

# Du driver Windows à l'EDR

Introduction au fonctionnement des drivers Windows et EDR's

# whoami /all



Éditer le profil

**Aurélien Chalot**  
@Defte\_

Hacker, sysadmin and security researcher @OrangeCyberdef 🖥️  
Calisthenic enthusiast 💪 and wannabe philosopher 📖  
🔥 Hide&Sec 🔥

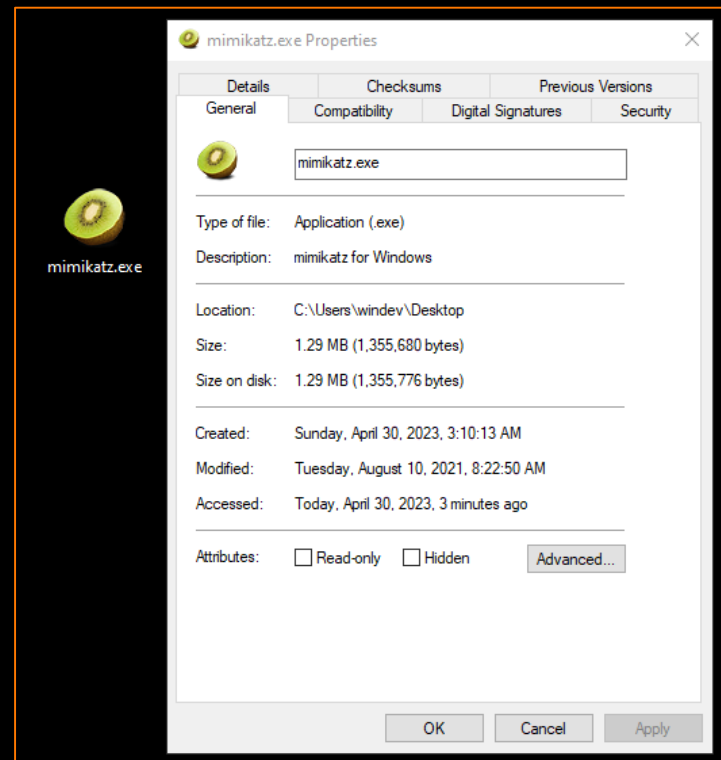
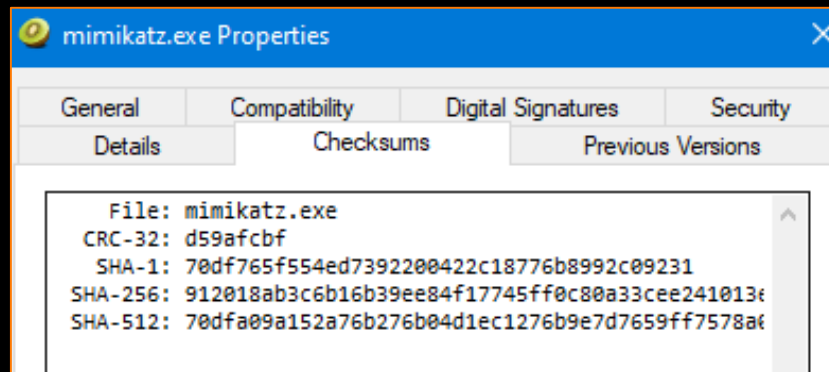
📍 The grid 🔗 [blog.whiteflag.io](https://blog.whiteflag.io) 📅 A rejoint Twitter en novembre 2017

**382** abonnements   **818** abonnés

# I / Anti-virus 101

# I / Anti-virus 101: analyse statique

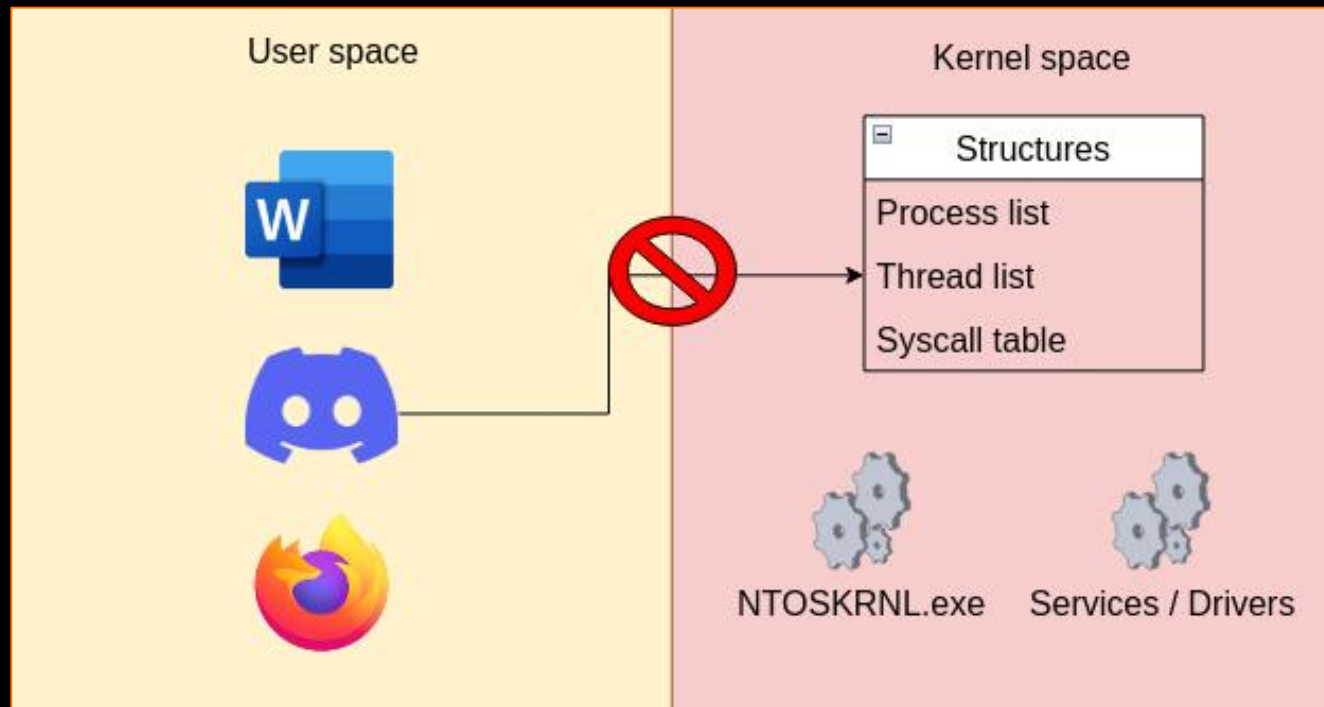
- L'antivirus détecte un virus de deux manières:
- Via une analyse heuristique simple ex:
  - Nom de l'exécutable
  - Metadata de l'exécutable
- Via une signature:



# I / Anti-virus 101: un peu de système

**User Space:** où tournent vos applications.

**Kernel space:** où tournent le cœur du système d'exploitation (le kernel), les services et les drivers.

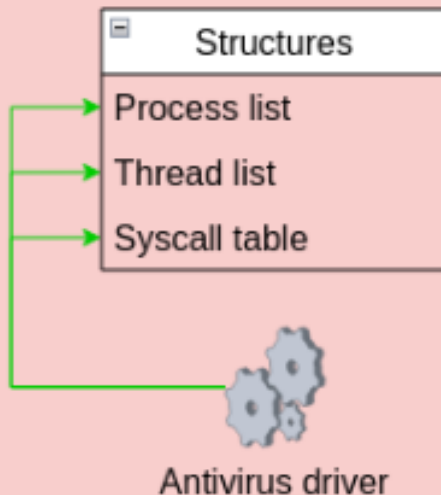


# I / Anti-virus 101: smart move

User space



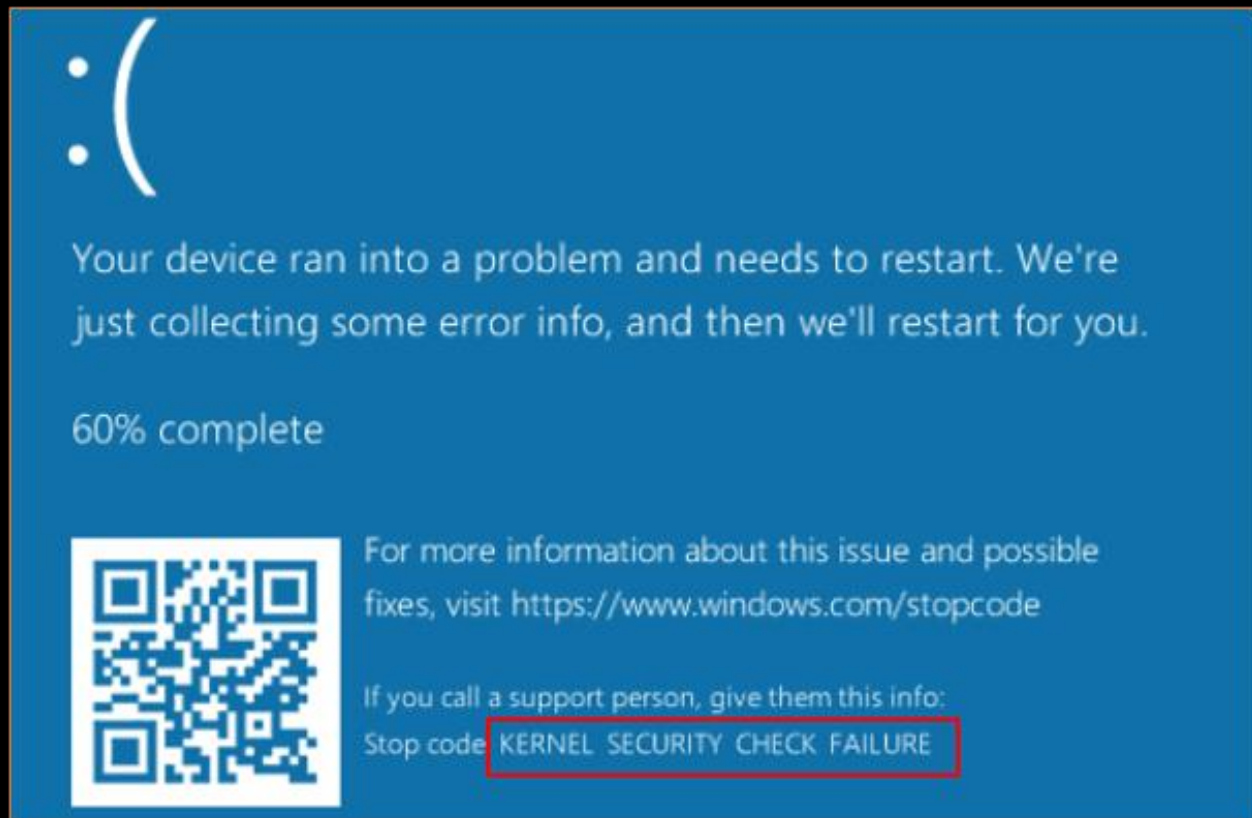
Kernel space



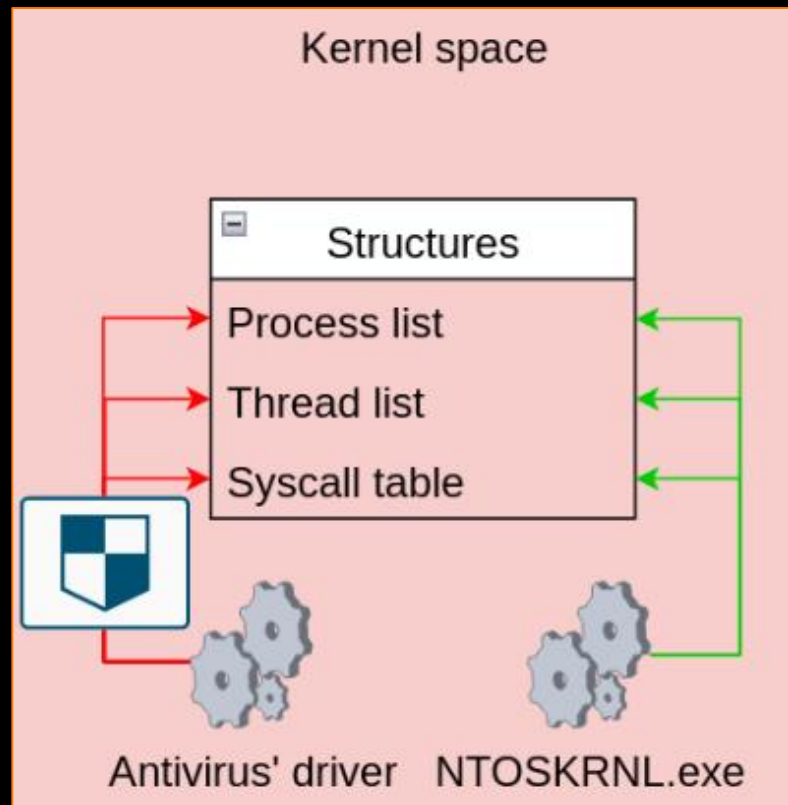
Et si on piochait /  
modifiait les  
informations  
contenues au sein du  
kernel ?



# I / Anti-virus 101: BSOD



# I / Anti-virus 101: the revenge of M\$



Microsoft déploia Patch Guard sur Windows XP/2003 afin d'empêcher l'accès et la modification de certaines structures critiques du kernel (dont celles utilisées par les antivirus)





# I / Anti-virus 101: seum over 9000

But this is all chaff to distract you from the real reason Symantec is blowing their horn so loudly. In News.com's report on the issue ("[Windows defense handcuffs the good guys](#)"), the Symantec spokesperson all but revealed the true reason for these reports:

"It seems a bit disingenuous of Microsoft. They are getting into the security market and are disallowing this whole class of security products that they don't have," McCorkendale said. "It does not feel like a level playing field at that point."

McCorkendale stopped short of saying that Symantec would sue Microsoft or complain to antitrust authorities. However, Yankee Group analyst Jaquith believes that step is getting closer, especially if Microsoft were to give its own security products a way to bypass PatchGuard.




<http://windows-now.com/blogs/robert/archive/2006/08/12/PatchGuard-and-Symantecs-Complaints-About-Windows-Vista.aspx>

# I / Anti-virus 101: la réponse de M\$

## Callback Objects

Article • 12/15/2021 • 3 contributors

 [Feedback](#)

The kernel's callback mechanism provides a general way for drivers to request and provide notification when certain conditions are satisfied.

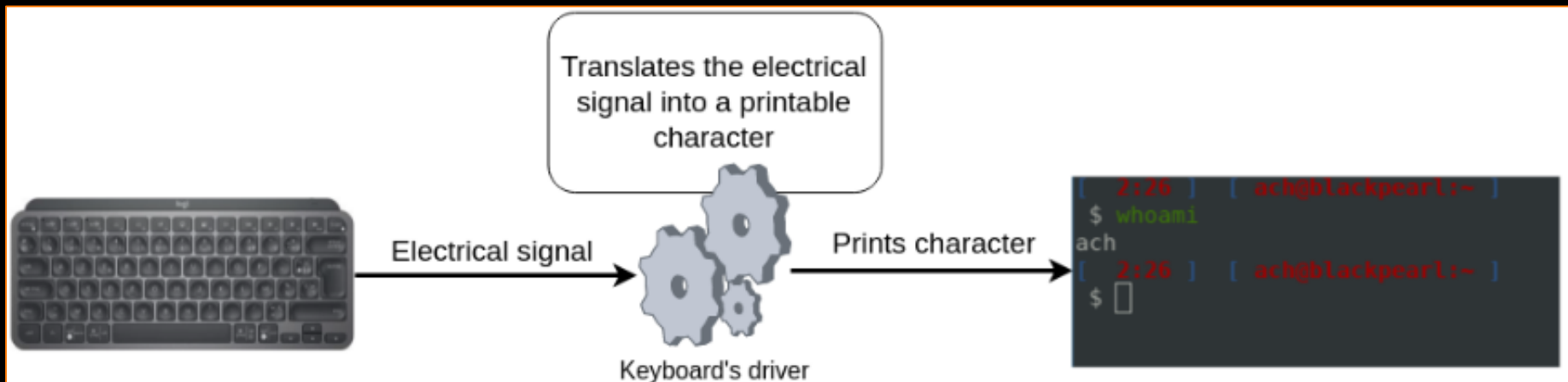
A driver can create a callback object, and other drivers can request notification for conditions associated with this driver-defined callback. In addition, the system defines three callback objects for driver use.

Par contre on va avoir besoin d'un driver...

## **II / Développer un driver Windows**

## II / Développer un driver Windows

On définit un driver comme étant un composant qui permet au kernel (un composant software) de communiquer avec un composant hardware:



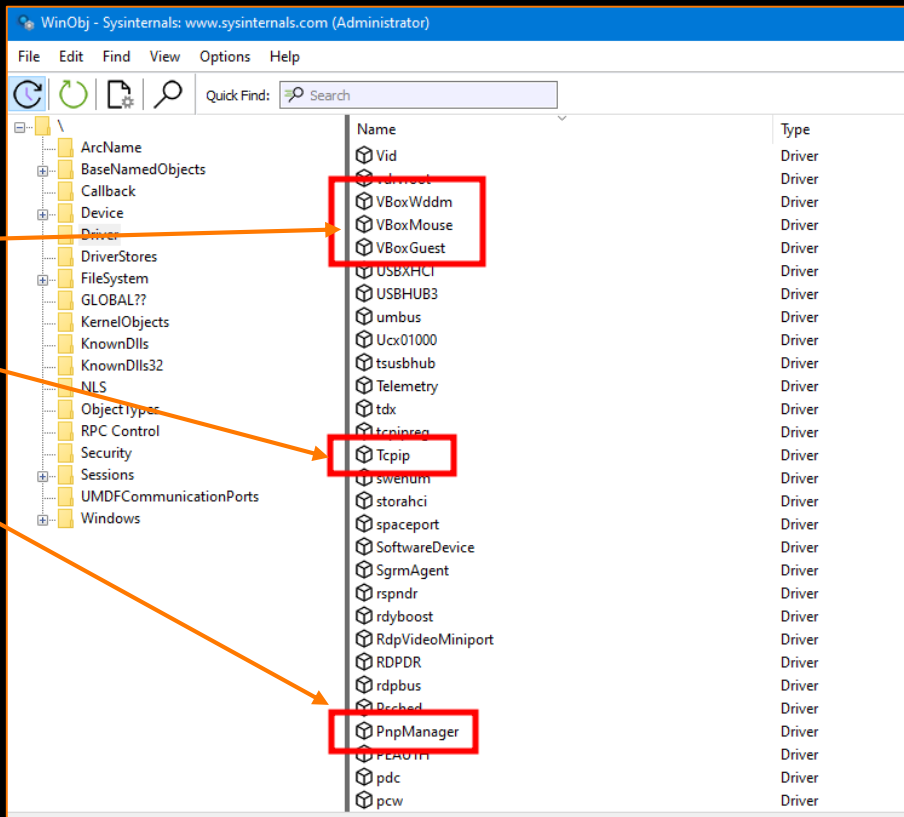
# II / Développer un driver simple

WinObj.exe de la suite SysInternals  
(<https://learn.microsoft.com/en-us/sysinternals/downloads/winobj>):

- Driver VirtualBox
- Driver de la pile TCPIP
- Driver Plug and Play

Mais aussi les drivers Bluetooth, wifi...  
Plus d'exemples ici ->

<https://github.com/microsoft/Windows-driver-samples>



## II / Développer un driver Windows

Microsoft propose le framework WDF (Windows Driver Frameworks) pour aider au développement de drivers. Ce dernier est constitué de deux sous framework qui ont chacun leurs spécificités:

	Pro's	Con's
KMDF (Kernel Mode Driver Framework)	Accès aux fonctionnalités avancées du kernel	Difficile à développer, une erreur = crash du kernel
UMDF (User Mode Driver Framework)	Simple à développer	Fonctionnalités limitées

# Code du driver Windows et démo

# III / Implémentation des kernel callbacks



# III / Implémentation des kernel callbacks

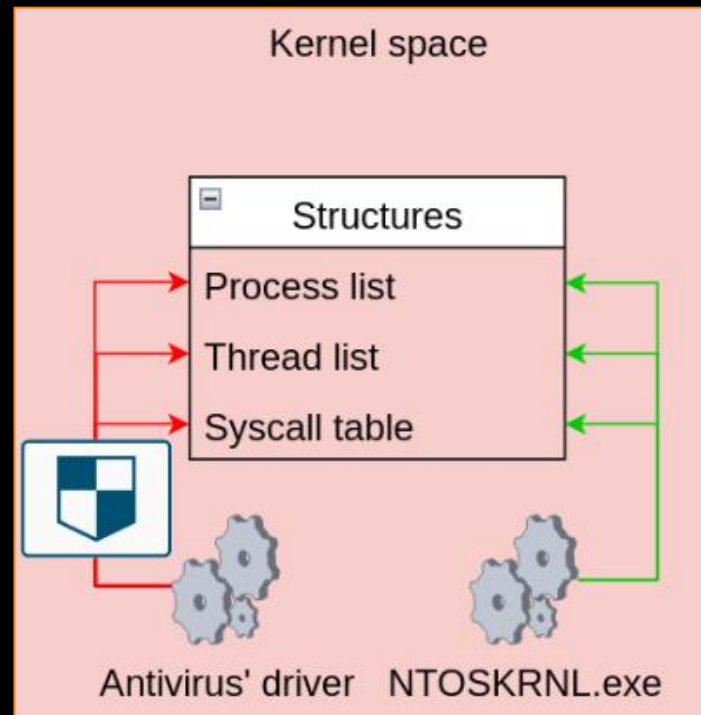
Un EDR a besoin de plusieurs informations pour détecter une menace.

Parmi ces informations il y a:

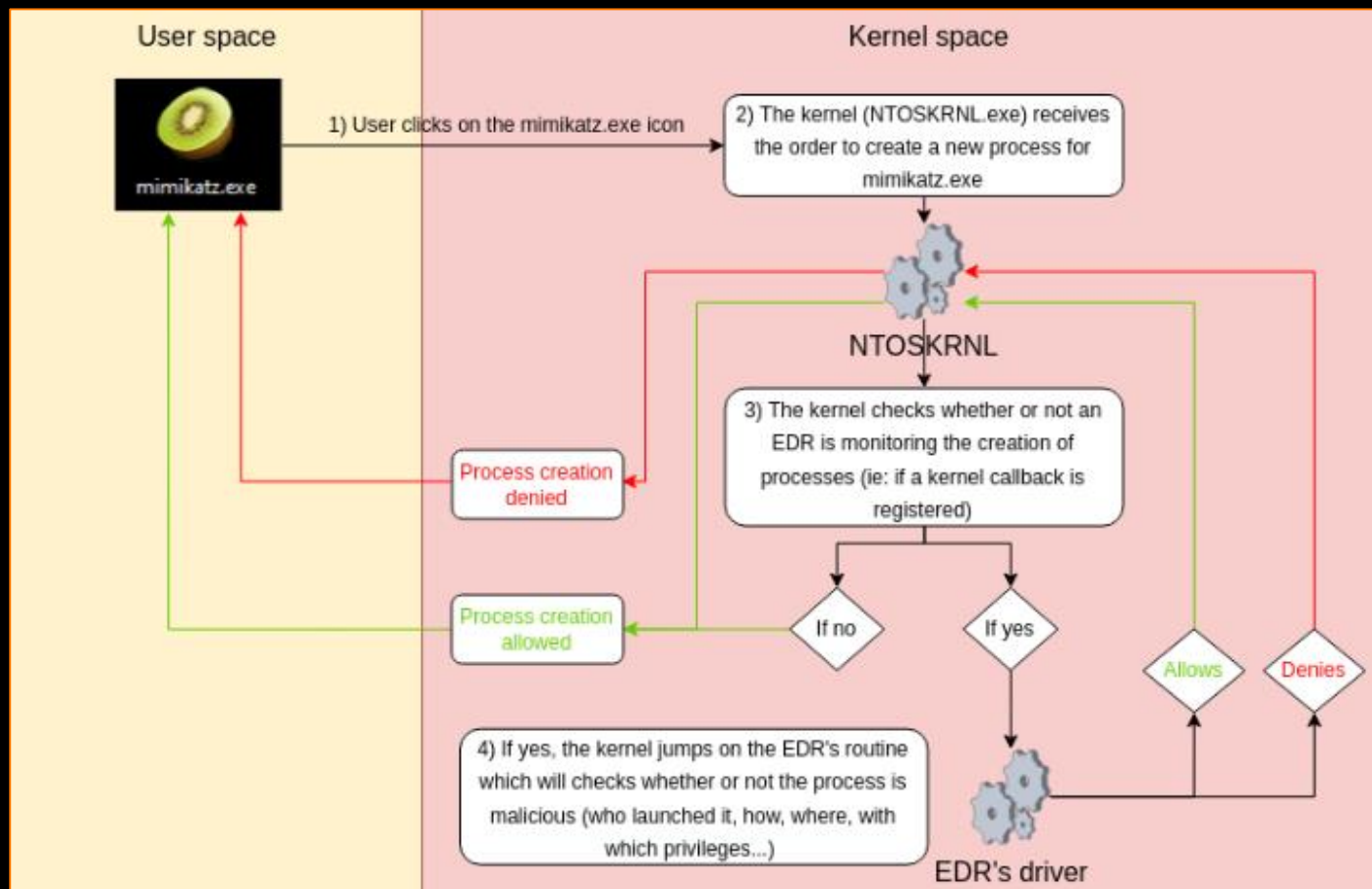
- Le nom des processus créés
- Le nom des DLL chargées et les fonctions utilisées par un binaire
- Les arguments passés à ces fonctions

Historiquement les antivirus avaient accès à ces informations en modifiant le comportement du kernel.

Mais ce n'est plus possible depuis Patch Guard



# III / Implémentation des kernel callbacks



# III / Implémentation des kernel callbacks

Pour enregistrer un kernel callback il faut utiliser une des fonctions légitimes proposées par l'OS Windows, par exemple:

- **PsSetCreateProcessNotifyRoutine**: monitore la creation de processus
- **PsSetThreadCreateNotifyRoutine**: monitore la creation de threads
- **PsSetLoadImageNotifyRoutine**: monitore le chargement de DLL's
- **ObRegisterCallbacks**: monitore l'accès aux ressources du système
- **CmRegisterCallbacks**: monitore la modification des clés de registre

## PsSetCreateProcessNotifyRoutine, fonction (ntddk.h)

Article • 24/09/2022 • 9 contributeurs

[Commentaires](#)

La routine **PsSetCreateProcessNotifyRoutine** ajoute une routine de rappel fournie par le pilote à, ou la supprime, une liste de routines à appeler chaque fois qu'un processus est créé ou supprimé.

# III / Implémentation des kernel callbacks

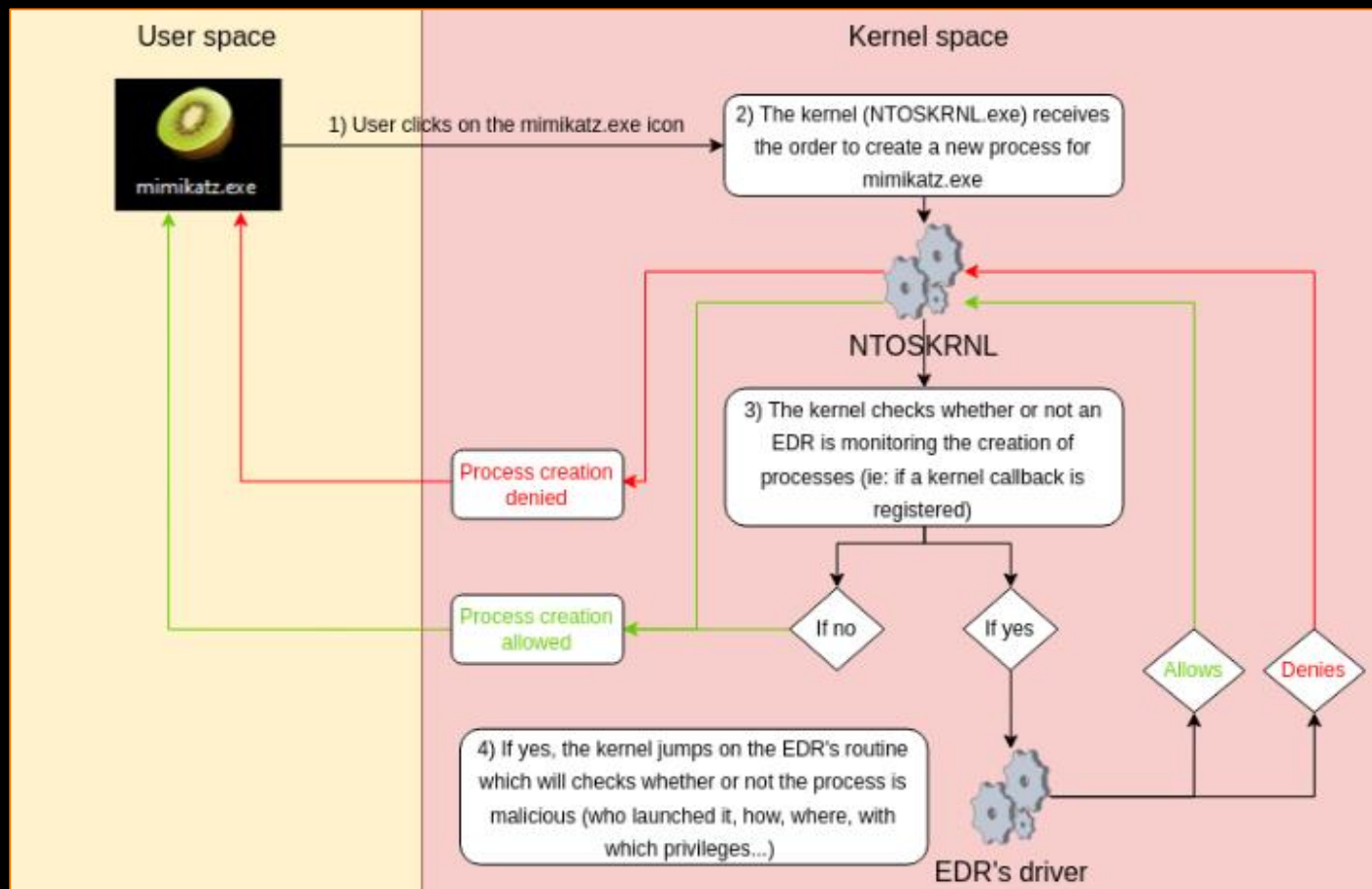
Plus précisément, un kernel callback c'est un pointeur stocké au sein du kernel dans un tableau.

Fonction	Nom de la structure	Nombre maximum de callbacks stockables
PsSetCreateProcessNotifyRoutine	PspCreateProcessNotifyRoutine	64
PsSetCreateThreadNotifyRoutine	PspCreateThreadNotifyRoutine	64
PsSetLoadImageNotifyRoutine	PspLoadImageNotifyRoutine	8
CmRegisterCallback	CmpCallBackVector	100

Contenu de la structure  
PspCreateProcessNotifyRoutine

```
1kd> dq nt!PspCreateProcessNotifyRoutine
fffff800`0c4ec120 fffffb30c`9b85018f fffffb30c`9b8ffabf
fffff800`0c4ec130 fffffb30c`9bf4d8df fffffb30c`9bf4de7f
fffff800`0c4ec140 fffffb30c`9e09179f fffffb30c`9e0912ef
fffff800`0c4ec150 fffffb30c`9e091a6f fffffb30c`a19b818f
fffff800`0c4ec160 fffffb30c`a19dc29f 00000000`00000000
fffff800`0c4ec170 00000000`00000000 00000000`00000000
fffff800`0c4ec180 00000000`00000000 00000000`00000000
fffff800`0c4ec190 00000000`00000000 00000000`00000000
```

# III / Implémentation des kernel callbacks



# Code kernel callback et démo

## **IV/ Développement des agents**

# IV / Développement des agents

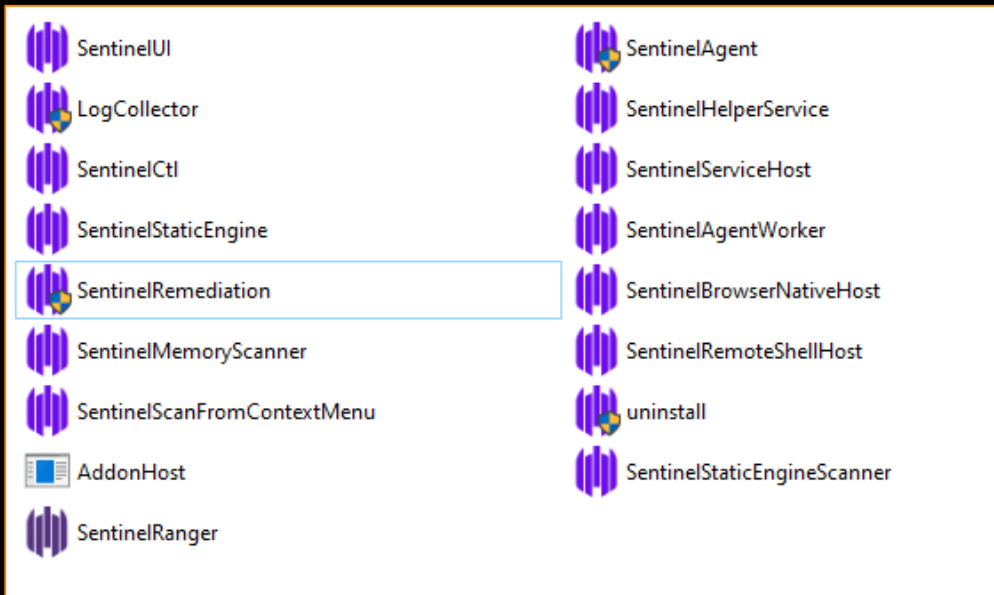
A ce stade, notre driver sait quand un nouveau processus est sur le point d'être créé.

On peut donc:

- Analyser l'exécutable en cours de création (analyse statique du binaire)
- Le modifier à la volée

Pour cela les EDR's disposent de plusieurs outils, par exemple:

- Analyse statique
- Analyse mémoire





# IV / Développement des agents

Pourquoi toute cette logique ne se fait pas directement au sein du driver ?



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

60% complete

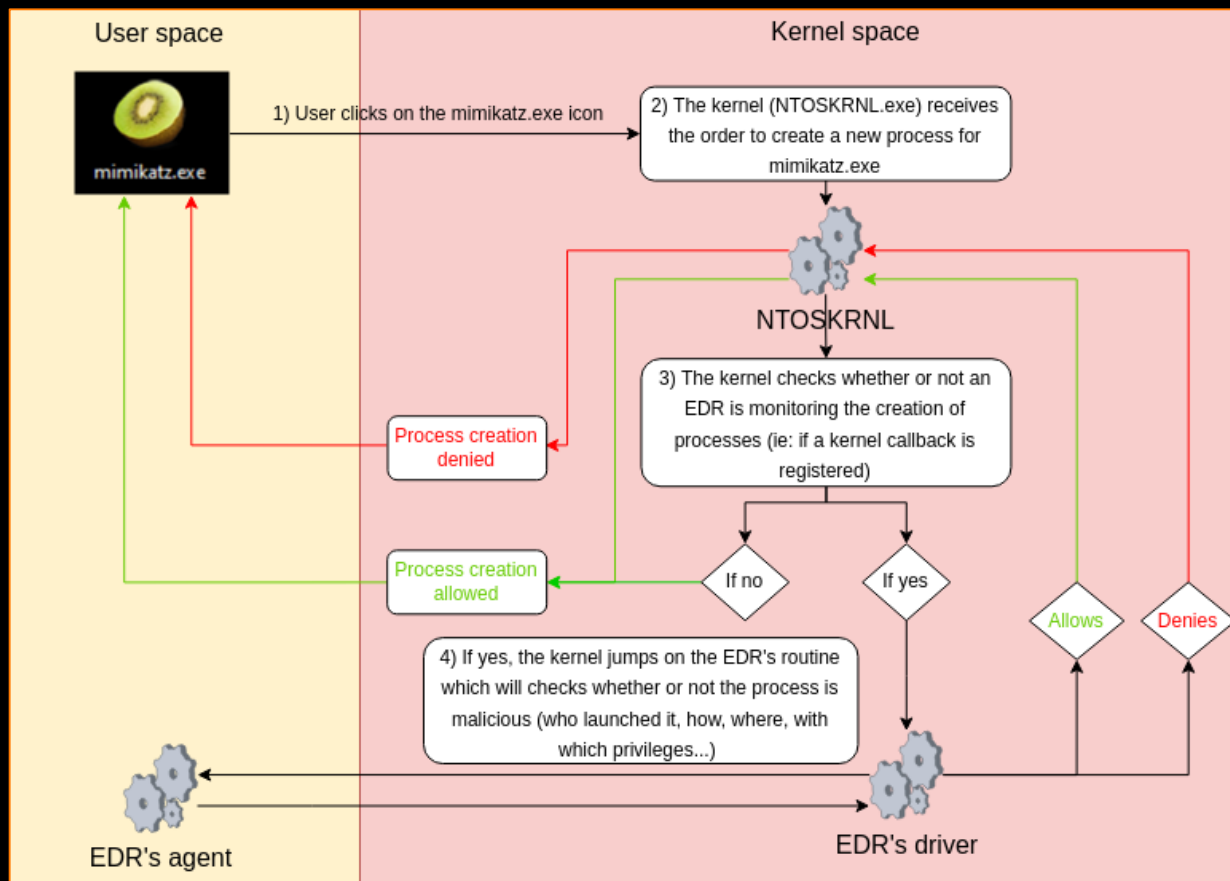


For more information about this issue and possible fixes, visit <https://www.windows.com/stopcode>

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

Stop code **KERNEL SECURITY CHECK FAILURE**

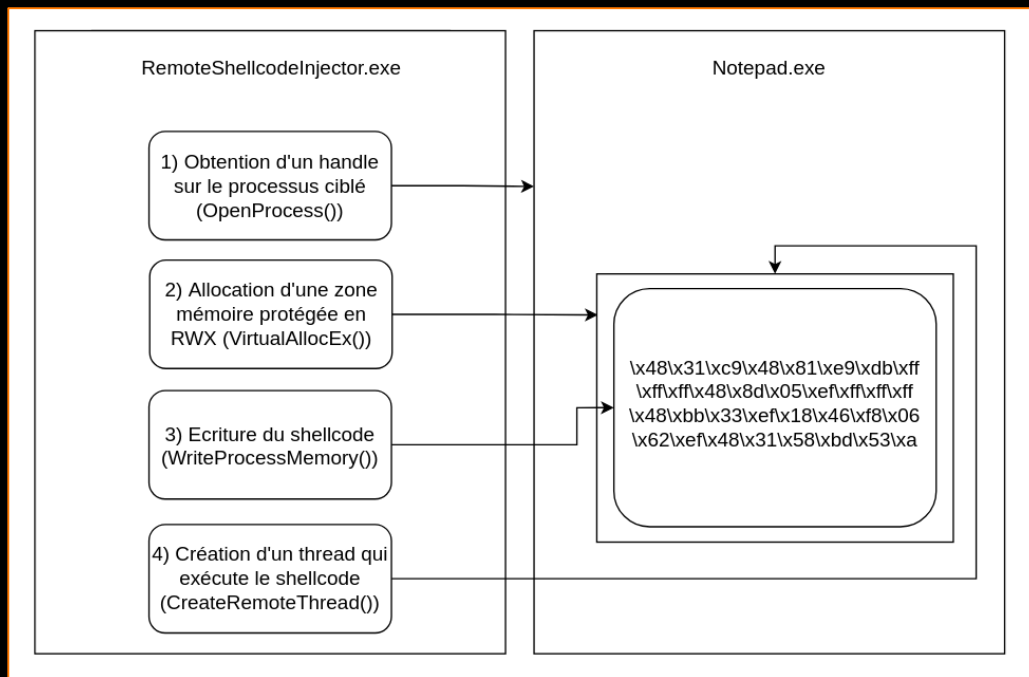
# IV / Développement des agents



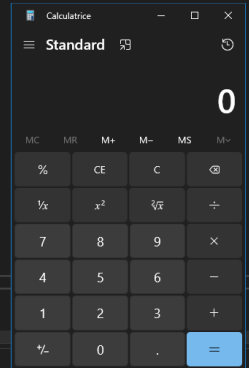
## IV / Développement des agents

Avant de développer un agent maison, il faut déterminer quel sera son utilité.

- > Empêcher l'injection à distance de shellcode.



```
printf("VirtualAllocEx\n");
// gsvf.exe -p windows\464\exec CMD=calc.exe -b %x90%ba%6d -f c
unsigned char shellcode[] =
{
    '\x48\x31\xC9\x48\x81\x99\xDB\xFF\xFF\xFF\xFF\x48\xDB\x85\xFF\xFF',
    '\xFF\xFF\x48\xDB\x33\xFF\x31\x48\x46\x86\x46\x62\x62\x31\x58',
    '\x27\x48\x24\xF8\xFF\xFF\xFF\xFF\x42\xF4\xCF\xA7\x9B\xA2\x88',
    '\xEA\x21\xEF\x33\xFF\x59\x12\x71\xB9\x65\x36\x86\x6A\x65\x74\x29',
    '\x46\x9D\x46\x53\xA7\x93\x14\x46\x46\x9D\x46\x13\xA7\x93\x46',
    '\x46\x9D\x46\x53\xA7\x93\x14\x46\x46\x9D\x46\x13\xA7\x93\x46',
    '\x3A\x3A\xF4\x24\x42\x46\xF2\x13\x46\x87\xF9\xF\x88\x82\x67\xA3',
    '\x49\x86\x47\x34\x46\x42\x64\x71\x31\x46\x47\x28\x8D\xE2\x67\x33',
    '\xEF\x31\x81\x96\x7D\x46\x81\x88\x75\x86\xC8\x16\x73\x46\xA6',
    '\xB8\xA7\x38\xF8\xF9\x66\xD1\xB9\x7B\x10\xD1\x90\x77\x32\xEA',
    '\xA7\x32\x31\x59\x55\x77\x31\x46\x53\x2F\x9F\xAE\xD9\x87\xF4',
    '\x63\xE2\x46\x8D\x67\xB4\x46\x53\x2F\x9F\xAE\xD9\x87\xF4',
    '\xD6\x3A\x46\xB8\xF3\xCF\x86\xF4\x46\x8D\x46\x84\xAE\xDB\x83\x59',
    '\x73\x46\x46\x46\x32\xF3\x59\xC2\xDF\xC8\x24\xAE\xE3\xAE',
    '\x97\xA9\x58\x36\xB5\x72\xB7\xB9\x31\xF9\x5C\x2A\xC6\x46\xCF',
    '\x97\xA9\x58\x36\xB5\x72\xB7\xB9\x31\xF9\x5C\x2A\xC6\x46\xCF',
    '\x32\x46\x46\xB9\x36\x53\x46\x56\x77\x93\x43\x6F\x46',
    '\x32\x46\x46\xB9\x36\x53\x46\x56\x77\x93\x43\x6F\x46',
    '\xD6\x65\xAE\xA2\x46\xDB\xFF\xA1\x60\x46\xA7\x9B\x82\xD3\xA',
    '\x64\x93\x39\x66\x31\xAE\x8D\x83\xD9\xA3\xD9\xD7\x46\xF2\xF8',
    '\x5F\x23\x61\x96\xF9\x10\xC2\x85\xC2\x5A\x35\x86\x5D\xB8\x77',
    '\x33\x61\x96\xF9\x10\xC2\x85\xC2\x5A\x35\x86\x5D\xB8\x77'
}
```



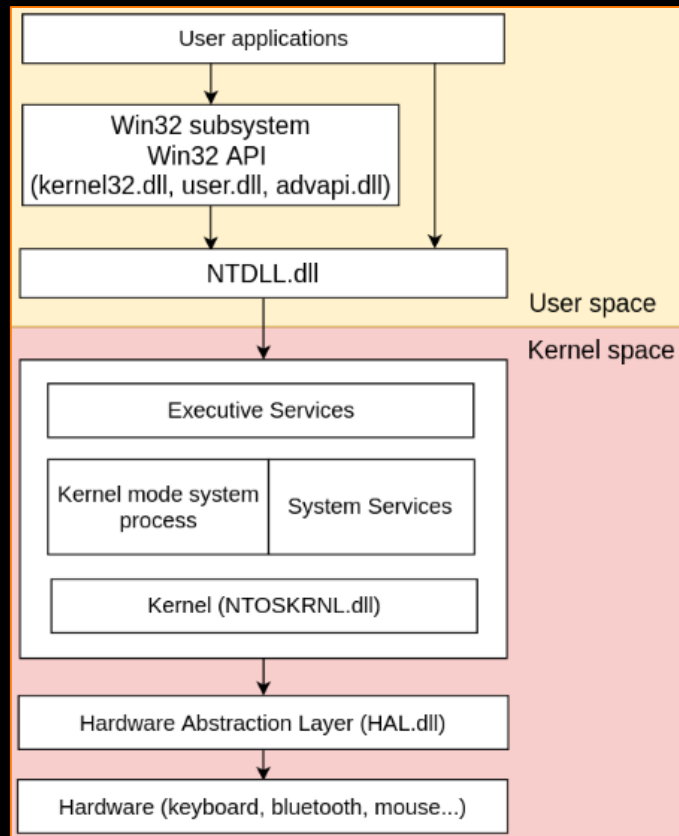
# Code l'agent StaticAnalyzer

# IV / Développement des agents

Le système d'exploitation Windows est composé de deux couches:

- User space
- Kernel space

Les applications Windows reposent la WinAPI, un ensemble de DLL's qui exposent des fonctions permettant d'ouvrir des fichiers, créer des connexions etc. La WinAPI repose elle-même, en partie, sur la NTDLL.dll.



# IV / Développement des agents

Prototype de la fonction VirtualAllocEx (kernel32.dll)

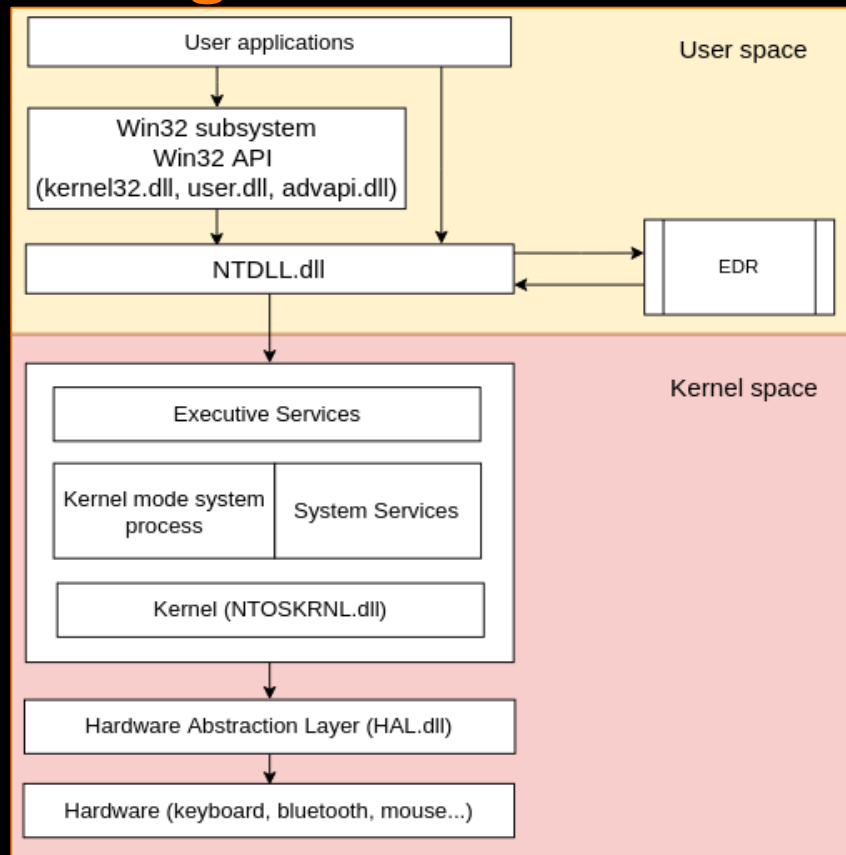
```
LPVOID VirtualAllocEx(  
    [in] HANDLE hProcess,  
    [in, optional] LPVOID lpAddress,  
    [in] SIZE_T dwSize,  
    [in] DWORD flAllocationType,  
    [in] DWORD flProtect  
);
```

Prototype de la fonction NtAllocateVirtualMemory (NTDLL.dll)

```
__kernel_entry NTSYSCALLAPI NTSTATUS NtAllocateVirtualMemory(  
    [in] HANDLE ProcessHandle,  
    [in, out] PVOID *BaseAddress,  
    [in] ULONG_PTR ZeroBits,  
    [in, out] PSIZE_T RegionSize,  
    [in] ULONG AllocationType,  
    [in] ULONG Protect  
);
```

# IV / Développement des agents

Depuis Patch Guard on ne peut plus modifier les structures contenues au sein du kernel. Par contre on peut modifier la NTDLL.dll qui est le dernier bloc avant l'accès au kernel



# IV / Développement des agents

Code assembleur légitime de la fonction NtAllocateVirtualMemory:

```
ntdll!NtAllocateVirtualMemory:
00007ffa`a074d350 4c8bd1      mov     r10, rcx
00007ffa`a074d353 b818000000  mov     eax, 18h
00007ffa`a074d358 f604250803fe7f01 test    byte ptr [7FFE0308h], 1
00007ffa`a074d360 7503        jne     ntdll!NtAllocateVirtualMemory+0x15
00007ffa`a074d362 0f05        syscall
00007ffa`a074d364 c3          ret
00007ffa`a074d365 cd2e        int     2Eh
00007ffa`a074d367 c3          ret
00007ffa`a074d368 0f1f840000000000 nop     dword ptr [rax+rax]
```

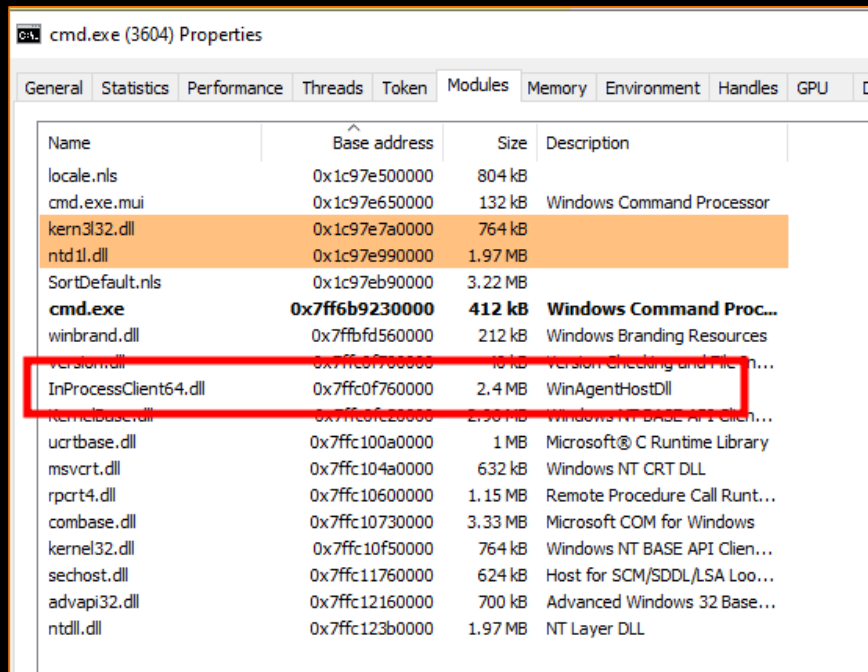
Code assembleur modifié par la DLL de l'EDR:

```
ntdll!NtAllocateVirtualMemory:
00007ffa`a074d350 e9813cf5ff  jmp     00007FFAA06A0FD6
00007ffa`a074d355 0000        add     byte ptr [rax], al
00007ffa`a074d357 00f6        add     dh, dh
00007ffa`a074d359 0425        add     al, 25h
00007ffa`a074d35b 0803        or      byte ptr [rbx], al
00007ffa`a074d35d fe          ???
00007ffa`a074d35e 7f01        jg      ntdll!NtAllocateVirtualMemory+0x11
00007ffa`a074d360 7503        jne     ntdll!NtAllocateVirtualMemory+0x15
00007ffa`a074d362 0f05        syscall
00007ffa`a074d364 c3          ret
00007ffa`a074d365 cd2e        int     2Eh
00007ffa`a074d367 c3          ret
00007ffa`a074d368 0f1f840000000000 nop     dword ptr [rax+rax]
```



# IV / Développement des agents

L'agent de l'EDR injecte une de ses DLL's au sein du programme. Cette DLL agit comme un proxy et permet à l'EDR de monitorer dynamiquement les fonctions appelées par un binaire ainsi que les paramètres qui lui sont envoyés.



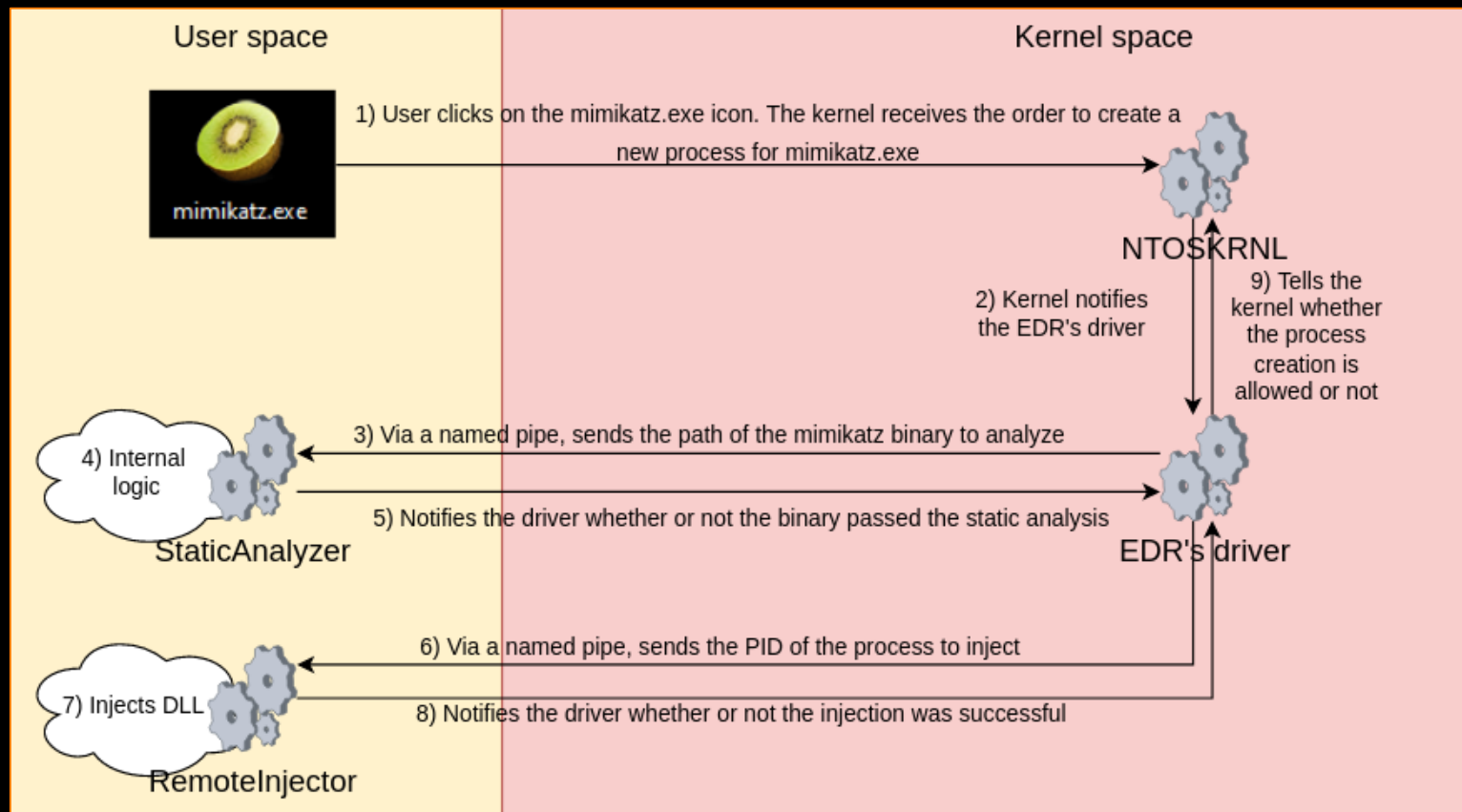
cmd.exe (3604) Properties

General Statistics Performance Threads Token Modules Memory Environment Handles GPU D

Name	Base address	Size	Description
locale.nls	0x1c97e500000	804 kB	
cmd.exe.mui	0x1c97e650000	132 kB	Windows Command Processor
kern32.dll	0x1c97e7a0000	764 kB	
ntdll.dll	0x1c97e990000	1.97 MB	
SortDefault.nls	0x1c97eb90000	3.22 MB	
<b>cmd.exe</b>	<b>0x7ff6b9230000</b>	<b>412 kB</b>	<b>Windows Command Proc...</b>
winbrand.dll	0x7ffbfd560000	212 kB	Windows Branding Resources
version.dll	0x7ffc0f700000	10 kB	Version Checking and File...
<b>InProcessClient64.dll</b>	<b>0x7ffc0f760000</b>	<b>2.4 MB</b>	<b>WinAgentHostDll</b>
kernelbase.dll	0x7ffc0f200000	2.56 MB	Windows NT BASE API Clie...
ucrtbase.dll	0x7ffc100a0000	1 MB	Microsoft® C Runtime Library
msvcrt.dll	0x7ffc104a0000	632 kB	Windows NT CRT DLL
rport4.dll	0x7ffc10600000	1.15 MB	Remote Procedure Call Runt...
combase.dll	0x7ffc10730000	3.33 MB	Microsoft COM for Windows
kernel32.dll	0x7ffc10f50000	764 kB	Windows NT BASE API Clie...
sechost.dll	0x7ffc11760000	624 kB	Host for SCM/SDDL/LSA Loo...
advapi32.dll	0x7ffc12160000	700 kB	Advanced Windows 32 Base...
ntdll.dll	0x7ffc123b0000	1.97 MB	NT Layer DLL

# Code l'agent RemoteInjector et DLL

# IV / Développement des agents



# Démo de l'EDR finalisé

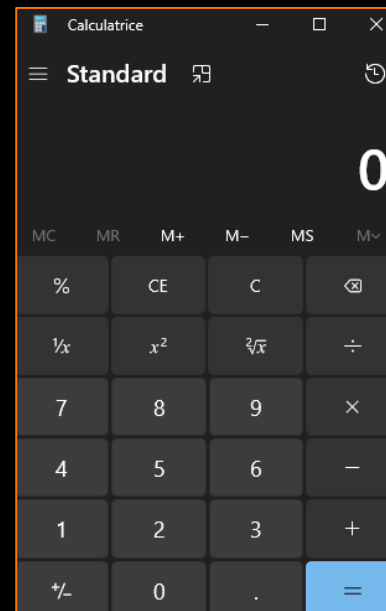
**V/ A vous de jouer**

# V / A vous de jouer

L'EDR va être publié sur Github (ASAP), est volontairement vulnérable (bugs logiques) et il est possible de le bypass de plusieurs manières.

Le but du jeu: injecter un shellcode au sein du processus notepad:

```
COMMANDO Tue 06/27/2023 13:33:30.03
Z:\windev\MyDumbEDR\x64\Debug+>ShellcodeInject.exe
Launching remote shellcode injection
SeDebugPrivilege owned
SeDebugPrivilege enabled.
Injecting to PID: 2988
VirtualAllocEx
WriteProcessMemory
CreateRemoteThread
Congratz dude! The flag is MyDumbEDR{H4ckTH3W0rld}
```



# Conclusion

# Des questions ?

Mail: [aurelien.chalot@orangecyberdefense.com](mailto:aurelien.chalot@orangecyberdefense.com)

Mon blog: <https://blog.whiteflag.io>

Twitter: [https://twitter.com/Defte\\_](https://twitter.com/Defte_)

Linkedin: <https://www.linkedin.com/in/aurelienchalotinc/>

<https://cyberdefense.orange.com>

