

Forum:[Hack Forums](#)

Thread: [\[C#\] Efficient Read EoF Function](#)

Zer0Fact0r - April 7, 2011, 8:57 am

This is essentially a stripped down version of the code in this thread (with the exception that this will work on .NET assemblies): <https://hackforums.net/showthread.php?tid=1191361>

I put this together a long time ago. It works and is much more memory efficient than other EoF functions I've seen here.

```
Code public static byte[] ReadEoF(string fileName)
{
    byte[] buffer = null;
    using (FileStream fs = new FileStream(fileName, FileMode.Open))
    {
        buffer = new byte[4];
        fs.Position = 0x3C;
        fs.Read(buffer, 0, 4);
        fs.Position = BitConverter.ToInt32(buffer, 0) + 0x6;
        buffer = new byte[2];
        fs.Read(buffer, 0, 2);
        fs.Position += 0x100 + ((BitConverter.ToInt16(buffer, 0) - 1) * 0x28);
        buffer = new byte[8];
        fs.Read(buffer, 0, 8);
        fs.Position = BitConverter.ToInt32(buffer, 0) + BitConverter.ToInt32(buffer, 4);
        buffer = new byte[fs.Length - fs.Position];
        fs.Read(buffer, 0, buffer.Length);
    }
    return buffer;
}
```

Example Usage:

```
Code File.AppendAllText("lol.exe", "123456");
Console.WriteLine(Encoding.ASCII.GetString(ReadEoF("lol.exe")));
// writes "123456" to the console
```

Vysse - April 7, 2011, 9:03 am

Very nice share this is quite the small snippet

DragQueen - April 7, 2011, 10:58 am

i tries my code on process explorer and i got 3229 byte[] with urs i got 6008 byte[]
the question is what function is not work my / urs

no1me - April 7, 2011, 11:03 am

his code working just fine i check it

DragQueen - April 7, 2011, 11:18 am

my mistake work on .net files Omg
and process explorer is not .net file

EDIT :: i liked this code i added it to my class. with source link + right credit.

mnbv - April 7, 2011, 11:58 am

Wow thanks very much, this is very short and efficient

But how can I use this in my crypter? First checking EOF and, if there is if then: ???
And also doing something with the EOF in the Stub?

After a quick search (btw Im using CodeDom, but it should be the same)

The steps: read EOF, crypt file, write it back to the file.

And quote of Vorfin

"Well, since EoF Preservation is a Function on the Builder, not the Stub, it doesn't matter if you use CodeDom or not. "

Zer0Fact0r - April 7, 2011, 12:15 pm

(Apr 7, 2011 10:58 AM)DragQueen Wrote: [\[CITE\]](#) i tries my code on process explorer and i got 3229 byte[] with urs i got 6008 byte[]
the question is what function is not work my / urs
Your (ported) code loads the contents of the executable into memory (File.ReadAllBytes). Mine doesn't.

mnbv - April 7, 2011, 12:33 pm

Build is crypting the file

Read Bytes, Convert to string, Encrypt, Write with CodeDom

Edit: this is not the good way

Code //Byte[] norm_file = System.IO.File.ReadAllBytes(i.Text); // Reads all bytes <This worked

Byte[] norm_file = ReadEoF(i.Text);

Can anyone offer some help?

DragQueen - April 7, 2011, 1:15 pm

(Apr 7, 2011 12:15 PM)Zer0Fact0r Wrote: [\[CITE\]](#) Your (ported) code loads the contents of the executable into memory (File.ReadAllBytes). Mine doesn't.

its still not explain the diffrent in the bytes []

in .net file it work like charm ... in other native files

i see big diffrent [6000 and 3000 its big diffrents]

and string its also weird ... with my class i get readable string

with this func i get unreadable string

ur methood work only on .net files, check it and see

mu methood work only with native with .net it fail

Zer0Fact0r - April 7, 2011, 3:11 pm

(Apr 7, 2011 01:15 PM)DragQueen Wrote: [\[CITE\]](#) its still not explain the diffrent in the bytes []

in .net file it work like charm ... in other native files

i see big diffrent [6000 and 3000 its big diffrents]

and string its also weird ... with my class i get readable string

with this func i get unreadable string

ur methood work only on .net files, check it and see

mu methood work only with native with .net it fail

I did... and it works fine...

You're testing it wrong.

mnbv - April 10, 2011, 12:07 pm

Anybody that can refere me or post a writeeof function that belongs to this read function?
Else, Its useless

no1me - April 10, 2011, 12:14 pm

```
File.AppendAllText("lol.exe", "123456");  
  
Console.WriteLine(Encoding.ASCII.GetString(ReadEoF("lol.exe")));  
  
// writes "123456" to the console
```

mnbv - April 10, 2011, 12:20 pm

This is with a console, but a crypted server for example is different, Actually thats the thing im searching (for days)
Thanks for the unput though ..
Anyone?

no1me - April 10, 2011, 12:28 pm

write to file
File.AppendAllText("lol.exe", "123456");
read the eof
msgbox(Encoding.ASCII.GetString(ReadEoF("lol.exe")));

the same for any file

mnbv - April 10, 2011, 12:37 pm

But what you are basically doing now is:
Make a file with name: lol.exe and eof: 123456

Then you read it and show it in a msgbox, console or whatever.

but this is not writing to a crypted file on the desktop for example right?

```
File.AppendAllText("cryptedfile.exe", "ReadedEofdata");  
If its like above , I get it
```

no1me - April 10, 2011, 12:46 pm

delete all ur reply and put just this

```
File.AppendAllText("cryptedfile.exe", "ReadedEofdata");  
If its like above , I get it
```

mnbv - April 10, 2011, 1:40 pm

Hmm its very strange and not working >.<
byte[] EOFbytes = ReadEoF(i.Text);
EOFstring = System.Text.Encoding.ASCII.GetString(EOFbytes);

```
if (checkBox1.Checked)
{
File.AppendAllText(Path, EOFstring);
}
```

EDIT LINE: Path is the path with SFD

I crypted several things and each output size stays exactly the same which is result of failed EOF-adding I think

no1me - April 10, 2011, 1:44 pm

u really dont know what u doing

```
File.AppendAllText(Path, EOFstring);
```

u add the same "EOFstring" and u want to get different output

mnbv - April 10, 2011, 1:50 pm

Im not adding the same EOFstring, because as you can see in my previous post it says ReadEof(i.Text) --> i.Text = the path of the file that I want to crypt.

And for writing it:

```
File.AppendAllText(Path, EOFstring); --> Path is the Path of the SFD e.g. Build(sfd.FileName);
```

Ill edit previous post Path have to be SFD instead of OFD, sorry my mistake

But then still .. it should work

no1me - April 10, 2011, 2:10 pm

```
EOFstring = ReadEof(i.Text)
```

u add to path (= i.Text) the EOFstring

```
File.AppendAllText(Path, EOFstring);
```

mnbv - April 10, 2011, 2:19 pm

This is not possible because ReadEof outputs a byte and not a string. Its public static byte
public static byte[] ReadEoF(string fileName)

so thats why I used get string

Vysse - April 10, 2011, 4:30 pm

I converted this to VB for my own purposes. I also created a small function to auto check if eof is needed...

```
Code Public Shared Function ReadEoF(ByVal fileName As String) As Byte()
Dim buffer As Byte() = Nothing
Using fs As New IO.FileStream(fileName, IO.FileMode.Open)
buffer = New Byte(3) {}
fs.Position = &H3C
fs.Read(buffer, 0, 4)
fs.Position = BitConverter.ToInt32(buffer, 0) + &H6
buffer = New Byte(1) {}
fs.Read(buffer, 0, 2)
fs.Position += &H100 + ((BitConverter.ToInt16(buffer, 0) - 1) * &H28)
buffer = New Byte(7) {}
fs.Read(buffer, 0, 8)
fs.Position = BitConverter.ToInt32(buffer, 0) + BitConverter.ToInt32(buffer, 4)
```

```

    buffer = New Byte(fs.Length - fs.Position - 1) {}
    fs.Read(buffer, 0, buffer.Length)
End Using
Return buffer
End Function
Code Public Sub AutoEOF()
    Dim EOFCheck As Byte() = ReadEoF(OFD.FileName)
    If EOFCheck.Length > 0 Then MessageBox.Show("EOF Found!") Else MessageBox.Show("EOF
Not Found!")
    End Sub

```

I am sure most can do this on their own but I figured why not share... Enjoy!

Zer0Fact0r - April 10, 2011, 4:40 pm

mnbv Wrote:Anybody that can refere me or post a writeeof function that belongs to this read function?

Else, Its useless (Apr 10, 2011 02:19 PM)mnbv Wrote: [\[CITE\]](#) This is not possible because ReadEof outputs a byte and not a string. Its public static byte
public static byte[] ReadEoF(string fileName)

so thats why I used get string

Are you retarded? You can convert the bytes to a string, and it will be like any other EoF reading function that you can copy and paste into your copy and pasted crypter. I provided an example in my original post and I believe someone else posted it again on the second page of this thread. What don't you understand?

But - this is what I would do (since why convert the data to a string if you don't have to):

```

Code whateverTheHellYouWant void WriteEoF(string fileName, byte[] data)
{
    using (FileStream fs = new FileStream(fileName, FileMode.Append, FileAccess.Write))
    {
        fs.Write(data, 0, data.Length);
    }
}

```

inb4"Invalid token 'void' in class, struct, or interface member declaration"

iamhere - April 10, 2011, 9:11 pm

Read if the server has Eof:

```

Code public string ReadEofData(string filePath)
{
    var readFile = File.ReadAllText(filePath);
    var IPos = (Strings.StrReverse(readFile)).IndexOf(GetNullBytes(30)) + 1;
    var eofBuf = Strings.StrReverse(readFile).Substring(0, Convert.ToInt32(IPos - 1));
    var returnValue = Strings.StrReverse(eofBuf);
    if (returnValue == "")
    {

    }
    return returnValue;
}

```

Usage:

```
Code TextBox1.Text = ReadEOFData(TextFile.Text)
    ReadEOFData(TextFile.Text)
```

mnbv - *April 11, 2011, 6:54 am*

lol the EOF of mine worked, but I had something else to fix
Thanks anyway