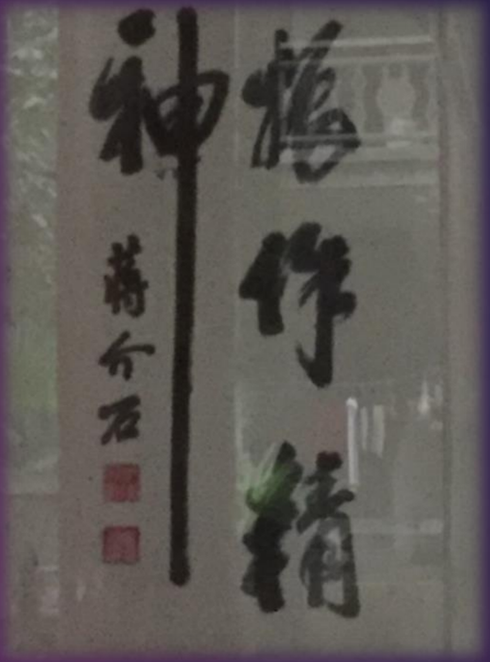
Windows 10探微

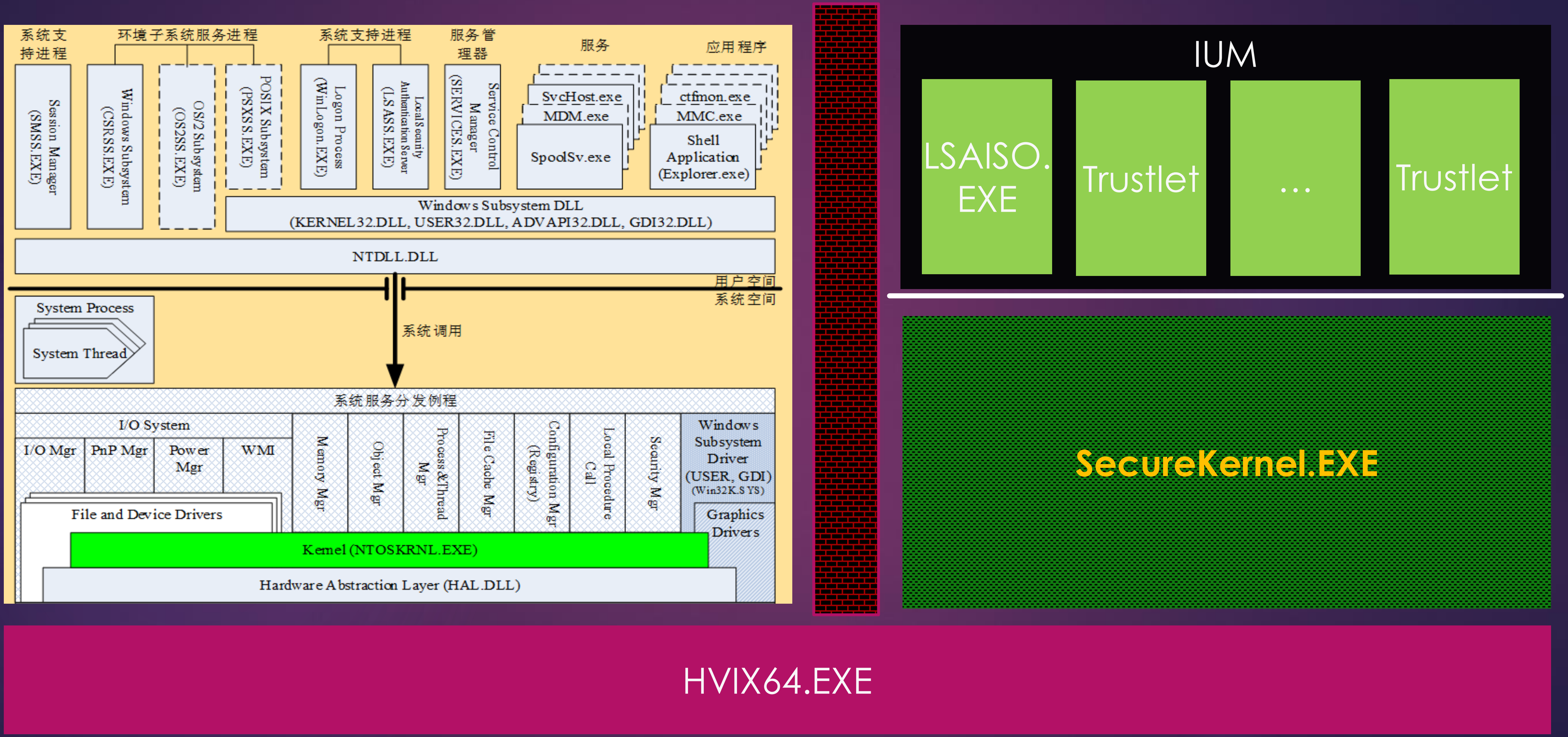


安全内核和IUM

格蠹老雷

2016/12/16 庐山中正行营

架构



系统支持进程

环境子系统服务进程

系统支持进程

服务管理器

服务

应用程序

Session Manager (SMSS.EXE)

Windows Subsystem (CSRSS.EXE)

OS/2 Subsystem (OS2SS.EXE)

POSIX Subsystem (PSXSS.EXE)

Logon Process (WinLogon.EXE)

Local Security Authentication Server (LSASS.EXE)

Service Control Manager (SERVICES.EXE)

SvcHost.exe

MDM.exe

SpoolSv.exe

ctfmon.exe

MMC.exe

Shell Application (Explorer.exe)

Windows Subsystem DLL
(KERNEL32.DLL, USER32.DLL, ADVAPI32.DLL, GDI32.DLL)

NTDLL.DLL

用户空间

系统空间

System Process

System Thread

系统调用

系统服务分发例程

I/O System

I/O Mgr

PnP Mgr

Power Mgr

WMI

File and Device Drivers

Memory Mgr

Object Mgr

Process&Thread Mgr

File Cache Mgr

Configuration Mgr (Registry)

Local Procedure Call

Security Mgr

Windows Subsystem Driver (USER, GDI) (Win32K.SYS)

Graphics Drivers

Kernel (NTOSKRNL.EXE)

Hardware Abstraction Layer (HAL.DLL)

IUM

LSAISO.EXE

Trustlet

…

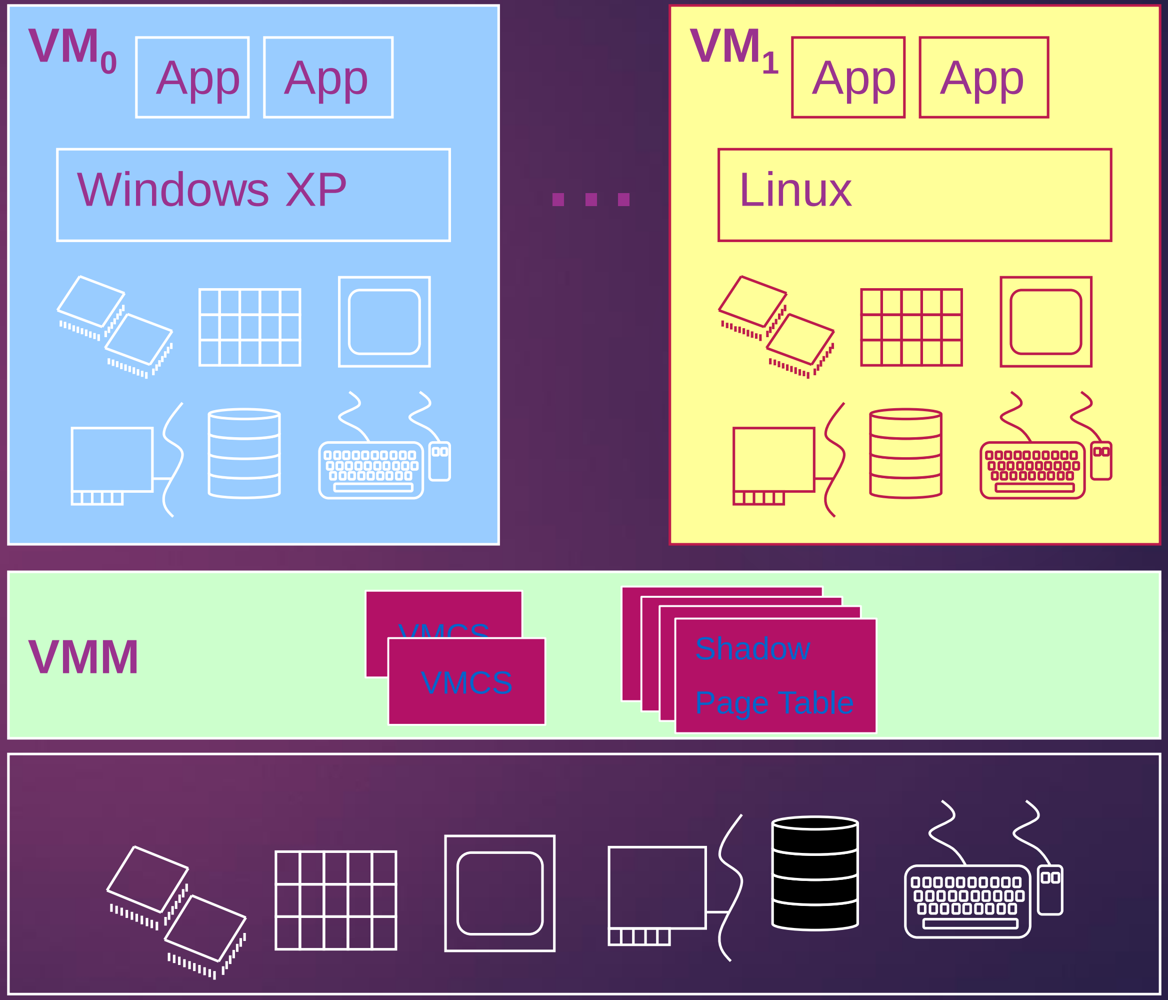
Trustlet

SecureKernel.EXE

HVIX64.EXE

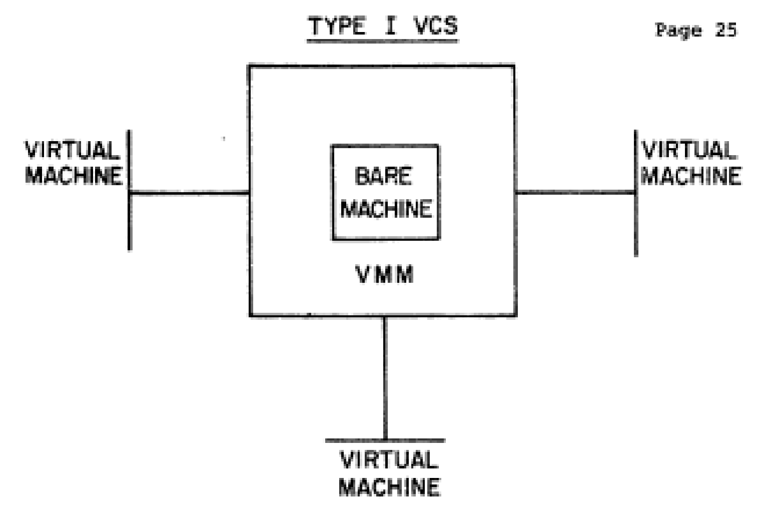
VT基础

- Virtual Machine Control Structures(VMCS)
- 管理VM的纲领
  - 每个VM至少一份
  - CPU相关
  - 必不可少
- 进出VM的规则
  - VM监管策略
- VMM， hypervisor
  - 最高领袖，ring -1
- CPU定义的数据结构
- IA-32卷3B

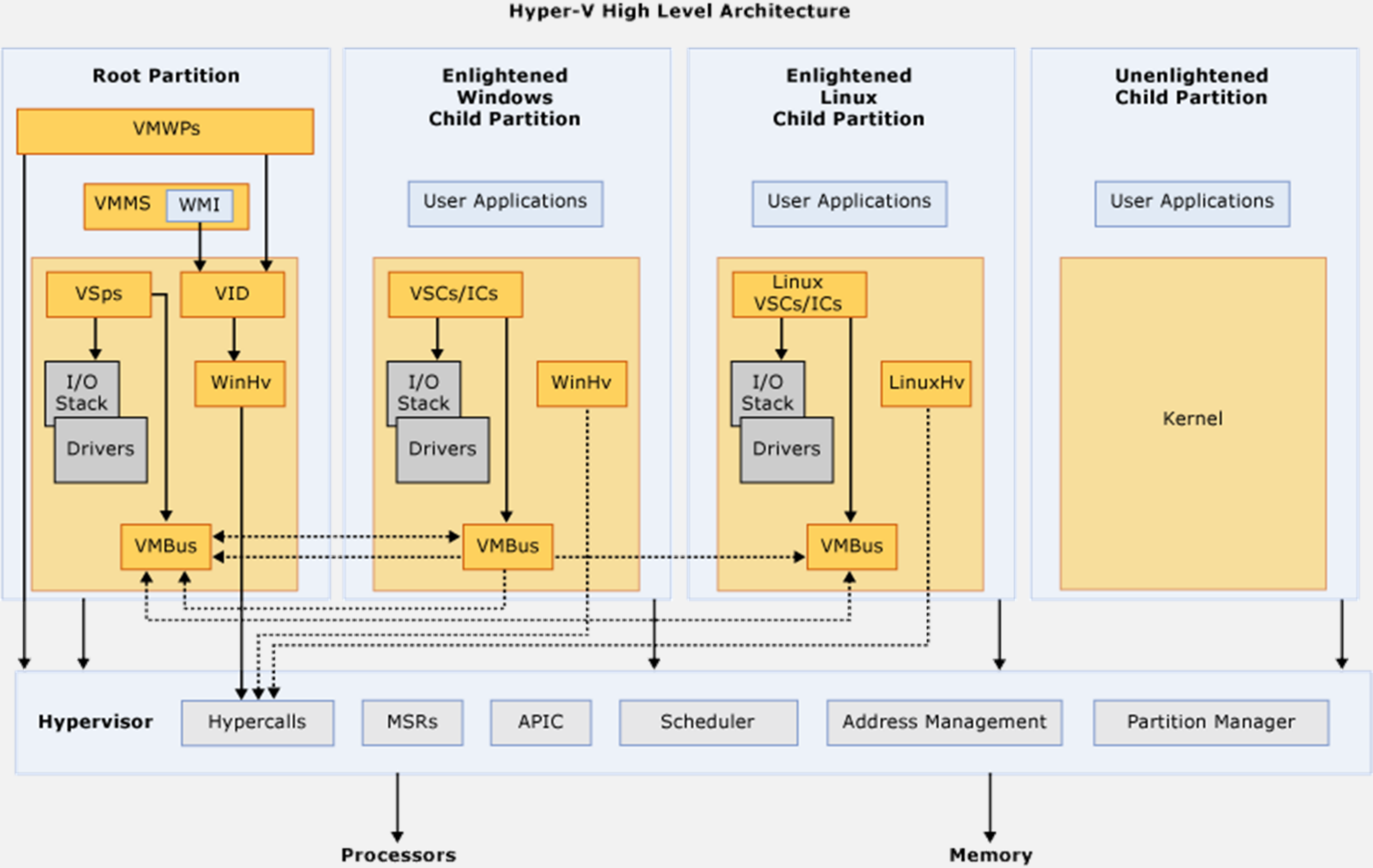


Hyper-V

- 类型I VMM，与XEN类似
- 主要用于服务器，Win8时引入到终端版本的Windows，称为client Hyper-V
- 与Windows 10和Server 2016对应的版本是Hyper-V 5.0，内建VSM（VIRTUAL SECURE MODE）支持



Hyper-V架构



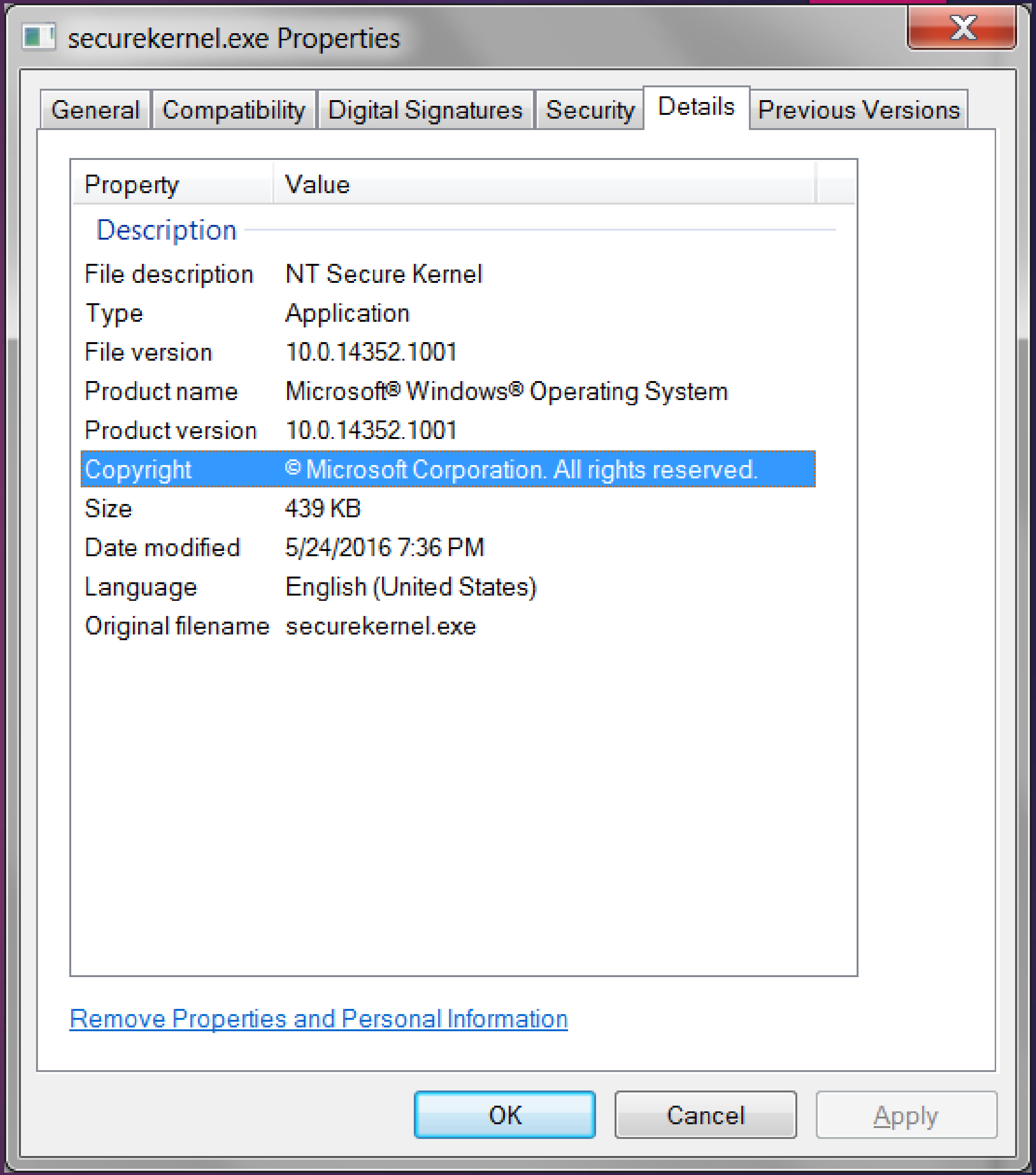
- VMMS – Virtual Machine Management Service
- VMWP – Virtual Machine Worker Process
- VSP – Virtualization Service Provider
- VSC – Virtualization Service Client
- WinHv – Windows Hypervisor Interface Library

Win10与Hyper-V

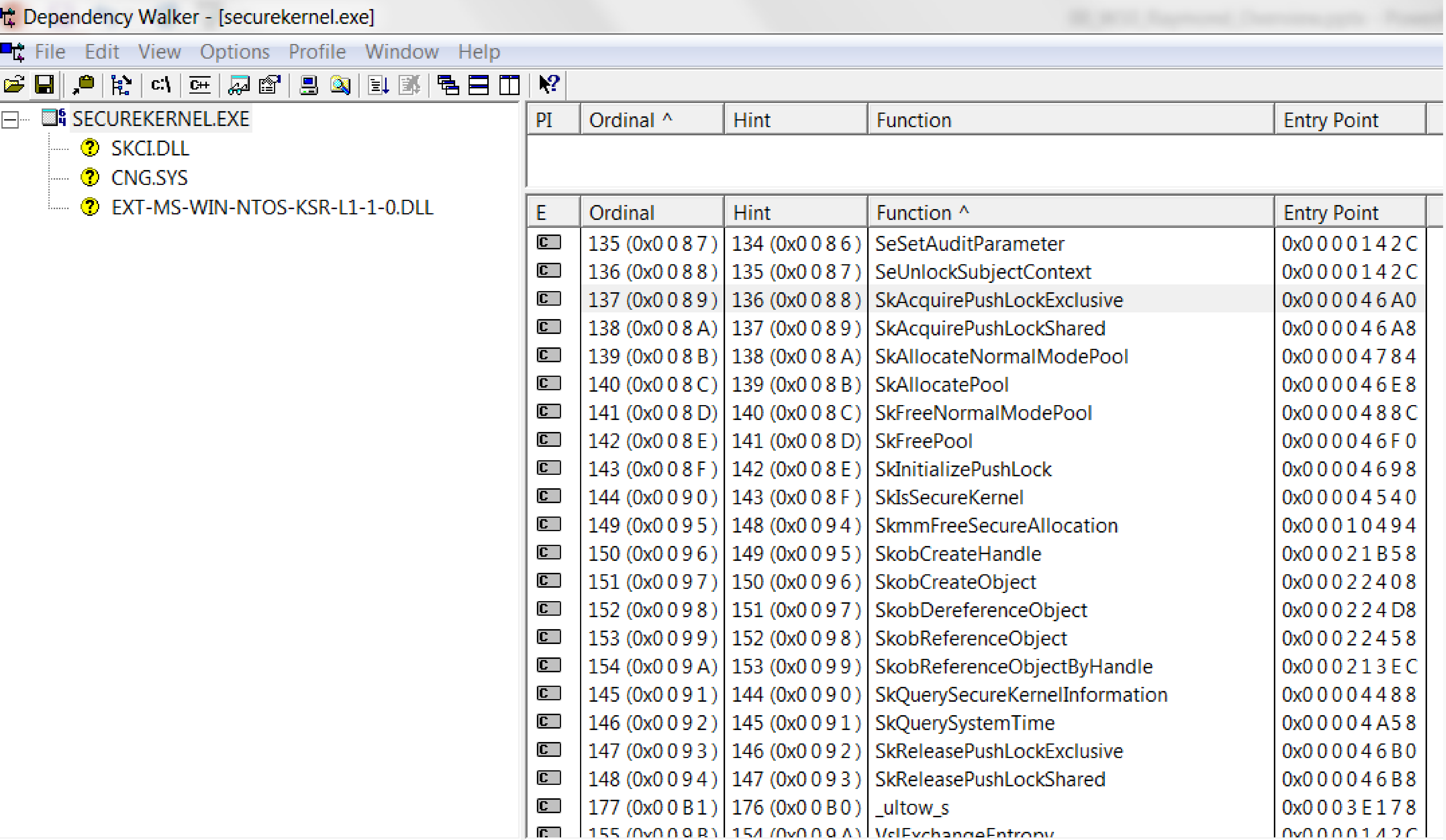
- 从总体架构角度看，Win10运行在Hyper-V的根分区中（Root Partition）
- 从软件发布形式看，Win10中包含了一份终端版本的Hyper-V 5.0，Hyper-V是Win10的一个功能组件（feature）
- 你中有我，我中有你

SECUREKERNEL.EXE

- 安全内核，简称SK，SKM
- 为IUM提供服务
- 从实现的功能来看，不是真正的内核，更像是内核的特别代理（proxy）
- 大约400KB



Property	Value
Description	
File description	NT Secure Kernel
Type	Application
File version	10.0.14352.1001
Product name	Microsoft® Windows® Operating System
Product version	10.0.14352.1001
Copyright	© Microsoft Corporation. All rights reserved.
Size	439 KB
Date modified	5/24/2016 7:36 PM
Language	English (United States)
Original filename	securekernel.exe



Dependency Walker - [securekernel.exe]

- SECUREKERNEL.EXE
  - SKCI.DLL
  - CNG.SYS
  - EXT-MS-WIN-NTOS-KSR-L1-1-0.DLL

PI	Ordinal ^	Hint	Function	Entry Point

E	Ordinal	Hint	Function ^	Entry Point
C	135 (0x0087)	134 (0x0086)	SeSetAuditParameter	0x0000142C
C	136 (0x0088)	135 (0x0087)	SeUnlockSubjectContext	0x0000142C
C	137 (0x0089)	136 (0x0088)	SkAcquirePushLockExclusive	0x000046A0
C	138 (0x008A)	137 (0x0089)	SkAcquirePushLockShared	0x000046A8
C	139 (0x008B)	138 (0x008A)	SkAllocateNormalModePool	0x00004784
C	140 (0x008C)	139 (0x008B)	SkAllocatePool	0x000046E8
C	141 (0x008D)	140 (0x008C)	SkFreeNormalModePool	0x0000488C
C	142 (0x008E)	141 (0x008D)	SkFreePool	0x000046F0
C	143 (0x008F)	142 (0x008E)	SkInitializePushLock	0x00004698
C	144 (0x0090)	143 (0x008F)	SkIsSecureKernel	0x00004540
C	149 (0x0095)	148 (0x0094)	SkmmFreeSecureAllocation	0x00010494
C	150 (0x0096)	149 (0x0095)	SkobCreateHandle	0x00021B58
C	151 (0x0097)	150 (0x0096)	SkobCreateObject	0x00022408
C	152 (0x0098)	151 (0x0097)	SkobDereferenceObject	0x000224D8
C	153 (0x0099)	152 (0x0098)	SkobReferenceObject	0x00022458
C	154 (0x009A)	153 (0x0099)	SkobReferenceObjectByHandle	0x000213EC
C	145 (0x0091)	144 (0x0090)	SkQuerySecureKernelInformation	0x00004488
C	146 (0x0092)	145 (0x0091)	SkQuerySystemTime	0x00004A58
C	147 (0x0093)	146 (0x0092)	SkReleasePushLockExclusive	0x000046B0
C	148 (0x0094)	147 (0x0093)	SkReleasePushLockShared	0x000046B8
C	177 (0x00B1)	176 (0x00B0)	_ultow_s	0x0003E178
C	155 (0x009B)	154 (0x009A)	VslExchangeEntropy	0x0000142C

SKCI.DLL

- Secure Kernel Code Integrity, 基于Hypervisor的代码完整性检查模块 (HYPERVISOR-BASED CODE INTEGRITY, HBCI)，其功能与CI.DLL类似
- 与SK一起加载，运行在安全内核空间中，输出以下函数：
- SkciCreateCodeCatalog
- SkciCreateSecureImage
- SkciFinalizeSecureImageHash
- SkciFinishImageValidation
- SkciFreeImageContext
- SkciInitialize
- SkciTransferVersionResource
- SkciValidateDynamicCodePages
- SkciValidateImageData

CNG.SYS

- BCryptCloseAlgorithmProvider
- BCryptCreateHash
- BCryptDecrypt
- BCryptDestroyHash
- BCryptDestroyKey
- BCryptEncrypt
- BCryptFinishHash
- BCryptGenRandom
- BCryptGenerateSymmetricKey
- BCryptGetProperty
- BCryptHashData
- BCryptImportKeyPair
- BCryptKeyDerivation
- BCryptOpenAlgorithmProvider

BCryptSetProperty
BCryptSignHash
CngGetFipsAlgorithmMode
EntropyPoolTriggerReseedForIum
EntropyProvideData
EntropyRegisterSource
SystemPrng

加解密服务

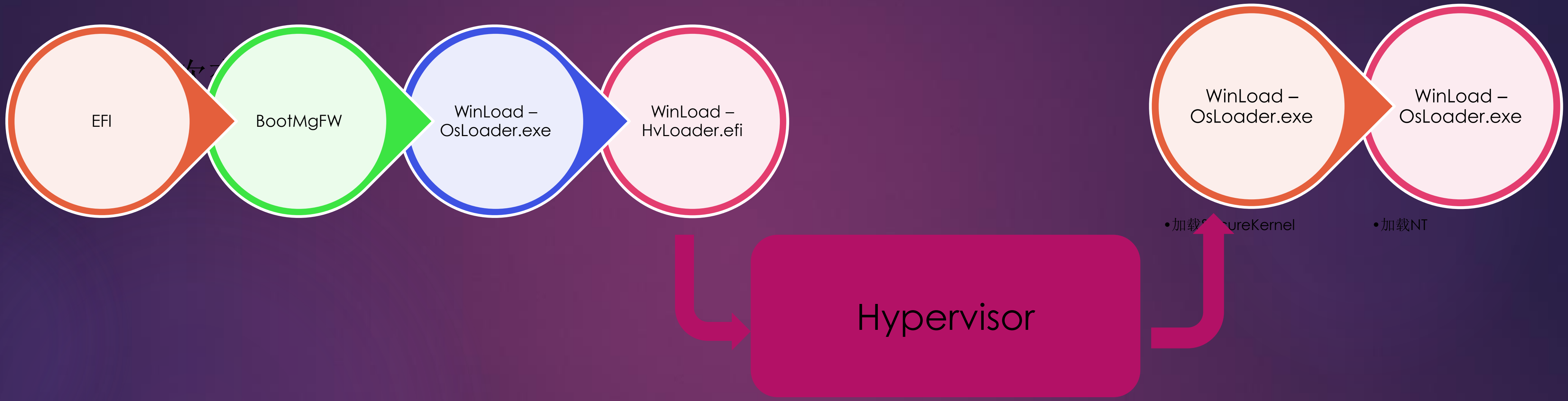
SK的组件（函数命名）

- CRT/RTL：memcpy, atoi, Rtlxxx, etc
- 经典NT内核函数的子集
- Etw 事件追踪
- Ex 执行体
- DbgPrintEx 调试信息输出
- Io 输入输出
- Ke 内核
- Ob 对象管理器
- Mm 内存管理器
- Ps 进程管理器
- Se 安全
- 安全内核的一般函数，SkXXX
- NT内核的代理函数
  - Skob，Skmm， Ske（Ski），Skps
- Ium

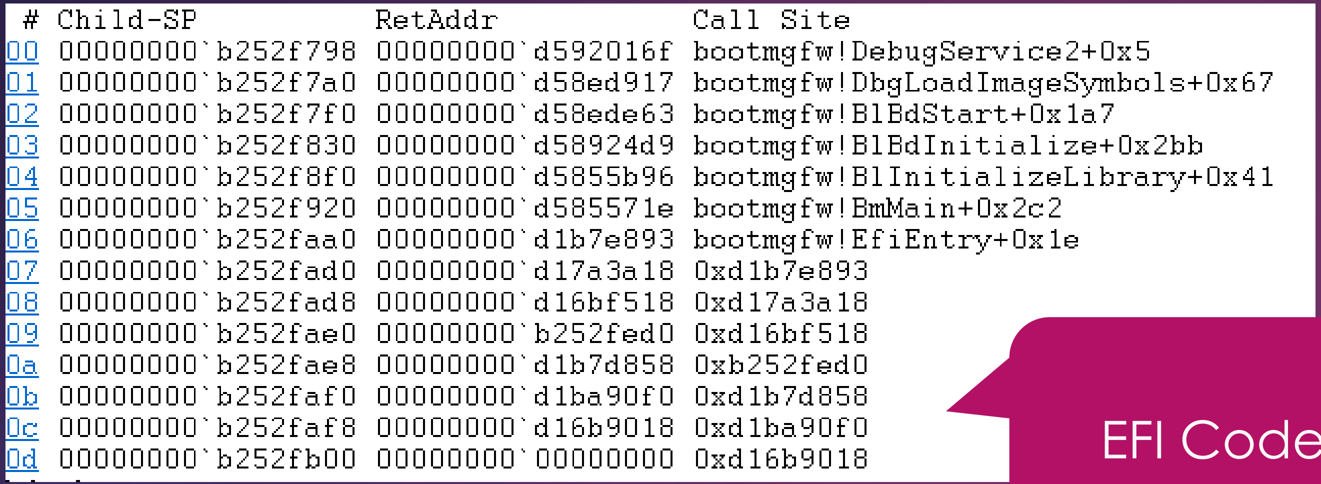
IUMDLL.DLL

- IUM与SKM的桥梁
- 公开如下系统调用
- 0x80000000 – IumGetIdk
- 0x80000001 – IumSetTrustletInstance
- 0x80000003 – IumCrypto
- 0x80000002 – IumPostMailbox
- 0x80000004 – IumStoragePut
- 0x80000005 – IumStorageGet

启动过程


EFI
BootMgFW
WinLoad – OsLoader.exe
WinLoad – HvLoader.efi
Hypervisor
WinLoad – OsLoader.exe
WinLoad – OsLoader.exe
•加载SecureKernel
•加载NT

启动过程 - BootMgFW



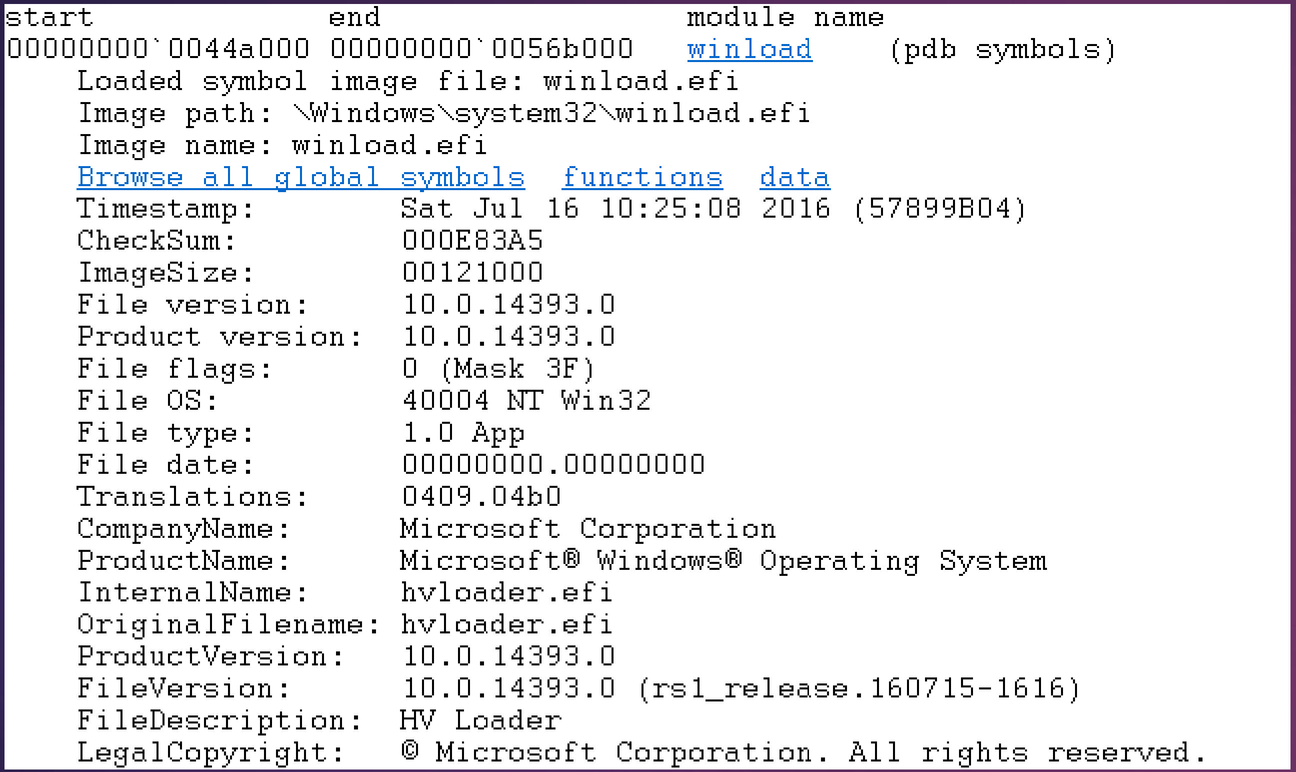
```
 # Child-SP          RetAddr           Call Site
00 00000000`b252f798 00000000`d592016f bootmgfw!DebugService2+0x5
01 00000000`b252f7a0 00000000`d58ed917 bootmgfw!DbgLoadImageSymbols+0x67
02 00000000`b252f7f0 00000000`d58ede63 bootmgfw!BlBdStart+0x1a7
03 00000000`b252f830 00000000`d58924d9 bootmgfw!BlBdInitialize+0x2bb
04 00000000`b252f8f0 00000000`d5855b96 bootmgfw!BlInitializeLibrary+0x41
05 00000000`b252f920 00000000`d585571e bootmgfw!BmMain+0x2c2
06 00000000`b252faa0 00000000`d1b7e893 bootmgfw!EfiEntry+0x1e
07 00000000`b252fad0 00000000`d17a3a18 0xd1b7e893
08 00000000`b252fad8 00000000`d16bf518 0xd17a3a18
09 00000000`b252fae0 00000000`b252fed0 0xd16bf518
0a 00000000`b252fae8 00000000`d1b7d858 0xb252fed0
0b 00000000`b252faf0 00000000`d1ba90f0 0xd1b7d858
0c 00000000`b252faf8 00000000`d16b9018 0xd1ba90f0
0d 00000000`b252fb00 00000000`00000000 0xd16b9018
```

EFI Code

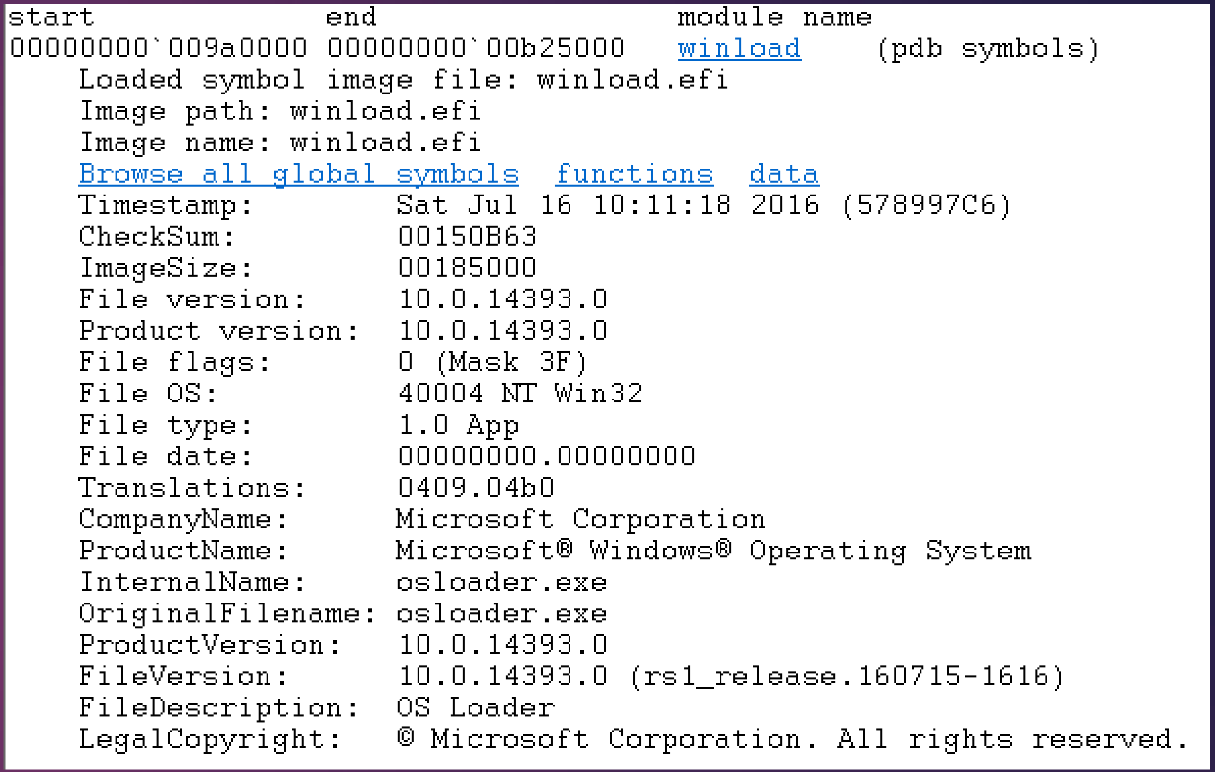
加载Hyper-V加载器

- winload!DebugService2
- winload!DbgLoadImageSymbols
- winload!BlBdStart
- winload!ImgArchEfiStartBootApplication
- winload!BlImgStartBootApplication
- winload!**HvlpLaunchHvLoader**
- winload!OslArchHypervisorSetup
- winload!OslPrepareTarget
- winload!OslpMain
- winload!OslMain

两个WinLoad



```
start             end                 module name
00000000`0044a000 00000000`0056b000   winload    (pdb symbols)
    Loaded symbol image file: winload.efi
    Image path: \Windows\system32\winload.efi
    Image name: winload.efi
    Browse all global symbols  functions  data
    Timestamp:        Sat Jul 16 10:25:08 2016 (57899B04)
    CheckSum:         000E83A5
    ImageSize:        00121000
    File version:     10.0.14393.0
    Product version:  10.0.14393.0
    File flags:       0 (Mask 3F)
    File OS:          40004 NT Win32
    File type:        1.0 App
    File date:        00000000.00000000
    Translations:     0409.04b0
    CompanyName:      Microsoft Corporation
    ProductName:      Microsoft® Windows® Operating System
    InternalName:     hvloader.efi
    OriginalFilename: hvloader.efi
    ProductVersion:   10.0.14393.0
    FileVersion:      10.0.14393.0 (rs1_release.160715-1616)
    FileDescription:  HV Loader
    LegalCopyright:   © Microsoft Corporation. All rights reserved.
```



```
start             end                 module name
00000000`009a0000 00000000`00b25000   winload    (pdb symbols)
    Loaded symbol image file: winload.efi
    Image path: winload.efi
    Image name: winload.efi
    Browse all global symbols  functions  data
    Timestamp:        Sat Jul 16 10:11:18 2016 (578997C6)
    CheckSum:         00150B63
    ImageSize:        00185000
    File version:     10.0.14393.0
    Product version:  10.0.14393.0
    File flags:       0 (Mask 3F)
    File OS:          40004 NT Win32
    File type:        1.0 App
    File date:        00000000.00000000
    Translations:     0409.04b0
    CompanyName:      Microsoft Corporation
    ProductName:      Microsoft® Windows® Operating System
    InternalName:     osloader.exe
    OriginalFilename: osloader.exe
    ProductVersion:   10.0.14393.0
    FileVersion:      10.0.14393.0 (rs1_release.160715-1616)
    FileDescription:  OS Loader
    LegalCopyright:   © Microsoft Corporation. All rights reserved.
```

加载阎罗王（-1层的老大）

- 00 winload!DbgBreakPointWithStatus
- 01 winload!vDbgPrintExWithPrefixIntern
- 02 winload!DbgPrint
- 03 winload!BalDebugPrint
- 04 winload!BtPrepareHypervisorLaunch
- 05 winload!HvlpPrepareHypervisorForL
- 06 winload!HvlMain
- 07 0x0

加载SK神秘内核

- winload!OslLoadImage
- winload!OslpVsmLoadModules
- winload!OslVsmSetup
- winload!OslPrepareTarget
- winload!OslpMain
- winload!OslMain
- 0x0

```
kd> dU r8
fffff800`5039ff90  "\Windows\system32\securekernel.e"
fffff800`5039ffd0  "xe"

kd> dU r8
fffff800`503a14f0  "\Windows\system32\skci.dll"

kd> dU r8
fffff800`503a14f0  "\Windows\system32\cng.sys"

fffff800`503a4250
"\Windows\System32\drivers\secure"
fffff800`503a4290  "kernel.exe"
```

DebugPrint

- SecureKernel virtual image base = 0xFFFFF80053200000 Image size = 0x7f000 Entry point = 0xFFFFF800532010C4

```
winload!BlBdPrint
winload!BlStatusPrint
winload!OslpVsmLoadModules
winload!OslVsmSetup
winload!OslPrepareTarget
winload!OslpMain
winload!OslMain
0x0
```

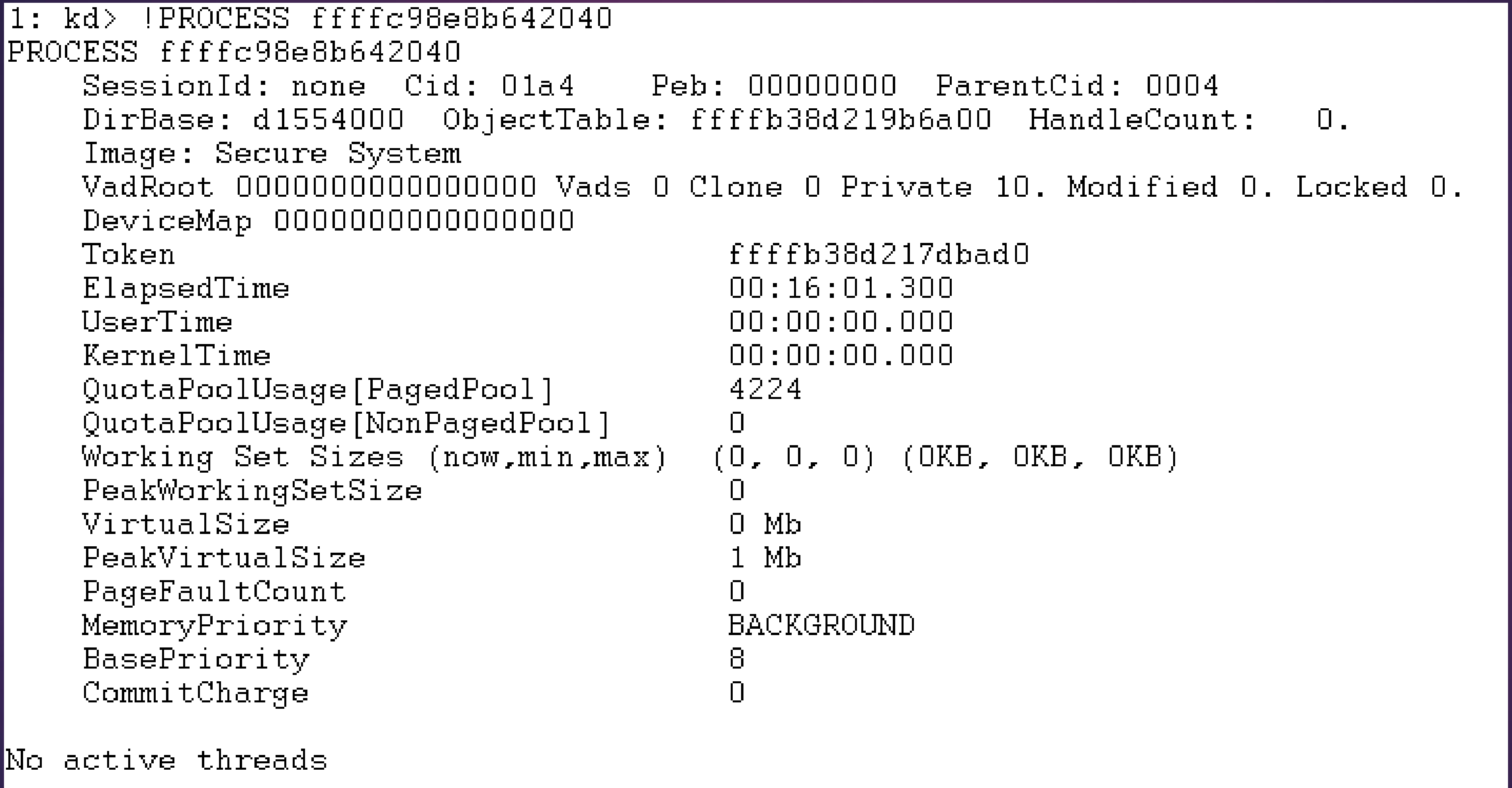
NTOS中初始化代理设施

- 00 nt!PsDispatchIumService
- 01 nt!VslpEnterIumSecureMode
- 02 **nt!VslpIumPhase0Initialize**
- 03 nt!VslInitSystem
- 04 nt!HvlPhase1Initialize
- 05 nt!InitBootProcessor
- 06 nt!ExpInitializeExecutive
- 07 nt!KiInitializeKernel
- 08 nt!KiSystemStartup

进程初始化

- 00 **nt!PspIumInitialize**
- 01 nt!PspInitPhase0
- 02 nt!InitBootProcessor
- 03 nt!ExpInitializeExecutive
- 04 nt!KiInitializeKernel
- 05 nt!KiSystemStartup

SK影子进程



```
1: kd> !PROCESS ffffc98e8b642040
PROCESS ffffc98e8b642040
    SessionId: none  Cid: 01a4    Peb: 00000000  ParentCid: 0004
    DirBase: d1554000  ObjectTable: ffffb38d219b6a00  HandleCount:   0.
    Image: Secure System
    VadRoot 0000000000000000 Vads 0 Clone 0 Private 10. Modified 0. Locked 0.
    DeviceMap 0000000000000000
    Token                             ffffb38d217dbad0
    ElapsedTime                       00:16:01.300
    UserTime                          00:00:00.000
    KernelTime                        00:00:00.000
    QuotaPoolUsage[PagedPool]         4224
    QuotaPoolUsage[NonPagedPool]      0
    Working Set Sizes (now,min,max)  (0, 0, 0) (0KB, 0KB, 0KB)
    PeakWorkingSetSize                0
    VirtualSize                       0 Mb
    PeakVirtualSize                   1 Mb
    PageFaultCount                    0
    MemoryPriority                    BACKGROUND
    BasePriority                      8
    CommitCharge                      0

No active threads
```

隔离增强安全

权力隔离

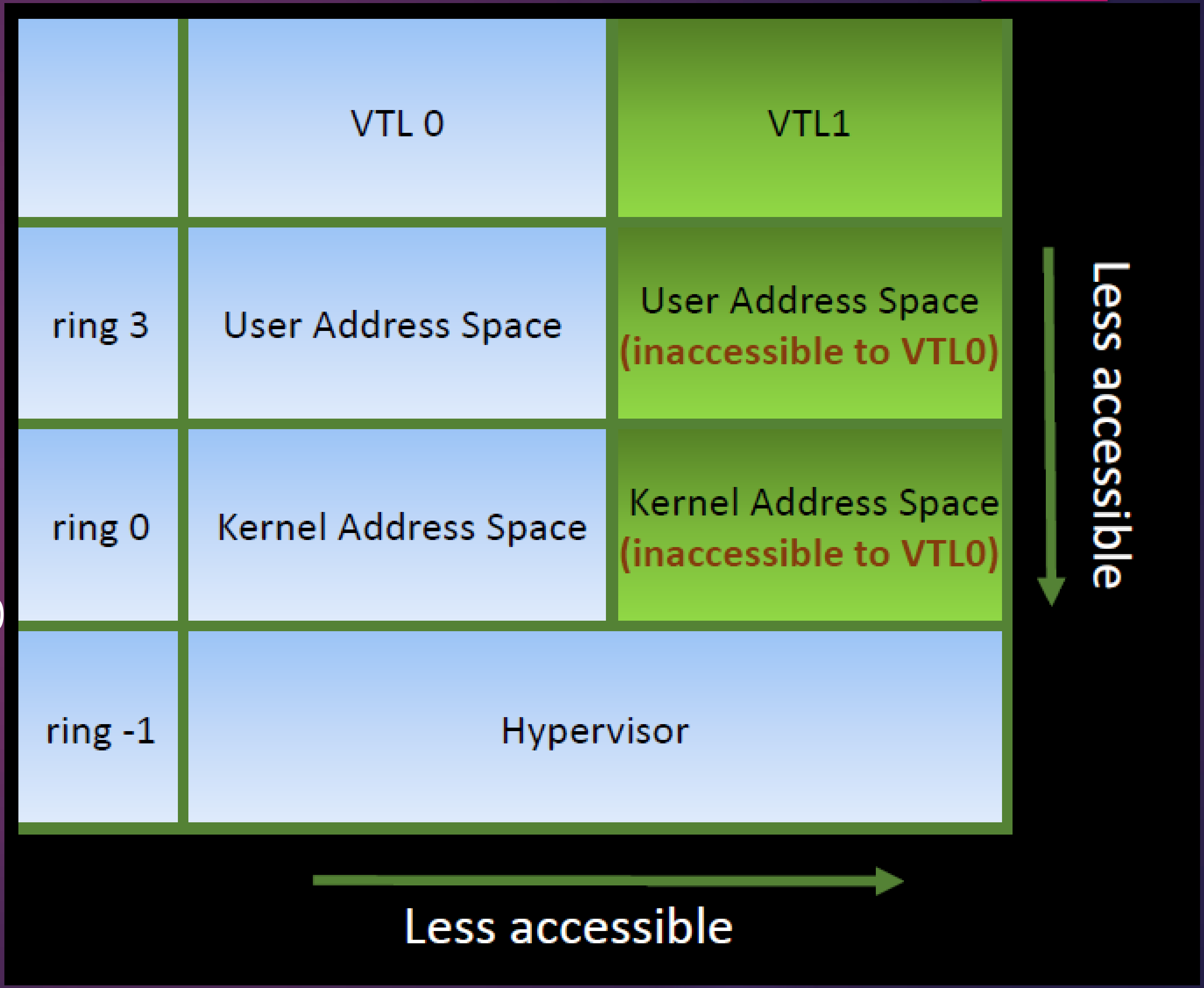
- Hypervisor具有最高权利，但是其职能单一，逻辑很少，攻击面小
- 虚拟机分区，机器边界，普通OS和安全OS运行在不同分区

角色隔离

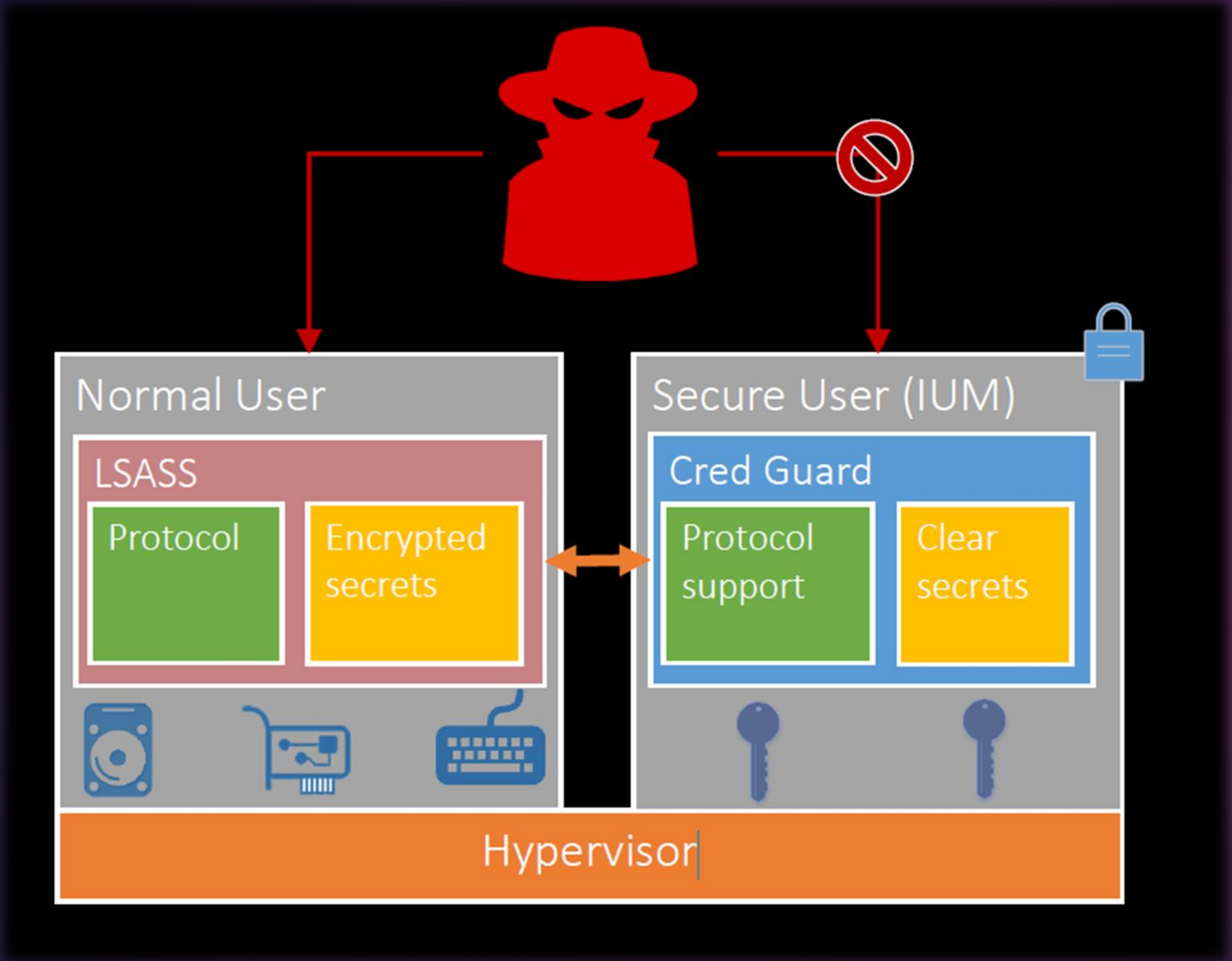
- IUM运行在特别设计的安全内核之上，不依赖普通内核
- IUM中的多个Trustlet相互隔离，不可以相互访问

VTL

- Virtual Trust Levels
- 使用VT和SLAT技术隔离内存
  - Second Level Address Translation (SLAT)
  - Guest virtual > Guest physical > System physical
- 常规的Windows 10运行在VTL 0
- 安全内核运行在VTL 1
- 将来可能扩展更多的VTL



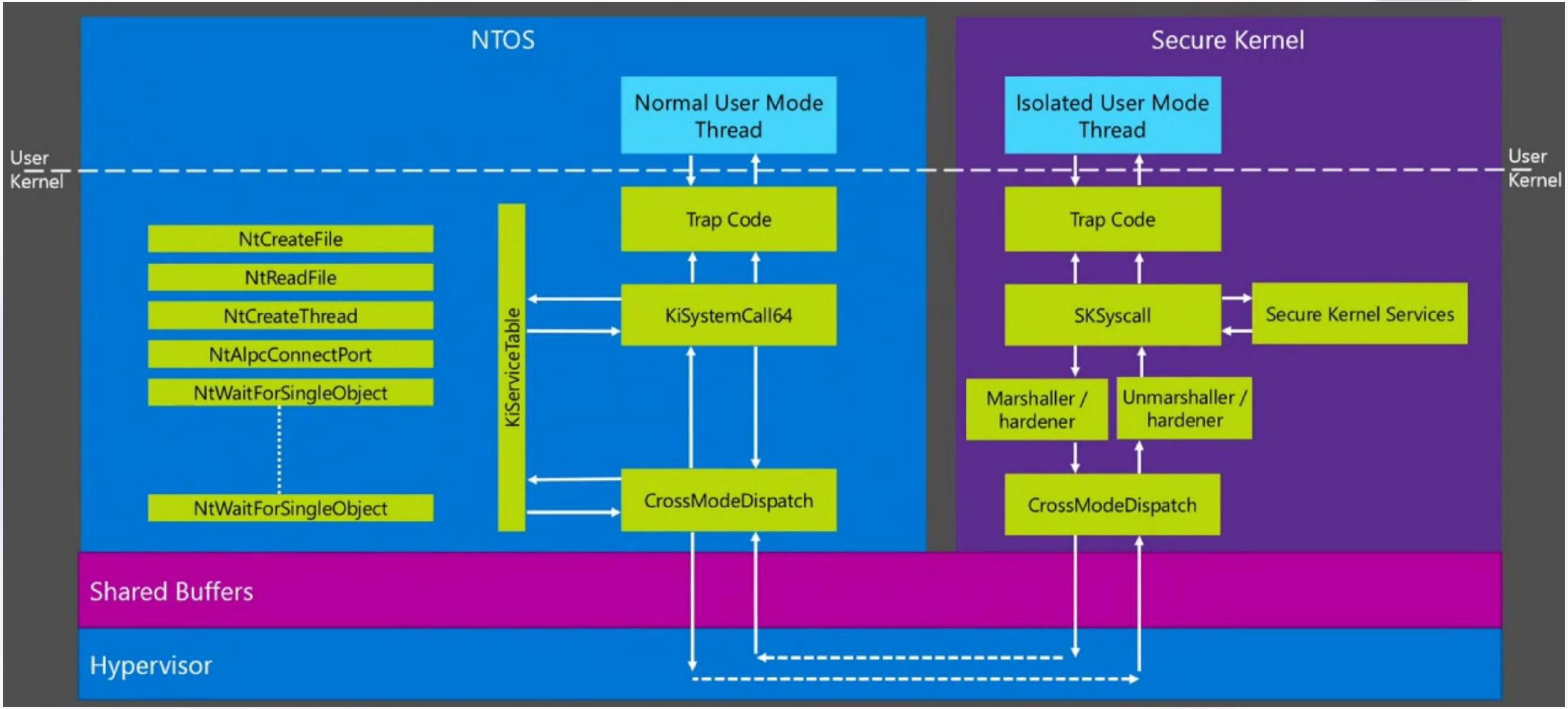
	VTL 0	VTL1
ring 3	User Address Space	User Address Space (inaccessible to VTL0)
ring 0	Kernel Address Space	Kernel Address Space (inaccessible to VTL0)
ring -1	Hypervisor	



原始密信数据（比如密码的HASH）保存在VTL 1，VTL 0中的恶件访问不到

加密后才传递到VTL 0

通信


NTOS
Secure Kernel
User
Kernel
Normal User Mode Thread
Isolated User Mode Thread
User
Kernel
Trap Code
Trap Code
NtCreateFile
NtReadFile
NtCreateThread
NtAlpcConnectPort
NtWaitForSingleObject
NtWaitForSingleObject
KiServiceTable
KiSystemCall64
SKSyscall
Secure Kernel Services
Marshaller / hardener
Unmarshaller / hardener
CrossModeDispatch
CrossModeDispatch
Shared Buffers
Hypervisor

调用SK的安全服务

- 00 nt!PsDispatchIumService
- 01 **nt!VslpEnterIumSecureMode**
- 02 nt!**VslFinishSecureImageValidation**
- 03 CI!CiHvciVerifyFileHashSignedFile
- 04 CI!CiHvciVerifyPageHashSignedFile
- 05 CI!CipGetPageHashesForFile
- 06 CI!CipValidatePageHash
- 07 CI!CipValidateImageHash
- 08 CI!CiValidateImageHeader
- 09 nt!SeValidateImageHeader
- 0a nt!MiValidateSectionCreate
- 0b nt!MiCreateNewSection
- 0c nt!MiCreateSection
- 0d nt!MmCreateSpecialImageSection
- 0e nt!PspLocateSystemDll
- 0f nt!PsLocateSystemDlls
- 10 nt!IoInitSystemPreDrivers
- 11 nt!IoInitSystem
- 12 nt!Phase1Initialization
- 13 nt!PspSystemThreadStartup

CI: Code Integrity

HVCI: HYPERVISOR-BASED CODE INTEGRITY

Vsl: Virtual Secure Library ?

内核函数

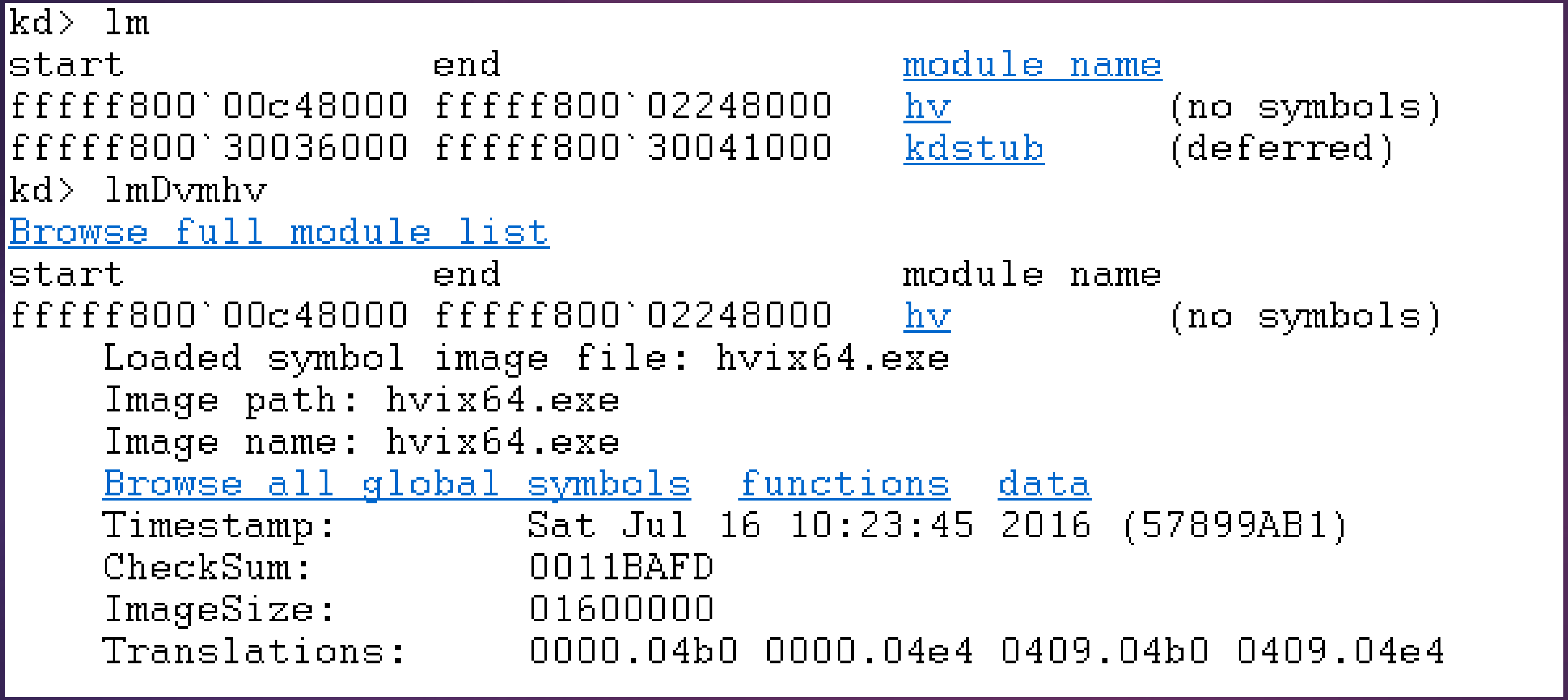
- 0: kd> x nt!??Ium*
- fffff803`6e744950 nt!PsIumSystemDllEnd
- fffff803`6e744958 nt!PsIumSystemDllStart
- 0: kd> x nt!???Ium*
- fffff803`6eab8ea0 nt!PspIumGetSystemDllMappingInfo
- fffff803`6eab9014 nt!PspIumInitializeNlsFiles
- fffff803`6e9b3f70 nt!PspIumGetSystemData
- fffff803`6e653908 nt!PspIumAllocateKernelPage
- fffff803`6eab8e60 nt!PspIumGetProcessorInfo
- fffff803`6eab8bcc nt!PspIumGetApiSetAndNlsSectionInformation
- fffff803`6e6539ac nt!PspIumGetImageMappingInfo
- fffff803`6eab8b40 nt!PspIumAllocateUserPage
- fffff803`6eab8ca0 nt!PspIumGetPhysicalPage
- fffff803`6e65395c nt!PspIumFreeKernelPage
- fffff803`6eab8b8c nt!PspIumFreePhysicalPage

!dh 0xFFFFF80053200000

```
kd> !dh 0xFFFFF80053200000

File Type: EXECUTABLE IMAGE
FILE HEADER VALUES
  8664 machine (X64)
     B number of sections
578997A3 time date stamp Sat Jul 16 10:10:43 2016

     0 file pointer to symbol table
     0 number of symbols
    F0 size of optional header
    22 characteristics
         Executable
         App can handle >2gb addresses

OPTIONAL HEADER VALUES
   20B magic #
 14.00 linker version
 49C00 size of code
 2EE00 size of initialized data
     0 size of uninitialized data
  10C4 address of entry point
  1000 base of code
       ----- new -----
```

```
0000000140000000 image base
    1000 section alignment
     200 file alignment
       1 subsystem (Native)
   10.00 operating system version
   10.00 image version
   10.00 subsystem version
   7F000 size of image
     400 size of headers
   7D069 checksum
0000000000080000 size of stack reserve
0000000000002000 size of stack commit
0000000000100000 size of heap reserve
0000000000001000 size of heap commit
     160  DLL characteristics
            High entropy VA supported
            Dynamic base
            NX compatible
   52150 [    16E4] address [size] of Export Directory
   53834 [      50] address [size] of Import Directory
   7D000 [     410] address [size] of Resource Directory
   60000 [    2D9C] address [size] of Exception Directory
   6D200 [    2160] address [size] of Security Directory
   7E000 [     180] address [size] of Base Relocation Directory
   4D5D0 [      38] address [size] of Debug Directory
       0 [       0] address [size] of Description Directory
       0 [       0] address [size] of Special Directory
```

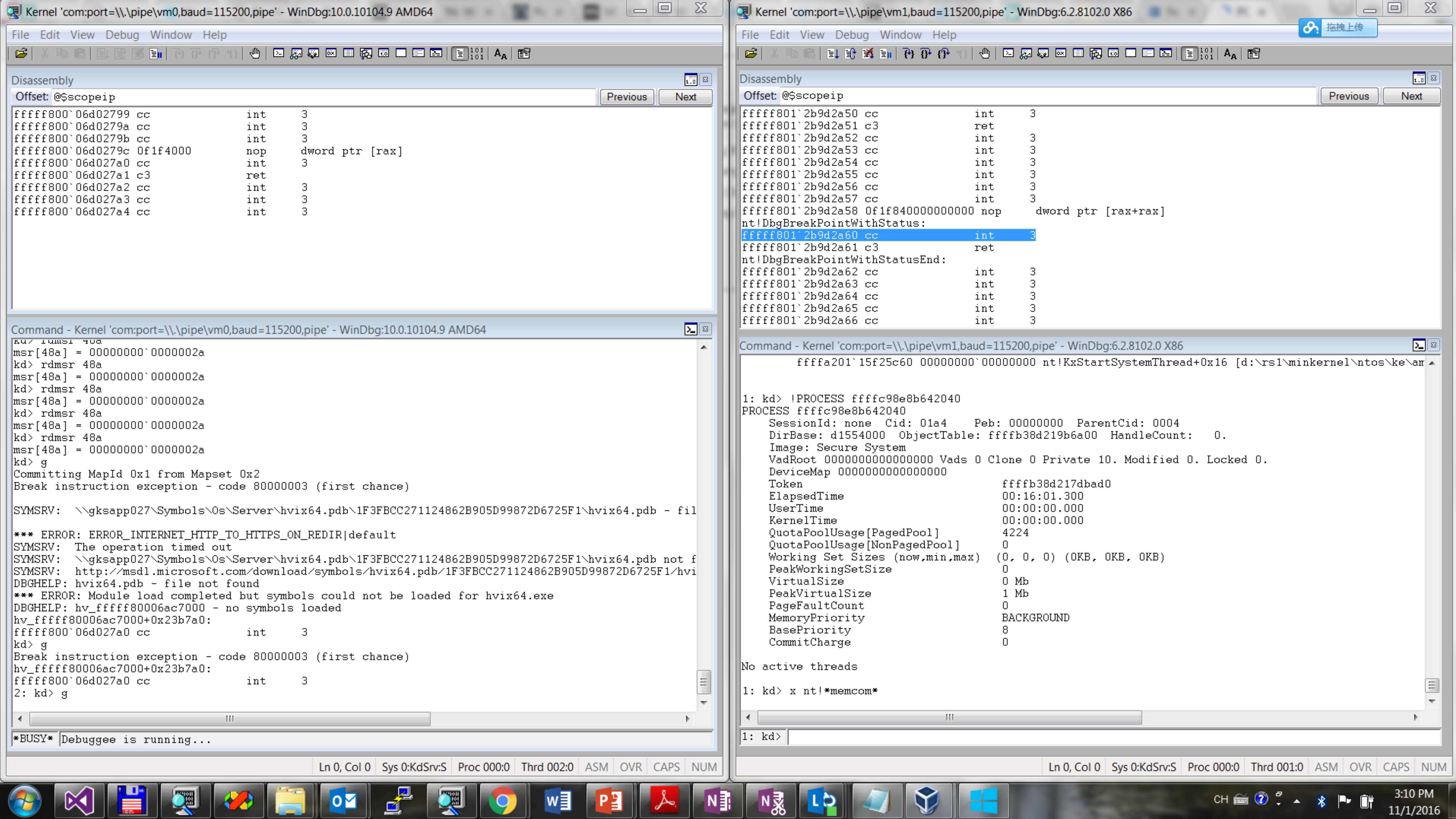
-1层的居民



```
kd> lm
start             end                 module name
fffff800`00c48000 fffff800`02248000   hv         (no symbols)
fffff800`30036000 fffff800`30041000   kdstub     (deferred)
kd> lmDvmhv
Browse full module list
start             end                 module name
fffff800`00c48000 fffff800`02248000   hv         (no symbols)
    Loaded symbol image file: hvix64.exe
    Image path: hvix64.exe
    Image name: hvix64.exe
    Browse all global symbols  functions  data
    Timestamp:        Sat Jul 16 10:23:45 2016 (57899AB1)
    CheckSum:         0011BAFD
    ImageSize:        01600000
    Translations:     0000.04b0 0000.04e4 0409.04b0 0409.04e4
```

调试之剑

- 目标端
- 调试VMM
  - bcdedit /hypervisorsettings serial DEBUGPORT:Port BAUDRATE:Baud
  - bcdedit /set hypervisordebug on
  - bcdedit /set hypervisorlaunchtype auto
- 调试Root Partition
  - bcdedit /set dbgtransport kdhvcom.dll
  - bcdedit /dbgsettings serial DEBUGPORT:Port BAUDRATE:Baud
  - bcdedit /debug on
- 主机端
- 启动vmdemux
  - vmdemux -src com:port=Port,baud=Baud
- 调试VMM
  - remote.exe /s "DbgPath\kd -k HVConnectionString -y SymPath" HyperV_HV
- 调试Root Partition
  - remote.exe /s “DbgPath\kd -k RPConnectionString -y SymPath” HyperV_ROOT


Kernel 'com:port=\\.\pipe\vm0,baud=115200,pipe' - WinDbg:10.0.10104.9 AMD64
File Edit View Debug Window Help
Disassembly
Offset: @$scopeip
Previous
Next

```
fffff800`06d02799 cc              int     3
fffff800`06d0279a cc              int     3
fffff800`06d0279b cc              int     3
fffff800`06d0279c 0f1f4000        nop     dword ptr [rax]
fffff800`06d027a0 cc              int     3
fffff800`06d027a1 c3              ret
fffff800`06d027a2 cc              int     3
fffff800`06d027a3 cc              int     3
fffff800`06d027a4 cc              int     3
```

Command - Kernel 'com:port=\\.\pipe\vm0,baud=115200,pipe' - WinDbg:10.0.10104.9 AMD64

```
kd> rdmsr 48a
msr[48a] = 00000000`0000002a
kd> rdmsr 48a
msr[48a] = 00000000`0000002a
kd> rdmsr 48a
msr[48a] = 00000000`0000002a
kd> rdmsr 48a
msr[48a] = 00000000`0000002a
kd> rdmsr 48a
msr[48a] = 00000000`0000002a
kd> g
Committing MapId 0x1 from Mapset 0x2
Break instruction exception - code 80000003 (first chance)

SYMSRV:  \\gksapp027\Symbols\Os\Server\hvix64.pdb\1F3FBCC271124862B905D99872D6725F1\hvix64.pdb - fil

*** ERROR: ERROR_INTERNET_HTTP_TO_HTTPS_ON_REDIR|default
SYMSRV:  The operation timed out
SYMSRV:  \\gksapp027\Symbols\Os\Server\hvix64.pdb\1F3FBCC271124862B905D99872D6725F1\hvix64.pdb not f
SYMSRV:  http://msdl.microsoft.com/download/symbols/hvix64.pdb/1F3FBCC271124862B905D99872D6725F1/hvi
DBGHELP: hvix64.pdb - file not found
*** ERROR: Module load completed but symbols could not be loaded for hvix64.exe
DBGHELP: hv_fffff80006ac7000 - no symbols loaded
hv_fffff80006ac7000+0x23b7a0:
fffff800`06d027a0 cc              int     3
kd> g
Break instruction exception - code 80000003 (first chance)
hv_fffff80006ac7000+0x23b7a0:
fffff800`06d027a0 cc              int     3
2: kd> g
```

BUSY Debuggee is running...
Ln 0, Col 0 Sys 0:KdSrv:S Proc 000:0 Thrd 002:0 ASM OVR CAPS NUM
Kernel 'com:port=\\.\pipe\vm1,baud=115200,pipe' - WinDbg:6.2.8102.0 X86
拖拽上传
File Edit View Debug Window Help
Disassembly
Offset: @$scopeip
Previous
Next

```
fffff801`2b9d2a50 cc              int     3
fffff801`2b9d2a51 c3              ret
fffff801`2b9d2a52 cc              int     3
fffff801`2b9d2a53 cc              int     3
fffff801`2b9d2a54 cc              int     3
fffff801`2b9d2a55 cc              int     3
fffff801`2b9d2a56 cc              int     3
fffff801`2b9d2a57 cc              int     3
fffff801`2b9d2a58 0f1f840000000000 nop     dword ptr [rax+rax]
nt!DbgBreakPointWithStatus:
fffff801`2b9d2a60 cc              int     3
fffff801`2b9d2a61 c3              ret
nt!DbgBreakPointWithStatusEnd:
fffff801`2b9d2a62 cc              int     3
fffff801`2b9d2a63 cc              int     3
fffff801`2b9d2a64 cc              int     3
fffff801`2b9d2a65 cc              int     3
fffff801`2b9d2a66 cc              int     3
```

Command - Kernel 'com:port=\\.\pipe\vm1,baud=115200,pipe' - WinDbg:6.2.8102.0 X86

```
        ffffa201`15f25c60 00000000`00000000 nt!KxStartSystemThread+0x16 [d:\rs1\minkernel\ntos\ke\am


1: kd> !PROCESS ffffc98e8b642040
PROCESS ffffc98e8b642040
    SessionId: none  Cid: 01a4    Peb: 00000000  ParentCid: 0004
    DirBase: d1554000  ObjectTable: ffffb38d219b6a00  HandleCount:   0.
    Image: Secure System
    VadRoot 0000000000000000 Vads 0 Clone 0 Private 10. Modified 0. Locked 0.
    DeviceMap 0000000000000000
    Token                             ffffb38d217dbad0
    ElapsedTime                       00:16:01.300
    UserTime                          00:00:00.000
    KernelTime                        00:00:00.000
    QuotaPoolUsage[PagedPool]         4224
    QuotaPoolUsage[NonPagedPool]      0
    Working Set Sizes (now,min,max)  (0, 0, 0) (0KB, 0KB, 0KB)
    PeakWorkingSetSize                0
    VirtualSize                       0 Mb
    PeakVirtualSize                   1 Mb
    PageFaultCount                    0
    MemoryPriority                    BACKGROUND
    BasePriority                      8
    CommitCharge                      0

No active threads

1: kd> x nt!*memcom*
```

1: kd>
Ln 0, Col 0 Sys 0:KdSrv:S Proc 000:0 Thrd 001:0 ASM OVR CAPS NUM
CH
3:10 PM
11/1/2016

“吾黨之小子狂簡，斐然成章，不知所以裁之。”

《论语 公冶长》

IUM是NT内核历史上最大的架构变化，我不清楚实现这个功能花多少时间，但很清楚消化和调试这一个变化所带来的问题需要更多的时间。格蠹老雷

切问而近思

欢迎关注格友公众号