Mitigation Bypass

The Past, Present, and Future

Who am I

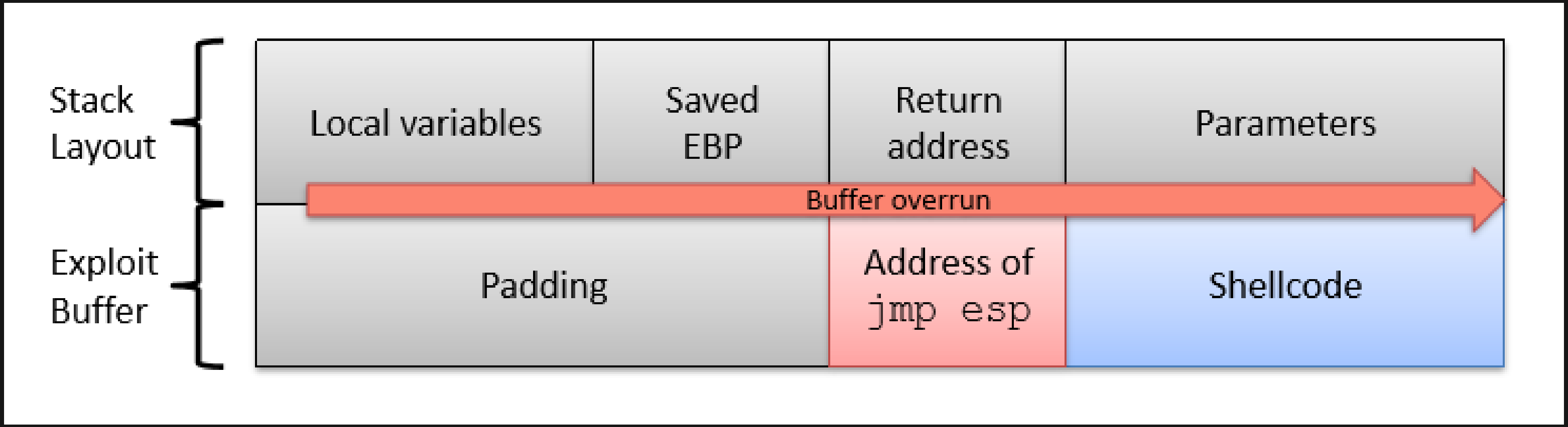
- Yunhai Zhang
- Twitter: @_f0rgetting_
- Researcher of NSFOCUS
- Winner of Mitigation Bypass Bounty: 2014 ~ 2018

The Good Old Days

- Before 2002
  - No mitigations at all
  - Exploit is just trivial

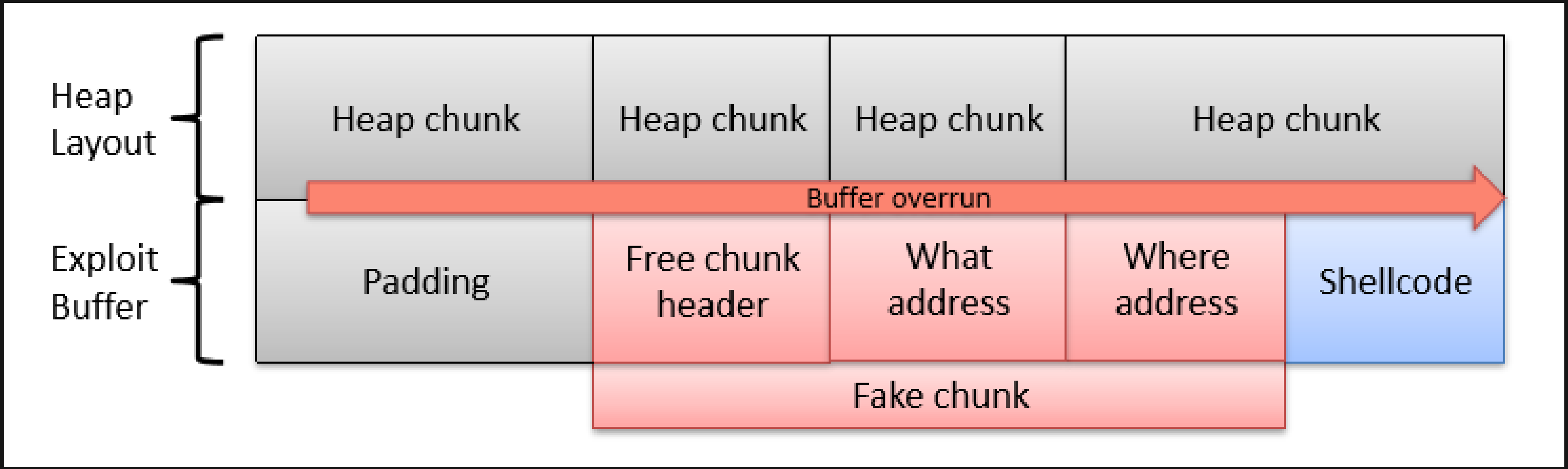
The Good Old Days

- Stack Buffer Overrun



The Good Old Days

- Heap Buffer Overrun

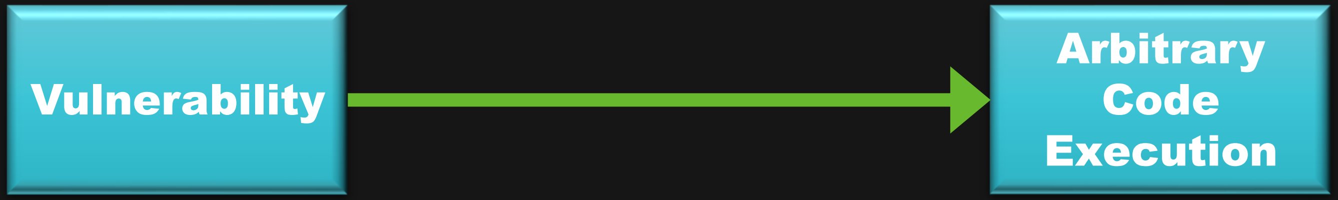


NSFOCUS

The Good Old Days

Vulnerability

The Good Old Days



The Rise of Mitigations

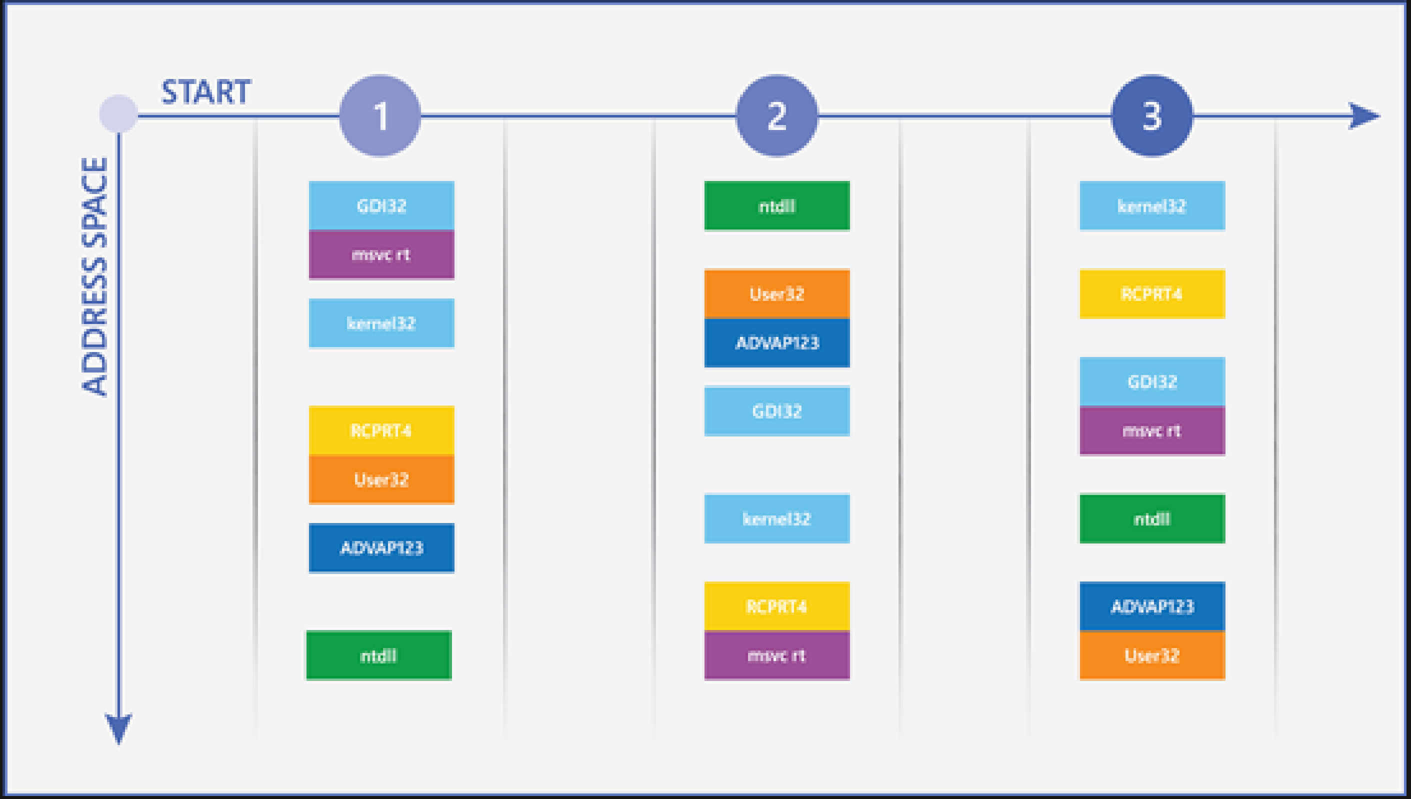
- Mitigations for Stack Buffer Overrun
  - 2002: GS v1
  - 2003: SafeSEH
  - 2005: GS v2
  - 2008: SEHOP
  - 2010: GS v3

The Rise of Mitigations

- Mitigations for Heap Buffer Overrun
  - 2004: Safe unlinking & Heap entry header cookie
  - 2006: Heap entry metadata randomization
  - 2012: Heap randomization & metadata protection

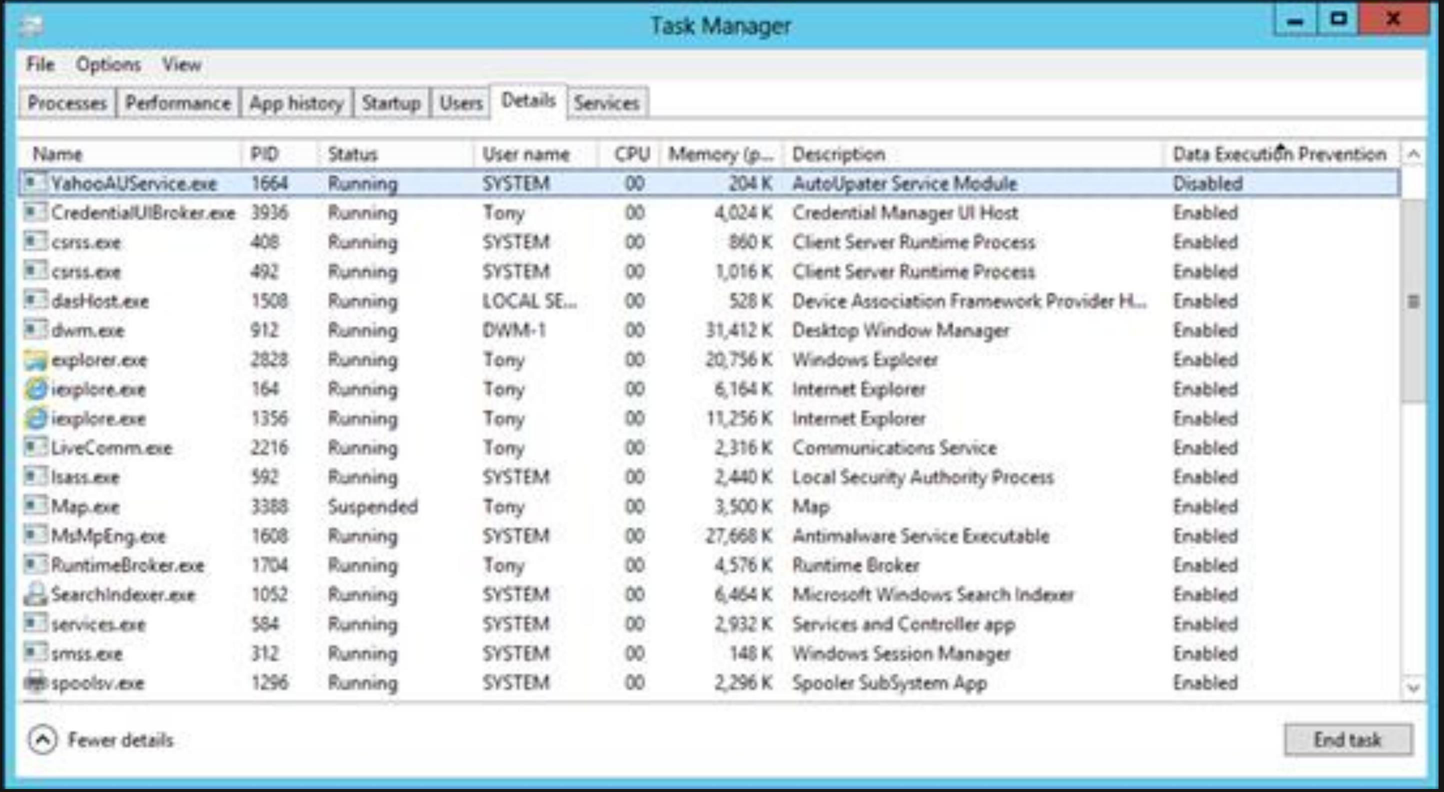
The Rise of Mitigations

- Address Space Layout Randomization (ASLR)



The Rise of Mitigations

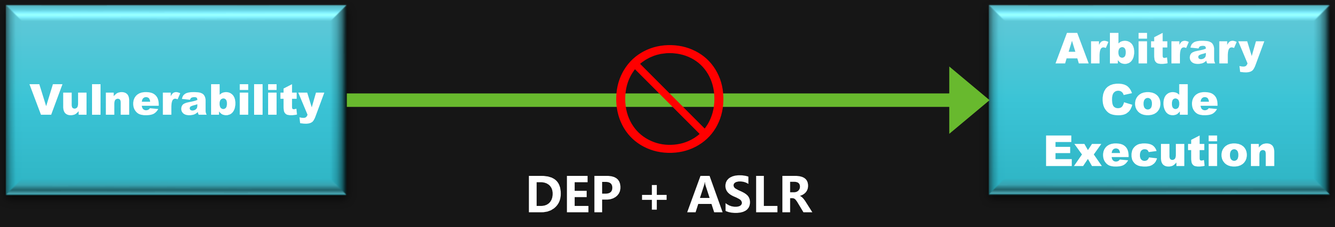
- Data Execution Prevention (DEP)


Task Manager
File Options View
Processes Performance App history Startup Users Details Services
Fewer details
End task

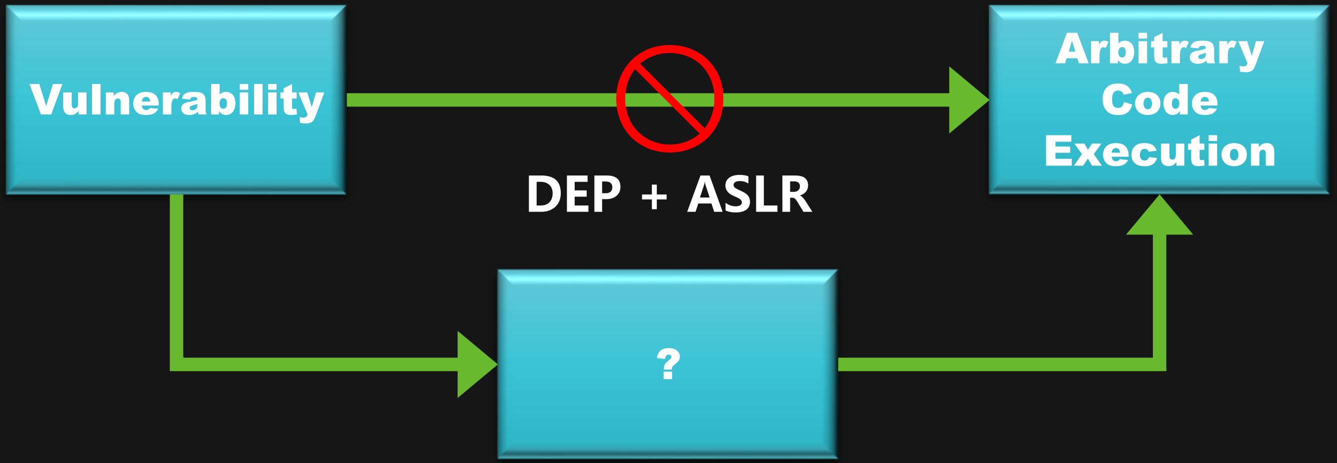
Name	PID	Status	User name	CPU	Memory (p...	Description	Data Execution Prevention
YahooAUService.exe	1664	Running	SYSTEM	00	204 K	AutoUpater Service Module	Disabled
CredentialUIBroker.exe	3936	Running	Tony	00	4,024 K	Credential Manager UI Host	Enabled
csrss.exe	408	Running	SYSTEM	00	860 K	Client Server Runtime Process	Enabled
csrss.exe	492	Running	SYSTEM	00	1,016 K	Client Server Runtime Process	Enabled
dasHost.exe	1508	Running	LOCAL SE...	00	528 K	Device Association Framework Provider H...	Enabled
dwm.exe	912	Running	DWM-1	00	31,412 K	Desktop Window Manager	Enabled
explorer.exe	2828	Running	Tony	00	20,756 K	Windows Explorer	Enabled
iexplore.exe	164	Running	Tony	00	6,164 K	Internet Explorer	Enabled
iexplore.exe	1356	Running	Tony	00	11,256 K	Internet Explorer	Enabled
LiveComm.exe	2216	Running	Tony	00	2,316 K	Communications Service	Enabled
lsass.exe	592	Running	SYSTEM	00	2,440 K	Local Security Authority Process	Enabled
Map.exe	3388	Suspended	Tony	00	3,500 K	Map	Enabled
MsMpEng.exe	1608	Running	SYSTEM	00	27,668 K	Antimalware Service Executable	Enabled
RuntimeBroker.exe	1704	Running	Tony	00	4,576 K	Runtime Broker	Enabled
SearchIndexer.exe	1052	Running	SYSTEM	00	6,464 K	Microsoft Windows Search Indexer	Enabled
services.exe	584	Running	SYSTEM	00	2,932 K	Services and Controller app	Enabled
smss.exe	312	Running	SYSTEM	00	148 K	Windows Session Manager	Enabled
spoolsv.exe	1296	Running	SYSTEM	00	2,296 K	Spooler SubSystem App	Enabled

NSFOCUS

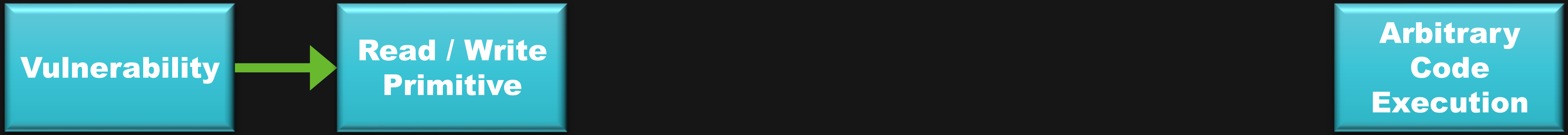
The Rise of Mitigations



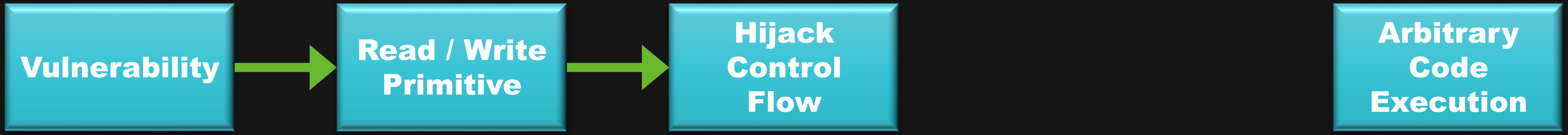
The Birth of Mitigation Bypass



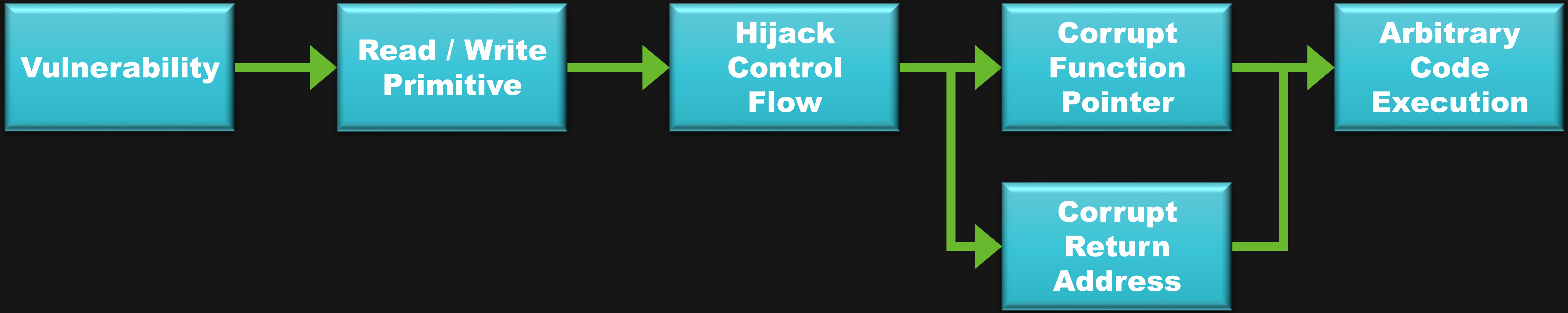
The Birth of Mitigation Bypass



The Birth of Mitigation Bypass



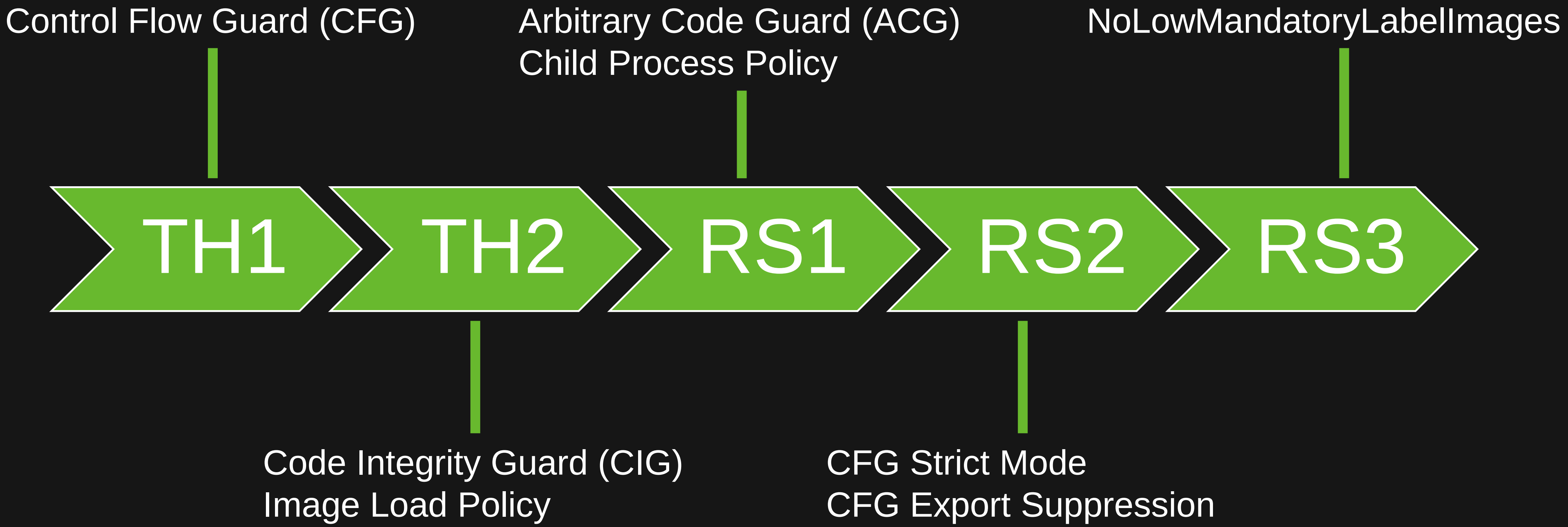
The Birth of Mitigation Bypass



The Era Of Windows 10

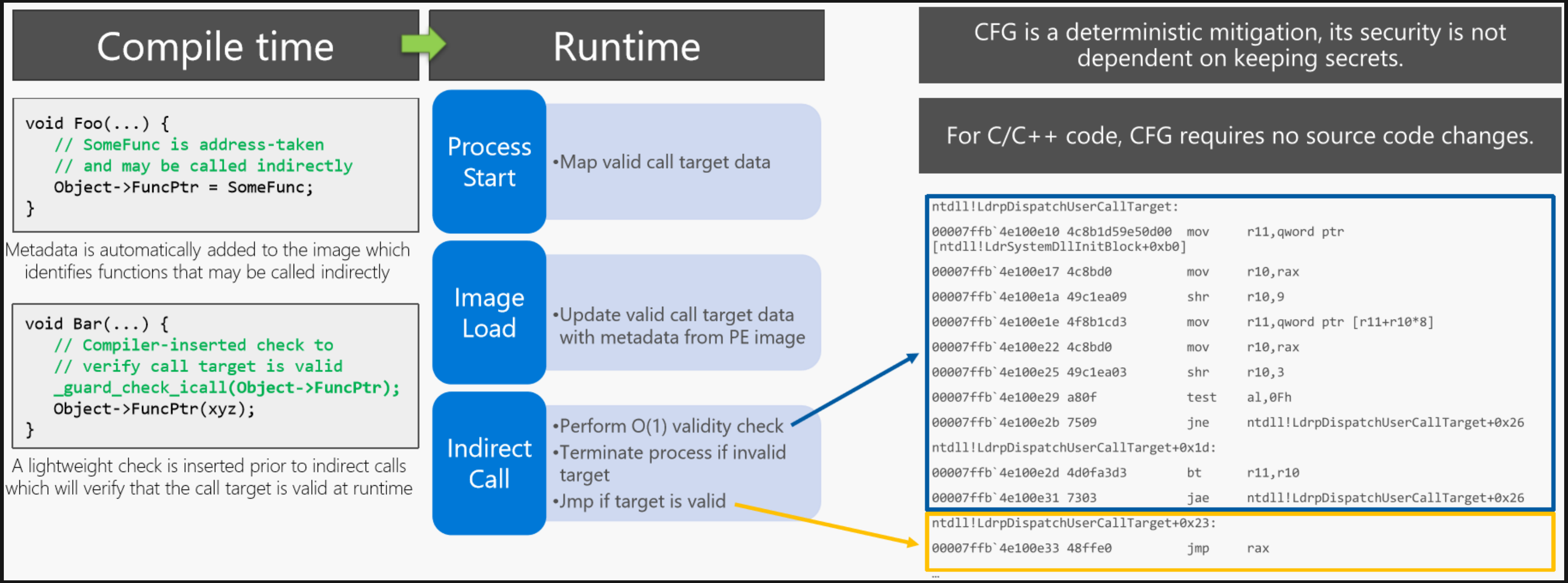
- New release strategy
  - 2 major updates every year
  - New mitigations each update

The Era Of Windows 10



The Era Of Windows 10

- Control Flow Guard (CFG)


Compile time
Runtime
Process Start
•Map valid call target data
Image Load
•Update valid call target data with metadata from PE image
Indirect Call
•Perform O(1) validity check
•Terminate process if invalid target
•Jmp if target is valid

```
void Foo(...) {
   // SomeFunc is address-taken
   // and may be called indirectly
   Object->FuncPtr = SomeFunc;
}
```

Metadata is automatically added to the image which identifies functions that may be called indirectly

```
void Bar(...) {
   // Compiler-inserted check to
   // verify call target is valid
   _guard_check_icall(Object->FuncPtr);
   Object->FuncPtr(xyz);
}
```

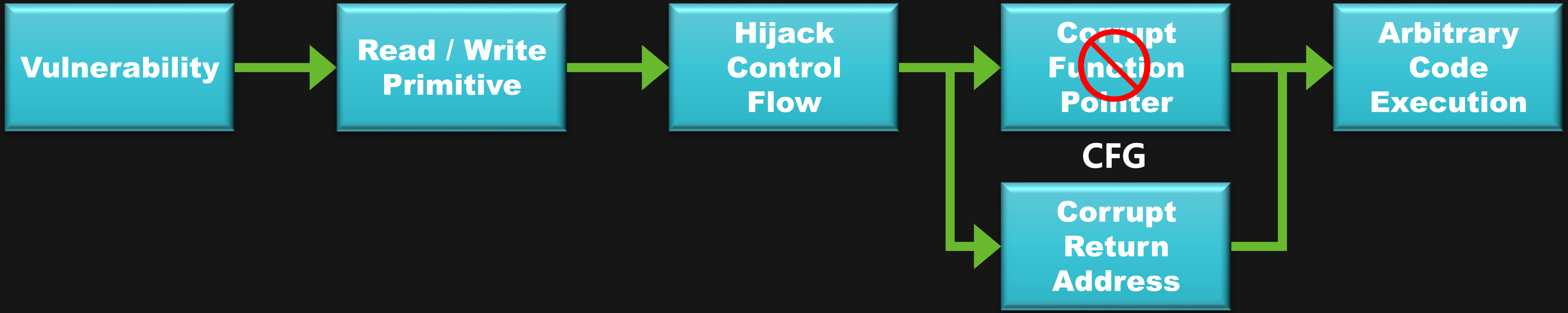
A lightweight check is inserted prior to indirect calls which will verify that the call target is valid at runtime

CFG is a deterministic mitigation, its security is not dependent on keeping secrets.

For C/C++ code, CFG requires no source code changes.

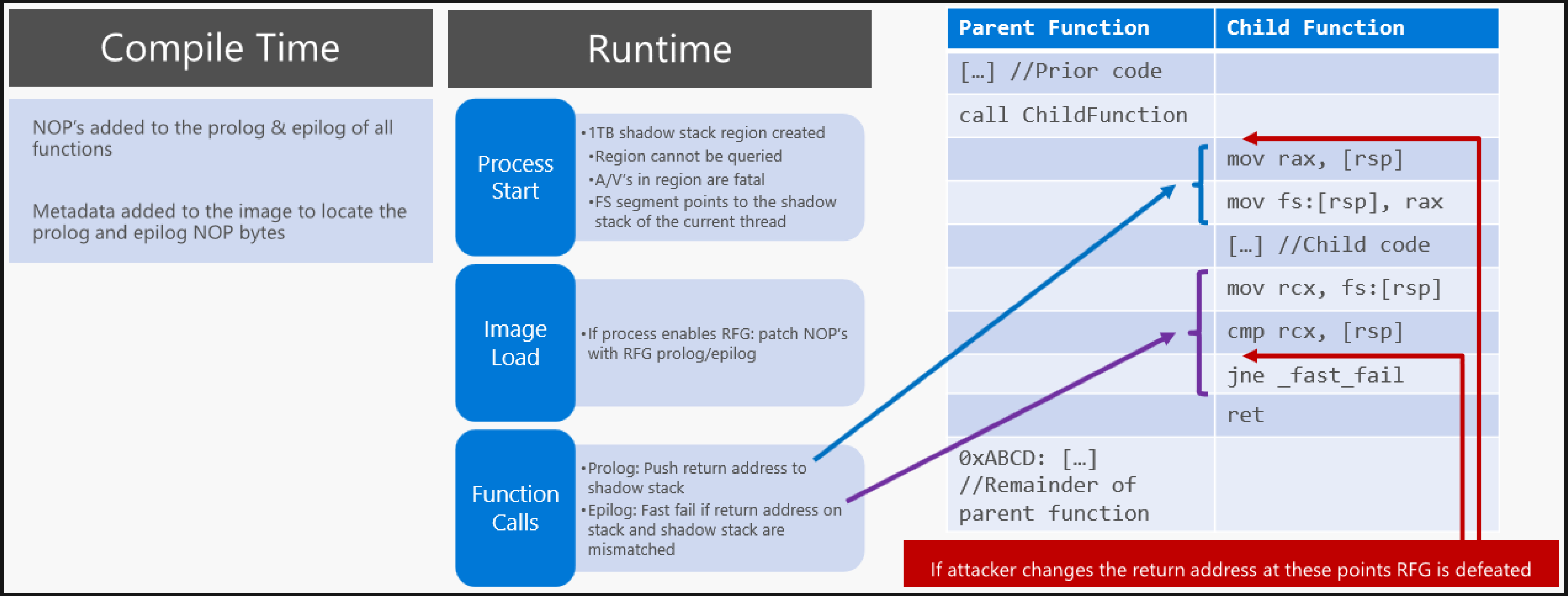
```
ntdll!LdrpDispatchUserCallTarget:
00007ffb`4e100e10 4c8b1d59e50d00  mov     r11,qword ptr
[ntdll!LdrSystemDllInitBlock+0xb0]
00007ffb`4e100e17 4c8bd0          mov     r10,rax
00007ffb`4e100e1a 49c1ea09        shr     r10,9
00007ffb`4e100e1e 4f8b1cd3        mov     r11,qword ptr [r11+r10*8]
00007ffb`4e100e22 4c8bd0          mov     r10,rax
00007ffb`4e100e25 49c1ea03        shr     r10,3
00007ffb`4e100e29 a80f            test    al,0Fh
00007ffb`4e100e2b 7509            jne     ntdll!LdrpDispatchUserCallTarget+0x26
ntdll!LdrpDispatchUserCallTarget+0x1d:
00007ffb`4e100e2d 4d0fa3d3        bt      r11,r10
00007ffb`4e100e31 7303            jae     ntdll!LdrpDispatchUserCallTarget+0x26
ntdll!LdrpDispatchUserCallTarget+0x23:
00007ffb`4e100e33 48ffe0          jmp     rax
...
```

The Era Of Windows 10



The Era Of Windows 10

Return Flow Guard (RFG)



Compile Time

NOP's added to the prolog & epilog of all functions

Metadata added to the image to locate the prolog and epilog NOP bytes

Runtime

Process Start

- 1TB shadow stack region created
  - Region cannot be queried
  - A/V's in region are fatal
  - FS segment points to the shadow stack of the current thread

Image Load

- If process enables RFG: patch NOP's with RFG prolog/epilog

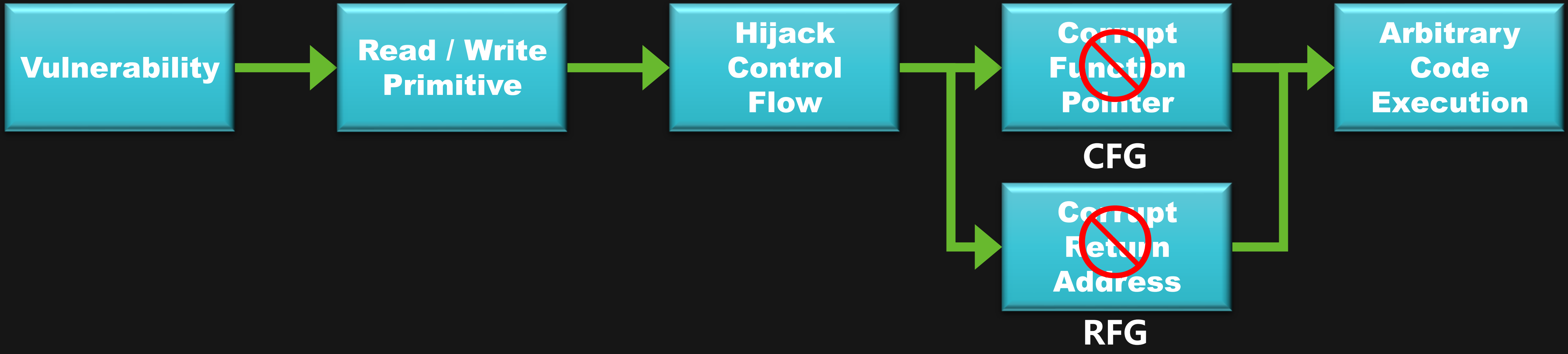
Function Calls

- Prolog: Push return address to shadow stack
- Epilog: Fast fail if return address on stack and shadow stack are mismatched

Parent Function	Child Function
[…] //Prior code	
call ChildFunction	
	mov rax, [rsp]
	mov fs:[rsp], rax
	[…] //Child code
	mov rcx, fs:[rsp]
	cmp rcx, [rsp]
	jne _fast_fail
	ret
0xABCD: […] //Remainder of parent function	

If attacker changes the return address at these points RFG is defeated

The Era Of Windows 10



The Era Of Windows 10

- Adversarial CFG Bypass Techniques
  - Unprotected code
  - Valid sensitive functions
  - Valid wrapper functions
- Alternative Bypass Techniques
  - Load Library
  - Abuse Feature

The Era Of Windows 10

- Unprotected code
  - Non-CFG module
  - JIT generated code
  - Indirect jump
  - setjmp / longjmp
  - Writeable IAT

The Era Of Windows 10

- Unprotected code
  - ~~Non-CFG module~~ CFG strict mode
  - JIT generated code
  - Indirect jump
  - setjmp / longjmp
  - Writeable IAT

The Era Of Windows 10

- Unprotected code
  - ~~Non-CFG module~~ CFG strict mode
  - ~~JIT generated code~~ CFG aware JIT
  - Indirect jump
  - setjmp / longjmp
  - Writeable IAT

The Era Of Windows 10

- Unprotected code
  - ~~Non-CFG module~~ CFG strict mode
  - ~~JIT generated code~~ CFG aware JIT
  - ~~Indirect jump~~ CFG aware jump
  - setjmp / longjmp
  - Writeable IAT

The Era Of Windows 10

- Unprotected code
  - ~~Non-CFG module~~ CFG strict mode
  - ~~JIT generated code~~ CFG aware JIT
  - ~~Indirect jump~~ CFG aware jump
  - ~~setjmp / longjmp~~ Longjmp hardening
  - Writeable IAT

The Era Of Windows 10

- Unprotected code
  - ~~Non-CFG module~~ CFG strict mode
  - ~~JIT generated code~~ CFG aware JIT
  - ~~Indirect jump~~ CFG aware jump
  - ~~setjmp / longjmp~~ Longjmp hardening
  - ~~Writable IAT~~

The Era Of Windows 10

- Valid sensitive functions
  - WinExec
  - NtContinue
  - VirtualAlloc
  - VirtualProtect
  - HeapCreate
  - MapViewOfFile
  - ...

The Era Of Windows 10

- Valid sensitive functions
  - WinExec
  - NtContinue
  - VirtualAlloc
  - VirtualProtect
  - HeapCreate
  - MapViewOfFile
  - ...

CFG Explicit Suppression

CFG Export Suppression

The Era Of Windows 10

- Valid wrapper functions
  - eshims!NS_ACGLockdownTelemetry::APIHook_VirtualProtect
  - d3d10warp!CodeStorageBlock::Protect
  - d3d10level9! CCodeWriter::~CCodeWriter
  - chakra!Memory::SmallHeapBlockT::ClearPageHeapState
  - ...

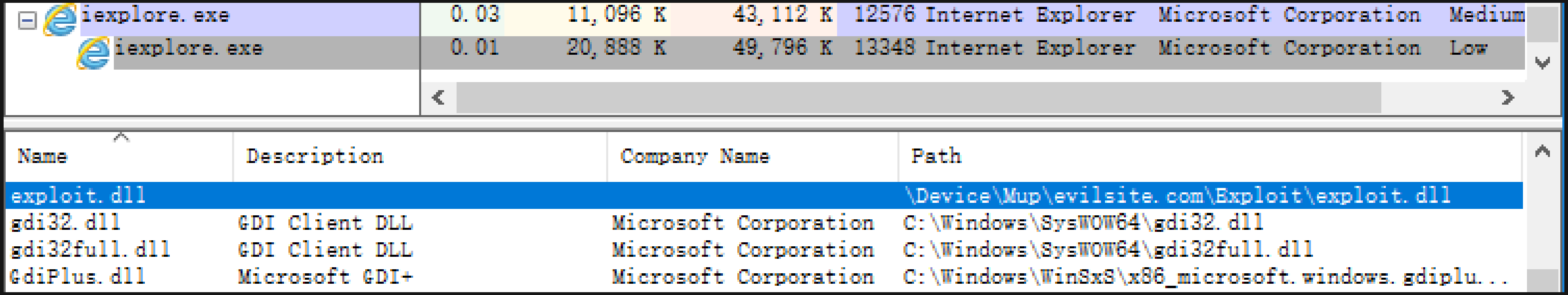
The Era Of Windows 10

- Load Library
  - It is trivial to call LoadLibraryA to load any library

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

The Era Of Windows 10

- Load Library
  - Use UNC path to load a remote library

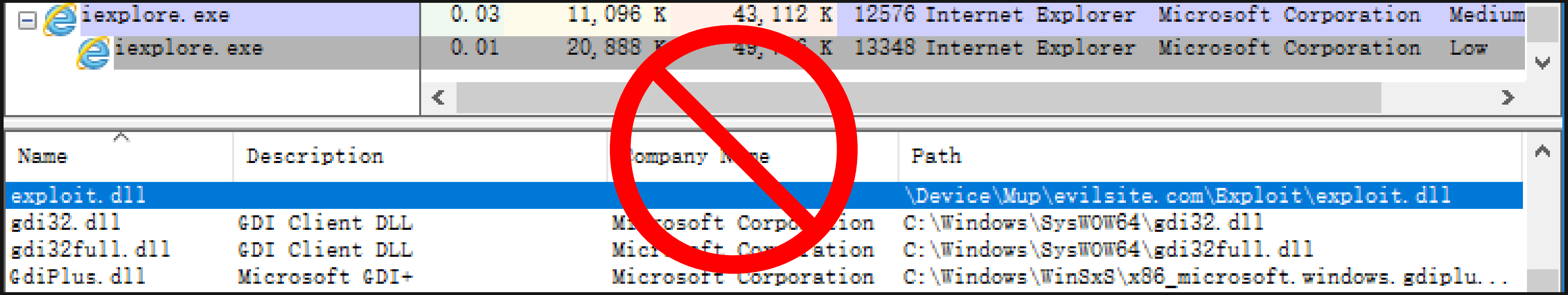


```
iexplore.exe      0.03   11,096 K   43,112 K  12576 Internet Explorer  Microsoft Corporation  Medium
  iexplore.exe    0.01   20,888 K   49,796 K  13348 Internet Explorer  Microsoft Corporation  Low
```

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...

The Era Of Windows 10

- Load Library
  - NoRemoteImages prevent load remote library

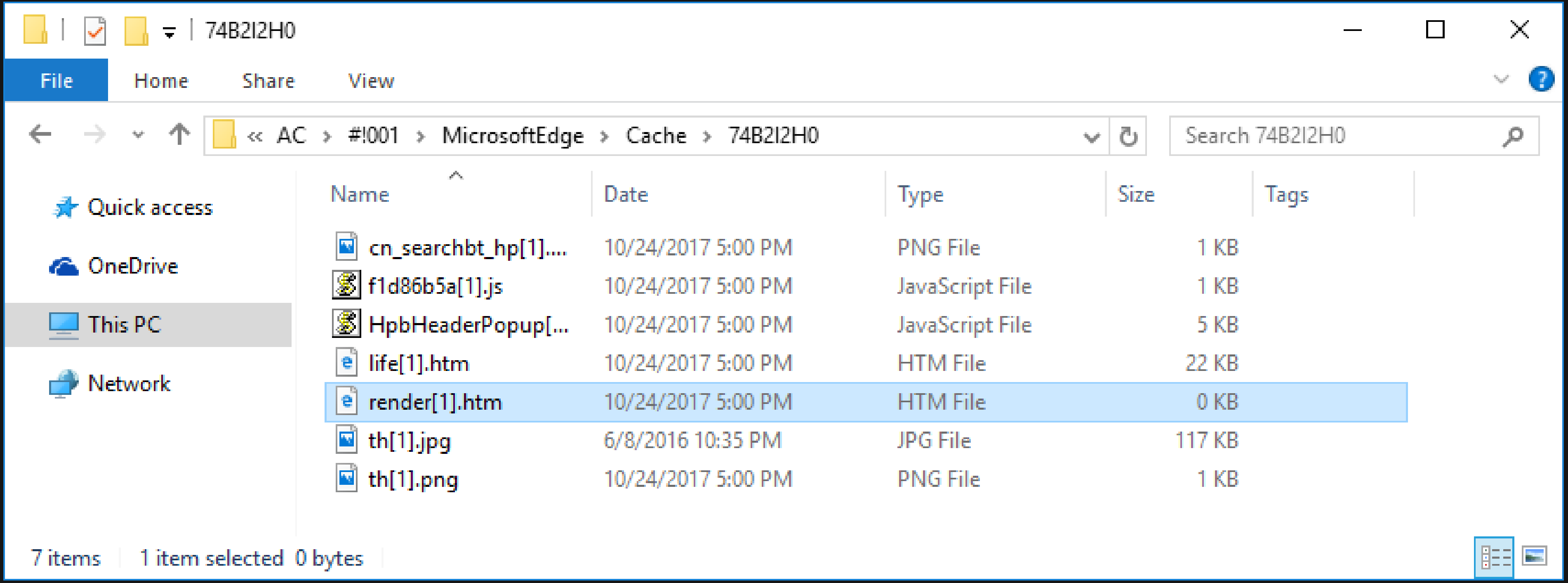


iexplore.exe	0.03	11,096 K	43,112 K	12576	Internet Explorer	Microsoft Corporation	Medium
iexplore.exe	0.01	20,888 K	49,[illegible] K	13348	Internet Explorer	Microsoft Corporation	Low

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...

The Era Of Windows 10

- Load Library
  - Browser cache will deliver the library to local



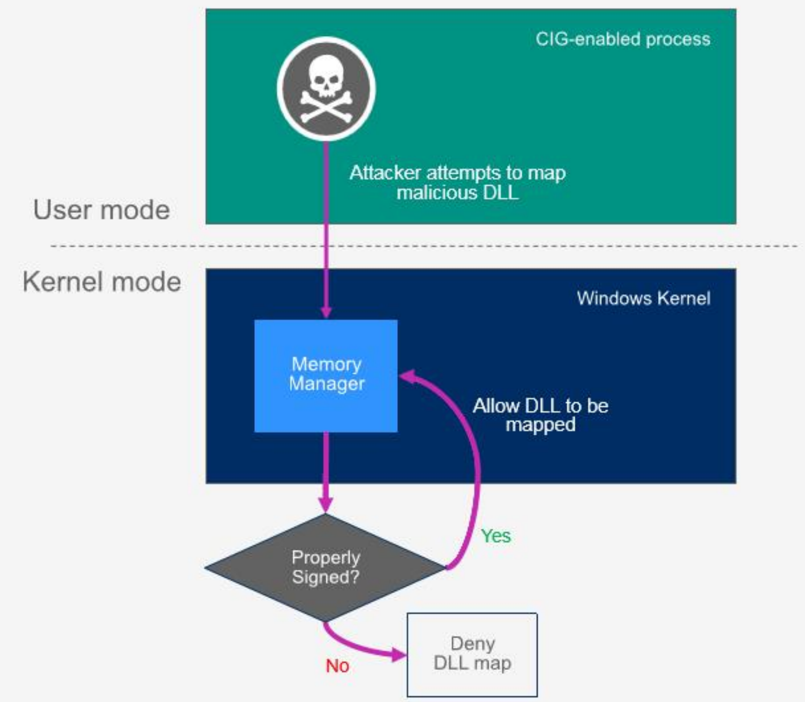
Name	Date	Type	Size	Tags
cn_searchbt_hp[1]....	10/24/2017 5:00 PM	PNG File	1 KB	
f1d86b5a[1].js	10/24/2017 5:00 PM	JavaScript File	1 KB	
HpbHeaderPopup[...	10/24/2017 5:00 PM	JavaScript File	5 KB	
life[1].htm	10/24/2017 5:00 PM	HTM File	22 KB	
render[1].htm	10/24/2017 5:00 PM	HTM File	0 KB	
th[1].jpg	6/8/2016 10:35 PM	JPG File	117 KB	
th[1].png	10/24/2017 5:00 PM	PNG File	1 KB	

7 items 1 item selected 0 bytes

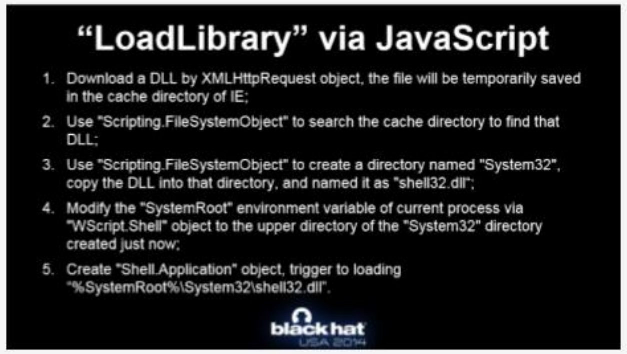
NSFOCUS

The Era Of Windows 10

- Load Library
  - CIG prevent load untrusted library



Example of such an attack provided by Yang Yu @ Black Hat USA 2014



"LoadLibrary" via JavaScript

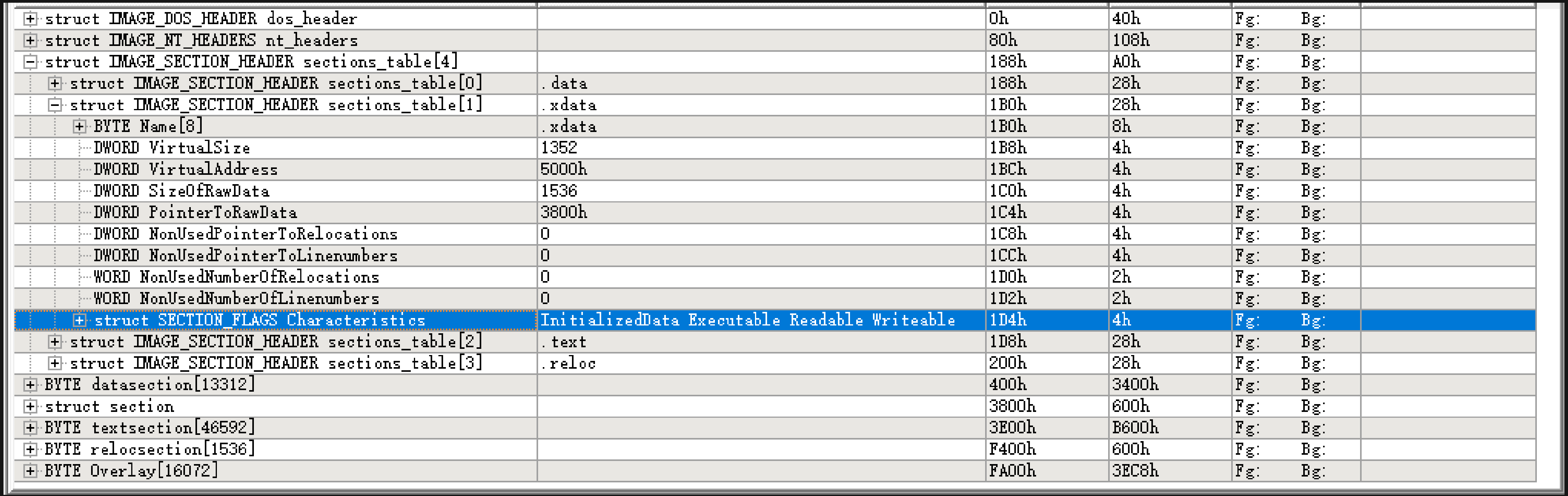
1. Download a DLL by XMLHttpRequest object, the file will be temporarily saved in the cache directory of IE;
2. Use "Scripting.FileSystemObject" to search the cache directory to find that DLL;
3. Use "Scripting.FileSystemObject" to create a directory named "System32", copy the DLL into that directory, and named it as "shell32.dll";
4. Modify the "SystemRoot" environment variable of current process via "WScript.Shell" object to the upper directory of the "System32" directory created just now;
5. Create "Shell.Application" object, trigger to loading "%SystemRoot%\System32\shell32.dll".

black hat USA 2014

Code Integrity Policies
Only allow Microsoft-signed DLLs
Only allow Microsoft, Store, or WHQL signed DLLs
Configurable via Device Guard policy

The Era Of Windows 10

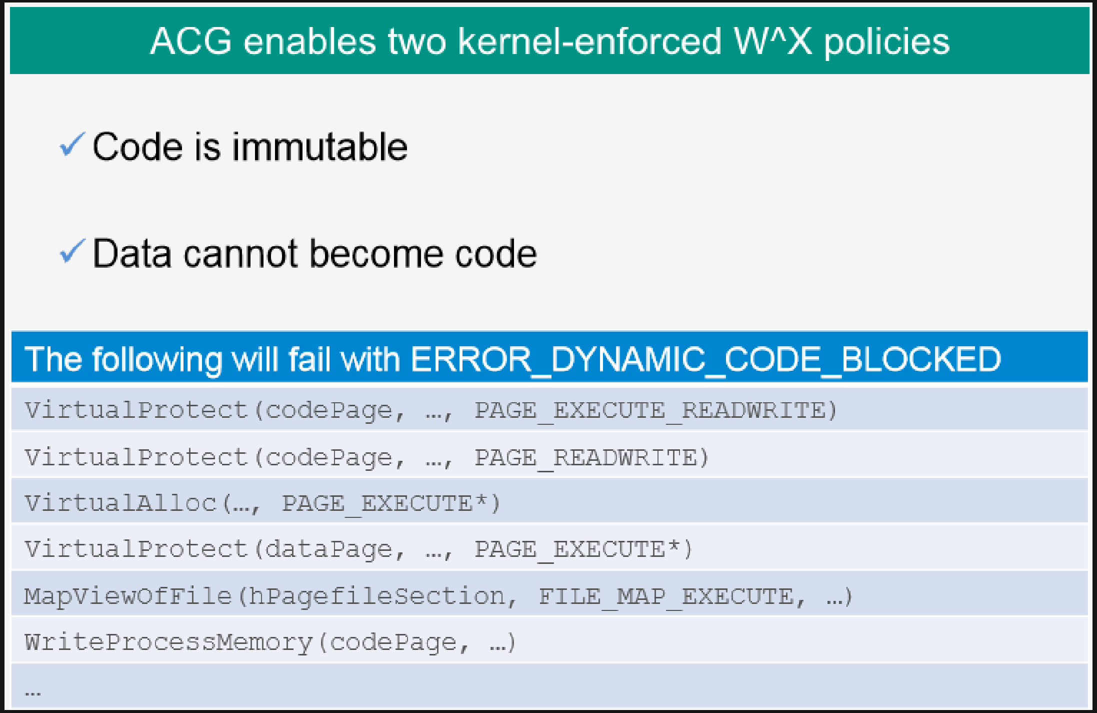
- Load Library
  - Microsoft signed .net Native Image contain a RWX .xdata section



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:

The Era Of Windows 10

- Load Library
  - ACG prevent create RWX section



ACG enables two kernel-enforced W^X policies

✓ Code is immutable

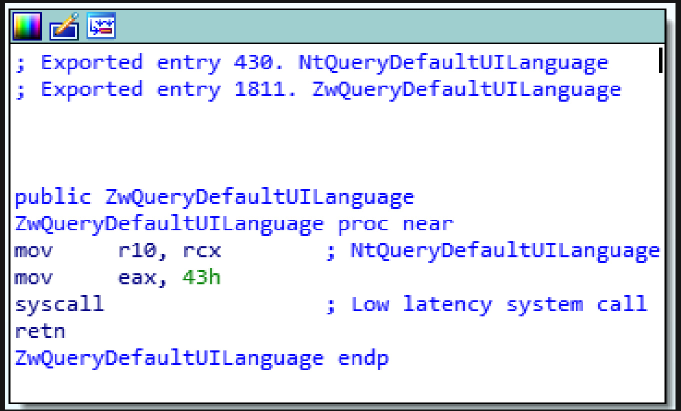
✓ Data cannot become code

The following will fail with ERROR_DYNAMIC_CODE_BLOCKED

```
VirtualProtect(codePage, …, PAGE_EXECUTE_READWRITE)
VirtualProtect(codePage, …, PAGE_READWRITE)
VirtualAlloc(…, PAGE_EXECUTE*)
VirtualProtect(dataPage, …, PAGE_EXECUTE*)
MapViewOfFile(hPagefileSection, FILE_MAP_EXECUTE, …)
WriteProcessMemory(codePage, …)
…
```

The Era Of Windows 10

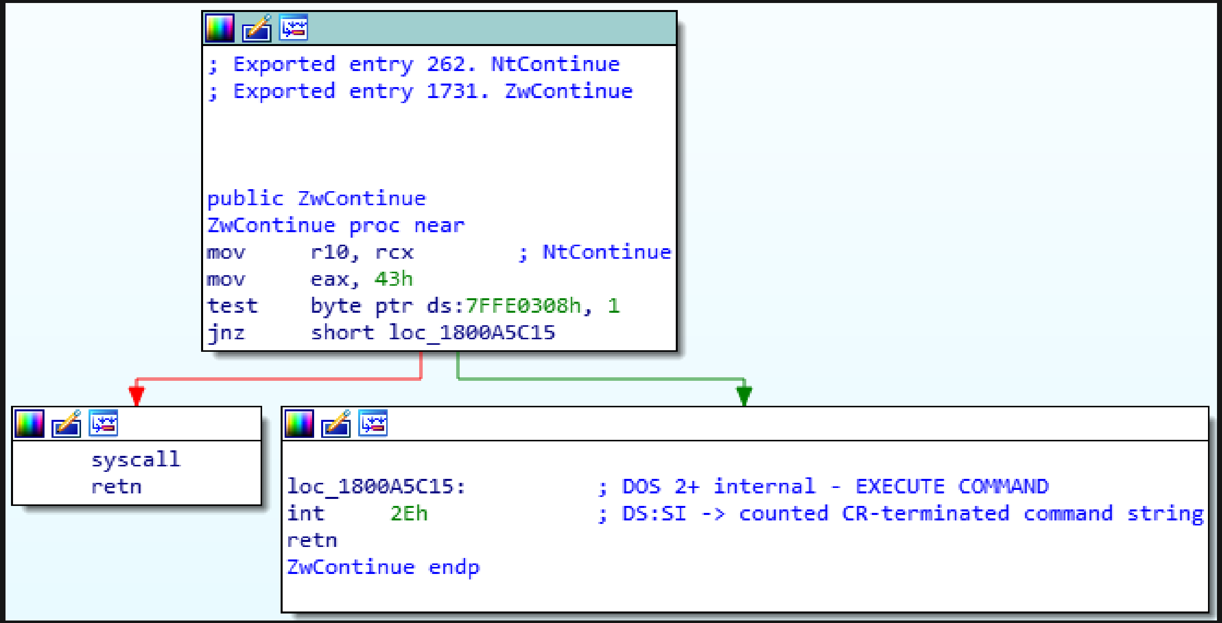
- Load Library
  - Load an old version of ntdll.dll to call NtContinue



```
; 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
```

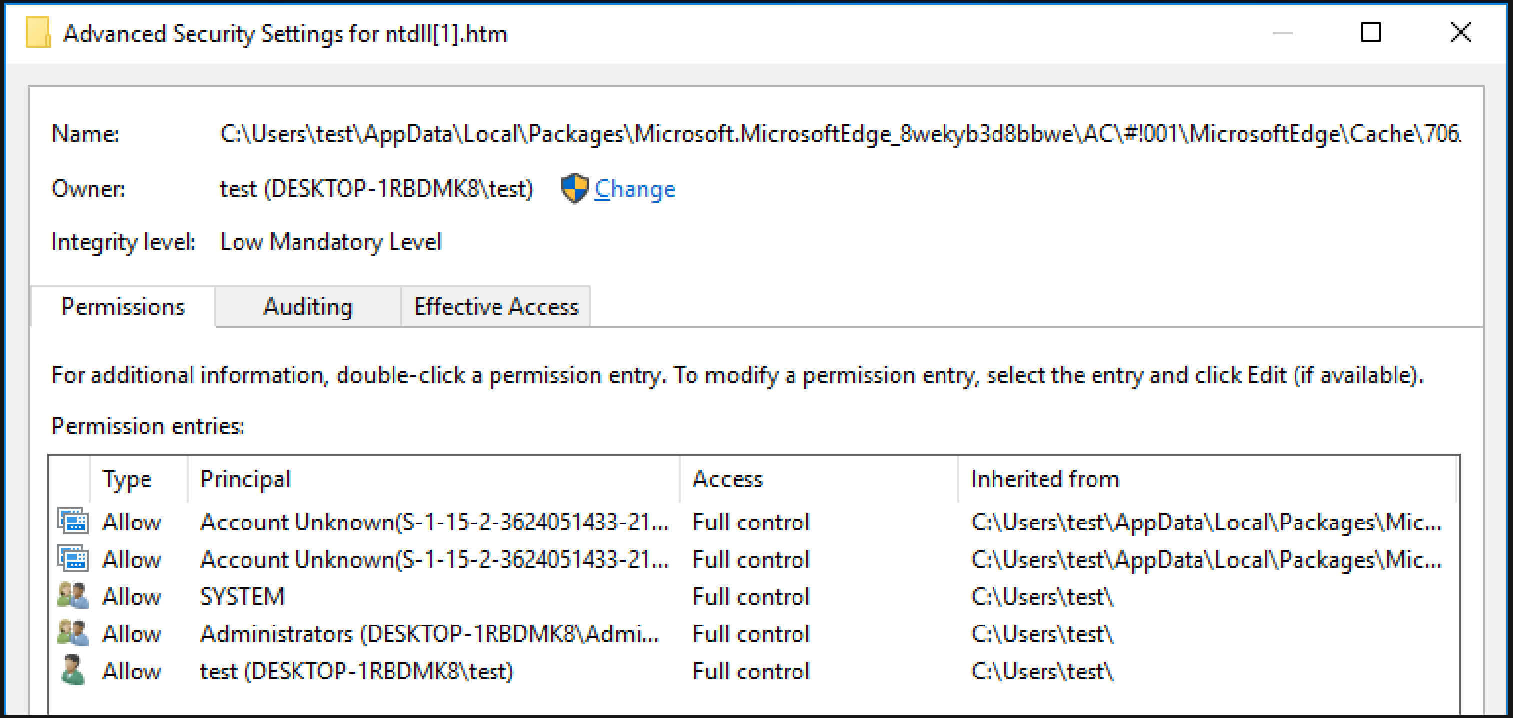
ntdll.dll version 10.0.15063.0

NSFOCUS

The Era Of Windows 10

Load Library

- NoLowMandatoryLabelImages prevent load cached files



Advanced Security Settings for ntdll[1].htm

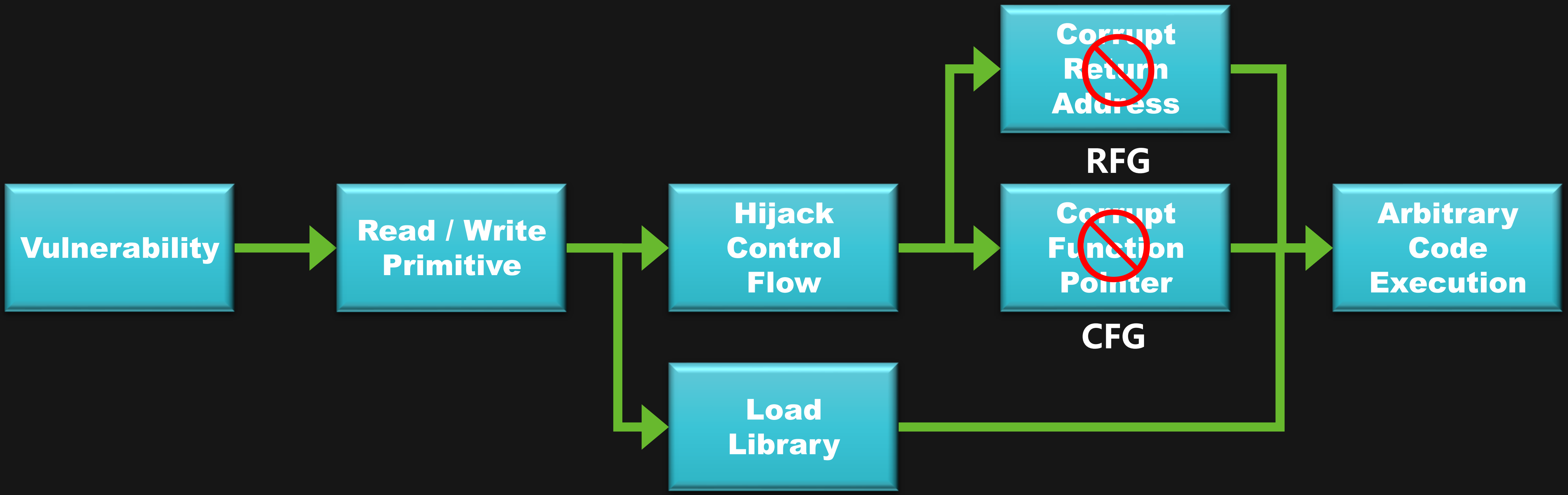
Name:	C:\Users\test\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\#!001\MicrosoftEdge\Cache\706
Owner:	test (DESKTOP-1RBDMK8\test) Change
Integrity level:	Low Mandatory Level

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

Permission entries:

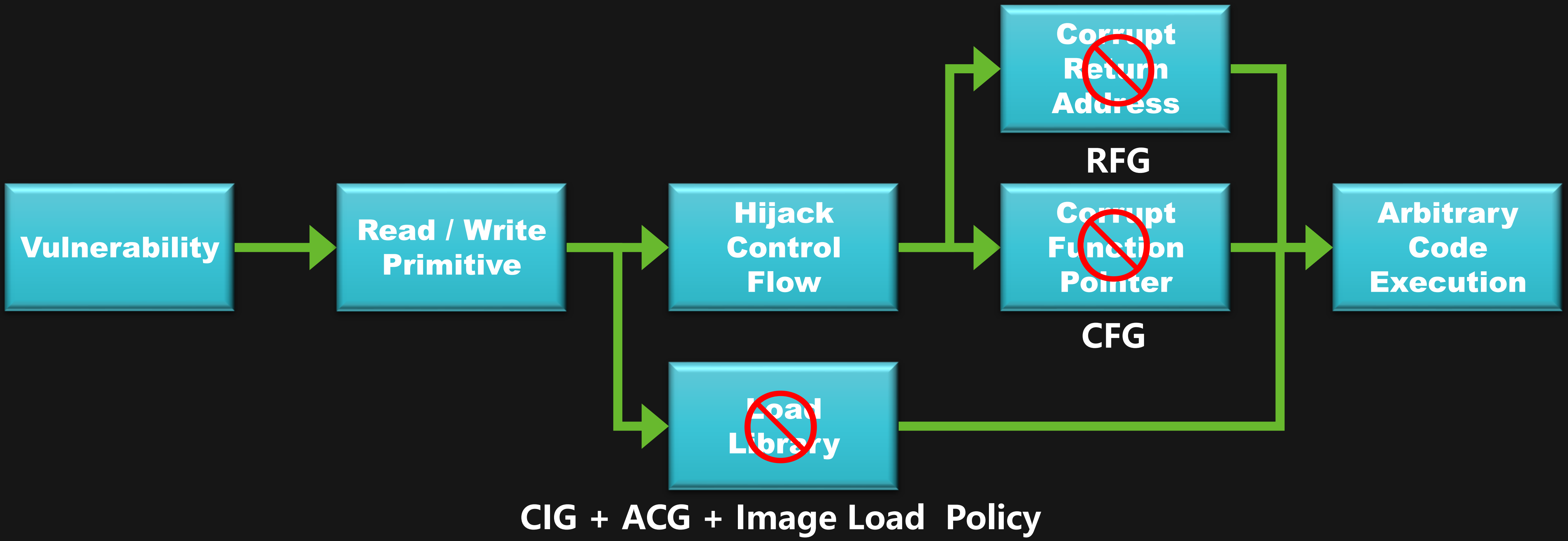
Type	Principal	Access	Inherited from
Allow	Account Unknown(S-1-15-2-3624051433-21...	Full control	C:\Users\test\AppData\Local\Packages\Mic...
Allow	Account Unknown(S-1-15-2-3624051433-21...	Full control	C:\Users\test\AppData\Local\Packages\Mic...
Allow	SYSTEM	Full control	C:\Users\test\
Allow	Administrators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

The Era Of Windows 10



NSFOCUS

The Era Of Windows 10



The Era Of Windows 10

- Abuse Feature
  - Unsafe COM object
  - JIT compile
  - Launch IE
  - Shim Hook

The Era Of Windows 10

- Abuse unsafe COM object
  - COM object can be used in different environments
  - Environments like wscript.exe allow unsafe COM object
  - Environments like browser disallow
  - Modify some flag to remove the restriction

NSFOCUS

The Era Of Windows 10

- Abuse unsafe COM object

31337

Microsoft Bounty Program
One Microsoft Way, Redmond, WA 98052

Pay to the order of James Forshaw $ 100,000

One Hundred Thousand and 00/100 Dollars USD

Memo Mitigation Bypass Technique

[illegible]
Katie Moussouris

The Era Of Windows 10

- Abuse unsafe COM object

The Era Of Windows 10

- Abuse unsafe COM object

Blue Frost Security

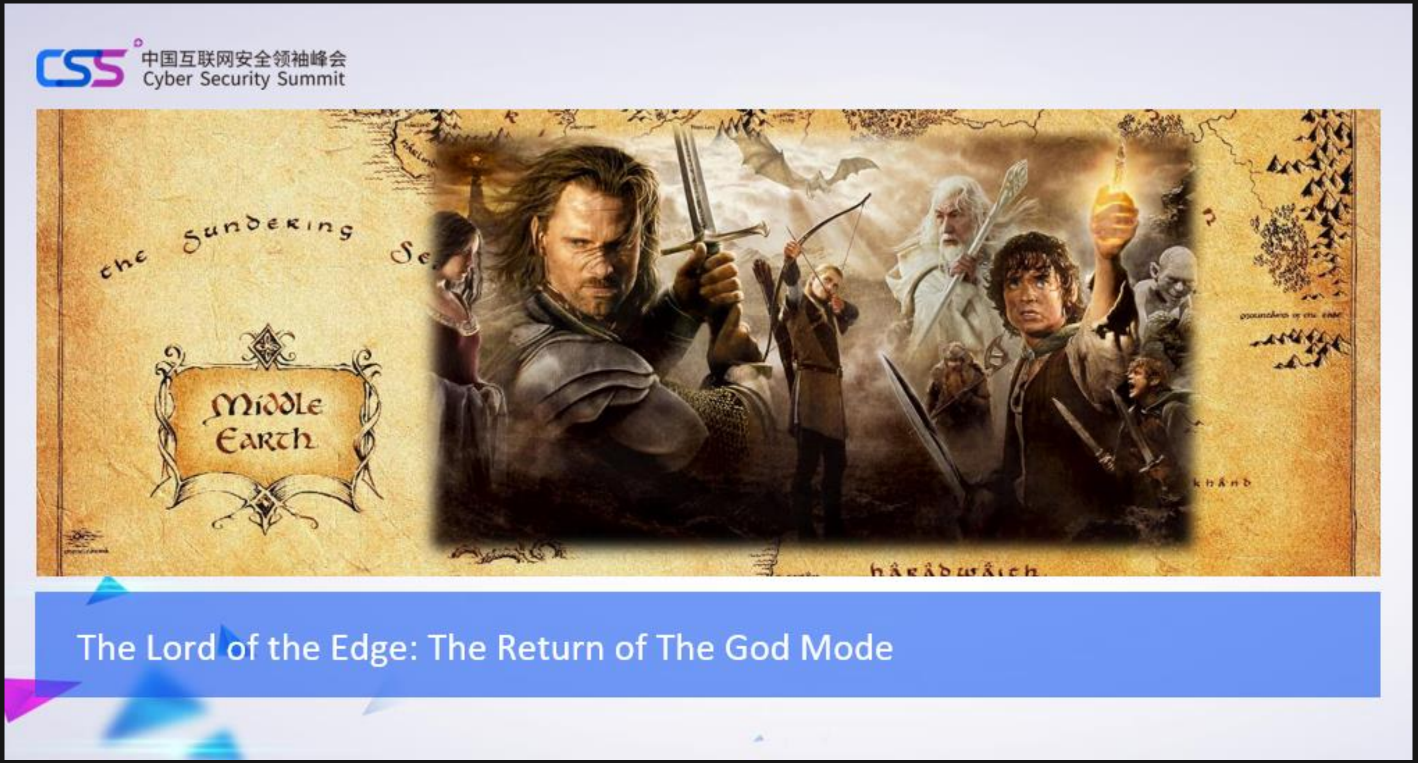
Look Mom, I don’t use Shellcode

Browser Exploitation Case Study for Internet Explorer 11

Moritz Jodeit (@moritzj)

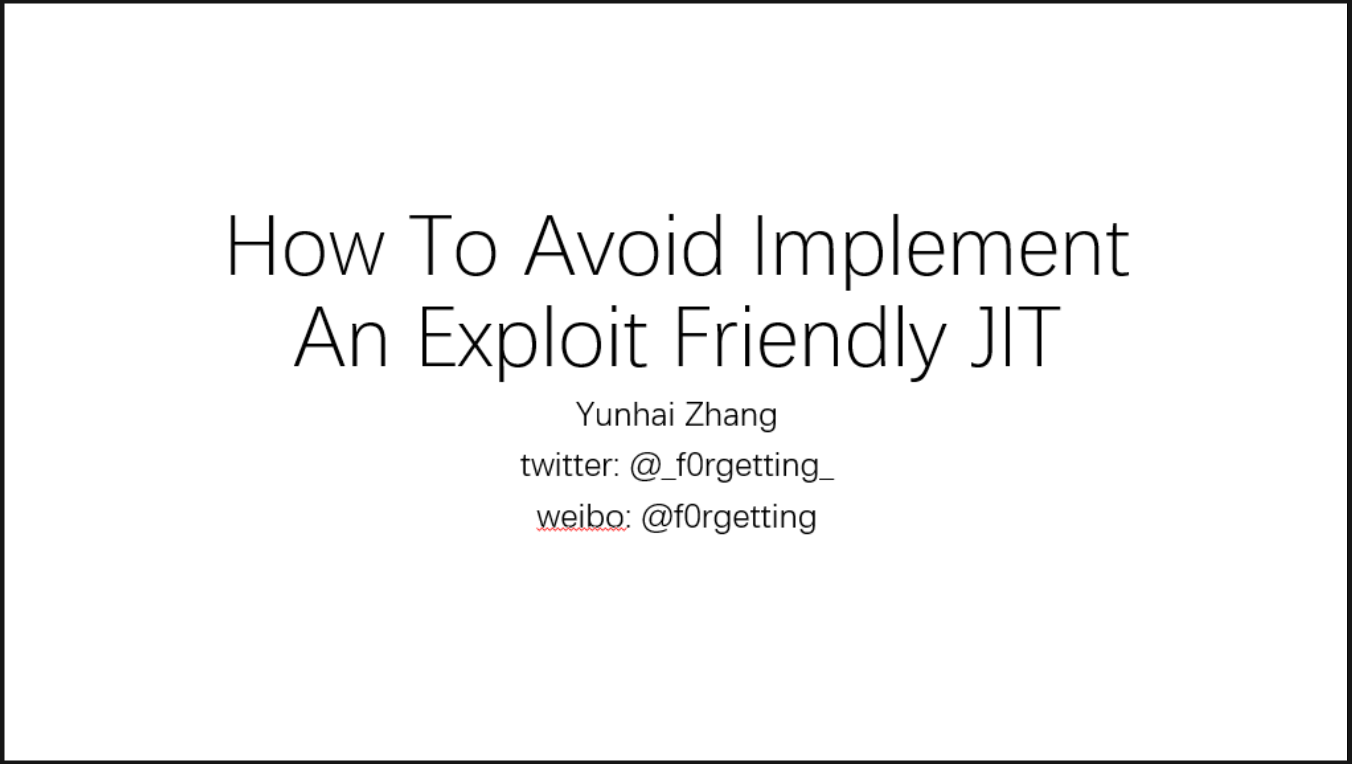
The Era Of Windows 10

- Abuse unsafe COM object



The Era Of Windows 10

- Abuse JIT compile



How To Avoid Implement An Exploit Friendly JIT

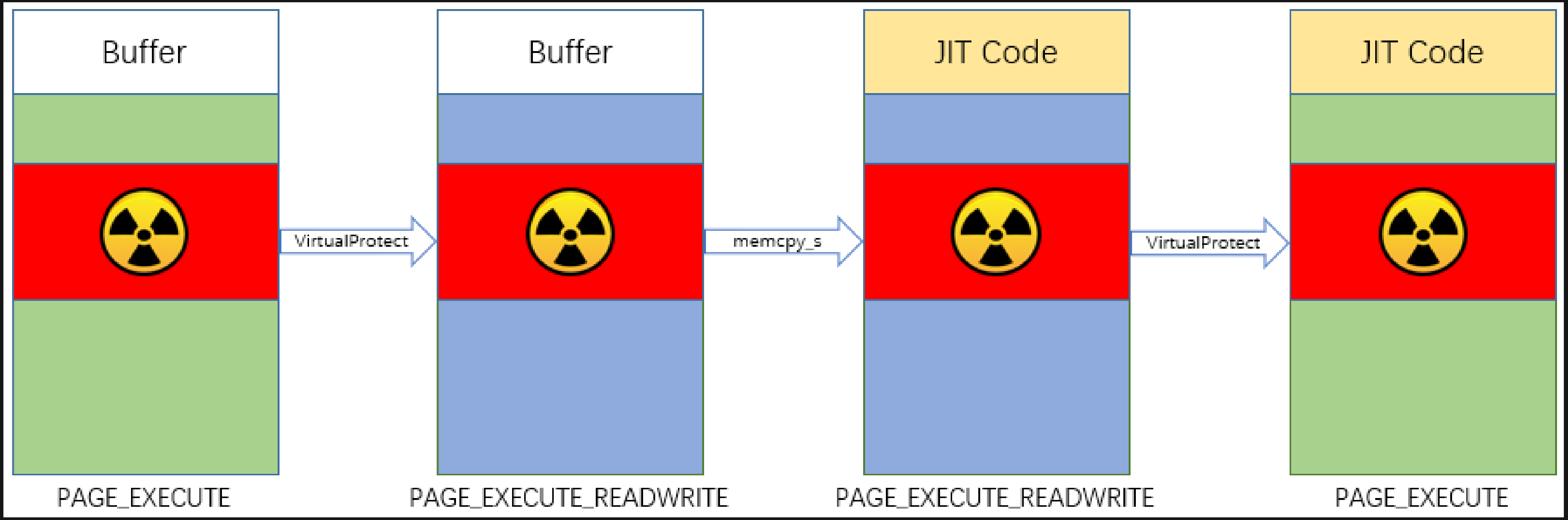
Yunhai Zhang

twitter: @_f0rgetting_

weibo: @f0rgetting

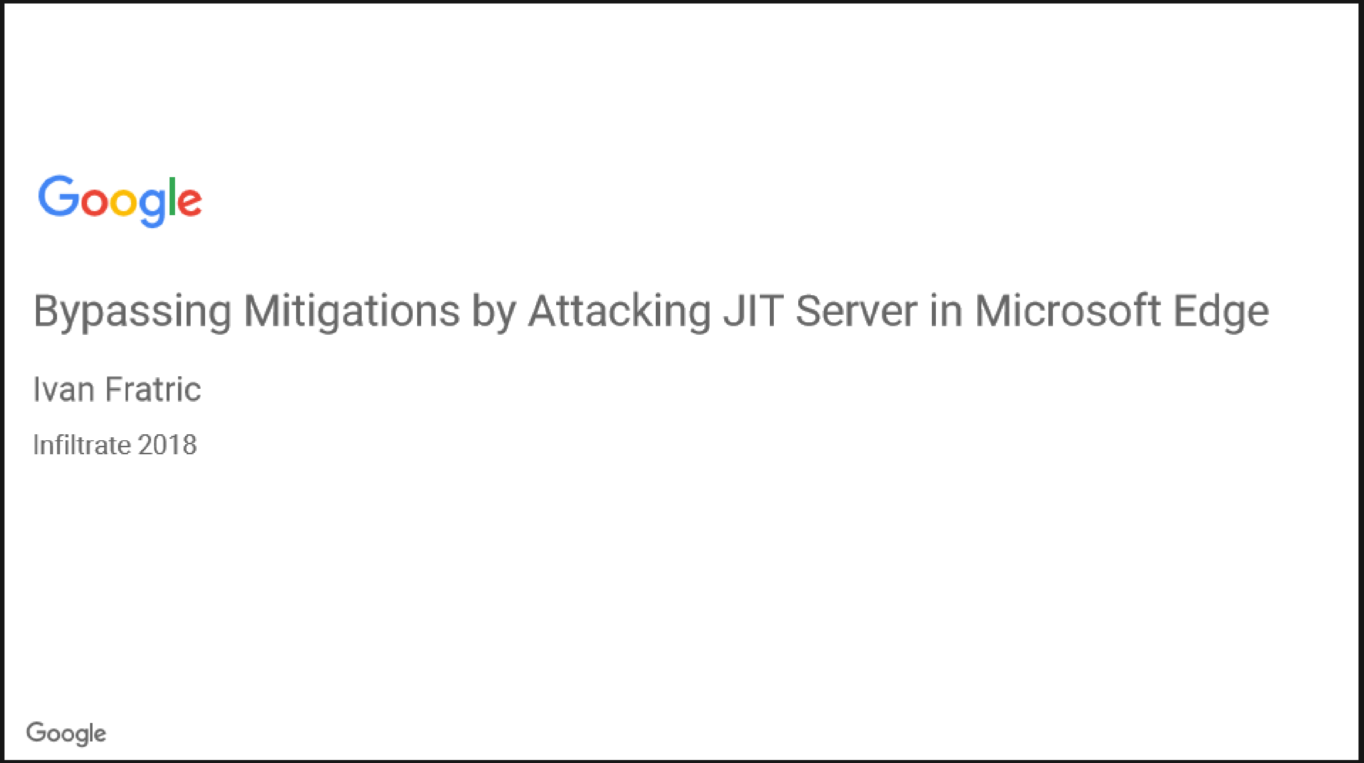
The Era Of Windows 10

- Abuse JIT compile



The Era Of Windows 10

- Abuse JIT compile



Google

Bypassing Mitigations by Attacking JIT Server in Microsoft Edge

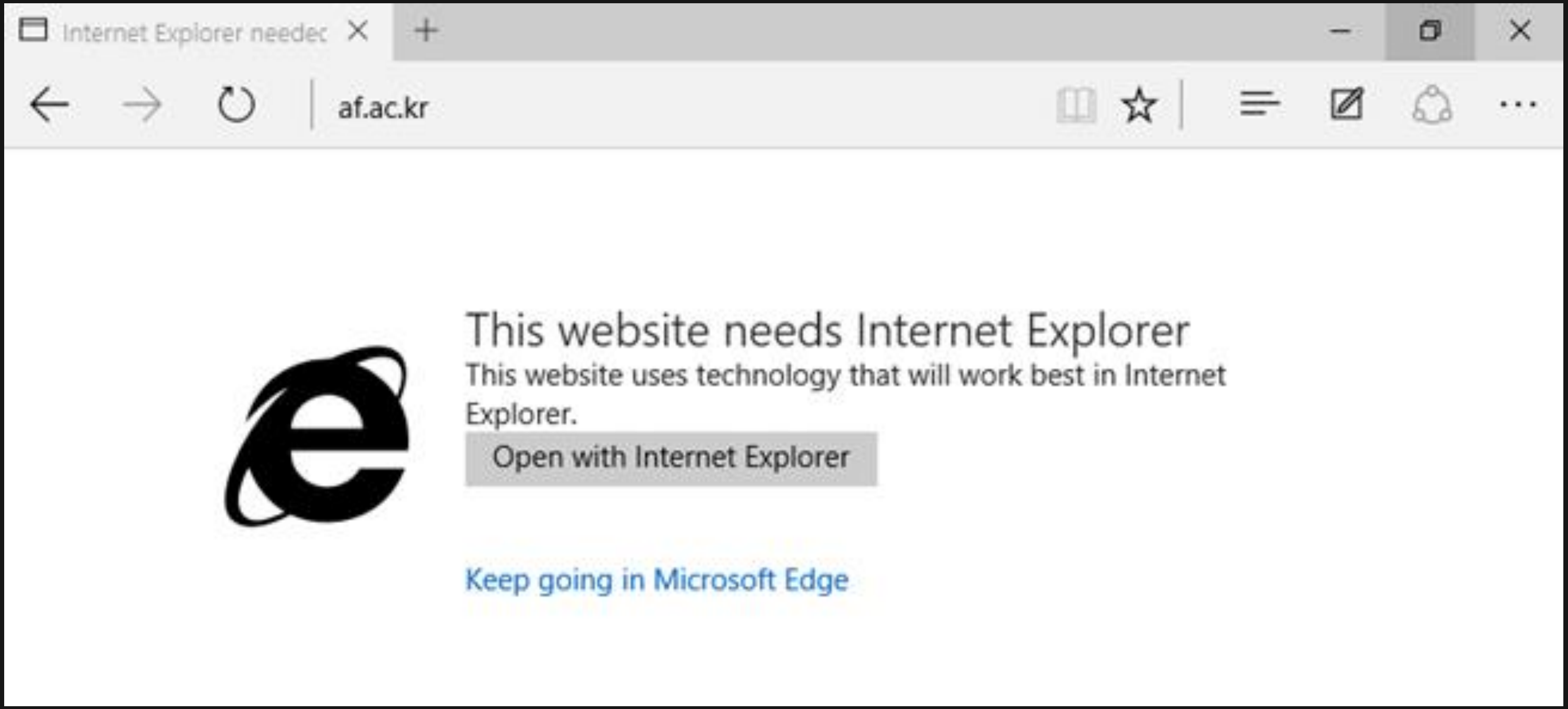
Ivan Fratric

Infiltrate 2018

Google

The Era Of Windows 10

- Abuse Launch IE

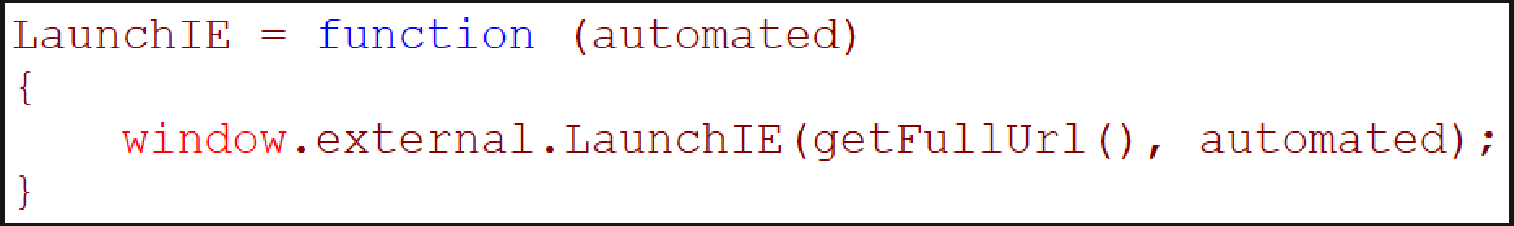


This website needs Internet Explorer

This website uses technology that will work best in Internet Explorer.

The Era Of Windows 10

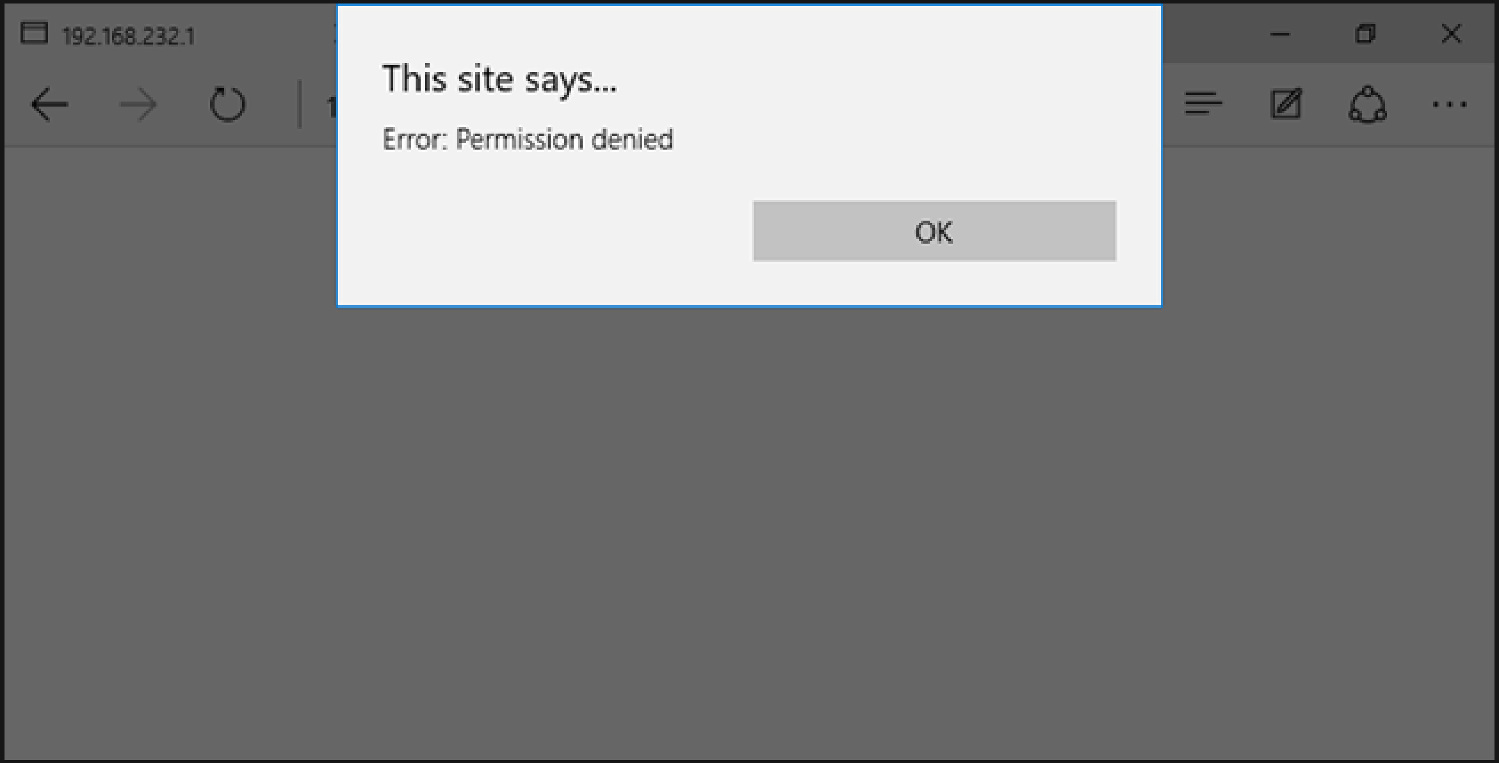
- Abuse Launch IE



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

The Era Of Windows 10

- Abuse Launch IE



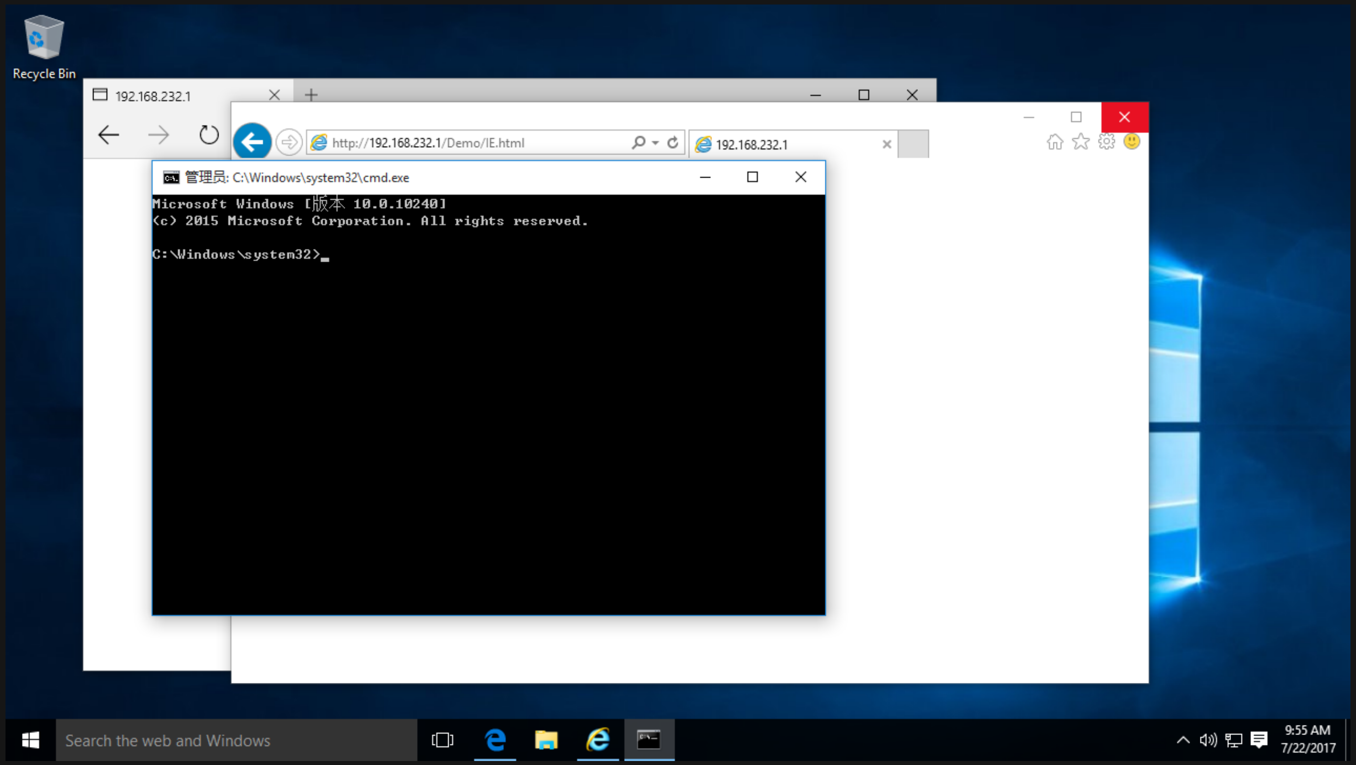
This site says...

Error: Permission denied

NSFOCUS

The Era Of Windows 10

- Abuse Launch IE

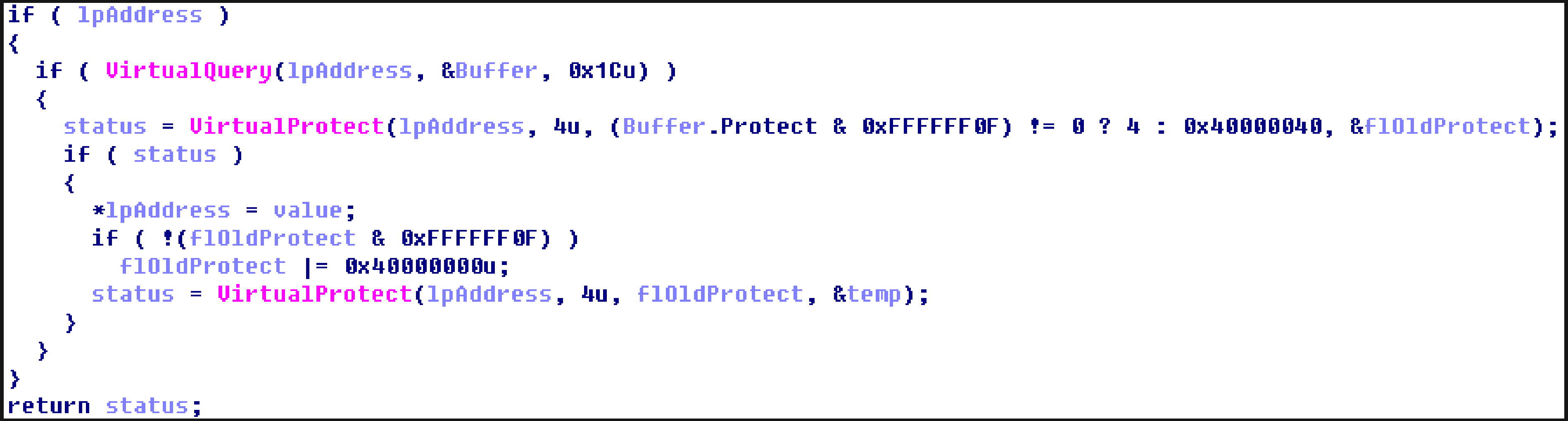

Recycle Bin
192.168.232.1
http://192.168.232.1/Demo/IE.html
192.168.232.1
管理员: C:\Windows\system32\cmd.exe
Search the web and Windows
9:55 AM
7/22/2017

```
Microsoft Windows [版本 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Windows\system32>_
```

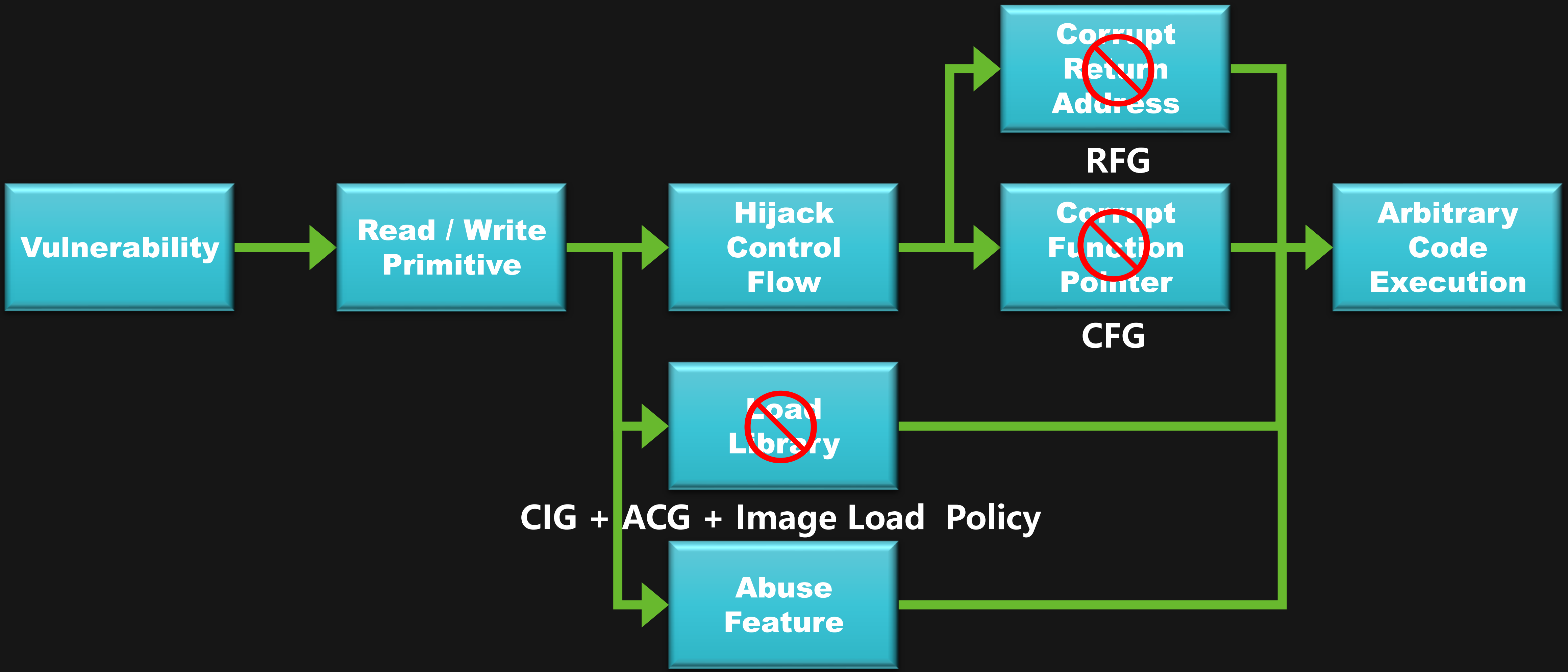
The Era Of Windows 10

- Abuse Shim Hook
  - EShims.dll will hook particular functions by modifying IAT
  - The hook is guided by meta data in BindingRef::s_pBindings
  - Modify the meta data to write any location including read-only areas

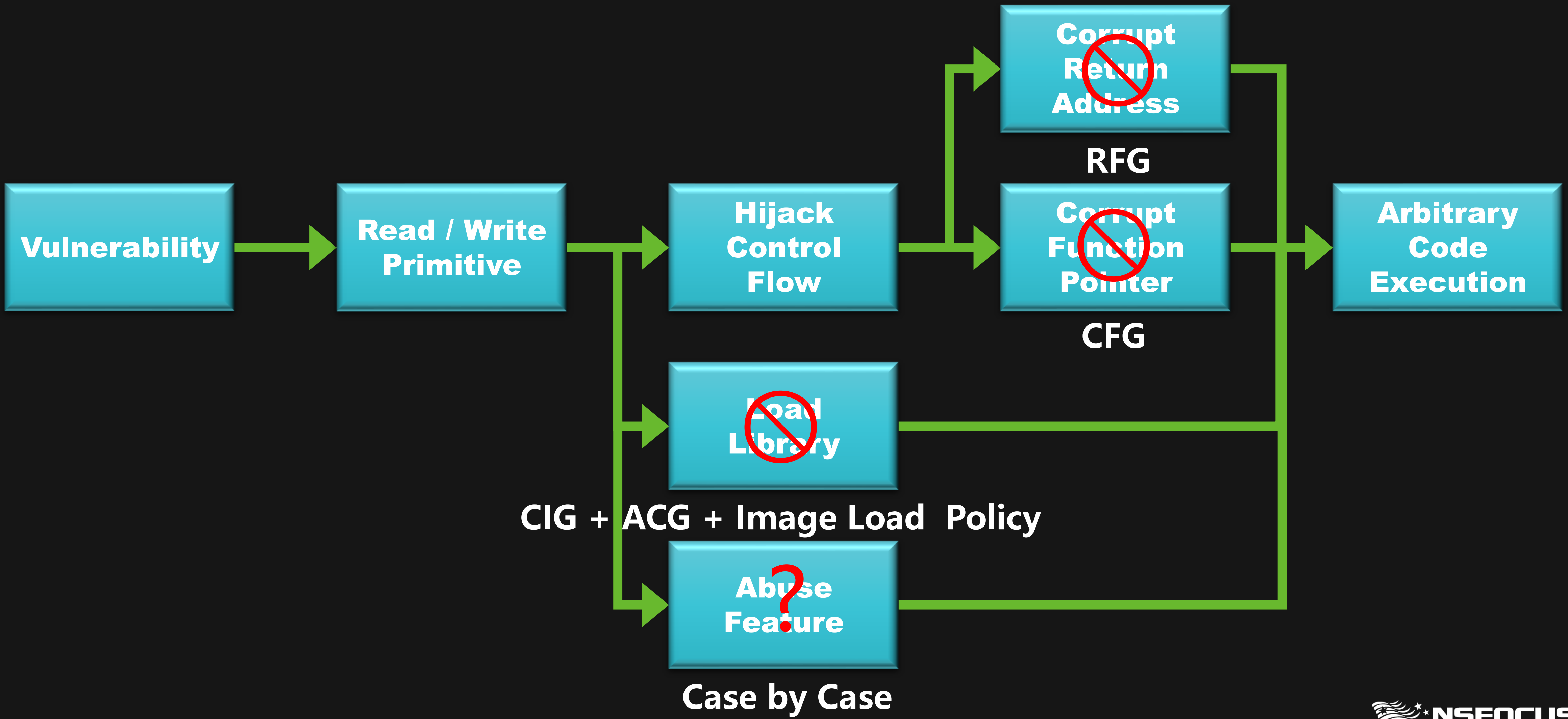


```
if ( lpAddress )
{
  if ( VirtualQuery(lpAddress, &Buffer, 0x1Cu) )
  {
    status = VirtualProtect(lpAddress, 4u, (Buffer.Protect & 0xFFFFFF0F) != 0 ? 4 : 0x40000040, &flOldProtect);
    if ( status )
    {
      *lpAddress = value;
      if ( !(flOldProtect & 0xFFFFFF0F) )
        flOldProtect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

The Era Of Windows 10

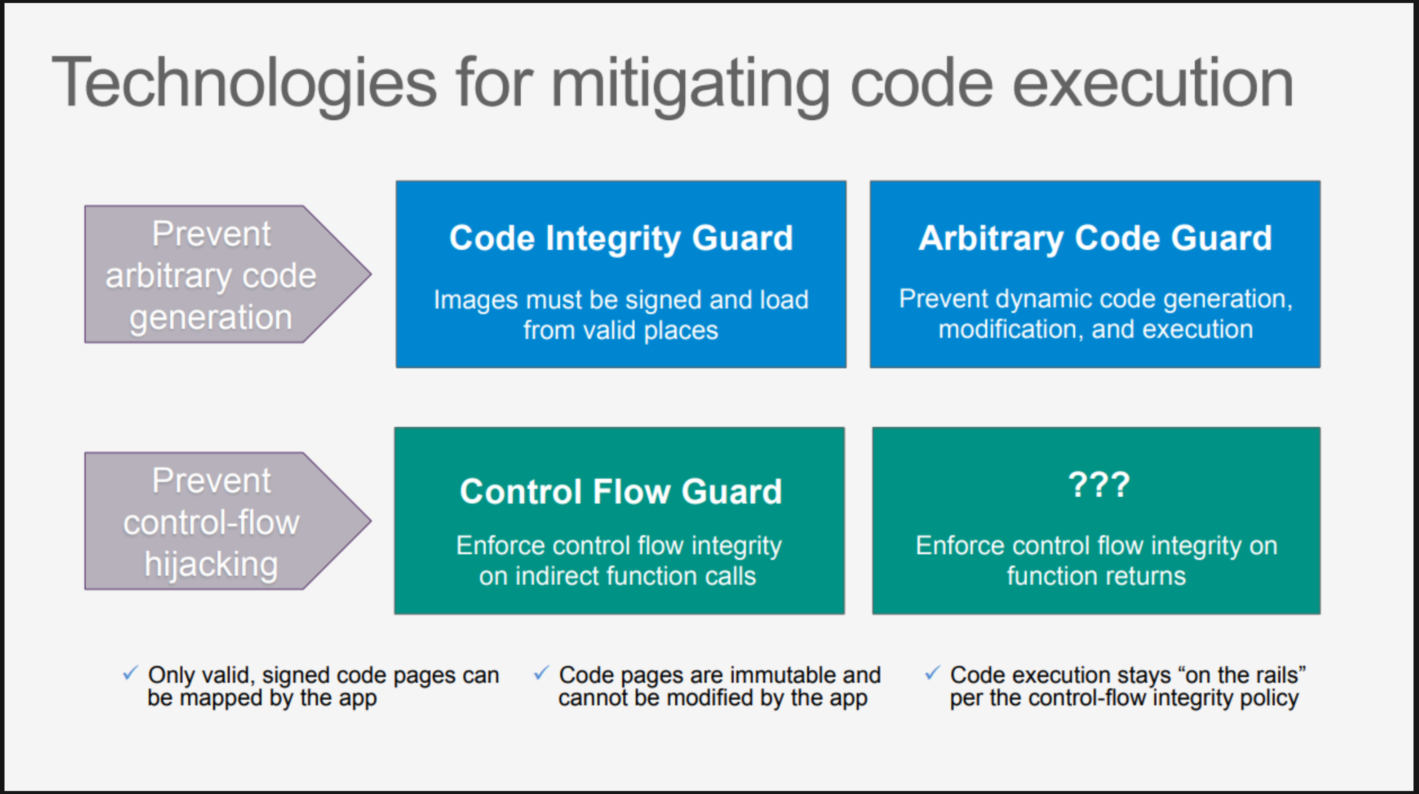


The Era Of Windows 10



The Future of Mitigation

- Intel CET based RFG



Technologies for mitigating code execution

Prevent arbitrary code generation
Code Integrity Guard
Images must be signed and load from valid places
Arbitrary Code Guard
Prevent dynamic code generation, modification, and execution
Prevent control-flow hijacking
Control Flow Guard
Enforce control flow integrity on indirect function calls
???
Enforce control flow integrity on function returns

✓ Only valid, signed code pages can be mapped by the app

✓ Code pages are immutable and cannot be modified by the app

✓ Code execution stays "on the rails" per the control-flow integrity policy

The Future of Mitigation

- Fine Grained CFG

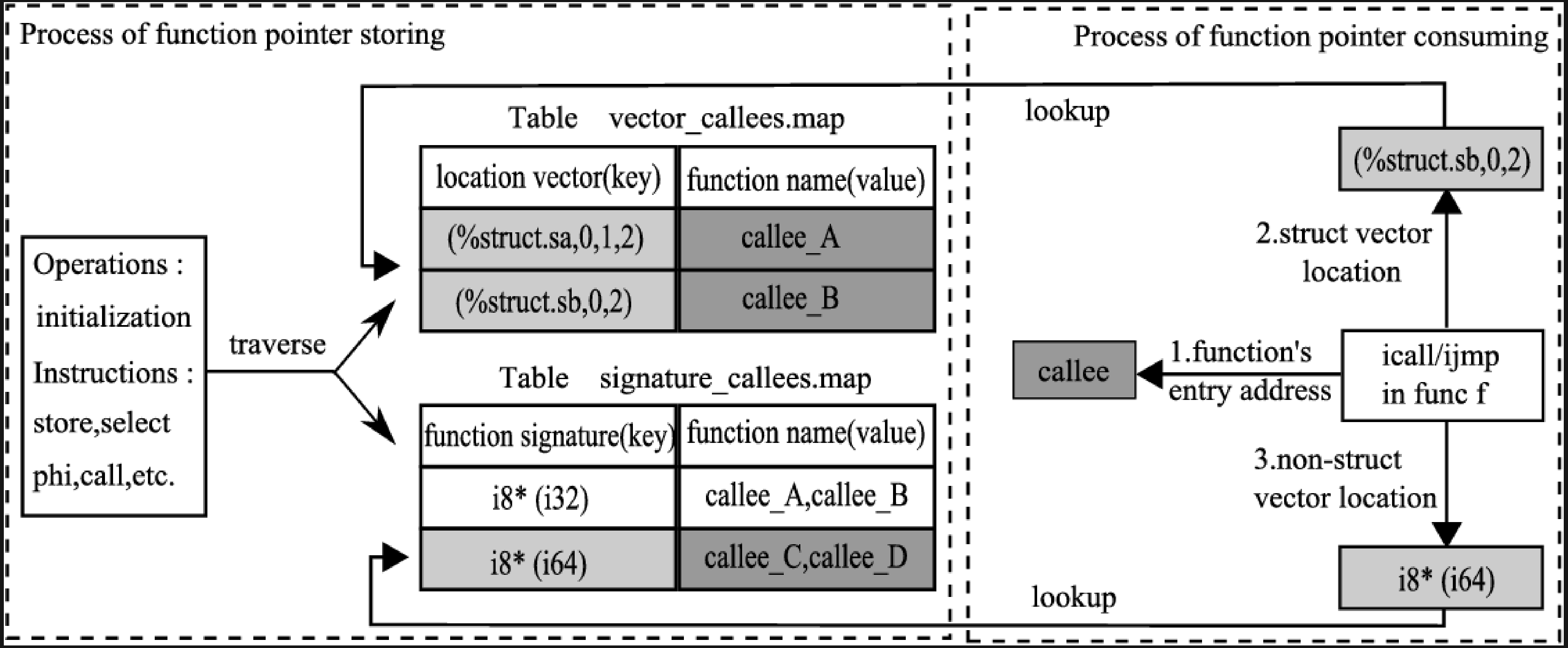

Process of function pointer storing
Process of function pointer consuming
Operations : initialization
Instructions : store,select phi,call,etc.
traverse
lookup
(%struct.sb,0,2)
2.struct vector location
callee
1.function's entry address
icall/ijmp in func f
3.non-struct vector location
i8* (i64)
lookup

Table vector_callees.map

location vector(key)	function name(value)
(%struct.sa,0,1,2)	callee_A
(%struct.sb,0,2)	callee_B

Table signature_callees.map

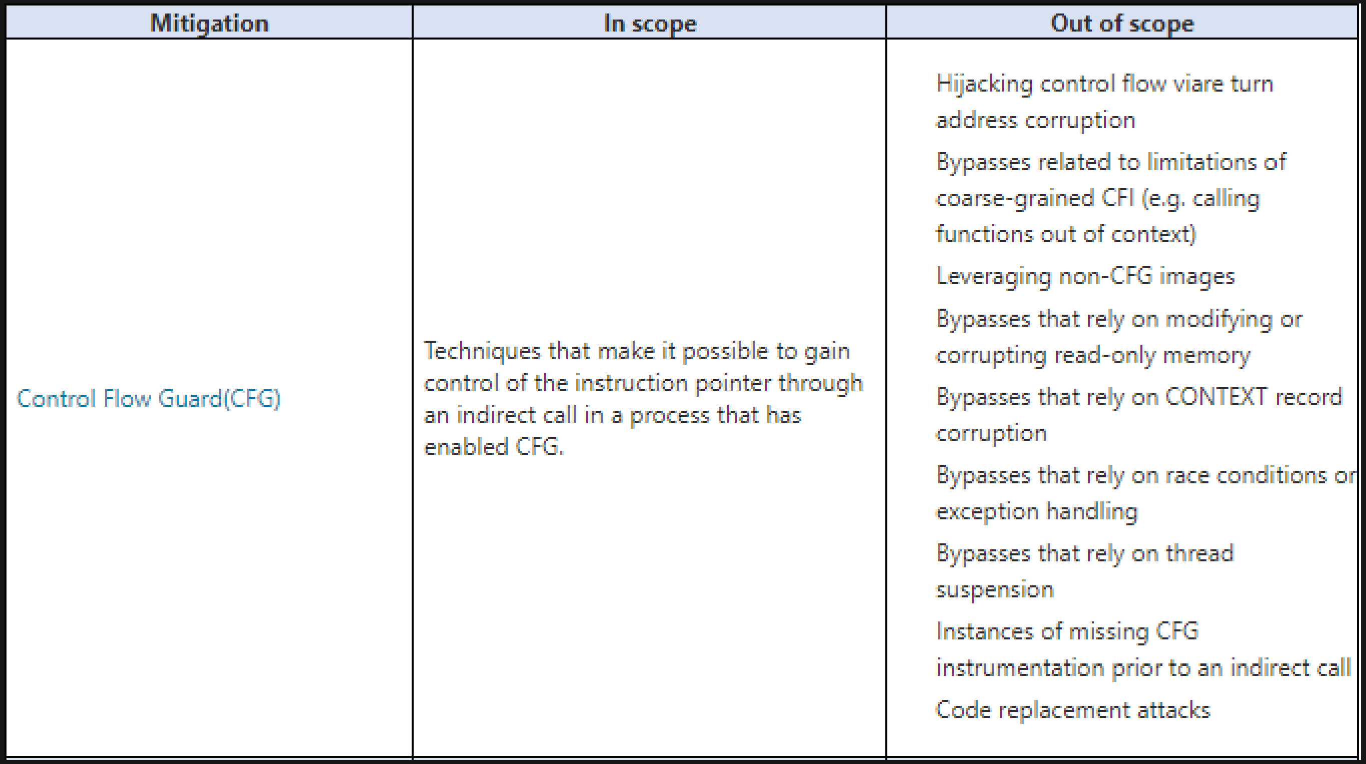
function signature(key)	function name(value)
i8* (i32)	callee_A,callee_B
i8* (i64)	callee_C,callee_D

The Future of Mitigation Bypass

- Hijack control flow is still possible in Fine Grained CFG
  - Nowadays non-trivial applications are complicated
  - Indirect calls with multiple valid targets exist
  - Some targets are not allow to be called for logical reason

The Future of Mitigation

- Existing CFG bypass techniques



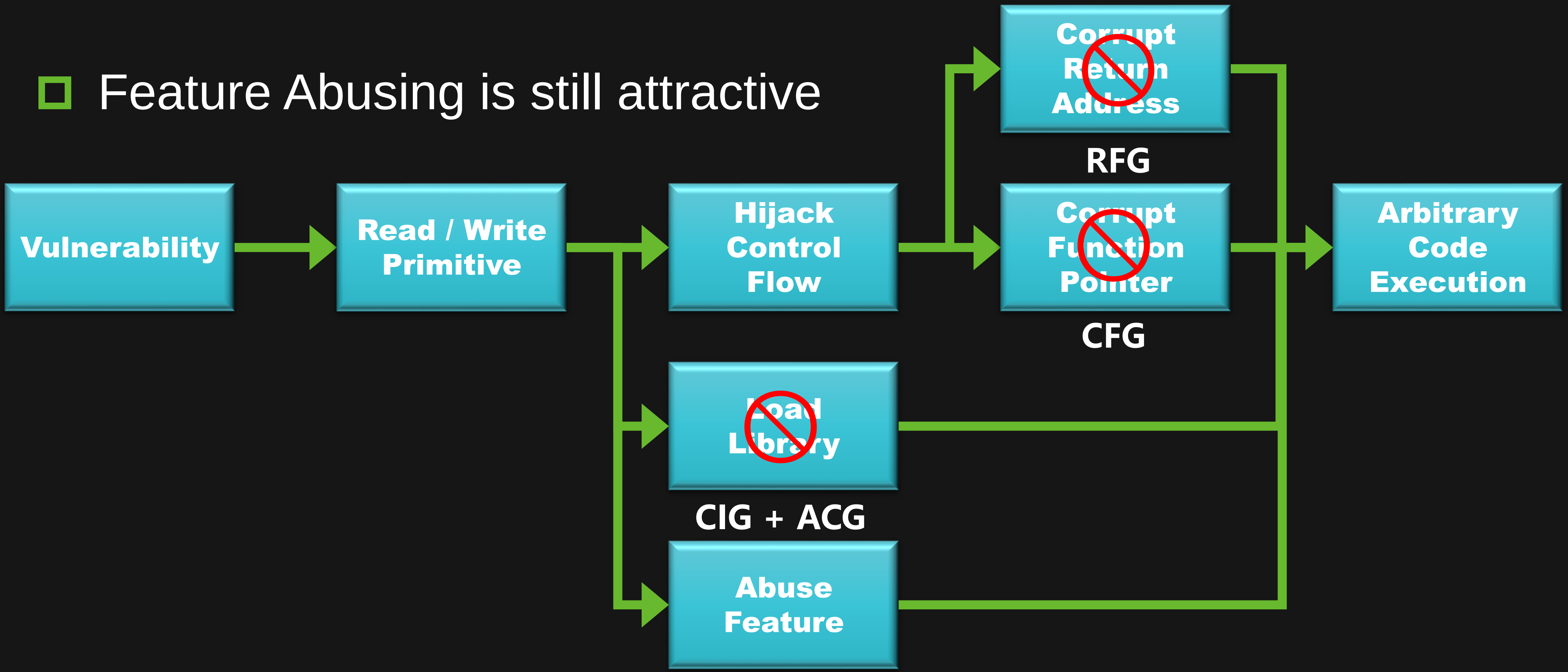
Mitigation	In scope	Out of scope
Control Flow Guard(CFG)	Techniques that make it possible to gain control of the instruction pointer through an indirect call in a process that has enabled CFG.	Hijacking control flow viare turn address corruption Bypasses related to limitations of coarse-grained CFI (e.g. calling functions out of context) Leveraging non-CFG images Bypasses that rely on modifying or corrupting read-only memory Bypasses that rely on CONTEXT record corruption Bypasses that rely on race conditions or exception handling Bypasses that rely on thread suspension Instances of missing CFG instrumentation prior to an indirect call Code replacement attacks

The Future of Mitigation Bypass

- Too many critical data are protected by read-only
  - __guard_dispatch_icall_fptr
  - IAT
  - NtUserPfn
  - Wow64ApcRoutine
  - ...

The Future of Mitigation Bypass

- Feature Abusing is still attractive



NSFOCUS

The Future of Mitigation Bypass

- Feature Abusing is still attractive
  - .net / PowerShell
  - Delay Load Library
  - OOP
  - Extension
  - ...

The Future of Mitigation Bypass

- No Silver Bullet

Q&A

NSFOCUS

Thanks!

NSFOCUS