# 内核中读写内存的方式有很多，典型的读写方式有CR3读写，MDL读写，以及今天要给大家分享的内存 拷贝实现读写，拷贝读写的核心是使用 MmCopyVirtualMemory 这个内核API函数实现，通过调用该函数即可很容易的实现内存的拷贝读写。

封装 KeReadProcessMemory() 内存读取。

#include <ntifs.h> #include <windef.h> #include <stdlib.h>



NTKERNELAPI NTSTATUS PsLookupProcessByProcessId(HANDLE ProcessId, PEPROCESS Process);

NTKERNELAPI CHAR PsGetProcessImageFileName(PEPROCESS Process);

NTSTATUS NTAPI MmCopyVirtualMemory(PEPROCESS SourceProcess, PVOID SourceAddress, PEPROCESS TargetProcess, PVOID TargetAddress, SIZE\_T BufferSize, KPROCESSOR\_MODE PreviousMode, PSIZE\_T ReturnSize);

// 定义全局EProcess结构

PEPROCESS Global\_Peprocess = NULL;

// 普通Ke内存读取

NTSTATUS KeReadProcessMemory(PVOID SourceAddress, PVOID TargetAddress, SIZE\_T Size)

{

try

{

PEPROCESS TargetProcess = PsGetCurrentProcess(); SIZE\_T Result;

if (NT\_SUCCESS(MmCopyVirtualMemory(Global\_Peprocess, SourceAddress, TargetProcess, TargetAddress, Size, KernelMode, &Result)))

return STATUS\_SUCCESS;

else

return STATUS\_ACCESS\_DENIED;

}

except (EXCEPTION\_EXECUTE\_HANDLER)

{

return STATUS\_ACCESS\_DENIED;

}

return STATUS\_ACCESS\_DENIED;

}

VOID UnDriver(PDRIVER\_OBJECT driver)

{

DbgPrint("Uninstall Driver Is OK \n");

}

// By:lyshark.cnblogs.com

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

{

DbgPrint("hello lyshark \n");

// 根据PID打开进程

DWORD PID = 6672;

NTSTATUS nt = PsLookupProcessByProcessId((HANDLE)PID, &Global\_Peprocess);

DWORD ref\_value = 0;

// 将地址处读取4字节到ref\_value中

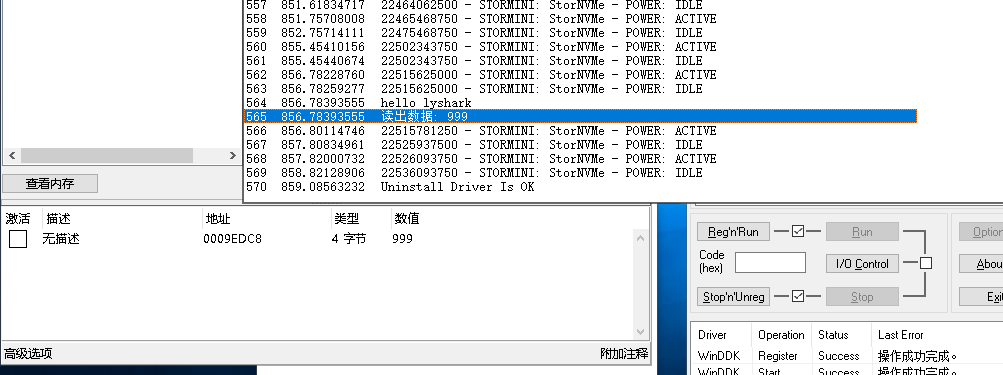
NTSTATUS read\_nt = KeReadProcessMemory((PVOID)0x0009EDC8, &ref\_value, 4);

DbgPrint("读出数据: %d \n", ref\_value);

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

}

# 读取效果如下：



封装 KeWriteProcessMemory() 内存读取。

#include <ntifs.h> #include <windef.h> #include <stdlib.h>



NTKERNELAPI NTSTATUS PsLookupProcessByProcessId(HANDLE ProcessId, PEPROCESS Process);

NTKERNELAPI CHAR PsGetProcessImageFileName(PEPROCESS Process);

NTSTATUS NTAPI MmCopyVirtualMemory(PEPROCESS SourceProcess, PVOID SourceAddress, PEPROCESS TargetProcess, PVOID TargetAddress, SIZE\_T BufferSize, KPROCESSOR\_MODE PreviousMode, PSIZE\_T ReturnSize);

// 定义全局EProcess结构

PEPROCESS Global\_Peprocess = NULL;

// 普通Ke内存写入

NTSTATUS KeWriteProcessMemory(PVOID SourceAddress, PVOID TargetAddress, SIZE\_T Size)

{

PEPROCESS SourceProcess = PsGetCurrentProcess(); PEPROCESS TargetProcess = Global\_Peprocess; SIZE\_T Result;

if (NT\_SUCCESS(MmCopyVirtualMemory(SourceProcess, SourceAddress, TargetProcess, TargetAddress, Size, KernelMode, &Result)))

return STATUS\_SUCCESS;

else

return STATUS\_ACCESS\_DENIED;

}

VOID UnDriver(PDRIVER\_OBJECT driver)

{

DbgPrint("Uninstall Driver Is OK \n");

}

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

{

DbgPrint("hello lyshark \n");

// 根据PID打开进程

DWORD PID = 6672;

NTSTATUS nt = PsLookupProcessByProcessId((HANDLE)PID, &Global\_Peprocess);

DWORD ref\_value = 10;

// 将地址处写出4字节

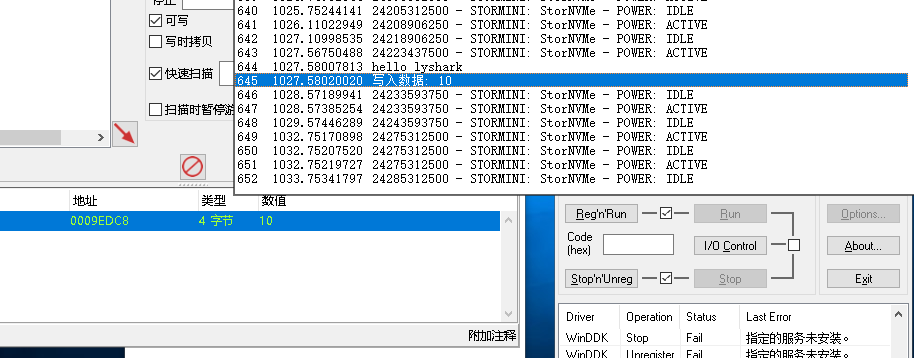
NTSTATUS read\_nt = KeWriteProcessMemory((PVOID)0x0009EDC8, &ref\_value, 4);

DbgPrint("写入数据: %d \n", ref\_value);

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

}

# 写出内存效果：



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