2019KCTF 晋级赛Q1 | 第十题点评及解题思路

小雪 看雪学院 5天前

好望角“Cape Of Good Hope”位于印度洋温暖的莫桑比克厄加勒斯洋流与南极洲水域寒冷的本格拉洋流相遇的地方，当航海者在经历过狂风暴雨的洗礼，越过这个风暴角，便会获得财富和希望。

好望角也是2019 KCTF晋级赛Q1最后一站，最后这道题，又会带给我们怎样的惊喜呢？让我们一起来看看吧！

本道题目算是一道入门题，有超过100支队伍破解出来，战队 **kali-go** 以47分钟率先破解本题。

出题团队

战队成员：widesoft

个人主页：https://bbs.pediy.com/user-68717.htm

看雪CTF crownless 评委 点评

《初入好望角》作为本次看雪CTF晋级赛Q1的最后一题，是一道简单的送分题，基于.net2.0，使用了AES加密。参赛者只需要编写简单的代码，即可解密得到flag。

题目设计思路

一道.net平台下入门题目。使用了较常用的网站用户名/密码加密方法。程序使用Visualstudio自带的Dotfuscator做了简单混淆，几乎没有什么强度。可以逆向查看代码或通过修改IL再重新编译的方式爆破。

破解思路

本题解题思路由看雪论坛 **SnowMzn** 提供

0x001-----程序逆向

由于程序是.net语言开发，因此使用DnSpy打开程序，启动程序后，使程序断在入口点处。

在这里可以看到代码

```
internal class a
{
    // Token: 0x06000004 RID: 4 RVA: 0x0000209B File Offset: 0x0000029B
    private static void a(string[] A_0)
    {
        Console.WriteLine("Please Input Serial:");
        if (global::a.a(Console.ReadLine(), "Kanxue2019") == "4RTlF9Ca2+oqExJwx68FiA==")
        {
            Console.WriteLine("Congratulations!  : )");
            Console.ReadLine();
        }
    }

    // Token: 0x06000005 RID: 5 RVA: 0x000020D4 File Offset: 0x000002D4
    public static string a(string A_0, string A_1)
    {
        byte[] bytes = Encoding.UTF8.GetBytes("Kanxue2019CTF-Q1");
        byte[] bytes2 = Encoding.UTF8.GetBytes(A_0);
        byte[] bytes3 = new PasswordDeriveBytes(A_1, null).GetBytes(32);
        ICryptoTransform transform = new RijndaelManaged
        {
            Mode = CipherMode.CBC
        }.CreateEncryptor(bytes3, bytes);
        MemoryStream memoryStream = new MemoryStream();
        CryptoStream cryptoStream = new CryptoStream(memoryStream, transform, CryptoStreamMode.Write);
        cryptoStream.Write(bytes2, 0, bytes2.Length);
        cryptoStream.FlushFinalBlock();
        byte[] inArray = memoryStream.ToArray();
        memoryStream.Close();
        cryptoStream.Close();
        return Convert.ToBase64String(inArray);
    }
```

代码逻辑很简单，将输入的字符串进行加密处理后，与字符串**4RTlF9Ca2+oqExJwx68FiA==**比较。

0x002-----算法分析

算法中的bytes和bytes3为固定值，bytes2为输入的值。

由于之前对C#语言并不是很了解，因此猜测算法中，是使用bytes和bytes3为密钥，使用CryptoStream加密bytes2，最后将加密后的字节数组进行Base64加密。

因此解密算法的思路为：

1、首先将 **4RTlF9Ca2+oqExJwx68FiA==**字符串解密为十六进制字符串；

2、使用CryptoStream解密，得出flag。

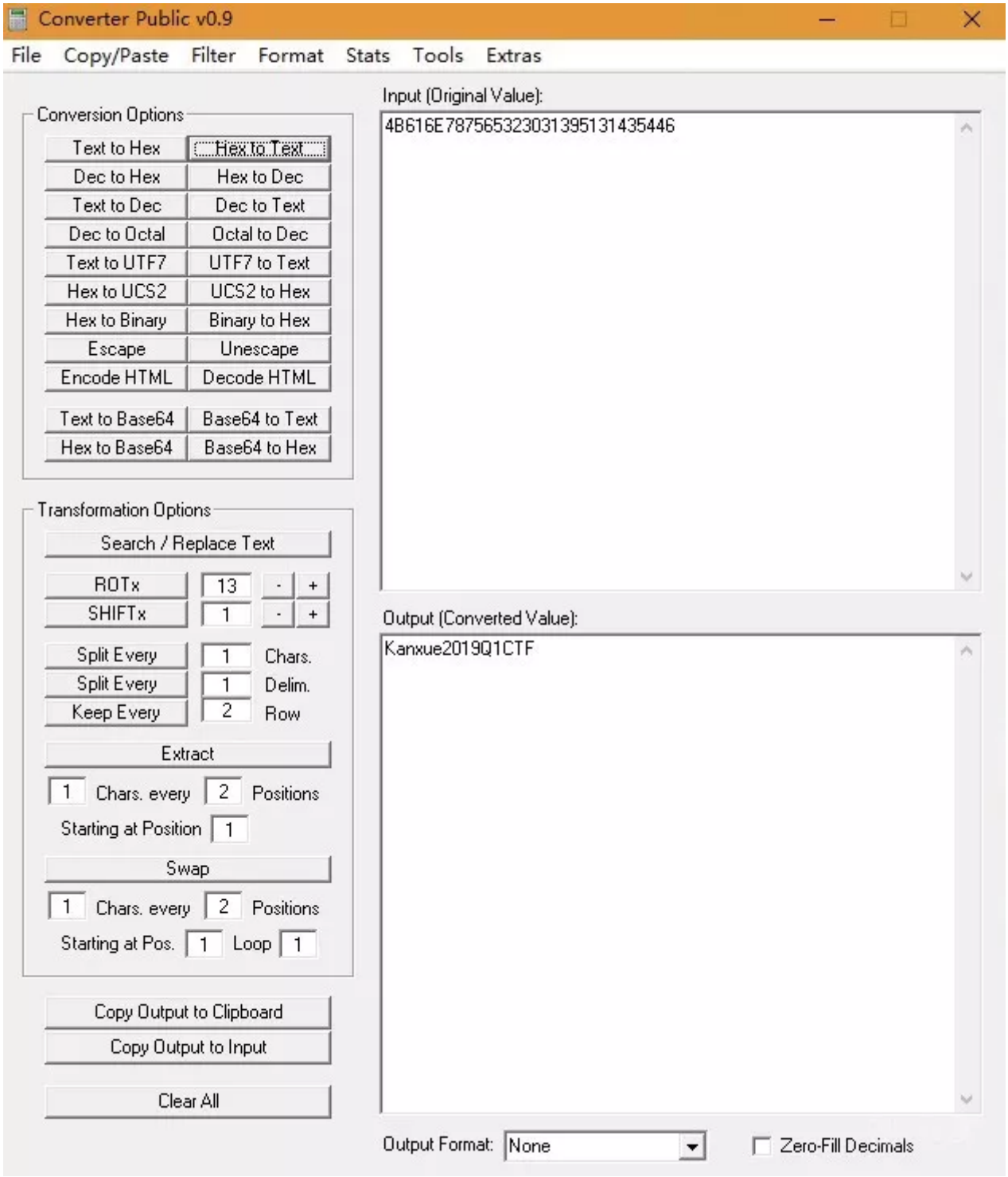
0x003-----编写脚本

使用C#编写解密脚本如下：

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Security.Cryptography;
using System.IO;

namespace ConsoleApplication1
{
class Program
{
static void Main(string[] args)
{
byte[] bytes = Encoding.UTF8.GetBytes("Kanxue2019CTF-Q1");
byte[] bytes3 = new PasswordDeriveBytes("Kanxue2019", null).GetBytes(32);

//string key_base64 = "4RTlF9Ca2+oqExJwx68FiA==";
//StringBuilder key = new StringBuilder();
//foreach (byte b in Convert.FromBase64String(key_base64).ToArray())
//{
// //Format as hex
```

```
// key.AppendFormat("{0:X2}", b);
//}
//Console.WriteLine("{0}", key.ToString());

string ret1 = "E114E517D09ADBEA2A131270C7AF0588";

ICryptoTransform transform = new RijndaelManaged
{
Mode = CipherMode.CBC
}.CreateDecryptor(bytes3, bytes);

byte[] inputByteArray = new byte[ret1.Length / 2];

for (int x = 0; x < ret1.Length / 2; x++)
{
int i = (Convert.ToInt32(ret1.Substring(x * 2, 2), 16));
inputByteArray[x] = (byte)i;
}

MemoryStream memoryStream = new MemoryStream();
CryptoStream cryptoStream = new CryptoStream(memoryStream, transform, CryptoStreamMode.
cryptoStream.Write(inputByteArray, 0, inputByteArray.Length);
cryptoStream.FlushFinalBlock();

byte[] inArray = memoryStream.ToArray();
memoryStream.Close();
cryptoStream.Close();

string returnStr = "";
for (int i = 0; i < inArray.Length; i++)
{
returnStr += inArray[i].ToString("X2");
}
Console.WriteLine("{0}", returnStr);
}
}
}
```

结果得到的是十六进制字符串，将结果转换为字符串即可。



看雪CTF晋级赛Q1 题解列表

1、2019KCTF 晋级赛Q1 | 第一题点评及解题思路

2、2019KCTF 晋级赛Q1 | 第二题点评及解题思路

3、2019 KCTF 晋级赛Q1 | 第三题点评及解题思路

4、2019KCTF 晋级赛Q1 | 第四题点评及解题思路

5、2019 KCTF 晋级赛Q1 | 第五题点评及解题思路

6、2019 KCTF 晋级赛Q1 | 第六题点评及解题思路

7、2019 KCTF 晋级赛Q1 | 第七题点评及解题思路

8、2019KCTF 晋级赛Q1 | 第八题点评及解题思路

9、2019KCTF 晋级赛Q1 | 第九题点评及解题思路

- End -

热门图书推荐


戳原文，查看更多精彩writeup！

阅读原文