Internet Explorer 11 Exploit Cheat Sheets

Author: Chen Zhang (@demi6od) <demi6d@gmail.com>
Date: 2014 May 23rd

1. Javascript Array

1.1 PoC

All PoCs are at https://github.com/demi6od
Array Spray.html

1.2 Data Structure

jscript9!Js::JavascriptArray

vTable	04de5850	00000000	00000005
length	pData	pData	00000000
00000000	00000000		

ArrayData

index	length	capacity	pNext
data[0]	data[1]	data[2]	data[3]
data[4]	data[5]	data[6]	data[7]
...	...	...	...

jscript9!LargeHeapBlock Entry

00000003	largeHeap BlockSize	00000000	00000000

1.3 Adjacent JavascriptNativeIntArray Spray

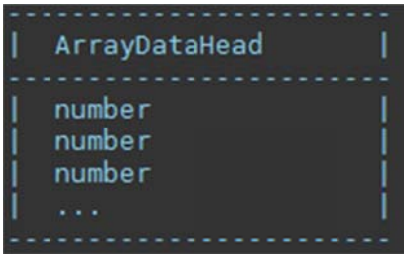
If size <= 0x100, allocate adjacent JavascriptNativeIntArray and ArrayData.

0:007> dd 02d0d000 l40
02d0d000 68662f54 02d362a0 00000000 00000005
02d0d010 00000030 02d0d028 02d0d028 00000000
02d0d020 00000001 02ab5e50 00000000 00000030
02d0d030 00000030 00000000 00adc0df 44444444
02d0d040 44444444 44444444 44444444 44444444
02d0d050 44444444 44444444 44444444 44444444
02d0d060 44444444 44444444 44444444 44444444
02d0d070 44444444 44444444 44444444 44444444
02d0d080 44444444 44444444 44444444 44444444
02d0d090 44444444 44444444 44444444 44444444
02d0d0a0 44444444 44444444 44444444 44444444
02d0d0b0 44444444 44444444 44444444 44444444
02d0d0c0 44444444 44444444 44444444 44444444
02d0d0d0 44444444 44444444 44444444 44444444
02d0d0e0 44444444 44444444 44444444 44444444
02d0d0f0 44444444 00adc0df 00000000 00000000

Sensitive length
JavascriptNativeIntArray
ArrayDataHead
Padding
Data

1.4 Separate small JavascriptNativeIntArray Spray

If 0x100 <= size <= 0x 300, allocate separate ArrayData.



0:007> dd 02820600

02820600 00000000 000000bc 000000bc 00000000

02820610 00adc0df 44444444 44444444 44444444

02820620 44444444 44444444 44444444 44444444

02820630 44444444 44444444 44444444 44444444

02820640 44444444 44444444 44444444 44444444

02820650 44444444 44444444 44444444 44444444

02820660 44444444 44444444 44444444 44444444

02820670 44444444 44444444 44444444 44444444

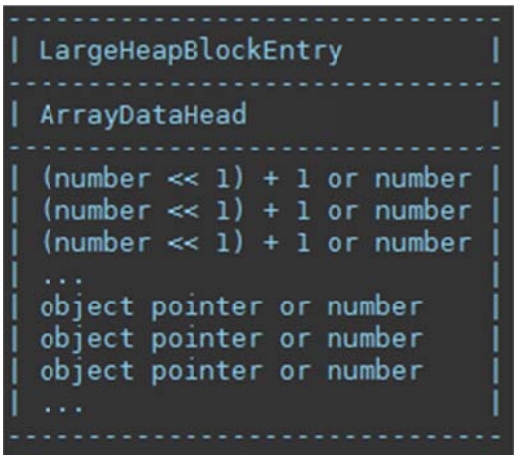
Sensitive length

ArrayDataHead

Data

1.5 Separate large JavascriptNativeIntArray or JavascriptArray Spray

If size >= 0x300, allocate separate ArrayData and put into LargeHeapBlock.



0:007> dd 0d0d0000

0d0d0000 00000000 0000fff0 00000000 00000000

0d0d0010 00000000 00003ff8 00003ff8 00000000

0d0d0020 0eadc0df 41410011 41410021 41410031

0d0d0030 41410041 41410051 41410061 41410071

0d0d0040 41410081 41410091 414100a1 414100b1

0d0d0050 414100c1 414100d1 414100e1 414100f1

0d0d0060 41410101 41410111 41410121 41410131

0d0d0070 41410141 41410151 41410161 41410171

Sensitive length

LargeHeapBlockEntry

ArrayDataHead

Data

2. Javascript Typed Array

2.1 PoC

VarArr+TypedArr Spray.html

2.2 Data Structure

jscript9!Js::TypedArray

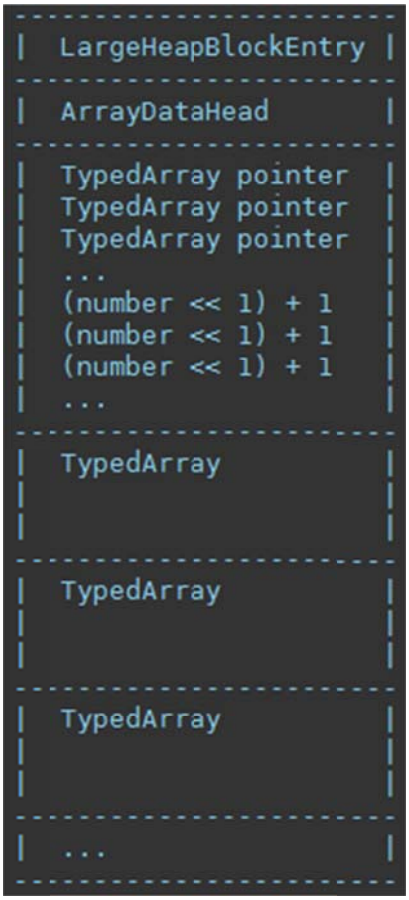
vTable	027264e0	00000000	00000000
00000004	00000000	length	pData
04156fe0	00000000	00000000	00000000

TypedArrayData

data[0]	data[1]	data[2]	data[3]
data[4]	data[5]	data[6]	data[7]
...	...	...	...

2.3 VarArray & TypedArray Spray

Heap fengshui to allocate adjacent VarArrayData and TypedArray




```
0:016> dd 0d0d0000
0d0d0000  00000000 0000eff0 00000000 00000000
0d0d0010  00000000 00003bf8 00003bf8 00000000
0d0d0020  0d0cff60 0d0cff90 0d0cffc0 0d0df000
0d0d0030  0d0df030 0d0df060 0d0df090 0d0df0c0
0d0d0040  0d0df0f0 0d0df120 0d0df150 0d0df180
0d0d0050  0d0df1b0 0d0df1e0 0d0df210 0d0df240
0d0d0060  0d0df270 0d0df2a0 0d0df2d0 0d0df300
0d0d0070  0d0df330 0d0df360 0d0df390 0d0df3c0
0:016> dd 0d0df000 - 10
0d0deff0  41410341 41410351 41410361 00adc0dd
0d0df000  686eb238 03bc6480 00000000 00000000
0d0df010  00000004 00000000 0000001a 01063648
0d0df020  03ee6fe0 00000000 00000000 00000000
0d0df030  686eb238 03bc6480 00000000 00000000
0d0df040  00000004 00000000 0000001a 01063648
0d0df050  03ee6fe0 00000000 00000000 00000000
```

Sensitive length

LargeHeapBlockEntry

ArrayDataHead

Data

3. HTML Element Property String

3.1 PoC

ElemProp Spray.html

3.2 HTML Element Property String Spray

If blockSize > 0x80000, allocate virtual memory directly instead of windows heap management.

```
0:016> dd 0d0a0000
0d0a0000  0d2b0000 0ce90000 00000000 00000000
0d0a0010  00100000 00100000 55f4959c 04000000
0d0a0020  deadc0de 3d0d0320 3d0d1320 ff0d2320
0d0a0030  ffffffff 3d0d43ff 3d0d5320 3d0d6320
0d0a0040  00200020 3d0d8300 0d0df300 3d0da320
0d0a0050  3d0db320 3d0dc320 3d0dd320 0d618820
0d0a0060  3d0df30d 3d1d0308 3d1d1320 3d1d2320
0d0a0070  3d1d3320 3d1d4320 3d0d0d64 3d1d6320
```

VirtualMemoryHead

4. Reference

[1] The Art of Leaks (@ga1ois)
https://cansecwest.com/slides/2014/The%20Art%20of%20Leaks%20-%20read%20version%20-%20Yoyo.pdf

[2] hacking ie11 32-bit: write once, bypass all (@bluerust)
http://hi.baidu.com/bluerust/item/8fffe0e5e60a623c86d9deff

[3] Windows 8.1 + IE 11 Exploit (@exp-sky)
http://www.exp-sky.org/windows-81-ie-11-exploit.html

[4] Exploiting Internet Explorer 11 64-bit on Windows 8.1 Preview (Ivan Fratric)
http://ifsec.blogspot.jp/2013/11/exploiting-internet-explorer-11-64-bit.html