

buuctf逆向rome

原创

g子荣 于 2021-10-13 13:07:24 发布 38 收藏

分类专栏: 笔记

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) 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/weixin_50847719/article/details/120741768

版权



笔记 专栏收录该内容

9 篇文章 0 订阅

订阅专栏

```
17 strcpy(v12, "Qsw3sj_lz4_Ujw@1");
18 printf("Please input:");
19 scanf("%s", &v2);
20 result = v2;
21 if ( v2 == 'A' )
22 {
23     result = v3;
24     if ( v3 == 'C' )
25     {
26         result = v4;
27         if ( v4 == 'T' )
28         {
29             result = v5;
30             if ( v5 == 'F' )
31             {
32                 result = v6;
33                 if ( v6 == 'I' )
34                 {
35                     result = v11;
36                     if ( v11 == 'J' )
37                     {
38                         v1[0] = v7;
39                         v1[1] = v8;
40                         v1[2] = v9;
41                         v1[3] = v10;
42                         *(DWORD *)&v12[17] = 0;
43                         while ( *(int *)&v12[17] <= 15 )
44                         {
45                             if ( *((char *)v1 + *(DWORD *)&v12[17]) > '@' && *((char *)v1 + *(DWORD *)&v12[17]) <= 'Z' )
46                                 *((_BYTE *)v1 + *(DWORD *)&v12[17]) = (*((char *)v1 + *(DWORD *)&v12[17]) - 51) % 26 + 65;
47                             if ( *((char *)v1 + *(DWORD *)&v12[17]) > '@' && *((char *)v1 + *(DWORD *)&v12[17]) <= 'z' )
48                                 *((_BYTE *)v1 + *(DWORD *)&v12[17]) = (*((char *)v1 + *(DWORD *)&v12[17]) - 79) % 26 + 97;
49                             ++*(DWORD *)&v12[17];
50                         }
51                         *(DWORD *)&v12[17] = 0;
52                         while ( *(int *)&v12[17] <= 15 )
53                         {
54                             result = (unsigned __int8)v12[*(DWORD *)&v12[17]];
55                             if ( *((_BYTE *)v1 + *(DWORD *)&v12[17]) != (_BYTE)result )
56                                 return result;
57                             ++*(DWORD *)&v12[17];
58                         }
59                         result = printf("You are correct!");
60                     }
61                 }
62             }
63         }
64     }
65 }
```

用大写举例
 $(x-51)\%26+65=y$
 $x=(y-65)+26+51$
 $=y+12$
就是大写字母平移 12 位
同理:
小写字母平移 8 位

如果大写
如果小写

凯撒加密得到的结果在 `v12[17]` 中
和 `"Qsw3sj_lz4_Ujw@1"` 比较, 如果相同返回 "正确".
标题是 rome (罗马), 暗示了加密方式

model = 'Qsw3sj_lz4_Ujw@1'
flag = ""

```

for i in model:
    if ord(i) > 64 and ord(i) <= 90:
        a = ord(i) + 12
        if a > 90:
            flag += chr(a - 26)
        else:
            flag += chr(a)
    elif ord(i) > 96 and ord(i) <= 122:
        a = ord(i) + 8
        if a > 122:
            flag += chr(a - 26)
        else:
            flag += chr(a)
    else:
        flag += i
print ('flag{'+flag+'}')

```

flag{Cae3ar_th4_Gre@t}

您早就解出这道题了

1/4

日期: 伪代码解释一下

```

*( _DWORD *)&v12[17] = 0;
while ( *(int *)&v12[17] <= 15 )
{
    if ( *((char *)v1 + *( _DWORD *)&v12[17]) > '@' && *((char *)v1 + *( _DWORD *)&v12[17]) <= 'Z' )
        *((_BYTE *)v1 + *( _DWORD *)&v12[17]) = *((char *)v1 + *( _DWORD *)&v12[17]) - 51 % 26 + 65;
    if ( *((char *)v1 + *( _DWORD *)&v12[17]) > '`' && *((char *)v1 + *( _DWORD *)&v12[17]) <= 'z' )
        *((_BYTE *)v1 + *( _DWORD *)&v12[17]) = *((char *)v1 + *( _DWORD *)&v12[17]) - 79 % 26 + 97;
    ++*( _DWORD *)&v12[17];
}

```

```

for ( i = 0; i <= 15; ++i )
{
    if ( *((_BYTE *)&v1 + i) > 64 && *((_BYTE *)&v1 + i) <= 90 )
        *((_BYTE *)&v1 + i) = *((char *)&v1 + i) - 51 % 26 + 65;
    if ( *((_BYTE *)&v1 + i) > 96 && *((_BYTE *)&v1 + i) <= 122 )
        *((_BYTE *)&v1 + i) = *((char *)&v1 + i) - 79 % 26 + 97;
}

```

v12[17] 就是 i

```

*( _DWORD *)&v12[17] = 0;
while ( *(int *)&v12[17] <= 15 )
{
    result = (unsigned __int8)v12[*( _DWORD *)&v12[17]];
    if ( *((_BYTE *)v1 + *( _DWORD *)&v12[17]) != (_BYTE)result )
        return result;
    ++*( _DWORD *)&v12[17];
}
result = printf("You are correct!");

```

```

for ( i = 0; i <= 15; ++i )
{
    result = (unsigned __int8)&v15 + i;
    if ( *((_BYTE *)&v1 + i) != (_BYTE)result )
        return result;
}
result = printf("You are correct!");

```

result = v15 = "Qsw3sj_lz4_Ujw@l" 原文

```

if ( *((_BYTE *)v1 + *( _DWORD *)&v12[17]) != (_BYTE)result )
    return result;
++*( _DWORD *)&v12[17];

```

```

if ( *((_BYTE *)&v1 + i) != (_BYTE)result )
    return result;

```

第一个 while (for) 循环的结果要和 相同

