第一图书网, tushu007.com

<<木马技术揭秘与防御>>

图书基本信息

书名：<<木马技术揭秘与防御>>

13位ISBN编号：9787121134715

10位ISBN编号：7121134713

出版时间：2011-9

出版时间：电子工业

作者：赵玉明

页数：240

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

第一图书网, tushu007.com

<<木马技术揭秘与防御>>

内容概要

《网络安全入门与提高:木马技术揭秘与防御》以microsoft visual c++ 6.0
为编程工具，全面介绍了c/c++语言网络编程和socket编程的基本方法。
重点剖析了目前流行木马的编程方法，揭露了黑客木马编程技术内幕。
本书的特色在于从整体入手，先介绍木马程序的整体框架雏形，然后一步一步地深入分析木马常用的隐藏技术、管道技术、反弹技术、内核级rootkit技术、钩子（hook）技术及远程注入技术等，全书结合众多生动案例，环环相扣，多种编程方法对比分析，深入浅出，使读者既能够从中领悟到一些编程技巧，而且还可以根据这些案例进行研究。
书中提供的案例都经过作者编译通过，完整无误。

《网络安全入门与提高:木马技术揭秘与防御》适合网络管理人员及其他相关领域的专业技术人员、管理人员阅读，也可作为高等院校相关专业的教学参考书。

第一图书网, tushu007.com

<<木马技术揭秘与防御>>

书籍目录

第1章 特洛伊木马发展历史
1.1 什么是木马程序
1.2 木马一直在变异
1.3 国内木马进化史
第2章 基础知识
2.1 常见的木马编程技术
2.2 socket 编程技术
2.2.1 基于tcp的socket技术
2.2.2 基于udp的socket技术
2.2.3 socket 实例分析
2.3 c++语言编程介绍
2.3.1 c++程序结构
2.3.2 visual c++编程介绍
2.3.3 visual c++使用小技巧
第3章 一个简单的木马程序分析
3.1 mini木马的基本原理
3.2 搭建实验环境
3.2.1 配置虚拟机环境
3.2.2 测试mini木马的功能
3.3 mini木马程序剖析
3.4 mini类木马的防御策略
第4章 木马隐藏技术分析
4.1 隐藏技术——注册表启动
4.1.1 测试注册表加载型木马door
4.1.2 door木马程序剖析
4.1.3 加载注册表木马的防御
4.2 隐藏技术——服务级木马
4.2.1 测试服务级木马svchost
4.2.2 svchost木马程序剖析
4.2.3 服务级木马程序防范
4.3 隐藏技术——进程注入木马
4.3.1 测试进程注入木马inject
4.3.2 inject注入木马程序剖析
4.3.3 inject注入木马程序防范
4.4 隐藏技术——内核级rootkit
4.4.1 测试rootkit 木马
4.4.2 rootkit木马程序剖析
4.4.3 rootkit木马程序防范
第5章 木马控制技术分析
5.1 管道技术
5.1.1 双管道木马程序剖析
5.1.2 简化双管道木马程序剖析
5.2 反弹木马技术
5.2.1 反弹木马的原理
5.2.2 反弹木马程序剖析

第一图书网, tushu007.com

5.2.3 反弹木马的防范策略
5.3 端口重用技术
5.3.1 端口重用技术实现
5.3.2 端口重用的防范
5.4 钩子（hook）技术
5.4.1 钩子技术实现
5.4.2 钩子程序防范
第6章 经典木马程序大解析
6.1 下载者程序剖析
6.2 关机程序剖析
6.3 进程查杀程序剖析
6.4 获取主机详细信息的代码
6.5 获取主机ip地址
6.6 单线程tcp扫描器
6.7 多线程dos攻击程序
第7章 综合木马程序剖析
7.1 正向连接木马程序剖析
7.2 反弹并隐藏木马程序剖析
7.3 winshell木马程序剖析
第8章 木马的查杀
8.1 自启动木马的查杀
8.1.1 注册表的基本知识
8.1.2 开机自启动木马
8.1.3 触发式启动木马
8.1.4 自动播放启动木马
8.2 进程木马的查杀
8.2.1 windows xp启动过程
8.2.2 进程的查看
8.2.3 进程的隐藏
8.3 文件木马的查杀
8.3.1 文件的基本知识
8.3.2 文件的隐藏、查找、保护与删除
8.3.3 利用系统本身的规则隐藏文件

第一图书网, tushu007.com

章节摘录

版权页：插图：3.载入内核阶段在这一阶段，ntldr会载入windows XP的内核文件Ntoskrnl.exe，但这里仅仅是载入，内核此时还不会被初始化。
随后被载入的是硬件抽象层（hal.d11）。
硬件抽象层其实是内存中运行的一个程序，这个程序在Windows XP内核和物理硬件之间起到了桥梁的作用。
在正常情况下，操作系统和应用程序无法直接与物理硬件打交道，只有Windows内核和少量内核模式的系统服务可以直接与硬件交互。
而其他大部分系统服务及应用程序如果想要与硬件交互，就必须通过硬件抽象层进行。
硬件抽象层的使用主要有两个原因：第一，忽略无效甚至错误的硬件调用，如果没有硬件抽象层，那么硬件上发生的所有调用甚至错误都将会反馈给操作系统，这可能会导致系统不稳定，而硬件抽象层就像工作在物理硬件和操作系统内核之间的一个过滤器，可以将认为会对操作系统产生危害的调用和错误全部过滤掉，这样直接提高了系统的稳定性；第二，多平台之间的转换翻译，这个原因可以列举一个形象的例子，假设每个物理硬件都使用不同的语言，而每个操作系统组件或者应用程序则使用了同样的语言，那么不同物理硬件和系统之间的交流将会是混乱而且很没有效率的。
如果有了硬件抽象层，等于给软／硬件之间安排了一位翻译，这位翻译懂所有硬件的语言，并会将硬件说的话用系统或者软件能够理解的语言原意转达给操作系统和软件。
通过这个机制，操作系统对硬件的支持可以得到极大的提高。

第一图书网, tushu007.com

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com