

Forum:[Hack Forums](#)

Thread: [\[INFO\] How To Protect Your Source Against Reflectors VB.NET/C# \[INFO\]](#)

shan - April 29, 2013, 12:16 am

???

Okay, so i found this pretty easy way of changing your PE which results in a reflector that is unable to open your file. I tried uploading a modded PE to VirusTotal, but seems it results in 2 false positives. Found out even a blank .NET PE file will also set off alarm to those 2 specific AVs. So I think this wouldn't cause a false positive on a random project you want to protect.

Into the PE header

So i found a specific header in the .NET PEs, which is default a byte with the value of 16. This byte is found in every .NET exe at the 244th byte. So just did some more research and found out when you change this byte into for example 15, a .NET reflector isn't able to read the .NET PE. I tried some more values, 0 to 5 will corrupt the exe file, 6 till 15 will do what we want, and higher than 16 I haven't tried, so that's up to you :].

Lets get started

So we want to create an app that changes that 244th byte, which is pretty easy.

Create a function/sub and add this little code into that function/sub [VB.NET]:

?

```
Code Dim tmpStream As New IO.FileStream("Path to your .NET PE", IO.FileMode.Open, IO.FileAccess.Write)
```

```
'We only need write permissions since we are only writing a single byte.
```

```
tmpStream.Seek(244, IO.SeekOrigin.Begin)
```

```
'This will place the file stream at the 244th byte, so we can overwrite that one
```

```
tmpStream.WriteByte(6)
```

```
'Writing our 244 byte in the PE file
```

```
tmpStream.Close
```

```
'Closing our file streamCode System.IO.FileStream tmpStream = new System.IO.FileStream(@"Path to your .NET PE",
```

```
System.IO.FileMode.Open, System.IO.FileAccess.Write);
```

```
tmpStream.Seek(244, System.IO.SeekOrigin.Begin);
```

```
tmpStream.WriteByte(6);
```

```
tmpStream.Close();
```

So as you can see, this is pretty easy. Now lets see the results:

before:

[img]<http://img26.imageshack.us/img26/7929/beforels.jpg>[img]

After Inserting The Code?

[img]<http://img607.imageshack.us/img607/397/afterb.jpg>[img]

As you can see, after i changed that PE header, the reflector can't open the file, however it does run as you can see ???

Hope you learned something and please let me know what you think ??

?

Sc?rf?c3_ - April 29, 2013, 12:20 am

i was wondering what exactly was BN.NET?

shan - April 29, 2013, 12:25 am

sorry my big mistake ..itz acutally vb.net ?

Washi - April 29, 2013, 12:33 am

Why do I think I have seen this tutorial some time ago already?

OP do not copy and paste tutorials.

Btw: this method is not working for almost all decompilers (including my decompiler)

AirKode - April 29, 2013, 12:45 am

As TUP said, this is a very old technique and only breaks Reflector. I'm also pretty sure that most people on this forum already know how to fix this.

Washi - April 29, 2013, 12:56 am

Besides that, if you want to use this method, you should handle 64 bit applications as well. The optional header of a 64 bit application differs from the optional header of a 32 bit application. The method you've posted does only work with 32 bit compiled applications.

li0nsar3c00l - April 29, 2013, 1:49 am

(Apr 29, 2013 12:33 AM)TheUnknownProgrammer Wrote: [\[CITE\]](#) Why do I think I have seen this tutorial some time ago already?

OP do not copy and paste tutorials.

Btw: this method is not working for almost all decompilers (including my decompiler)

In other words: net reflector is the only decompiler i know that gets fooled by thir threadyou obviously copy pasted

shan - April 29, 2013, 1:50 am

Tup .. yea thats correct ,, :) ,,,,

Mohatu - May 4, 2013, 7:14 am

Copied from here: <http://leetcoders.org/showthread.php?tid=364>

Give credit.