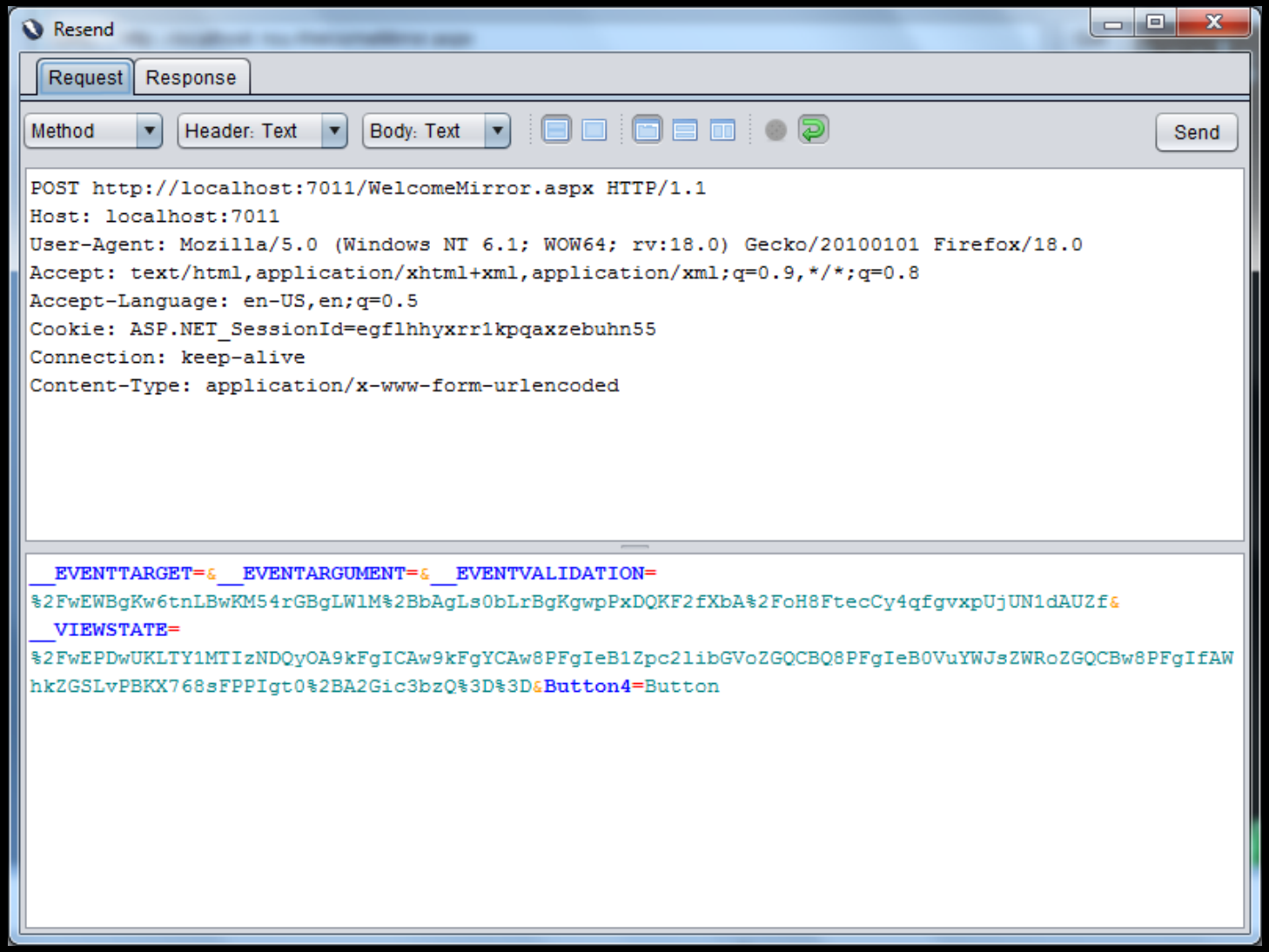
- **Prerequisites (ASP.Net / Mono) - Commented Out Controls:**
  - The developer should rely solely on the fact that the control is commented.
  - The attacker can simply “uncomment” the HTML control and execute the embedded event, or send the appropriate values directly.
- **Advantages**
  - Exploit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

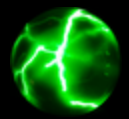


Activating Events of Commented Controls (Cont.)



Activating Events of Commented Controls (Cont.)

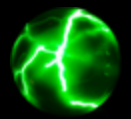




Disabled Purge

Activating 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**.

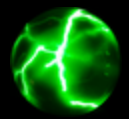


Activating Events of Disabled Controls (Cont.)

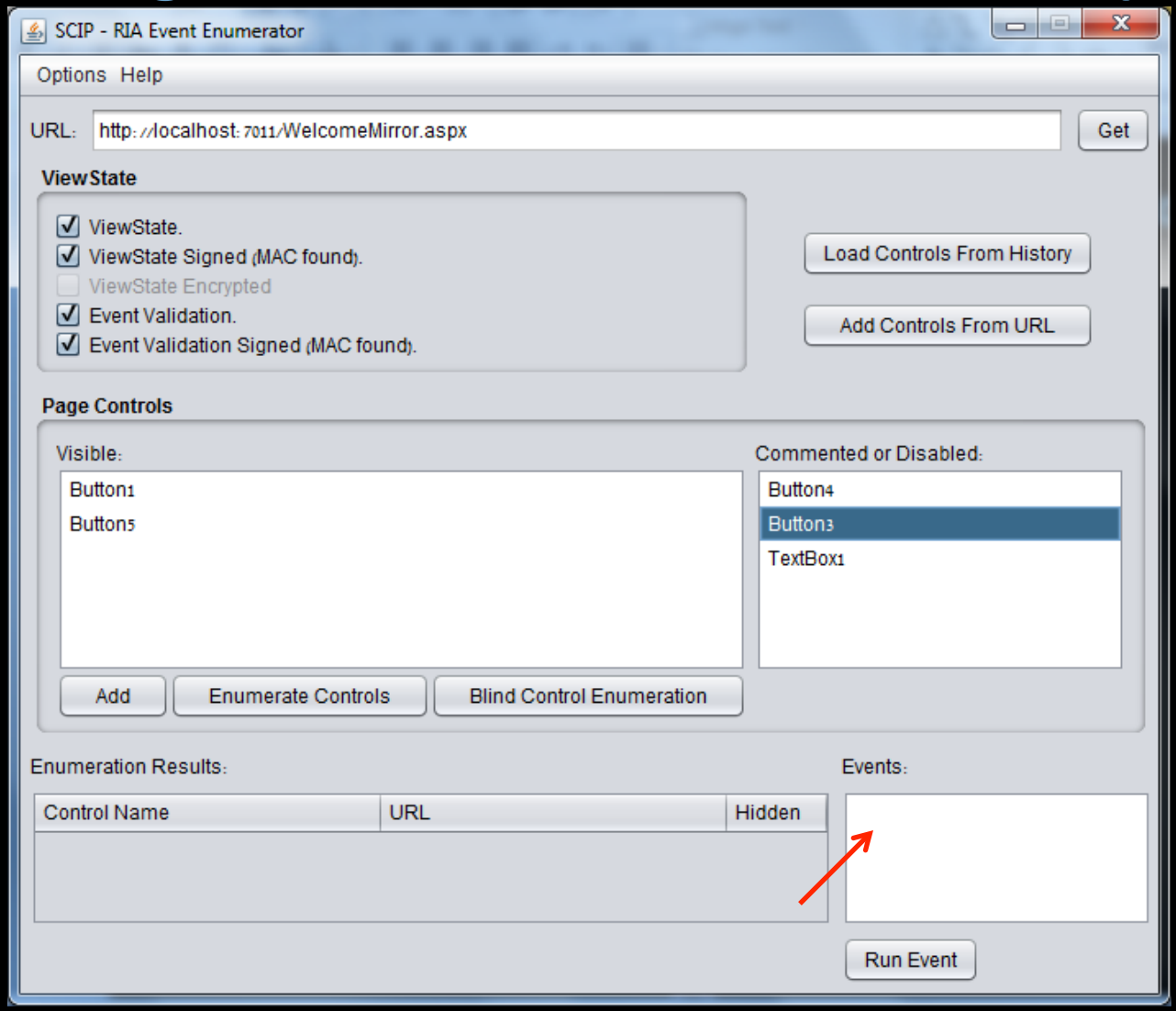
- **The Process of Forging a PostBack / CallBack Method**
  - Why does it work?
    - Using temporarily disabled controls in ASP.Net is **a feature**
    - Controls might be disabled without any relation to security, and thus, are currently not protected like invisible controls
  - How does it work?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



Activating Events of Disabled Controls (Cont.)



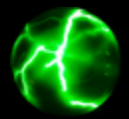
Invisibility Purge!

Activating 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>
```

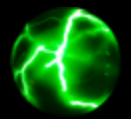
  - (II) The developer should rely solely on the control invisibility for protecting the invisible control events.



Activating Events of Invisible Controls (Cont.)

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

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

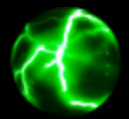



Activating Events of Invisible Controls (Cont.)

- **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" …%>
```

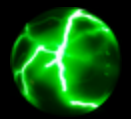
– In all cases, we still need to obtain the control / event name…



Activating Events of Invisible Controls (Cont.)

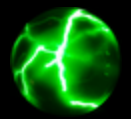
- **The Process of Server Control Enumeration**
  - In this scenario, the control leaves no client-side traces:
    - Control Name Fuzzing
    - Core Controls vs. Custom Controls

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



Common Control Naming Conventions

- **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 Control 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.

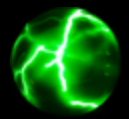
Exception Details: System.ArgumentException: Invalid postback or callback argument. Event validation is enabled using <pages enableEventValidation="true"/> in configuration or <%@ Page EnableEventValidation="true" %> in a page. For security purposes, this feature verifies that arguments to postback or 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:

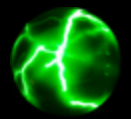
```
[ArgumentException: Invalid postback or callback argument.  Event validation is enabled using <pages enableEventValidation="true"/> in
   System.Web.UI.ClientScriptManager.ValidateEvent(String uniqueId, String argument) +8644649
   System.Web.UI.Control.ValidateEvent(String uniqueID, String eventArgument) +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
```

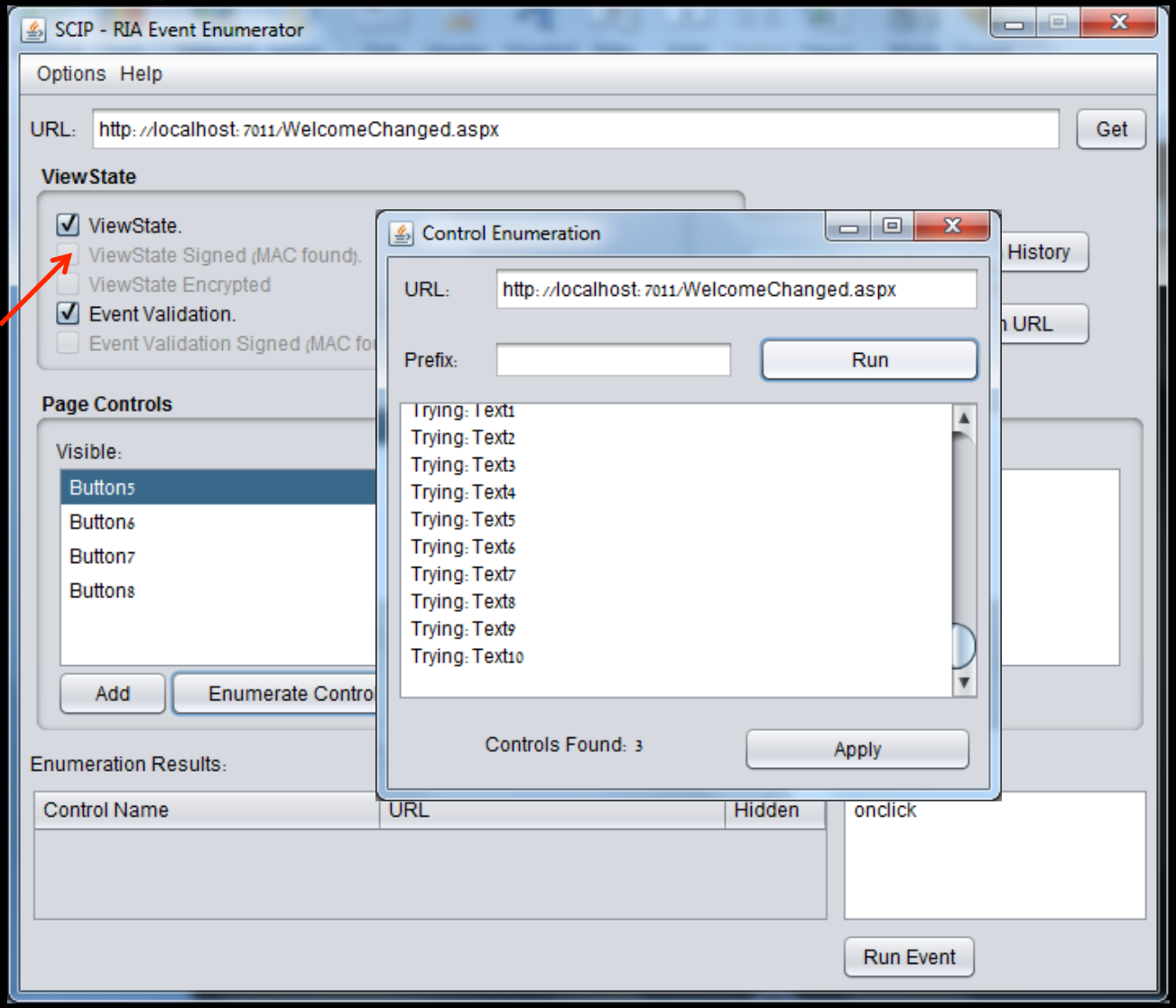
Blind Control Enumeration

- Basic Blind Differentiation Formula:

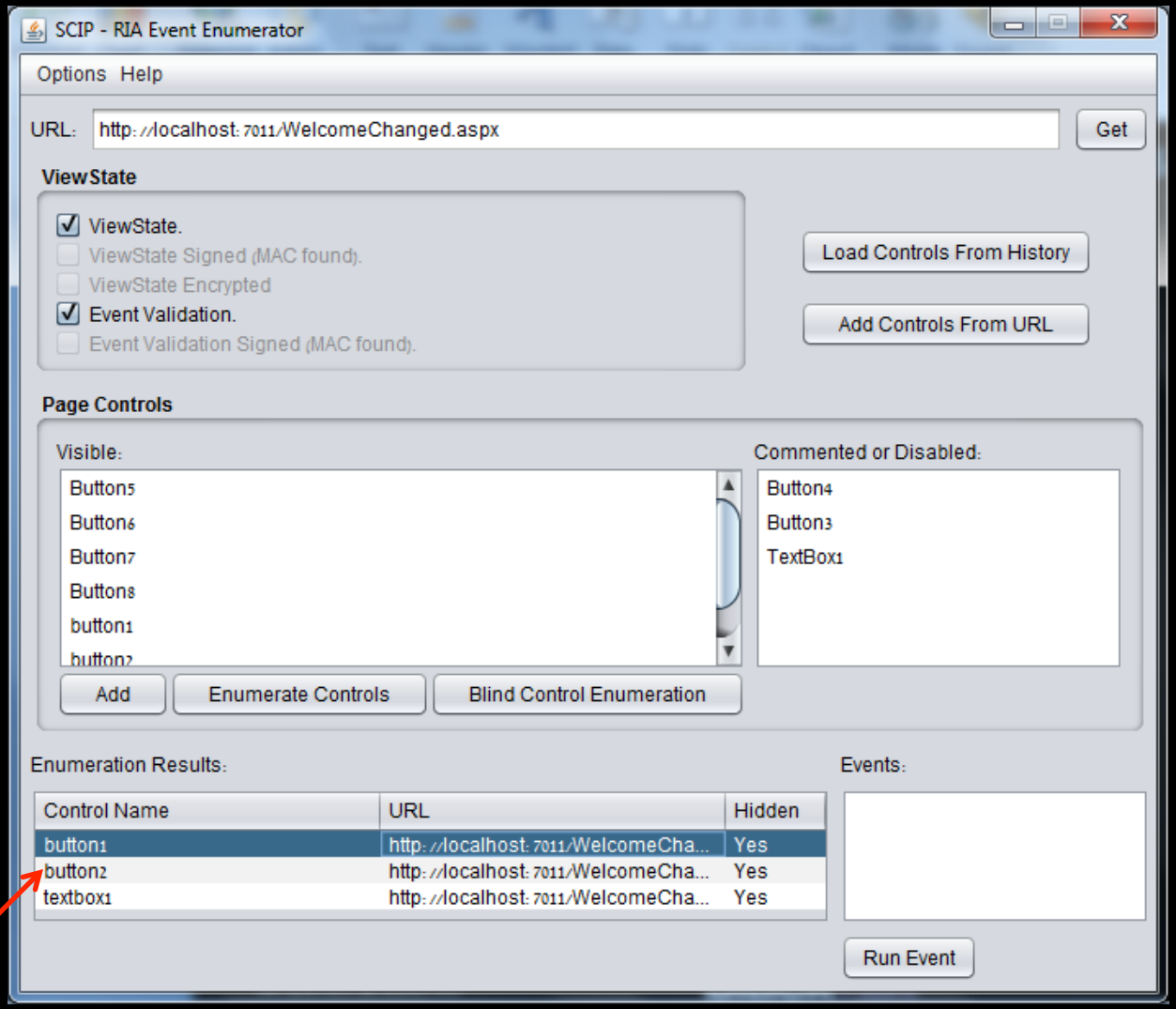
```
ValidControlEvent = False;

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;
```

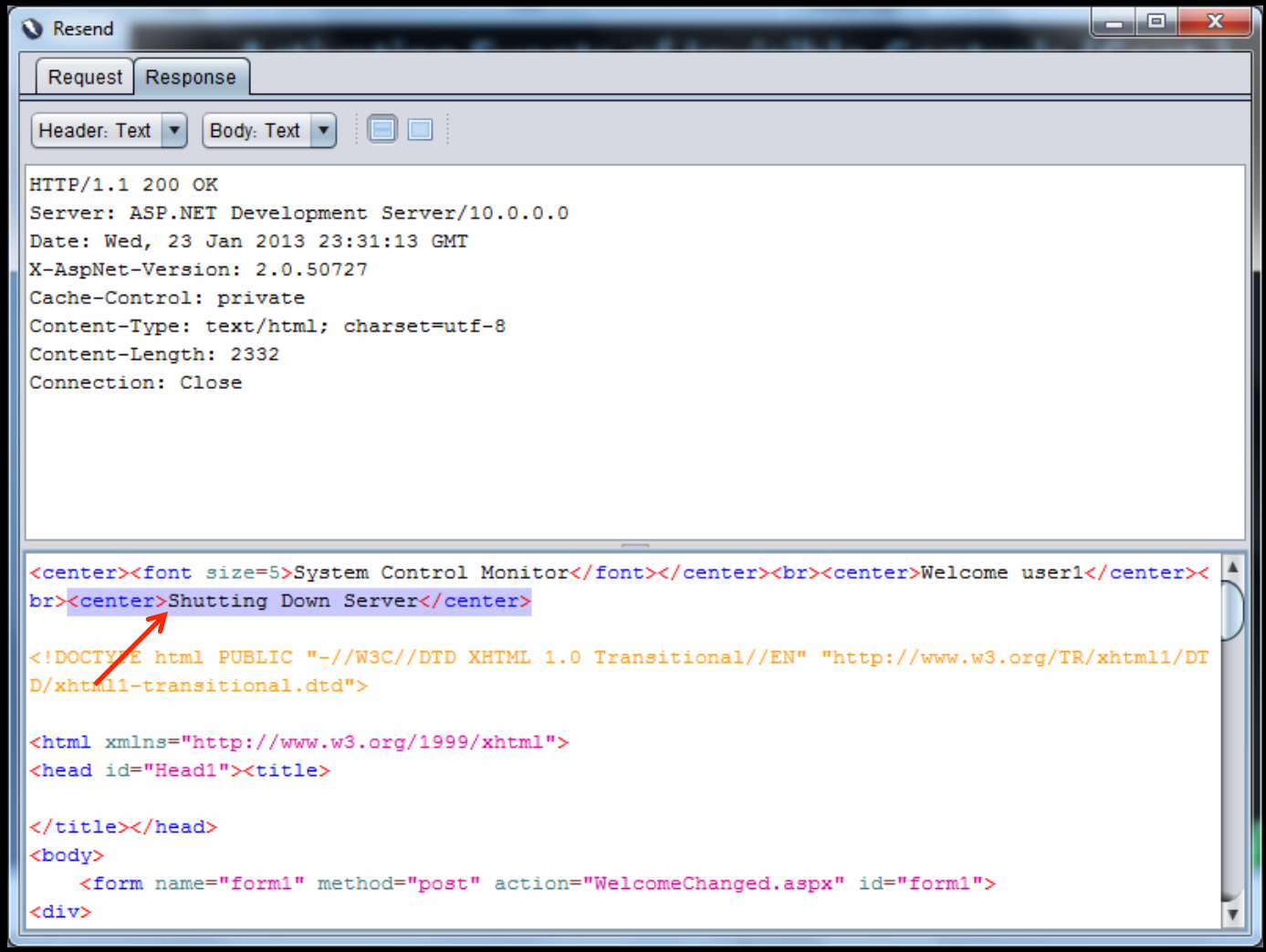
Activating Events of Invisible Controls (Cont.)



Activating Events of Invisible Controls (Cont.)



Activating Events of Invisible Controls (Cont.)



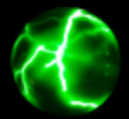
Locating Hidden Optional Events

SKILL LEVEL: ULTRA-VIOLENCE

Activating Hidden Optional Events

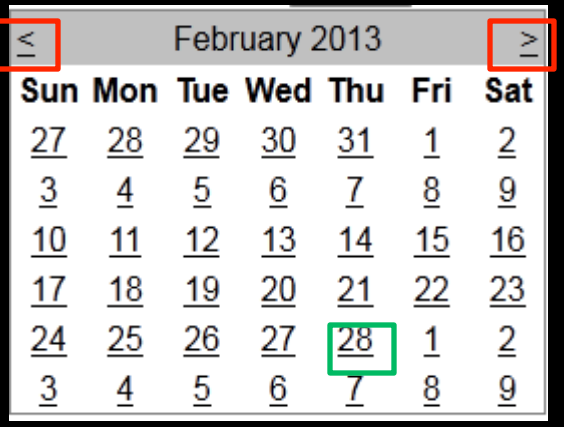
- **Prerequisites – Multiple Dormant Events of a Single Control:**
  - By default, only a limited amount of basic controls support multiple events (not including custom controls).
  - The hidden control must be assigned with multiple valid events (example: Calendar control).
  - In addition to fuzzing 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



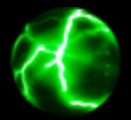
Activating Hidden Optional Events

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




```
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"
```

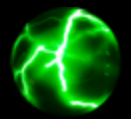



Advanced SCIP Methods

Executing Events of Invisible Controls

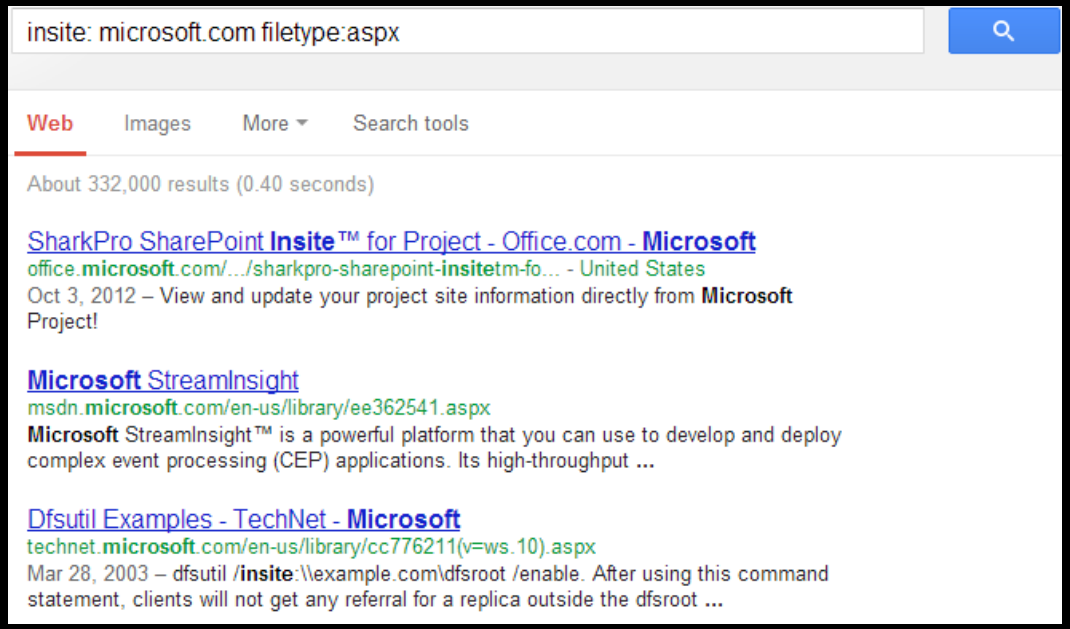
DESPITE

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!)

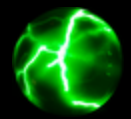


Advanced SCIP Methods (Cont.)

about:cache

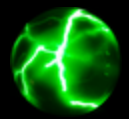
File Edit View Help

```
12 <input type="hidden" name="__EVENTTARGET" id="__EVENTTARGET" value="" />
13 <input type="hidden" name="__EVENTARGUMENT" id="__EVENTARGUMENT" value="" />
14 <input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
   value="/wEPDwUKLTY1MTIzNDQyOA9kFgICAw9kFgYCAw8PFgIeB1Zpc2libGVoZGQCBQ8PFgIeB0VuYWJsZWRoZG
   QCBw8PFgIfAWhkZGSLvPBKX768sFPPIgt0+A2Gic3bzQ==" />
15 </div>
16
17 <script type="text/javascript">
18 //<![CDATA[
19 var theForm = document.forms['form1'];
20 if (!theForm) {
21     theForm = document.form1;
22 }
23 function __doPostBack(eventTarget, eventArgument) {
24     if (!theForm.onsubmit || (theForm.onsubmit() != false)) {
25         theForm.__EVENTTARGET.value = eventTarget;
26         theForm.__EVENTARGUMENT.value = eventArgument;
27         theForm.submit();
28     }
29 }
30 //]]>
31 </script>
32
33
34 <div>
35
36     <input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION"
   value="/wEWBgKw6tnLBwKM54rGBgLWlM+bAgLs0bLrBgKgwpPxDQKF2fXbA/oH8FtecCy4qfgvxpUjUN1dAUZf"
   />
```

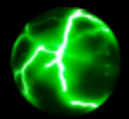
Advanced SCIP Methods (Cont.)

- 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

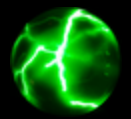


Advanced SCIP Methods (Cont.)

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

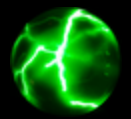


Risk Mitigation



SCIP Mitigation - Secure Coding Practices

- **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



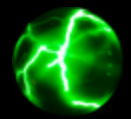
Event / Privilege Validation

- 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

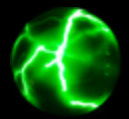
- Disable Browser/Proxy Cache (Sample Code)

```
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();
```

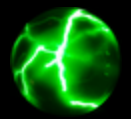
- Restrict SE access in robots.txt (Sample Config)
  - http://www.robotstxt.org/robotstxt.html

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

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

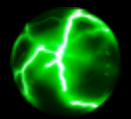


The Original Theory



The Original Research

- 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

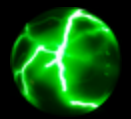


Summary


THE GANTLET
FINISHED
KILLS 92%
ITEMS 100%
SECRET 100%
TIME 19:14
PAR 02:00

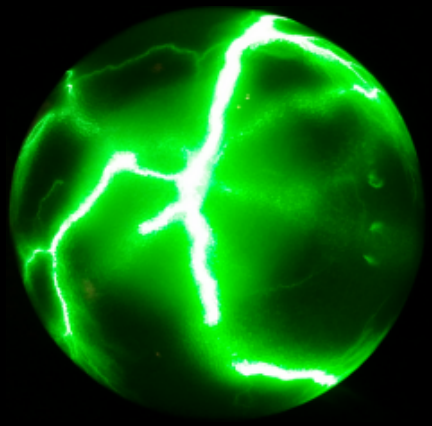
The SCIP Project

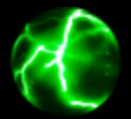
- **SCIP**
  - **Homepage:** http://code.google.com/p/ria-scip/
  - OWASP ZAP extension (v2.0+), currently focused at **ASP.net**
  - **Current Features:**
    - disabled/commented control event execution
    - error-based detection of invisible controls
    - manual execution of target events
    - Manual parameter tampering even when event validation is ON (while viewstate MAC is off)
  - **Upcoming features:** cache scraping / reuse, blind event enumeration
  - Relies on **Diviner** diff methods for Blind Control Enumeration



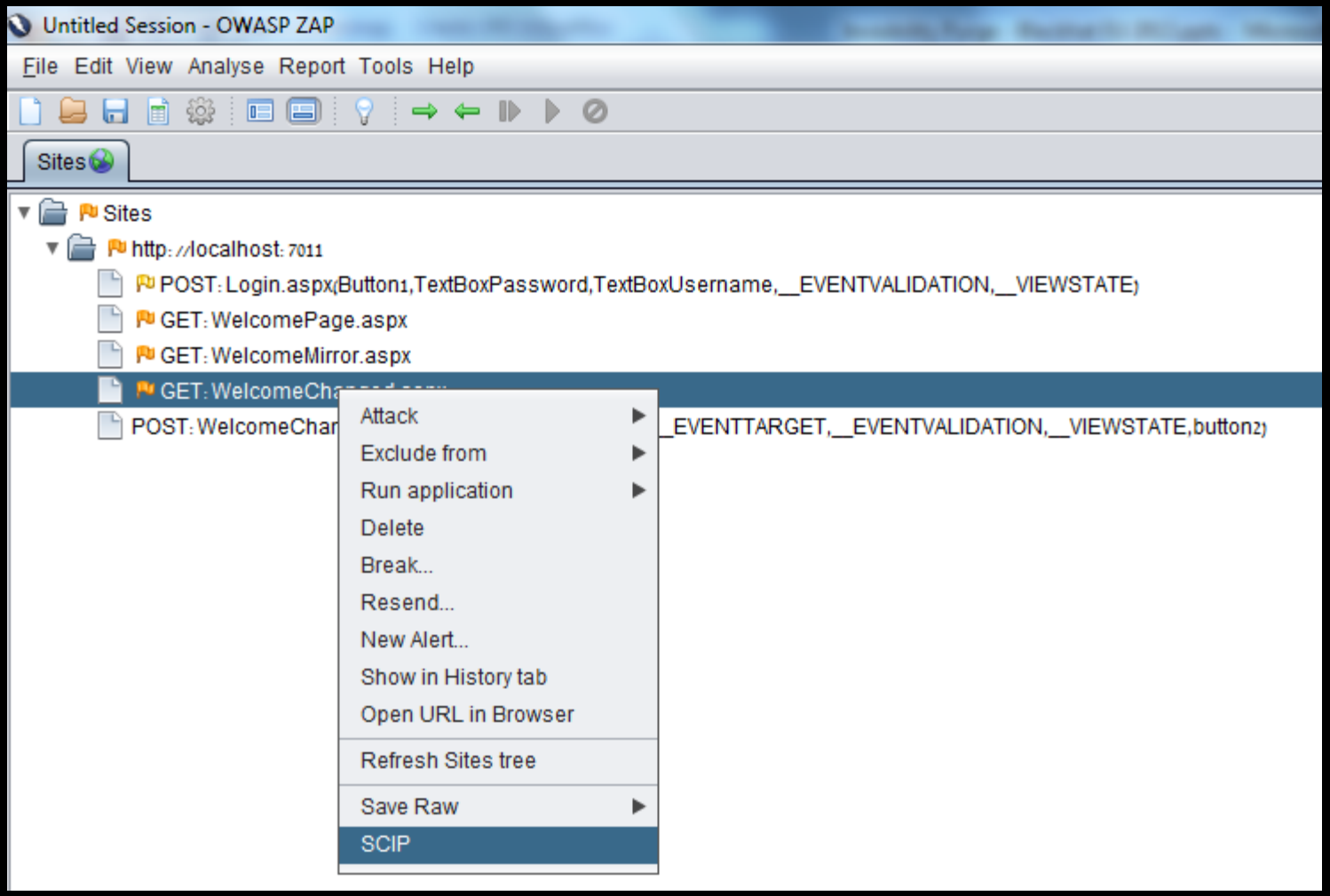
The Diviner Project

- **Diviner**
  - **Homepage:** http://code.google.com/p/diviner/
  - OWASP ZAP extension (v1.4+/v2.0+)
  - Requires ZAP to run with Java 1.7+



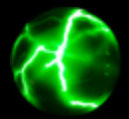


Activating SCIP in ZAP



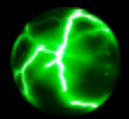
Summary & Conclusions

- 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

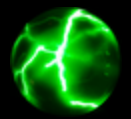


Summary & Conclusions (Cont.)

- 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

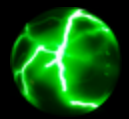


And Finally…



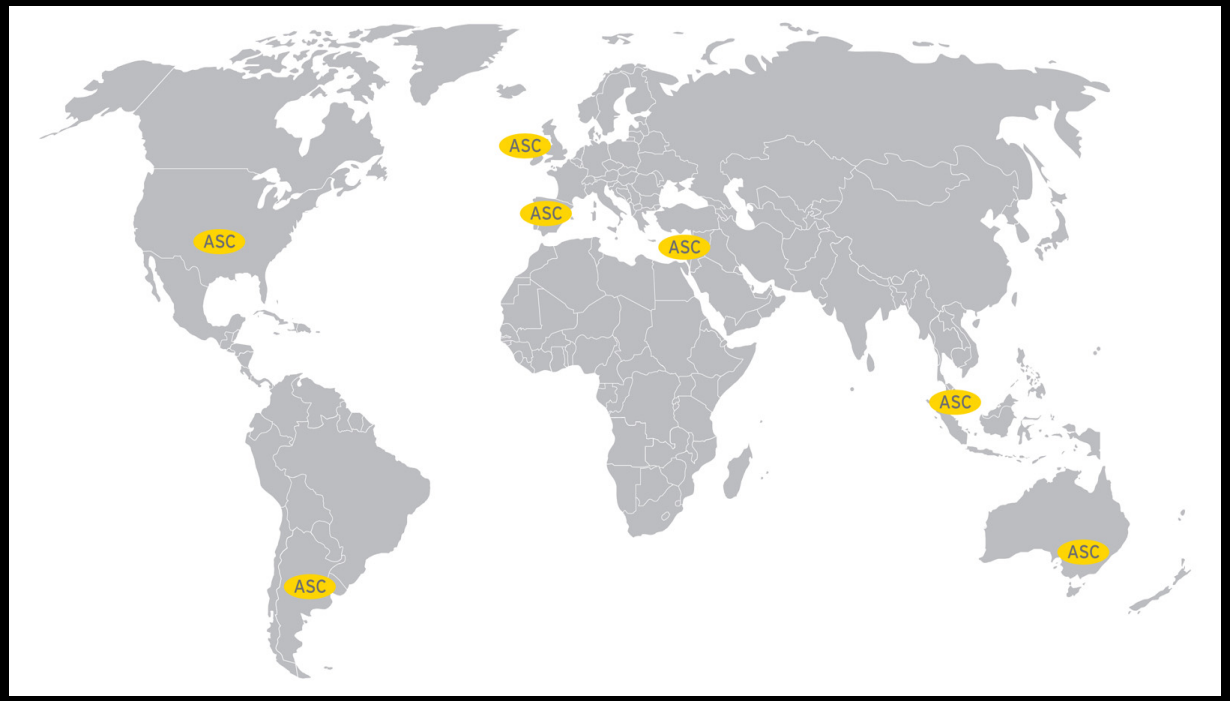
Additional Resources

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



Ernst & Young Advanced Security Centers

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



Ernst & Young

Assurance | Tax | Transactions | Advisory

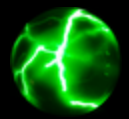
About Ernst & Young
Ernst & Young is a global leader in assurance, tax, transaction and advisory services. Worldwide, our 130,000 people are united by our shared values and an unwavering commitment to quality. We make a difference by helping our people, our clients and our wider communities achieve potential.

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
Quality In Everything We Do

black hat EU 2013



Questions?

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