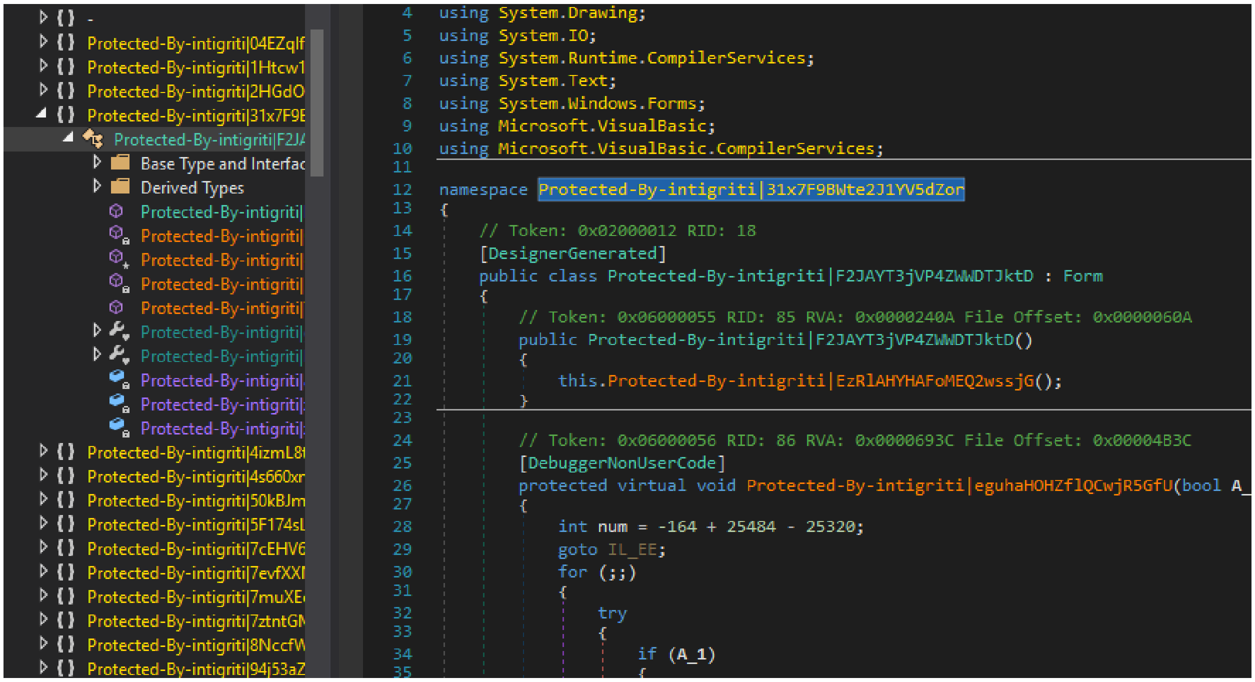
SadPower

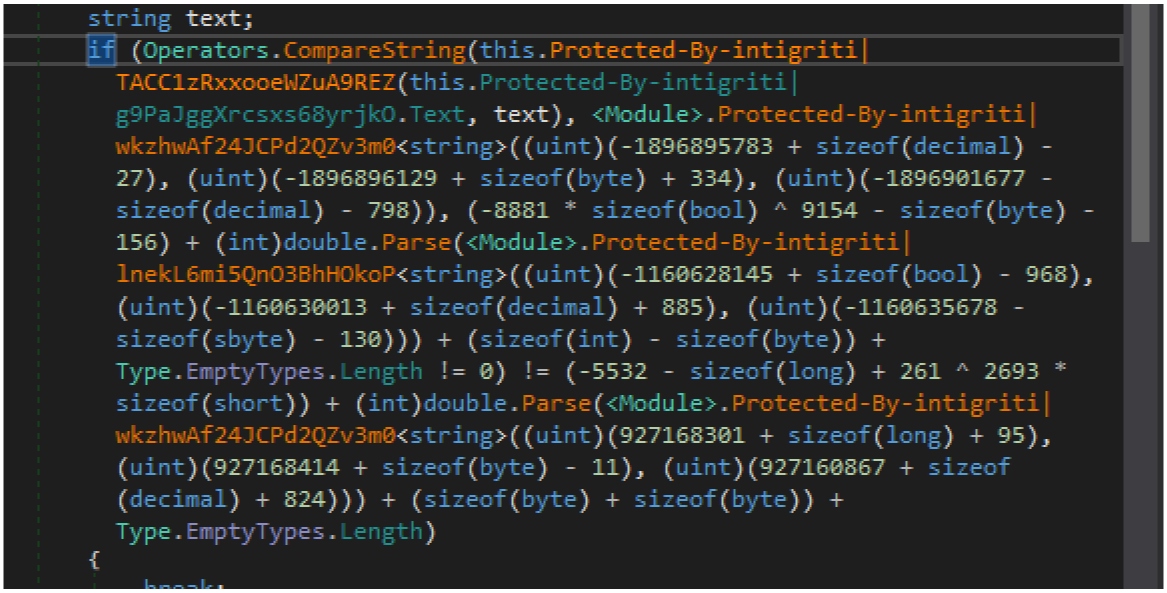
In this challenge the exe will have to decrypt the key to compare it with the input, so the main challenge in this exe is trying to understand and navigate an obfuscated .net sample to find a key to decrypt the enc file.

Looking through the exe the only real classes are Protected-By-intigriti|31x7F9BWte2J1YV5dZor and settings


-
Protected-By-intigriti|04EZqlf
Protected-By-intigriti|1Htcw1
Protected-By-intigriti|2HGdO
Protected-By-intigriti|31x7F9E
Protected-By-intigriti|F2JA
Base Type and Interfac
Derived Types
Protected-By-intigriti|
Protected-By-intigriti|
Protected-By-intigriti|
Protected-By-intigriti|
Protected-By-intigriti|
Protected-By-intigriti|
Protected-By-intigriti|
Protected-By-intigriti|
Protected-By-intigriti|
Protected-By-intigriti|
Protected-By-intigriti|4izmL8
Protected-By-intigriti|4s660xr
Protected-By-intigriti|50kBJm
Protected-By-intigriti|5F174sL
Protected-By-intigriti|7cEHV6
Protected-By-intigriti|7evfXXI
Protected-By-intigriti|7muXE
Protected-By-intigriti|7ztntGN
Protected-By-intigriti|8NccfW
Protected-By-intigriti|94j53aZ

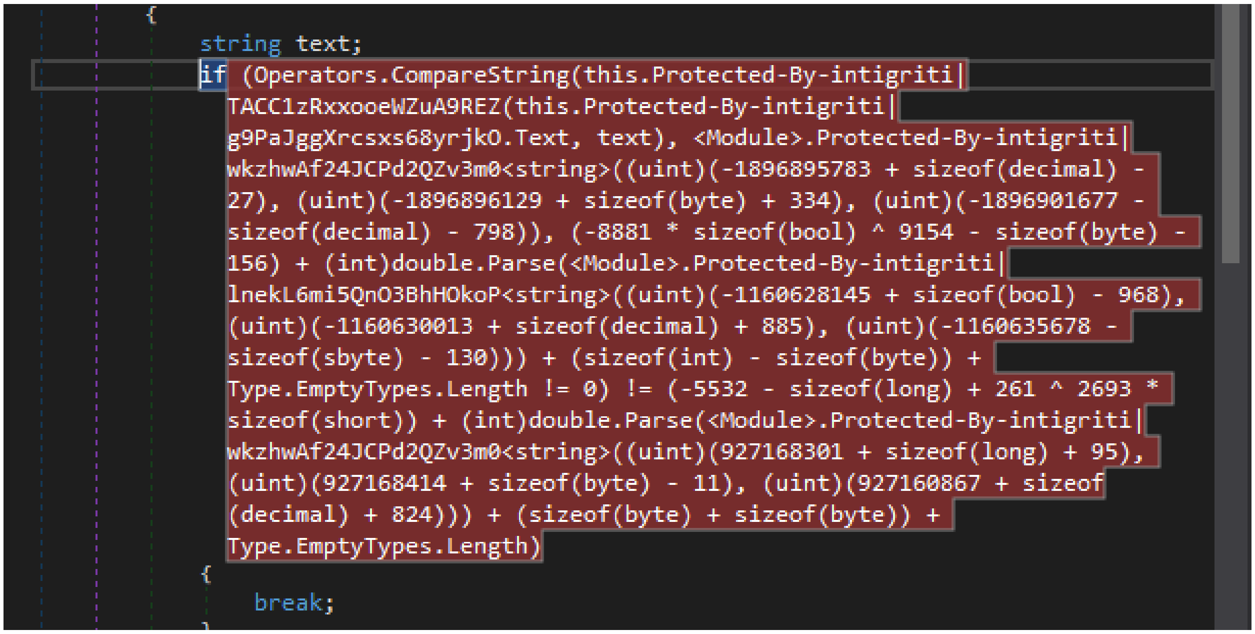
```
using System.Drawing;
using System.IO;
using System.Runtime.CompilerServices;
using System.Text;
using System.Windows.Forms;
using Microsoft.VisualBasic;
using Microsoft.VisualBasic.CompilerServices;

namespace Protected-By-intigriti|31x7F9BWte2J1YV5dZor
{
    // Token: 0x02000012 RID: 18
    [DesignerGenerated]
    public class Protected-By-intigriti|F2JAYT3jVP4ZWWDTJktD : Form
    {
        // Token: 0x06000055 RID: 85 RVA: 0x0000240A File Offset: 0x0000060A
        public Protected-By-intigriti|F2JAYT3jVP4ZWWDTJktD()
        {
            this.Protected-By-intigriti|EzRlAHYHAFoMEQ2wssjG();
        }

        // Token: 0x06000056 RID: 86 RVA: 0x0000693C File Offset: 0x00004B3C
        [DebuggerNonUserCode]
        protected virtual void Protected-By-intigriti|eguhaHOHZflQCwjR5GfU(bool A_
        {
            int num = -164 + 25484 - 25320;
            goto IL_EE;
            for (;;)
            {
                try
                {
                    if (A_1)
                    {
```

When looking through 31x7F9BWte2J1YV5dZor we search for any point where the exe compares and try to see if any strings appear.



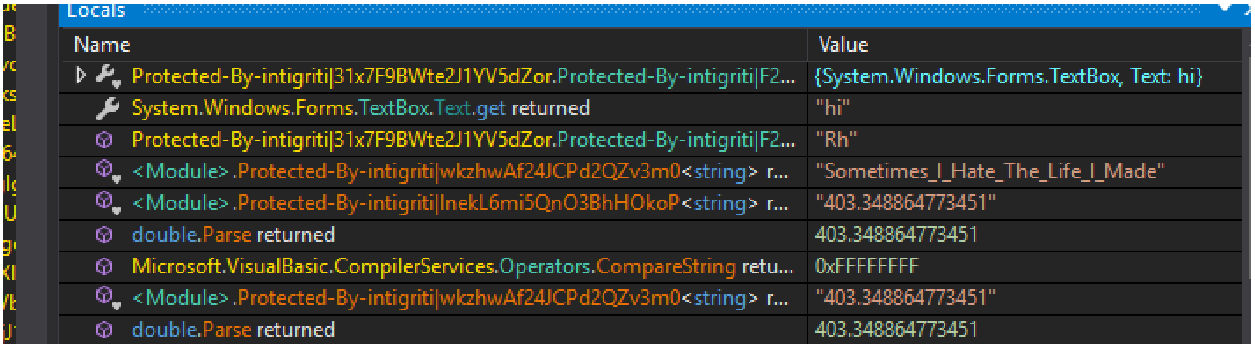
```
string text;
if (Operators.CompareString(this.Protected-By-intigriti|
  TACC1zRxxooeWZuA9REZ(this.Protected-By-intigriti|
  g9PaJggXrcsxs68yrjkO.Text, text), <Module>.Protected-By-intigriti|
  wkzhwAf24JCPd2QZv3m0<string>((uint)(-1896895783 + sizeof(decimal) -
  27), (uint)(-1896896129 + sizeof(byte) + 334), (uint)(-1896901677 -
  sizeof(decimal) - 798)), (-8881 * sizeof(bool) ^ 9154 - sizeof(byte) -
  156) + (int)double.Parse(<Module>.Protected-By-intigriti|
  lnekL6mi5QnO3BhHOkoP<string>((uint)(-1160628145 + sizeof(bool) - 968),
  (uint)(-1160630013 + sizeof(decimal) + 885), (uint)(-1160635678 -
  sizeof(sbyte) - 130))) + (sizeof(int) - sizeof(byte)) +
  Type.EmptyTypes.Length != 0) != (-5532 - sizeof(long) + 261 ^ 2693 *
  sizeof(short)) + (int)double.Parse(<Module>.Protected-By-intigriti|
  wkzhwAf24JCPd2QZv3m0<string>((uint)(927168301 + sizeof(long) + 95),
  (uint)(927168414 + sizeof(byte) - 11), (uint)(927160867 + sizeof
  (decimal) + 824))) + (sizeof(byte) + sizeof(byte)) +
  Type.EmptyTypes.Length)
{
    break;
```

We find the above portion in the method Protected-By-intigriti|Chmp9auSrtjuD5cVeNQW where it tries to compare between strings and we place a breakpoint on it



```
        {
            string text;
            if (Operators.CompareString(this.Protected-By-intigriti|
              TACC1zRxxooeWZuA9REZ(this.Protected-By-intigriti|
              g9PaJggXrcsxs68yrjkO.Text, text), <Module>.Protected-By-intigriti|
              wkzhwAf24JCPd2QZv3m0<string>((uint)(-1896895783 + sizeof(decimal) -
              27), (uint)(-1896896129 + sizeof(byte) + 334), (uint)(-1896901677 -
              sizeof(decimal) - 798)), (-8881 * sizeof(bool) ^ 9154 - sizeof(byte) -
              156) + (int)double.Parse(<Module>.Protected-By-intigriti|
              lnekL6mi5QnO3BhHOkoP<string>((uint)(-1160628145 + sizeof(bool) - 968),
              (uint)(-1160630013 + sizeof(decimal) + 885), (uint)(-1160635678 -
              sizeof(sbyte) - 130))) + (sizeof(int) - sizeof(byte)) +
              Type.EmptyTypes.Length != 0) != (-5532 - sizeof(long) + 261 ^ 2693 *
              sizeof(short)) + (int)double.Parse(<Module>.Protected-By-intigriti|
              wkzhwAf24JCPd2QZv3m0<string>((uint)(927168301 + sizeof(long) + 95),
              (uint)(927168414 + sizeof(byte) - 11), (uint)(927160867 + sizeof
              (decimal) + 824))) + (sizeof(byte) + sizeof(byte)) +
              Type.EmptyTypes.Length)
            {
                break;
```

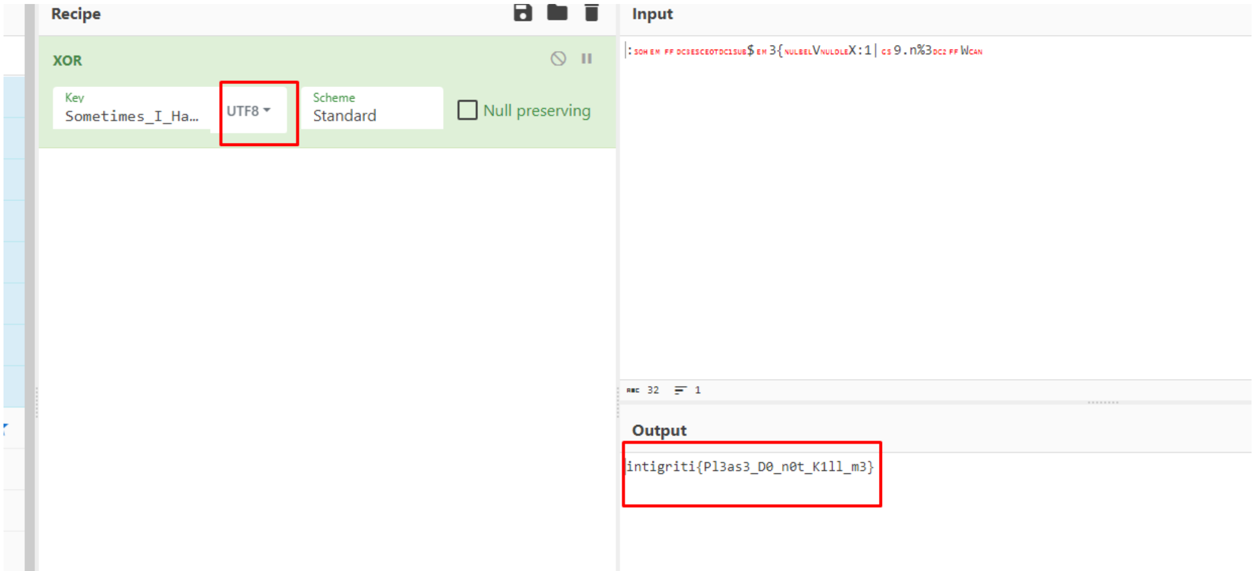
We run the program and input the word “hi”



Name	Value
Protected-By-intigriti|31x7F9BWte2J1YV5dZor.Protected-By-intigriti|F2...	{System.Windows.Forms.TextBox, Text: hi}
System.Windows.Forms.TextBox.Text.get returned	"hi"
Protected-By-intigriti|31x7F9BWte2J1YV5dZor.Protected-By-intigriti|F2...	"Rh"
<Module>.Protected-By-intigriti|wkzhwAf24JCPd2QZv3m0<string> r...	"Sometimes_I_Hate_The_Life_I_Made"
<Module>.Protected-By-intigriti|lnekL6mi5QnO3BhHOkoP<string> r...	"403.348864773451"
double.Parse returned	403.348864773451
Microsoft.VisualBasic.CompilerServices.Operators.CompareString retu...	0xFFFFFFFF
<Module>.Protected-By-intigriti|wkzhwAf24JCPd2QZv3m0<string> r...	"403.348864773451"
double.Parse returned	403.348864773451

We find that it is trying to compare between the key "Sometimes_I_Hate_The_Life_I_Made" and the input

Using cyberchef we can xor the enc file with the key to get the flag while keeping it as UTF8



XOR

Key
Sometimes_I_Ha…

UTF8

Scheme
Standard

□ Null preserving

Input

```
:SOHEMFFDC3ESCEOTDC1SUB$EM3{NULBELVNULDLEX:1|GS9.n%3DC2FFWCAN
```

Output

```
intigriti{Pl3as3_D0_n0t_K1ll_m3}
```