看雪ctf晋级赛第十题wp

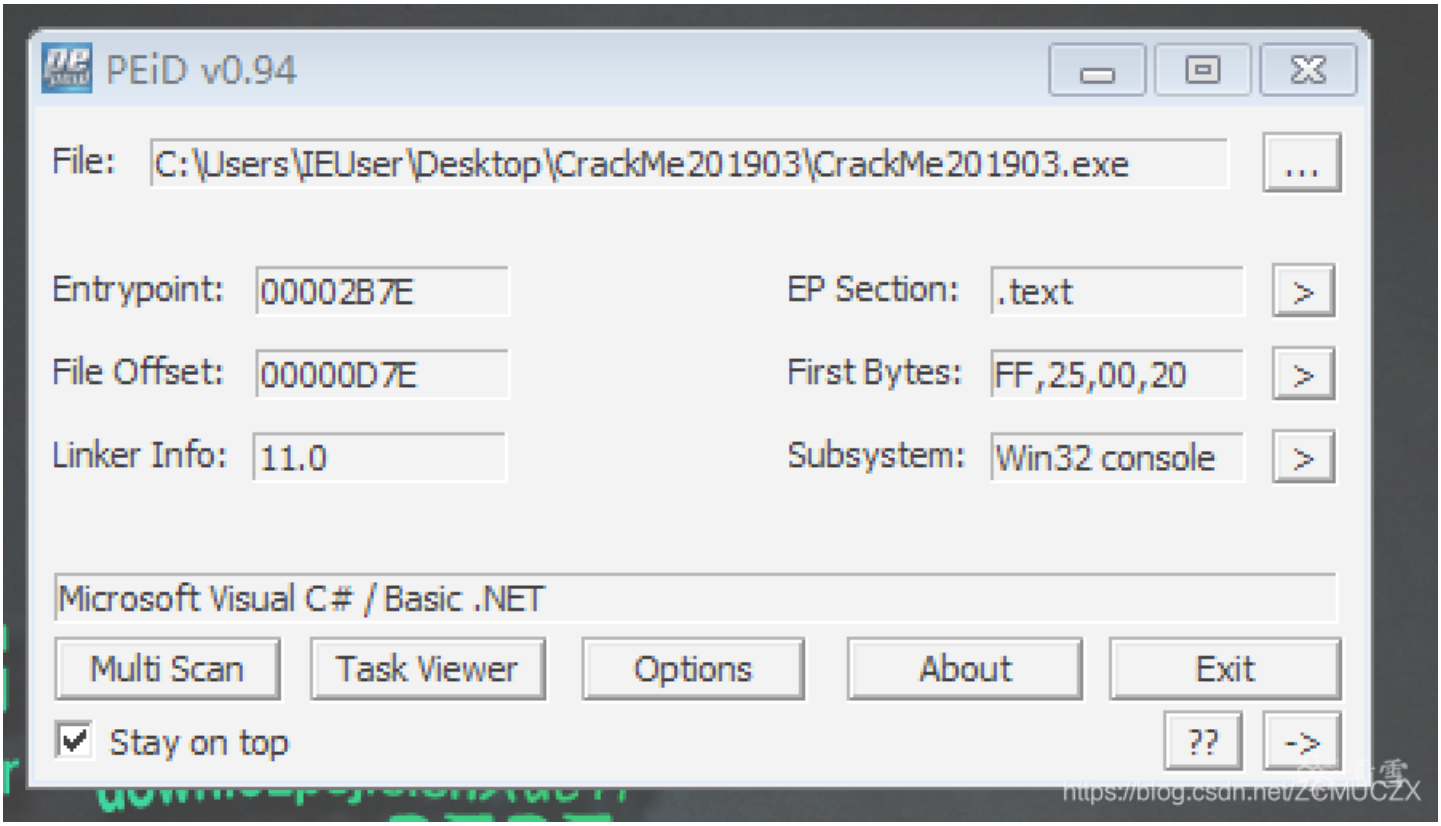
原创

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

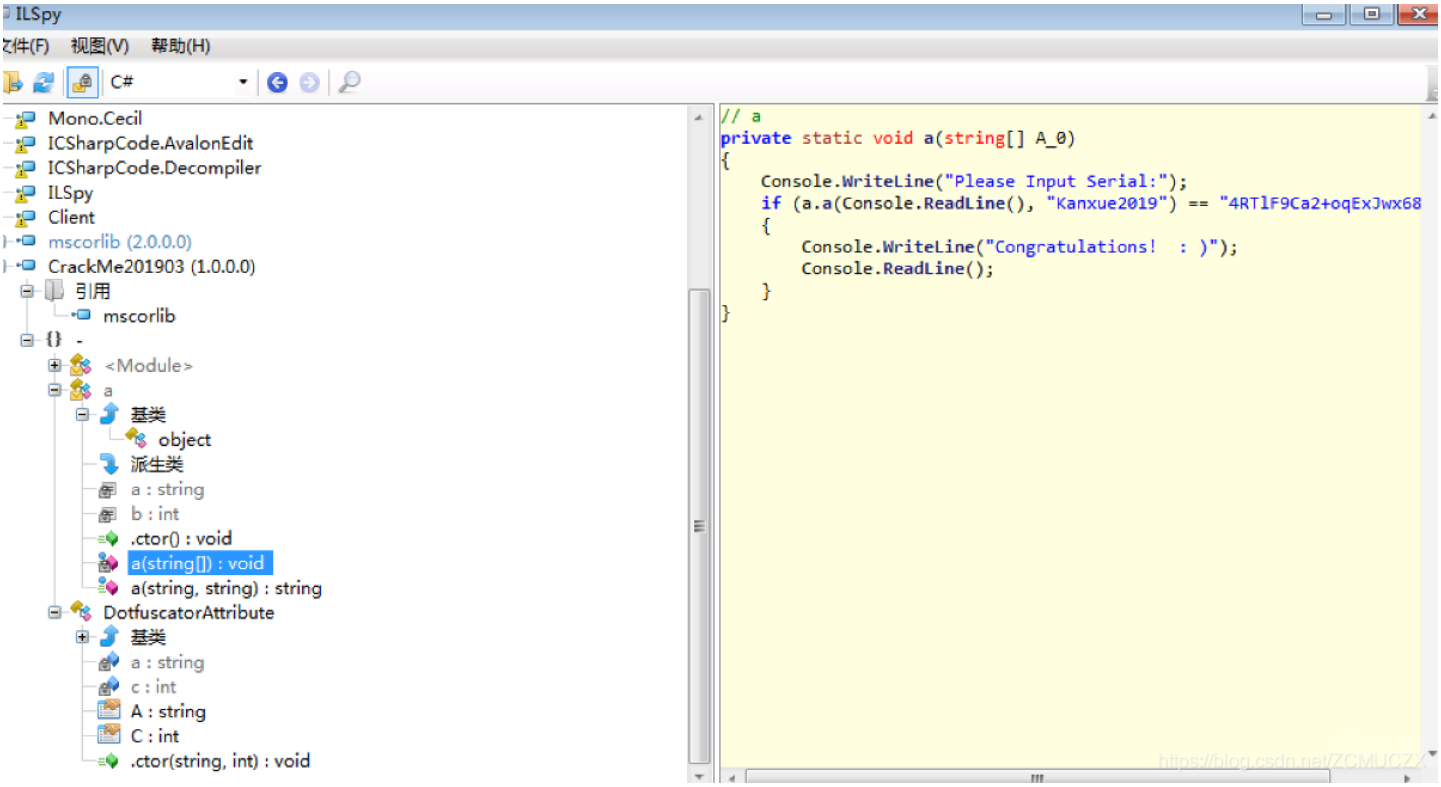
分类专栏： 安全相关

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

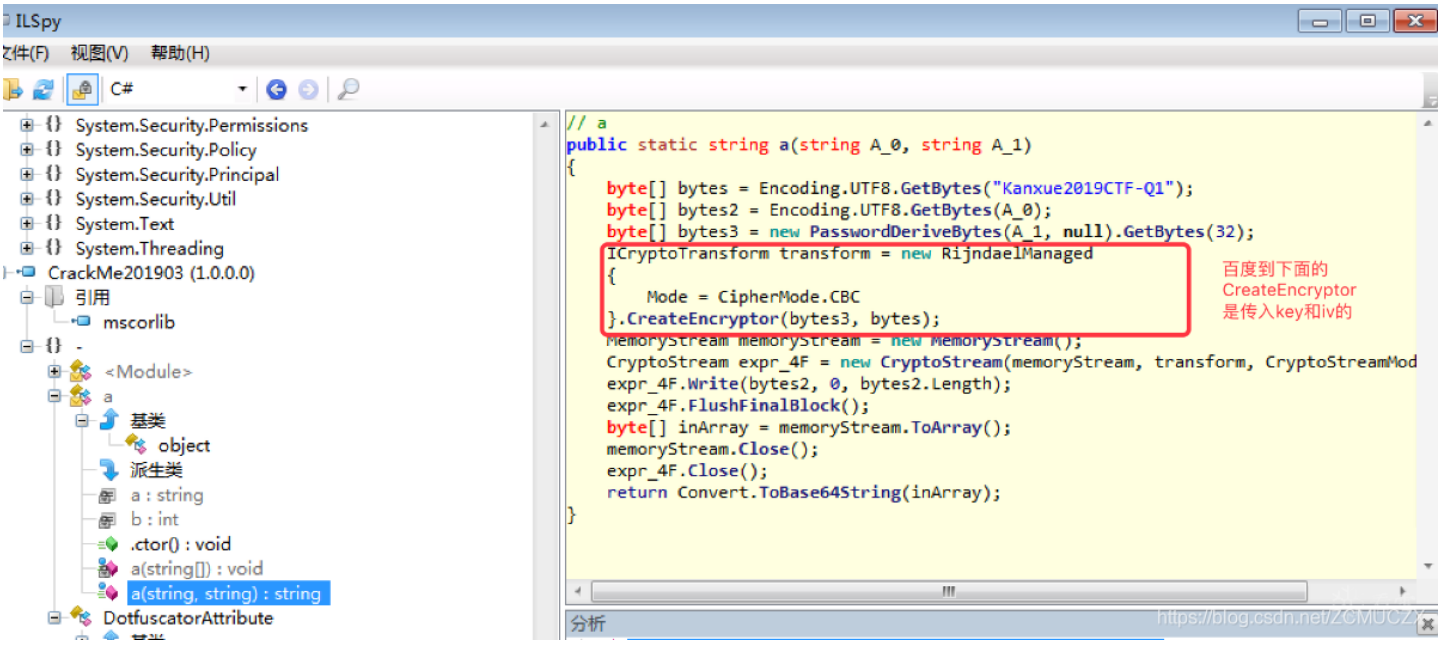
> 使用PEiD查看下是.net程序，这个程序是可以直接进行反编译的



> 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==



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

```
12
13      }
14      public static string AESDecrypt(string text, string key, string iv)
15          {
16              RijndaelManaged rijndaelCipher = new RijndaelManaged();
17              rijndaelCipher.Mode = CipherMode.CBC;
18              rijndaelCipher.Padding = PaddingMode.PKCS7;
19              rijndaelCipher.KeySize = 128;
20              rijndaelCipher.BlockSize = 128;
21              byte[] encryptedData = Convert.FromBase64String(text);
22              byte[] keyBytes = new PasswordDeriveBytes("Kanxue2019", null).GetBytes(32);
23              rijndaelCipher.Key = keyBytes;
24              byte[] ivBytes = System.Text.Encoding.UTF8.GetBytes(iv);
25              rijndaelCipher.IV = Encoding.UTF8.GetBytes("Kanxue2019CTF-Q1");
26              ICryptoTransform transform = rijndaelCipher.CreateDecryptor();
27              byte[] plainText = transform.TransformFinalBlock(encryptedData, 0, encryptedData.Length);
28              return Encoding.UTF8.GetString(plainText);
29          }
30
31  }
32
```

run (ctrl+x) 输入 分享当前代码 出现故障，请使用这个点击这里

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
```


所以就成功拿到flag了