

ATTACKING WINDOWS BY WINDOWS

**Yin Liang & Zhou Li
Tencent PC Manager**

Outline

- About us
- How to exploit old Windows OS
- Windows 10`s limit
- New exploit method



About us

Team member:

- xin, godz, ki, michael, kelvin, willj

Achieve:

- Attacking Adobe Flash in Pwn2Own 2016



- Attacking Surface Pro 4 in GeekPwn Macao



● 46 acknowledgments !

4月7日

腾讯电脑管家发现 Adobe flash 漏洞 14 个

Tencent PC Manager working with Trend Micro's ZDI (CVE-2016-1018)
willj of Tencent PC Manager (CVE-2016-1020, CVE-2016-1021, CVE-2016-1022, CVE-2016-1023, CVE-2016-1024, CVE-2016-1025, CVE-2016-1026, CVE-2016-1027, CVE-2016-1028, CVE-2016-1029, CVE-2016-1031, CVE-2016-1032, CVE-2016-1033)

5月12日

腾讯电脑管家发现 Adobe Flash 漏洞 7 个

willj of Tencent PC Manager (CVE-2016-4109, CVE-2016-4110, CVE-2016-4111, CVE-2016-4112, CVE-2016-4113, CVE-2016-4114, CVE-2016-4115)

5月

腾讯电脑管家发现 微软 漏洞 2 个

CVE-2016-0174, Liang Yin of Tencent PC Manager working with Trend Micro's Zero Day Initiative (ZDI)

CVE-2016-0175, Liang Yin of Tencent PC Manager working with Trend Micro's Zero Day Initiative (ZDI)

5月5日

腾讯电脑管家发现 Adobe Reader 漏洞 6 个

kelvinwang of Tencent PC Manager (CVE-2016-1081, CVE-2016-1082, CVE-2016-1083, CVE-2016-1084, CVE-2016-1085, CVE-2016-1086)

6月16日

腾讯电脑管家发现 Adobe Flash 漏洞 12 个

willj of Tencent PC Manager (CVE-2016-4122, CVE-2016-4123, CVE-2016-4124, CVE-2016-4125, CVE-2016-4127, CVE-2016-4128, CVE-2016-4129, CVE-2016-4130, CVE-2016-4131, CVE-2016-4134, CVE-2016-4166)

kelvinwang of Tencent PC Manager (CVE-2016-4133)

How to exploit?

Q1: Where to write?

Q2: What to write?

Q3: What can we do now?



Old days..

Q1: Where to write?

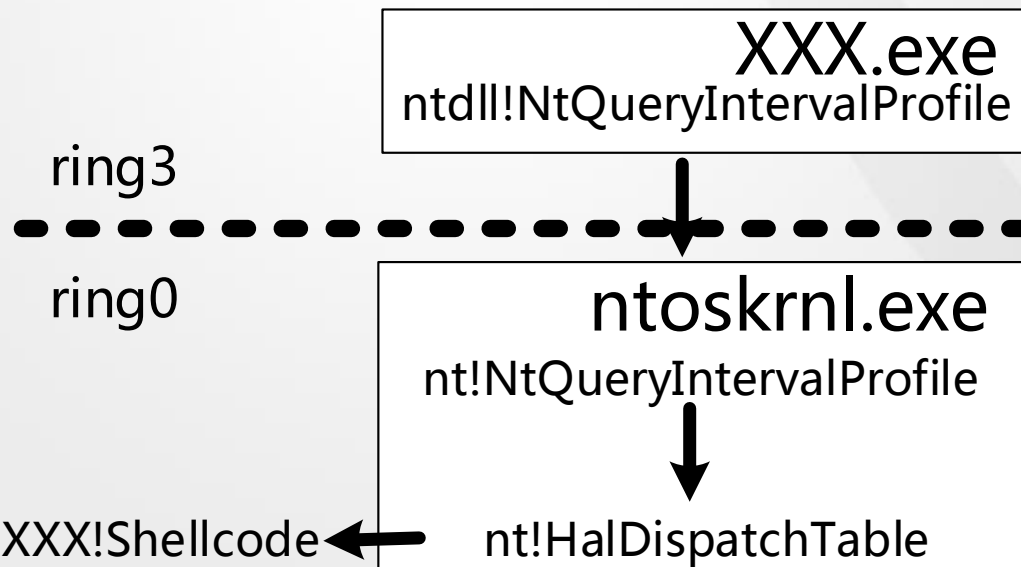
- nt!HalDispatchTable

```
NTSTATUS WINAPI NtQuerySystemInformation(  
    _In_     SYSTEM_INFORMATION_CLASS SystemInformationClass,  
    _Inout_ PVOID                     SystemInformation,  
    _In_     ULONG                     SystemInformationLength,  
    _Out_opt_ PULONG                  ReturnLength  
);
```

Q2: What to write?

- Userland shellcode address

Q3: What can we do now?

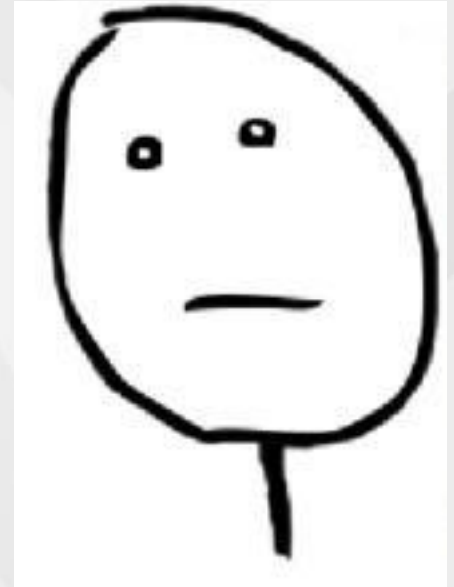
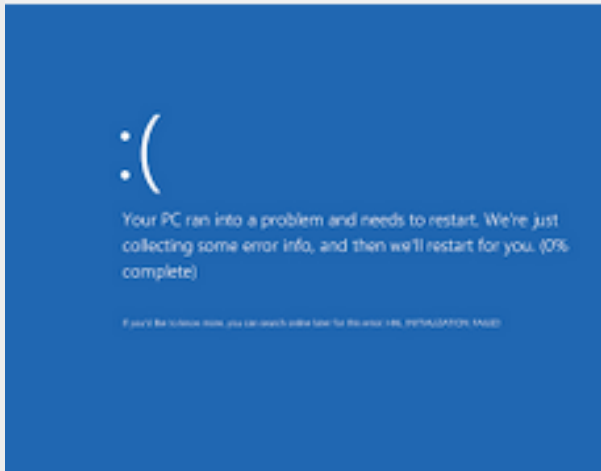


Now..

- Forbid low integrity user

```
loc_14043EF36:          ; jumtable 0000000014043EADA case 11
movzx  ecx, r10b
call   ExIsRestrictedCaller
test   eax, eax
jnz    loc_1405D4CBB
```

- SMEP



Where to write?

gSharedInfo



What to write?

Basic object: Window

```
typedef struct tagWNDCLASSW {
    UINT        style;
    WNDPROC     lpfnWndProc;
    int         cbClsExtra;
    int         cbWndExtra;
    HINSTANCE   hInstance;
    HICON       hIcon;
    HCURSOR     hCursor;
    HBRUSH      hbrBackground;
    LPCWSTR     lpstrMenuName;
    LPCWSTR     lpstrClassName;
} WNDCLASSW, *PWNDCLASSW;
```

```
HWND WINAPI CreateWindowExW(
    __in DWORD dwExStyle,
    __in_opt LPCWSTR lpClassName,
    __in_opt LPCWSTR lpWindowName,
    __in DWORD dwStyle,
    __in int X,
    __in int Y,
    __in int nWidth,
    __in int nHeight,
    __in_opt HWND hWndParent,
    __in_opt HMENU hMenu,
    __in_opt HINSTANCE hInstance,
    __in_opt LPVOID lpParam);
```

tagWND

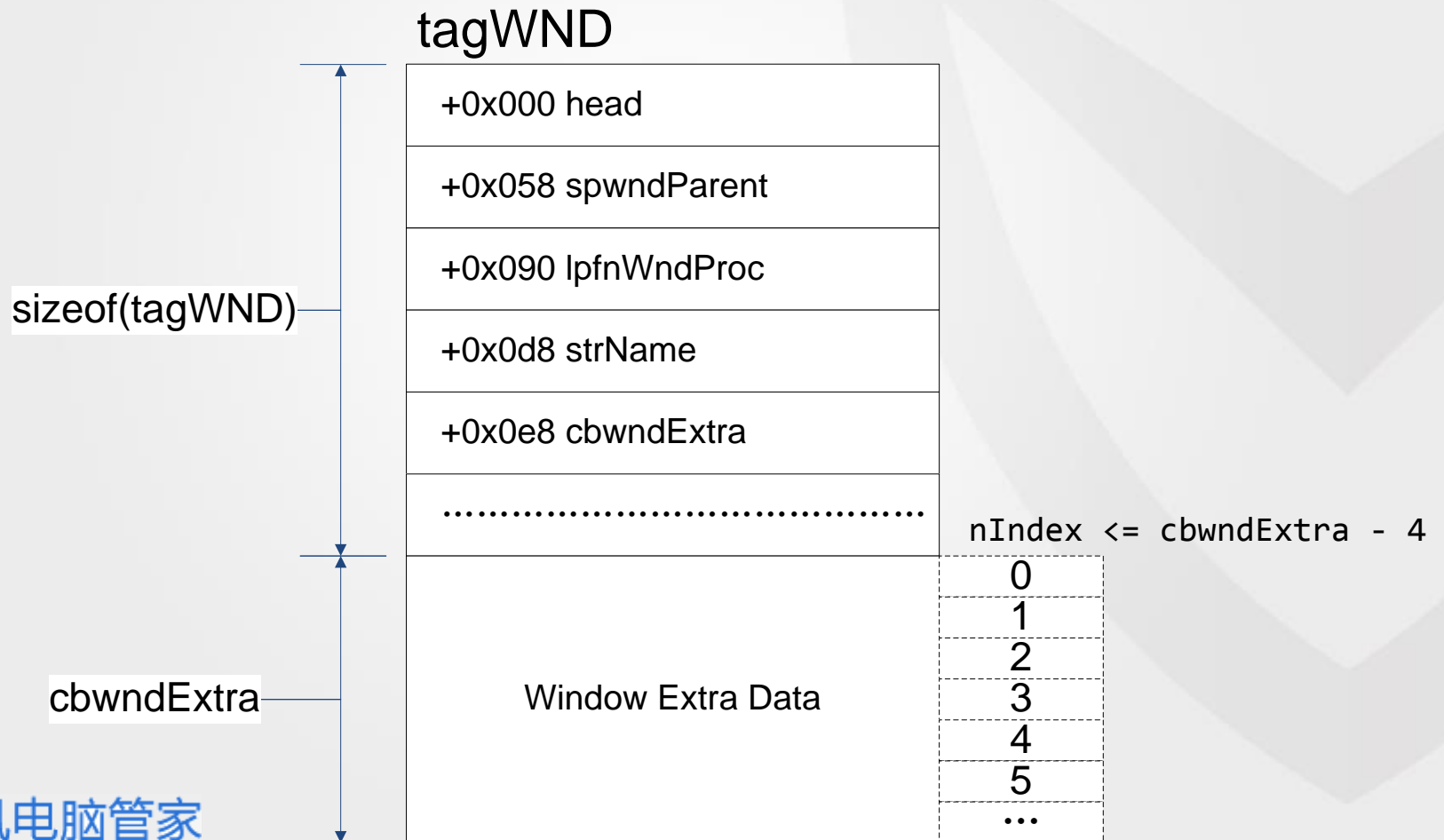
+0x000 head
+0x058 spwndParent
+0x090 lpfnWndProc
+0x0d8 strName
+0x0e8 cbwndExtra
.....
Window Extra Data

Window Extra Data

● Two APIs:

```
LONG WINAPI SetWindowLongW( HWND hWnd, int nIndex, LONG dwNewLong);
```

```
LONG WINAPI GetWindowLongW( HWND hWnd, int nIndex);
```

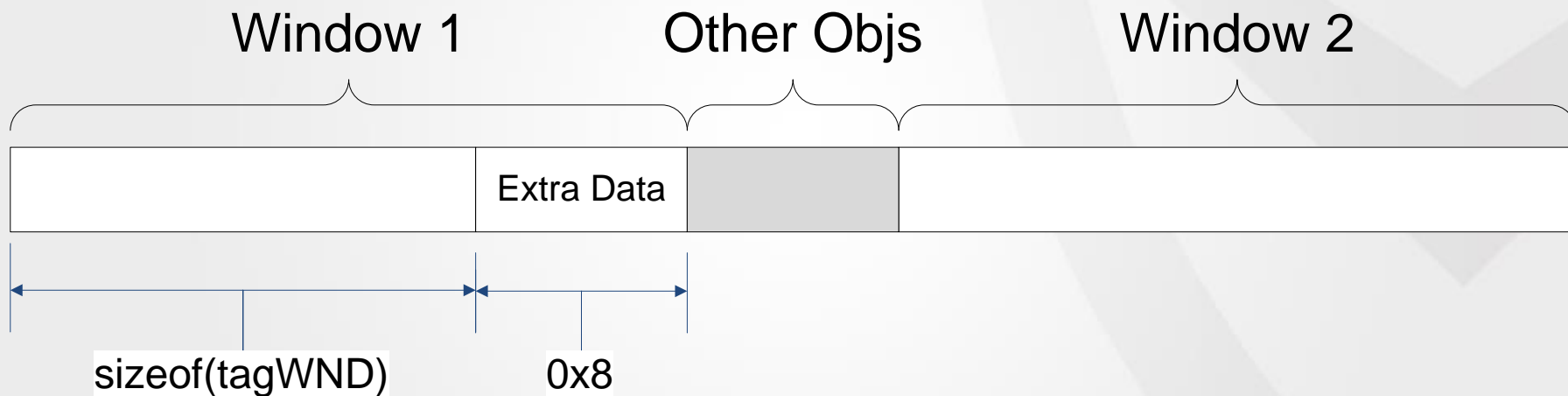


Normal Case

- `cbwndExtra = 0x8`

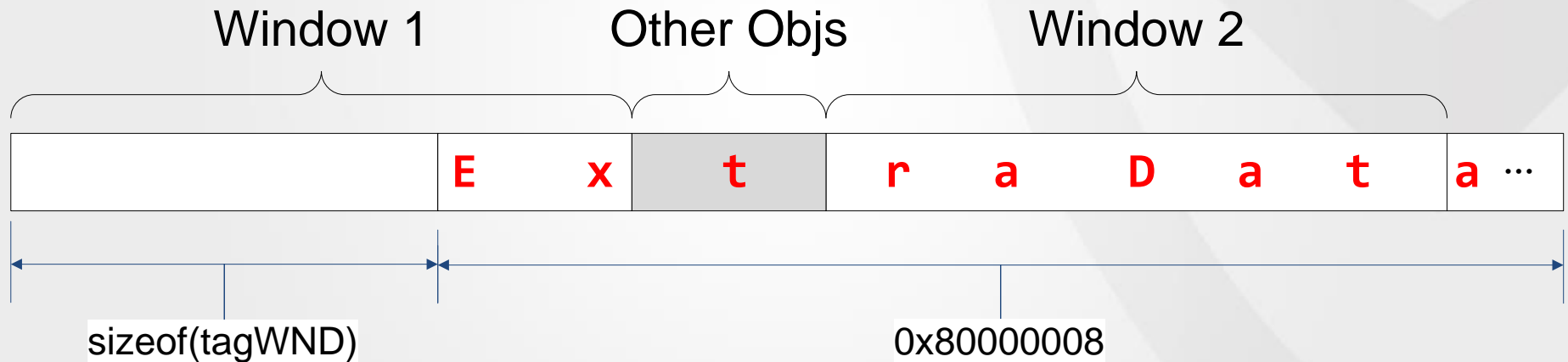
- Hex: `0x8`

- Bin: `0000 0000 0000 0000 0000 0000 0000 1000`
pos 31 15 0



If we change a bit...

- Bin: **1**000 0000 0000 0000 0000 0000 0000 1000
pos 31 15 0
- Hex: 0x80000008
- cbwndExtra = 0x80000008



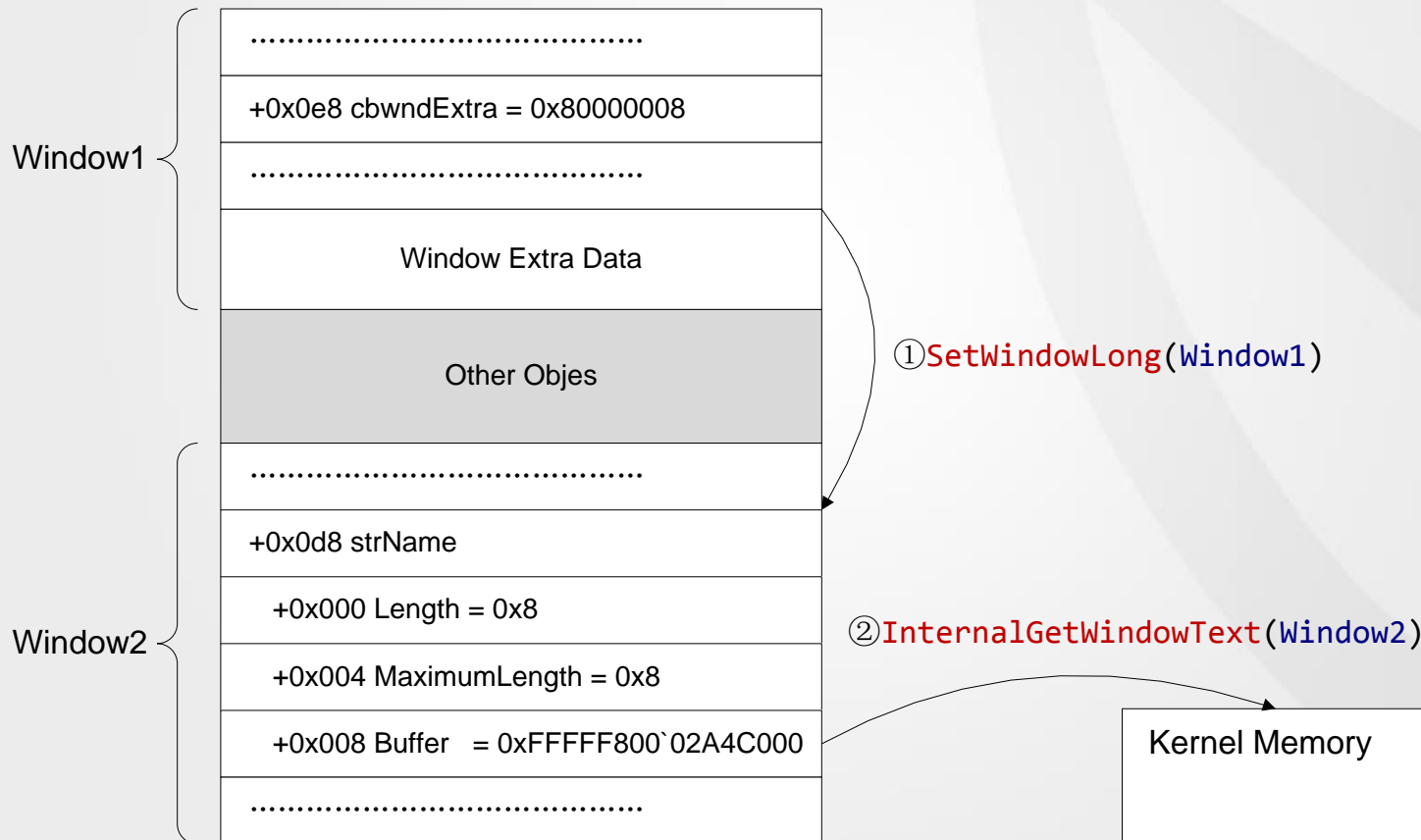
What can we do now?

Read from anywhere

● Two APIs:

```
LONG WINAPI SetWindowLongW( HWND hWnd, int nIndex, LONG dwNewLong);
```

```
int WINAPI InternalGetWindowText( HWND hWnd, LPWSTR lpString, int nMaxCount);
```

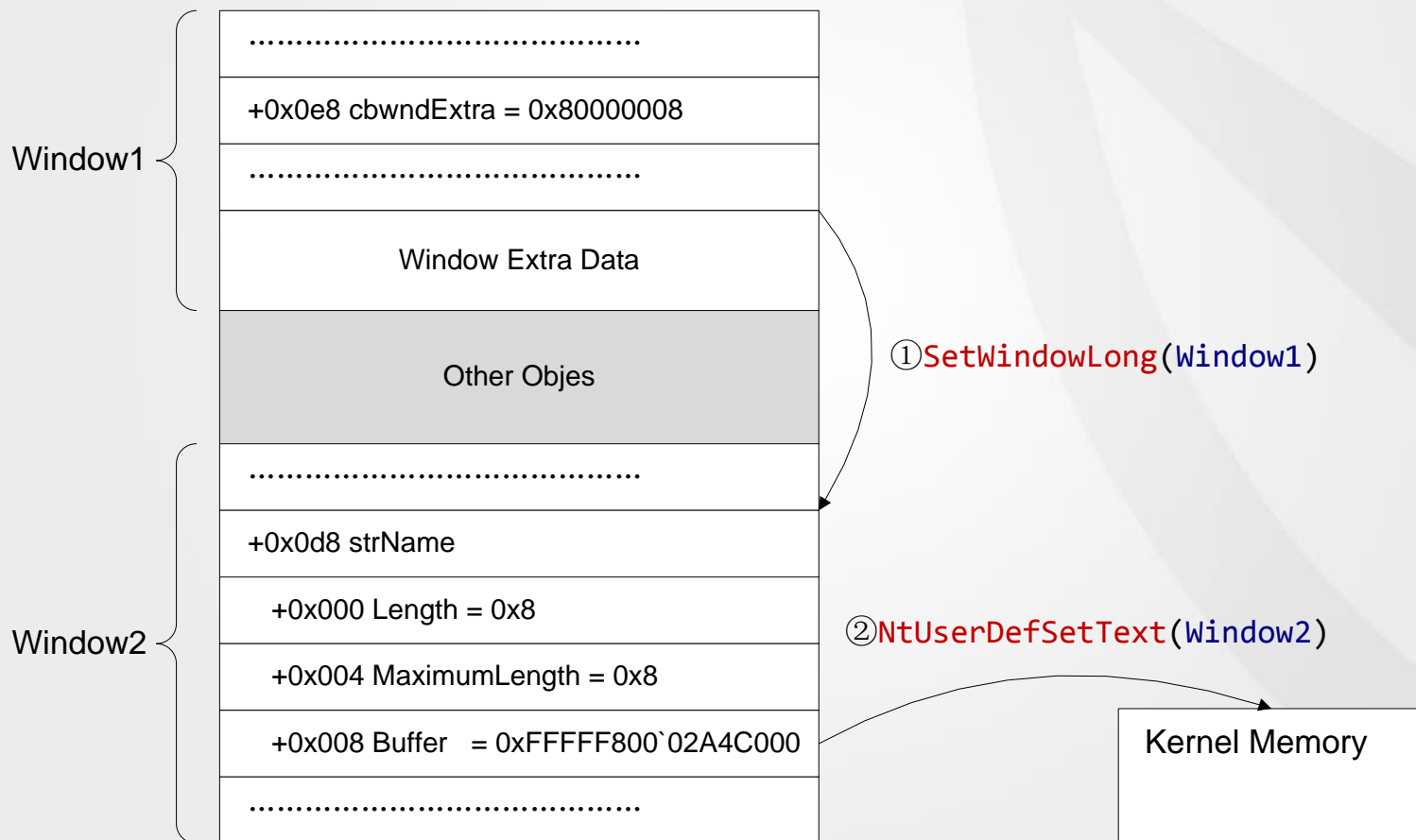


Write to anywhere

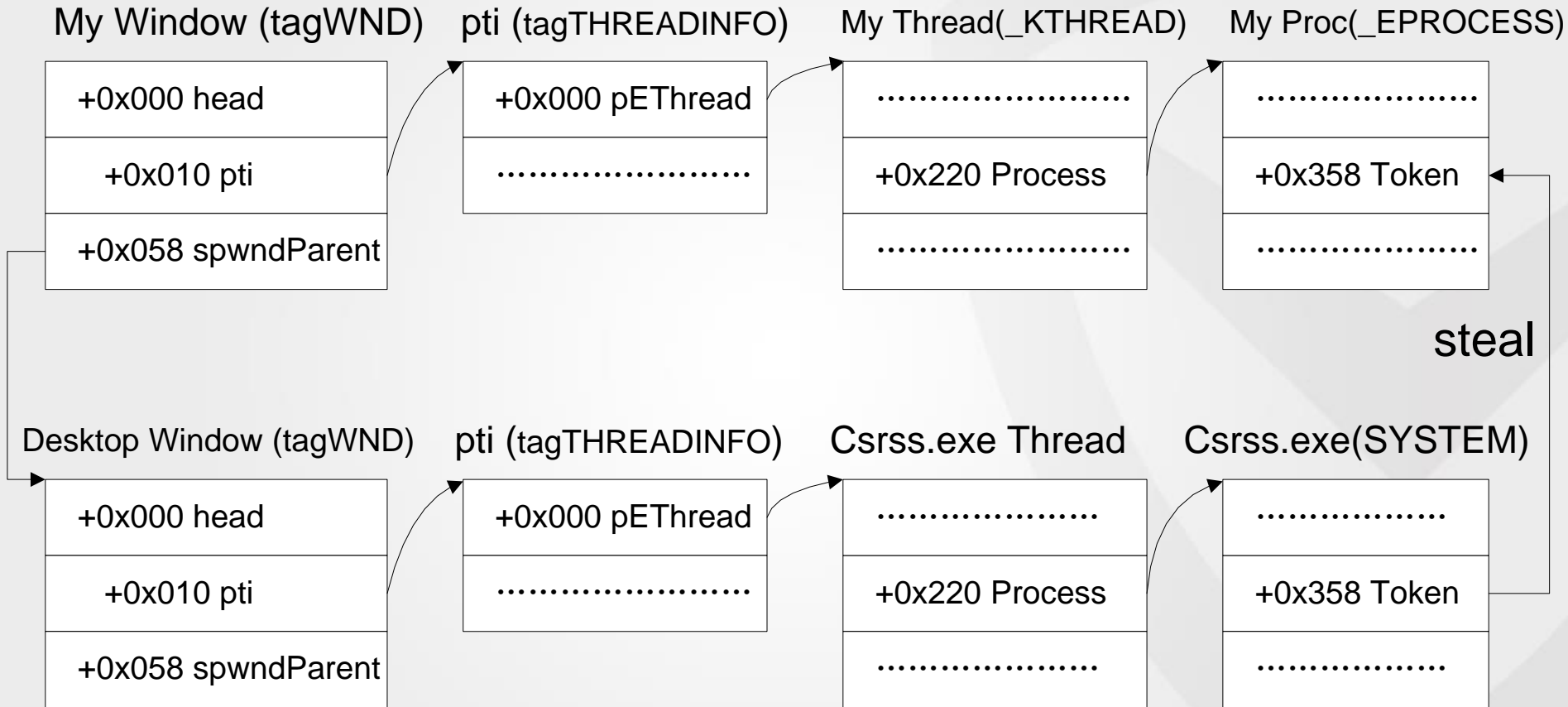
● Two APIs:

```
LONG WINAPI SetWindowLongW( HWND hWnd, int nIndex, LONG dwNewLong);
```

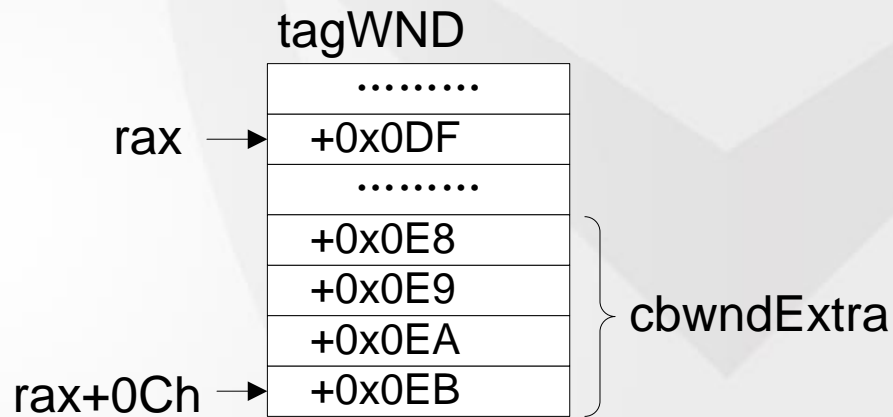
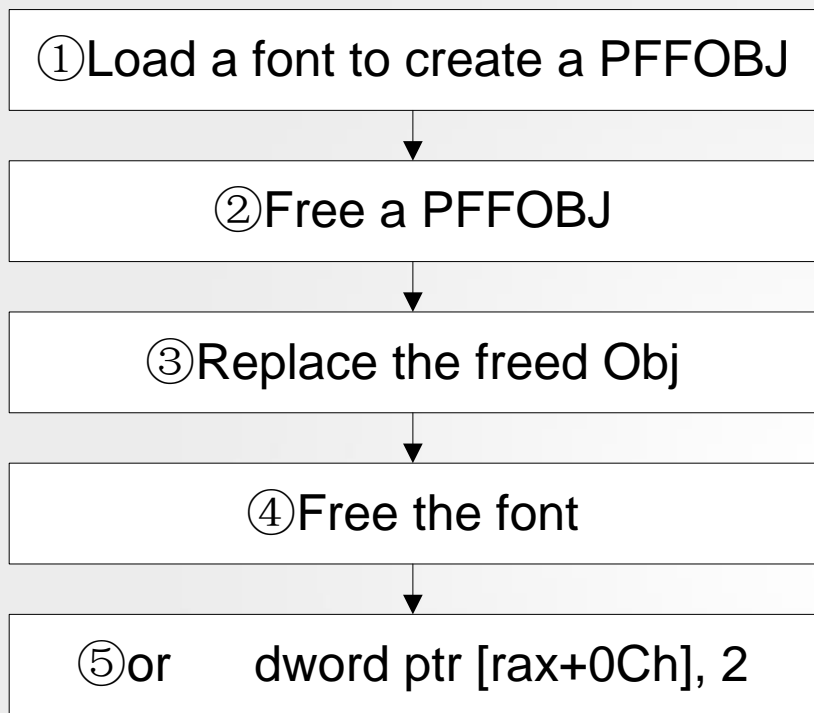
```
BOOL NtUserDefSetText( HWND hWnd, PLARGE_STRING pstrText );
```



Steal SYSTEM Token

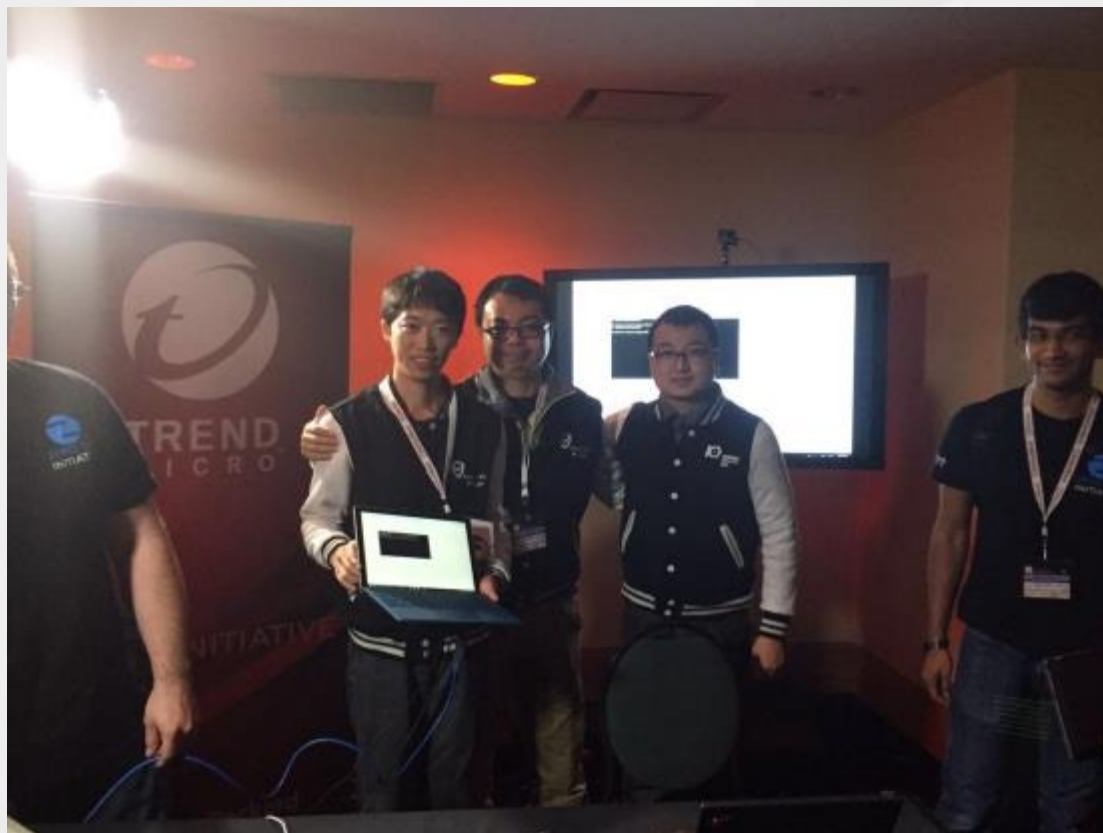


Real Case: CVE-2016-0174



offset	0xEB	0xEA	0xE9	0xE8
● Bin:	0000 0000	0000 0000	0000 0000	0000 0000
● Bin:	0000 00 1 0	0000 0000	0000 0000	0000 0000
● cbwndExtra:	0	→	0x20000000	

Pwn2Own 2016 Flash



Other Case

dec dword ptr [rax]

- | offset | 0xEB | 0xEA | 0xE9 | 0xE8 |
|---------------|----------------|-----------|-----------|-----------|
| ● Bin: | 0000 0000 | 0000 0000 | 0000 0000 | 0000 0000 |
| ● Bin: | FFFF FFFF | FFFF FFFF | FFFF FFFF | FFFF FFFF |
| ● cbwndExtra: | 0-1=0xFFFFFFFF | | | |

inc dword ptr [r10+8]

- | offset | 0xEB | 0xEA | 0xE9 | 0xE8 |
|---------------|----------------|-----------|-----------|-----------|
| ● Bin: | 0000 0000 | 0000 0000 | 0000 0000 | 0000 0000 |
| ● Bin: | 0000 1000 | 0000 0000 | 0000 0000 | 0000 0000 |
| ● cbwndExtra: | 0 → 0x80000000 | | | |

How to exploit 0 → 1

Q1: Where to write?

- tagWND.cbwndExtra

Q2: What to write?

- A big value

Q3: What can we do now?

- Read from anywhere
- Write any value to anywhere
- Steal csrss.exe`s token



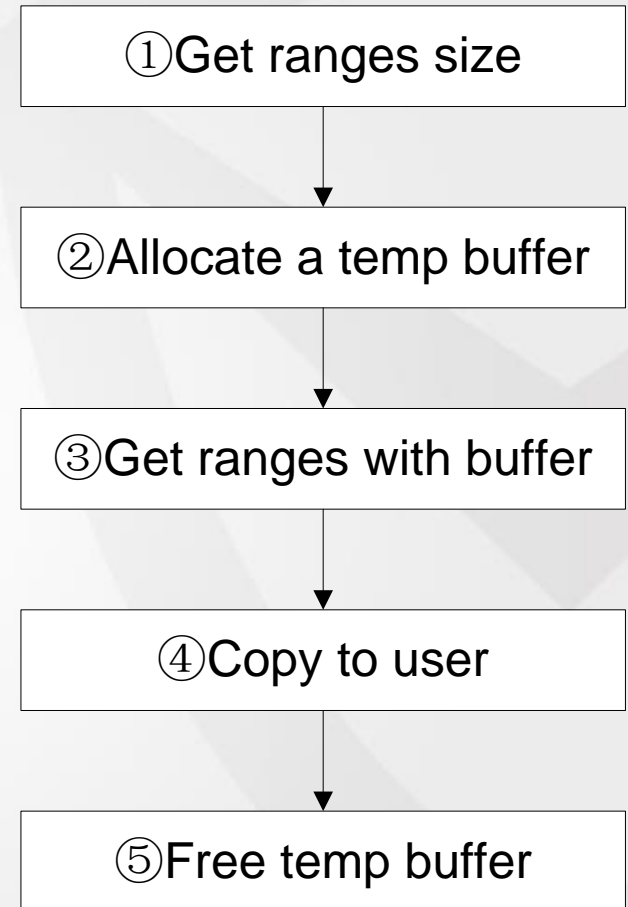
1 → 0



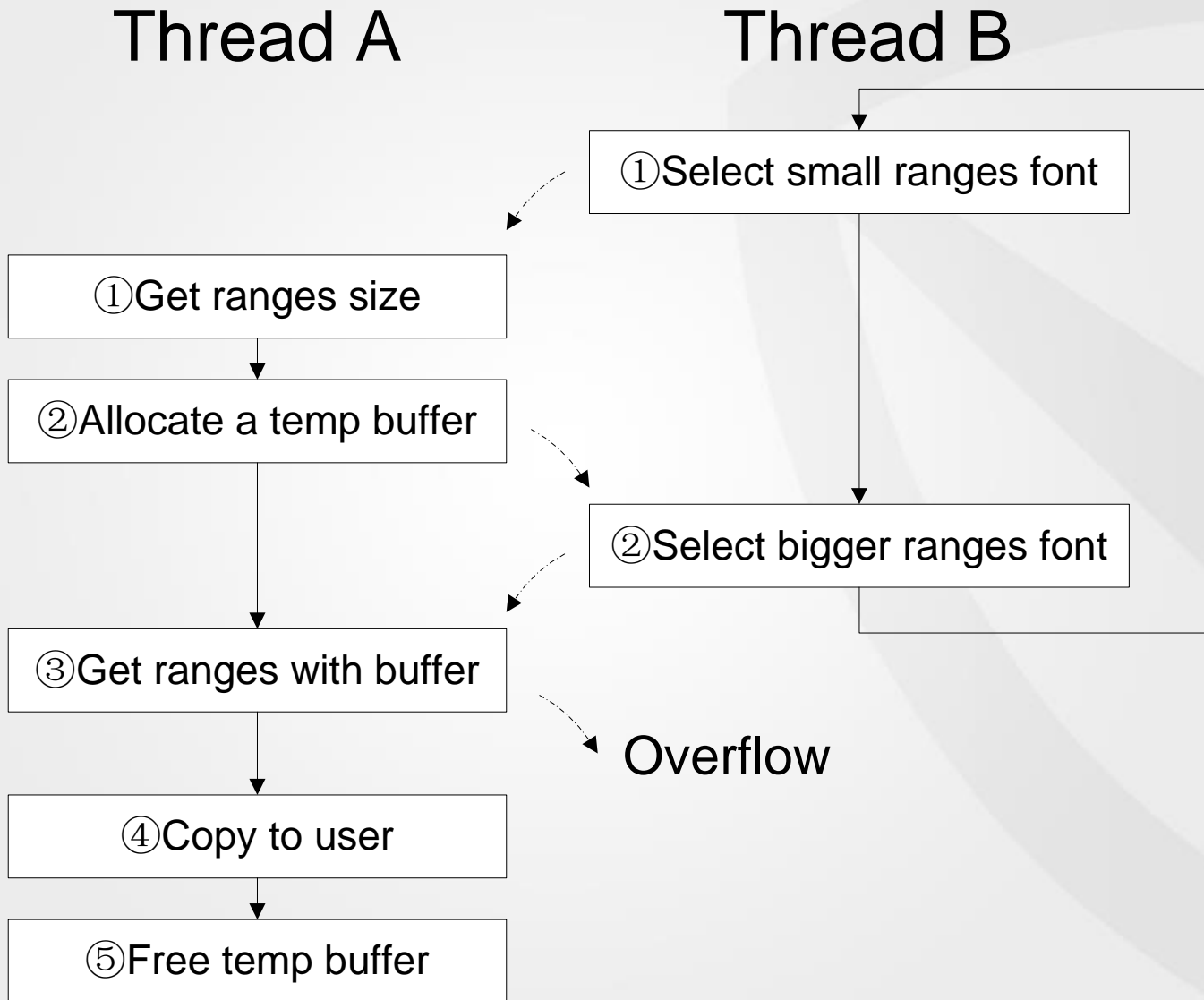
The Function - CVE-2016-3355

```
DWORD NtGdiGetFontUnicodeRanges(HDC hdc, LPGLYPHSET lpgs)
{
    DWORD PreSize;
    DWORD PosSize;
    DWORD ReturnSize;
    PVOID pTmpBuf;

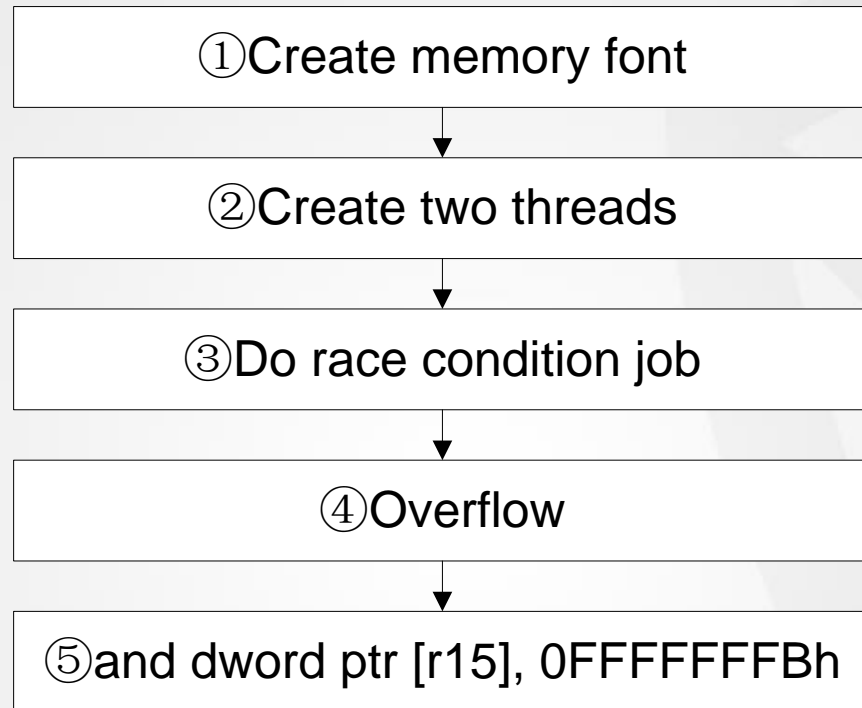
    ReturnSize = 0;
    PreSize = GreGetFontUnicodeRanges(hdc, 0);
    if ( PreSize && lpgs )
    {
        pTmpBuf = AllocFreeTmpBuffer(PreSize);
        if ( pTmpBuf )
        {
            PosSize = GreGetFontUnicodeRanges(hdc, pTmpBuf);
            if ( PosSize && PreSize == PosSize )
            {
                ProbeAndWriteBuffer(lpgs, pTmpBuf, PreSize);
                ReturnSize = PreSize;
            }
            FreeTmpBuffer(pTmpBuf);
        }
    }
    return ReturnSize;
}
```



The Problem



My Exploit



● Hex: 0xFFFFFFFFFB

● Bin: 1111 1111 1011
Pos 31 7 3 2 1 0

My Exploit

- Length? 0x4 → 0x0, 0x14 → 0x10

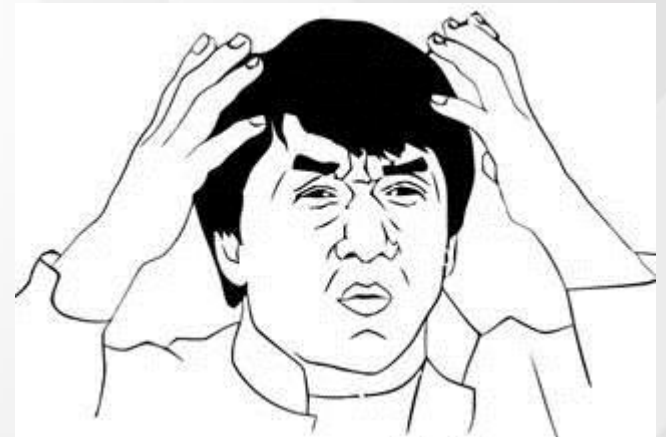
- Flag?

```
1: kd> dt win32k!tagWND
```

```
.....  
+0x014 bHasHorizontalScrollbar : Pos 2, 1 Bit  
.....  
+0x018 bStartPaint           : Pos 2, 1 Bit  
.....  
+0x01c bWS_EX_NOPARENTNOTIFY : Pos 2, 1 Bit  
.....  
+0x0ac bRedirectedForPrint   : Pos 2, 1 Bit
```

- Type?

```
TYPE_CLIPDATA (6) → TYPE_MENU (2)  
TYPE_MONITOR (0xC) → TYPE_ACCELTABLE (0x8)  
TYPE_KBDFILE (0xE) → TYPE_DDECONV (0xA)
```



The Refcount!

● Win32k object:

```
1: kd> dt win32k!tagWND -b
+0x000 head          : _THRDESKHEAD
+0x000 h             : Ptr32
+0x004 cLockObj     : Uint4B
+0x008 pti          : Ptr32
+0x00c rpdesk       : Ptr32
+0x010 pSelf        : Ptr32
```

```
1: kd> dt win32k!tagMENU -b
+0x000 head          : _PROCDESKHEAD
+0x000 h             : Ptr32
+0x004 cLockObj     : Uint4B
+0x008 hTaskWow     : Uint4B
+0x00c rpdesk       : Ptr32
+0x010 pSelf        : Ptr32
```

```
1: kd> dt win32k!tagMONITOR -b
+0x000 head          : _HEAD
+0x000 h             : Ptr32
+0x004 cLockObj     : Uint4B
```

● Modify cLockObj:

```
PVOID FASTCALL HMAssignmentLock(PVOID *ppobj, PVOID pobj);
```

```
PVOID FASTCALL HMAssignmentUnlock( PVOID *ppobj );
```

● Check cLockObj:

```
BOOL HMMarkObjectDestroy( PVOID pobj);
```

Basic object: Menu

● Two APIs:

```
HMENU CreateMenu ();
```

```
BOOL AppendMenu(HMENU hMenu,UINT uFlags,UINT_PRT uIDNewItem,LPCWSTR lpNewItem);
```

Menu1 (tagMENU)

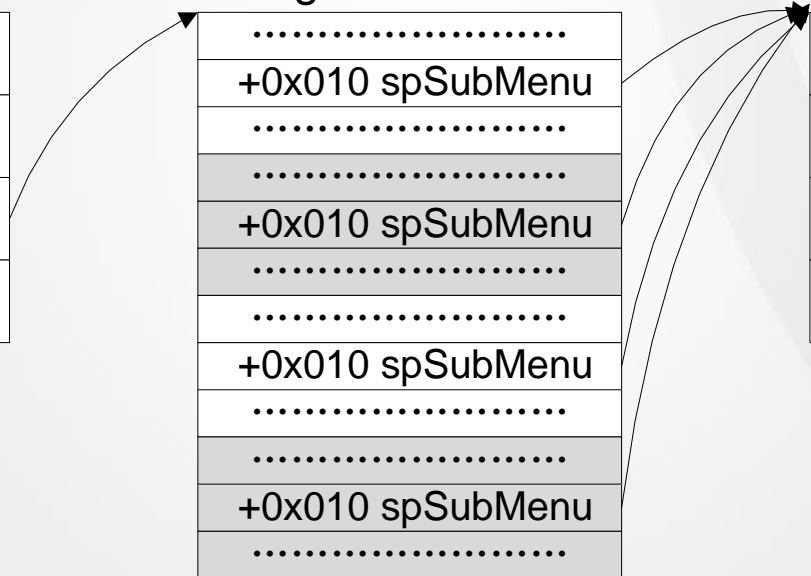
+0x008 cLockObj
+0x034 cItems = 4
+0x050 rgItems
+0x068 dwMenuData

4 tagITEMs

.....
+0x010 spSubMenu
.....
.....
+0x010 spSubMenu
.....
.....
+0x010 spSubMenu
.....
.....
+0x010 spSubMenu
.....

Menu2 (tagMENU)

+0x008 cLockObj = 4
+0x034 cItems = 0
+0x050 rgItems = NULL
+0x068 dwMenuData



Make a Use-After-Free

● API:

```
BOOL WINAPI DestroyMenu ();
```

Menu2(tagMENU)

+0x008 cLockObj = 4
+0x034 cltems
+0x050 rgltems
+0x068 dwMenuData

Menu2(tagMENU)

+0x008 cLockObj = 0
+0x034 cltems
+0x050 rgltems
+0x068 dwMenuData

Freed Mem



and dword ptr [r15], 0FFFFFFFBh

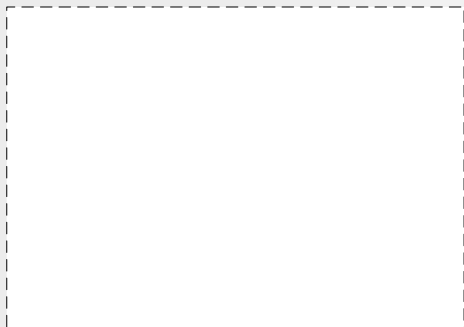
DestroyMenu(Menu2)

Take Pos

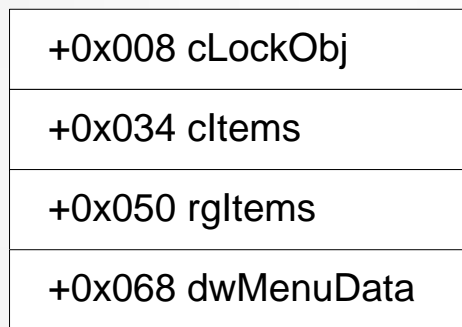
● Native API:

```
BOOL NtUserDefSetText( HWND hWnd, PLARGE_STRING pstrText );
```

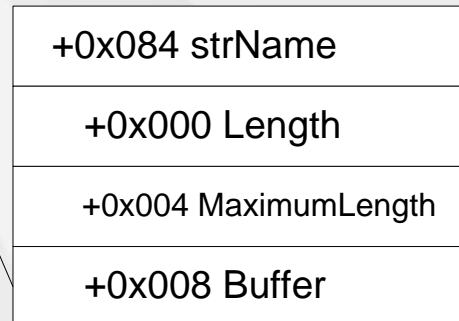
Freed Mem



Window Text (Fake tagMENU)



tagWND

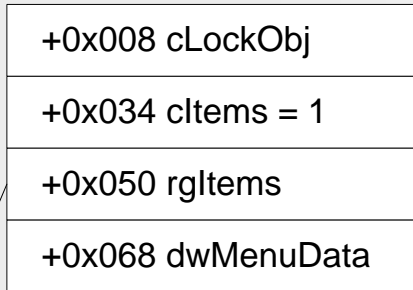


NtUserDefSetText

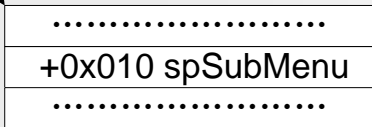


Fake tagMENU

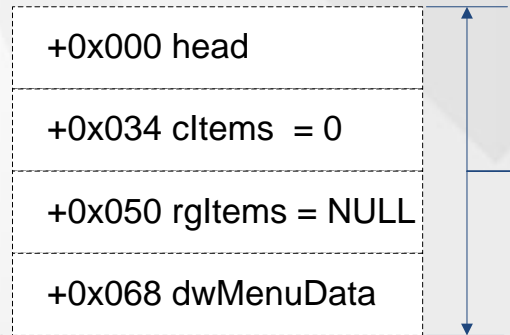
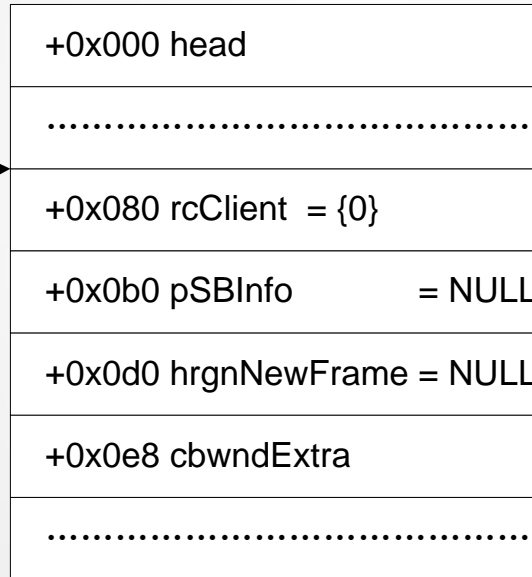
Window Text (Fake tagMENU)



1 tagITEM



Window Low (tagWND)



Virtual Menu

Change the cbwndExtra

● API:

```
MenuInfo.fMask      = MIM_MENUDATA | MIM_APPLYTOSUBMENUS;  
MenuInfo.dwMenuData = 0xFFFFFFFF;  
SetMenuInfo(Menu1, &MenuInfo);
```

Menu1(tagMENU)

+0x008 cLockObj
+0x034 cItems = 4
+0x050 rgItems
+0x068 dwMenuData

Window Text (Fake tagMENU)

+0x008 cLockObj
+0x034 cItems = 1
+0x050 rgItems
+0x068 dwMenuData

Window Low (tagWND)

+0x080 rcClient	+0x000 head
+0x0b0 pSBInfo	+0x034 cItems = 0
+0x0d0 hrgnNewFrame	+0x050 rgItems = NULL
+0x0e8 cbwndExtra	+0x068 dwMenuData

4 tagITEMs

.....
+0x010 spSubMenu
.....
+0x010 spSubMenu
.....
+0x010 spSubMenu
.....
+0x010 spSubMenu
.....
+0x010 spSubMenu
.....

1 tagITEM

.....
+0x010 spSubMenu
.....

Let`s rule them all!

 Windows 10

 Windows Server 2016

 Windows 8

 Windows Server 2012

 Windows 7

 Windows Server 2008 R2

 Windows Vista™

 Windows Server 2008

 Microsoft Windows xp

 Microsoft Windows Server 2003

 Microsoft Windows 2000

How to exploit 1 → 0

Q1: Where to write?

- tagMENU.cLockObj

Q2: What to write?

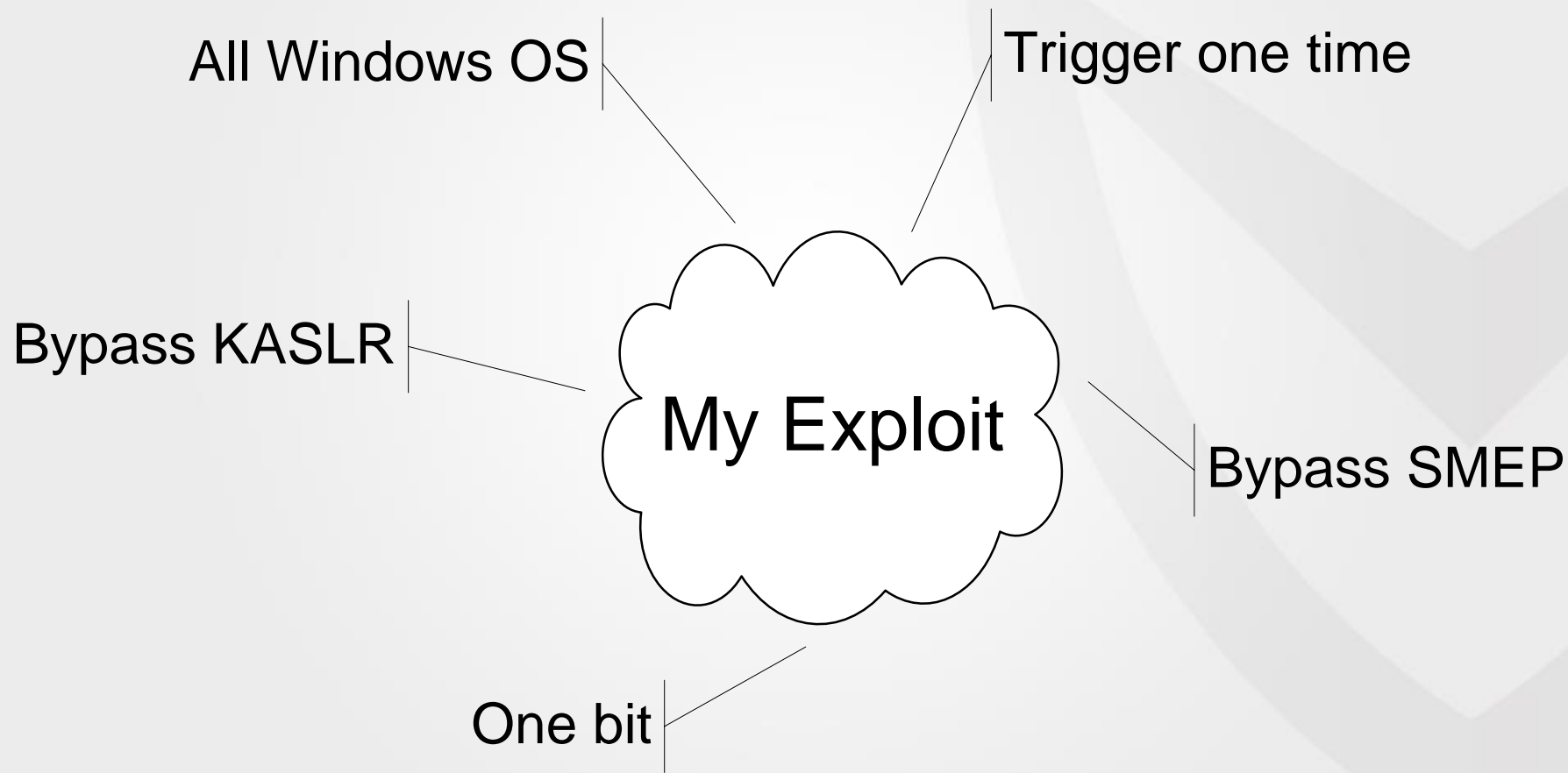
- 0

Q3: What can we do now?

- Make a UAF
- Control the pointer
- Write a big value to tagWND.cbwndExtra



Summary



Q & A

Pwn2Own 🍌 hotmail.com