How to control/monitor Webrelay with Visual Basic 2005

1. Introduction

Webrelay can be monitored and controlled through the use of TCP and XML. TCP is used to create a network connection with the Webrelay unit, and XML is the language used to communicate with it. Visual Basic 2005 comes with a new class called System.Net.Sockets, which makes it easier to setup a TCP connection and communicate over it.

2. Handling the TCP connection

In order to communicate with Webrelay, the Visual Basic application needs to do four things:

1 - The application needs to establish a TCP connection with Webrelay
2 - The application needs to send a XML message to Webrelay
3 - The application needs to wait for a response from Webrelay
4 - The application needs to close the TCP connection with Webrelay

Visual Basic 2005 comes with a class that handles these four tasks. It is called System.Net.Sockets. To use this class in your application, make sure to import it at the beginning of the code file.

```
Imports System.Net.Sockets
```

Once imported into the project, establishing a TCP connection is as simple as declaring a tcpClient object and calling the objects open method with Webrelay's IP address and port number.

```
Dim tcpClient As New System.Net.Sockets.TcpClient()

'Connect to webrelay
tcpClient.Connect("192.168.1.2", 80)
```

After a connection has been established, sending and receiving messages from Webrelay is achieved by creating a Network Stream object that points to the tcpClient object's network stream, and using it's read and write methods.

```
'Create a network stream object
Dim netStream As NetworkStream = tcpClient.GetStream()
```

```
'Create the XML command to send to Webrelay
Dim sendBytes As [Byte]() = Encoding.ASCII.GetBytes("GET
/state.xml&noReply=0 HTTP/1.1" & vbCrLf & "Authorization: Basic
bm9uZTp3ZWJyZWxheQ==" &              vbCrLf & vbCrLf)

'Send the command to webrelay
netStream.Write(sendBytes, 0, sendBytes.Length)

'Get the response from webrelay
Dim bytes(tcpClient.ReceiveBufferSize) As Byte
netStream.Read(bytes, 0, Cint(tcpClient.ReceiveBufferSize))
```

Notice the line where the XML request is generated. The constant vbCrLf is used to add the appropriate carriage return and line feed characters to the command so Webrelay will recognize it. More explanation of the XML request can be found in part 3.

Once a response has been received from Webrelay, we can parse through it to find the current state of the unit. In our example, we update two text boxes with the state of the Webrelay unit. The current state of the relay is found at an offset of 66 bytes from the beginning of the response, and the current state of the input is found at an offset of 94 bytes from the beginning of the response.

```
'Convert the bytes received from Webrelay into a string
Dim returndata As String = Encoding.ASCII.GetString(bytes)

'Convert the string into and array
Dim array1 As Char() = returndata.ToCharArray()

'Relay State found at index 66 of array
If array1(66) = "1" Then
     relayState.Text = "ON"
Else
     relayState.Text = "OFF"
End If

'Input State found at index 94 of array
If array1(94) = "1" Then
     inputState.Text = "ON"
Else
     inputState.Text = "OFF"
End If
```

When we are done communicating with Webrelay, we close the TCP connection using the tcpClient objects method close.

```
'Close the connection
tcpClient.Close()
```

3. Format of XML request.

Note: The following notation is used to indicate a carriage return and line feed in the XML requests.

<CR> = Carriage return (hex 0d)
<LF> = Line feed (hex 0a)

These characters aren't normally visible, but they must be present for correct operation of the Webrelay unit.

There are 4 different XML request that are recognized by Webrelay :

1. To retrieve the state of the Webrelay unit, send the following

GET /state.xml HTTP/1.1<CR><LF>
Authorization: Basic bm9uZTp3ZWJyZWxheQ==<CR><LF><CR><LF>

2. To turn the relay of the Webrelay unit on, send the following

GET /state.xml?relayState=1 HTTP/1.1<CR><LF>
Authorization: Basic bm9uZTp3ZWJyZWxheQ==<CR><LF><CR><LF>

3. To turn the relay of the Webrelay unit off, send the following

GET /state.xml?relayState=0 HTTP/1.1<CR><LF>
Authorization: Basic bm9uZTp3ZWJyZWxheQ==<CR><LF><CR><LF>

4. To pulse the relay of the Webrelay unit, send the following

GET /state.xml?relayState=2 HTTP/1.1<CR><LF>
Authorization: Basic bm9uZTp3ZWJyZWxheQ==<CR><LF><CR><LF>

Note: The Second Line of each of the XML requests above specifies the password to be used. The password needs to be in base64 form. For example, the password "webrelay" is "bm9uZTp3ZWJyZWxheQ==" in base64 form. To encode a password, go to the webpage http://www.controlbyweb.com/encoder.html.

4. Format of XML response

Each XML response from Webrelay is 157 bytes long. Below is an example XML response from a webrelay unit whose input is off, and relay state is on.

```
<?xml version='1.0' encoding='utf-8'?>
<datavalues>
<relaystate>1</relaystate>
<inputstate>0</inputstate>
<rebootstate>0</rebootstate>
</datavalues>
```

Each XML response begins with a header that indicates the version of XML used and the character encoding used. Next, there is a tag labeled <datavalues>. It is in between the tags <datavalues> and </datavalues> that the state of Webrelay can be found. The tags <relaystate></relaystate> surround the byte that signifies the relay state of the webrelay unit.

1 = RELAY ON
0 = RELAY OFF

Similarly, the tags <inputstate></inputstate> surround the byte that signifies the input state of the webrelay unit.

1 = INPUT ON
0 = INPUT OFF

The relay state is found at an offset of 66 bytes from the beginning of the response. The input state is found at an offset of 94 bytes from the beginning of the response.