

re -22 春秋杯 Snake

原创

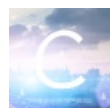
[weixin_45847059](#) 于 2021-12-01 18:18:08 发布 141 收藏

分类专栏: [re](#) 文章标签: [语言](#) [开发语言](#) [python](#)

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_45847059/article/details/121661122

版权



[re](#) 专栏收录该内容

24 篇文章 0 订阅

订阅专栏

脱壳后, ida打开找到flag检验函数

```
int flag_check(void)
{
    unsigned __int8 v1[256]; // [esp+18h] [ebp-910h] BYREF
    char v2[2048]; // [esp+118h] [ebp-810h] BYREF
    int j; // [esp+918h] [ebp-10h]
    int i; // [esp+91Ch] [ebp-Ch]

    gotoxy(22, 18);
    scanf("%s", v1);
    for ( i = 0; v1[i]; ++i )
        ;
    doubt(v1, i);
    memset(v2, 0, sizeof(v2));
    my_encode(v1, v2, i);
    gotoxy(22, 20);
    for ( j = 0; v2[j]; ++j )
    {
        if ( v2[j] != flag[j] )
            return puts(&Buffer);
    }
    return puts(&byte_405016);
}
```

CSDN @weixin_45847059

发现两个关键函数doubt, my_encode, my_encode显然是base64加密, 联想到表是否被改变, 对table变量查看交叉引用, 找到改变表的函数

```
int tablechange(void)
{
    int result; // eax
    char v1; // [esp+13h] [ebp-15h]
    signed int v2; // [esp+14h] [ebp-14h]
    int j; // [esp+18h] [ebp-10h]
    int i; // [esp+1Ch] [ebp-Ch]

    result = change_flag
```

```

9   result = change_flag;
10  if ( !change_flag )
11  {
12      v2 = strlen(table);
13      for ( i = 0; v2 / 2 > i; ++i )
14      {
15          for ( j = 0; v2 - i - 1 > j; ++j )
16          {
17              if ( table[j] > table[j + 1] )
18              {
19                  v1 = table[j];
20                  table[j] = table[j + 1];
21                  table[j + 1] = v1;
22              }
23          }
24      }
25      return ++change_flag;
26  }
27  return result;
28 }

```

CSDN @weixin_45847059

利用脚本跑出真正的表

```

#include<iostream>
#include<cstdio>
using namespace std;
char table[100] = "ABCDEFGHJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/" ;
int main()
{
    int result; // eax
    char v1; // [esp+13h] [ebp-15h]
    signed int v2; // [esp+14h] [ebp-14h]
    int j; // [esp+18h] [ebp-10h]
    int i; // [esp+1Ch] [ebp-Ch]

    v2 = strlen(table);
    for (i = 0; v2 / 2 > i; ++i)
    {
        for (j = 0; v2 - i - 1 > j; ++j)
        {
            if (table[j] > table[j + 1])
            {
                v1 = table[j];
                table[j] = table[j + 1];
                table[j + 1] = v1;
            }
        }
    }
    for (int i = 0; i < 64; i++)
    {
        cout << table[i];
    }
}

```

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789+/UVWXYZabcdefghijklmnopqrstuvwxyz

最随后的if判断中的flag数组中拿出加密后的值，对其进行解密

```

import base64
import string
str1 = "7G5d5bAy+TMdLWlu5CdkMT1cJnwkNUgb2AQL3CcmPpVf6DAp72sc0S1b"
string1 = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789+/UVWXYZabcdefghijklmnopqrstuvwxyz"
string2 = "ABCDEFGHIJKLMNOPQRSTUVWXYZUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
print(base64.b64decode(str1.translate(str.maketrans(string1, string2))))
print(str(base64.b64decode(str1.translate(str.maketrans(string1, string2))), 'utf-8'))

```

得出:

```
b"lfifp>y3),zd*p1<h'?06\x0b'X\x04\x0b\*2?Xkh05mn(9,g"
```

中间有三个\x代表一个16进制数

拿到加密前的字符串，再对doubt分析

对其写脚本解密得到flag

```

#include<iostream>
#include<cstdio>
using namespace std;
char b[100] = "lfifp>y3),zd*p1<h'?06'X\\*2?Xkh05mn(9,g";
int main()
{
    int j; // [esp+8h] [ebp-Ch]
    int i; // [esp+Ch] [ebp-8h]

    for (i = 1; i <= 10; ++i)
    {
        for (j = 0; j < 42; ++j)
        {
            if (42 % i)
                b[j] ^= i + j;
            else
                b[j] ^= (j % i) + j;
        }
    }
    for (int i = 0; i <= 42; i++)
        cout << b[i] ;
}

```

```
flag{5e2200bc-f21a-5421-a90b-57dec19fe196}
```