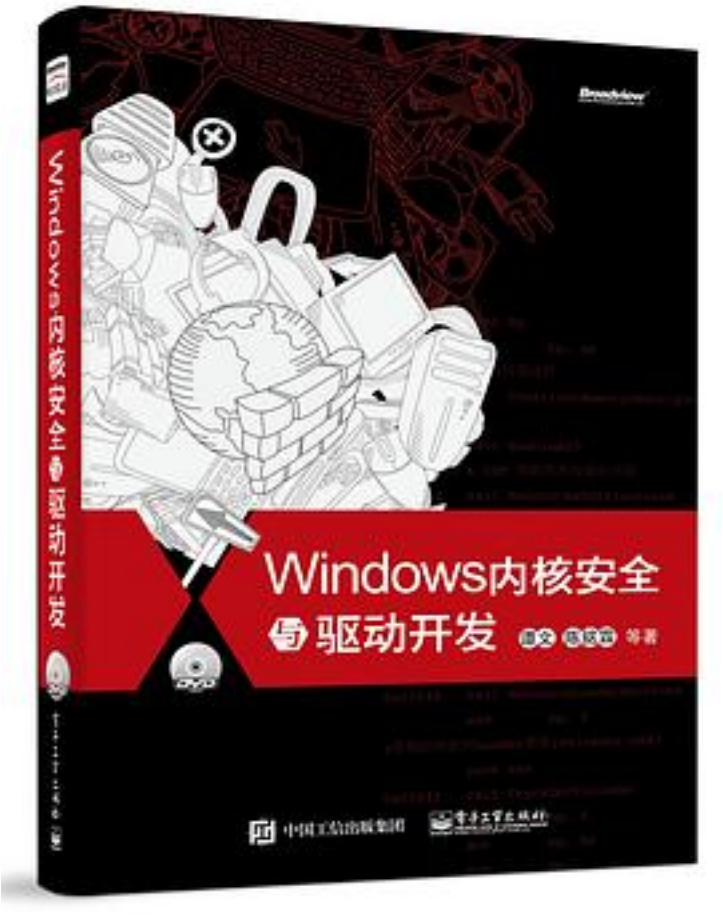
Windows内核安全与驱动开发(含CD光盘1张)



Windows内核安全与驱动开发(含CD光盘1张)_下载链接1_

著者:谭文

出版者:电子工业出版社

出版时间:2015-6-1

装帧:平装

isbn:9787121262159

《Windows内核安全与驱动开发(含CD光盘1张)》的前身是《天书夜读——从汇编语言到Windows内核编程》和《寒江独钓——Windows内核安全编程》。与Windows客户端安全软件开发相关的驱动程序开发是本书的主题。书中的程序使用环境从32位到64位

，从Windows XP到Windows 8都有涉及，大部分程序不经过修改即可在Windows 10上运行。同时本书也深入浅出地介绍了进行内核安全编程所需要的操作系统、汇编等基础知识。

《Windows内核安全与驱动开发(含CD光盘1张)》共分三篇，基础篇囊括了驱动开发的基础知识，降低了入门的难度；开发篇介绍了在实际工作中可能遇到的各种开发需求的技术实现，包括：串口的过滤、键盘的过滤、磁盘的虚拟、磁盘的过滤、文件系统的过滤与监控、文件系统透明加密、文件系统微过滤驱动、网络传输层过滤、Windows过滤平台、NDIS协议驱动、NDIS小端口驱动、NDIS中间层驱动、IA-32汇编基础、IA-32体系中的内存地址、处理器权限级别切换、IA-32体系结构中的中断和Windows内核挂钩；高级篇包含了汇编语言、操作系统原理、处理器体系架构相关的内容。本书是由长期从事这个行业的工程师自己写的，所以处处以实用为准。对细节的考究主要体现在对实际问题的解决，而不是知识的详尽程度上。

《Windows内核安全与驱动开发(含CD光盘1张)》适合计算机安全软件从业人员、计算机相关专业院校学生以及有一定C语言和操作系统基础知识的编程爱好者阅读。

作者介绍:

谭文，网名楚狂人，长期从事客户端安全的开发工作。先后在NEC、英特尔亚太研发有限公司、腾讯科技任职。曾经从事过企业安全软件、x86版Android、腾讯电脑管家等开发工作。编写了《天》与《寒》的大部分章节，并为本书重写了部分章节，添加了一些新的内容。

陈铭霖，腾讯电脑管家团队高级工程师，长期从事Windows系统研究，目前从事客户端安全的开发工作。编写了本书中部分新的章节，并统稿全书，重新整理和编辑了《天》和《寒》的全部内容。对本书的面世做出了巨大的贡献。

张佩，Windows驱动开发技术专家，长期从事声卡、显卡等硬件驱动程序的开发、调试工作。目前在英特尔亚太研发有限公司平板电脑相关的部门工作。曾著有《竹林蹊径——深入浅出Windows驱动开发》一书。为《寒》贡献了若干个网络驱动相关的章节，这些内容大多原版整合到了本书中。

杨潇，曾为Windows客户端安全工程师，先后在上海贝尔和北京Comodo工作。后来离职创业，目前为西安一家医疗科技公司的CEO。为《寒》一书编写了磁盘驱动相关的章节，这些章节也整合到了本书中。

邵坚磊，网名wowocock，业内著名的Windows安全技术专家。长期从事Windows安全相关的内核开发工作。目前在奇虎360任职。为《天》和《寒》都贡献了部分章节和代码实例，这些内容有一部分整合到了本书中。

卢冠豪，中国台湾人。毕业于辅仁大学资讯工程学系。长期从事C、C++、网络与通信程序设计的工作；参与过“端点安全”、“资产管理”、“网络流量分析”等项目的开发与维护；擅长Windows项目开发。编写了《寒》一书中的文件系统微端口过滤一章，此章也整合到了本书中。

目录: 基 础 篇
第1章 内核上机指导 2
1.1 下载和使用WDK 2
1.1.1 下载并安装WDK 2
1.1.2 编写第一个C文件 4
1.1.3 编译一个工程 5

1.2 安装与运行 6
1.2.1 下载一个安装工具 6
1.2.2 运行与查看输出信息 7
1.2.3 在虚拟机中运行 8
1.3 调试内核模块 9
1.3.1 下载和安装WinDbg 9
1.3.2 设置Windows XP调试执行 9
1.3.3 设置Vista调试执行 10
1.3.4 设置VMware的管道虚拟串口 11
1.3.5 设置Windows内核符号表 12
1.3.6 实战调试first 13
第2章 内核编程环境及其特殊性 16
2.1 内核编程的环境 16
2.1.1 隔离的应用程序 16
2.1.2 共享的内核空间 17
2.1.3 无处不在的内核模块 18
2.2 数据类型 19
2.2.1 基本数据类型 19
2.2.2 返回状态 19
2.2.3 字符串 20
2.3 重要的数据结构 21
2.3.1 驱动对象 21
2.3.2 设备对象 22
2.3.3 请求 24
2.4 函数调用 25
2.4.1 查阅帮助 25
2.4.2 帮助中有的几类函数 26
2.4.3 帮助中没有的函数 28
2.5 Windows的驱动开发模型 29
2.6 WDK编程中的特殊点 30
2.6.1 内核编程的主要调用源 30
2.6.2 函数的多线程安全性 30
2.6.3 代码的中断级 32
2.6.4 WDK中出现的特殊代码 32
第3章 字符串与链表 35
3.1 字符串操作 35
3.1.1 使用字符串结构 35
3.1.2 字符串的初始化 36
3.1.3 字符串的拷贝 37
3.1.4 字符串的连接 38
3.1.5 字符串的打印 38
3.2 内存与链表 40
3.2.1 内存的分配与释放 40
3.2.2 使用LIST_ENTRY 41
3.2.3 使用长长整型数据 43
3.3 自旋锁 44
3.3.1 使用自旋锁 44
3.3.2 在双向链表中使用自旋锁 45
3.3.3 使用队列自旋锁提高性能 46
第4章 文件、注册表、线程 47
4.1 文件操作 47
4.1.1 使用OBJECT_ATTRIBUTES 47
4.1.2 打开和关闭文件 48
4.1.3 文件读/写操作 51

4.2 注册表操作 53
4.2.1 注册表键的打开 53
4.2.2 注册表键值的读 55
4.2.3 注册表键值的写 57
4.3 时间与定时器 58
4.3.1 获得当前“滴答”数 58
4.3.2 获得当前系统时间 58
4.3.3 使用定时器 59
4.4 线程与事件 62
4.4.1 使用系统线程 62
4.4.2 在线程中睡眠 63
4.4.3 使用同步事件 64
第5章 应用与内核通信 67
5.1 内核方面的编程 68
5.1.1 生成控制设备 68
5.1.2 控制设备的名字和符号链接 70
5.1.3 控制设备的删除 71
5.1.4 分发函数 72
5.1.5 请求的处理 73
5.2 应用方面的编程 74
5.2.1 基本的功能需求 74
5.2.2 在应用程序中打开与关闭设备 75
5.2.3 设备控制请求 75
5.2.4 内核中的对应处理 77
5.2.5 结合测试的效果 79
5.3 阻塞、等待与安全设计 80
5.3.1 驱动主动通知应用 80
5.3.2 通信接口的测试 81
5.3.3 内核中的缓冲区链表结构 83
5.3.4 输入：内核中的请求处理中的安全检查 84
5.3.5 输出处理与卸载清理 85
第6章 64位和32位内核开发差异 88
6.1 64位系统新增机制 88
6.1.1 WOW64子系统 88
6.1.2 PatchGuard技术 91
6.1.3 64位驱动的编译、安装与运行 91
6.2 编程差异 92
6.2.1 汇编嵌入变化 92
6.2.2 预处理与条件编译 93
6.2.3 数据结构调整 93
开 发 篇
第7章 串口的过滤 96
7.1 过滤的概念 96
7.1.1 设备绑定的内核API之一 97
7.1.2 设备绑定的内核API之二 98
7.1.3 生成过滤设备并绑定 98
7.1.4 从名字获得设备对象 100
7.1.5 绑定所有串口 101
7.2 获得实际数据 102
7.2.1 请求的区分 102
7.2.2 请求的结局 103
7.2.3 写请求的数据 104
7.3 完整的代码 105
7.3.1 完整的分发函数 105

7.3.2 如何动态卸载 106
7.3.3 代码的编译与运行 107
第8章 键盘的过滤 109
8.1 技术原理 110
8.1.1 预备知识 110
8.1.2 Windows中从击键到内核 110
8.1.3 键盘硬件原理 112
8.2 键盘过滤的框架 112
8.2.1 找到所有的键盘设备 112
8.2.2 应用设备扩展 115
8.2.3 键盘过滤模块的DriverEntry 117
8.2.4 键盘过滤模块的动态卸载 117
8.3 键盘过滤的请求处理 119
8.3.1 通常的处理 119
8.3.2 PNP的处理 120
8.3.3 读的处理 121
8.3.4 读完成的处理 122
8.4 从请求中打印出按键信息 123
8.4.1 从缓冲区中获得KEYBOARD_INPUT_DATA 123
8.4.2 从KEYBOARD_INPUT_DATA中得到键 124
8.4.3 从MakeCode到实际字符 124
8.5 Hook分发函数 126
8.5.1 获得类驱动对象 126
8.5.2 修改类驱动的分发函数指针 127
8.5.3 类驱动之下的端口驱动 128
8.5.4 端口驱动和类驱动之间的协作机制 129
8.5.5 找到关键的回调函数的条件 129
8.5.6 定义常数和数据结构 130
8.5.7 打开两种键盘端口驱动寻找设备 131
8.5.8 搜索在KbdClass类驱动中的地址 133
8.6 Hook键盘中断反过滤 135
8.6.1 中断：IRQ和INT 136
8.6.2 如何修改IDT 136
8.6.3 替换IDT中的跳转地址 137
8.6.4 QQ的PS/2反过滤措施 139
8.7 直接用端口操作键盘 139
8.7.1 读取键盘数据和命令端口 139
8.7.2 p2cUserFilter的最终实现 140
第9章 磁盘的虚拟 143
9.1 虚拟的磁盘 143
9.2 一个具体的例子 143
9.3 入口函数 144
9.3.1 入口函数的定义 144
9.3.2 Ramdisk驱动的入口函数 145
9.4 EvtDriverDeviceAdd函数 146
9.4.1 EvtDriverDeviceAdd的定义 146
9.4.2 局部变量的声明 146
9.4.3 磁盘设备的创建 147
9.4.4 如何处理发往设备的请求 148
9.4.5 用户配置的初始化 149
9.4.6 链接给应用程序 151
9.4.7 小结 152
9.5 FAT12/16磁盘卷初始化 152
9.5.1 磁盘卷结构简介 152

9.5.2 Ramdisk对磁盘的初始化 154
9.6 驱动中的请求处理 160
9.6.1 请求的处理 160
9.6.2 读/写请求 160
9.6.3 DeviceIoControl请求 162
9.7 Ramdisk的编译和安装 164
9.7.1 编译 164
9.7.2 安装 164
9.7.3 对安装的深入探究 165
第10章 磁盘的过滤 167
10.1 磁盘过滤驱动的概念 167
10.1.1 设备过滤和类过滤 167
10.1.2 磁盘设备和磁盘卷设备过滤驱动 167
10.1.3 注册表和磁盘卷设备过滤驱动 168
10.2 具有还原功能的磁盘卷过滤驱动 168
10.2.1 简介 168
10.2.2 基本思想 169
10.3 驱动分析 169
10.3.1 DriverEntry函数 169
10.3.2 AddDevice函数 170
10.3.3 PnP请求的处理 174
10.3.4 Power请求的处理 178
10.3.5 DeviceIoControl请求的处理 178
10.3.6 bitmap的作用和分析 182
10.3.7 boot驱动完成回调函数和稀疏文件 187
10.3.8 读/写请求的处理 190
第11章 文件系统的过滤与监控 199
11.1 文件系统的设备对象 200
11.1.1 控制设备与卷设备 200
11.1.2 生成自己的一个控制设备 201
11.2 文件系统的分发函数 202
11.2.1 普通的分发函数 202
11.2.2 文件过滤的快速IO分发函数 203
11.2.3 快速IO分发函数的一个实现 205
11.2.4 快速IO分发函数逐个简介 206
11.3 设备的绑定前期工作 207
11.3.1 动态地选择绑定函数 207
11.3.2 注册文件系统变动回调 208
11.3.3 文件系统变动回调的一个实现 209
11.3.4 文件系统识别器 211
11.4 文件系统控制设备的绑定 212
11.4.1 生成文件系统控制设备的过滤设备 212
11.4.2 绑定文件系统控制设备 213
11.4.3 利用文件系统控制请求 215
11.5 文件系统卷设备的绑定 217
11.5.1 从IRP中获得VPB指针 217
11.5.2 设置完成函数并等待IRP完成 218
11.5.3 卷挂载IRP完成后的工作 221
11.5.4 完成函数的相应实现 223
11.5.5 绑定卷的实现 224
11.6 读/写操作的过滤 226
11.6.1 设置一个读处理函数 226
11.6.2 设备对象的区分处理 227
11.6.3 解析读请求中的文件信息 228

11.6.4 读请求的完成 230
11.7 其他操作的过滤 234
11.7.1 文件对象的生存周期 234
11.7.2 文件的打开与关闭 235
11.7.3 文件的删除 237
11.8 路径过滤的实现 238
11.8.1 取得文件路径的三种情况 238
11.8.2 打开成功后获取路径 238
11.8.3 在其他时刻获得文件路径 240
11.8.4 在打开请求完成之前获得路径名 240
11.8.5 把短名转换为长名 242
11.9 把sfilter编译成静态库 243
11.9.1 如何方便地使用sfilter 243
11.9.2 初始化回调、卸载回调和绑定回调 244
11.9.3 绑定与回调 245
11.9.4 插入请求回调 246
11.9.5 如何利用sfilter.lib 249
第12章 文件系统透明加密 252
12.1 文件透明加密的应用 252
12.1.1 防止企业信息泄密 252
12.1.2 文件透明加密防止企业信息泄密 253
12.1.3 文件透明加密软件的例子 253
12.2 区分进程 254
12.2.1 机密进程与普通进程 254
12.2.2 找到进程名字的位置 255
12.2.3 得到当前进程的名字 256
12.3 内存映射与文件缓冲 257
12.3.1 记事本的内存映射文件 257
12.3.2 Windows的文件缓冲 258
12.3.3 文件缓冲：明文还是密文的选择 259
12.3.4 清除文件缓冲 260
12.4 加密标识 263
12.4.1 保存在文件外、文件头还是文件尾 263
12.4.2 隐藏文件头的大小 264
12.4.3 隐藏文件头的设置偏移 266
12.4.4 隐藏文件头的读/写偏移 267
12.5 文件加密表 267
12.5.1 何时进行加密操作 267
12.5.2 文件控制块与文件对象 268
12.5.3 文件加密表的数据结构与初始化 269
12.5.4 文件加密表的操作：查询 270
12.5.5 文件加密表的操作：添加 271
12.5.6 文件加密表的操作：删除 272
12.6 文件打开处理 273
12.6.1 直接发送IRP进行查询与设置操作 274
12.6.2 直接发送IRP进行读/写操作 276
12.6.3 文件的非重入打开 277
12.6.4 文件的打开预处理 280
12.7 读/写加密和解密 285
12.7.1 在读取时进行解密 285
12.7.2 分配与释放MDL 286
12.7.3 写请求加密 287
12.8 crypt_file的组装 289
12.8.1 crypt_file的初始化 289

12.8.2 crypt_file的IRP预处理 290
12.8.3 crypt_file的IRP后处理 293
第13章 文件系统微过滤驱动 297
13.1 文件系统微过滤驱动简介 297
13.1.1 文件系统微过滤驱动的由来 297
13.1.2 Minifilter的优点与不足 298
13.2 Minifilter的编程框架 298
13.2.1 微文件系统过滤的注册 299
13.2.2 微过滤器的数据结构 300
13.2.3 卸载回调函数 303
13.2.4 预操作回调函数 303
13.2.5 后操作回调函数 306
13.2.6 其他回调函数 307
13.3 Minifilter如何与应用程序通信 309
13.3.1 建立通信端口的方法 310
13.3.2 在用户态通过DLL使用通信端口的范例 311
13.4 Minifilter的安装与加载 314
13.4.1 安装Minifilter的INF文件 314
13.4.2 启动安装完成的Minifilter 316
第14章 网络传输层过滤 317
14.1 TDI概要 317
14.1.1 为何选择TDI 317
14.1.2 从socket到Windows内核 318
14.1.3 TDI过滤的代码例子 319
14.2 TDI的过滤框架 319
14.2.1 绑定TDI的设备 319
14.2.2 唯一的分发函数 320
14.2.3 过滤框架的实现 322
14.2.4 主要过滤的请求类型 323
14.3 生成请求：获取地址 324
14.3.1 过滤生成请求 324
14.3.2 准备解析IP地址与端口 326
14.3.3 获取生成的IP地址和端口 327
14.3.4 连接终端的生成与相关信息的保存 329
14.4 控制请求 330
14.4.1 TDI_ASSOCIATE_ADDRESS的过滤 330
14.4.2 TDI_CONNECT的过滤 332
14.4.3 其他的次功能号 333
14.4.4 设置事件的过滤 334
14.4.5 TDI_EVENT_CONNECT类型的设置事件的过滤 336
14.4.6 直接获取发送函数的过滤 337
14.4.7 清理请求的过滤 339
14.5 本书例子tdifw.lib的应用 341
14.5.1 tdifw库的回调接口 341
14.5.2 tdifw库的使用例子 342
第15章 Windows过滤平台 345
15.1 WFP简介 345
15.2 WFP框架 345
15.3 基本对象模型 347
15.3.1 过滤引擎 347
15.3.2 垫片 347
15.3.3 呼出接口 347
15.3.4 分层 348
15.3.5 子层 349

15.3.6 过滤器 350
15.3.7 呼出接口回调函数 354
15.4 WFP操作 359
15.4.1 呼出接口的注册与卸载 360
15.4.2 呼出接口的添加与移除 360
15.4.3 子层的添加与移除 361
15.4.4 过滤器的添加 362
15.5 WFP过滤例子 362
第16章 NDIS协议驱动 370
16.1 以太网包和网络驱动架构 370
16.1.1 以太网包和协议驱动 370
16.1.2 NDIS网络驱动 371
16.2 协议驱动的DriverEntry 372
16.2.1 生成控制设备 372
16.2.2 注册协议 374
16.3 协议与网卡的绑定 375
16.3.1 协议与网卡的绑定概念 375
16.3.2 绑定回调处理的实现 376
16.3.3 协议绑定网卡的API 378
16.3.4 解决绑定竞争问题 379
16.3.5 分配接收和发送的包池与缓冲池 380
16.3.6 OID请求的发送和请求完成回调 381
16.3.7 ndisprotCreateBinding的最终实现 385
16.4 绑定的解除 390
16.4.1 解除绑定使用的API 390
16.4.2 ndisprotShutdownBinding的实现 392
16.5 在用户态操作协议驱动 395
16.5.1 协议的收包与发包 395
16.5.2 在用户态编程打开设备 396
16.5.3 用DeviceIoControl发送控制请求 397
16.5.4 用WriteFile发送数据包 399
16.5.5 用ReadFile发送数据包 400
16.6 在内核态完成功能的实现 402
16.6.1 请求的分发与实现 402
16.6.2 等待设备绑定完成与指定设备名 402
16.6.3 指派设备的完成 403
16.6.4 处理读请求 406
16.6.5 处理写请求 408
16.7 协议驱动的接收回调 412
16.7.1 和接收包有关的回调函数 412
16.7.2 ReceiveHandler的实现 413
16.7.3 TransferDataCompleteHandler的实现 417
16.7.4 ReceivePacketHandler的实现 418
16.7.5 接收数据包的入队 420
16.7.6 接收数据包的出队和读请求的完成 422
第17章 NDIS小端口驱动 427
17.1 小端口驱动的应用与概述 427
17.1.1 小端口驱动的应用 427
17.1.2 小端口驱动示例 428
17.1.3 小端口驱动的运作与编程
概述 429
17.2 小端口驱动的初始化 429
17.2.1 小端口驱动的DriverEntry 429
17.2.2 小端口驱动的适配器结构 431

17.2.3 配置信息的读取 433
17.2.4 设置小端口适配器上下文 433
17.2.5 MPInitialize的实现 434
17.2.6 MPHalt的实现 437
17.3 打开ndisprot设备 438
17.3.1 IO目标 438
17.3.2 给IO目标发送DeviceIoControl请求 439
17.3.3 打开ndisprot接口并完成配置设备 441
17.4 使用ndisprot发送包 443
17.4.1 小端口驱动的发包接口 443
17.4.2 发送控制块（TCB） 444
17.4.3 遍历包组并填写TCB 446
17.4.4 写请求的构建与发送 449
17.5 使用ndisprot接收包 451
17.5.1 提交数据包的内核API 451
17.5.2 从接收控制块（RCB）提交包 452
17.5.3 对ndisprot读请求的完成函数 454
17.5.4 读请求的发送 456
17.5.5 用于读包的WDF工作任务 457
17.5.6 ndisedge读工作任务的生成与入列 459
17.6 其他的特征回调函数的实现 461
17.6.1 包的归还 461
17.6.2 OID查询处理的直接完成 462
17.6.3 OID设置处理 465
第18章 NDIS中间层驱动 467
18.1 NDIS中间层驱动概述 467
18.1.1 Windows网络架构总结 467
18.1.2 NDIS中间层驱动简介 468
18.1.3 NDIS中间层驱动的应用 469
18.1.4 NDIS包描述符结构深究 470
18.2 中间层驱动的入口与绑定 473
18.2.1 中间层驱动的入口函数 473
18.2.2 动态绑定NIC设备 474
18.2.3 小端口初始化（MpInitialize） 475
18.3 中间层驱动发送数据包 477
18.3.1 发送数据包原理 477
18.3.2 包描述符“重利用” 478
18.3.3 包描述符“重申请” 481
18.3.4 发送数据包的异步完成 482
18.4 中间层驱动接收数据包 484
18.4.1 接收数据包概述 484
18.4.2 用PtReceive接收数据包 485
18.4.3 用PtReceivePacket接收 490
18.4.4 对包进行过滤 491
18.5 中间层驱动程序查询和设置 494
18.5.1 查询请求的处理 494
18.5.2 设置请求的处理 496
18.6 NDIS句柄 498
18.6.1 不可见的结构指针 498
18.6.2 常见的NDIS句柄 499
18.6.3 NDIS句柄误用问题 500
18.6.4 一种解决方案 502
18.7 生成普通控制设备 503
18.7.1 在中间层驱动中添加普通设备 503

18.7.2 使用传统方法来生成控制设备 505
第19章 IA-32汇编基础 511
19.1 x86内存、寄存器与堆栈 511
19.1.1 _asm关键字 511
19.1.2 x86中的mov指令 512
19.1.3 x86中的寄存器与内存 512
19.1.4 赋值语句的实现 513
19.2 x86中函数的实现 514
19.2.1 一个函数的例子 514
19.2.2 堆栈的介绍 515
19.2.3 寄存器的备份和恢复 516
19.2.4 内部变量与返回值 518
19.3 x86中函数的调用与返回 521
19.3.1 函数的调用指令call 521
19.3.2 通过堆栈传递参数 521
19.3.3 从函数返回 523
19.3.4 三种常见的调用协议 524
19.4 从32位汇编到64位汇编 526
19.4.1 Intel 64与IA-32体系架构简介 526
19.4.2 64位指令与32位指令 526
19.4.3 通用寄存器 527
19.5 64位下的函数实现 528
19.5.1 函数概览 528
19.5.2 32位参数的传递 529
19.5.3 64位参数与返回值 530
19.5.4 栈空间的开辟与恢复 531
第20章 IA-32体系中的内存地址 534
20.1 内存的虚拟地址 534
20.1.1 C语言中的内存地址 534
20.1.2 虚拟地址的构成 535
20.1.3 段的选择 536
20.2 全局描述符表和段描述符 538
20.2.1 全局描述符表 538
20.2.2 段类型 539
20.2.3 段寄存器与段选择子 540
20.2.4 64位模式下的段 541
20.3 分段编程实践 542
20.3.1 系统表寄存器的结构 542
20.3.2 在汇编语言中获取全局描述表的位置 543
20.3.3 调试范例：sgdt指令的错误使用 545
20.3.4 在64位下获得全局描述符表 547
20.4 线性地址基础 549
20.4.1 分页控制机制 550
20.4.2 线性地址的转换 551
20.4.3 混合页面大小 552
20.4.4 32位物理地址的页目录和页表项 552
20.5 各种特殊分页方式 555
20.5.1 PAE分页方式 555
20.5.2 PSE-36分页机制 558
20.5.3 IA-32e模式下的线性地址 559
20.6 分页编程实践 562
20.6.1 页目录和页目录指针表的获取 562
20.6.2 页表的获取 564
20.6.3 线性地址的结构 567

第21章 处理器权限级别切换 571
21.1 Ring0和Ring3权限级别 571
21.2 保护模式下的分页内存保护 572
21.3 分页内存不可执行保护 574
21.3.1 不可执行保护原理 574
21.3.2 不可执行保护的漏洞 575
21.3.3 上机实践 577
21.4 权限级别的切换 579
21.4.1 调用门及其漏洞 579
21.4.2 sysenter和sysexit指令 581
21.4.3 上机实践 583
第22章 IA-32体系结构中的中断 585
22.1 中断基础知识 585
22.1.1 中断描述符表 585
22.1.2 中断处理过程 587
22.1.3 64位模式下的中断处理机制 589
22.1.4 多核下的中断 589
22.2 Windows中断机制 593
22.3 中断编程实践 596
22.3.1 IDT Hook 596
22.3.2 巧用IDT Hook实现安全防护 598
第23章 Windows内核挂钩 601
23.1 系统服务描述符表挂钩 602
23.1.1 系统服务描述符表（SSDT） 602
23.1.2 系统服务描述符表挂钩的意图 603
23.1.3 寻找要挂钩的函数的地址 604
23.1.4 函数被挂钩的过程 605
23.1.5 具体实现的代码 606
23.2 函数导出表挂钩 608
23.2.1 内核函数的种类 608
23.2.2 挂钩IoCallDriver 610
23.2.3 对跳转地址进行修改 611
23.3 Windows 7系统下IofCallDriver的跟踪 612
23.4 Windows 7系统下内联挂钩 615
23.4.1 写入跳转指令并拷贝代码 615
23.4.2 实现中继函数 617
高 级 篇
第24章 Windows通知与回调 620
24.1 Windows的事件通知与回调 620
24.2 常用的事件通知 620
24.2.1 创建进程通知 621
24.2.2 创建线程通知 625
24.2.3 加载模块通知 626
24.2.4 注册表操作通知 629
24.3 Windows回调机制 636
24.3.1 回调对象 636
24.3.2 回调对象的创建 637
24.3.3 回调对象的注册 637
24.3.4 回调的通告 638
24.4 安全的死角，回调的应用 639
第25章 保护进程 640
25.1 内核对象简介 640
25.2 内核对象的结构 641
25.3 保护内核对象 642

25.3.1 处理对象的打开 643
25.3.2 处理句柄的复制 644
25.3.3 处理句柄的继承 646
25.4 进程的保护 652
25.4.1 保护原理 652
25.4.2 Vista以后的进程对象保护 654
25.4.3 进程的其他保护 655
附录A 如何使用本书的源码光盘 656
附录B 练习题 659
・・・・・・(收起)

Windows内核安全与驱动开发(含CD光盘1张)_下载链接1_

标签

windows

驱动

内核

操作系统

操作系统编程

计算机

develope

评论

本书第一评

各个方面都不错！但是不是入门书，想入门不推荐看这本。

计算机内核

Windows内核安全与驱动开发(含CD光盘1张)_下载链接1_

书评

Windows内核安全与驱动开发(含CD光盘1张)_下载链接1_