中国互联网安全领袖峰会
Cyber Security Summit

攻破 Windows 的马奇诺防线

张云海
绿盟科技
研究员

1 Windows 的缓解措施

2 功能重用攻击

3 The Lord of the Edge: The Two Browsers

4 The Lord of the Edge: The Return of The God Mode

5 The Lord of the Edge: The Shims of the Edge

Windows 的缓解措施

缓解措施是什么？

What is a Security Mitigation?

- A feature to disrupt exploitation.
- Mitigations make certain exploitation techniques and vulnerability classes harder or impossible to use.
- Different class of mitigations:
  - **Hard mitigations:** Harder or impossible to bypass. Typically disrupts an entire vulnerability class.
  - **Soft mitigations:** Makes exploitation harder but can be bypassed with stronger primitives.
  - **Tactical mitigations:** Aimed at disrupting specific exploit techniques.

控制流完整性缓解措施

数据执行保护 (DEP)

控制流防护 (CFG)

返回流防护 (RFG)

代码完整性缓解措施

任意代码防护 (ACG)

代码完整性防护 (CIG)

辅助性缓解措施

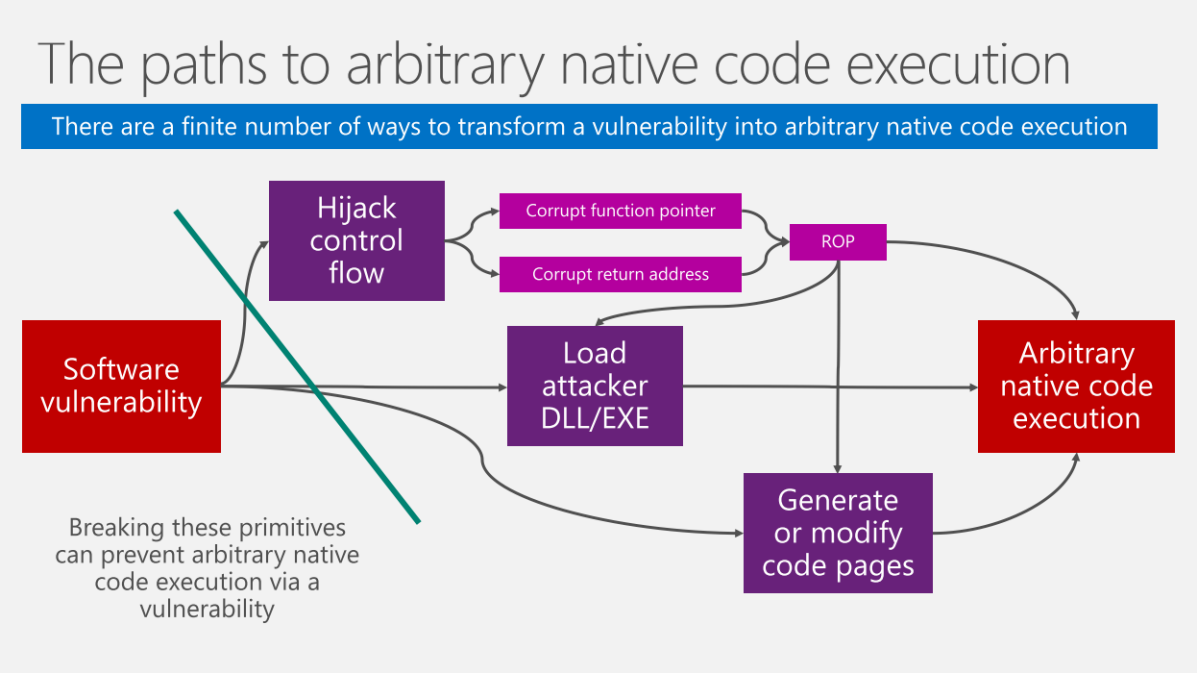
子进程策略

地址空间布局随机化 (ASLR)

SEHOP/SafeSEH

堆随机化和元数据保护

缓解措施的弱点


The paths to arbitrary native code execution
There are a finite number of ways to transform a vulnerability into arbitrary native code execution
Hijack control flow
Corrupt function pointer
Corrupt return address
ROP
Software vulnerability
Load attacker DLL/EXE
Arbitrary native code execution
Generate or modify code pages
Breaking these primitives can prevent arbitrary native code execution via a vulnerability

功能重用攻击 Feature Reuse Attack

功能重用攻击是什么？

现代操作系统与应用中包含有众多的功能

其中的部分功能具有特别的能力

通常这些能力在敏感应用中是受限的

解除这些限制就可以重用这些功能来执行期望的操作

The Lord of the Edge: The Two Browsers

Windows 10 有两个浏览器

Internet Explorer

Microsoft Edge

Microsoft Edge 更快更安全

Windows 10 Browsing Engines

Microsoft Edge: EdgeHTML

- Interoperability for Windows 10
- Up-to-date web engine
- Software as a service for modern websites
- More secure, with no binary extensions

Internet Explorer 11: MSHTML

- Compatibility for Windows 7, Windows 8.1, Windows 10
- Versioned "document modes"
  IE11 | IE10 | IE9 | IE8 | IE7 | IE5.5
- Software as a product for web apps, intranet sites
- Compatible with ActiveX controls, binary extensions

Internet Explorer 11 兼容性更好

Windows 10 Browsing Engines

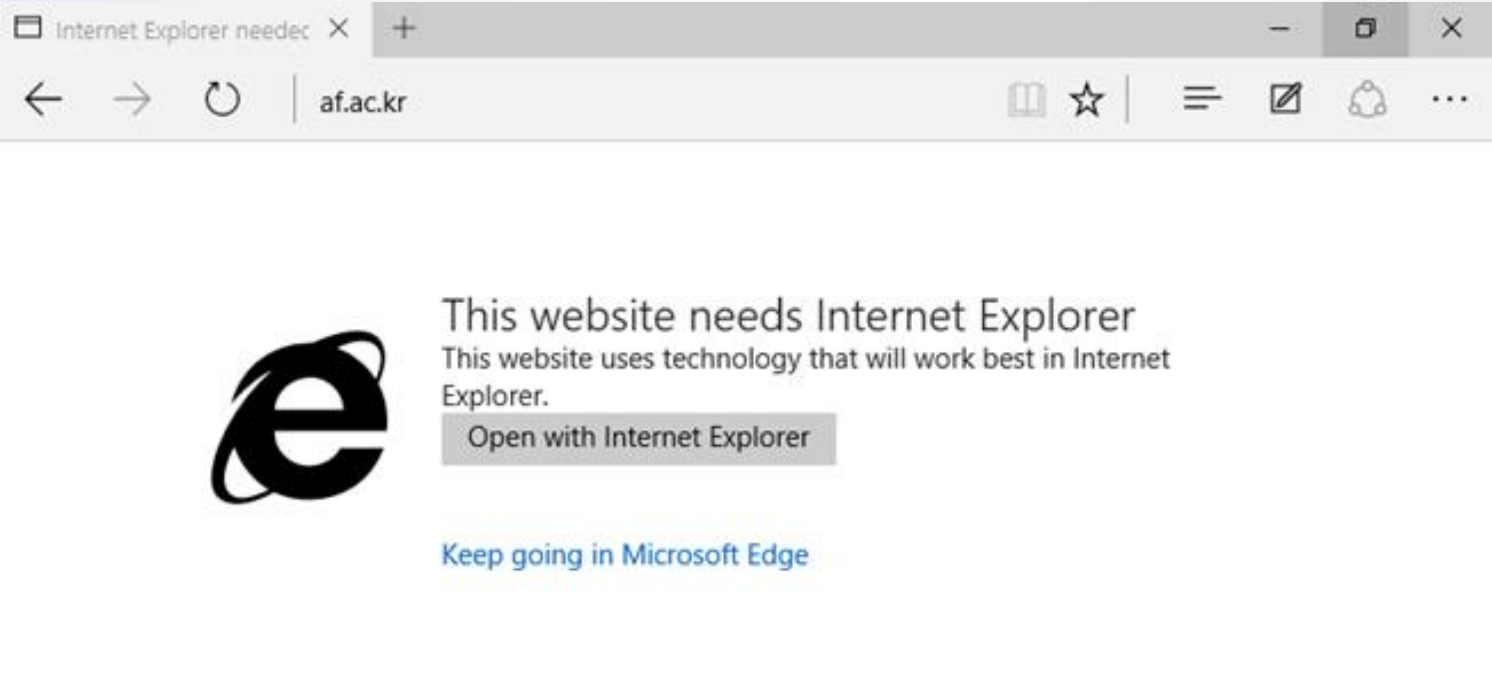
Microsoft Edge: EdgeHTML

- Interoperability for Windows 10
- Up-to-date web engine
- Software as a service for modern websites
- More secure, with no binary extensions

Internet Explorer 11: MSHTML

- Compatibility for Windows 7, Windows 8.1, Windows 10
- Versioned "document modes"
  IE11 | IE10 | IE9 | IE8 | IE7 | IE5.5
- Software as a product for web apps, intranet sites
- Compatible with ActiveX controls, binary extensions

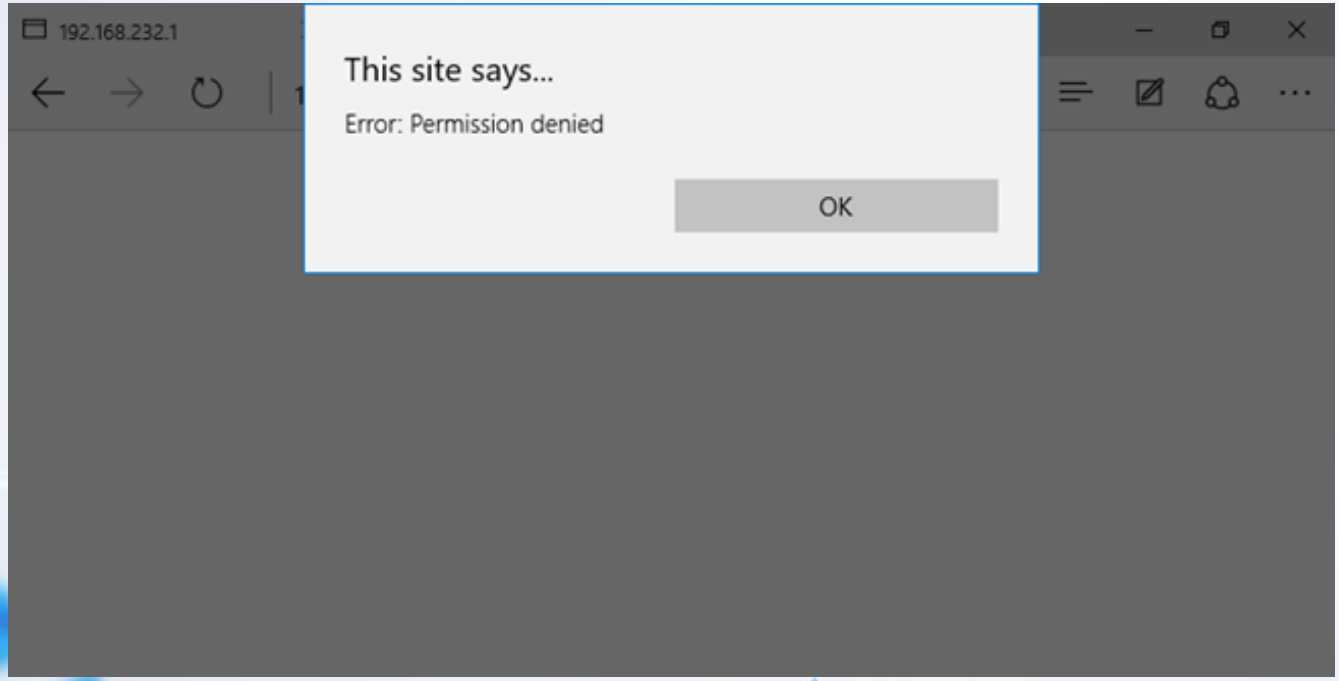
Microsoft Edge 的兼容性处理



Microsoft Edge 的兼容性处理

```
LaunchIE = function (automated)
{
    window.external.LaunchIE(getFullUrl(), automated);
}
```

限制1：只能在 NeedIE 页面中执行



限制1：只能在 NeedIE 页面中执行

```
BOOL __stdcall CBrowserTab::CWPCHost::CExternalDispatch::IsNeedIEPage(const unsigned __int16 *lpUrl)
{
  int v2; // esi@1
  int v3; // eax@1

  v2 = IsErrorUrl(lpUrl);
  v3 = StrStrIW(lpUrl, L"/assets/errorpages/needie.html") != 0;
  return v2 && v3;
}
```

```
if ( !StrCmpNW(v1, L"ms-appx-web://", 14) )
  v2 = StrStrIW(v1, L"/assets/errorpages/") != 0;
return v2;
```

解决方案：修改当前页面的 URL

ms-appx-web://microsoft.microsoftedge/assets/errorpages/needie.html

限制2：重定向策略

```
__int32 __stdcall CShdocvwBroker::LaunchIE(CShdocvwBroker *this, const unsigned __int16 *url, int a3)
{
  __int32 status; // esi@1 MAPDST
  IUnknownVtbl *v4; // esi@4
  int v8; // [sp+0h] [bp-10h]@4
  int policy; // [sp+8h] [bp-8h]@1
  IUnknown *pUnk; // [sp+Ch] [bp-4h]@3 MAPDST

  status = 0x80070005;
  policy = 0;
  if ( LCIEGetRedirectionPolicyForURL(url, 0, 1, 0x10000u, 0, (unsigned __int32 *)&policy, 0) >= 0
    && policy & 0x3C000000 )
  {
    pUnk = 0;
    status = GetBrowserBrokerInterface((struct IBrowserBrokerFactory **)&pUnk);
    if ( status >= 0 )
    {
      CoAllowSetForegroundWindow(pUnk, 0);
      v4 = pUnk->lpVtbl;
      __guard_check_icall_fptr(pUnk->lpVtbl[6].Release);
      status = ((int (__stdcall *)(IUnknown *, const unsigned __int16 *, int))v4[6].Release)(pUnk, url, a3);
      if ( &v8 != &v8 )
        __fastfail(4u);
    }
    ATL::CComPtr<IOpenServiceActivity>::~CComPtr<IOpenServiceActivity>(&pUnk);
  }
  return status;
}
```

限制2：重定向策略

Enterprise Site List

Intranet

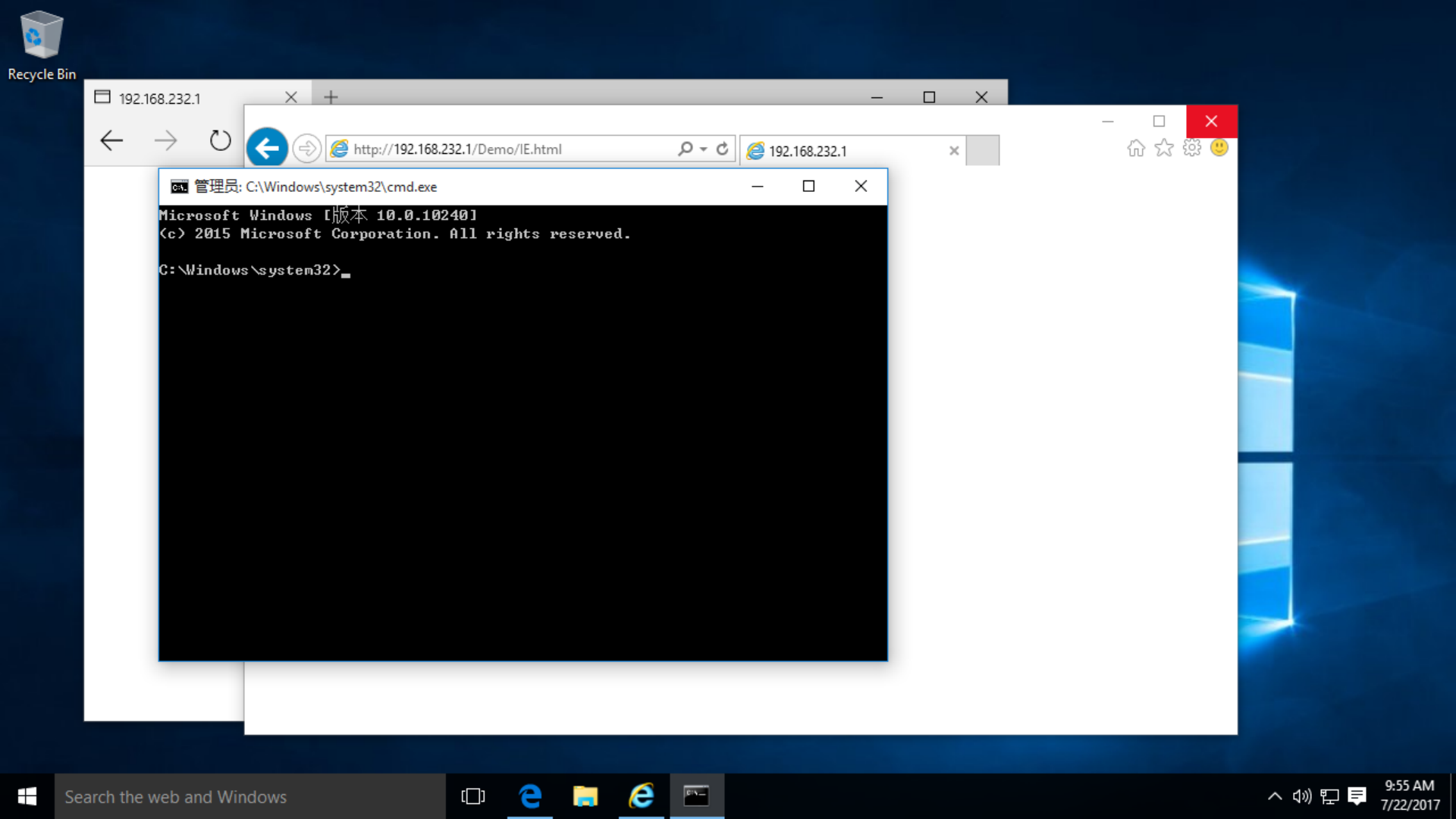
Compatibility view list

解决方案：利用 URL 跳转

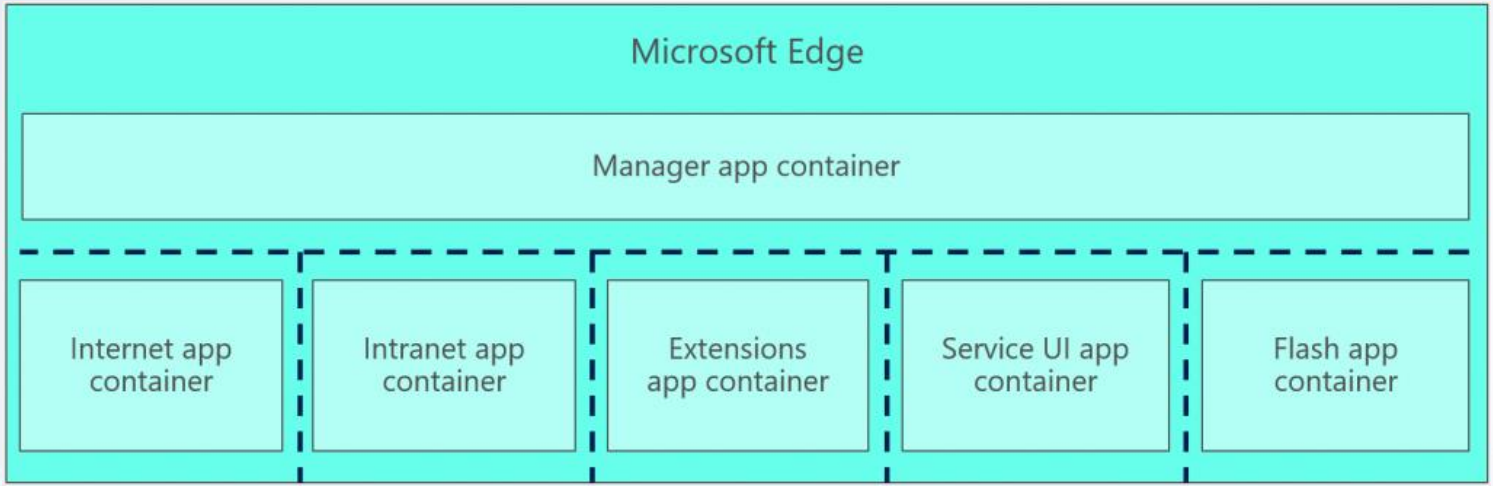
Compatibility view list 中有上千个网站

这些网站中的任意 URL 都符合重定向策略

部分网站中存在特殊的 URL 可以跳转到指定的目标


Recycle Bin
192.168.232.1
http://192.168.232.1/Demo/IE.html
192.168.232.1
管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.
C:\Windows\system32>
Search the web and Windows
9:55 AM
7/22/2017

问题修复


Microsoft Edge
Manager app container
Internet app container
Intranet app container
Extensions app container
Service UI app container
Flash app container

The Lord of the Edge: The Return of The God Mode

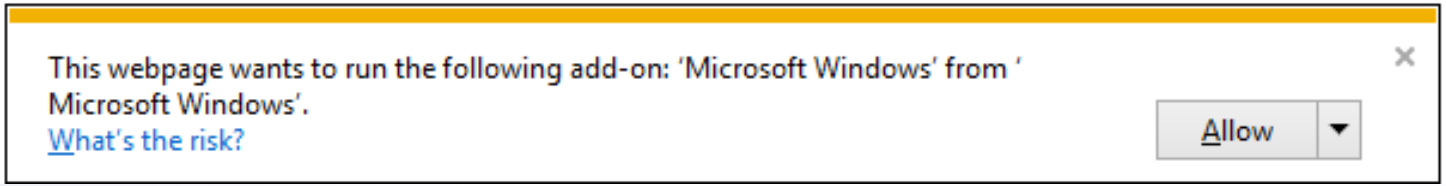
God Mode 是什么？

JavaScript / VBScript 可以实例化一些敏感的 ActiveX 对象

- Shell.Application
- WScript.Shell
- Scripting.FileSystemObject
- ADODB.Stream
- ……

God Mode 是什么？

此功能在浏览器中是受限的



God Mode 是什么？

修改特定的标志位可以解除此限制

```
int __thiscall COleScript::InSafeMode(COleScript *this, const struct _GUID *a2)
{
  signed int v2; // esi@1

  v2 = 0;
  if ( *((_DWORD *)this + 0x5D) & 0xB || !COleScript::IsUnsafeAllowed(a2) )
    v2 = 1;
  return v2;
}
```

God Mode 是什么？


black hat
USA 2014
Write Once, Pwn Anywhere
Yang Yu
Twitter: @tombkeeper

God Mode 是什么？

Bypassing Windows 8.1 Mitigations using Unsafe COM Objects

By James Forshaw, 25 June 2014

In October last year I was awarded the first $100,000 bounty for a Mitigation Bypass in Microsoft Windows. My original plan was to not discuss it in any depth until Microsoft had come up with a sufficient changes to reduce the impact of the bypass. However as other researchers have basically come up with variants of the same technique, some of which are publically disclosed with proof-of-concept code it seemed silly to not discuss my winning entry. So what follows is some technical detail about the bypass itself.

I am not usually known for finding memory corruption vulnerabilities, mainly because I don't go looking for them. Still I know my way around and so I knew the challenges I would face trying to come up with a suitable mitigation bypass entry. I realised that about the only way of having a successful entry would be to take a difficult to exploit memory corruption vulnerability and try and find a way of turning that into reliable code execution.

For that reason I settled on investigating the exploitation of a memory overwrite where the only value you could write was the number 0. Converting a 0 overwrite of this sort, while not impossible to exploit, certainly presents some challenges. I also stated that I could not disclose the existing contents of memory. If you have an information disclosure vulnerability then it is generally game over anyway, so I was confident that would not pass for a winning entry.

Microsoft Edge 不支持 ActiveX

MAY 6, 2015 11:00 AM

A break from the past, part 2: Saying goodbye to ActiveX, VBScript, attachEvent...

By Microsoft Edge Team

SHARE TWEET SHARE SHARE SKYPE

We recently posted "A break from the past: the birth of Microsoft's new web rendering engine", an in-depth look at the background and motivation behind building a new rendering engine to power Microsoft Edge. A key factor described was the ability to make a break from legacy Internet Explorer-specific technologies that had been built up over the years.

Microsoft Edge 不支持 ActiveX

Edge attack surface reduction

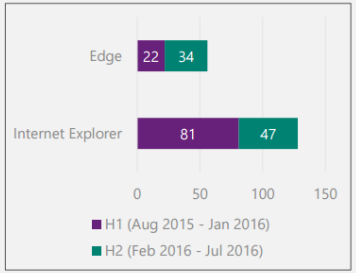
With the Edge browser, we also seized the opportunity to drastically reduce the attack surface exposed to the web

- ✓No legacy document modes
- ✓No legacy script engines (VBScript, JScript)
- ✓No Vector Markup Language (VML)
- ✓No Toolbars
- ✓No Browser Helper Objects (BHOs)
- ✓No ActiveX controls

In the past year

Edge had 56% fewer RCE CVEs compared to Internet Explorer

Internet Explorer RCE CVEs decreased 40% H/H



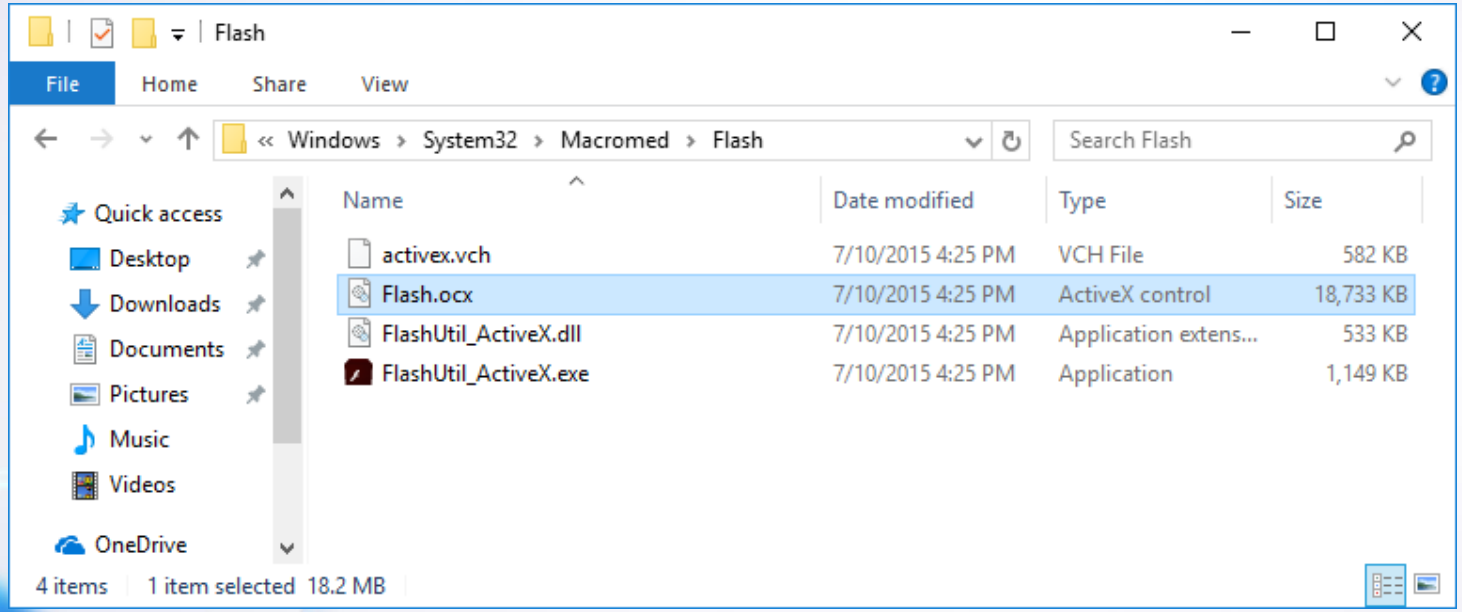
Tons of code was removed as a result!

Tactic	Applies to	First shipped
Eliminate entire classes of vulnerabilities	Edge on Windows 10	July, 2015 (Windows 10 RTM)

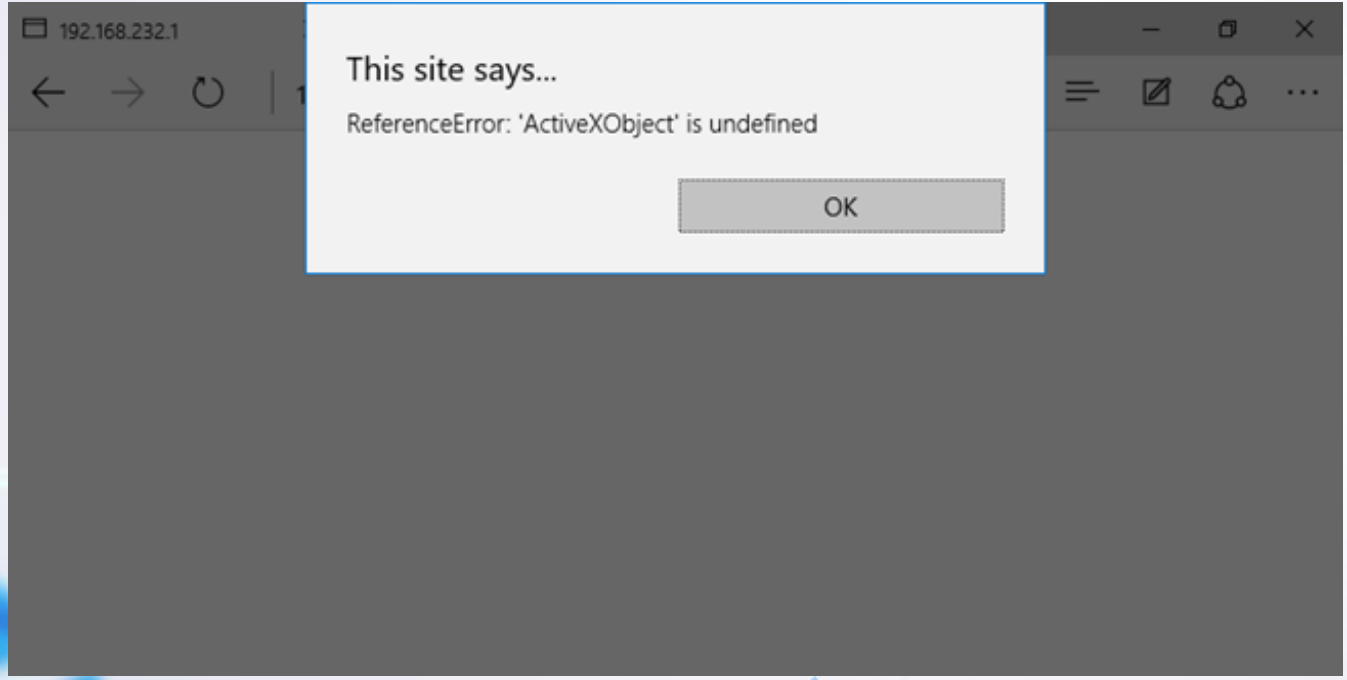
Microsoft Edge 不支持 ActiveX

```
0:018> lmvm flash
Browse full module list
start    end        module name
0fc60000 11231000   Flash      (deferred)
    Image path: C:\Windows\System32\Macromed\Flash\Flash.ocx
    Image name: Flash.ocx
    Browse all global symbols  functions  data
    Timestamp:        Tue Jan 31 04:16:58 2017 (588F9F3A)
    CheckSum:         014DEB1A
    ImageSize:        015D1000
    File version:     24.0.0.221
    Product version:  24.0.0.221
    File flags:       0 (Mask 3F)
    File OS:          4 Unknown Win32
    File type:        2.0 Dll
    File date:        00000000.00000000
    Translations:     0409.04b0
    CompanyName:      Adobe Systems, Inc.
    ProductName:      Shockwave Flash
    InternalName:     Adobe Flash Player 24.0
    OriginalFilename: Flash.ocx
    ProductVersion:   24,0,0,221
    FileVersion:      24,0,0,221
    FileDescription:  Adobe Flash Player 24.0 r0
    LegalCopyright:   Adobe® Flash® Player. Copyright © 1996-2017 Adobe Systems Incorporated.
    LegalTrademarks:  Adobe Flash Player
```

Microsoft Edge 不支持 ActiveX



限制1: 没有 ActiveXObject 对象



解决方案：使用 object 元素

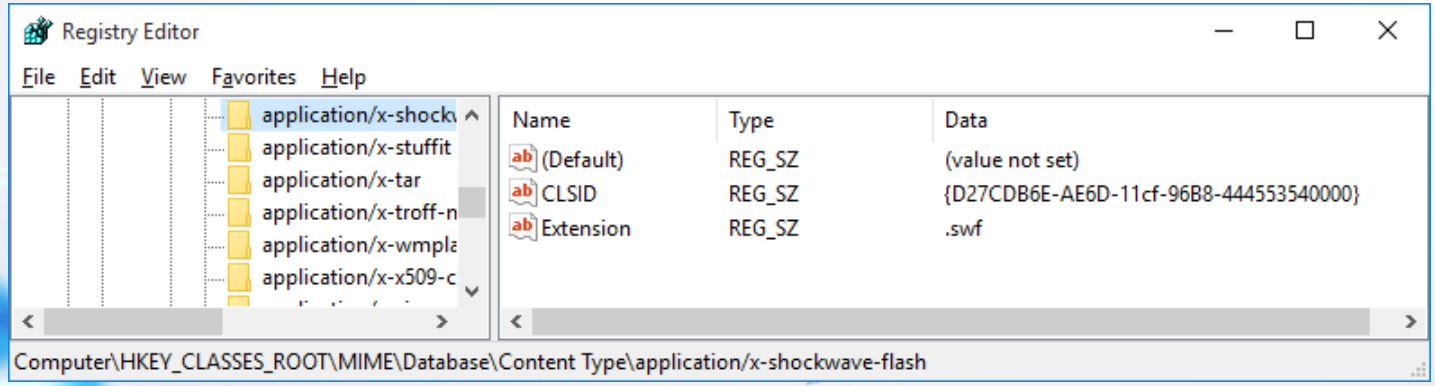
```
var o = new ActiveXObject("msxml2.xmlhttp.3.0");
```

↓

```
var o = document.createElement("object");
o.classid="clsid:F5078F35-C551-11D3-89B9-0000F81FE221";
document.body.appendChild(o);
```

限制2：没有 classid 属性

```
var o = document.createElement("object");
o.type="application/x-shockwave-flash";
document.body.appendChild(o);
```



解决方案：利用遗产代码

CObjectElement 对象中仍然保留有 classid 的成员变量

CObjectElement::RetrieveClassidAndData 函数优先使用该成员变量

获得任意地址读写能力后可以直接设置 classid 的成员变量

限制3：ActiveX 白名单

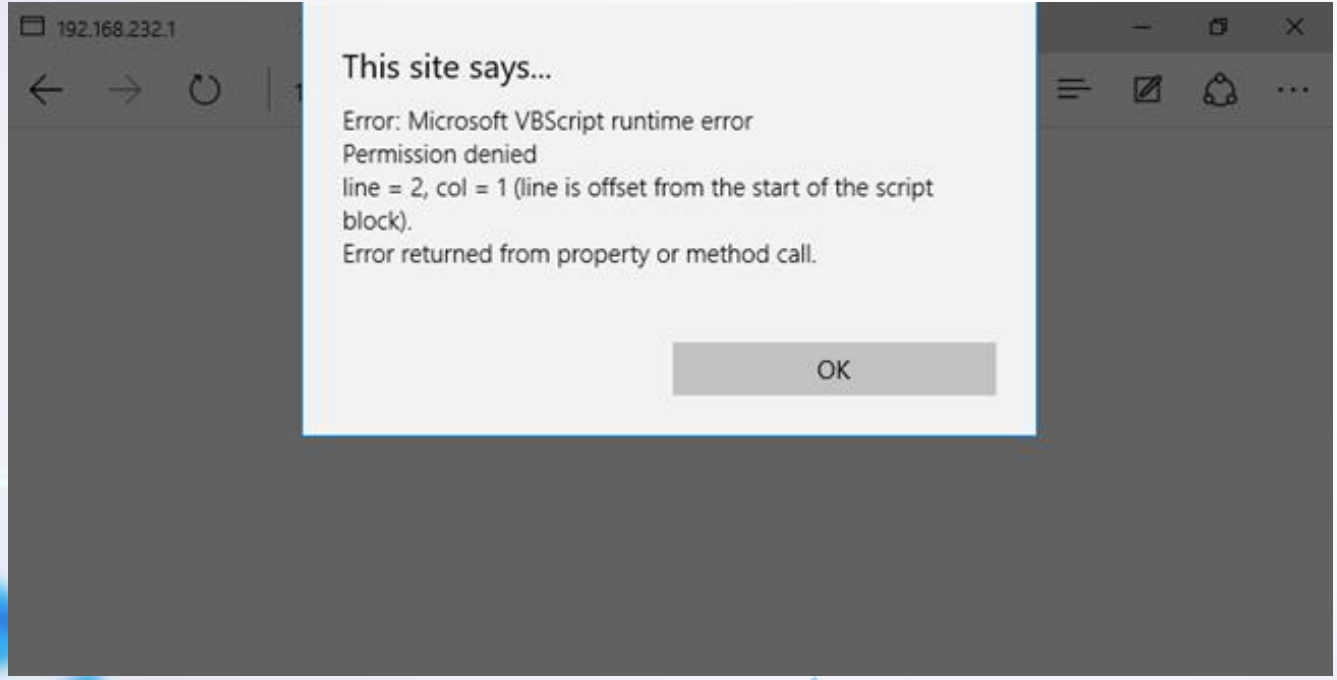
```
.data:1A4F0128 ; _GUID c_rgsActiveXTrustedList[29]
.data:1A4F0128 c_rgsActiveXTrustedList dd 0F6D90F11h          ; Data1
.data:1A4F0128                                          ; DATA XREF: IsAppContainerCompatible
.data:1A4F0128                                          ; Ext_IsActiveXImmersiveCompatible(_G
.data:1A4F0128                 dw 9C73h                 ; Data2
.data:1A4F0128                 dw 11D3h                 ; Data3
.data:1A4F0128                 db 0B3h, 2Eh, 0, 0C0h, 4Fh, 99h, 0Bh, 0B4h; Data4
.data:1A4F0128                 dd 0F6D90F12h            ; Data1
.data:1A4F0128                 dw 9C73h                 ; Data2
.data:1A4F0128                 dw 11D3h                 ; Data3
.data:1A4F0128                 db 0B3h, 2Eh, 0, 0C0h, 4Fh, 99h, 0Bh, 0B4h; Data4
.data:1A4F0128                 dd 2933BF91h             ; Data1
.data:1A4F0128                 dw 7B36h                 ; Data2
.data:1A4F0128                 dw 11D2h                 ; Data3
.data:1A4F0128                 db 0B2h, 0Eh, 0, 0C0h, 4Fh, 98h, 3Eh, 60h; Data4
.data:1A4F0128                 dd 373984C9h             ; Data1
.data:1A4F0128                 dw 0B845h                ; Data2
.data:1A4F0128                 dw 449Bh                 ; Data3
.data:1A4F0128                 db 91h, 0E7h, 45h, 0ACh, 83h, 3, 6Ah, 0DEh; Data4
```

解决方案：使用 MSXML2.XMLHTTP.3.0

MSXML2.XMLHTTP.3.0 正好在白名单内

James Forshaw 的利用技术只需要使用 MSXML2.XMLHTTP.3.0

限制4：AppContainer



解决方案：使用 WScript.Shell.Exec

问题的本质是调用 GetShellProcessHandle 函数时因权限不够而失败

WScript.Shell.Exec 直接调用 KERNELBASE!CreateProcessW 函数

限制5：代码完整性

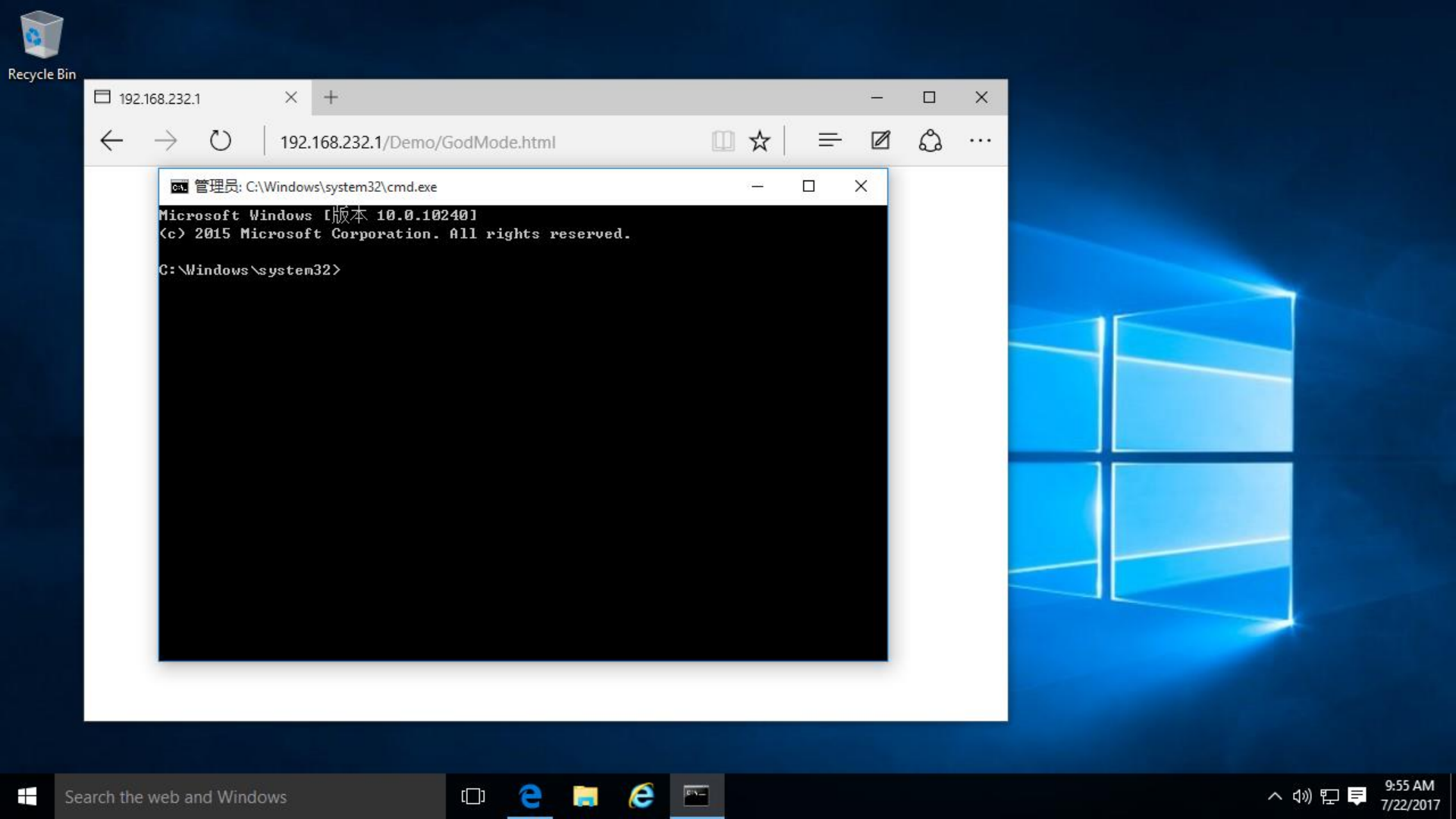

192.168.232.1
This site says...
Error: WshShell.Exec
Windows cannot verify the digital signature for this file. A recent hardware or software change might have installed a file that is signed incorrectly or damaged, or that might be malicious software from an unknown source.
line = 2, col = 1 (line is offset from the start of the script block).
Error returned from property or method call.
OK

解决方案：使用 rundll32.exe

rundll32.exe 具有 Microsoft Windows 签名

rundll32.exe 加载 dll 时并不检查代码完整性

rundll32.exe 可以在 AppContainer 内执行


Recycle Bin
192.168.232.1
192.168.232.1/Demo/GodMode.html
管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.
C:\Windows\system32>
Search the web and Windows
9:55 AM
7/22/2017

问题修复

WIP 15048 引入 COleSite::IsSupportedControl 进行过滤

仅支持 4 个控件

- CLSID_MacromediaSwFlash
- IID_DRTSurfacePresenterFlipCtrlOOP
- IID_DRTSurfacePresenterFlipCtrl
- IID_DRTPluginInterfaceTester

The Lord of the Edge: The Shims of the Edge

Shims 是什么？

用于处理应用兼容性的一个中间层

基于 API Hooking 技术实现

Microsoft Edge 使用 Shims 来实现其沙箱

Microsoft Edge 中 Shims 如何工作？

初始化时注册 Dll Notification 回调函数

EShims!IEShims_Initialize

EShims!CShimBindings::Initialize

EShims!CShimBindings::_Register

ntdll!LdrRegisterDllNotification

Microsoft Edge 中 Shims 如何工作？

初始化时注册 Dll Notification 回调函数

```
ntdll = GetModuleHandleW(L"NTDLL.DLL");
LdrRegisterDllNotification = GetProcAddress(ntdll, "LdrRegisterDllNotification");
if ( LdrRegisterDllNotification )
{
  __guard_check_icall_fptr(LdrRegisterDllNotification, v14);
  status = (LdrRegisterDllNotification)(
             0,
             CShimBindings::_LdrNotificationCallback,
             0,
             v10 + 0x3C,
```

Microsoft Edge 中 Shims 如何工作？

在回调函数中替换模块入口函数

EShims!CShimBindings::_LdrNotificationCallback

EShims!ShimBindings::OnModuleLoaded

Microsoft Edge 中 Shims 如何工作？

在回调函数中替换模块入口函数

```
if ( LdrDataEntry )
{
  i = this->ModuleCount;
  if ( i < 0x200 )
  {
    this->Modules[i].DllBase = DllBase;
    this->Modules[this->ModuleCount].DllEnd = DllBase + LdrDataEntry->SizeOfImage;
    *(&this->DefaultFlag + 8 * (this->ModuleCount + 0xB)) = LdrDataEntry->EntryPoint;
    NeededShims = CShimBindings::_GetNeededShims(this, LdrDataEntry->BaseDllName.Buffer);
    i_ = this->ModuleCount;
    this->Modules[i_].NeededShims = NeededShims;
    this->Modules[i_].Unknown = v10;
    LOBYTE(this->Modules[this->ModuleCount++].Patched) = 0;
    _InterlockedIncrement(&BindingRef::s_bindingsRefCount);
    LdrDataEntry->EntryPoint = CShimBindings::s_DllMainHook;
    this->NeedPatch = 1;
  }
}
```

Microsoft Edge 中 Shims 如何工作？

在模块入口函数中修改导入表

EShims!CShimBindings::s_DllMainHook

EShims!CShimBindings::DllMainHook

EShims!CShimBindings::ApplyShims

EShims!CShimBindings::_PatchNewModule

EShims!CShimBindings::LUPatchVerify

Microsoft Edge 中 Shims 如何工作？

在模块入口函数中修改导入表

```
if ( lpAddress )
{
  if ( VirtualQuery(lpAddress, &Buffer, 0x1Cu) )
  {
    status = VirtualProtect(lpAddress, 4u, (Buffer.Protect & 0xFFFFFF0F) != 0 ? 4 : 0x40000040, &flOldProtect);
    if ( status )
    {
      *lpAddress = value;
      if ( !(flOldProtect & 0xFFFFFF0F) )
        flOldProtect |= 0x40000000u;
      status = VirtualProtect(lpAddress, 4u, flOldProtect, &temp);
    }
  }
}
return status;
```

限制1：写入的数据不可控

记录 Hook API 信息的数据保存在 .mrdata 段中

通过 IEShims_VerifyWithinMrdata 函数进行校验

```
void __fastcall IEShims_VerifyWithinMrdata(int address)
{
  if ( address < g_pMrdataBase || address >= g_pMrdataBase + g_ulMrdataSize )
    __fastfail(5u);
}
```

限制1：写入的数据不可控

不能直接修改 g_pMrdataBase 或者 g_ulMrdataSize

```
void __fastcall IEShims_MakeMrdataReadOnly(char a1)
{
  DWORD v1; // ebx@2
  DWORD flOldProtect; // [sp+0h] [bp-8h]@4

  if ( g_fMrdataReadOnly != a1 )
  {
    v1 = 2 * (a1 == 0) + 2;
    if ( v1 == 2 )
      g_fMrdataReadOnly = a1;
    VirtualProtect(g_pMrdataBase, g_ulMrdataSize, v1, &flOldProtect);
    if ( v1 == 4 )
      g_fMrdataReadOnly = a1;
  }
}
```

解决方案：直接修改代码

函数 IEShims_VerifyWithinMrdata 通过地址访问 g_ulMrdataSize

```
EShims!IEShims_VerifyWithinMrdata:
70db776e 8b158404dc70     mov     edx,dword ptr [EShims!g_pMrdataBase (70dc0484)]
70db7774 3bca             cmp     ecx,edx
70db7776 720b             jb      EShims!IEShims_VerifyWithinMrdata+0x15 (70db7783)  Branch

EShims!IEShims_VerifyWithinMrdata+0xa:
70db7778 a1ac02dc70       mov     eax,dword ptr [EShims!g_ulMrdataSize (70dc02ac)]
70db777d 03c2             add     eax,edx
70db777f 3bc8             cmp     ecx,eax
70db7781 7205             jb      EShims!IEShims_VerifyWithinMrdata+0x1a (70db7788)  Branch

EShims!IEShims_VerifyWithinMrdata+0x15:
70db7783 6a05             push    5
70db7785 59               pop     ecx
70db7786 cd29             int     29h

EShims!IEShims_VerifyWithinMrdata+0x1a:
70db7788 c3               ret  Branch
```

解决方案：直接修改代码

将该地址修改为 Hook API 的地址以扩大能通过校验的地址范围

```
EShims!IEShims_VerifyWithinMrdata:
70db776e 8b158404dc70    mov     edx,dword ptr [EShims!g_pMrdataBase (70dc0484)]
70db7774 3bca            cmp     ecx,edx
70db7776 720b            jb      EShims!IEShims_VerifyWithinMrdata+0x15 (70db7783)  Branch

EShims!IEShims_VerifyWithinMrdata+0xa:
70db7778 a1504fdb70      mov     eax,dword ptr [EShims!NS_LRIECoCreate::APIHook_CoCreateInstance (70db4f50)]
70db777d 03c2            add     eax,edx
70db777f 3bc8            cmp     ecx,eax
70db7781 7205            jb      EShims!IEShims_VerifyWithinMrdata+0x1a (70db7788)  Branch

EShims!IEShims_VerifyWithinMrdata+0x15:
70db7783 6a05            push    5
70db7785 59              pop     ecx
70db7786 cd29            int     29h

EShims!IEShims_VerifyWithinMrdata+0x1a:
70db7788 c3              ret  Branch
```

限制2：ACG

启用 ACG 后将不能直接修改代码

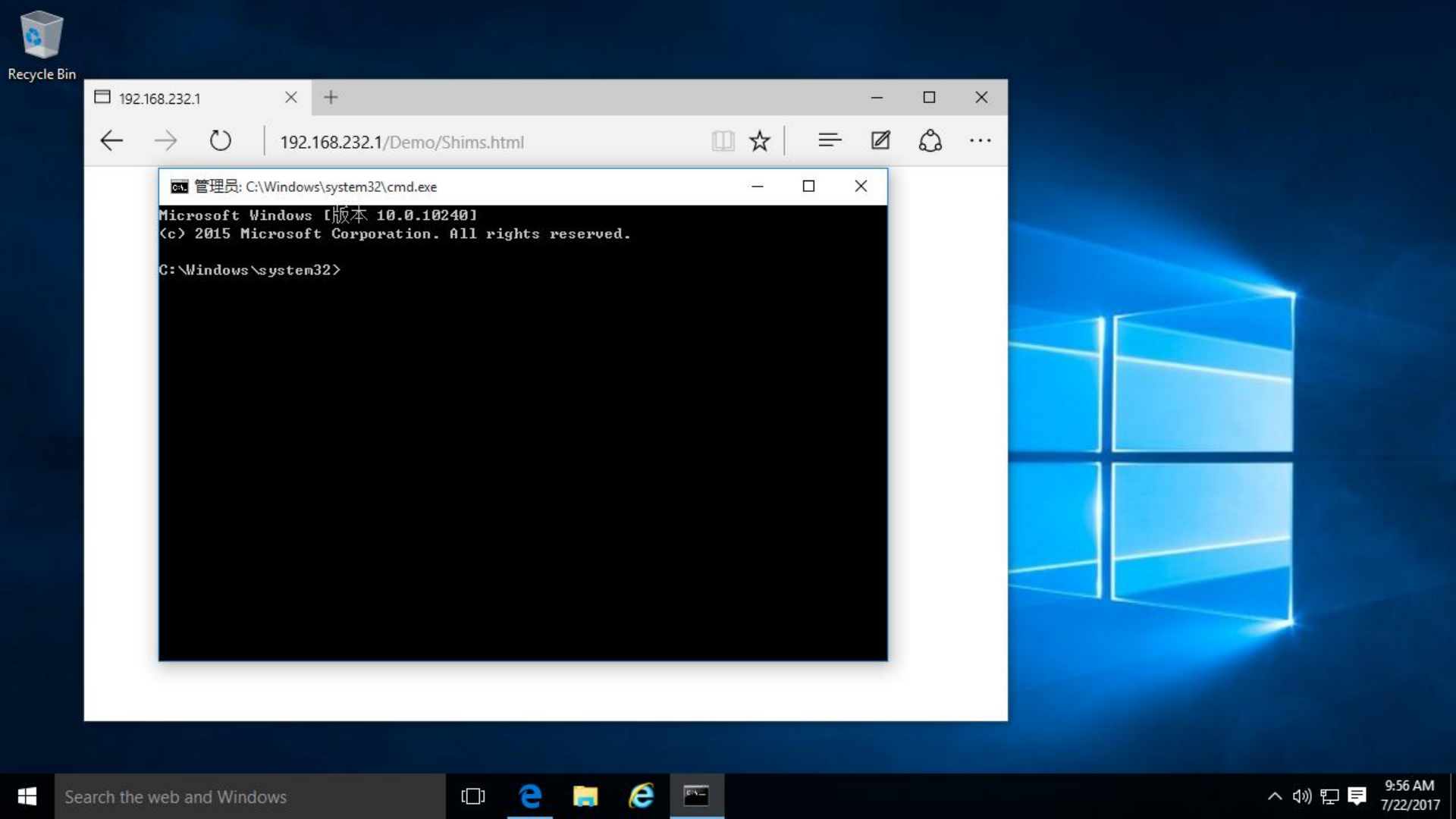
```
if ( lpAddress )
{
  if ( VirtualQuery(lpAddress, &Buffer, 0x1Cu) )
  {
    status = VirtualProtect(lpAddress, 4u, (Buffer.Protect & 0xFFFFFF0F) != 0 ? 4 : 0x40000040, &flOldProtect);
    if ( status )
    {
      *lpAddress = value;
      if ( !(flOldProtect & 0xFFFFFF0F) )
        flOldProtect |= 0x40000000u;
      status = VirtualProtect(lpAddress, 4u, flOldProtect, &temp);
    }
  }
}
return status;
```

解决方案：利用 ThreadOptOut

NS_ACGLockdownTelemetry::APIHook_VirtualProtect 自动进行 ThreadOptOut

```
BOOL __cdecl NS_ACGLockdownTelemetry::APIHook_VirtualProtect(LPVOID lpAddress,
{
  int status; // esi@3
  char lockdown; // [sp+Fh] [bp-Dh]@1
  int v1; // [sp+18h] [bp-4h]@1

  lockdown = 0;
  v1 = 0;
  if ( flNewProtect & 0x70 )
    CACGLockdown::Enable(&lockdown);
  status = VirtualProtect(lpAddress, dwSize, flNewProtect, lpflOldProtect);
  if ( status != 1 && GetLastError() == 0x677 )
    ReportACGLockdownTelemetryViolation();
  CACGLockdown::~CACGLockdown(&lockdown);
  return status;
}
```


Recycle Bin
192.168.232.1
192.168.232.1/Demo/Shims.html
管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.
C:\Windows\system32>
Search the web and Windows
9:56 AM
7/22/2017

问题修复

Creator Update 引入 CShimBindings::_VerifyNewModule

```
void __stdcall CShimBindings::_VerifyNewModule(const struct LOADED_MODULE *module)
{
  struct _MEMORY_BASIC_INFORMATION info; // [sp+4h] [bp-1Ch]@1

  if ( !VirtualQuery(module->DllBase, &info, 0x1Cu)
    || module->DllBase != info.BaseAddress
    || module->DllEnd != info.BaseAddress + *(info.BaseAddress + *(info.BaseAddress + 0xF) + 0x50)
    || !(info.Type & 0x1000000)                 // MEM_IMAGE
    || info.Protect & 4                         // PAGE_READWRITE
    || !(info.State & 0x1000) )                 // MEM_COMMIT
  {
    RaiseFailFastException(0, 0, 1u);
  }
}
```

Q & A

THANKS!