内核级后门RootKit技术

江苏大学　信息安全系　xyzreg
E-mail：xyzreg@yahoo.com.cn
URL ： http://www.xyzreg.net

声　明

- 本PPT原为在我们系讨论会上我所作的演讲，因CVC里朋友们对Rootkit技术的热爱与渴求，先把PPT放出来，希望对有些朋友有所启发。其中有些敏感技术信息已删去，不过相信对有些朋友还是有用的。
- PS: EST ： ww.eviloctal.com
  CVC ： www.retcvc.com
  江苏大学信息安全系: www.cnWhiteHat.org

前 言

- 什么是 Rootkit [此处只讨论基于Windows平台的]

  Rootkit与普通木马后门以及病毒的区别

- Rootkit宗旨：隐蔽

  通信隐蔽、自启动项隐藏、文件隐藏、进程/模块隐藏、注册表隐藏、服务隐藏、端口隐藏 etc.

- 研究内核级后门 Rootkit 技术的必要性

  事物两面性；信息战、情报战 ……

Rootkit技术发展

1. Ring3（用户态）-> Ring0（核心态）
2. MEP（Modify Execution Path）-> DKOM（Direct Kernel Object Manipulation）
3. 越来越深入系统底层，挖掘未公开系统内部数据结构
4. 非纯技术性的各种新思路..

技术总揽 之 隐藏篇

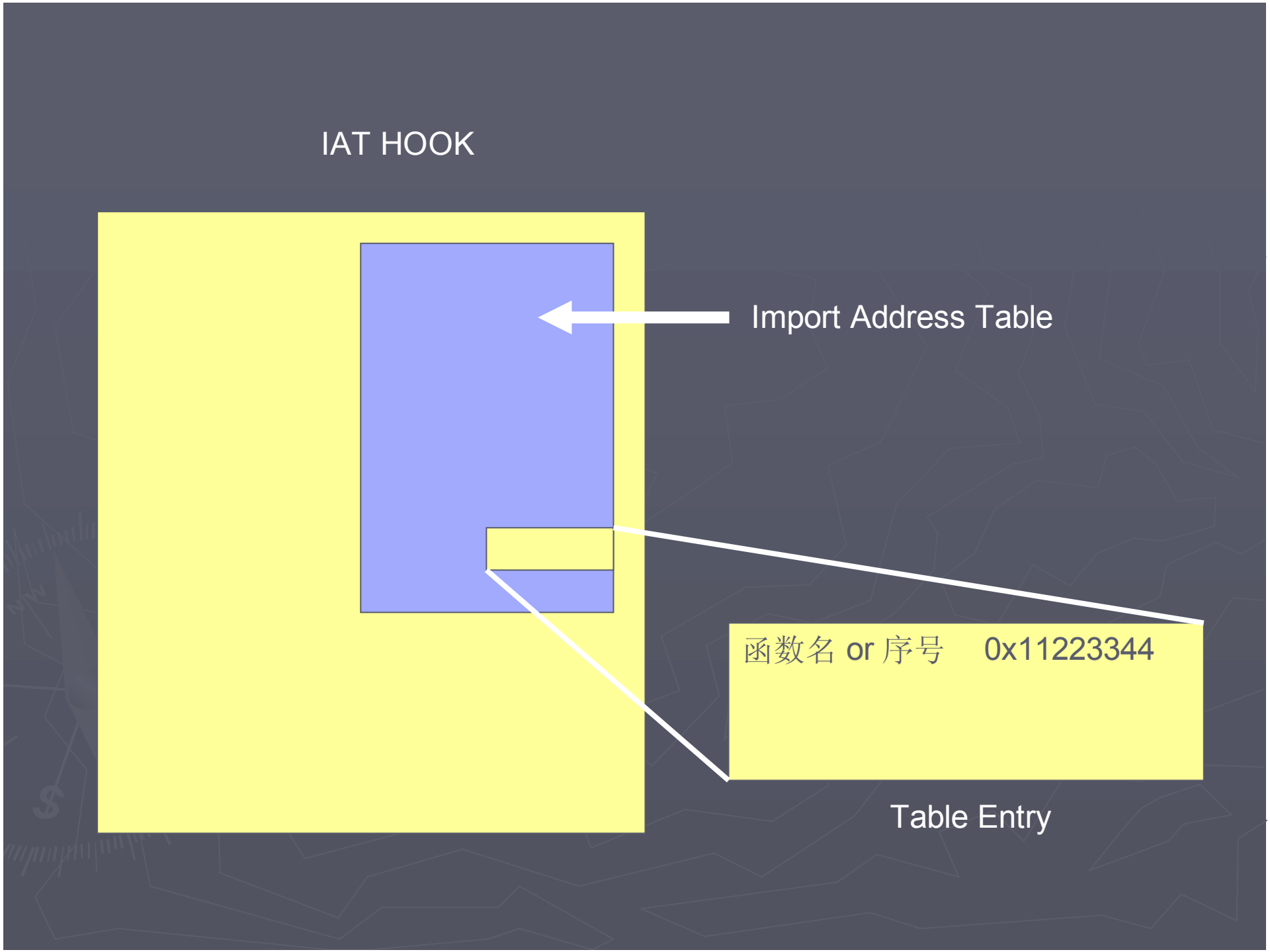
- MEP (Modify Execution Path) 行为拦截挂钩技术

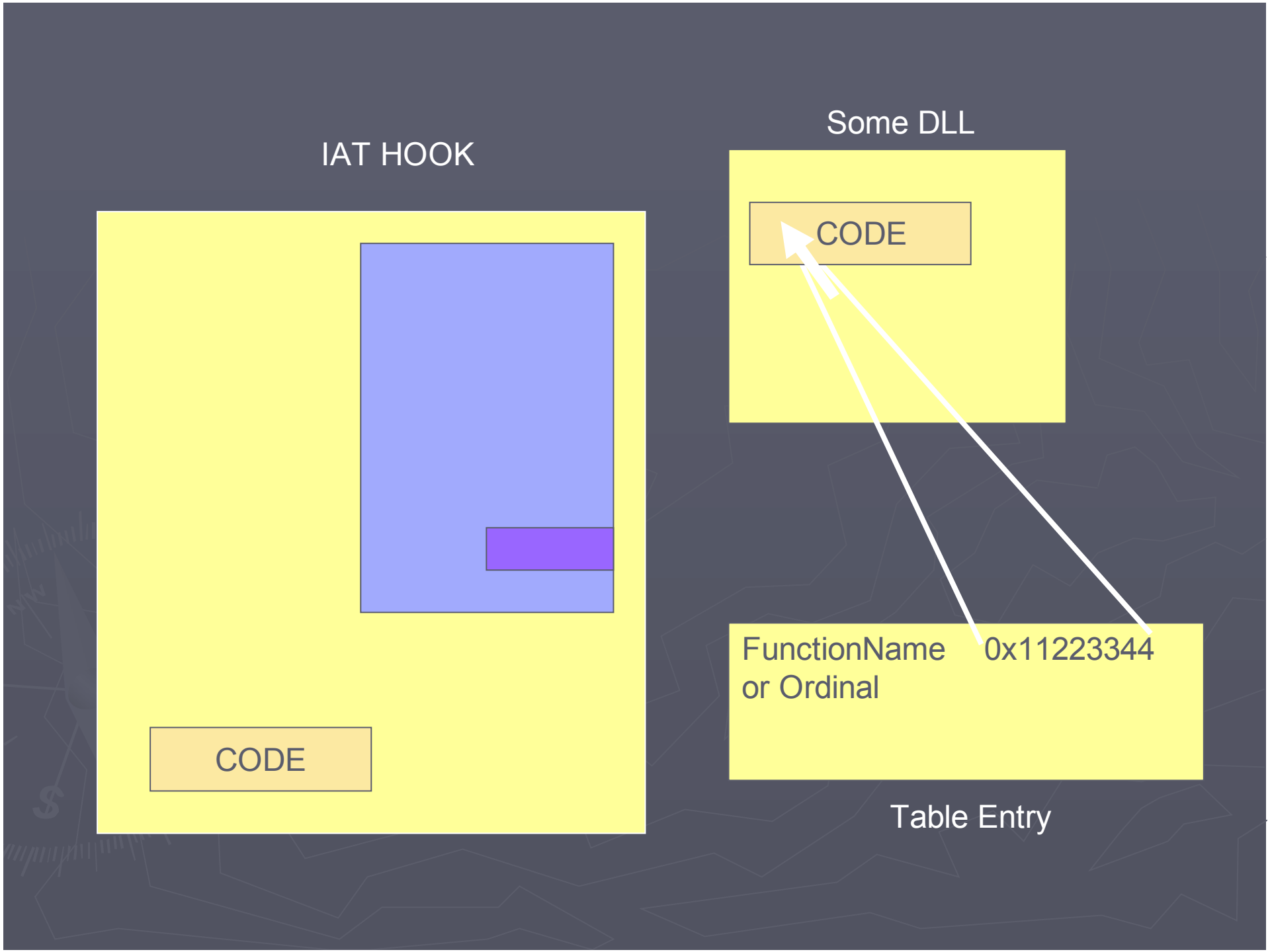
  Hooks（挂钩、挂接的意思）：

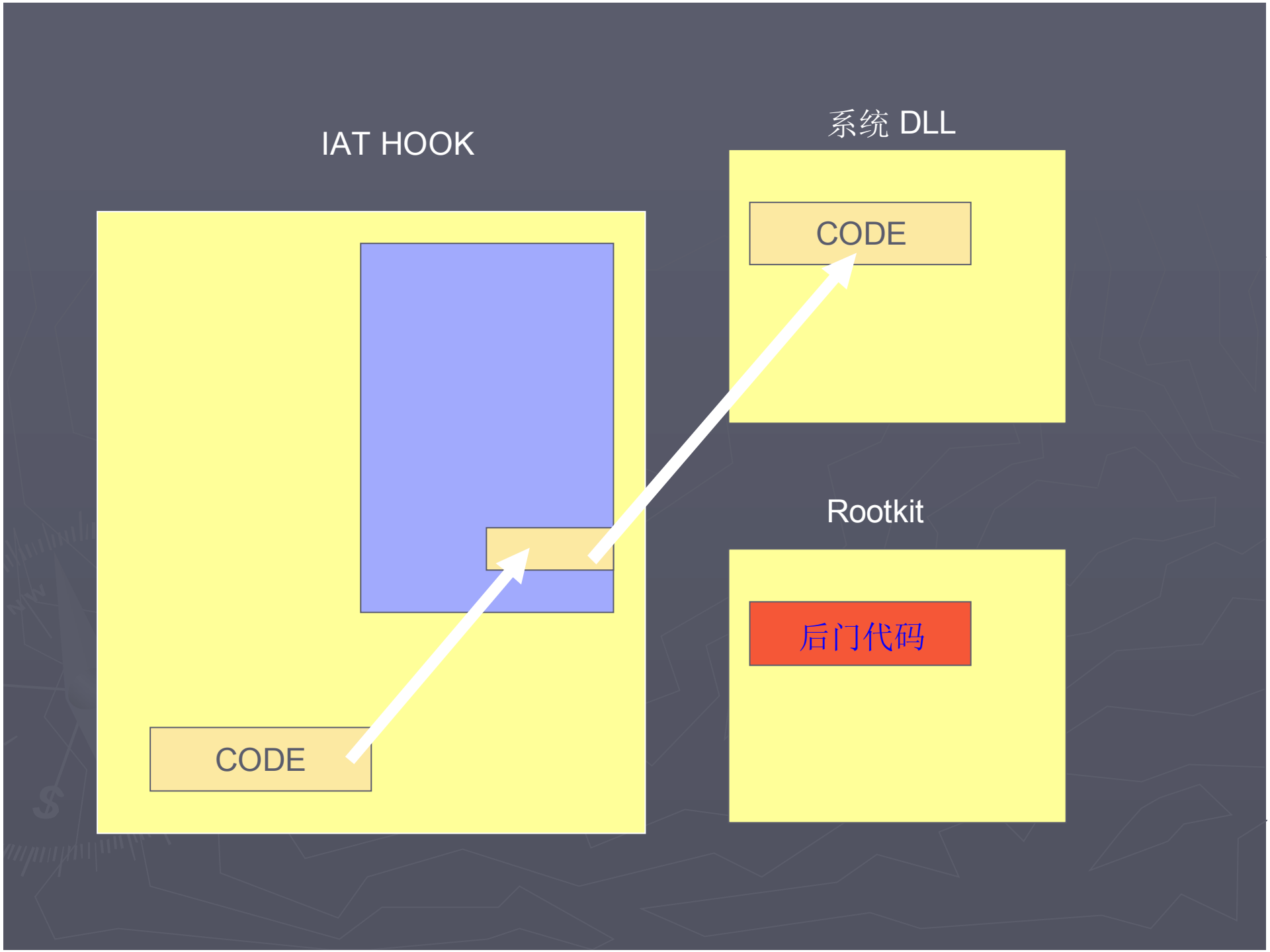
  目的：拦截系统函数或相关处理例程，先转向我们自己的函数处理，这样就可以实现过滤参数或者修改目标函数处理结果的目的，实现进程、文件、注册表、端口之类的隐藏

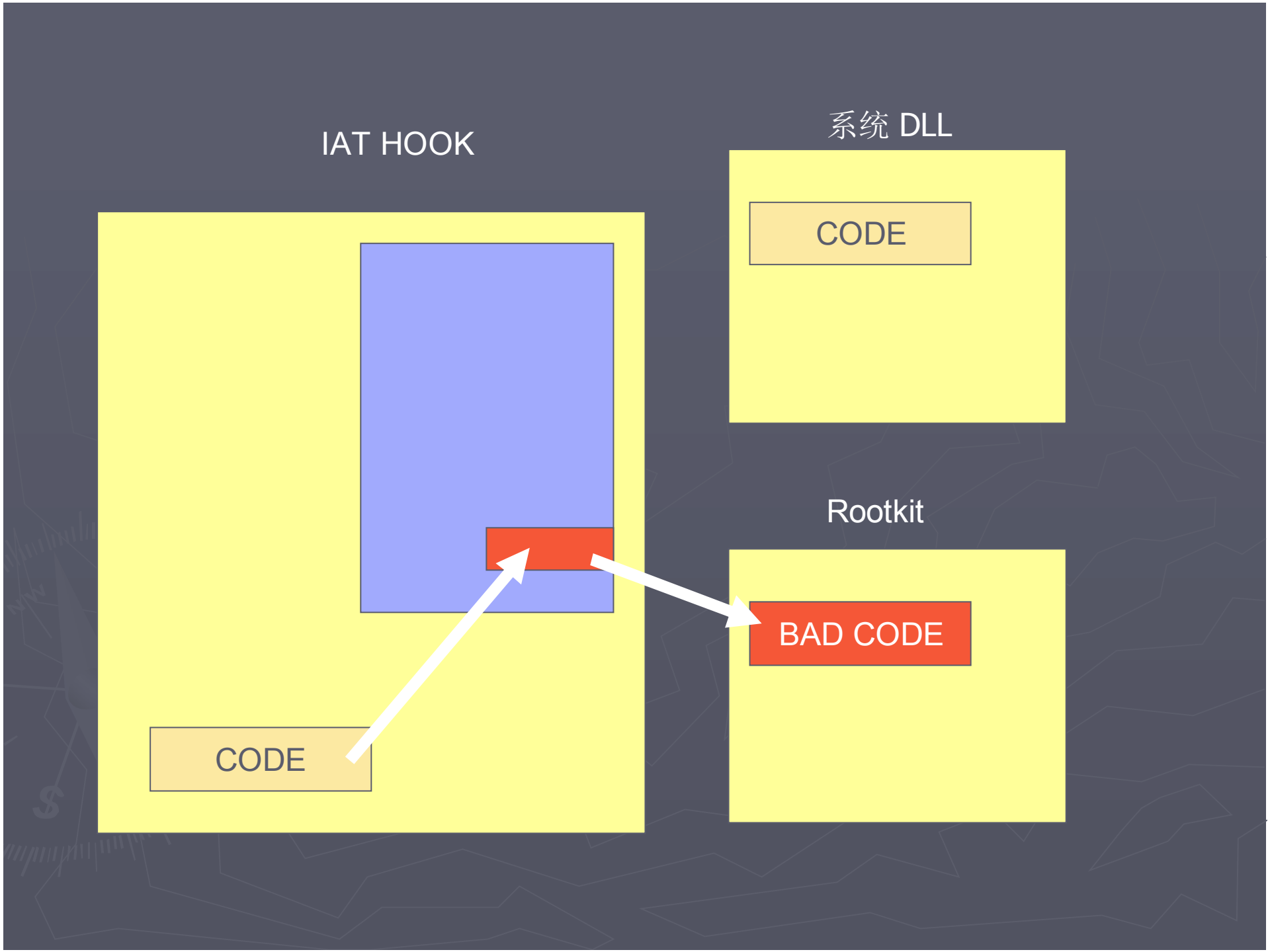
  Hook技术分类：

  - Inline Hook(比如修改目标函数前几个字节为 jmp至我们的函数)
  - IAT (Import Address Table)
  - SSDT (KeServiceDescriptor Table)
  - IDT (Interrupt Descriptor Table)
  - Filter Driver (I/O Request Packet (IRP))
  - Hook IRP Function, etc…


IAT HOOK
Import Address Table
函数名 or 序号 0x11223344
Table Entry


IAT HOOK
CODE
Some DLL
CODE
FunctionName
or Ordinal
0x11223344
Table Entry


IAT HOOK
系统 DLL
CODE
Rootkit
后门代码
CODE


IAT HOOK
系统 DLL
CODE
Rootkit
BAD CODE
CODE

技术总揽 隐藏篇 之 代码注入

- **Ring3**: a.CreateRemoteThread + WriteProcessMemory
  1.线程注入 2.代码注入
  b.SetWindowsHookEx
  c. HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs
  d.Winlogon通知包
  e.感染PE文件（1、全部插入 2、感染IAT ）
  f. DebugActiveProcess + SetThreadContext
  etc. (Activx,SPI,BHO…)
- **Ring0**: KeAttachProcess…

技术总揽 隐藏篇 之 DKOM

- DKOM (Direct Kernel Object Manipulation)

1. 直接修改系统内核数据实现隐藏
2. 因为是修改系统内核数据，所以要写驱动（或采用其他技巧）进入管态（Ring0）
3. 使用WinDbg、SoftICE、IDA Pro等工具自己挖掘未公开的Windows系统内部结构，从而实现比较好的效果…
4. 对Windows系统机制非常熟悉，前置课程《操作系统原理》

技术总揽 之 非常规进Ring0

1. 通过中断门/任务门/调用门/内存映射等技巧(只适用*Wn9x*,比如CIH)
2. \Device\PhysicalMemory对象
3. SetSystemInformation函数中SystemLoadAndCallImage参数,加载驱动
4. 感染HAL.DLL或者Win32k.sys等文件,添加调用门
5. 常规调用操作Windows服务的函数加载驱动(因常规而不隐蔽)
6. 直接调用本机函数NtLoadDriver加载驱动

技术总揽 之 自启动

1. 注册表中Run相关项
2. 相关ini文件
3. BHO（浏览器插件）
4. Winlogon通知包
5. SPI
6. Avticvx...
7. 注册为Windows服务
8. 替换现有服务
9. 其他各种隐蔽的注册表项（知道的人少，但可利用的却很多 ...）
10. 感染系统文件
11. 写入硬盘/网卡固件
12. 写入BIOS

Etc...

技术总揽 之 网络通信篇

- 总之越底层越隐蔽，穿透防火墙的几率就越高

A.

1. 代码注入到防火墙默认允许访问网络的系统进程(如IE)
2. Hook Socket API 或者 SPI技术 或基于TDI等实现端口复用
3. TDI层面通信
4. 在NDIS层面上通信… （pt,mp…）[难点：自己实现的细节多，自己写TCP/IP协议栈，当然也效果最好，能穿透软件防火墙)

B．http隧道；伪装为DNS协议包。为了穿透边界防火墙…

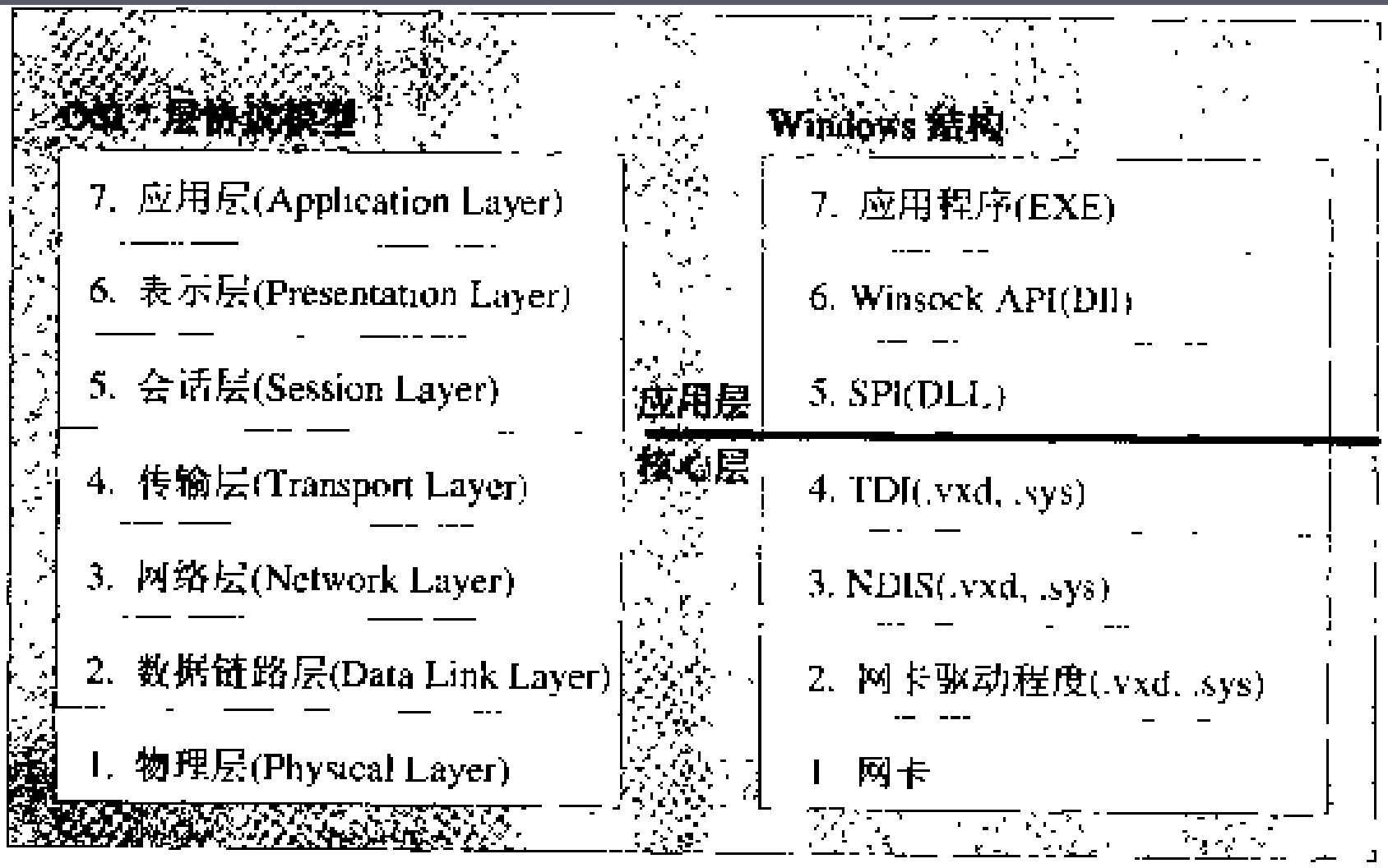


图 1.4　OSI 7 层协议与 Windows 结构的概略映射

具体实现 之 进程隐藏

1. 代码注入（DLL注入，线程注入，进程注入…),实现**无进程**
2. 挂钩应用层上的Process32First、Process32First等函数
3. 挂钩系统服务NtQuerySystemInformation
4. 从进程控制块中的活动进程链表（ ActiveProcessLinks ）中摘除自身
5. 从csrss.exe进程中的句柄表中摘除自身
6. 挂钩SwapContext，自己实现线程调度
7. 从PspCidTable表中摘除自身 （此招目前能对付IceSword等AntiRootkit工具的检测）

etc…

```
typedef struct _HANDLE_TABLE {
    PVOID p_hTable;
    PEPROCESS QuotaProcess;
    PVOID UniqueProcessId;
    EX_PUSH_LOCK HandleTableLock [4];
    LIST_ENTRY HandleTableList;
    EX_PUSH_LOCK HandleContentionEvent;
    PHANDLE_TRACE_DEBUG_INFO DebugInfo;
    DWORD ExtraInfoPages;
    DWORD FirstFree;
    DWORD LastFree;
    DWORD NextHandleNeedingPool;
    DWORD HandleCount;
    DWORD Flags;
};
```

通过特征码搜索系统中PspCidTable地址

```
PsLookupProcessByProcessId:
        mov edi, edi
        push ebp
        mov ebp, esp
        push ebx
        push esi
        mov eax, large fs:124h
        push [ebp+arg_4]
        mov esi, eax
        dec dword ptr [esi+0D4h]
        push PspCidTable
        call ExMapHandleToPointer
```

具体实现 之 文件隐藏

1. 采用病毒技术，感染寄生于其他文件,实现无文件
2. 挂钩应用层上的FindFirstFile、FindNextFirst等函数
3. 挂钩内核态中系统服务ZwQueryDirectoryFile
4. 文件过滤驱动
5. 修改 FSD IRP Fuction 函数地址，再对相关IRP处理...
6. Inline Hook FSD 相关例程～，再对相关IRP处理...

etc…

类似地

- 具体实现 之 注册表隐藏...
- 具体实现 之 服务隐藏...
- 具体实现 之 模块隐藏...
- 具体实现 之 端口隐藏...

PS...

RK技术新挑战

- 突破主动防御以及进程行为监控(绕过注册表监控、代码注入监控、驱动加载监控等)

1. 对付卡巴6、SSM、GSS等的方法
2. 突破各大比较强的防火墙的方法，ZoneAlarm，Outpost，Kerio，BlackICE，
3. 对付杀毒软件的通用方法...
4. 突破进程行为监控的终极通用方法...

RK技术新挑战

- KOH技术...

- DEMO1

我的与Rootkit相关的项目(有关信息隐去）

1. 代号为XXXX
2. 代号为XXXX
3. 代号为XXXX （属于Ring3新思路系统级型Rootkit）
4. 代号为XXXX （Rootkit工具，能隐藏指定进程文件端口服务注册表等，并且结合相关底层技术能从系统底层对付杀毒软件 ）
5. 代号为XXXX （属于Ring0 纯驱动级别内核级Rootkit，包括网络通信部分）

重点：不光注重隐蔽性，还得注重稳定性，网络通信方面得考虑各种复杂的网络架构情况

后续研究：通过写BIOS或者硬盘固件，后门硬件化，实现后门的高度隐蔽和长期生存，即使重装系统，后门存在于目标电脑中…

- DEMO2

安全检测以及防范

时间关系，此话题这次不细谈了，有兴趣的话以后聊。不过上次讲的我那个入侵防护系统的进程行为监控模块就能从源头防范。

参考资料

1. http://www.rootkit.com
2. 《Subverting the Windows Kernel》
3. RAIDE: Rootkit Analysis Identification Elimination
4. 《Windows防火墙与封包拦截技术》

谢谢观看