# 微软在 x64 系统中推出了 DSE 保护机制，DSE全称 (Driver Signature Enforcement) ，该保护机制的核心就是任何驱动程序或者是第三方驱动如果想要在正常模式下被加载则必须要经过微软的认证，当 驱动程序被加载到内存时会验证签名的正确性，如果签名不正常则系统会拒绝运行驱动，这种机制也被 称为驱动强制签名，该机制的作用是保护系统免受恶意软件的破坏，是提高系统安全性的一种手段。

该验证机制即便是在调试模式也需要强制签名，对于一名 驱动开发者 来说是很麻烦的一件事情，而签名的验证则是在加载时验证驱动入口 \_KLDR\_DATA\_TABLE\_ENTRY 里面的 Flags 标志，如果此标志被pLdrData->Flags | 0x20 置位，则在调试模式下就不会在验证签名了，省去了重复签名的麻烦。

# 代码的实现非常容易，如下所示:

// 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 <ntifs.h>

// 绕过签名检查

BOOLEAN BypassCheckSign(PDRIVER\_OBJECT pDriverObject)

{

#ifdef \_WIN64

typedef struct \_KLDR\_DATA\_TABLE\_ENTRY

{

LIST\_ENTRY listEntry; ULONG64 Undefined1; ULONG64 Undefined2; ULONG64 Undefined3; ULONG64 NonPagedDebugInfo; ULONG64 DllBase;

ULONG64 EntryPoint; ULONG SizeOfImage; UNICODE\_STRING path; UNICODE\_STRING name;

ULONG Flags; USHORT LoadCount;

USHORT Undefined5;

ULONG64 Undefined6; ULONG CheckSum; ULONG padding1; ULONG TimeDateStamp; ULONG padding2;

} KLDR\_DATA\_TABLE\_ENTRY, PKLDR\_DATA\_TABLE\_ENTRY;

#else

typedef struct \_KLDR\_DATA\_TABLE\_ENTRY

{

LIST\_ENTRY listEntry; ULONG unknown1; ULONG unknown2; ULONG unknown3; ULONG unknown4; ULONG unknown5; ULONG unknown6;



ULONG unknown7; UNICODE\_STRING path; UNICODE\_STRING name;

ULONG Flags;

} KLDR\_DATA\_TABLE\_ENTRY, PKLDR\_DATA\_TABLE\_ENTRY;

#endif

PKLDR\_DATA\_TABLE\_ENTRY pLdrData = (PKLDR\_DATA\_TABLE\_ENTRY)pDriverObject-

>DriverSection;

pLdrData->Flags = pLdrData->Flags | 0x20;

return TRUE;

}

VOID UnDriver(PDRIVER\_OBJECT driver)

{

}

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

{

NTSTATUS status;

// 绕过签名检查

// LINKER\_FLAGS=/INTEGRITYCHECK

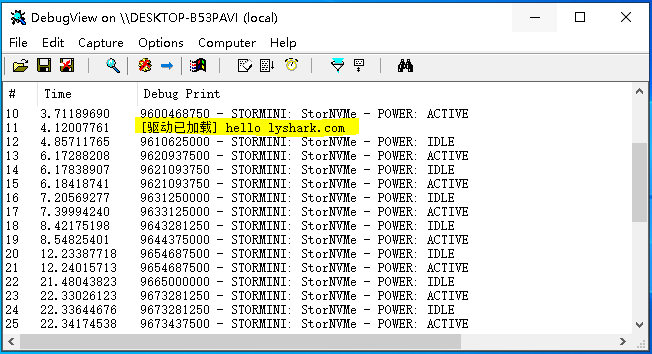
BypassCheckSign(Driver);

DbgPrint("[驱动已加载] hello lyshark.com \n"); Driver->DriverUnload = UnDriver;

return STATUS\_SUCCESS;

}

# 将程序拖入到虚拟机，直接运行即可加载，无需再继续签名：



当然这种方式只能在测试模式下使用，在正常模式也是无效的，只是为了方便测试驱动。