

Forum:[Hack Forums](#)

Thread: [Anti debugger/sniffer bypasser](#)

Crakxproo - June 21, 2018, 8:59 am

```
Code    private void ChangeProcName(object sender, EventArgs e)
        {
            var p = Process.GetProcessById(Convert.ToInt32(textBox1.Text));
            while(true)
            {
                Thread.Sleep(100); // Set this to 1 For Antis / Or just delete it
                SetWindowText(p.MainWindowHandle, textBox2.Text);
            }
        }
```

Uses SetWindowText API From kernel32

The Delay Is there for reduced memory usage
Should be easy to convert this to C/Nim for extra speed

So if you were to start fiddler and use this, on a app, run it unless it does multiple checks

Dotfuscator employs Managed/nmanaged deugger checks at the beginning in case your retarded

Pengers - June 21, 2018, 10:43 am

I don't see the reason to repeat it every 100 milliseconds, you said it was because it will be memory intensive, but if the title reset, it could be like that for over 99 milliseconds which I'm sure gives the program you're attempting to bypass enough time to detect it.

:thunk:

Crakxproo - June 21, 2018, 11:00 am

(Jun 21, 2018 10:43 AM)Pengers Wrote: [\[CITE\]](#) I don't see the reason to repeat it every 100 milliseconds, you said it was because it will be memory intensive, but if the title reset, it could be like that for over 99 milliseconds which I'm sure gives the program you're attempting to bypass enough time to detect it.

:thunk:

Without the delay, That will never happen, especially if you convert it to C like i did..

Or Intercept Title Changing code like my gf did

Menalix - June 23, 2018, 8:04 am

Don't loop, it's such an easy hook "SetWindowText", you just need to ret a bool and ignore the parameters. then the program won't change its title again.

Crakxproo - June 23, 2018, 8:08 am

(Jun 23, 2018 08:04 AM)Menalix Wrote: [\[CITE\]](#) Don't loop, it's such an easy inline hook

"SetWindowText", you just need to ret a bool and ignore the parameters. then the program won't change its title again.

Thanks my gf

Washi - June 23, 2018, 9:30 am

Right...

Why is your method accepting two parameters that you are not using anywhere in the program? Is this an event handler of a Windows Forms Application or something? Then why is there an infinite loop in an event handler? Good luck freezing your UI thread.

In your code you also assume that we have a few textboxes. One containing the target process ID and one containing the new text for the main window.

Your source code also lacks quite a bit of error handling. I wonder what happens when I enter anything but an int32 in that first textbox, or a non-existing process ID...

The SetWindowText DllImport is missing from your "tutorial".

Only really weak anti debugger techniques query the text of all opened windows to see if there is a blacklisted application running. I am afraid your anti anti is not that effective :/.

Your code is in C#, but you posted in the VB section

Also, why do you keep on telling everyone in all your threads that you can easily convert, or that you already have converted to C as well? How is that relevant? Are you trying to show off here or something? You are in the .NET section, not in the C section. Oh well good luck with translating it to C and hogging your CPU with a nice infinite loop ?

Spoiler ([Click to View](#))

Let me guess, this source code was also meant for skids?

Crakxproo - June 23, 2018, 9:35 am

(Jun 23, 2018 09:30 AM)Washi Wrote: [\[CITE\]](#) Right...

Why is your method accepting two parameters that you are not using anywhere in the program? Is this an event handler of a Windows Forms Application or something? Then why is there an infinite loop in an event handler? Good luck freezing your UI thread.

In your code you also assume that we have a few textboxes. One containing the target process ID and one containing the new text for the main window.

Your source code also lacks quite a bit of error handling. I wonder what happens when I enter anything but an int32 in that first textbox, or a non-existing process ID...

The SetWindowText DllImport is missing from your "tutorial".

Only really weak anti debugger techniques query the text of all opened windows to see if there is a blacklisted application running. I am afraid your anti anti is not that effective :/.

Your code is in C#, but you posted in the VB section

Also, why do you keep on telling everyone in all your threads that you can easily convert, or that you already have converted to C as well? How is that relevant? Are you trying to show off here or something? You are in the .NET section, not in the C section. Oh well good luck with translating it to C and hogging your CPU with a nice infinite loop ?

Spoiler ([Click to View](#))

Let me guess, this source code was also meant for skids?

it was a snippet, plus i used a loop with a delay because most private debuggers i have dont work without the loop

Menalix - June 23, 2018, 12:10 pm

(Jun 23, 2018 09:35 AM)Crakxproo Wrote: [\[CITE\]](#) it was a snippet, plus i used a loop with a delay because most private debuggers i have dont work without the loop

If it was snippet, you shouldn't include UI elements and use inf loops.

Instead you should do a clean func, with parameters for the text and selected process.

That way people can use the function however they want to, eventually decide for themself if the function should be within a inf loop, or if they want to use textboxes.

EdwardSn0wdent - June 23, 2018, 2:35 pm

(Jun 23, 2018 12:10 PM)Menalix Wrote: [\[CITE\]](#) If it was snippet, you shouldn't include UI elements and use inf loops.

Instead you should do a clean func, with parameters for the text and selected process.

That way people can use the function however they want to, eventually decide for themself if the function should be within a inf loop, or if they want to use textboxes.

What standard of snippet says you can not use controls...

Menalix - June 23, 2018, 5:13 pm

(Jun 23, 2018 02:35 PM)EdwardSn0wdent Wrote: [\[CITE\]](#) What standard of snippet says you can not use controls...

None, but for his post it did not make sense, if it really was just a snippet as he says.

It's a standard that snippets is not longer than necessary, and that it should only contain the important. Because that is what a snippet is.

It is not important when changing the window title to use a textbox control, it's important to have a string for the SetWindowText parameter.

In most cases you would inline the string value or just give the function a string parameter. Same goes for the PID.

It would have made more sense like this, if it was a snippet, and not a badass application he thought he had made:

```
private void ChangeProcName(string title, int pid)
{
    var p = Process.GetProcessById(pid);
    SetWindowText(p.MainWindowHandle, title);
}
```

His comment about changing sleep(100) to sleep(1), clearly shows his intention was for people to directly use his code, aka not a snippet.

Just because his application is shit, he can't excuse it by saying: it was a snippet.

Instead he should have said: Thanks Washi for learning me something, that was nice of you.

