

<<C++黑客编程揭秘与防范>>

图书基本信息

书名：<<C++黑客编程揭秘与防范>>

13位ISBN编号：9787115280640

10位ISBN编号：7115280649

出版时间：2012-6

出版时间：人民邮电出版社

作者：冀云

页数：265

字数：406000

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>

<<C++黑客编程揭秘与防范>>

内容概要

《C++黑客编程揭秘与防范》旨在通过简单的语法知识及常用的系统函数编程，完成一些有特定功能的安全工具，让读者对系统知识等各方面有一个全面的了解，并且在笔者的带领下一步步完成书中的实例。

本书主要内容为：第1章了解黑客编程，主要讲解了VC(Visual C++的缩写)和Windows下安全编程方面的基础知识。

第2章从剖析简单的木马说起，讲解有关的网络编程和协议知识。

第3章

Windows应用编程基础，讲解API编程的技术。

第4章加密与解密，讲解PE等加密有关的知识。

第5章

HOOK编程，讲解了与钩子有关的知识。

第6章

黑客编程剖析，剖析了病毒的原理和攻防技术，以及安全工具的开发。

第7章最后的旅程——简单驱动开发及逆向。

《C++黑客编程揭秘与防范》适合网络安全人员、黑客爱好者，以及相关的程序员阅读。

<<C++黑客编程揭秘与防范>>

书籍目录

第1章 黑客编程入门

- 1.1 编程语言和开发环境的选择
 - 1.1.1 何为SDK、API和MFC
 - 1.1.2 VC6和SDK的配置
- 1.2 应用程序的调试
 - 1.2.1 编写我们的第一个程序
 - 1.2.2 用VC6调试第一个程序
 - 1.2.3 专业的应用程序调试工具——OllyDbg
- 1.3 简单API的介绍
 - 1.3.1 复制自身程序到Windows目录和系统目录下
 - 1.3.2 获得系统的相关信息
 - 1.3.3 Debug和Release的编译方式
 - 1.3.4 查看函数定义
- 1.4 总结

第2章 木马开发剖析

- 2.1 网络通信基础
 - 2.1.1 IP地址的作用与分类
 - 2.1.2 端口的作用与分类
- 2.2 网络编程基础知识
 - 2.2.1 通信模型
 - 2.2.2 Winsock
 - 2.2.3 Winsock的相关函数
 - 2.2.4 字节顺序
- 2.3 简单的通信程序
 - 2.3.1 基于TCP协议的“Hello World!”
 - 2.3.2 基于UDP协议的“Hello World!”
- 2.4 实现一个C/S模式的简单木马
 - 2.4.1 木马服务器端的实现
 - 2.4.2 木马客户端的实现
- 2.5 总结

第3章 Windows应用编程基础

- 3.1 文件
 - 3.1.1 打开文件
 - 3.1.2 文件操作
- 3.2 AutoRun免疫程序的编写
 - 3.2.1 AutoRun免疫原理
 - 3.2.2 AutoRun免疫程序的代码实现
 - 3.2.3 界面设置
 - 3.2.4 代码相关部分
- 3.3 注册表操作
 - 3.3.1 注册表
 - 3.3.2 与注册表操作相关的常用API函数
 - 3.3.3 注册表启动项的管理
 - 3.3.4 程序的界面设置及相关代码
 - 3.3.5 启动项的枚举

<<C++黑客编程揭秘与防范>>

- 3.3.6 添加启动项的代码
 - 3.3.7 删除启动项的代码
 - 3.4 服务相关的编程
 - 3.4.1 如何查看系统服务
 - 3.4.2 服务控制管理器的开发
 - 3.4.3 枚举服务的相关API函数
 - 3.4.4 服务的停止
 - 3.4.5 停止服务的相关API函数
 - 3.4.6 服务的启动
 - 3.5 进程与线程
 - 3.5.1 进程
 - 3.5.2 进程的创建
 - 3.5.3 “下载者”的简单演示
 - 3.5.4 CreateProcess()函数介绍与程序创建
 - 3.5.5 进程的结束
 - 3.5.6 进程的枚举
 - 3.5.7 调整当前进程的权限
 - 3.5.8 进程的暂停与恢复
 - 3.5.9 多线程
 - 3.6 DLL编程
 - 3.6.1 什么是DLL
 - 3.6.2 编写一个简单的DLL程序
 - 3.6.3 对DLL程序的调用方法一
 - 3.6.4 对DLL程序的调用方法二
 - 3.7 远程线程
 - 3.7.1 DLL注入
 - 3.7.2 DLL卸载
 - 3.7.3 无DLL的代码注入
 - 3.8 总结
- 第4章 加密与解密
- 4.1 PE文件结构
 - 4.1.1 PE文件结构全貌
 - 4.1.2 MZ头部
 - 4.1.3 PE头部
 - 4.1.4 节表
 - 4.1.5 节表数据
 - 4.2 详解PE文件结构
 - 4.2.1 DOS头部详解IMAGE_DOS_HEADER
 - 4.2.2 PE头部详解IMAGE_NT_HEADERS
 - 4.2.3 IMAGE_FILE_HEADER
 - 4.2.4 IMAGE_OPTIONAL_HEADER
 - 4.2.5 节区详解IMAGE_SECTION_HEADER
 - 4.2.6 与PE结构相关的3种地址
 - 4.2.7 3种地址的转换
 - 4.3 PE查看器
 - 4.4 简单的查壳工具
 - 4.5 地址转换器

<<C++黑客编程揭秘与防范>>

- 4.6 添加节区
 - 4.6.1 手动添加一个节区
 - 4.6.2 通过编程添加节区
- 4.7 破解基础知识及调试API函数的应用
 - 4.7.1 CrackMe程序
 - 4.7.2 用OD破解CrackMe
- 4.8 文件补丁及内存补丁
 - 4.8.1 文件补丁
 - 4.8.2 内存补丁
- 4.9 调试API函数的使用
 - 4.9.1 常见的3种断点方法
 - 4.9.2 调试API函数及相关结构体介绍
 - 4.9.3 判断是否处于被调试状态
 - 4.9.4 断点异常函数
 - 4.9.5 调试事件
 - 4.9.6 调试循环
 - 4.9.7 内存的操作
 - 4.9.8 线程环境相关API及结构体
- 4.10 打造一个密码显示器
- 4.11 总结
- 第5章 HOOK编程
 - 5.1 HOOK知识前奏
 - 5.2 内联钩子——Inline Hook
 - 5.2.1 Inline Hook的原理
 - 5.2.2 Inline Hook的实现
 - 5.2.3 HOOK MessageBoxA
 - 5.2.4 HOOK CreateProcessW
 - 5.2.5 7字节Inline Hook
 - 5.2.6 Inline Hook的注意事项
 - 5.3 导入地址表钩子——IAT HOOK
 - 5.3.1 导入表简介
 - 5.3.2 导入表的数据结构定义
 - 5.3.3 手动分析导入表
 - 5.3.4 枚举导入地址表
 - 5.3.5 IAT HOOK介绍
 - 5.3.6 IAT HOOK之CreateFileW()
 - 5.4 Windows钩子函数
 - 5.4.1 钩子原理
 - 5.4.2 钩子函数
 - 5.4.3 键盘钩子实例
 - 5.4.4 使用钩子进行DLL注入
 - 5.5 总结
- 第6章 黑客编程剖析
 - 6.1 恶意程序剖析
 - 6.1.1 恶意程序的自启动
 - 6.1.2 木马的配置生成与反弹端口
 - 6.1.3 代码实现剖析

<<C++黑客编程揭秘与防范>>

- 6.2 简单病毒剖析
 - 6.2.1 病毒的感染剖析
 - 6.2.2 缝隙搜索的实现
 - 6.2.3 感染目标程序文件剖析
 - 6.2.4 添加感染标志
 - 6.2.5 自删除功能的实现
- 6.3 隐藏DLL文件
 - 6.3.1 启动WinDBG
 - 6.3.2 调试步骤
 - 6.3.3 编写枚举进程中模块的函数
 - 6.3.4 指定模块的隐藏
- 6.4 安全工具开发基础
 - 6.4.1 行为监控工具开发基础
 - 6.4.2 专杀工具
 - 6.4.3 U盘防御软件
 - 6.4.4 目录监控工具
- 6.5 引导区解析
 - 6.5.1 通过WinHex来手动解析引导区
 - 6.5.2 通过程序解析MBR
 - 6.5.3 自定义MBR的各种结构体
 - 6.5.4 解析MBR的程序实现
- 6.6 加壳与脱壳
 - 6.6.1 手动加壳
 - 6.6.2 编写简单的加壳工具
- 第7章 最后的旅程——简单驱动开发及逆向
 - 7.1 驱动版的“Hello World”
 - 7.2 驱动下的进程遍历
 - 7.2.1 配置VMware和WinDbg进行驱动调试
 - 7.2.2 EPROCESS和手动遍历进程
 - 7.2.3 编程实现进程遍历
 - 7.3 HOOK SSDT(系统服务描述表)
 - 7.3.1 SSDT简介
 - 7.3.2 HOOK SSDT
 - 7.3.3 Inline HOOK SSDT
 - 7.4 应用程序与驱动程序的通信
 - 7.4.1 创建设备
 - 7.4.2 应用程序与驱动程序的通信方式
 - 7.4.3 应用程序与驱动程序的通信实例
 - 7.5 C语言代码逆向基础
 - 7.5.1 函数的识别
 - 7.5.2 if...else...分支结构
 - 7.5.3 switch分支结构
 - 7.5.4 for循环结构
 - 7.5.5 do...while与while...循环结构
- 参考文献

章节摘录

版权页：插图：从图6—6中看到了反弹木马的工作原理。

通常情况下攻击者的IP地址是变动的，那么“小白”是如何连接到“黑客”的主机的呢？

一般情况下黑客要把自己的IP地址动态地保存到某个固定的IP地址下（比如保存到网上FTP空间中），然后木马通过读取该IP地址下保存的黑客的IP地址进行连接，同样用图来说明，如图6—7所示。

从图6—7中可以看出，黑客开启木马客户端后，首先会更新服务器上保存着的自己的IP地址。

“小白”会去读取服务器中保存着的黑客的IP地址，然后“小白”去连接“黑客”的主机，主动地让黑客去控制它，这就是木马中的“自动上线”。

关于反弹端口的介绍就到这里。

有了思路，通过前面学习的Wmsock的知识自己可以试着实现一下，这里就不做更多的介绍了。

二、木马的配置生成与配置信息的保护 木马写好以后，通常会发布一个程序，在木马程序中通过配置一些相关的内容和参数后，会生成一个木马的服务器端程序。

为什么木马的客户端会生成木马的服务端程序呢？

其实木马的客户端和服务端本来就是两个程序，只是通过某种方式使其成为了一个文件而已。

让木马的服务端和客户端成为一个文件可以有多种方法，常见的有资源法和文件附加数据法两种。

在PE文件结构中有一个数据目录称作资源，资源可以是图片、图标、音频、视频等内容。

资源法也就是把服务端以资源的形式连接到客户端的程序中，然后客户端通过一些操作资源的函数将资源读取出来并生成文件。

文件附加数据法是将服务端保存到客户端的末尾，然后通过文件操作函数，直接将服务端读取出来并生成新的文件。

反弹端口连接是要访问某个固定的IP地址去读取保存着黑客的动态IP地址的信息，而这个固定的IP地址是保存在木马程序中的。

也就是说，我们的客户端在把服务端生成以后，会把一些配置信息写入服务端程序的指定位置中，服务端程序会读取指定位置的信息来进行使用。

配置信息的写入与读出必须要一致，否则就没有意义了。

对于配置信息中往往会存在一些比较敏感的信息，比如邮箱账号、密码等内容。

比如，我们在分析盗QQ的木马时会发现接收QQ密码的邮箱，由于现在很多邮箱都需要SMTP的验证，因此在配置信息中也会看到邮箱的账号及密码信息。

这样配置信息中的这些敏感信息很容易被人获取到，甚至接收QQ密码邮箱的账号和密码也会被别人获取到，真是“偷鸡不成蚀把米”。

对于此类情况，正确的做法是对配置信息进行加密。

也就是说客户端往服务端中写配置信息前需要加密后再写入，而服务端在使用这些信息前需要先解密再使用。

关于配置生成客户端与配置信息的保护上面已经介绍得差不多了，接下来应该把重点放在代码的实现上了。

我们的代码是模拟实现上面的内容，而不是真的去生成木马。

<<C++黑客编程揭秘与防范>>

编辑推荐

- 1、讲解windows安全和网络编程知识；
- 2、讲解内核编程以及软件逆向知识；

<<C++黑客编程揭秘与防范>>

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>