170611 逆向-gctf的debug的writeup

原创

奈沙夜影 于 2017-06-12 13:15:02 发布 471 收藏
分类专栏： CTF

本文链接：https://blog.csdn.net/whklhhhh/article/details/73098069

1625-5 王子昂 总结《2017年6月11日》【连续第252天总结】

A.gctf reverse（1）

B.下载到debug.exe后拖入PEiD，发现是C#做的

刚好之前在实验吧的练习中学到了C#的逆向方法：使用ILSpy工具进行反编译

轻松得到源码，接下来就是分析即可

虽然函数名字都显示为乱码了，但是最终还是在一个类中找到了相关的函数

最外层函数：

```
//
private static void  (string[] A_0)
{
    string b = null;
    string value =string.Format("{0}", DateTime.Now.Hour +1);
    string a_ = "CreateByTenshine";
     . (a_, Convert.ToInt32(value),ref b);
    string a = Console.ReadLine();
    if (a == b)
    {
        Console.WriteLine("u got it!");
        Console.ReadKey(true);
    }
    else
    {
        Console.Write("wrong");
    }
    Console.ReadKey(true);
}
```

可以看出，该程序将当前时间和一个字符串传入了一个函数进行处理，保存在临时变量b中，然后与输入字符串进行比较。因此

处理函数：

```
//
private static void  (string A_0, int A_1, refstring A_2)
{
    int num = 0;
    if (0 < A_0.Length)
    {
        do
        {
            char c = A_0[num];
            int num2 = 1;
            do
            {
                c = Convert.ToChar( . (Convert.ToInt32(c), num2));
                num2++;
            }
            while (num2 < 15);
            A_2 += c;
            num++;
        }
        while (num < A_0.Length);
    }
    A_2 =  . (A_2);
}
```

在这个函数中，传入的时间参数A_1完全没有用到，看来是用来混淆的（也可能是每小时换一个flag太麻烦了吧

本函数中为对之前的字符串每个字符依次处理，先转换为INT类型然后在下一层函数中处理

将处理结果拼接起来，再最后处理一次

再内层的函数分别是一个由数组列表，两个数字参数一个用来作为索引，一个用来作为按位异或的操作数，结果返回的函数

和一个MD5处理的函数

大体算法就是这样，但由于没学过C#，不明白各个库函数的意思，还是有点麻烦的，百度了一番也没明白ToInt32和chr到底是按ASCII转换还是直接转换为对应内容的字符，用python照着写了一遍，结果却不对

请教了一下他人得知可以直接照抄到在线C#编译器中解决：

```
using System;
using System.Security.Cryptography;
using System.Text;

public class Test
{
public static void Main()
 {
 string b = null;
 string value = string.Format("{0}", DateTime.Now.Hour + 1);
 string a_ = "CreateByTenshine";
 AAA(a_, Convert.ToInt32(value), ref b);
 string a = Console.ReadLine();
    Console.WriteLine(b);
 }



 private static void AAA(string A_0, int A_1, ref string A_2)
    {
    int num = 0;
    if (0 < A_0.Length)
    {
    do
    {
    char c = A_0[num];
    int num2 = 1;
    do
    {
    c = Convert.ToChar(CCC(Convert.ToInt32(c), num2));
    num2++;
    }
    while (num2 < 15);
    A_2 += c;
    num++;
    }
    while (num < A_0.Length);
    }
```

```
    }
Console.WriteLine(A_2);
    A_2 = BBB(A_2);
    }

        private static string BBB(string A_0)
    {
    byte[] bytes = Encoding.ASCII.GetBytes(A_0);
    return "flag{" + BitConverter.ToString(new MD5CryptoServiceProvider().ComputeHash(bytes)).Replace("-", "") + "}";
    }
```

```
        private static int CCC(int A_0, int A_1)
    {
    return (new int[]
    {
    2,
    3,
    5,
    7,
    11,
    13,
    17,
    19,
    23,
    29,
    31,
    37,
    41,
    43
```

```
        43,
        47,
        53,
        59,
        61,
        67,
        71,
        73,
        79,
        83,
        89,
        97,
        101,
        103,
        107,
        109,
        113
    })[A_1] ^ A_0;
    }
}
```

即可得到输出：

字符串为：[j}yl}ZaL}vkpqv}

MD5加密再拼接后的flag：flag{967DDDFBCD32C1F53527C221D9E40A0B}

验证后正确

明天后天再研究下第二个rev的题目，hackme格式，记事本打开后发现了ELF头，简单查了一下是LINUX的可执行文件格式

C.明日计划

计算思维/.NET理论