**The interception of x64 kernel HOOK technology(x64 커널 HOOK 기술 가로채기)**

**Intercept process(프로세스 가로채기), intercept thread(스레드 가로채기), intercept module(모듈 가로채기)**

1. HOOK 기술을 설명하는 이유

예를 들어 32bit 시스템에서 SSDT 테이블을 HOOK 하려면 CR0의 메모리 보호 속성에 대해 직접 이야기하고 제거하십시오. 테이블의 주소 수정에 대해 직접 이야기하십시오.

그러나 64비트 시스템에서는 이것이 불가능합니다.

첫 번째 이유는 SSDT 테이블이 암호화되어 있기 때문입니다.

두 번째는 해독하더라도 SSDT 테이블입니다. 그러면 API 주소가 멀어집니다. SSDT 테이블은 4G 이외의 주소에 맞을 수 없습니다. 따라서 배치할 수 없습니다.

그리고 지금. 바이러스 백신 소프트웨어가 커널에서 몇 가지 트릭을 수행해야 한다고 상상해 보십시오. HOOK을 할 수 없다면 어떻게 발전해야 할까요?

따라서 win7(64비트) 시스템에서 Microsoft는 이를 실행하는 것을 두려워합니다. 직접 말씀하세오. 당신은 API를 HOOK하고 싶어, 나는 당신을 위해 그것을 할 것 입니다. 당신은 그것을 할 필요가 없습니다. 그리고 내 것은 안정적입니다. 블루 스크린이 없습니다.

따라서 x64에서 HOOK은 매우 간단합니다.

예를 들어 xxx 바이러스 백신 소프트웨어는 프로세스 생성을 모니터링해야 합니다. 그러자 마이크로소프트가 말했다. API를 호출하고 콜백을 제공합니다. 생성되면 알려드리겠습니다. 반환 값도 귀하의 것 입니다.

이런 식으로 API를 사용할 수 있습니다. Thread, Modules은 모두 이렇습니다.

1. 프로세스 모니터링 및 차단 HOOK

API: PsSetCreateProcessNotifyRoutineEx

함수 프로로타입:

NTSTATUS PsSetCreateProcessNotifyRoutineEx(

PCREATE\_PROCESS\_NOTIFY\_ROUTINE\_EX NotifyRoutine,

BOOLEAN Remove

);

이 함수의 목적은 프로세스가 생성될 때 알려주는 것 입니다.

매개변수 콜백 함수 프로토타입

VOID CreateProcessNotifyEx(

\_Inout\_  PEPROCESS              Process,

\_In\_     HANDLE                 ProcessId,

\_In\_opt\_ PPS\_CREATE\_NOTIFY\_INFO CreateInfo

);

Process extra information 구조체

typedef struct \_PS\_CREATE\_NOTIFY\_INFO {

SIZE\_T Size;

union {

ULONG Flags;

struct {

ULONG FileOpenNameAvailable : 1;

ULONG IsSubsystemProcess : 1;

ULONG Reserved : 30;

};

};

HANDLE ParentProcessId;

CLIENT\_ID CreatingThreadId;

struct \_FILE\_OBJECT \*FileObject;

PCUNICODE\_STRING ImageFileName;

PCUNICODE\_STRING CommandLine;

NTSTATUS CreationStatus;

} PS\_CREATE\_NOTIFY\_INFO, \*PPS\_CREATE\_NOTIFY\_INFO;

반환 상태에서 작업하는 경우. 그런 다음 값을 실패로 반환하면 프로세스가 생성되지 않습니다. 그것은 또한 우리가 프로세스 생성을 방지한다는 것을 깨닫습니다.

커널 코드의 실제 작동으로 인해 계산기가 생성되지 않습니다.

**Third, thread monitoring and interception HOOK(셋째, 스레드 모니터링 및 가로채기 HOOK)**

스레드 가로채기는 프로세스 가로채기와 비슷하지만 64비트 미만에서는 Microsoft에서 제공하는 HOOK 기능이 강력하지 않습니다.

Thread monitoring API

Monitroing: PsSetCreateThreadNotifyRoutine

Cancel monitoring: PsRemoveCreateThreadNotifyRoutine

Montitoring prototype:

NTSTATUS PsSetCreateThreadNotifyRoutine(

PCREATE\_THREAD\_NOTIFY\_ROUTINE NotifyRoutine

);

Uninstall function prototype:

NTSTATUS PsRemoveCreateThreadNotifyRoutine(

PCREATE\_THREAD\_NOTIFY\_ROUTINE NotifyRoutine

);

핵심은 배열에 함수 주소를 등록하는 것 입니다.

이 두 함수는 볼 수만 있습니다. 그러나 그들은 작업할 수 없습니다. 그러나 여러 번 확인하는 것으로 충분합니다.

Thread callback prototype:

void PcreateThreadNotifyRoutine(

HANDLE ProcessId,

HANDLE ThreadId,

BOOLEAN Create

)

보시다시피 스레드 HOOK 이후에는 프로세스 ID와 스레드 ID만 있을 뿐 정보가 거의 없습니다. Create 플래그도 있습니다. 아무것도 남아 있지 않습니다. 이제 HOOK으로 이동합니다.

HOOK ideas:

1. EPROCESS object를 통해 process ID 찾기
2. ETHREAD object를 통해 thread ID 찾기
3. EPROCESS object를 통해 process path(프로세스 경로) 얻기
4. process path(프로세스 경로)를 통해 corresponding process name(해당 프로세스 이름) 찾기
5. process name(프로세스 이름)이 동일한지 확인

The same(동일): ETHREAD + offset을 통해 thread callback function(스레드 콜백 함수)의 주소를 찾습니다. 그리고 callback address

are not the same(동일하지 않음): 동일하지 않고 종료합니다.

위의 HOOK idea에서 다섯 번째는 그다지 중요하지 않습니다. ETHREAD + offset. 그럼 어떤 멤버가 검색되나요?

그런 다음 ETHREAD에서 어떤 멤버가 중요한지 알아야 합니다.

WinDbg에서 EPROCESS 구조체 보는 명령어:

dt nt!\_EPROCESS

||0:kd> dt nt!\_EPROCESS

+0x000 Pcb : \_KPROCESS

+0x0a0 ProcessLock : \_EX\_PUSH\_LOCK

+0x0a8 CreateTime : \_LARGE\_INTEGER

+0x0b0 RundownProtect : \_EX\_RUNDOWN\_REF

+0x0b4 UniqueProcessId : Ptr32 Void

+0x0b8 ActiveProcessLinks : \_LIST\_ENTRY

+0x0c0 Flags2 : Uint4B

+0x0c0 JobNotReallyActive : Pos 0, 1 Bit

+0x0c0 AccountingFolded : Pos 1, 1 Bit

+0x0c0 NewProcessReported : Pos 2, 1 Bit

+0x0c0 ExitProcessReported : Pos 3, 1 Bit

+0x0c0 ReportCommitChanges : Pos 4, 1 Bit

+0x0c0 LastReportMemory : Pos 5, 1 Bit

+0x0c0 ForceWakeCharge : Pos 6, 1 Bit

+0x0c0 CrossSessionCreate : Pos 7, 1 Bit

+0x0c0 NeedsHandleRundown : Pos 8, 1 Bit

+0x0c0 RefTraceEnabled : Pos 9, 1 Bit

+0x0c0 DisableDynamicCode : Pos 10, 1 Bit

+0x0c0 EmptyJobEvaluated : Pos 11, 1 Bit

+0x0c0 DefaultPagePriority : Pos 12, 3 Bits

+0x0c0 PrimaryTokenFrozen : Pos 15, 1 Bit

+0x0c0 ProcessVerifierTarget : Pos 16, 1 Bit

+0x0c0 StackRandomizationDisabled : Pos 17, 1 Bit

+0x0c0 AffinityPermanent : Pos 18, 1 Bit

+0x0c0 AffinityUpdateEnable : Pos 19, 1 Bit

+0x0c0 PropagateNode : Pos 20, 1 Bit

+0x0c0 ExplicitAffinity : Pos 21, 1 Bit

+0x0c0 ProcessExecutionState : Pos 22, 2 Bits

+0x0c0 DisallowStrippedImages : Pos 24, 1 Bit

+0x0c0 HighEntropyASLREnabled : Pos 25, 1 Bit

+0x0c0 ExtensionPointDisable : Pos 26, 1 Bit

+0x0c0 ForceRelocateImages : Pos 27, 1 Bit

+0x0c0 ProcessStateChangeRequest : Pos 28, 2 Bits

+0x0c0 ProcessStateChangeInProgress : Pos 30, 1 Bit

+0x0c0 DisallowWin32kSystemCalls : Pos 31, 1 Bit

+0x0c4 Flags : Uint4B

+0x0c4 CreateReported : Pos 0, 1 Bit

+0x0c4 NoDebugInherit : Pos 1, 1 Bit

+0x0c4 ProcessExiting : Pos 2, 1 Bit

+0x0c4 ProcessDelete : Pos 3, 1 Bit

+0x0c4 ControlFlowGuardEnabled : Pos 4, 1 Bit

+0x0c4 VmDeleted : Pos 5, 1 Bit

+0x0c4 OutswapEnabled : Pos 6, 1 Bit

+0x0c4 Outswapped : Pos 7, 1 Bit

+0x0c4 ForkFailed : Pos 8, 1 Bit

+0x0c4 Wow64VaSpace4Gb : Pos 9, 1 Bit

+0x0c4 AddressSpaceInitialized : Pos 10, 2 Bits

+0x0c4 SetTimerResolution : Pos 12, 1 Bit

+0x0c4 BreakOnTermination : Pos 13, 1 Bit

+0x0c4 DeprioritizeViews : Pos 14, 1 Bit

+0x0c4 WriteWatch : Pos 15, 1 Bit

+0x0c4 ProcessInSession : Pos 16, 1 Bit

+0x0c4 OverrideAddressSpace : Pos 17, 1 Bit

+0x0c4 HasAddressSpace : Pos 18, 1 Bit

+0x0c4 LaunchPrefetched : Pos 19, 1 Bit

+0x0c4 Background : Pos 20, 1 Bit

+0x0c4 VmTopDown : Pos 21, 1 Bit

+0x0c4 ImageNotifyDone : Pos 22, 1 Bit

+0x0c4 PdeUpdateNeeded : Pos 23, 1 Bit

+0x0c4 VdmAllowed : Pos 24, 1 Bit

+0x0c4 ProcessRundown : Pos 25, 1 Bit

+0x0c4 ProcessInserted : Pos 26, 1 Bit

+0x0c4 DefaultIoPriority : Pos 27, 3 Bits

+0x0c4 ProcessSelfDelete : Pos 30, 1 Bit

+0x0c4 SetTimerResolutionLink : Pos 31, 1 Bit

+0x0c8 ProcessQuotaUsage : [2] Uint4B

+0x0d0 ProcessQuotaPeak : [2] Uint4B

+0x0d8 PeakVirtualSize : Uint4B

+0x0dc VirtualSize : Uint4B

+0x0e0 SessionProcessLinks : \_LIST\_ENTRY

+0x0e8 ExceptionPortData : Ptr32 Void

+0x0e8 ExceptionPortValue : Uint4B

+0x0e8 ExceptionPortState : Pos 0, 3 Bits

+0x0ec Token : \_EX\_FAST\_REF

+0x0f0 WorkingSetPage : Uint4B

+0x0f4 AddressCreationLock : \_EX\_PUSH\_LOCK

+0x0f8 PageTableCommitmentLock : \_EX\_PUSH\_LOCK

+0x0fc RotateInProgress : Ptr32 \_ETHREAD

+0x100 ForkInProgress : Ptr32 \_ETHREAD

+0x104 CommitChargeJob : Ptr32 \_EJOB

+0x108 CloneRoot : \_RTL\_AVL\_TREE

+0x10c NumberOfPrivatePages : Uint4B

+0x110 NumberOfLockedPages : Uint4B

+0x114 Win32Process : Ptr32 Void

+0x118 Job : Ptr32 \_EJOB

+0x11c SectionObject : Ptr32 Void

+0x120 SectionBaseAddress : Ptr32 Void

+0x124 Cookie : Uint4B

+0x128 VdmObjects : Ptr32 Void

+0x12c WorkingSetWatch : Ptr32 \_PAGEFAULT\_HISTORY

+0x130 Win32WindowStation : Ptr32 Void

+0x134 InheritedFromUniqueProcessId : Ptr32 Void

+0x138 LdtInformation : Ptr32 Void

+0x13c OwnerProcessId : Uint4B

+0x140 Peb : Ptr32 \_PEB

+0x144 Session : Ptr32 Void

+0x148 AweInfo : Ptr32 Void

+0x14c QuotaBlock : Ptr32 \_EPROCESS\_QUOTA\_BLOCK

+0x150 ObjectTable : Ptr32 \_HANDLE\_TABLE

+0x154 DebugPort : Ptr32 Void

+0x158 PaeTop : Ptr32 Void

+0x15c DeviceMap : Ptr32 Void

+0x160 EtwDataSource : Ptr32 Void

+0x168 PageDirectoryPte : Uint8B

+0x170 ImageFileName : [15] UChar

+0x17f PriorityClass : UChar

+0x180 SecurityPort : Ptr32 Void

+0x184 SeAuditProcessCreationInfo : \_SE\_AUDIT\_PROCESS\_CREATION\_INFO

+0x188 JobLinks : \_LIST\_ENTRY

+0x190 HighestUserAddress : Ptr32 Void

+0x194 ThreadListHead : \_LIST\_ENTRY

+0x19c ActiveThreads : Uint4B

+0x1a0 ImagePathHash : Uint4B

+0x1a4 DefaultHardErrorProcessing : Uint4B

+0x1a8 LastThreadExitStatus : Int4B

+0x1ac PrefetchTrace : \_EX\_FAST\_REF

+0x1b0 LockedPagesList : Ptr32 Void

+0x1b8 ReadOperationCount : \_LARGE\_INTEGER

+0x1c0 WriteOperationCount : \_LARGE\_INTEGER

+0x1c8 OtherOperationCount : \_LARGE\_INTEGER

+0x1d0 ReadTransferCount : \_LARGE\_INTEGER

+0x1d8 WriteTransferCount : \_LARGE\_INTEGER

+0x1e0 OtherTransferCount : \_LARGE\_INTEGER

+0x1e8 CommitCharge : Uint4B

+0x1ec Vm : \_MMSUPPORT

+0x268 MmProcessLinks : \_LIST\_ENTRY

+0x270 ExitStatus : Int4B

+0x274 VadRoot : \_RTL\_AVL\_TREE

+0x278 VadHint : Ptr32 Void

+0x27c VadCount : Uint4B

+0x280 VadPhysicalPages : Uint4B

+0x284 VadPhysicalPagesLimit : Uint4B

+0x288 AlpcContext : \_ALPC\_PROCESS\_CONTEXT

+0x298 TimerResolutionLink : \_LIST\_ENTRY

+0x2a0 TimerResolutionStackRecord : Ptr32 \_PO\_DIAG\_STACK\_RECORD

+0x2a4 RequestedTimerResolution : Uint4B

+0x2a8 SmallestTimerResolution : Uint4B

+0x2b0 ExitTime : \_LARGE\_INTEGER

+0x2b8 ActiveThreadsHighWatermark : Uint4B

+0x2bc LargePrivateVadCount : Uint4B

+0x2c0 ThreadListLock : \_EX\_PUSH\_LOCK

+0x2c4 WnfContext : Ptr32 Void

+0x2c8 Spare0 : Uint4B

+0x2cc SignatureLevel : UChar

+0x2cd SectionSignatureLevel : UChar

+0x2ce Protection : \_PS\_PROTECTION

+0x2cf SpareByte20 : [1] UChar

+0x2d0 Flags3 : Uint4B

+0x2d0 Minimal : Pos 0, 1 Bit

+0x2d0 RestrictIndirectBranchPrediction : Pos 1, 1 Bit

+0x2d0 SpeculativeStoreBypassDisable : Pos 2, 1 Bit

+0x2d0 Flags3Unused : Pos 3, 11 Bits

+0x2d0 AddressPolicyFrozen : Pos 14, 1 Bit

+0x2d4 SvmReserved : Int4B

+0x2d8 SvmReserved1 : Ptr32 Void

+0x2dc SvmReserved2 : Uint4B

+0x2e0 LastFreezeInterruptTime : Uint8B

+0x2e8 DiskCounters : Ptr32 \_PROCESS\_DISK\_COUNTERS

+0x2ec KeepAliveCounter : Uint4B

+0x2f0 NoWakeKeepAliveCounter : Uint4B

+0x2f8 DeepFreezeStartTime : Uint8B

+0x300 CommitChargeLimit : Uint4B

+0x304 CommitChargePeak : Uint4B

+0x308 HighPriorityFaultsAllowed : Uint4B

+0x310 SequenceNumber : Uint8B

+0x318 CreateInterruptTime : Uint8B

+0x320 CreateUnbiasedInterruptTime : Uint8B

+0x328 SecurityDomain : Uint8B

+0x330 ModifiedPageCount : Uint4B

WinDbg에서 ETHREAD 구조체 보는 명령어:

dt nt!\_ETHREAD

||0:kd> dt nt!\_ETHREAD

+0x000 Tcb : \_KTHREAD

+0x338 CreateTime : \_LARGE\_INTEGER

+0x340 ExitTime : \_LARGE\_INTEGER

+0x340 KeyedWaitChain : \_LIST\_ENTRY

+0x348 ChargeOnlySession : Ptr32 Void

+0x34c PostBlockList : \_LIST\_ENTRY

+0x34c ForwardLinkShadow : Ptr32 Void

+0x350 StartAddress : Ptr32 Void

+0x354 TerminationPort : Ptr32 \_TERMINATION\_PORT

+0x354 ReaperLink : Ptr32 \_ETHREAD

+0x354 KeyedWaitValue : Ptr32 Void

+0x358 ActiveTimerListLock : Uint4B

+0x35c ActiveTimerListHead : \_LIST\_ENTRY

+0x364 Cid : \_CLIENT\_ID

+0x36c KeyedWaitSemaphore : \_KSEMAPHORE

+0x36c AlpcWaitSemaphore : \_KSEMAPHORE

+0x380 ClientSecurity : \_PS\_CLIENT\_SECURITY\_CONTEXT

+0x384 IrpList : \_LIST\_ENTRY

+0x38c TopLevelIrp : Uint4B

+0x390 DeviceToVerify : Ptr32 \_DEVICE\_OBJECT

+0x394 Win32StartAddress : Ptr32 Void

+0x398 LegacyPowerObject : Ptr32 Void

+0x39c ThreadListEntry : \_LIST\_ENTRY

+0x3a4 RundownProtect : \_EX\_RUNDOWN\_REF

+0x3a8 ThreadLock : \_EX\_PUSH\_LOCK

+0x3ac ReadClusterSize : Uint4B

+0x3b0 MmLockOrdering : Int4B

+0x3b4 CmLockOrdering : Int4B

+0x3b8 CrossThreadFlags : Uint4B

+0x3b8 Terminated : Pos 0, 1 Bit

+0x3b8 ThreadInserted : Pos 1, 1 Bit

+0x3b8 HideFromDebugger : Pos 2, 1 Bit

+0x3b8 ActiveImpersonationInfo : Pos 3, 1 Bit

+0x3b8 HardErrorsAreDisabled : Pos 4, 1 Bit

+0x3b8 BreakOnTermination : Pos 5, 1 Bit

+0x3b8 SkipCreationMsg : Pos 6, 1 Bit

+0x3b8 SkipTerminationMsg : Pos 7, 1 Bit

+0x3b8 CopyTokenOnOpen : Pos 8, 1 Bit

+0x3b8 ThreadIoPriority : Pos 9, 3 Bits

+0x3b8 ThreadPagePriority : Pos 12, 3 Bits

+0x3b8 RundownFail : Pos 15, 1 Bit

+0x3b8 UmsForceQueueTermination : Pos 16, 1 Bit

+0x3b8 ReservedCrossThreadFlags : Pos 17, 15 Bits

+0x3bc SameThreadPassiveFlags : Uint4B

+0x3bc ActiveExWorker : Pos 0, 1 Bit

+0x3bc MemoryMaker : Pos 1, 1 Bit

+0x3bc ClonedThread : Pos 2, 1 Bit

+0x3bc KeyedEventInUse : Pos 3, 1 Bit

+0x3bc SelfTerminate : Pos 4, 1 Bit

+0x3c0 SameThreadApcFlags : Uint4B

+0x3c0 HardFaultBehavior : Pos 0, 1 Bit

+0x3c0 StartAddressInvalid : Pos 1, 1 Bit

+0x3c0 EtwCalloutActive : Pos 2, 1 Bit

+0x3c0 OwnsProcessWorkingSetExclusive : Pos 3, 1 Bit

+0x3c0 OwnsProcessWorkingSetShared : Pos 4, 1 Bit

+0x3c0 OwnsSystemCacheWorkingSetExclusive : Pos 5, 1 Bit

+0x3c0 OwnsSystemCacheWorkingSetShared : Pos 6, 1 Bit

+0x3c0 OwnsSessionWorkingSetExclusive : Pos 7, 1 Bit

+0x3c1 OwnsSessionWorkingSetShared : Pos 0, 1 Bit

+0x3c1 OwnsProcessAddressSpaceExclusive : Pos 1, 1 Bit

+0x3c1 OwnsProcessAddressSpaceShared : Pos 2, 1 Bit

+0x3c1 SuppressSymbolLoad : Pos 3, 1 Bit

+0x3c1 Prefetching : Pos 4, 1 Bit

+0x3c1 OwnsVadExclusive : Pos 5, 1 Bit

+0x3c1 OwnsChangeControlAreaExclusive : Pos 6, 1 Bit

+0x3c1 OwnsChangeControlAreaShared : Pos 7, 1 Bit

+0x3c2 OwnsPagedPoolWorkingSetExclusive : Pos 0, 1 Bit

+0x3c2 OwnsPagedPoolWorkingSetShared : Pos 1, 1 Bit

+0x3c2 OwnsSystemPtesWorkingSetExclusive : Pos 2, 1 Bit

+0x3c2 OwnsSystemPtesWorkingSetShared : Pos 3, 1 Bit

+0x3c2 TrimTrigger : Pos 4, 2 Bits

+0x3c2 Spare2 : Pos 6, 2 Bits

+0x3c3 SystemPagePriorityActive : Pos 0, 1 Bit

+0x3c3 SystemPagePriority : Pos 1, 3 Bits

+0x3c3 Spare3 : Pos 4, 4 Bits

+0x3c4 CacheManagerActive : UChar

+0x3c5 DisablePageFaultClustering : UChar

+0x3c6 ActiveFaultCount : UChar

+0x3c7 LockOrderState : UChar

+0x3c8 AlpcMessageId : Uint4B

+0x3cc AlpcMessage : Ptr32 Void

+0x3cc AlpcReceiveAttributeSet : Uint4B

+0x3d0 ExitStatus : Int4B

+0x3d4 AlpcWaitListEntry : \_LIST\_ENTRY

+0x3dc CacheManagerCount : Uint4B

+0x3e0 IoBoostCount : Uint4B

+0x3e4 BoostList : \_LIST\_ENTRY

+0x3ec DeboostList : \_LIST\_ENTRY

+0x3f4 BoostListLock : Uint4B

+0x3f8 IrpListLock : Uint4B

+0x3fc ReservedForSynchTracking : Ptr32 Void

+0x400 CmCallbackListHead : \_SINGLE\_LIST\_ENTRY

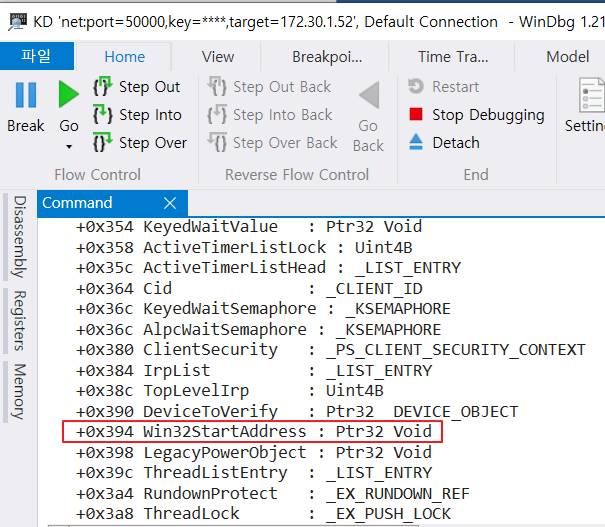
+0x404 ActivityId : Ptr32 \_GUID

+0x408 SeLearningModeListHead : \_SINGLE\_LIST\_ENTRY

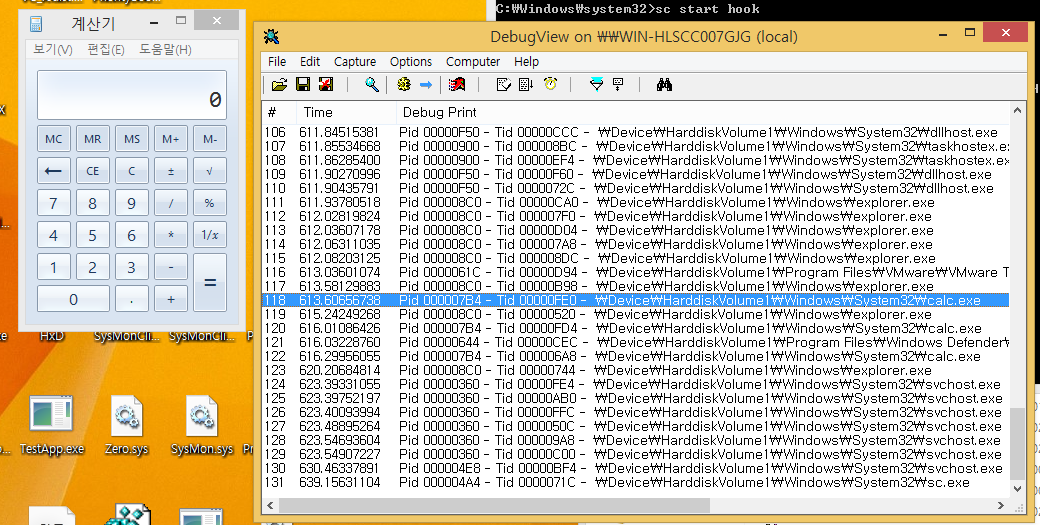
+0x40c VerifierContext : Ptr32 Void

+0x410 KernelStackReference : Uint4B

+0x414 AdjustedClientToken : Ptr32 Void



ETHREAD를 분석하면 offset 0x394에서 ring3 아래의 thread callback function(스레드 콜백 함수)의 주소가 저장된다는 결론을 내릴 수 있습니다. 그런 다음 이 오프셋을 조작하여 스레드 생성을 위장한 채로 가로채는 ret로 바꿀 수 있습니다.



위의 프로세스 ID를 통해 EPROCESS 등을 얻기 위한 API는 모두 Microsoft에서 공개하지 않습니다. 그러나 exported. 선언하기만 하면 됩니다. 하지만 공개되지 않았기 때문에 마이크로소프트가 어떻게 사용하느냐에 따라 달라진다. Microsoft에서 사용한 후 참조를 계산하므로 동일한 작업을 수행해야 합니다. 메모리가 C3으로 변경되면 스레드를 생성할 수 없습니다.