RealEzRE

发表于 2021-03-05 分类于 Challenge ， 2020 ， SWPU CTF 2020 ， Reverse
Challenge | 2020 | SWPU CTF 2020 | Reverse | RealEzRE

点击此处获得更好的阅读体验

WriteUp来源

官方WP

题目考点

- *RC4 加密*

解题思路

SMC

进入主函数后发现几个被加密后的字符串，通过IDA的Findcrypt 插件发现Base64的表，猜测这些字符串是经过Base64编码的，解密后得知是这里是一些提示信息

下面的操作是对关键函数区块进行smc解密操作，首先通过sub_401B80函数找到要解密的区块始末地址和大小

目标区块和0x67异或解密

IDC 脚本

```
auto key = 0x67;
auto from = 0x401cc0 + 0x20;
auto size = 0x359;
auto i, x;
for ( i = 0; i < size; ++i )
{
    x = Byte(from);
    x = (x^(key + (i & 0xF)));
    PatchByte(from,x);
    from = from + 1;
}
Message("\n Success \n");
```

关键函数

通过一些操作还原了关键函数，查看伪C代码后发现输入的字符串加密后和特定数值分三次比较

加密函数

通过分析知道这是个RC4加密

解密脚本

```
#include <stdio.h>

struct rc4
{
    int x, y, m[256];
};

typedef unsigned char uint8;

int main()
{
    uint8 flag[100] = { 0 };
    uint8 cmp[] = { 0x12,0xa7,0xf5,0xde,0x75,0x2a,0x6e,0x4a,0x6e,0x73,0xe6,0x62,0x50,0xbf,0x2a,0x98,0xfe,0x2b,0xdd,0x7b,0xba,0xb6,0x5,0x13,0x63,0x57,0x2d,0xd4,0x45,0xb8,0x
    uint8 key[8] = { 0x01,0x23,0x45,0x67,0x89,0xAB,0xCD,0xEF };
    int i, j, k, a,b;
    struct rc4 s;
    int length = 8;
    memset(&s, 0, sizeof(s));
    s.x = 0;
    s.y = 0;
    for (i = 0; i < 256; i++)
        s.m[i] = i;
    j = k = 0;
    for (i = 0; i < 256; i++)
    {
        a = s.m[i];
        j = (uint8)(j + a + key[k]);
        s.m[i] = s.m[j];
        s.m[j] = a;
        if (++k >= length)
            k = 0;
    }

    length = 32;
    for (i = 0; i < length; i++)
    {
        s.x = (uint8)(s.x + 1);
        a = s.m[s.x];
        s.y = (uint8)(s.y + a);
        s.m[s.x] = b = s.m[s.y];
        s.m[s.y] = a;
        flag[i] = cmp[i] ^ s.m[(uint8)(a + b)];
    }
    printf("flag{%s}", flag);
}
```

Flag

```
1 flag{f379eaf3c831b04de153469d1bec345e}
```

- ***本文作者：*** *CTFHub*
- ***本文链接：*** *https://writeup.ctfhub.com/Challenge/2020/SWPU-CTF-2020/Reverse/r3e1FLGA5jXFuGWSVyVKko.html*
- ***版权声明：*** *本博客所有文章除特别声明外，均采用 BY-NC-SA 许可协议。转载请注明出处！*

 # Challenge # 2020 # Reverse # SWPU CTF 2020
359度防护网站
Stangeapk