A password viewer program with Visual Basic

邓双成　焦向东　张沛
(北京石油化工学院机械工程系　102600)

用VB编写密码查看程序

摘要：本文论述了用Visual Basic编写密码查看程序的方法，给出了具体的程序实例。

关键词：Visual Basic　密码

1 引言

网络越来越普及是一件好事，它可以带来很大的方便，但有时也会带来不少麻烦。譬如要记住那么多的密码，就不是一件容易的事。笔者申请的免费EMAIL帐号就有七八个，如果每次上网收邮件时都输入这么多的密码确实令人发憷，因此我采取的是一劳永逸的办法，就是让The Bat记住这些帐号和密码。这样上网时就不用一一输入每个帐号的密码了，但时间长了容易忘记密码。网上有一个叫revelll.exe (Revelation v1.1)的软件，能够查看那些********所代表的密码明文，但revelll运行时"占地"太大，WIN98的桌面被它的主窗口占了四分之一之多，操作起来不很方便。针对这种情况，笔者用VB编写了一个类似的软件。

这个用VB编写的软件只有一个文件PassView.exe。运行PassView.exe，按"查看密码"按钮，再在要查看的密码密文(一般为********)上单击，密码密文********就显示为明文。下面两图是用PassView.exe查看"拨号网络"的密码。图1中的密码显示为********，真正的密码显示在图2中，为"dengdeng"。由图可见，PassView.exe所占的桌面空间非常小。(注意：每次查看新的密码前都要先按"查看密码"按钮，然后再单击要查看的密码)。

下面详述在编写此软件的过程中遇到的几个难点。

2 显示密码明文的原理

正如大家在"拨号网络"中所看到的那样，输入密码一般是利用TextBox控件(也叫Edit控件)。TextBox控件实际上也是一个窗口，有自己的窗口Style。输入密码用的TextBox控件的Style中，其EN-PASSWORD这一位被置位，这样在输入密码时，密码明文就显示不出来了，而是用********代替。这样做可以防止输入密码时被人偷窥。

如上所叙，要显示密码明文似乎很简单，只要能从密码所在的TextBox控件的Style中去掉EN-PASSWORD风格就行。但笔者在试图这么做时却未成功。

好在还有另外一种办法，可以让TextBox控件中的密码明文"原形毕露"。只要找到此TextBox控件的窗口句柄hWnd，我们就可以向其发送一条EM_SETPASSWORDCHAR消息，让WINDOWS自动显示密码明文。

向某个窗口发送消息要用到如下函数：

```
Declare Function SendMessage Lib" user32" -
Alias "SendMessageA"(By Val hWnd As Long,-
ByVal wMsg As Long,-
ByVal wParam As Long,IParam As Any) As Long
```

在发送EM-SETPASSWORDCHAR消息时，只要让wParam和IParam都为0，就可以显示密码明文。由于IParam被声明为Any，故IParam应设为ByVal 0&。实际的语句如下：

```
SendMessage hWnd,EM_SETPASSWORDCHAR,
0,ByVal 0&
```

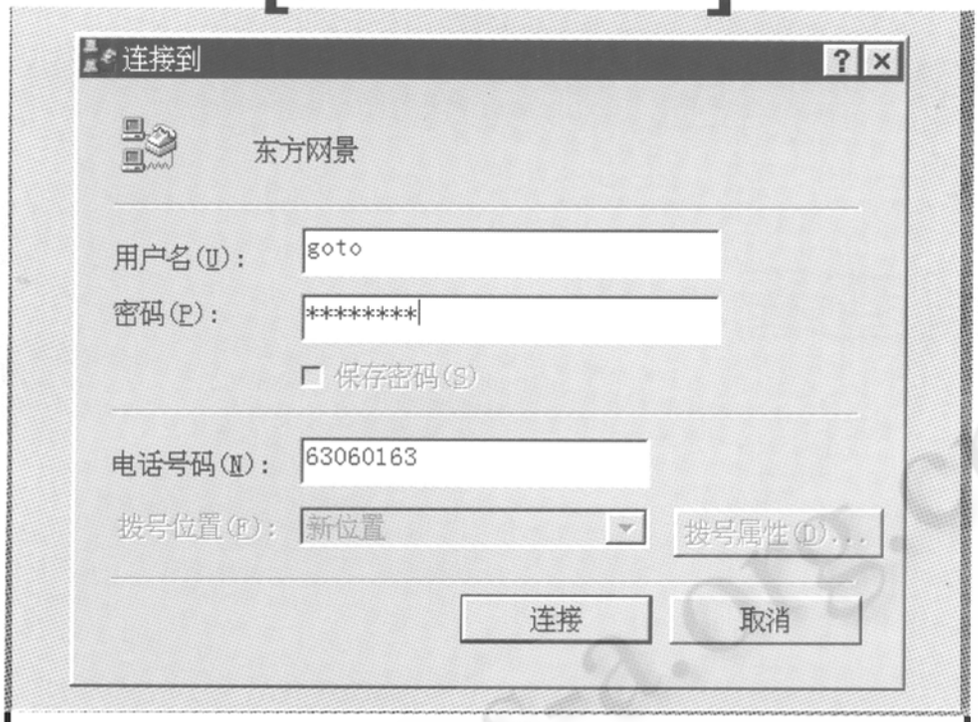

图 1 查看密码前

3 确定密码所在TextBox控件的窗口句柄

为了使程序具有通用性，我们必须让用户能够以某种方式找到密码所在TextBox控件的窗口句柄hWnd。此程序采取的方法是让用户按"查看密码"按钮后，单击密码所在TextBox控件来确定用户要显示的密码。

PassView程序通过WM—LBUTTONUP消息来截获用户单击密码所在TextBox控件的动作。在WM-LBUTTONUP消息处理程序中，可以通过GetCursorPos函数来获取用户单击动作所在的(x,y)坐标，然后用WindowFromPoint函数来取得比(x,y)所属的TextBox控件的窗口句柄hWnd。

这里存在着一个问题：密码所在的TextBox控件肯定位于别的窗口之内，也即位于PassView程序主窗体之外。PassView程序主窗体通常只能截获到发生在其范围之内的鼠标消息，如何才能让它截获到用户对密码所在TextBox的单击动作呢?这就要通过鼠标捕捉来实现。

4 捕捉鼠标

此程序的鼠标捕捉是在按钮控件Command1的Click事件中完成的。通过用SetCapture函数捕捉鼠标，可以让PassView程序截获发生在其主窗体之外的所有鼠标消息。这样它就能截获到用户对密码所在TextBox的单击动作，并根据单击动作发生时的(x,y)坐标确定TextBox的窗口句柄，向其发送EM-SETPASSWORDCHAR消息以显示密码明文。一旦完成密码明文显示，就应该用ReleaseCapture函数解除鼠标捕捉状态。

虽然PassView可以截获到发生在其主窗体范围之外的WM-LBUTTONLP消息了，但在Form-Click、Form-MouseDown、Form-MouseUp等事件过程里是收不到此消息了。因此，必须通过窗口子类化技术来截获此消息并对其进行处理。

5 窗口子类化技术

关于窗口子类化技术的资料很多，笔者就不在此重复了。VB中窗口子类化的实现可以借助于一些控件(如Message Blaster、Message Hook等)来实现，也可以用VB自带的AddressOf语句来实现。此程序在标准模块Pass View.bas中定义了一个新的窗口函数，用AddressOf取得其地址后，可以用它替换VB预设的窗口函数。替换工作是通过调用SetWindowLong函数完成的。程序结束时应再次调用此函数恢复VB预设的窗口函数，否则会造成系统崩溃。

对象	属性	值
Form	Caption	Password Viewer
	ControlBox	False
Command1	Caption	查看密码
Command2	Caption	退出

表1

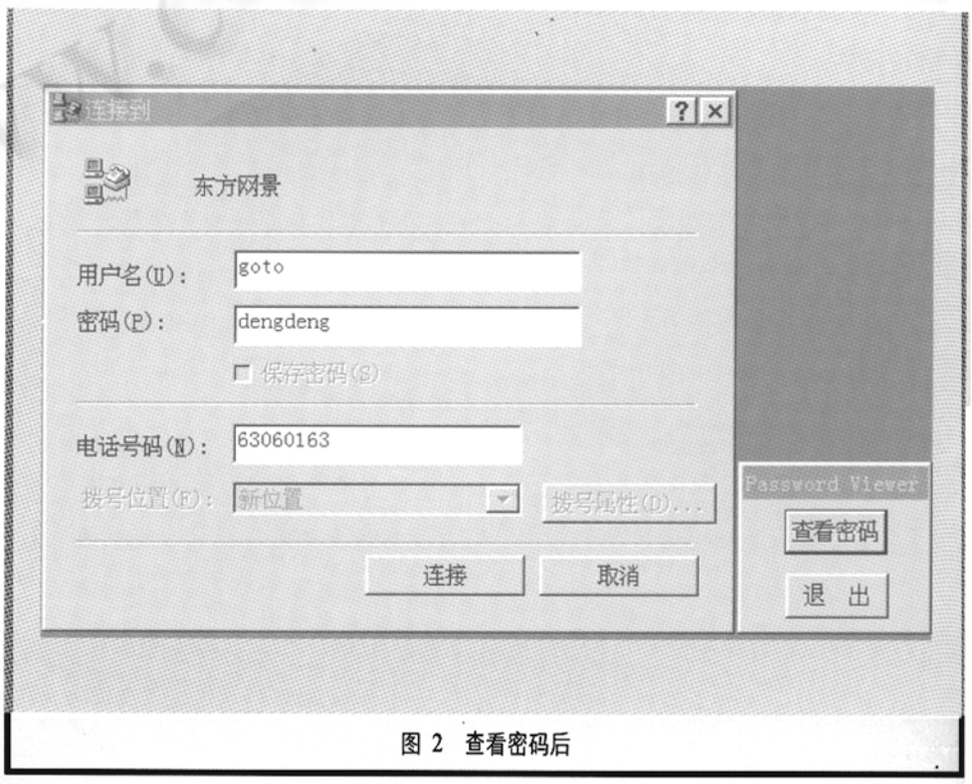

图 2 查看密码后

6 程序清单

此程序有一个标准窗体(PassView.frm)和一个标准模块(Pass View.bas)。

6.1 窗体

窗体PassView.frm只有两个按钮控件。

6.2 窗体中(PassView.frm)的代码

```
Option Explicit
Private Sub Form-Load()
bMouseCaptured=False
' 用新的窗口过程替换原来的窗口过程 OldWndProc=GetWindowLong (Me.hwnd,GWL-WNDPROC)
SetWindowLong Me.hwnd,GWL-WNDPROC,AddressOfNewWndProc
End Sub

Private Sub Form-Unload (Cancel As Integer)
' 恢复原来的窗口过程
SetWindowLong Me.hwnd,GWL-WNDPROC,OldWndProc
End Sub

Private Sub Command1-Click()
' 捕捉鼠标
SetCapture Me.hwnd
bMouseCaptured=True
End Sub

Private Sub Command2_Click()
' 结束程序
UnloadMe End Sub
```

6.3 标准模块(PassView.bas)中的代码

```
Option Explicit
' 用于将Form1子类化
Declare Function GetWindowLong Lib" -user32" -
Alias "GetWindowLongA" -
(By Val hwnd As Long,By Val nIndex As Long) As Long
Declare Function SetWindowLong Lib "user32" -
Alias" SetWindowLongA" -
(By Val hwnd As Long,By Val nIndex As Long,-
By Val dwNewLong As Long)As Long
Declare Function CallWindowProc Lib
ser32" -Alias "CallWindowProcA" -
(By Val lpPrevWndFunc As Long,
ByVal hwnd As Long,-
By Val Msg As Long,-
By Val wParam As Long,ByVal lParam As Long) As Long
Public Const GWL-WNDPROC=(-4)
'要截取的鼠标消息
Public Const WM-LBUTTONUP=&H202

' 要截取的鼠标捕捉
Declare Function SetCapture Lib" user32" -
(By Val hwnd As Long) As Long
Declare Function ReleaseCapture Lib" user32"() As Long
'用于得到光标所在的位置的屏幕坐标
Declare Function GetCursorPos Lib "user32" (lpPoint As POINTAPI) As Long
Type POINTAPI
x As Long
y As Long
End Type
' 用于获取鼠标所在位置的窗口的句柄
Declare Function WindowFromPoint Lib" ser32" -
(By Val xPoint As Long,By Val yPoint As Long) As Long
'Declare Function ClientToScreen Lib" user32" -
(By Val hwnd As Long,lpPoint As POINTAPI) As Long
'用于发送消息的API的函数
Declare Function SendMessage Lib "user32" -
Alias "SendMessageA"(By Val hwnd As Long,- By Val wMsg As Long,-
By Val wParam As Long,lParam As Any) As Long Public Const EM-SETPASSWORDCHAR=&HCC
Public Const EM-SETSEL=&HB1
' 声明全局变量
Public OldWndProc As Long '保存原来的窗口过程
Public bMouseCaptured As Booleam
' 保存鼠标的捕捉状态
' 新的窗口过程
Public Function NewWndProc (By Val hwnd As Long,- By Val wMsg As Long,-
By Val wParam As Long,-
ByVallParam As Long) As Long
Select Case wMsg
Case Wm-LBUTTONUP '截获鼠标单击消息
Dim p As POINTAPI
Dim hWindow As Long
'释放鼠标
IF bMouseCaptured Then
ReleaseCapture
bMouseCaptured=False
End If
'GetCursorPos函数得到的光标的位置总是屏幕坐标,
' 不受包含光标的窗口的映射方式的影响
GetCursorPosp
'得到(x,y)处的窗口句柄
hWindow=WindowFromPoint(p.x,p,y)
If h Window<>0 Then
向此窗口发送EM-SETPASSWORDCHAR消息,以显示密码
SendMessage h Window,EM-SETPASSWORD-CHAR,0,By,Val 0&
' 再发送如下的消息,以保证立即显示密码
'(若不发送此消息,有时似乎要双击才能显示密码)
SendMessage h Window,EM-SETSEL,0,By Val-1
End If Exit Function End Select
' 将所有未截取的其他消息传递给原来的窗口过程处理
New WndProc=CallWindowProc (OldWndProc,hwnd,wMsg,wParam,lParam)
End Function
```


参考文献

1 (美) Evangelos Petroutsos、Kevin Hough 著,Visual Basic6 高级开发指南,电子工业 出版社,1999。