## 看雪ctf晋级赛第十题wp



rv0p111 于 2019-10-08 13:37:42 发布 161 ~ 收藏

分类专栏: 安全相关

版权声明:本文为博主原创文章,遵循 CC 4.0 BY-SA 版权协议,转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/ZCMUCZX/article/details/102380604

版权

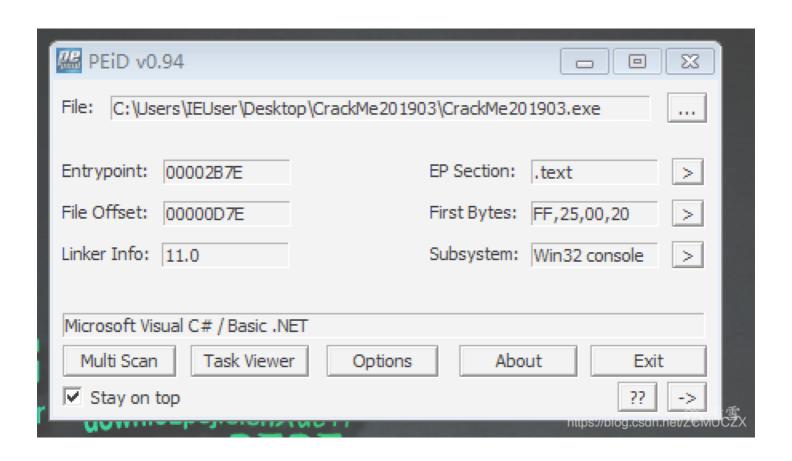


安全相关 专栏收录该内容

16 篇文章 0 订阅

订阅专栏

使用PEiD查看下是.net程序,这个程序是可以直接进行反编译的



所以我们使用ILSpy进行反编译下,我们可以看到程序就是叫我们输入一个字符串,然后和Kanxue2019一起给传入a函数当中,算结果为4RTIF9Ca2+oqExJwx68FiA==

```
ILSpy
ζ件(F) 视图(V) 帮助(H)
🖺 💋 🔑 C#
                        · (3 (5) (2)
 Mono Cecil
                                                                          private static void a(string[] A_0)
 ICSharpCode.AvalonEdit
 ICSharpCode.Decompiler
                                                                               Console.WriteLine("Please Input Serial:");
if (a.a(Console.ReadLine(), "Kanxue2019") == "4RTlF9Ca2+oqExJwx68"
 ILSpy
 📜 Client
⊢ mscorlib (2.0.0.0)
                                                                                   Console.WriteLine("Congratulations! : )");
├ • CrackMe201903 (1.0.0.0)
                                                                                   Console.ReadLine();
 卓⊪引用

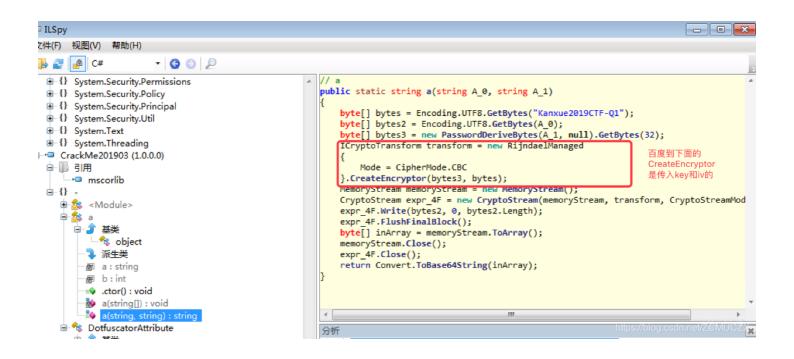
─ mscorlib

 ⊟-{} -

    ★ < Module >

    😑 🎎 а
       🖹 🧷 基类
           └� object
         🜗 派生类
         a:string
         - ∰ b:int
         -=• .ctor() : void
         🚵 a(string[]) :
         a(string, string): string
    ■ ♣ DotfuscatorAttribute
       由 🤰 基类
         _ a:string
         -@e c:int
         🕍 A : string
         C:int
         -= void .ctor(string, int) : void
```

所以下面我们看下a函数,ICryptoTransform是拿来做加密转换的,RijndaelManaged类是拿来做AES算法的一个类,然后又有cbc的提示, 所以我们就可以知道其实就是AES CBC的加密算法,所以我们只需要写出其相应的解密算法就可以了



由于没有C#环境,所以我们就去了http://www.dooccn.com/csharp/当中去在线编译,得到了下面的结果,下面就是解密的代码

```
public static string AESDecrypt(string text, string key, string iv)
                     RijndaelManaged rijndaelCipher = new RijndaelManaged();
                     rijndaelCipher.Mode = CipherMode.CBC;
                     rijndaelCipher.Padding = PaddingMode.PKCS7;
rijndaelCipher.KeySize = 128;
rijndaelCipher.BlockSize = 128;
                     byte[] encryptedData = Convert.FromBase64String(text);
byte[] keyBytes = new PasswordDeriveBytes("Kanxue2019", null).GetBytes(32);
rijndaelCipher.Key = keyBytes;
  23
                          e[] ivBytes = System.Text.Encoding.UTF8.GetBytes(iv);
  24
                     rijndaelCipher.IV = Encoding.UTF8.GetBytes("Kanxue2019CTF-Q1");
                     ICryptoTransform transform = rijndaelCipher.CreateDecryptor();
                     byte[] plainText = transform.TransformFinalBlock(encryptedData, 0, encryptedData.Length);
                     return Encoding.UTF8.GetString(plainText);
                }
       }
                                              出现故障, 请使用这个点击这里
run (ctrl+x)
               输入
                                分享当前代码
○ 文本方式显示 ○ html方式显示
Compilation succeeded - 1 warning(s)
Kanxue2019Q1CTF
```

/usercode/file.cs(24,20): warning CS0219: The variable `ivBytes' is assigned but its value is never used

https://blog.csdn.net/ZCMUCZX

所以就成功拿到flag了