

# buctf逆向rome

原创

g子荣 于 2021-10-13 13:07:24 发布 收藏 38

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 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](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

The screenshot shows a note-taking application interface with various tools at the top. Below the toolbar, there is a large area for writing notes.

**Handwritten Notes:**

- A red arrow points from the text "用大写举例" to the formula  $(X-51)/26+65=Y$ .
- A blue arrow points from the formula  $X=(Y-65)+26+51$  to the equation  $= Y+12$ .
- The text "就是大写字母平移 12位" is written below the equations.
- A green arrow points from the text "同理：" to the handwritten note "小写字母平移 8位".
- The text "凯撒加密得到的结果在 v12[17] 中" is written below the green circle.
- The text "和 \"Qsw3sj\_1z4\_Ujw@1\" 比较, 如果相同返回 '正确'" is written below the green circle.
- The text "标题是 rome (罗马), 暗示了加密方式" is written at the bottom in blue ink.

**Redacted Code Snippet:**

```
01 strcpy(v12, "Qsw3sj_1z4_Ujw@1");
02 printf("Please input:");
03 scanf("%s", &v2);
04 result = v2;
05 if ( v2 == 'A' )
06 {
07     result = v3;
08     if ( v3 == 'C' )
09     {
10         result = v4;
11         if ( v4 == 'T' )
12         {
13             result = v5;
14             if ( v5 == 'F' )
15             {
16                 result = v6;
17                 if ( v6 == '{' )
18                 {
19                     result = v11;
20                     if ( v11 == '}' )
21                     {
22                         v1[0] = v7;
23                         v1[1] = v8;
24                         v1[2] = v9;
25                         v1[3] = v10;
26                         *(DWORD *)&v12[17] = 0;
27                         while ( *(int *)&v12[17] <= 15 ) 如果大写
28                         {
29                             if ( *((char *)v1 + *(DWORD *)&v12[17]) > '@' && *((char *)v1 + *(DWORD *)&v12[17]) <= 'Z' )
30                                 *(BYTE *)v1 + *(DWORD *)&v12[17] = *((char *)v1 + *(DWORD *)&v12[17]) - 51 % 26 + 65;
31                             if ( *((char *)v1 + *(DWORD *)&v12[17]) > '=' && *((char *)v1 + *(DWORD *)&v12[17]) <= 'z' )
32                                 *((char *)v1 + *(DWORD *)&v12[17]) = *((char *)v1 + *(DWORD *)&v12[17]) - 79 % 26 + 97;
33                             *(DWORD *)&v12[17] = 0;
34                         }
35                         while ( *(int *)&v12[17] <= 15 )
36                         {
37                             result = (unsigned _int8)v12[*(DWORD *)&v12[17]];
38                             if ( *(BYTE *)v1 + *(DWORD *)&v12[17] != (BYTE)result )
39                                 return result;
40                             ++*(DWORD *)&v12[17];
41                         }
42                         result = printf("You are correct!");
43                     }
44                 }
45             }
46         }
47     }
48 }
49 }
```

00000740 \_func:41 (401340)

```

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\_1z4\_Ujw@1" 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

[https://blog.csdn.net/weixin\\_50847719](https://blog.csdn.net/weixin_50847719)