在笔者上一篇文章 《驱动开发：内核注册并监控对象回调》 介绍了如何运用 ObRegisterCallbacks 注册 进

程与线程 回调，并通过该回调实现了 拦截 指定进行运行的效果，本章 LyShark 将带大家继续探索一个新

的回调注册函数， PsSetLoadImageNotifyRoutine 常用于注册 LoadImage 映像监视，当有模块被系统加载时则可以第一时间获取到加载模块信息，需要注意的是该回调函数内无法进行拦截，如需要拦截则 需写入返回指令这部分内容将在下一章进行讲解，本章将主要实现对模块的监视功能。

# 监视模块加载与卸载需要费别使用两个函数，这两个函数的参数传递都是自己的回调地址。

PsSetLoadImageNotifyRoutine 设置回调

PsRemoveLoadImageNotifyRoutine 移除回调

此处 MyLySharkLoadImageNotifyRoutine 回调地址必须有三个参数传递组成，其中 FullImageName

代表完整路径， ModuleStyle 代表模块类型，一般来说 ModuleStyle=0 表示加载SYS驱动，如果

ModuleStyle=1 则表示加载的是 DLL ，最后一个参数 ImageInfo 则是映像的详细参数结构体。

VOID MyLySharkLoadImageNotifyRoutine(PUNICODE\_STRING FullImageName, HANDLE ModuleStyle, PIMAGE\_INFO ImageInfo)

那么如何实现监视映像加载呢，来看如下完整代码片段，首先 PsSetLoadImageNotifyRoutine 注册回调，当有模块被加载则自动执行 MyLySharkLoadImageNotifyRoutine 回调函数，其内部首先判断

ModuleStyle 得出是什么类型的模块，然后再通过 GetDriverEntryByImageBase 拿到当前进程详细参

# 数并打印输出。



// 署名权

// right to sign one's name on a piece of work

// PowerBy: LyShark

// Email: [me@lyshark.com](mailto:me@lyshark.com) #include <ntddk.h> #include <ntimage.h>

// 未导出函数声明

PUCHAR PsGetProcessImageFileName(PEPROCESS pEProcess);

// 获取到镜像装载基地址

PVOID GetDriverEntryByImageBase(PVOID ImageBase)

{

PIMAGE\_DOS\_HEADER pDOSHeader; PIMAGE\_NT\_HEADERS64 pNTHeader;

PVOID pEntryPoint;

pDOSHeader = (PIMAGE\_DOS\_HEADER)ImageBase;

pNTHeader = (PIMAGE\_NT\_HEADERS64)((ULONG64)ImageBase + pDOSHeader-

>e\_lfanew);

pEntryPoint = (PVOID)((ULONG64)ImageBase + pNTHeader-

>OptionalHeader.AddressOfEntryPoint); return pEntryPoint;

}

// 获取当前进程名

UCHAR GetCurrentProcessName()

{

PEPROCESS pEProcess = PsGetCurrentProcess(); if (NULL != pEProcess)

{



UCHAR lpszProcessName = PsGetProcessImageFileName(pEProcess); if (NULL != lpszProcessName)

{

return lpszProcessName;

}

}

return NULL;

}

// 设置自己的回调函数

VOID MyLySharkLoadImageNotifyRoutine(PUNICODE\_STRING FullImageName, HANDLE ModuleStyle, PIMAGE\_INFO ImageInfo)

{

PVOID pDrvEntry;

// MmIsAddress 验证地址可用性

if (FullImageName != NULL && MmIsAddressValid(FullImageName))

{

// ModuleStyle为零表示加载sys if (ModuleStyle == 0)

{

// 得到装载主进程名

UCHAR load\_name = GetCurrentProcessName();

pDrvEntry = GetDriverEntryByImageBase(ImageInfo->ImageBase); DbgPrint("[LyShark SYS加载] 模块名称:%wZ --> 装载基址:%p --> 镜像长度: %d

--> 装载主进程: %s \n", FullImageName, pDrvEntry, ImageInfo->ImageSize,

load\_name);

}

// ModuleStyle非零表示加载DLL else

{

// 得到装载主进程名

UCHAR load\_name = GetCurrentProcessName();

pDrvEntry = GetDriverEntryByImageBase(ImageInfo->ImageBase); DbgPrint("[LyShark DLL加载] 模块名称:%wZ --> 装载基址:%p --> 镜像长度: %d

--> 装载主进程: %s \n", FullImageName, pDrvEntry, ImageInfo->ImageSize,

load\_name);

}

}

}

VOID UnDriver(PDRIVER\_OBJECT driver)

{

PsRemoveLoadImageNotifyRoutine((PLOAD\_IMAGE\_NOTIFY\_ROUTINE)MyLySharkLoadImageNot ifyRoutine);

DbgPrint("[LyShark.com] 驱动卸载完成...");

}

NTSTATUS DriverEntry(IN PDRIVER\_OBJECT Driver, PUNICODE\_STRING RegistryPath)

{

DbgPrint("hello lyshark.com \n");

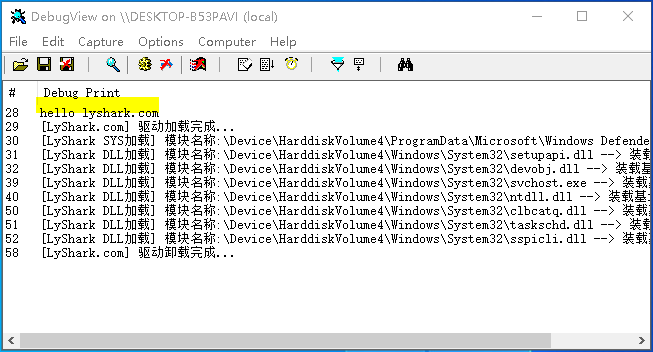
PsSetLoadImageNotifyRoutine((PLOAD\_IMAGE\_NOTIFY\_ROUTINE)MyLySharkLoadImageNotify Routine);

DbgPrint("[LyShark.com] 驱动加载完成...");

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

}

# 运行这个驱动程序，则会输出被加载的驱动详细参数。



作者： 王瑞 (LyShark)

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

版权声明：本博客文章与代码均为学习时整理的笔记，文章 [均为原创] 作品，转载文章请遵守

《中华人民共和国著作权法》相关法律规定或遵守《署名CC BY-ND 4.0国际》规范，合理合规携带原创出处转载，如果不携带文章出处，并恶意转载多篇原创文章被本人发现，本人保留起诉权！