**w02057's CrackMe#2 by w02057**

Site: [http://www.crackmes.de/users/w02057/crackme2_by_w02057/](http://www.crackmes.de/users/w02057/crackme2_by_w02057/)

**Solution by costy**

When you load the crackme you can notice that the clue button is disabled.

Inside Reflector you can see

```vbnet
Private Sub clue_textbox_Click(ByVal sender As Object, ByVal e As EventArgs)
    Interaction.MsgBox(Me.clue_textbox.Text, MsgBoxStyle.Information, "Clue")
End Sub
```

So it should be interesting to see the text inside `clue_textbox`.

Let's see this text.

*In the InitializeComponent you can see:*

```vbnet
point = New Point(9, &H88)
Me.clue_textbox.Location = point
Me.clue_textbox.Name = "clue_textbox"
size = New Size(&HF4, 13)
Me.clue_textbox.Size = size
Me.clue_textbox.TabIndex = 6
Me.clue_textbox.Text = "The number is between 10000000 and 99999999."
```

"The number is between 10000000 and 99999999." is a great help. Without it solving the crackme is impossible. (Too much time for bruteforcing).

The "point" indicates when the text appears. You can see the text enlarging the window.
Ok. Let's see how the check works.

The text you type should be a number. In fact, the secret help suggests to type a number between 10000000 and 99999999. If you type a number, the crackme creates two strings: 'str' is a constant "abcdefghijklmnopqrstuvwxyz". In a first time, 'sString' is equal to (number_you_type+57842967). Then the program appends to the end of this string some letters from 'str'. Each mid instruction gets a letter from 'str'. The second parameter of mid is the position of the letter. The third parameter is the length (it takes only a character). Then the crackme generates a value from the string using the sha function. It inserts 4 "-" in this string in the position 4,9,14,19.

It compares this value with the string in the combobox. If they are equal, you win.

In order to solve the crackme, I coded a brute-forcer.

It works in a simply way.

Instead of taking the name from a textbox, it has a loop and try every number from 10000000 and 99999999. Then it compares the number generate with sha with each number in the combobox.

I added the source to the zip file.

Anyway the valid couple are:

60135393 E9A1-CFE4-A2D6-D19B-E1E2
76032106 F68A-8002-D1FD-1BF6-E6C4
59568847 6F96-AFB3-2A2D-F692-1A27
34134732 5FE6-10E5-FF3F-01AE-232E
10820813 E04B-D618-90EF-373B-307D
22107279 6C22-0B1F-7C4A-6ADE-2796