

Forum:[Hack Forums](#)

Thread: [Hide UI of any process!](#)

SIM-4704770 - May 28, 2020, 11:44 pm

I found this method when making my hvnc module as I wanted to hide multiple windows that popup. This is very simple and I will be showing how to fully implement this, as you can use this to hide anything from task managers UI to web browsers that you don't have access to. This can be very useful for blocking certain apps from showing their UI and should be used only for Educational Purposes.

Lets begin, first you need these references.

```
Code using System;
using System.Diagnostics;
using System.Runtime.InteropServices;
using System.Threading;
```

Now here is how to hide process that are running

```
Code [DllImport("User32.dll")]
private static extern int ShowWindow(int hWnd, int nCmdShow);

public static void Hide(string processName)
{
    try
    {
        int hWnd;
        Process[] processRunning = Process.GetProcesses();
        foreach (Process pr in processRunning)
        {
            if (pr.ProcessName == processName)
            {
                hWnd = pr.MainWindowHandle.ToInt32();
                ShowWindow(hWnd, 0);
            }
        }
    }
    catch (Exception)
    {
    }
}
```

Here is some examples for how to call the code correctly.

```
Code Hide("explorer");
//Or if you want to have it hide explorer every time it is relaunched.
Task.Run(() =>
{
    while(1==10
    {
        Thread.Sleep(1);
        Hide("explorer");
    }
}
```

```
}
```

});This loop method is very crude but it does work and doesn't lag at all, even on a 2 core rdp with only 4gb of ram!

Anyways this is a full tutorial on how to paste a working hide method. Hope you learn something from this!

Depending on the feedback I receive I will be creating more content!

VisualSoft - May 29, 2020, 3:40 am

Nice share buddy, Keep up the good work.

Xamarin - May 29, 2020, 4:07 am

(May 28, 2020 11:44 PM)MarkSkith4 Wrote: [\[CITE\]](#) I found this method when making my hvnc module as I wanted to hide multiple windows that popup.

This is very simple and I will be showing how to fully implement this, as you can use this to hide anything from task managers UI to web browsers that you don't have access to. This can be very useful for blocking certain apps from showing their UI and should be used only for Educational Purposes.

Lets begin, first you need these references.

```
Code using System;
using System.Diagnostics;
using System.Runtime.InteropServices;
using System.Threading;
```

Now here is how to hide process that are running

```
Code [DllImport("User32.dll")]
private static extern int ShowWindow(int hwnd, int nCmdShow);

public static void Hide(string processName)
{
    try
    {
        int hwnd;
        Process[] processRunning = Process.GetProcesses();
        foreach (Process pr in processRunning)
        {
            if (pr.ProcessName == processName)
            {
                hwnd = pr.MainWindowHandle.ToInt32();
                ShowWindow(hwnd, 0);
            }
        }
    }
    catch (Exception)
    {
    }
}
```

Here is some examples for how to call the code correctly.

```
Code Hide("explorer");
//Or if you want to have it hide explorer every time it is relaunched.
Task.Run(() =>
```

```
{  
while(1==10  
{  
Thread.Sleep(1);  
Hide("explorer");  
}
```

}); This loop method is very crude but it does work and doesn't lag at all, even on a 2 core rdp with only 4gb of ram!

Anyways this is a full tutorial on how to paste a working hide method. Hope you learn something from this!

Depending on the feedback I receive I will be creating more content!
Two things I would like to say about your code.

- 1) It would be better if your hide() function actually returned true or false upon success/failure, so that your try/catch actually has a meaningful existence other than an exception suppressor.
- 2) Instead of running this computationally expensive loop every time, it would be a good idea to use [WMI's event watcher](#), so that you can run your code on an event basis.

Other than that, good job.

SIM-4704770 - May 29, 2020, 7:39 am

Thanks for the tips Xamarin! This is just a crude way of doing it that works for me.

MavenGradle - May 29, 2020, 8:06 am

You are using a native Windows function. This Windows function has a return value. I would suggest that you actually take that into account and start using it:

Quote:Return value

Type: BOOL

If the window was previously visible, the return value is nonzero.

If the window was previously hidden, the return value is zero.

<https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-showwindow>

The proper return type in C# would be a boolean, not an int. Granted, a boolean can be considered as an integer that can only have the value 0 or 1, but since you are working in C# you should implement the proper return type.

In addition, the ShowWindow function takes in an IntPtr as the handle to the window.:

<https://docs.microsoft.com/en-us/dotnet/api/system.intptr?view=netframework-4.8>

You are, additionally, 'catching them all' so to speak which is one of the worst practices to utilize when it comes to programming. You should, instead, only catch the exceptions that are relevant to your use case and utilize them (by for example logging or returning a value to the caller of your function). Here's what I mean:

Code try

```
{  
// Your error-prone code code here  
}  
catch(SubTypeException ex)  
{  
// Do something with the exception  
}
```

The ShowWindow function, however, does not return an error if it cannot perform its functionality because of, for example, an incorrect handle..

I'm going to assume that you made a typo (actually multiple in the following code block): You are also creating an infinite loop and utilizing Thread.Sleep() , Two prime examples of bad programming practices in essentially the same block of code (assuming while (1 == 10 should be while(10 == 10)).

```
Code while(1==10
```

```
{  
Thread.Sleep(1);  
Hide("explorer");  
}
```

The implementation of this 'infinite loop' (if my assumptions are correct) means that the process of your application will never be able to shut down gracefully. You should never use infinite loops. Thread.Sleep(1) is completely pointless, you are introducing an artificial waiting period of 1 millisecond on every single iteration, which makes the process unresponsive even. You might be hiding it using a Task but why? To hide your bad practices? This entire code block is an exercise in pointlessness.

However, if you really did mean to type while (1 ==10:

Your code won't even compile and even if you added the extra bracket that's missing from your example, it would never be executed because obviously 1 does not equal 10 and as such you have just delivered us with 5 useless lines of code that should be removed. Actually, even if my assumption about the infinite loop is correct, those 5 lines should be removed from existence, regardless of the errors you made. See the reasons stated above if you want to know why.

This isn't good code by any standards. Please correct it.

SIM-4704770 - May 29, 2020, 9:02 am

Thanks for the tips and response, I need feedback like this to help me become a better programmer and not make mistakes like this in the future.

MavenGradle - May 29, 2020, 9:05 am

(May 29, 2020 09:02 AM)MarkSkith4 Wrote: [\[CITE\]](#) Thanks for the tips and response, I need feedback like this to help me become a better programmer and not make mistakes like this in the future.

You're welcome. Thank you for taking it like a champion. Good luck and feel free to share other code snippets you have right here. If you need help, don't hesitate to ask!

SIM-4704770 - May 29, 2020, 9:31 am

Thank you!

I appreciate you taking the time to fix some of my mistakes, I have been coding c# applications for about 1 year now and still have a ways to go before I get to where you are.

SaketSaket - May 29, 2020, 9:33 am

keep sharing buddy... nice work...

Shadow_Cloud - May 29, 2020, 9:33 am

(May 29, 2020 09:31 AM)MarkSkith4 Wrote: [\[CITE\]](#) Thank you!

I appreciate you taking the time to fix some of my mistakes

This is the right mindset to have to learn things. keep it up :)