INVISIBILITY PURGE
Manipulating Properties of Invisible & Dormant ASP.Net Server-Side Web Controls

Shay Chen, CTO @sectooladdict
Hacktics ASC, Ernst & Young
October 27th, 2013

Alex Mor, Team Leader @nashcontrol
About

► Formerly a boutique company that provided 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

VEHICLE!

Viewstate Hidden Control Enumerator
(formerly known as ria-scip)

A project based on a research by Niv Sela and Shay Chen,
ZAP Extension Implementation by Alex Mor.
ABUSING ASP.NET MECHANICS

- Identify Hidden Server Controls
- Enumerate Control Type
- Inject / Override Control Properties
- Cached ViewState / Validation Mining
- Web Control Event Execution
Property Injection / Property Override

- **Fingerprint** visible & hidden control types
- **Override** server side control properties
- **Access** additional data sources in the backend databases and systems
- **Execute** application attacks while bypassing ASP.net security mechanisms
Event Execution Exploits (EodSec)

► 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
Risk Factors

► Hidden Server Control Identification
► Server Control Type Fingerprinting
► Server Control Property Manipulation
► Server Control Property Injection
► Execution of Dormant Server Controls
► Cached Viewstate Reuse
The Attack Surface of ASP.net / Mono
Security Features in ASP.net/Mono

► Event Validation

► Digital Signatures / MAC
  ► Limit to List, Manipulation Prevention

► Security Filter (XSS)

► Sandbox

► Built-in Regular Expressions

► Etc
Purpose: Locating Code to Abuse

- Web Pages
- Web Service Methods
- Global Modules (Filters, Handlers, Etc)
- ...
- *Events of Server Web Controls*
Server-Side Web Controls

- Rendered into HTML/JS code
- Include server side implementation
- Core Controls and Custom Controls (e.g. ascx)
Server-Side Control Events

- A triggered **server-side code segment**, containing optional functionality (**PostBack/CallBack** in ASP.Net)
- Client triggering mechanism rely on **EVENTTARGET**, **EVENTARGUMENT** and **VIEWSTATE**
- Sample Server Side Implementation (**C#**, **ASP.Net**):
  
  **.aspx**:
  ```xml
  <asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Button" />
  ```

  **.aspx.cs**:
  ```csharp
  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";
    }
  }
  ```
Sample client-side implementation (ASP.Net postback):

```html
<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="/wEPdW1UKLY1M:
  </div>
  <input type="button" name="Button1" value="View Service Status" onclick="javascript:__doPostBack('Button1','')"

<script type="text/javascript">
  <!--
  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>
```
The Structure of the Viewstate Field

Viewstate HTML Structure:

```html
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
value="/wEPdWULLTEyNTITyMjExOTQP2BYCAgMP2BYIAgEPDxYCHgdWaNpYmxlAGkAgMDxYCHwBo2GQCBQ8PFgIeB0VuYJWJSzWROsGQCBw8PFgIfAWhkgQd1dSR00Eayc+I/Kt9v2TA3JvsHg==" />
```

- Serialized into Base64*
- Signed (MAC), clear-text or encrypted
Event Validation Drill Down

- Independent Events (e.g. buttons with usesubmitbehavior=false, etc)
- Programmatic vs. Declarative
The Event Validation Mechanism

► Name/Value HashCode Formula

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

► MachineKey and MAC

► Control Name/Value Verification, Prior to Event Execution

► Include viewstate hashcode

► Included in the HTML:

```html
<input type="hidden" name="__EVENTVALIDATION" value="/wEWCAK+2oTRBwLw1M+bAqLs0DlrBgwpPxDQKF2f:/l6KxF32QYvu1QC0yedG1oA7" />
```
Evidence of Hidden Controls & Type

Visible/Enabled Controls:

Control Panel - Zone 1

- View Service Status
  - Green
- Shutdown Service
  - Red
- Send Event Notification
  - Gray
- Logout

Invisible / Disabled Controls (Properties in Viewstate):

Control Panel - Zone 1

- View Service Status
  - Green
- Send Event Notification
  - Gray
- Server Is Up

EventValidation (Viewstate Decoder):

ViewState v2.0 compatible (MAC enabled)

- Pair
  - String 65025323
  - Null
  - Null

EventValidation (Viewstate Decoder):

ViewState v2.0 compatible (MAC enabled)

- List
  - Int 1677238116
  - Int 1757590412
  - Int -2134092357
  - Int 594790998
  - Int 1835837676
  - Int 998075525
  - Int -568008416
Server Controls / Props: Application Building Blocks
Fingerprinting Server Controls

Control Viewstate Presence
- Collection of Properties for Each Control
- Boolean, Numeric, Text, etc
- Properties reloaded from the viewstate

Rule of Thumb – Properties in Viewstate
- Programmatic value manipulation
- Significant In-Design Modifications
Mapping Viewstate Reloaded Props

```csharp
C:\Documents and Settings\Administrator\Desktop\withViewState\.\BaseDataList.cs

78 object obj = this.ViewState["DataSourceID"];  
79 C:\Documents and Settings\Administrator\Desktop\withViewState\.\BaseDataList.cs
80 this.ViewState["DataSourceID"] = value;
81 C:\Documents and Settings\Administrator\Desktop\withViewState\.\BaseDataList.cs
82 object obj = this.ViewState["UseAccessibleHeader"];  
83 C:\Documents and Settings\Administrator\Desktop\withViewState\.\BaseDataList.cs
84 this.ViewState["UseAccessibleHeader"] = value;
85 C:\Documents and Settings\Administrator\Desktop\withViewState\.\BaseDataList.cs
86 C:\Documents and Settings\Administrator\Desktop\withViewState\.\BaseDataList.cs

log.txt

1056 C:\Documents and Settings\Administrator\Desktop\withViewState\.\HotSpot.cs  
1057 string text = (string)this.ViewState["AccessKey"];  
1058 C:\Documents and Settings\Administrator\Desktop\withViewState\.\HotSpot.cs
1059 this.ViewState["AccessKey"] = value;
1060 C:\Documents and Settings\Administrator\Desktop\withViewState\.\HotSpot.cs
1061 object obj = this.ViewState["AlternateText"];  
1062 C:\Documents and Settings\Administrator\Desktop\withViewState\.\HotSpot.cs
1063 this.ViewState["AlternateText"] = value;
```
Demo:
Inject / Override the GridView sqlDataSource and key fields

Employees List
Data Source: SqlDataSource2
Data Field: account_number

<table>
<thead>
<tr>
<th>account_id</th>
<th>account_number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>234234234</td>
</tr>
<tr>
<td>2</td>
<td>675675675</td>
</tr>
<tr>
<td>3</td>
<td>345345345</td>
</tr>
</tbody>
</table>

[Manipulate ViewState] [Manipulate ViewState (Client)]
ViewState Editor

- Pair
  - SparseArray
  - Object
  - Pair
    - ArrayList
      - IndexedString
        - DataBound
      - True
      - True
      - IndexedString
        - ItemCount
      - Int32
        - 3
      - IndexedString
        - DataSourceID
    - String
    - USER INPUT:
      - Null
      - null
      - Null
      - null
      - Null
      - null
      - Null
Overriding Control Properties

► Reusing Valid Viewstates
  ► MAC Compatibility
  ► Mining Sources
  ► Structure Similarity

► Reconstructing Unsigned Viewstates
  ► Structure and Order of Properties
  ► Insertion of Value Pair Blocks
Mining VS/EV from Cache Sources
Hidden Web Controls: Archetypes
Dormant Server Web Controls, 1 of 3

- **Commented Out Controls**
  - Commented out using HTML comments (<!-- -->)
  - **Rendered inside an HTML comment**, but the server code is still active.

```csharp
protected void Button4_Click(object sender, EventArgs e)
{
    Response.Write("<center><b>Active Users</b></center>" Rental;)
```
Dormant Server Web Controls, 2 of 3

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

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

- Dormant Events of Visible Controls
  - **Declarative**: Optional events of controls with multiple events
  - **Programmatic**: Optional event listeners registered in the code level for controls with at least one active event
Dormant & Invisible Control Execution
Commented Control Event Execution

► **Prerequisites (ASP.Net)** - Commented Out Controls:
  ► Comment server control using HTML comments
  ► Event code does not include privilege validation

► **Process**
  ► “uncomment” the HTML control and execute the event, or send the appropriate values directly.

► **Advantages**
  ► Exploit works despite the **ViewState MAC** AND **EventValidation**.
Commented Control Execution, Cont.
Commented Control Execution, Cont.
Disabled Control Event Execution

➤ **Prerequisites (ASP.Net / Mono)** - Disabled Controls:
  ➤ The control “enabled” server property is set to FALSE
  ➤ Event code does not include privilege validation

➤ **Process**
  ➤ ASP.Net/Mono: Forge a postback / callback call, or send the appropriate values directly.
  ➤ Mono: delete the viewstate (!)

➤ **Advantages**
  ➤ Exploit works despite an active **ViewState MAC** AND **EventValidation**.
Forging a Disabled Control Event

► Forging a PostBack / CallBack Method Call

► Why does it work?
  ► ASP.Net: Temporarily disabled controls are a feature
  ► Mono: EventValidation’ Viewstate hash-code issue

► How does it work?
  ► The control name is exposed in the disabled control

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

  ► Use an interception proxy to “inject” postback calls into HTML control events, or craft requests manually.
Disabled Control Execution, Cont.
Invisible Control Event Execution 1/4

Prerequisites (ASP.Net / Mono) - Invisible Controls:

(I) Either the ViewstateMAC 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 control “visible” server property is set to FALSE

(III) Event code does not include privilege validation
Invisible Control Event Execution 2/4

► EventValidation is OFF
  ► No event validation = ANY event can be executed, regardless of MAC

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

► Process
  ► 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
Invisible Control Event Execution 3/4

- **EventValidation** is ON, **ViewState** MAC is OFF
  - Forge valid viewstate / eventvalidation fields (no MAC)

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

- **Process**
  - Craft a request using **VEHICLE** or other eventvalidation editors
Hashcode Generation

```csharp
internal static unsafe int GetStringHashCode(string s)
{
    fixed (char* chPtr = s)
    {
        int num1 = 352654597;
        int num2 = num1;
        int* numPtr = (int*)chPtr;
        int length = s.Length;
        while (length > 0)
        {
            num1 = (num1 << 5) + num1 + (num1 >> 27) ^ *numPtr;
            if (length > 2)
            {
                num2 = (num2 << 5) + num2 + (num2 >> 27) ^ numPtr[1];
                numPtr += 2;
                length -= 4;
            }
            else
            {
                break;
            }
        }
        return num1 + num2 * 1566083941;
    }
}
```
Error-Based Control Enumeration

- Accessing invalid control names will **NOT** raise exceptions
- Accessing protected **will** - if the 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:**

Blind Control Enumeration

Basic Blind Differentiation Formula:

```plaintext
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 = OriginalRespone - ReflectedValues - InconsistentContent - TimestampTokens;

If (Diff (ClearResponse, EventExecResponse ) > 0) ValidControlEvent = True;
```
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
Hidden/Optional Event Execution 1/2

Prerequisites – Multiple Dormant Events of Controls:
- Control assigned with multiple events.
- (Calendar control, Custom Controls, etc)

Process:
- Fuzz the eventargument field, in addition to eventtarget
- Eventargument variation can execute different server events (for example - V[value] vs. [value])

Advanced:
- Core Events vs. Custom Events
- Example: Click, Command, onSelectionChanged, OnVisibleMonthChanged, Etc
Hidden/Optional Event Execution 2/2
Invisible Control Execution, Cont.
Invisible Control Execution, Cont.

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

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
```
Advanced Event Execution Methods

Executing Events of Invisible Controls DESPITE EventValidation and Viewstate MAC
Missing CallBack Code Validation

► **Prerequisites** – Improper CallBack Implementation
  ► Event validation not performed manually in CallBack code
  ► Relies on the CALLBACKID and CALLBACKPARAM

```csharp
public void RaiseCallbackEvent(string eventArgument)
{
    try
    {
        //Page.ClientScript.ValidateEvent(this.UniqueID);
        this.OnClick(new EventArgs());
    }
    catch (Exception ex)
    {
        this._eventArg = ex.Message;
        this.OnClick(new EventArgs());
    }
}
```

► **Process:**
  ► Ignore Event Validation when Forging Event Call

► **Advantage:**
  ► Works **despite** of Viewstate MAC & EventValidation.
Mining Cached Viewstate Values 1/2

 ► **Prerequisites** – Reusing Signed Cached Values
   ► Obtain control names from cached / indexed content: search engines, proxies, browser cache of privileged users
   
     ![Search Result](https://example.com)

     **Web**
     ![Images](https://example.com)
     ![More](https://example.com)
     ![Search tools](https://example.com)
     
     About 332,000 results (0.40 seconds)

     ![Search Result](https://example.com)
     Shapr3o SharePoint InSite™ for Project - Office com - Microsoft
doce.microsoft.com/.../shapr3o-sharepoint-insiteinfo... - United States
     Oct 3, 2012 – View and update your project site information directly from Microsoft Project!

     ![Search Result](https://example.com)
     Microsoft Streaminsight
     **Microsoft Streaminsight™** is a powerful platform that you can use to develop and deploy complex event processing (CEP) applications. Its high-throughput ...

     ![Search Result](https://example.com)
     Dfsutil Examples - TechNet - Microsoft
tochnet.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 ...

 ► **Process:**
   ► Reuse the cached VIEWSTATE, EVENTTARGET, EVENTARGUMENT and EVENTVALIDATION
Mining VS/EV from Cache Sources

![VEHICLE - Viewstate Hidden Control Enumerator](image)

- **Options Help**
  - **Vehicle**
  - **Deja Vu**

**URL:** http://localhost:1442/ControlPanelSection7.aspx

**Date**
- **Select Year:** 2011

**Deja Vu Analysis**
- Viewstate
- Event Validation

**Deja Vu Controls**
- Run Event
- Post Back
Advantages

Exploit works DESPITE the Viewstate MAC AND EventValidation

Shared Hosting Attack Model:

Can bypass Viewstate MAC and EventValidation

Scenarios for Shared / Isolated Application Pool
Risk Mitigation
Secure Coding Guidelines

Preventing Event Execution & Property Override

► 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**
► Replace the **Machine Key** (to prevent cache reuse)
Event Level Privilege Validation

► Explicit Privilege Validation in Event Code

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

► Enable Event Validation / MAC

```html
<%@ Page Language="C#" AutoEventWireup="true"
EnableEventValidation="true" EnableViewStateMac="true" ...%>
```
Disable Cache & Prevent Indexing

► Disable Browser/Proxy Cache (Sample Code)

```csharp
HttpContext.Current.Response.Cache.SetValidUntilExpires(false);
```

► Restrict SE access in robots.txt (Sample Config)

- [http://www.robotstxt.org/robotstxt.html](http://www.robotstxt.org/robotstxt.html)

  User-agent: *
  Disallow: /

► Restrict SE caching/crawling via meta tags

- [http://www.robotstxt.org/meta.html](http://www.robotstxt.org/meta.html)
## Dormant Control Execution Summary

<table>
<thead>
<tr>
<th>Control / Event Type</th>
<th>ONLY Event Validation Is ON</th>
<th>ONLY Viewstate MAC is ON</th>
<th>Event Validation AND Viewstate MAC are ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commented</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
</tr>
<tr>
<td>Disabled</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
</tr>
<tr>
<td>Invisible</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![Ban]</td>
</tr>
<tr>
<td>Optional</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![Ban]</td>
</tr>
</tbody>
</table>
Advanced Invisible Control Execution

► Execute Control Events DESPITE MAC/Validation
  ► Reuse cached / indexed ViewState and EventValidation fields of the same Page
  ► Abuse insecure callback implementations
  ► Abuse MachineKey in Shared Hosting Model
The VEHICLE Project (a.k.a RIA-SCIP)

- **Homepage:** [http://www.github.com/hacktics/vehicle/](http://www.github.com/hacktics/vehicle/)
- **Alternate:** [http://code.google.com/p/ria-scip/](http://code.google.com/p/ria-scip/)
- OWASP ZAP extension (v2.0+), requires Java 1.7
- Included in ZAP’s Marketplace
- Currently focused at **ASP.net**
Activating the Vehicle Extension in ZAP
The Vehicle Project, Cont.

► **Current Features**
  ► Passively identifies traces of Invisible / Dormant Controls
  ► Error-based invisible control name enumeration
  ► Blind-based invisible control name enumeration (NEW!)
  ► Disabled/commented control event execution
  ► Invisible control event execution
  ► Viewstate manipulation for manual parameter tampering, when MAC is OFF (NEW!), Control Injection Templates
  ► Manual execution of target events
  ► Cached viewstate/eventvalidation scraping - déjà vu (NEW!)

► **Upcoming Features**
  ► Control type fingerprinting & exploitation
Additional Resources

► déjà vu (cache analysis) ZAP Extension: https://github.com/hacktics/deja-vu

► Woanware Viewstate Hacker: http://www.woanware.co.uk/application/viewstatehacker.html

► OWASP ZAP: https://code.google.com/p/zaproxy/

► James Jardine blog posts: http://www.jardinesoftware.net/
EY Advanced Security Centers

- **Asia Pacific**
  - Melbourne
  - Singapore

- **Americas**
  - Hacktics IL
  - Houston
  - New York
  - Buenos Aires

- **EMEIA**
  - Dublin
  - Barcelona
QUESTIONS?

Shay Chen (@sectooladdict)
Niv Sela (@nivselatwit)
Alex Mor (@nashcontrol)