Code Engn
www.CodeEngn.com

Hook the Planet

July 21, **2007**

Sammuel koo
dual5651@hotmail.com

Null @ Root

Agenda

1) What is a kernel hook used for?
2) Get into the Ring0!
3) Review
4) Hook the planet!
5) Hook me, If you can!
6) ?

What is a kernel hook used for?

Code Engn
www.CodeEngn.com

1) Kernel hooks are global (relatively speaking).

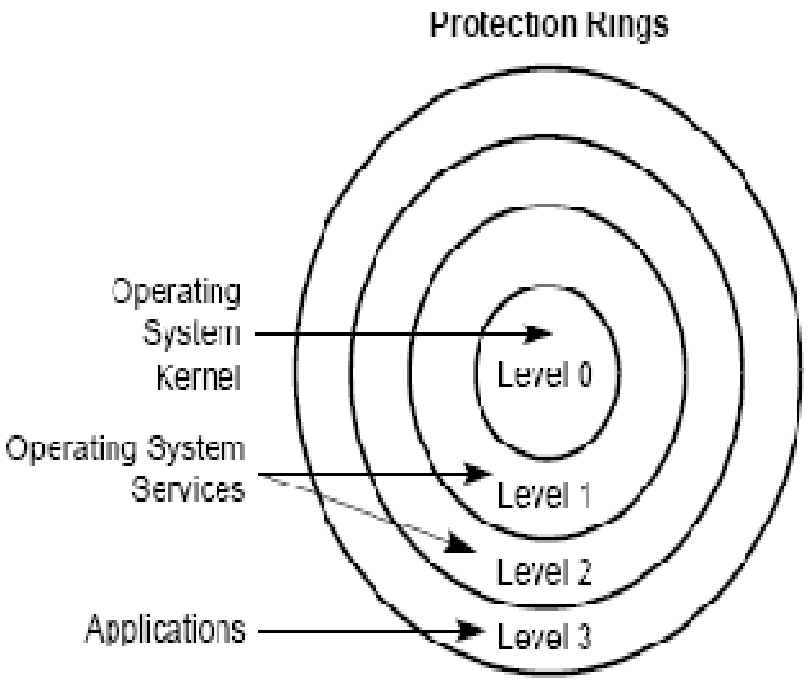
2) Rootkit / Protection / Detection software are both in Ring Zero.

3) By using kernel hook, We can study the behavior of the system.

4) By using kernel hook, we can get performance data for specific tasks and generating statics.

Get into the Ring0!

CodeEngn
www.CodeEngn.com



How to make code run on Ring0?

- **Call gate**
: CPL change 3 from to 0

- **Software Interrupt**
: API Call, Exception, ..

- **Device Driver**
: It doesn`t mean Usermode driver but Kernelmode drvier, because Kernelmode driver run on Level0, however, Usermode driver run on Level3.

Get into the Ring0!

Code Engn
www.CodeEngn.com

- Code of exe file is located into Paged Memory,
  In contrast, Code of sys file is located into NonPaged Memory.

- Both exe file and sys file has PE(Portable Executable) file format.

Header of 1stDriver.sys :

EntryPoint:	00001000	Subsystem:	0001
ImageBase:	00010000	NumberOfSections:	0006
SizeOfImage:	00007000	TimeDateStamp:	466BAB7E
BaseOfCode:	00001000	SizeOfHeaders:	00000400
BaseOfData:	00002000	Characteristics:	010E
SectionAlignment:	00001000	Checksum:	0000685F
FileAlignment:	00000200	SizeOfOptionalHeader:	00E0
Magic:	010B	NumOfRvaAndSizes:	00000010

Header of cmd.exe :

EntryPoint:	00005056	Subsystem:	0003
ImageBase:	4AD00000	NumberOfSections:	0003
SizeOfImage:	00078000	TimeDateStamp:	41107EBE
BaseOfCode:	00001000	SizeOfHeaders:	00000400
BaseOfData:	0001F000	Characteristics:	010F
SectionAlignment:	00001000	Checksum:	0007DCF6
FileAlignment:	00000200	SizeOfOptionalHeader:	00E0
Magic:	010B	NumOfRvaAndSizes:	00000010

- DPL of CS in sys file has value 0, however, DPL of CS in exe file has value 3. It means CS of sys file can access all of instructions and memorys, In contrast, CS of exe file has limited access right.

Get into the Ring0!

CodeEngn
www.CodeEngn.com

1. Compile C code by *DDK*

```
NTSTATUS DriverEntry (
            IN PDRIVER_OBJECT pDriverObject,
            IN PUNICODE_STRING pRegistryPath     ) {

    NTSTATUS status;
    int i;
    PDEVICE_OBJECT Device_Object = NULL;
    UNICODE_STRING Device_Name;
    UNICODE_STRING  Win32NameString;
    UNICODE_STRING uFileName,uFileName2;
    ANSI_STRING logNameString,logNameString2;
    OBJECT_ATTRIBUTES obj_attrib,obj_attrib2;
    IO_STATUS_BLOCK file_status,file_status2;
    IO_STATUS_BLOCK io_status,io_status2;
    LARGE_INTEGER timeout;

    RtlInitUnicodeString(&Device_Name,WIN_DEVICE_NA
    status = IoCreateDevice(pDriverObject,       //C
                                    0,
```

```
2007-01-31  오전 08:29          66,878 KFCI.zip
2003-02-04  오전 10:23             267 MAKEFILE
2007-01-31  오전 08:22    <DIR>        obj
2007-01-31  오전 08:22    <DIR>        objchk
2007-01-31  오전 08:48    <DIR>        objfre
2007-01-31  오전 08:26             67 SOURCES
              7개 파일         115,496 바이트
              5개 디렉터리  5,756,407,808 바이트 남음

C:₩saruengang>build -Cez
BUILD: Object root set to: ==> objchk
BUILD: Adding /Y to COPYCMD so xcopy eps won't hang.
BUILD: /i switch ignored
BUILD: Compile and Link for i386
BUILD: Examining c:₩saruengang directory for files to compile.
BUILD: Compiling c:₩saruengang directory
Compiling - kfci.c for i386
BUILD: Linking c:₩saruengang directory
Linking Executable - objchk₩i386₩kfci.sys for i386
BUILD: Done

    2 files compiled
    1 executable built

C:₩saruengang>
```

2. Assemble Asm code by *KmdKit*

```
DriverEntry proc pDriverObject:PDRIVER_OBJECT, pusRegi

  invoke DbgPrint, $CTA0("\nFindShadowTable: Entering

  mov eax, KeServiceDescriptorTable
  mov eax, [eax]
  invoke DbgPrint, $CTA0("FindShadowTable: ServiceDesc

  invoke GetServiceDescriptorTableShadowAddress
  .if eax != NULL
    invoke DbgPrint, $CTA0("FindShadowTable: ServiceDe
  .endif

  invoke DbgPrint, $CTA0("FindShadowTable: Leaving Dri
```

```
C:₩Documents and Settings₩구사무엘₩바탕 화면₩KmdKit₩KmdKit₩examples₩simple₩GetKe
rnelBase>goto make

C:₩Documents and Settings₩구사무엘₩바탕 화면₩KmdKit₩KmdKit₩examples₩simple₩GetKe
rnelBase>set drv-GetKernelBase

C:₩Documents and Settings₩구사무엘₩바탕 화면₩KmdKit₩KmdKit₩examples₩simple₩GetKe
rnelBase>₩masm32₩bin₩ml /nologo /c /coff GetKernelBase.bat
 Assembling: GetKernelBase.bat

C:₩Documents and Settings₩구사무엘₩바탕 화면₩KmdKit₩KmdKit₩examples₩simple₩GetKe
rnelBase>₩masm32₩bin₩link /nologo /driver /base:0x10000 /align:32 /out:GetKernel
Base.sys /subsystem:native GetKernelBase.obj

C:₩Documents and Settings₩구사무엘₩바탕 화면₩KmdKit₩KmdKit₩examples₩simple₩GetKe
rnelBase>del GetKernelBase.obj

C:₩Documents and Settings₩구사무엘₩바탕 화면₩KmdKit₩KmdKit₩examples₩simple₩GetKe
rnelBase>echo.


C:₩Documents and Settings₩구사무엘₩바탕 화면₩KmdKit₩KmdKit₩examples₩simple₩GetKe
rnelBase>pause
계속하려면 아무 키나 누르십시오 . . .
```

Get into the Ring0!

You can link DDK to Visual Studio.
If you use Visual Studio.net as IDE, You can use this way.

1. Download ddkbuild.bat from
   http://www.hollistech.com/Resources/ddkbuild/ddkbuild.htm

2. Copied it at c:₩progra~₩microsoft visu~₩vc7₩bin₩

3. Edit ddkbuild.bat
   set WNETBASE= C:₩WinDDK (Directory that DDK installed)
4. Making Project
   1. Select makefile project.
   2. Build cmdline : ddkbuild -WNET checked .
   3. output / arrangement cmd skip
   4. Build cmdline : ddkbuild -WNET chekced . -cZ

Get into the Ringo!

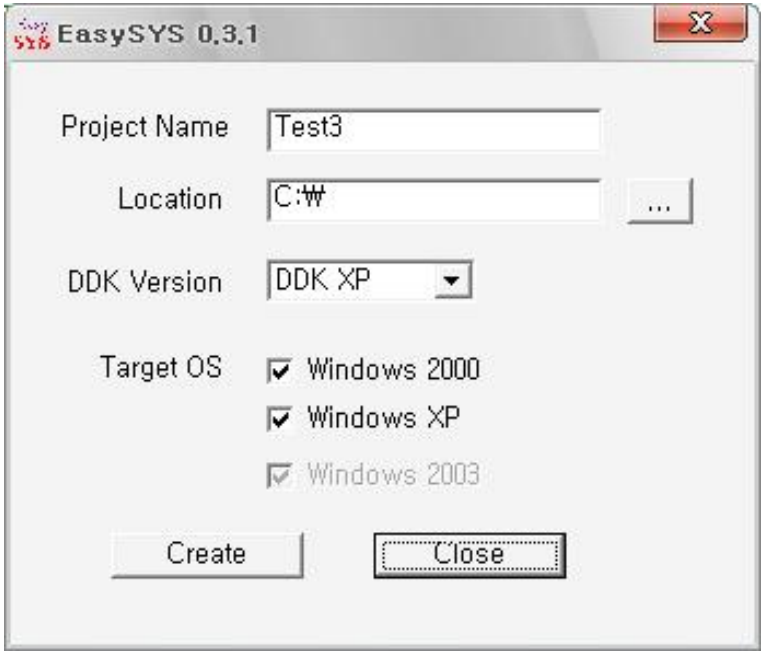
Code Engn
www.CodeEngn.com

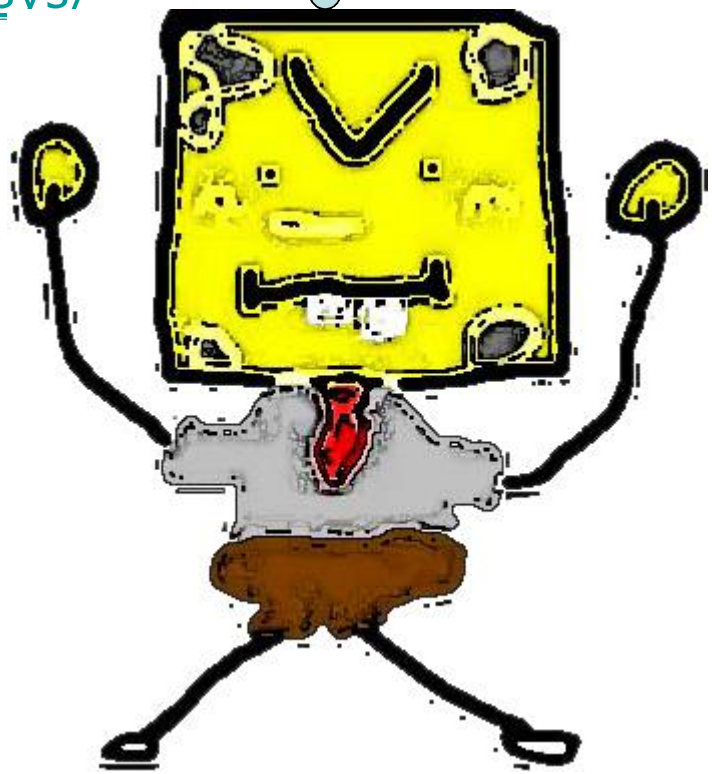
If you use Visual Studio 6.0 as IDE, You can use this way

It`s
So easy!!

1) Download easysys from this url :
   http://sourceforge.net/projects/easysvs/

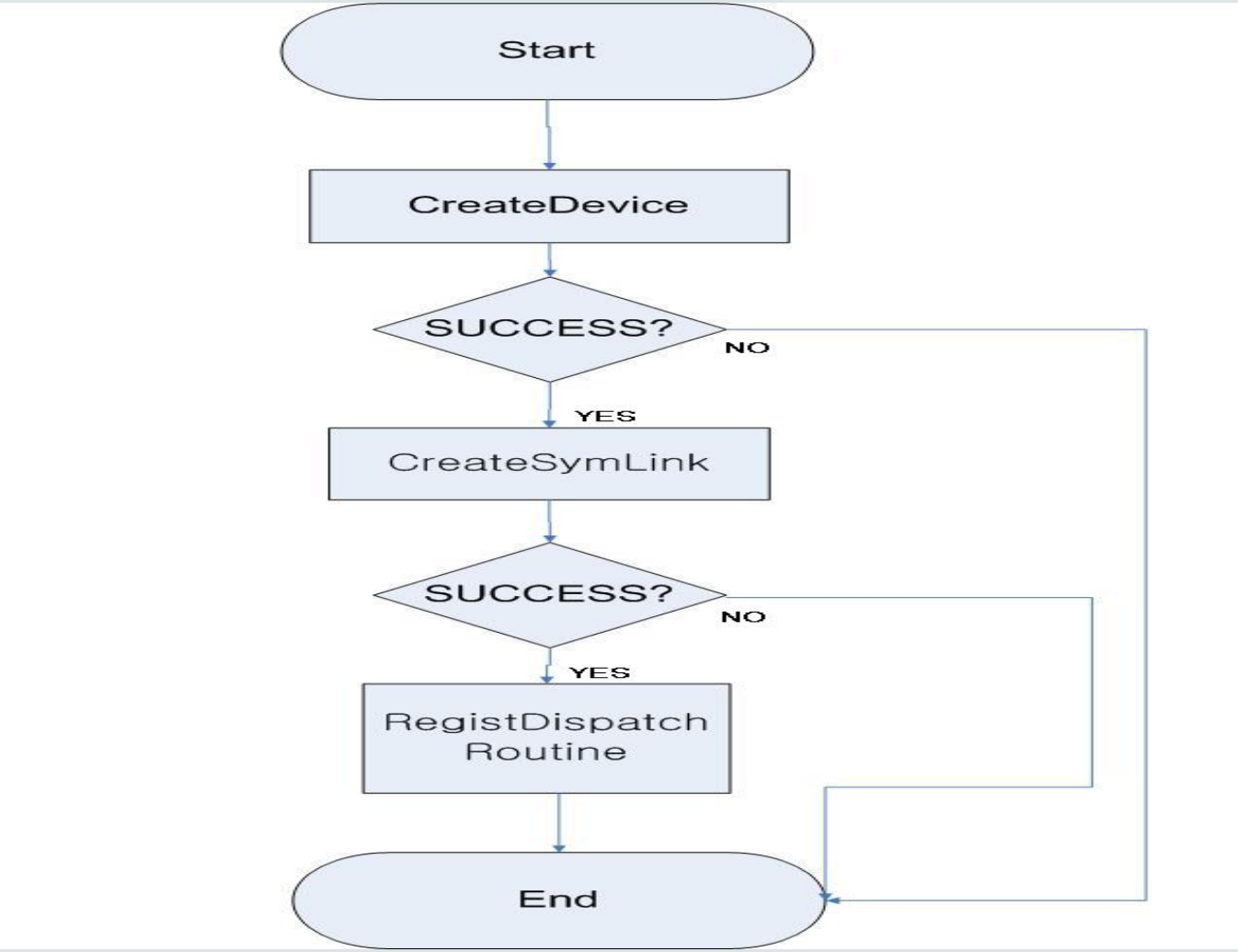
2) Easysys





Get into the Ring0!

Simple Device Driver :



Get into the Ring0!

CodeEngn
www.CodeEngn.com

1. Using SCM API

1) When a driver is loaded using the SCM, it is non-pageable.
This means your callback functions, IRP-handling functions, and other Important code will not vanish from Memory.
2) You can select start mode of Driver.

SERVICE_BOOT_START(0x0)
: Driver will be loaded by system loader.

SERVICE_SYSTEM_START(0x1)
: Load Driver when IoInitSystem is called.

SERVICE_AUTO_START(0x2)
: Load by SCM

SERVICE_DEMAND_START(0x3)
: Load by calling StartService() API.

SERVICE_DISABLED(0x4)
: Makes driver can not be loaded.

2. Using Undocumented API

Pro :
By using this way, you can load a Driver into the kernel without having to create registry key.

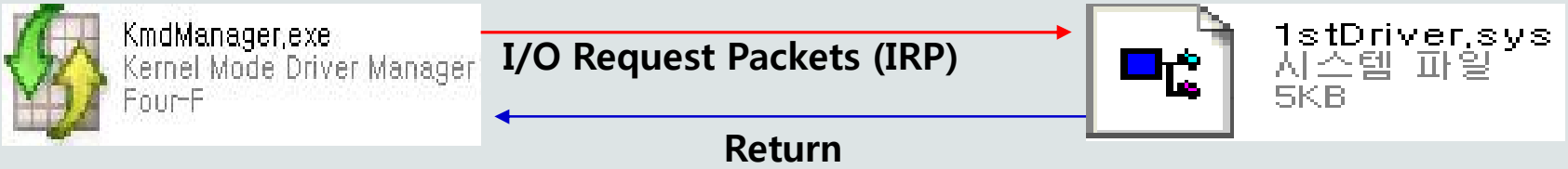
Con :
1) The problem with this approach is That the driver will be pageable. Sometimes when memory is paged out, it cannot be accessed; It will occur **BSOD**(Blue Screen Of Death) with system crash.
2) Once it is loaded, it cannot be Unloaded until reboot.

#44u6115f

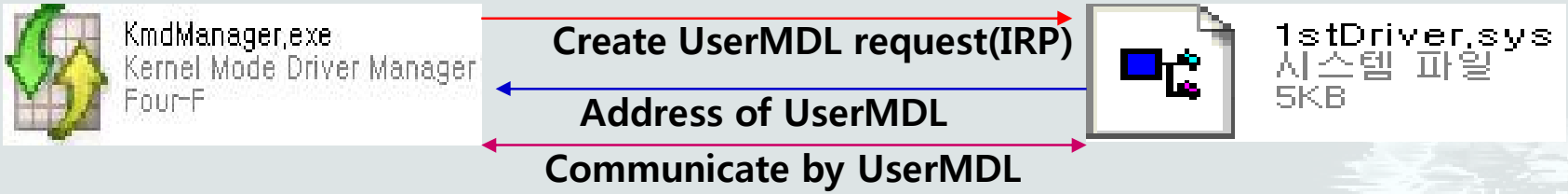
Get into the Ring0!

CodeEngn
www.CodeEngn.com

□ Communicate by IRP



□ Communicate by IRP & UserMDL



□ Communicate by IRP & Event

#44u6115f

Review

Code Engn
www.CodeEngn.com

Review

CodeEngn
www.CodeEngn.com

1) KPCR (Kernel`s Processor Control Region)

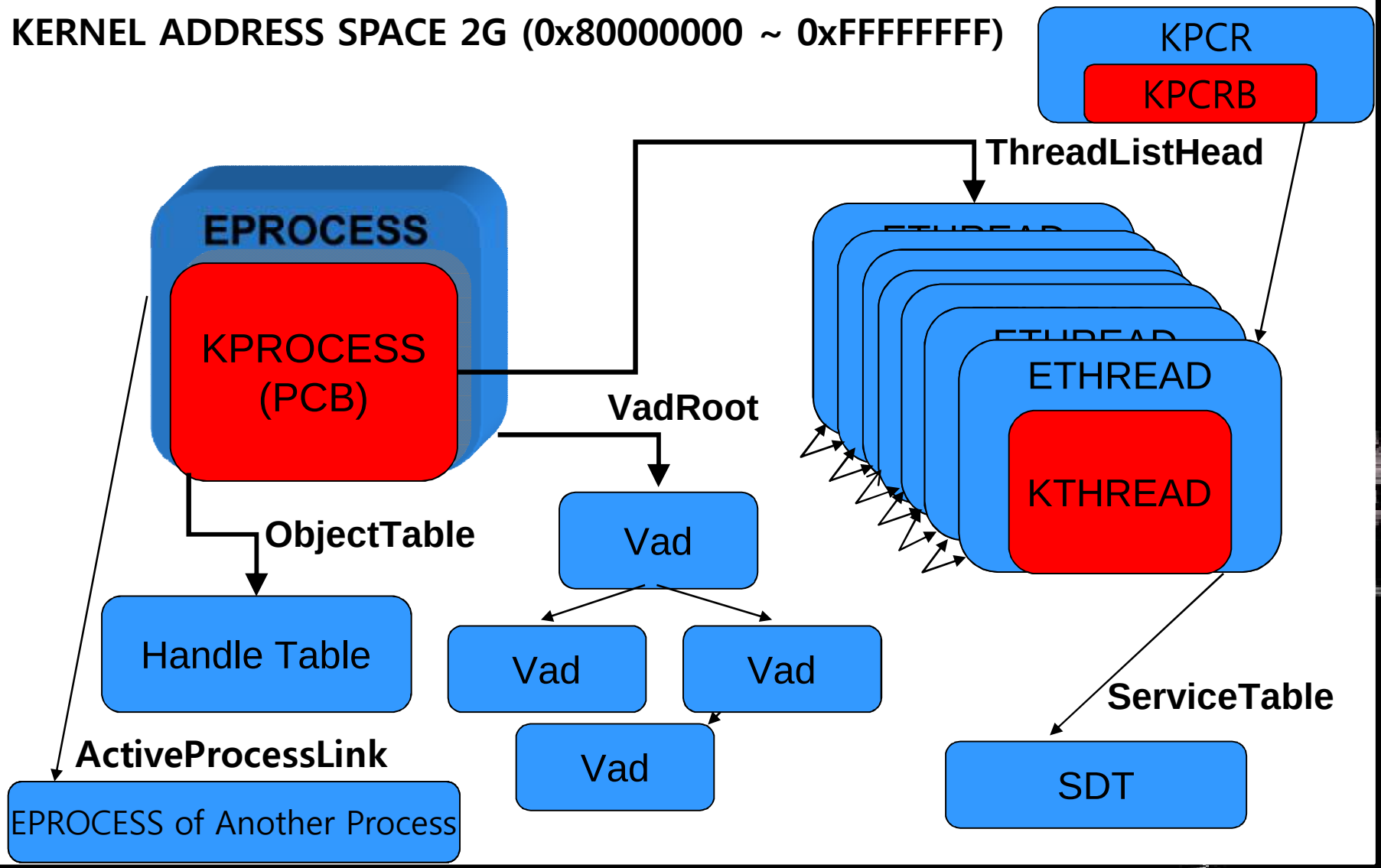
2) EPROCESS (Executable Process)

3) TEB (Thread Environment Block)

4) ETHREAD (Executable Thread)

Review

Code Engn
www.CodeEngn.com



Review

CodeEngn
www.CodeEngn.com

KPCR (Kernel`s Processor Control Region)

In Kernel Level, FS:0 = KPCR = 0xFFDFF000

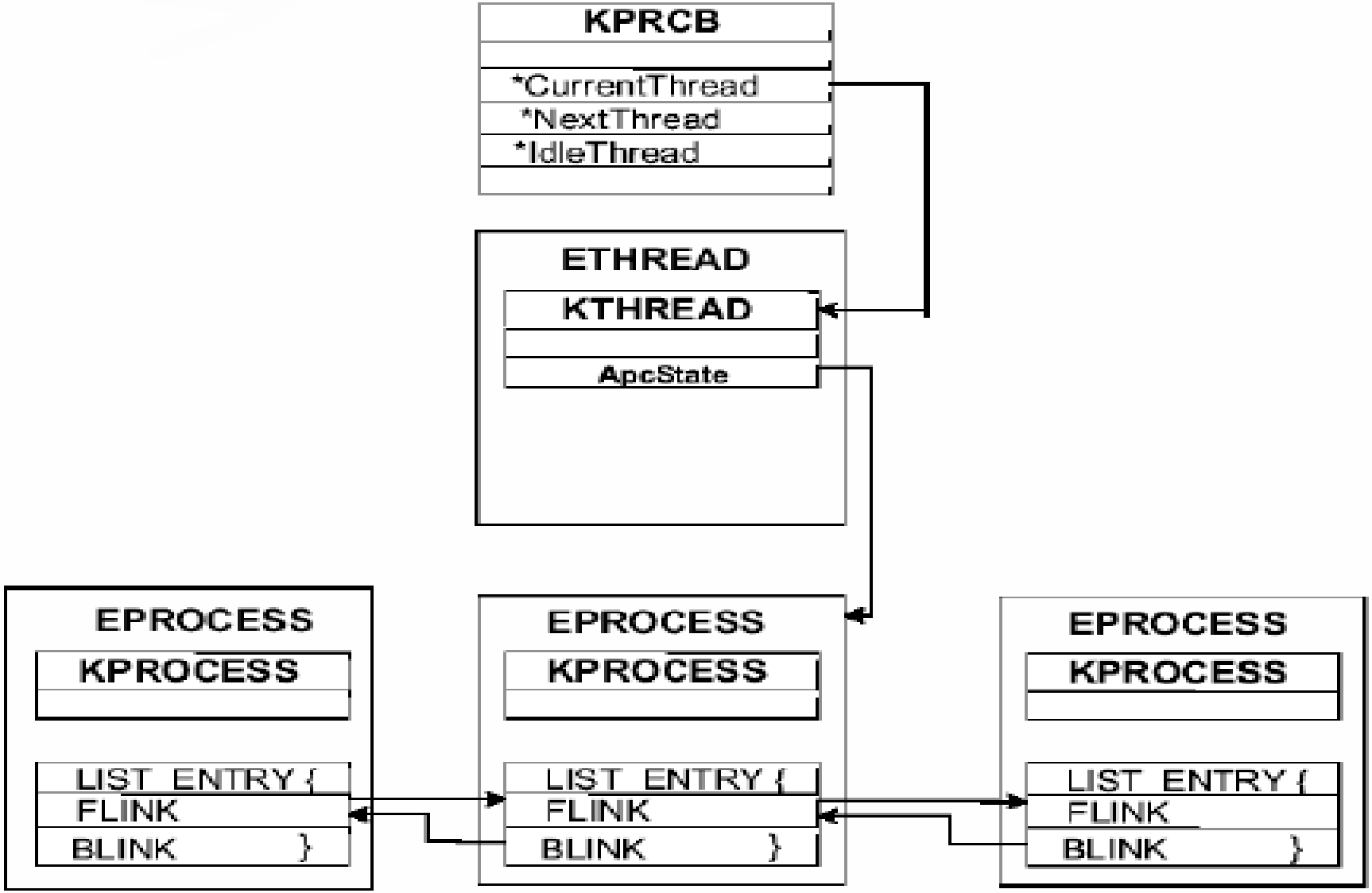
NtTib
SelfPcr
Prcb
Irql
IRR
IrrActive
IDR
Reserved2
IDT
GDT
TSS
MajorVersion
MinorVersion
SelfMember
StallScaleFactor
DebugActive
Number

ExceptionList
StackBase
StackLimit
SubSystemTib
ArbitraryUserPointer
SelfTib

MinorVersion
MajorVersion
CurrentThread
NextThread
IdleThread
Reserved
BuildType
SetMember
RestartBlock

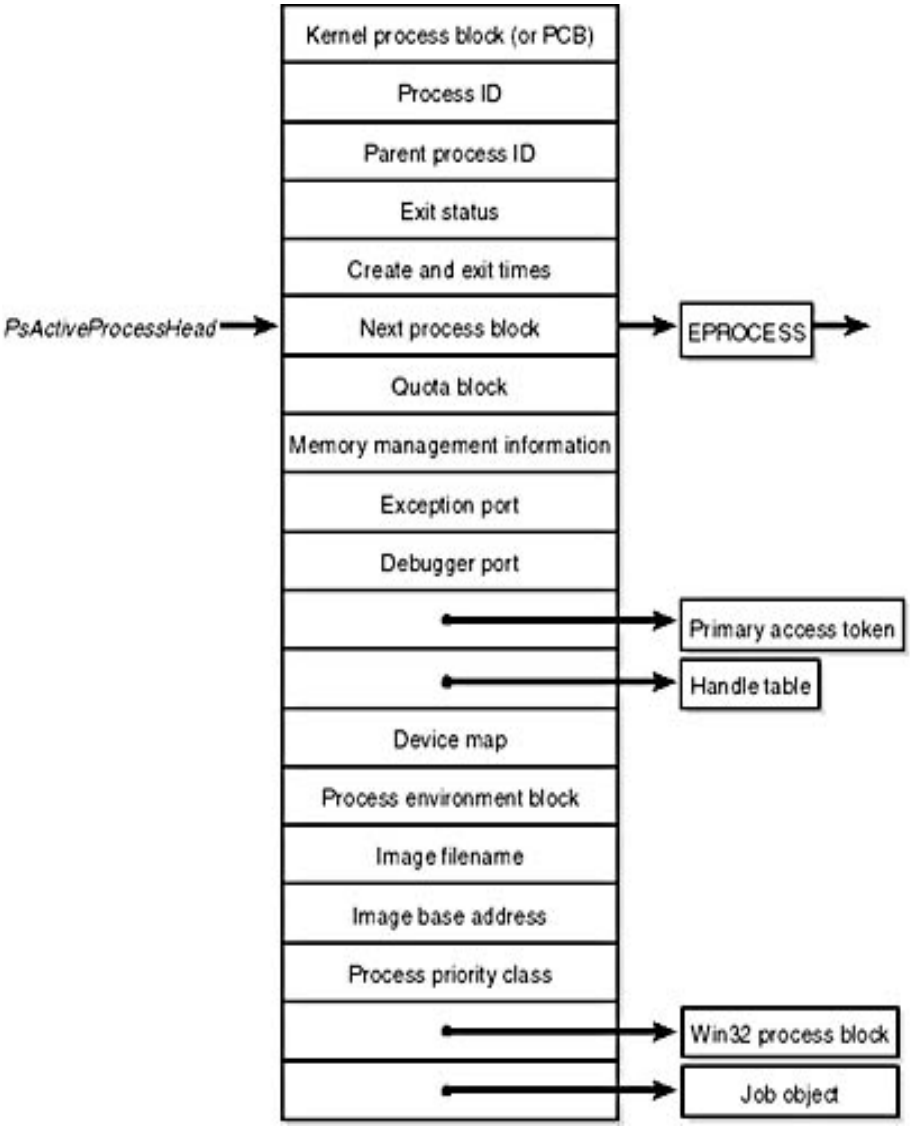
Review

Code Engn
www.CodeEngn.com



Review

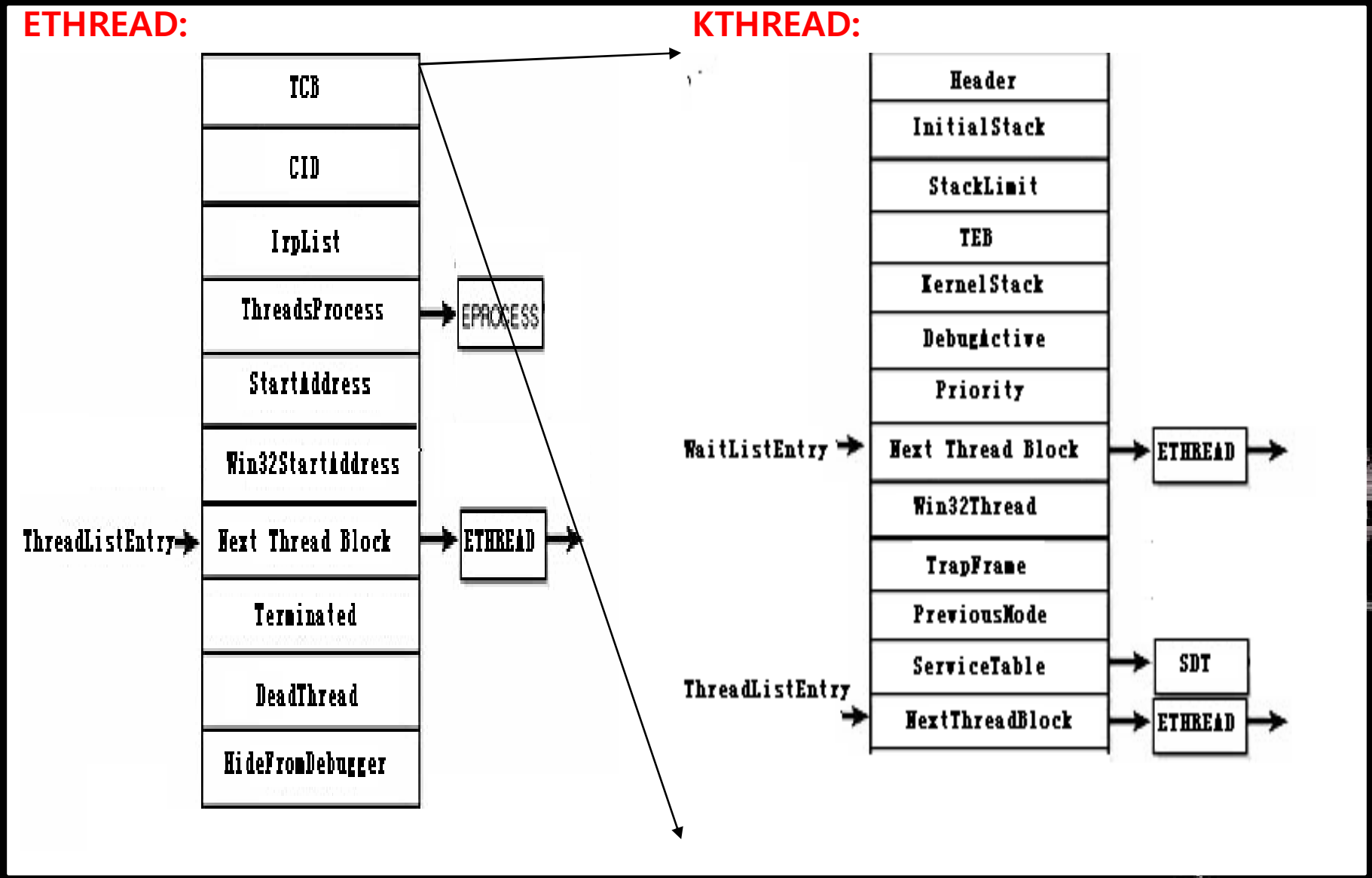
CodeEngn
www.CodeEngn.com



- In Windows Kernel, Each process is represented as an EPROCESS into Kernel Memory.
- List of active processes is obtained by traversing a doubly linked list referenced in the EPROCESS Structure of each process.
- We can access EPROCESS by CurrentThread of KPRCB structure.
- The Windows scheduling algorithm is not executed at process.

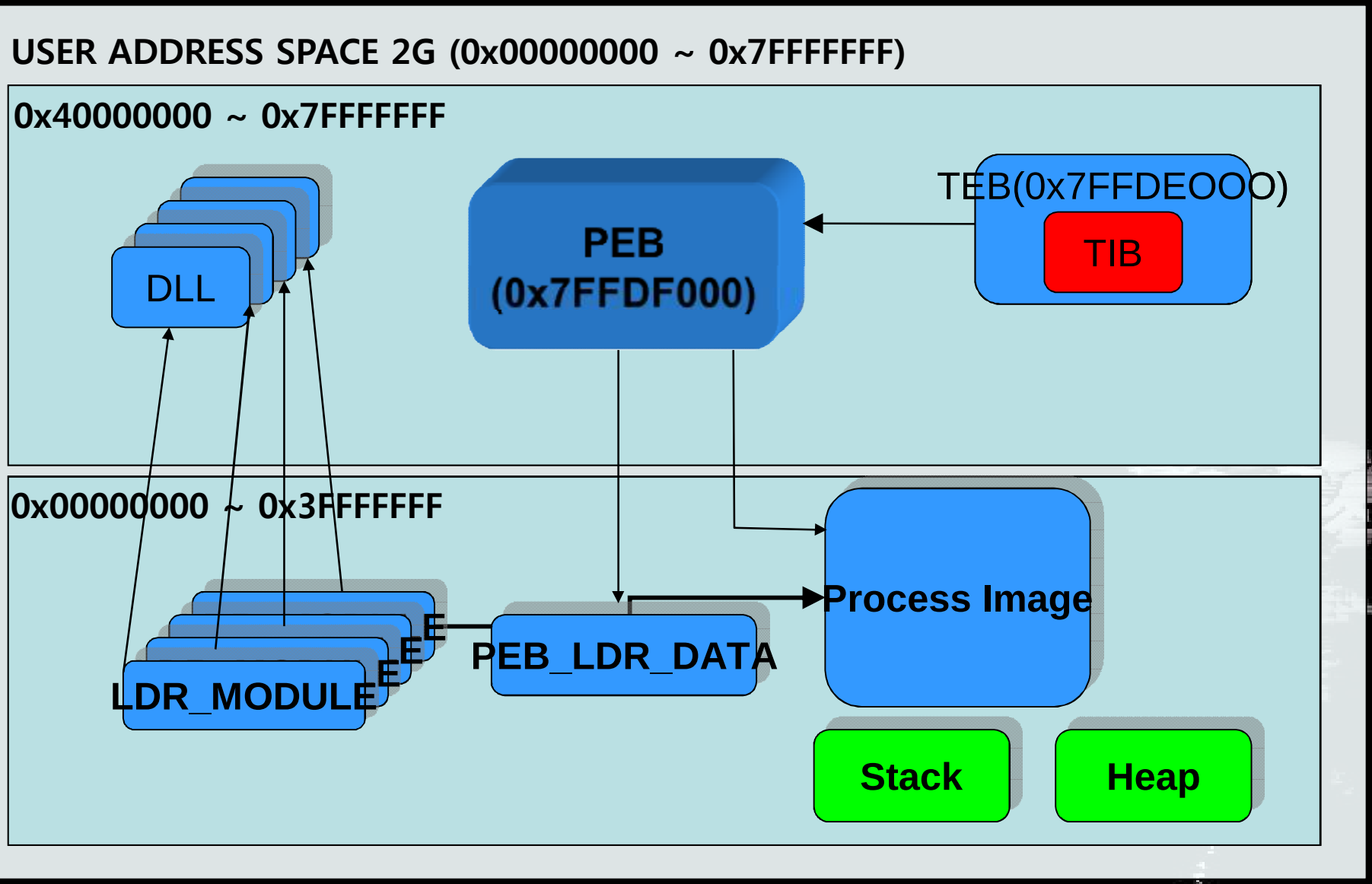
Review

Code Engn
www.CodeEngn.com



Review

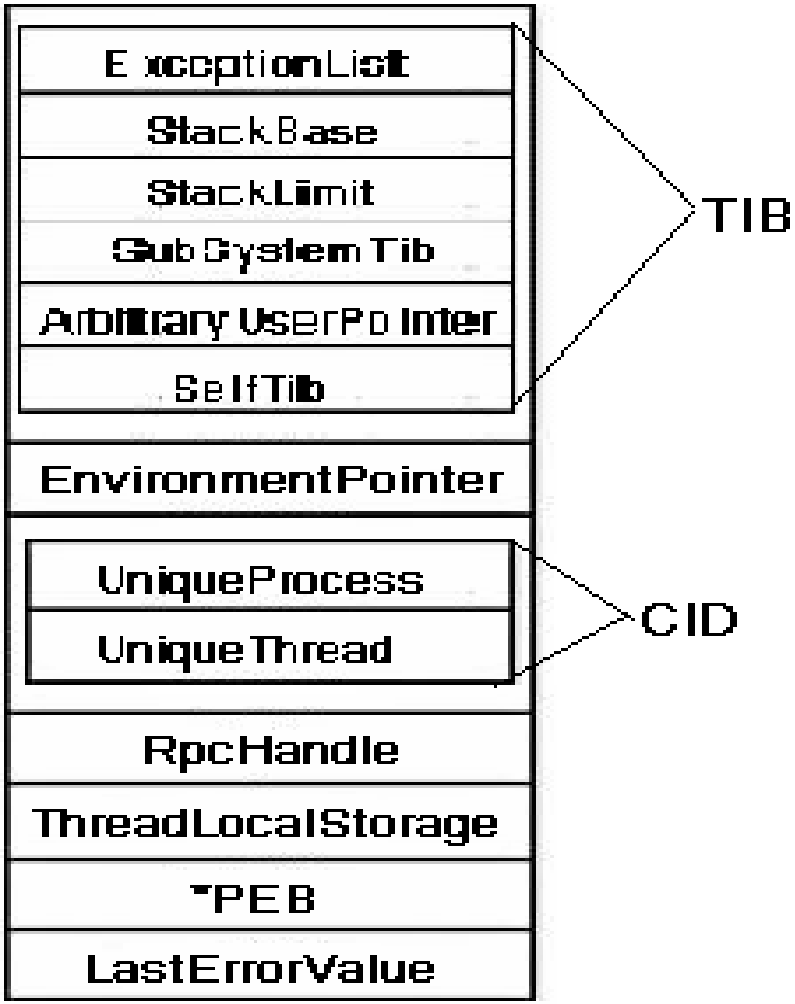
Code Engn
www.CodeEngn.com



Review

CodeEngn
www.CodeEngn.com

TEB (Thread Environment Block)



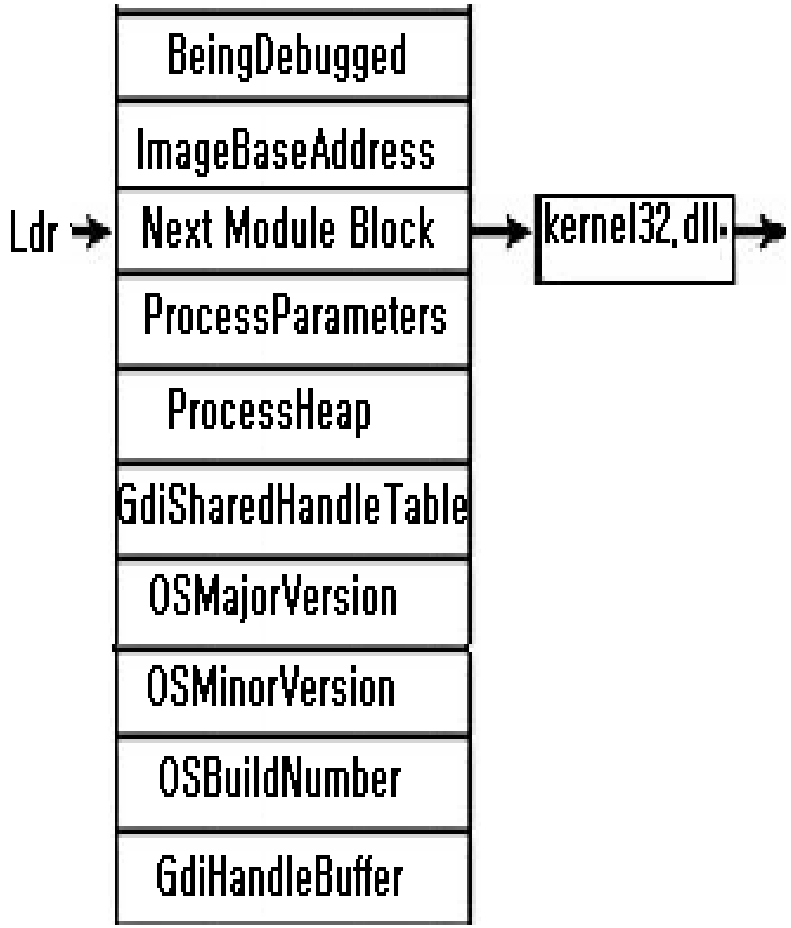
In UserLevel, FS:0 = TEB = 0x7ffde000 (1st Process)

- ☐ UserMode Application can access this structure directly.
- ☐ GetCurrentProcessId() function use this structure for getting PID.
- ☐ We can get PEB address from this structure.

Review

CodeEngn
www.CodeEngn.com

PEB (Process Environment Block)



- ☐ UserMode Application can access this structure directly.
- ☐ This structure have loaded modules list.
- ☐ This structure have ProcessParameters information.
- ☐ IsDebuggerPresent() function use this structure.

Hook the planet!

Code Engn
www.CodeEngn.com

Hook the planet!

CodeEngn
www.CodeEngn.com

1) Win32 UserLevel API Global Hooking

2) SSDT(System Service Distpach Table) hooking

3) IDT(Interrupt Descriptor Table) hooking

4) 0ne byte hooking

5) Blind hooking by using DRx

Hook the planet!

Win32 UserLevel API Global Hooking Motivation

- Is there no way to Dll Injection without read/write process memory, and CreateRemoteThread() API?
- Is there no way to hook Win32 API globaly?

Hook the planet!

Concept

- In Windows, DLL memory and mapped memory are shared.
- If we can modify memory of DLL without PageFault(Copy-On-Write), It will be applied to all processes.
- If we have chance to execute our code by victim process self, We don`t need CreateRemoteThread() anymore.
- In Windows, There is a region that will get mapped into every process address space, The name of this area is KUSER_SHARED_DATA.

Hook the planet!

Code Engn
www.CodeEngn.com

Process

1. Get the EPROCESS of explorer.exe
2. Attach to explorer.exe
3. Get the PEB from EPROCESS
4. Find Kernel32.dll address from Ldr in PEB.
5. Find IsDebuggerPresent() address from IMAGE_EXPORT_DIRECTORY

   of kernel32.dll
6. Modify IsDebuggerPresent() code to jmp KUSER_SHARED_DATA+0x8

   with CR0 trick.
7. Now just waiting for victim process call IsDebuggerPresent()

Hook the planet!

Code Engn
www.CodeEngn.com

Original IsDebuggerPresent() code:

```
MOV EAX,DWORD PTR FS:[18]
MOV EAX,DWORD PTR DS:[EAX+30]
MOVZX EAX,BYTE PTR DS:[EAX+2]
```

↓

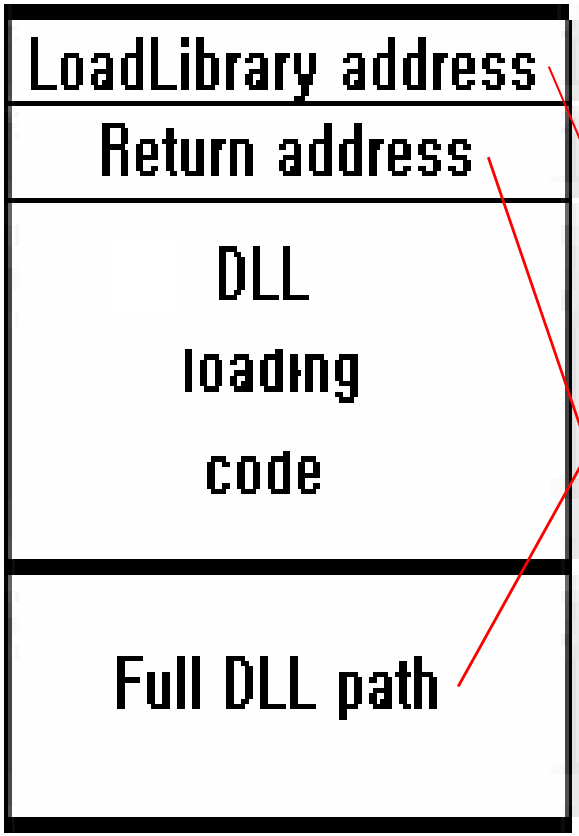
Modifed IsDebuggerPresent() code:

```
MOV EAX , KUSER_SHARED_DATA+0x8
JMP EAX
```

Hook the planet!

Code Engn
www.CodeEngn.com

PAYLOAD:



DLL loading code :

```
pushad
pushfd
push KSD + 0x8 + sizeof(DLL load
 code)
call [KUSER_SHARED_DATA]
popfd
popad
jmp [KUSER_SHARED_DATA+0x4]
```

Hook the planet!

Demonstration

Hook the planet!

SSDT(System Service Distpach Table) hooking Motivation

- ☐ Is there no way to hook Native API as easier as IAT hook?
- ☐ Is there no way to hook Native API globaly?

Hook the planet!

Concept

- Native system services`s addresses are listed in SSDT.
- KeServiceDescriptorTable is exported by the ntoskrnl.exe
- Windows XP later versions of the OS make the SSDT read-only, but we can bypass this protection with CR0 trick or MDL.
- We can get index number of Native API that we want to hook from ntdll.dll

Hook the planet!

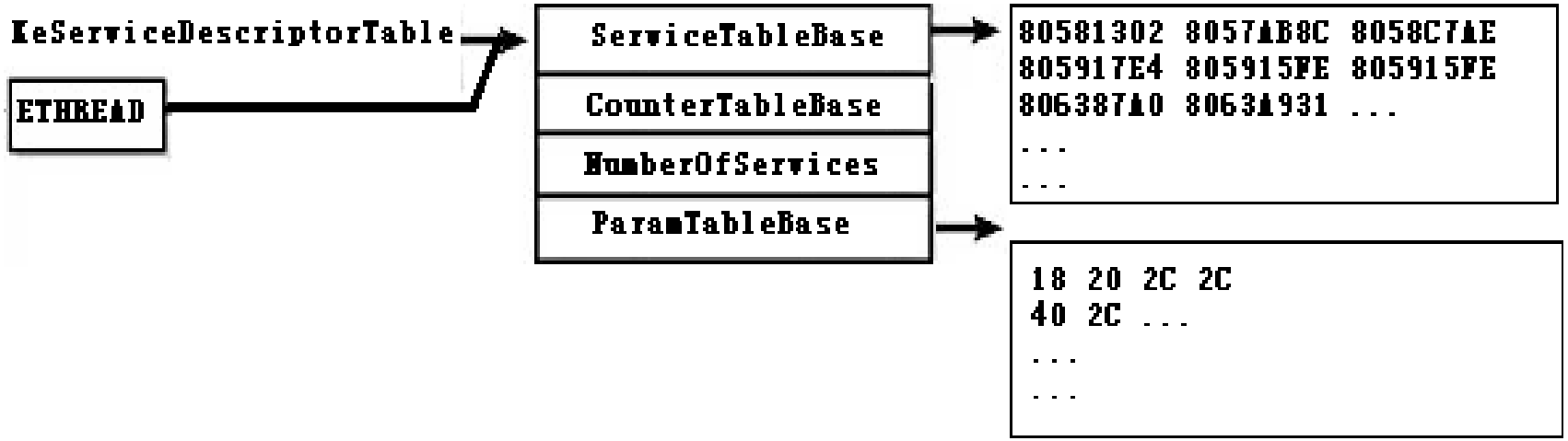
Code Engn
www.CodeEngn.com

Process

1. Build the function that has same prototype with Native API to hook.
2. Get the index number of Native API.
3. Get address of SSDT by referencing KeServiceDescriptorTable.
4. Make SSDT memory area writeable.
5. KeServiceDescriptorTable->ServiceTableBase[index] = HookFunction;

Hook the planet!

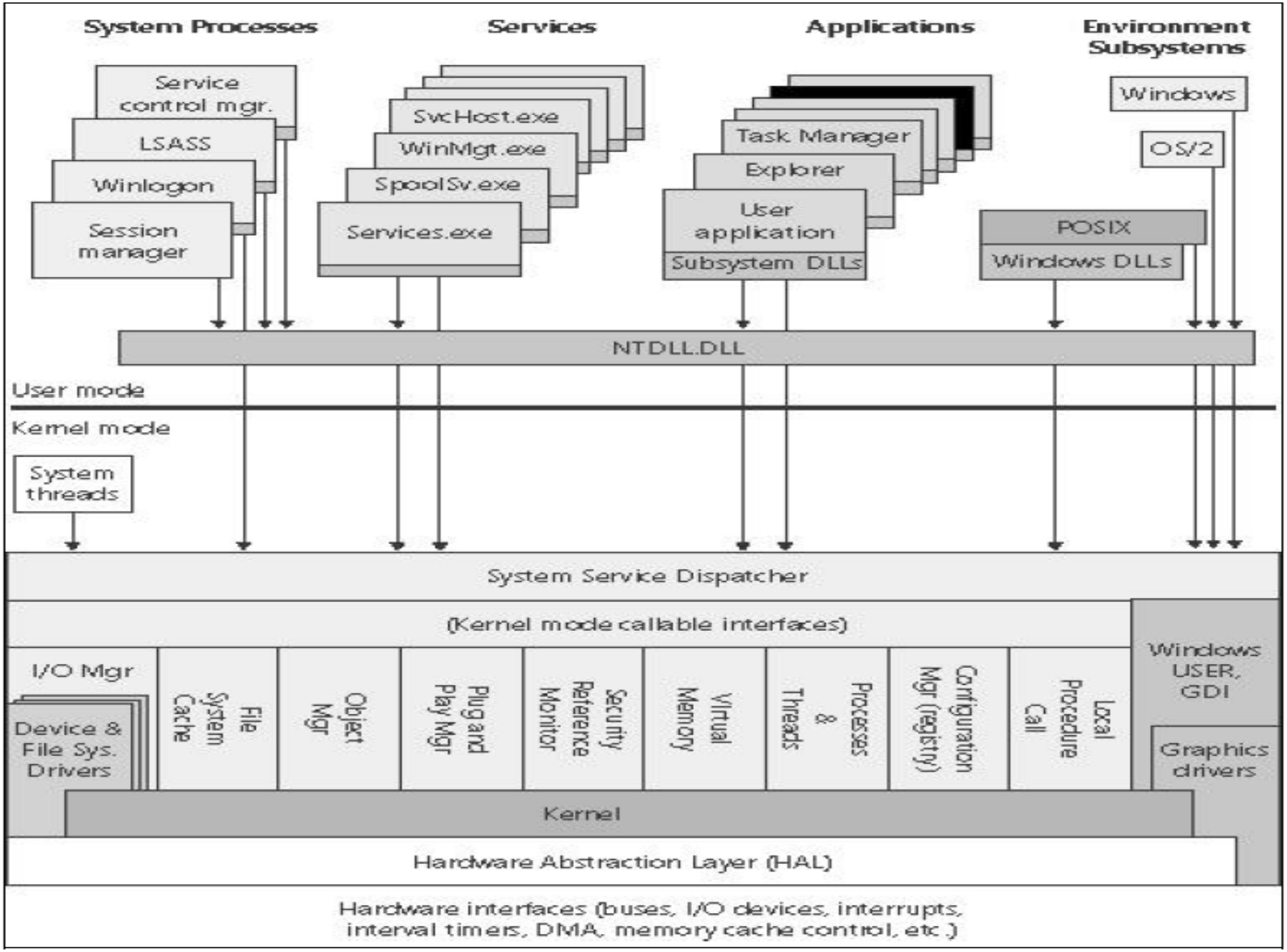
CodeEngn
www.CodeEngn.com



- The KeServiceDescriptorTable is a table exported by the kernel,
  This table contains the core system services implemented in **ntoskrnl.exe**.

- There is another table in Windows Kernel, called
  KeServiceDescriptorTableShadow,
  that contains the address of USER & GDI services implemented in
  win32k.sys.
  This table is not exported.

- ServiceTable pointer of ETHREAD is not always have same value with KSDT.


System Processes
Services
Applications
Environment Subsystems
Service control mgr.
LSASS
Winlogon
Session manager
SvcHost.exe
WinMgt.exe
SpoolSv.exe
Services.exe
Task Manager
Explorer
User application
Subsystem DLLs
Windows
OS/2
POSIX
Windows DLLs
NTDLL.DLL
User mode
Kernel mode
System threads
System Service Dispatcher
(Kernel mode callable interfaces)
I/O Mgr
Device & File Sys. Drivers
File System Cache
Object Mgr
Plug and Play Mgr
Security Reference Monitor
Virtual Memory
Processes & Threads
Configuration Mgr (registry)
Local Procedure Call
Windows USER, GDI
Graphics drivers
Kernel
Hardware Abstraction Layer (HAL)
Hardware interfaces (buses, I/O devices, interrupts, interval timers, DMA, memory cache control, etc.)

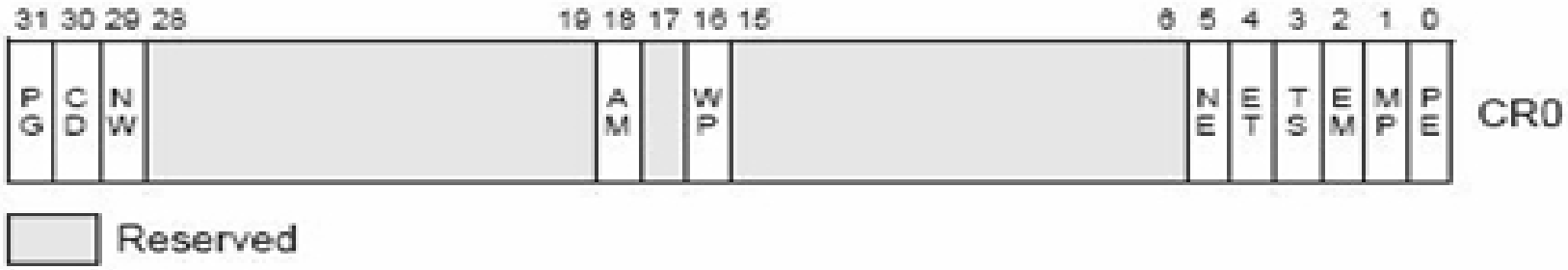
Hook the planet!

- A system service dispatch is triggered when an INT 2E or SYSENTER instruction is called.
- Lower version than Windows 2000, Windows use INT 2E for service dispatch, Higher than Windows XP, Windows use SYSENTER for service dispatch.
- Normally, System call from UserLevel through ntdll.dll, but direct system call is possible.
- In Windows XP, we can use INT 2E instruction.
- System call number is contained in the EAX register.

Hook the planet!

Code Engn
www.CodeEngn.com

What is the CR0 trick?



- The WP bit controls whether the processor will allow writes to memory pages marked as read-only.
- Setting WP(Write Protection) to zero disables memory protection.

Hook the planet!

CodeEngn
www.CodeEngn.com

What is the MDL?

MDL (Memory Descriptor List):

Next
Size
MdlFlags
Process
MappedSystemVa
StartVa
ByteCount
ByteOffset

1) Create the memory into our domain.
   - MmCreateMdl()

2) MDL build for NonPage
   - MmBuildMdlForNonPagedPool()

3) Change the flags of the MDL
   - MdlFlags |= MDL_MAPPED_TO_SYSTEM_VA

4) Lock that page
   - MmMapLockedPages()

Hook the planet!

CodeEngn
www.CodeEngn.com

How to get index number?

ZwWriteFile in Ntdll.dll :

```
7C93E9F3   B8 12010000   MOV EAX,112
7C93E9F8   BA 0003FE7F   MOV EDX,7FFE0300
7C93E9FD   FF12          CALL DWORD PTR DS:[EDX]
7C93E9FF   C2 2400       RETN 24
```

InvalidateRect in user32.dll :

```
77CFB5F5   B8 C2110000   MOV EAX,11C2
77CFB5FA   BA 0003FE7F   MOV EDX,7FFE0300
77CFB5FF   FF12          CALL DWORD PTR DS:[EDX]
77CFB601   C2 0C00       RETN 0C
```

Hook the planet!

Demonstration

Hook the planet!

IDT(Interrupt Descriptor Table) hooking Motivation

- ☐ IDT is used to handle interrupts, so there are many tasty things.
- ☐ IDT hooking is more powerful than SDT hooking.

Hook the planet!

Concept

- We can use sidt,lidt instruction for getting and saving IDT information.

Hook the planet!

Code Engn
www.CodeEngn.com

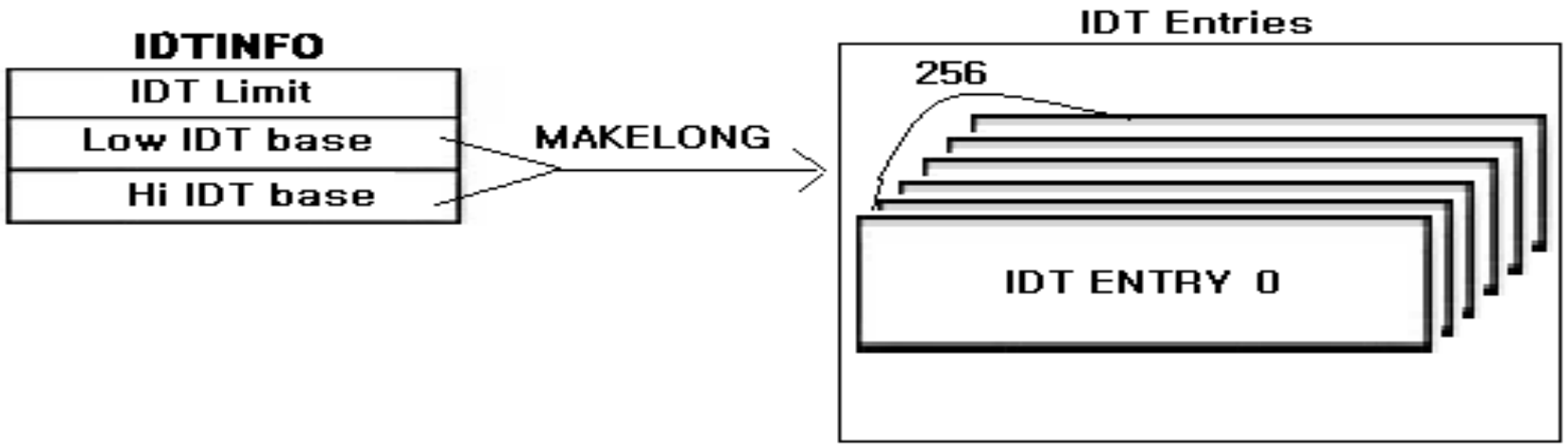
IDTENTRY

LowOffset
selector
unused_lo
Type
Always0
DPL
present bit
HiOffset

- Handler Address = MAKELONG(LowOffset,HiOffset);
- If DPL value of gate is 3, It can be called from both User and Kernel level.
- There are 3 types of gate.
  - Interrupt Gate (X 1 1 0)
  - Trap Gate (X 1 1 1)
  - Task Gate (0 1 0 1)

Hook the planet!

Code Engn
www.CodeEngn.com



- The SIDT instruction is used to find the IDT in memory,
  It returns the address of the IDTINFO structure.
  The LIDT instruction is used for saving information into IDT.

- The IDT specifies how to process interrupts such as those fired when a key pressed, when a page fault occurs.

- The Total number of IDT gates is 256.

Hook the planet!

Code Engn
www.CodeEngn.com

Selector :

```
#define KGDT_NULL        0
#define KGDT_R0_CODE     8
#define KGDT_R0_DATA     16
#define KGDT_R3_CODE     24
#define KGDT_R3_DATA     32
#define KGDT_TSS         40
#define KGDT_R0_PCR      48
#define KGDT_R3_TEB      56
#define KGDT_VDM_TILE    64
#define KGDT_LDT         72
#define KGDT_DF_TSS      80
#define KGDT_NMI_TSS     88
```

Hook the planet!

Code Engn
www.CodeEngn.com

There are so many interrupts :

Int.	DPL	P	Descriptor Type	Description	
0000	0	P	Interrupt Gate	Fault	Divide Error
0001	0	P	Interrupt Gate	Fault/Trap	Debug
0002	0	P	Task Gate	Interrupt	NMI Interrupt
0003	3	P	Interrupt Gate	Trap	Breakpoint
0004	3	P	Interrupt Gate	Trap	Overflow
0005	0	P	Interrupt Gate	Fault	BOUND Range Exceeded
0006	0	P	Interrupt Gate	Fault	Invalid Opcode (Undefined Opcode). Was intr
0007	0	P	Interrupt Gate	Fault	Device Not Available (No Math Coprocessor)
0008	0	P	Task Gate	Abort	Double Fault

Hook the planet!

Demonstration

Hook the planet!

0ne byte hooking Motivation

- I`m so lazy person to restore original codes of hooked function **J**
- Inline hooking is so static.

Hook the planet!

Concept

- Some functions in Windows XP later versions of the OS have *MOV EDI, EDI* *instruction.*
- *MOV EDI,EDI* doesn`t take the effect to code flow.
- *MOV EDI,EDI* (=0x8B,0xFF) -> *INT 0xFF* (=0xCD,0xFF)
- We can set handler at *INT 0xFF* *manually.*

Hook the planet!

Code Engn
www.CodeEngn.com

Process

1. Get address of function.
2. Hook IDT (INT 0xFF) for all processors.
3. Make memory region of function to hook writeable.
4. Overwrite MOV EDI,EDI with 0xCD, the 'int' opcode making INT 0xFF.

Hook the planet!

Demonstration

Hook the planet!

Blind hooking by using DRx Motivation

- ☐ I want to hook function without memory or table patching.
- ☐ Are there no way to hook dinamically?

Hook the planet!

Concept

- ☐ We can use DR0 ~ DR3 for set addresses to hook.
- ☐ We can set handler at *INT 0x01* (Debug Exception).

Hook the planet!

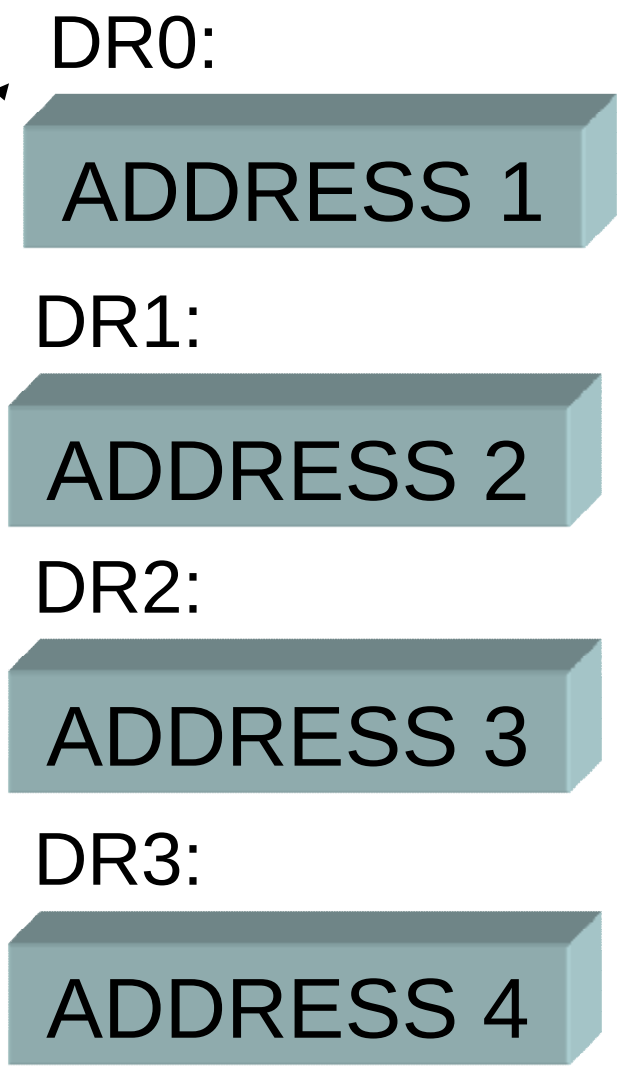
Process

1. Get address of function.
2. Hook IDT (INT 0x01) for all processors.
3. Set addresses at DR0 ~ DR3.
4. Set hook type(E/W/RW) at DR7.

Hook the planet!

CodeEngn
www.CodeEngn.com

```
typedef struct tagDebugReg7
    {
    unsigned L0 :1; //
    unsigned G0 :1; //
    unsigned L1 :1; //
    unsigned G1 :1; //
    unsigned L2 :1; //
    unsigned G2 :1; //
    unsigned L3 :1; //
    unsigned G3 :1; //
    unsigned GL :1; //
    unsigned GE :1; //
    unsigned undefined1 :3; // 001
    unsigned GD :1; //
    unsigned undefined2 :2; // 00
    unsigned RW0 :2;
    unsigned LEN0 :2;
    unsigned RW1 :2;
    unsigned LEN1 :2;
    unsigned RW2 :2;
    unsigned LEN2 :2;
    unsigned RW3 :2;
    unsigned LEN3 :2;
    } DebugReg7;
```



Hook the planet!

Code Engn
www.CodeEngn.com

Lx?

Gx?

GD?

RWx? 00(execute), 01(write), 11(read & write).

LENx? 00(byte), 01(word), 11(dword).

Hook the planet!

Code Engn
www.CodeEngn.com

```
typedef struct
    DebugReg6
{
unsigned B0 :1;
unsigned B1 :1;
unsigned B2 :1;
unsigned B3 :1;
unsigned undefined1 :9;
unsigned BD :1;
unsigned BS :1;
unsigned BT :1;
unsigned undefined2 :16;
} DebugReg6;
```

When Interrupt1 was occurred by DR0 ~ DR3.

When Interrupt1 was occurred by Gd bit.

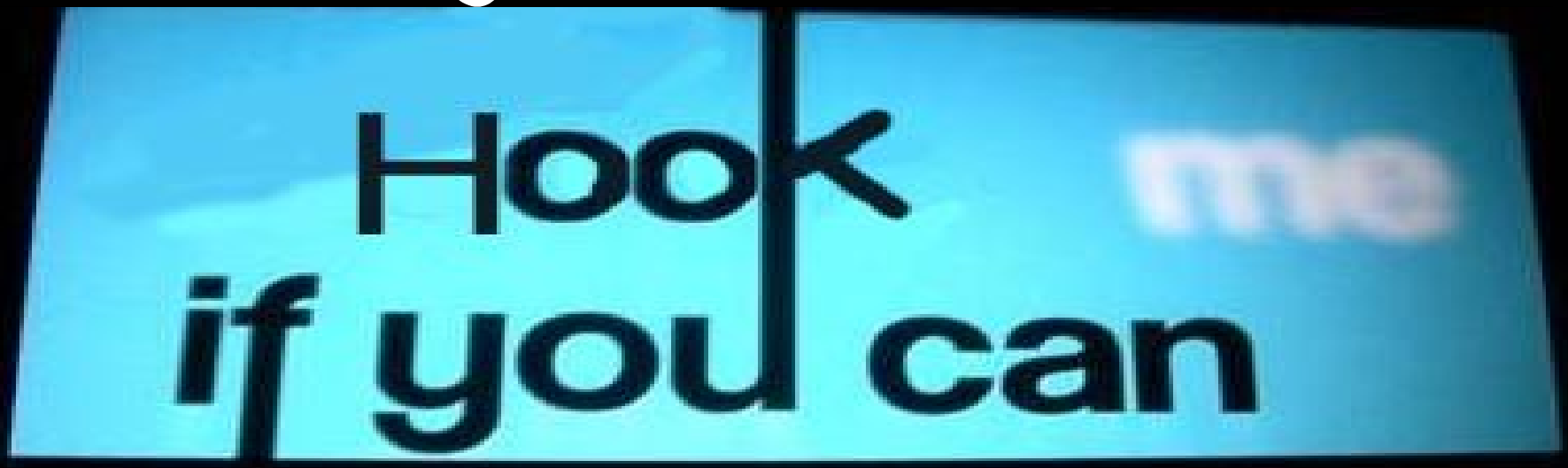
When Interrupt1 was occurred by TF bit.

Hook the planet!

Demonstration

Hook me, if you can!

Code Engn
www.CodeEngn.com



Hook me, if you can!

1) SDT Restore

2) SDT Relocation

3) KiFastCallEntry imitation

4) Bypassing Sysenter hook

Hook me, if you can!

SDT Restore Motivation

- ☐ I want to be free from SDT hooking!
- ☐ Is there no way to restore SDT?

Hook me, if you can!

Concept

- We can write to Kernel memory from User space by writing directly to *₩device₩physicalmemory*

- We can get original copy of the ServiceTable by loading ntoskrnl.exe into memory.

Hook me, if you can!

Code Engn
www.CodeEngn.com

Process

1. Use NtOpenSection to get a handle ₩device₩Physicalmemory with SECTION_MAP_READ | SECTION_MAP_WRITE access.
2. Load ntoskrnl.exe into memory.
3. Use NtMapViewOfSection to map in the physical memory page.
4. Get the address of ServiceTable from the page.
5. Use the address of ServiceTable to offset into the loaded ntoskrnl.exe
6. comparing the copy in the kernel memory with the copy in the loaded ntoskrnl.exe

Hook me, if you can!

Demonstration

Hook me, if you can!

SDT Relocation Motivation

- ☐ How about change SDT as new thing?
- ☐ Is there no way to make SDT hook localy?

Hook me, if you can!

Concept

- Windows doesn`t use KeServiceDescriptorTable for getting address of ServiceTable. Actually, Windows use ETHREAD->ServiceTable to get address of ServiceTable.

- When new thread is created, we can know this by using PsSetCreateThreadNotifyRoutine()

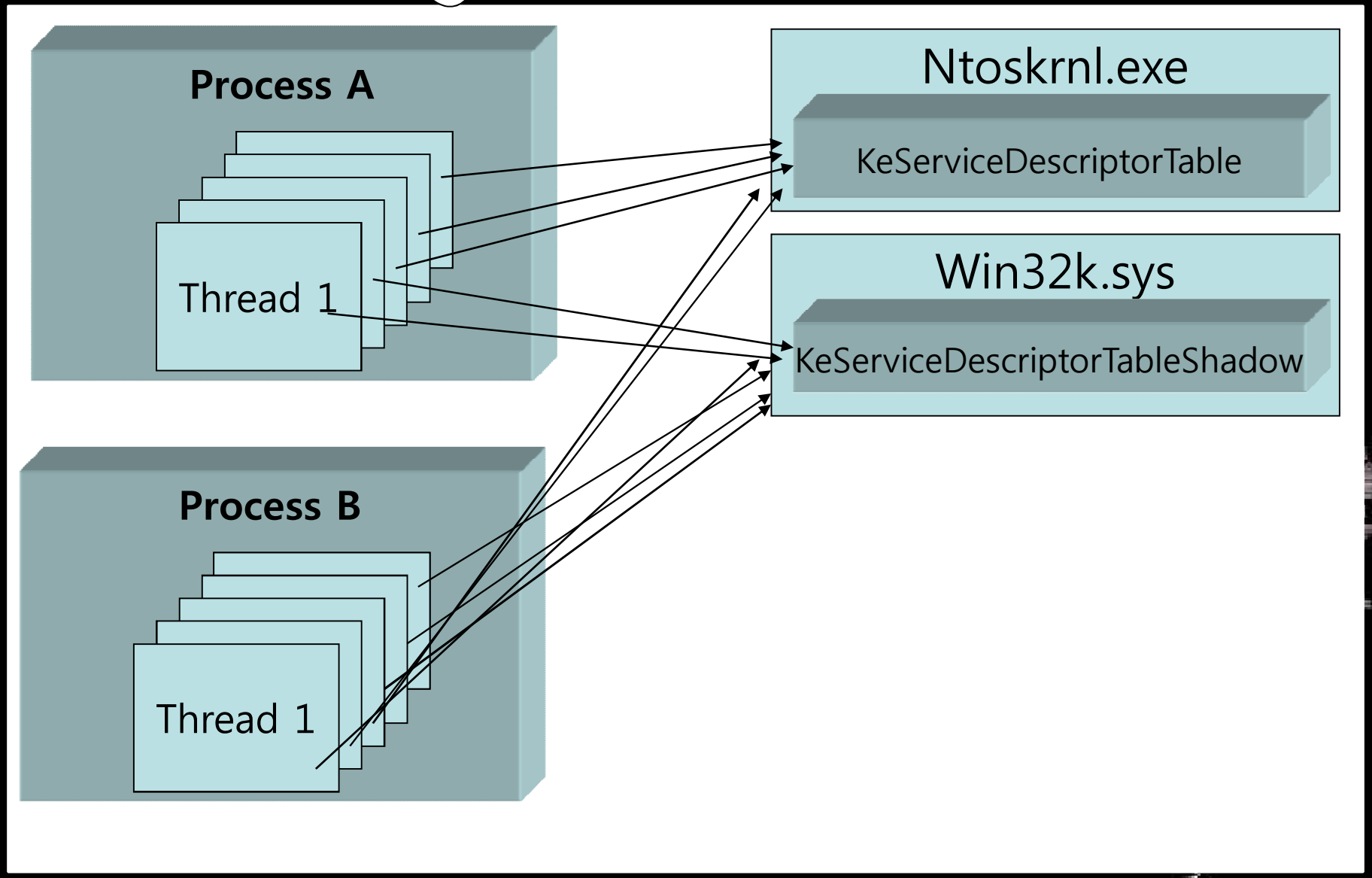
Hook me, if you can!

Process

1. Copy the SerivceTable from ntoskrnl.exe
2. Tracing threads of seleted process, and make NewSDT at each thread.
3. Regist PsSetCreateThreadNotifyRoutine()

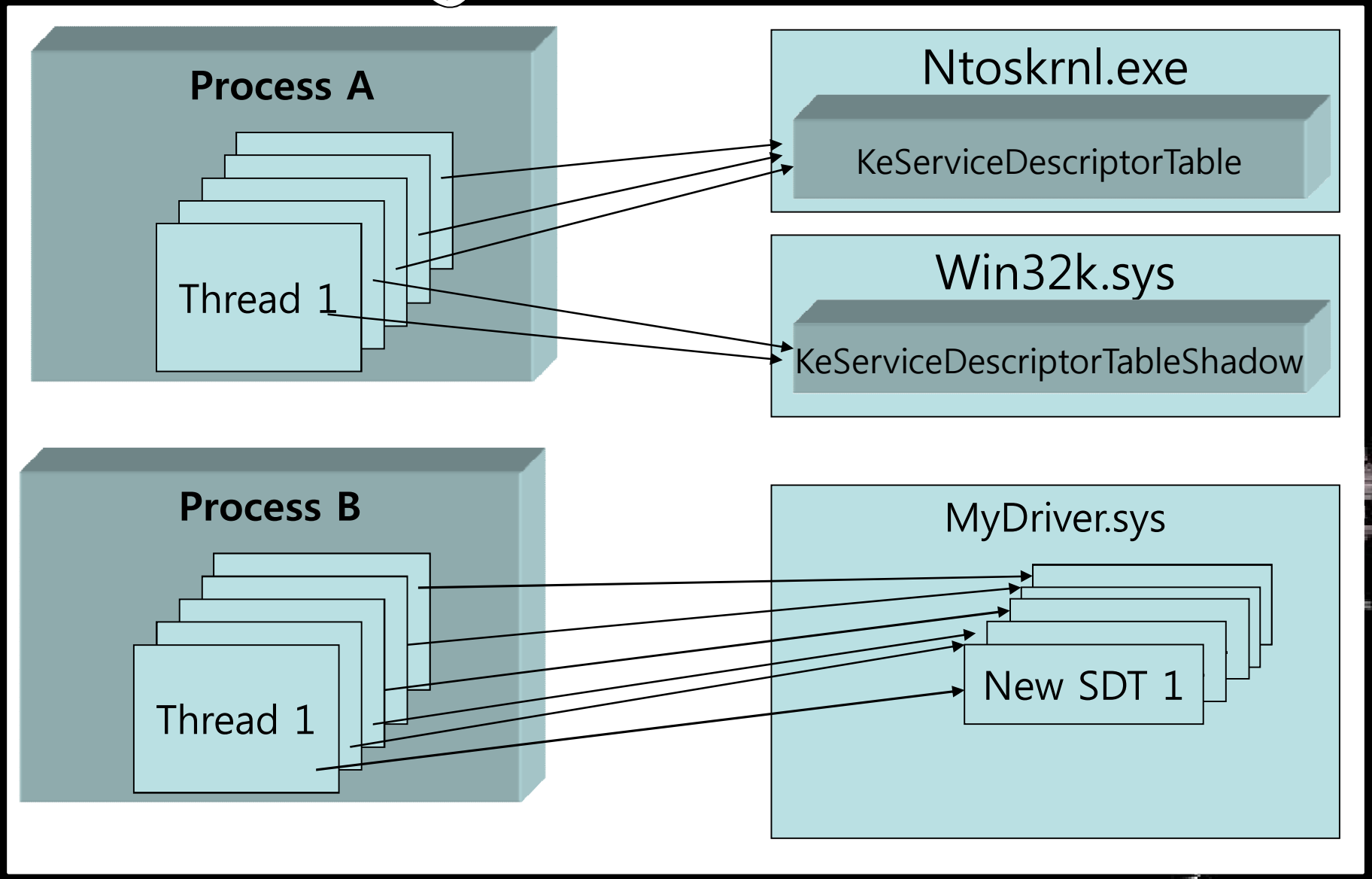
Hook me, if you can!

Code Engn
www.CodeEngn.com



Hook me, if you can!

Code Engn
www.CodeEngn.com



Hook me, if you can!

What is the problem?

- When UserThread is maded, ETHREAD->ServiceTable of that UserThread doesn`t have KeServiceDescriptorTableShadow.

Where does problem come from?

- This problem is caused by PsConvertToGuiThread() that called by KiFastCallEntry().

Hook me, if you can!

Code Engn
www.CodeEngn.com

PsConvertToGuiThread():

```
//Check Kernel Mode
cmp byte ptr[esi+0x140],bl  //KTHREAD_PREVIOUS_MODE,kernelmode
jz InvalidParam

cmp dword ptr[0xBF820ECE],ebx //PspWin32ProcessCallback?
jz AccessDenied

cmp dword ptr[esi+0xE0],KeServiceDescriptorTable    //KTHREAD_SERVICE_TABLE
jnz AlreadyWin32

.............................................
```

Hook me, if you can!

Demonstration

Hook me, if you can!

KiFastCallEntry Imitation Motivation

- ☐ Is there no way to bypass SDT hooking by hooking?
- ☐ Is there no way to handle syscall by myself?

Hook me, if you can!

Concept

- The SYSENTER instruction passes control to the address specified in one of the Model-Specific Registers(MSRs). The Name of this register is IA32_SYSENTER_EIP, We can read and write to this register, By using RDMSR , WRMSR instruction.

Hook me, if you can!

Process

1. Copy the KeServiceDescriptorTable from ntoskrnl.exe,
   Copy the KeServiceDescriptorTableShadow from Win32k.sys

2. Change IA32_SYSENTER_EIP for all processors.

3. Hadling system call **J**

Hook me, if you can!

CodeEngn
www.CodeEngn.com

How we can do it?

KiFastCallEntry():

```
.............................
mov ebx,dword ptr[edi+0xC]  //SERVICE_DESCRIPTOR_NUMBER
xor ecx,ecx
mov cl,byte ptr [eax+ebx]

//Get the function point
mov edi,dword ptr[edi]  //Service_Descriptor_Base
.............................
```

Hook me, if you can!

Code Engn
www.CodeEngn.com

What is the thing that I did?

```
        //Get the function point
        mov edi,dword ptr[edi]  //Service_Descriptor_Base

        //Blocking SSDT Hooking
        cmp edi,Orginal_ServiceTable
        jne Shadow

        mov edi,NewServiceTable
        jmp NotShadow
Shadow:
        mov edi,NewShadowTable
NotShadow:
        mov ebx,dword ptr[edi+eax*4]
```

Hook me, if you can!

Demonstration

Hook me, if you can!

Bypassing sysenter hook Motivation

- ☐ Is there no way to bypass sysenter hook?
- ☐ Is there more way to bypass SDT hook?

Hook me, if you can!

Concept

- In Windows XP, Defaultly, Windows use sysenter to handle system call, but using int 0x2e is possible.

- We can modify the code of KiFastSystemCall() in the ntdll.dll by using Win32 UserLevel API global hook.

Hook me, if you can!

Process

1. Hook IDT (INT 0x2E) for all processors.
2. Modify the code of KiFastSystemCall()
3. Hadling system call **J**

Hook me, if you can!

Code Engn
www.CodeEngn.com

Before:

```
7C93EB8B | 8BD4 | MOV EDX,ESP
7C93EB8D | 0F34 | SYSENTER
```

After:

```
7C93EB8B | 8BD4 | MOV EDX,ESP
7C93EB8D | CD 2E | INT 2E
```

Hook me, if you can!

How we can do it?

```
__declspec(naked) MyKiSystemService()
{
  __asm
  {
    .............
    ExitService:
        sti
        jmp SyscallEntry

    .............
```

Hook me, if you can!

CodeEngn
www.CodeEngn.com

```
SysCallEntry:
        mov edi,eax
        shr edi,0x8 //SERVICE_TABLE_SHIFT
        and edi,0x30    //SERVICE_TABLE_MASK
        mov ecx,edi

        ................

        mov cl,byte ptr [eax+ebx]

        //Get the function point
        mov edi,dword ptr[edi]  //Service_Descriptor_Base

        .............
```

Hook me, if you can!

What is the thing that I did?

```
        //Get the function point
        mov edi,dword ptr[edi]  //Service_Descriptor_Base

        //Blocking SSDT Hooking
        cmp edi,Orginal_ServiceTable
        jne Shadow

        mov edi,NewServiceTable
        jmp NotShadow
Shadow:
        mov edi,NewShadowTable
NotShadow:
        mov ebx,dword ptr[edi+eax*4]
```

?

http://dualpage.muz.ro