



Hacking ASP.Net: Tips and Tricks

James Jardine

james@secureideas.com

(866) 404-7837

@JardineSoftware

James Jardine

- Principal Security Consultant at Secure Ideas
- .Net Developer Since the Beta Release
- SANS Instructor and Author
 - Dev544: Secure Coding in .Net
- Open Source Projects
 - Web Config Security Analyzer - <http://sourceforge.net/projects/wcsa/>
 - EventValMod - <http://sourceforge.net/projects/eventvalmod>
- Podcaster
 - Professionally Evil Perspective
 - Down the Rabbit Hole
- Blogs
 - .Net Security - <http://www.jardinesoftware.net/>
 - Gen. Security – <http://blog.secureideas.net>



Topics

- ASP.Net
- RequestValidation
- ViewState
- EventValidation
- GET/POST & Postback
- Conclusion



ASP.Net

```
$js_command_hist ?>); var last = 0; function key(e) { ?> va r current_line = 0; var command_hist = New array (<?php
```

Versions

- 1.1
- 2.0
- 3.0
- 3.5
- 4.0
- 4.5 *

Editions

- WebForms
- MVC
- Web Pages
- Web API
- WCF



Testing ASP.Net

- Similar to other technologies

- GETs/POSTs, etc
- AJAX
- Cookies, Hidden Fields, Forms
- Session State, Authentication

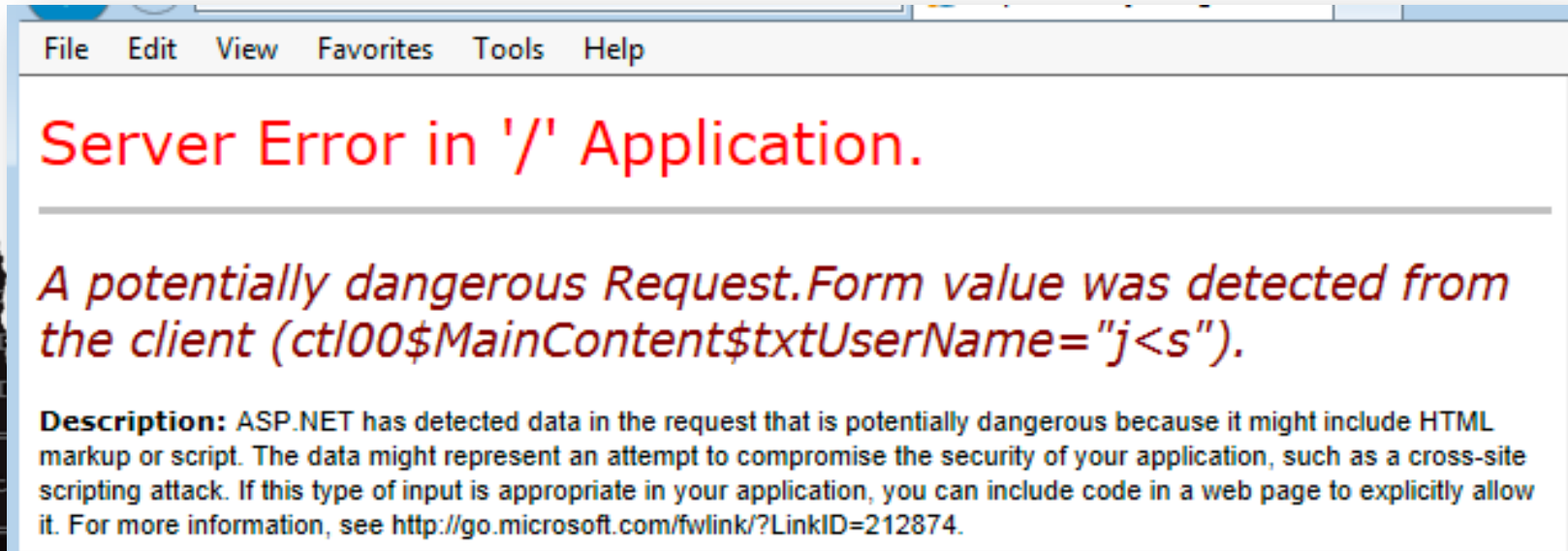
- Differentiators

- Request Validation
- View State
- Event Validation
- Other Built In Controls



Request Validation

- Attempt to block XSS Attacks
- In 2.0+ only works for **HTML** Context
 - <[char], <!, <?, </, &#
- Prior to 2.0 most likely disabled



The screenshot shows a web browser window with a menu bar (File, Edit, View, Favorites, Tools, Help) and a red error message. The error message reads: "Server Error in '/' Application." followed by a detailed description of a potentially dangerous request.

Server Error in '/' Application.

A potentially dangerous Request. Form value was detected from the client (ctl00\$MainContent\$txtUserName="j<s").

Description: ASP.NET has detected data in the request that is potentially dangerous because it might include HTML markup or script. The data might represent an attempt to compromise the security of your application, such as a cross-site scripting attack. If this type of input is appropriate in your application, you can include code in a web page to explicitly allow it. For more information, see <http://go.microsoft.com/fwlink/?LinkID=212874>.



Request Validation Bypass

- Not Really!
- If database stores data as varchar (not nvarchar)
- Use unicode-wide %uFF1C (<)
- RequestValidation doesn't detect this but...
- Database will convert it to the < character

Of course output encoding does block this as well



Request Validation Bypass 2

- Addition of % Character (<%tagname>)
- Reported to work in IE (I was unsuccessful)
- Reported by Zamir Paltiel (<http://www.securityfocus.com/archive/1/524043>)
- An older bypass was to use a null character like <%00tagname>
- Browser specific and doesn't really work anywhere

Of course output encoding does block this as well



Request Validation Config

- Set in the Web.Config File

```
<system.web>
```

```
<pages validateRequest="true" />
```

```
</system.web>
```

- Set at the Page Level

```
<%@ ValidateRequest="true" %>
```



Yes, It's Interesting

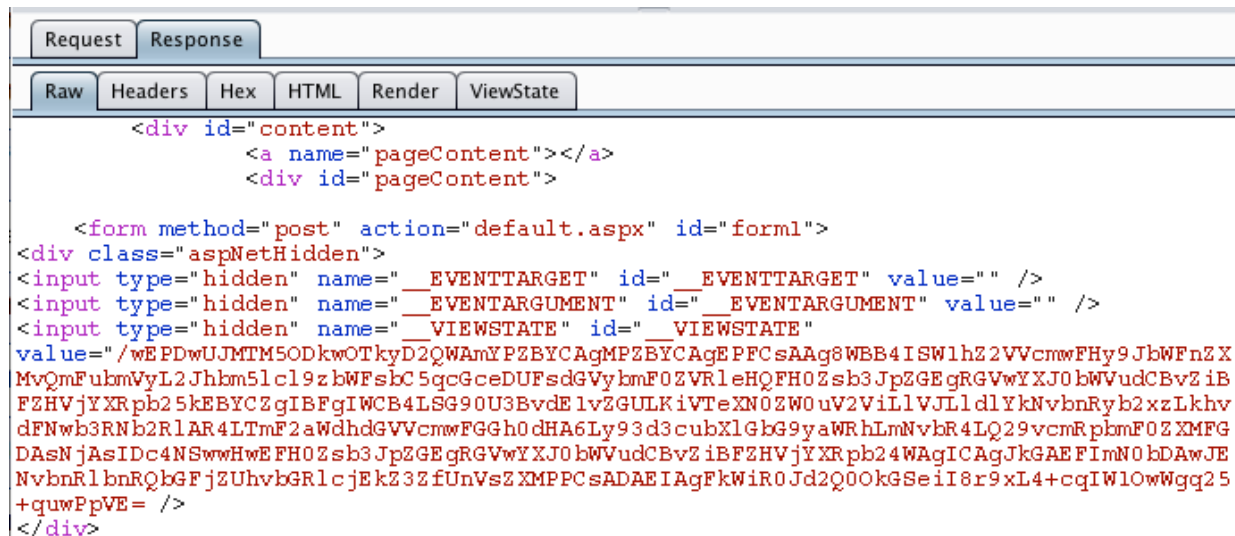
```
$js_command_hist ?>); var last = 0; function key(e) { ?> va r current_line = 0; var command_hist = New array (<?php
```

ViewState v2.0 compatible [MAC is not enabled]



ViewState

- Base64 Encoded By Default
 - Can be encrypted
- Vulnerabilities
 - Parameter Tampering, XSS, Info Leakage



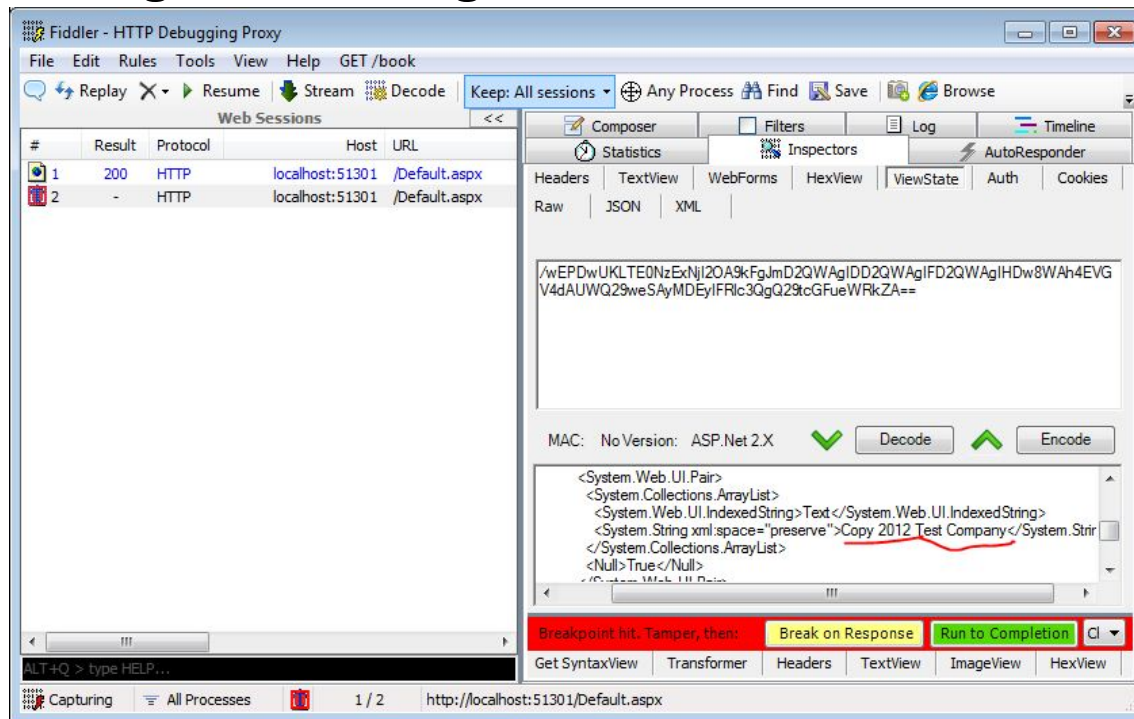
```
Request Response
Raw Headers Hex HTML Render ViewState
<div id="content">
  <a name="pageContent"></a>
  <div id="pageContent">
    <form method="post" action="default.aspx" id="form1">
<div class="aspNetHidden">
<input type="hidden" name="__EVENTTARGET" id="__EVENTTARGET" value="" />
<input type="hidden" name="__EVENTARGUMENT" id="__EVENTARGUMENT" value="" />
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
value="/wEPDwUJMTM5ODkwOTkyD2QWAmYpZBYCAgMPZBYCAgEPFCsAAg8WBB4ISW1hZ2VvcmwFH9JbWFnZXMvQmFubmVYL2Jhbm5lc19zbWFSbC5qcGceDUFsdGVybmF0ZVRleHQFH0Zsb3JpZGEgRGVwYXJ0bWVudCBvZiBFZHVjYXRpb25kEYCB4LSG90U3BvdE1vZGULKiVTeXN0ZW0uV2ViLlVJL1dlYkNvb3R5b2xzLkhvZFNwb3Rnb2RlAR4LTmF2aWdhGVVcmwFGGh0dHA6Ly93d3cubXlgbG9yaWRhLnNvbR4LQ29vcwRpbmF0ZXMFGDAsNjAsIDc4NSwwHwE FH0Zsb3JpZGEgRGVwYXJ0bWVudCBvZiBFZHVjYXRpb24WAQICAgJkGAEFIIn0kdAwJE Nvb3RlbnRQbGFjZUhhbGRlcjEkb2Z3fUUnVsZXMPPCsADAEIAGfKwIR0Jd2Q00kgSeii8r9xL4+cqIW10wWgq25 +quwPpVE= />
</div>
```



ViewState Manipulation

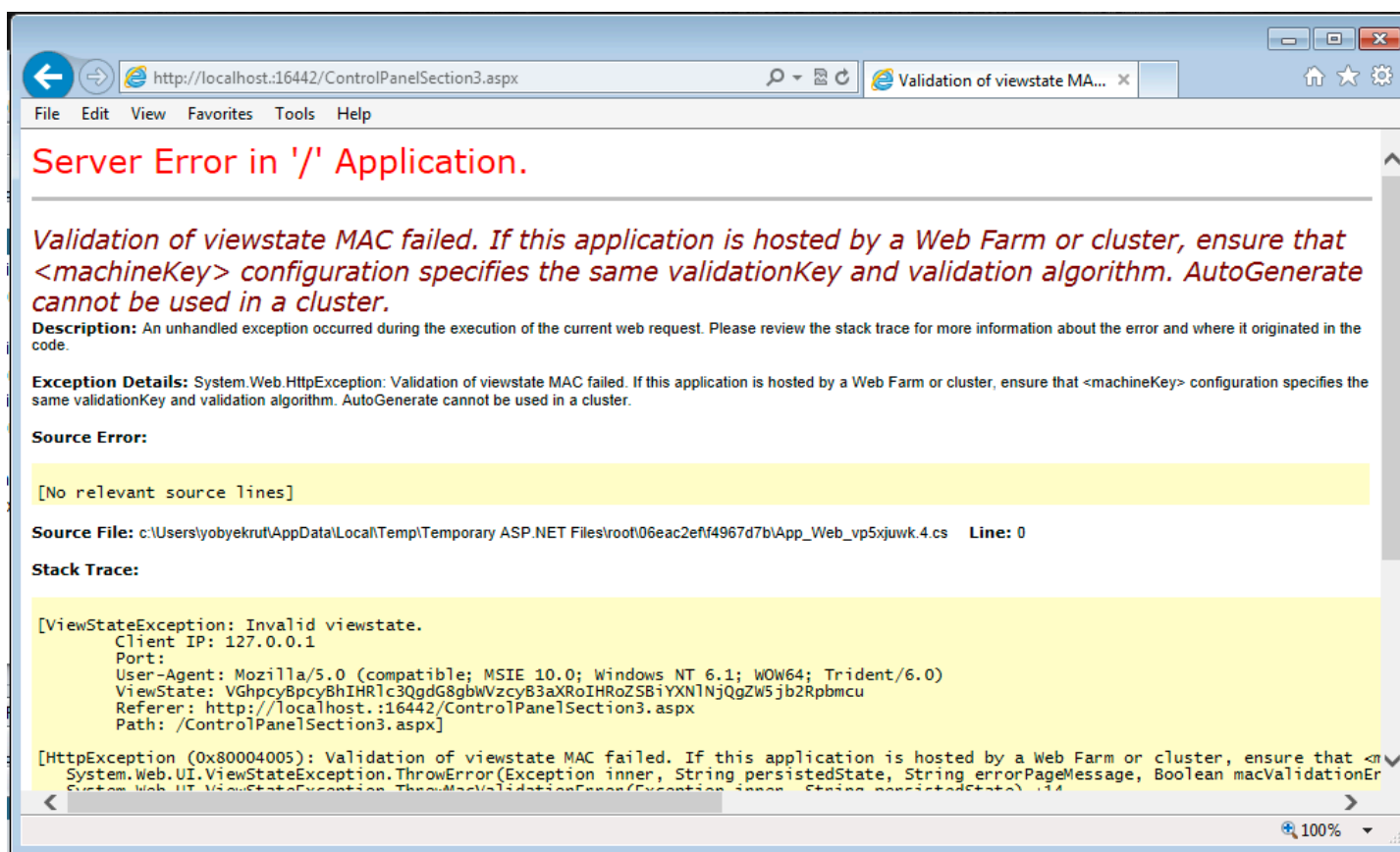
```
Sjs_command_hist ?>); var last = 0; function key(e) { ?> va r current_line = 0; var command_hist = New array (<?php
```

- ViewStateViewer - <http://labs.neohapsis.com/2009/08/03/viewstateviewer-a-gui-tool-for-deserializingreserializing-viewstate/>



ViewState - Protected

```
Sjs_command_hist ?>); var last = 0; function key(e) { ?> va r current_line = 0; var command_hist = New array (<?php
```



The Problem

`$js_command_hist ?>); var last = 0; function key(e) { ?> va r current_line = 0; var command_hist = New array (<?php`

This is **wrong** common advice!!

Solution 1

Sign Up to vote  

Refer to the following thread for a discussion about this issue:

<http://forums.asp.net/t/955145.aspx>

You can fix this by setting the `EnableViewStateMAC` property to false. Refer to more information about `EnableViewStateMAC` in the link given below:

<http://msdn.microsoft.com/en-us/library/system.web.ui.page.enableviewstatemac.aspx>

Posted 24-Sep-12 16:39pm

[Permalink](#)

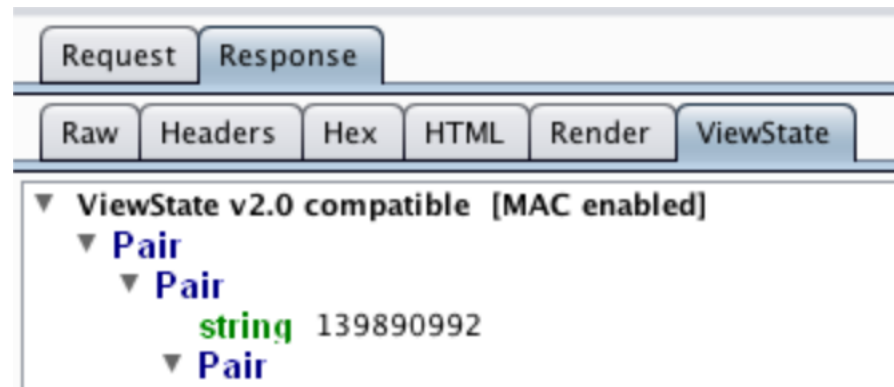
<http://www.codeproject.com/Questions/464873/Validation-of-viewstate-MAC-failed>



ViewStateMac

```
$js_command_hist ?>); var last = 0; function key(e) { ?> va r current_line = 0; var command_hist = New array (<?php
```

- Provides Tamper Protection for:
 - ViewState
 - EventValidation



Web.Config

```
<pages enableViewStateMac="true"/>
```

Page Level

```
<%@ Page Language="C#" EnableViewStateMac="true"...
```



Event Validation

- Protects Drop Down Lists
- Protects against forged post backs
- Protected by ViewStateMac
- Creates an array of numeric hashes
- Not User Specific
 - Doesn't Protect against CSRF

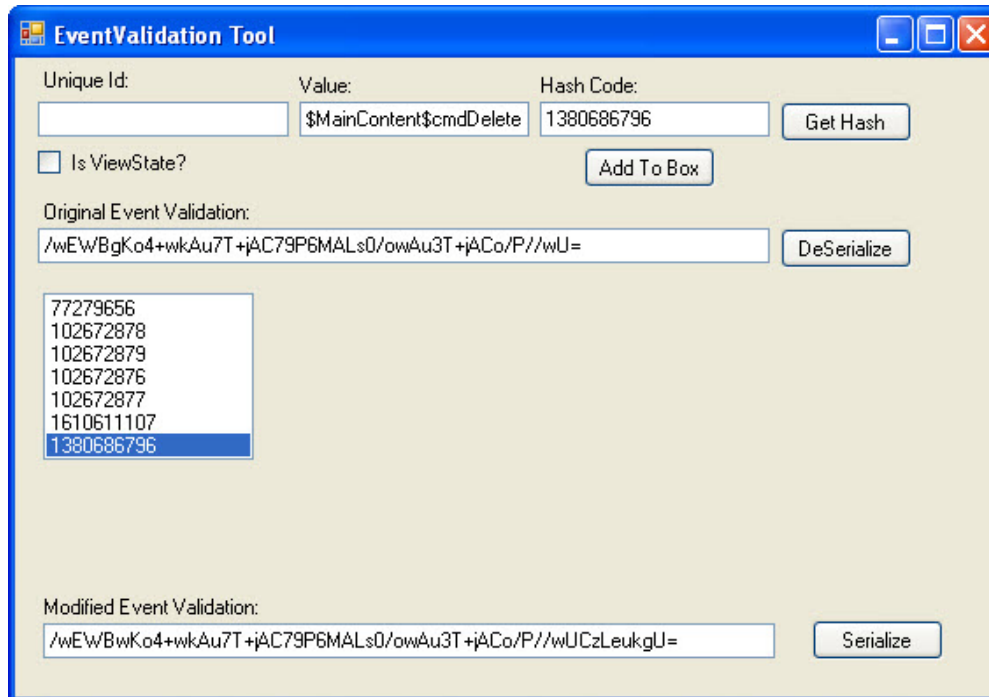
```
<input type="hidden" name="__EVENTVALIDATION" value="/wEWBALsIL0qAu3wv7QBAqnOkfQNAoznisYG"/>
```



Event Validation - EventValMod

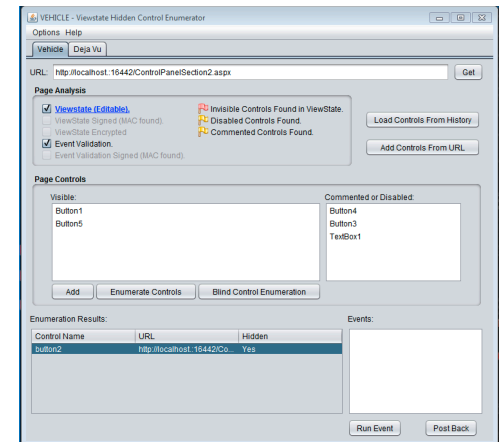
```
$js_command_hist ?>); var last = 0; function key(e) { ?> va r current_line = 0; var command_hist = New array (<?php
```

- Modifies the Event Validation field
- Stand Alone App / Written in .Net
- <http://sourceforge.net/projects/eventvalmod>



Event Validation - VEHICLE

- ViewState Hidden Event Enumerator
 - Formerly known as ria-scip
- Works with ZAP
- Features
 - Event Execution of Disabled/Invisible Controls
 - Server Control Property Injection
 - Edit the ViewState Field
 - Error-Based Control Name Enum
 - ViewState/EventValidation Reconstruction
- <https://github.com/hacktics/vehicle>



EventValidation Config

- Set in the Web.Config File

```
<system.web>
```

```
<pages enableEventValidation="true" />
```

```
</system.web>
```

- Set at the Page Level

```
<%@ EnableEventValidation="true" %>
```



Bad, Bad, Bad!!

```
$js_command_hist ?>); var last = 0; function key(e) { ?> va r current_line = 0; var command_hist = New array (<?php
```

ViewState v2.0 compatible [MAC is not enabled]



ViewStateUserKey

- Protects against Cross Site Request Forgery
 - Provides a user "salt" to ViewStateMac
- Not enabled by default
- Only works for requests with ViewState
 - <http://www.testsite.mm/deleteuser.aspx?id=5> (doesn't work)

- Recommendation:

```
protected void Page_Init(object sender, EventArgs e)
{
    Page.ViewStateUserKey = Session.SessionId;
}
```



PostBack

- Webforms are based around "PostBacks"
- Caused by Events (ex. button_click)
- Triggered by __ViewState or __EventTarget

```
if (!Page.IsPostBack){  
    // Authorization/Populate Data  
    lblCopy.Text = "copy 2013";  
    if(!User.IsInRole("Admin"))  
        Response.Redirect("Unauthorized.aspx");  
}  
else{  
    // Execute Events  
}
```



Postback Attacks

```
$js_command_hist ?>); var last = 0; function key(e) { ?> va r current_line = 0; var command_hist = New array (<?php
```

- Authorization Bypass

```
if(!User.IsInRole("Admin"))  
    Response.Redirect("Unauthorized.aspx");
```

Recommendation:

- Check Authorization on Every Request

- XSS (ViewState Tampering)

```
lblCopy.Text = "copy 2013";
```

Recommendation:

- Enable ViewStateMac
- Set text on every request



GET/POST Exchange

- Server Control GETs and POSTs are Interchangeable
 - TextBox
 - ListBox
 - ViewState/EventValidation
 - Etc.
- Based on Request Type
- Can Call POST requests with GET
 - Good for CSRF
- Can Trigger Postback with GET request



GET/POST Fix

```
$js_command_hist ?>); var last = 0; function key(e) { ?> va r current_line = 0; var command_hist = New array (<?php
```

- WebForms
if(Request.RequestType == "POST")
- MVC
[HttpPost]
void DoSomething()



Authentication Cookie

- HTTPOnly (Hard Coded)
- Secure Flag may not be set
 - Sometimes there is an error if behind a Load Balancer that strips SSL
 - Should Recommend Manually setting this value
- Self-Contained – Not tracked on server
 - Timeout is key. Lives until the timeout expires on the cookie
 - FormsAuthentication.Logout only removes cookie from the browser (doesn't kill it)



Misc. Files

- Trace.axd
- Elmah.axd
- Use URL Authorization in the Web.config
- Web.config (crown jewels) – GOOD LUCK!!
 - IIS is set up to not serve this file



Conclusion

- ASP.Net has good security features
 - You have to understand them
- ViewStateMac is IMPORTANT!
 - EventValidation
 - ViewState
 - ViewStateUserKey
- Developers are not up to speed on these things
 - Share this info with developers





Hacking ASP.Net: Tips and Tricks

James Jardine

james@secureideas.com

(866) 404-7837

@JardineSoftware