

# [2019红帽杯]easyRE writeup

原创

禾兮兮 于 2021-11-25 11:18:47 发布 37 收藏

文章标签: 其他 经验分享

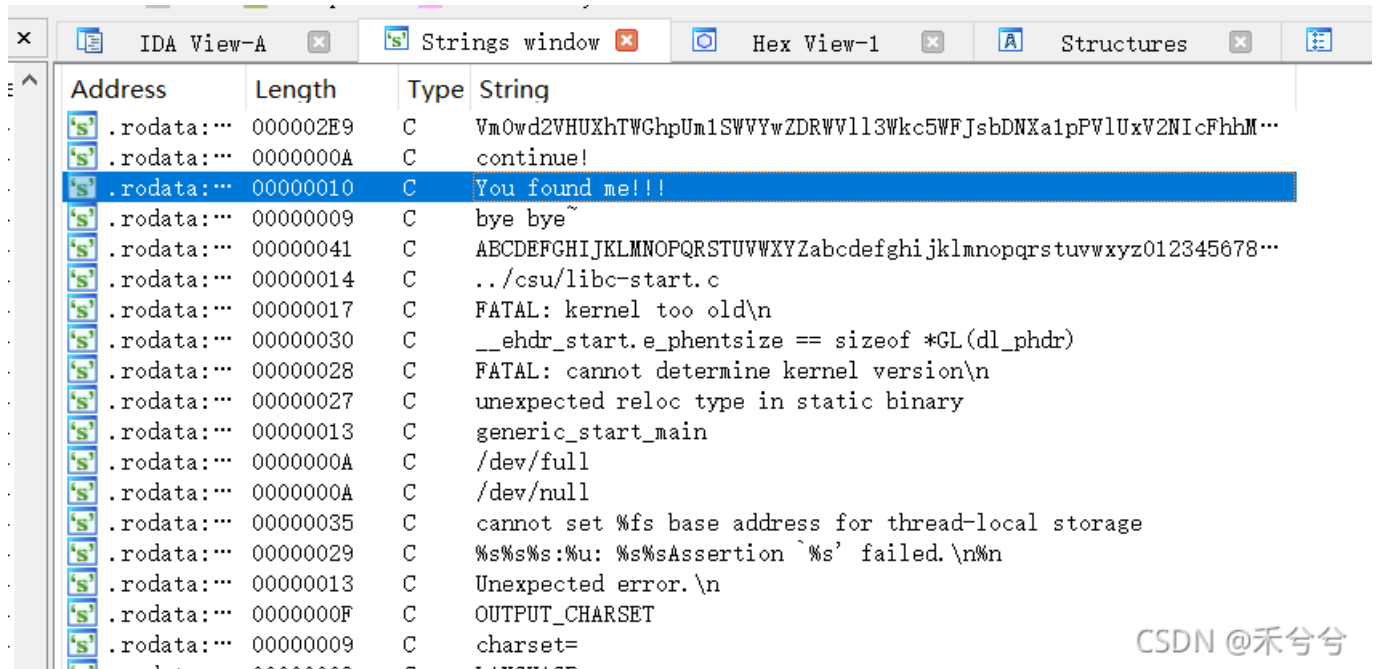
版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/HLi1219/article/details/121533211>

版权

很想挑战难题, 发现自己连writeup也看不懂

用64bit的ida打开, 查找字符



Address	Length	Type	String
.rodata:...	000002E9	C	Vm0wd2VHUXhTWGhpUm1SWVYwZDRWV1l3Wkc5WFJsbDNXa1pPV1UxV2NIcFhhM...
.rodata:...	0000000A	C	continue!
.rodata:...	00000010	C	You found me!!!
.rodata:...	00000009	C	bye bye~
.rodata:...	00000041	C	ABCDEFGHIJKLMNopQRSTUVWXYZabcdefghijklmnopqrstuvwxyz012345678...
.rodata:...	00000014	C	../csu/libc-start.c
.rodata:...	00000017	C	FATAL: kernel too old\n
.rodata:...	00000030	C	__ehdr_start.e_phentsize == sizeof *GL(dl_phdr)
.rodata:...	00000028	C	FATAL: cannot determine kernel version\n
.rodata:...	00000027	C	unexpected reloc type in static binary
.rodata:...	00000013	C	generic_start_main
.rodata:...	0000000A	C	/dev/full
.rodata:...	0000000A	C	/dev/null
.rodata:...	00000035	C	cannot set %fs base address for thread-local storage
.rodata:...	00000029	C	%s%s:%u: %sAssertion '%s' failed.\n\n
.rodata:...	00000013	C	Unexpected error.\n
.rodata:...	0000000F	C	OUTPUT_CHARSET
.rodata:...	00000009	C	charset=

CSDN @禾兮兮

找到函数:

```
signed __int64 sub_4009C6()
{
    signed __int64 result; // rax
    __int64 v1; // ST10_8
    __int64 v2; // ST18_8
    __int64 v3; // ST20_8
    __int64 v4; // ST28_8
    __int64 v5; // ST30_8
    __int64 v6; // ST38_8
    __int64 v7; // ST40_8
    __int64 v8; // ST48_8
    __int64 v9; // ST50_8
    __int64 v10; // ST58_8
    int i; // [rsp+Ch] [rbp-114h]
    char v12; // [rsp+60h] [rbp-C0h]
    char v13; // [rsp+61h] [rbp-BFh]
    char v14; // [rsp+62h] [rbp-BEh]
    char v15; // [rsp+63h] [rbp-BDh]
    char v16; // [rsp+64h] [rbp-BCh]
    char v17; // [rsp+65h] [rbp-BBh]
    char v18; // [rsp+66h] [rbp-BAh]
    char v19; // [rsp+67h] [rbp-B9h]
```

```
char v20; // [rsp+68h] [rbp-B8h]
char v21; // [rsp+69h] [rbp-B7h]
char v22; // [rsp+6Ah] [rbp-B6h]
char v23; // [rsp+6Bh] [rbp-B5h]
char v24; // [rsp+6Ch] [rbp-B4h]
char v25; // [rsp+6Dh] [rbp-B3h]
char v26; // [rsp+6Eh] [rbp-B2h]
char v27; // [rsp+6Fh] [rbp-B1h]
char v28; // [rsp+70h] [rbp-B0h]
char v29; // [rsp+71h] [rbp-AFh]
char v30; // [rsp+72h] [rbp-AEh]
char v31; // [rsp+73h] [rbp-ADh]
char v32; // [rsp+74h] [rbp-ACH]
char v33; // [rsp+75h] [rbp-ABh]
char v34; // [rsp+76h] [rbp-AAh]
char v35; // [rsp+77h] [rbp-A9h]
char v36; // [rsp+78h] [rbp-A8h]
char v37; // [rsp+79h] [rbp-A7h]
char v38; // [rsp+7Ah] [rbp-A6h]
char v39; // [rsp+7Bh] [rbp-A5h]
char v40; // [rsp+7Ch] [rbp-A4h]
char v41; // [rsp+7Dh] [rbp-A3h]
char v42; // [rsp+7Eh] [rbp-A2h]
char v43; // [rsp+7Fh] [rbp-A1h]
char v44; // [rsp+80h] [rbp-A0h]
char v45; // [rsp+81h] [rbp-9Fh]
char v46; // [rsp+82h] [rbp-9Eh]
char v47; // [rsp+83h] [rbp-9Dh]
char v48[32]; // [rsp+90h] [rbp-90h]
int v49; // [rsp+B0h] [rbp-70h]
char v50; // [rsp+B4h] [rbp-6Ch]
char v51; // [rsp+C0h] [rbp-60h]
char v52; // [rsp+E7h] [rbp-39h]
char v53; // [rsp+100h] [rbp-20h]
unsigned __int64 v54; // [rsp+108h] [rbp-18h]
```

```
v54 = __readfsqword(0x28u);
```

```
v12 = 73;
v13 = 111;
v14 = 100;
v15 = 108;
v16 = 62;
v17 = 81;
v18 = 110;
v19 = 98;
v20 = 40;
v21 = 111;
v22 = 99;
v23 = 121;
v24 = 127;
v25 = 121;
v26 = 46;
v27 = 105;
v28 = 127;
v29 = 100;
v30 = 96;
v31 = 51;
v32 = 119;
v33 = 125;
v34 = 110;
```

```

v34 = 119;
v35 = 101;
v36 = 107;
v37 = 57;
v38 = 123;
v39 = 105;
v40 = 121;
v41 = 61;
v42 = 126;
v43 = 121;
v44 = 76;
v45 = 64;
v46 = 69;
v47 = 67;
memset(v48, 0, sizeof(v48));
v49 = 0;
v50 = 0;
sub_4406E0(0LL, v48, 37LL);
v50 = 0;
if ( sub_424BA0(v48) == 36 )
{
    for ( i = 0; i < (unsigned __int64)sub_424BA0(v48); ++i )
    {
        if ( (unsigned __int8)(v48[i] ^ i) != *(&v12 + i) )
        {
            result = 4294967294LL;
            goto LABEL_13;
        }
    }
    sub_410CC0("continue!");
    memset(&v51, 0, 0x40uLL);
    v53 = 0;
    sub_4406E0(0LL, &v51, 64LL);
    v52 = 0;
    if ( sub_424BA0(&v51) == 39 )
    {
        v1 = sub_400E44(&v51);
        v2 = sub_400E44(v1);
        v3 = sub_400E44(v2);
        v4 = sub_400E44(v3);
        v5 = sub_400E44(v4);
        v6 = sub_400E44(v5);
        v7 = sub_400E44(v6);
        v8 = sub_400E44(v7);
        v9 = sub_400E44(v8);
        v10 = sub_400E44(v9);
        if ( !(unsigned int)sub_400360(v10, off_6CC090) )
        {
            sub_410CC0("You found me!!!");
            sub_410CC0("bye bye~");
        }
        result = 0LL;
    }
    else
    {
        result = 4294967293LL;
    }
}
else
{

```

```

    result = 0xFFFFFFFF;
}
LABEL_13:
if ( __readfsqword(0x28u) != v54 )
    sub_444020();
return result;
}

```

发现有两个加密的函数，一个是简单的异或，一个是base64的十次加密，密文如下

```

db 0
db 0
huxhtwg db 'Vm0wd2VHUXhTWGhpUm1SWVYwZDRWm1l3Wkc5WFJsbDNXa1pPVlUxV2NIcFhhMk0xV'
; DATA XREF: .data:off_6CC090↓o
db 'mpKS1NHVkdXbFpOYmtKVZtCtEtTMU15VGtsaVJtUk9ZV3hhZVZadGVHdFRNVTYVYW'
db '01T2FGSnRVbGhhVjNoaFZWwmtWMBFVWxSTmJFcElWbTAXVDJGV1NuTlhia0pXWwx'
db 'ob1dGUnJXbXRXTVZaeVdrWm9hVlpyV1hwV1IzaGhXVmRHVjFodVVsWmlhMHBZV1ZS'
db 'R1lWZEdVbFZTYlhSWFRWwndNRlZ0TVc5VWJGcFZWbXR3VjJKSFVYZFdhapXWlZaT'
db '2NtRkhhRk5pVjJowVYxZDBhMVV3Tlh0a1JscFlZbGhTY1ZsclduZGxiR1J5VmxSR1'
db 'ZXS1ZjRWhaTUZKaFZqSktWVkJZYUZkV1JWcFlwV3BHYTEkZFRiV3hvVFVoQ1d'
db 'sWXhaRFJpTwtSM1RVaG9hbEpYYUUhOVmJUVkrZekZy1ZKcmRGtk5Wa3A2VjJ0U1Ex'
db 'WlhTbFpquldoYVRVWndkbFpxUmtwbGJVWklZVWprYUdFeGNHOVhXSEJIWkRGS2RGS'
db 'nJhR2hTYXpWdlZGVm9RMlJzV25STldHUlZUVlpXTlZadE5VOVdiVXBjVld4c1dtS1'
db 'lUWGHXTUzwell6RmFkRkpzVWxOaVNFSktWa1phVTFFeFduUlRhMlJxVWxad1YxWnR'
db 'lRXXTVZaSFVsUnNVVlZVTURrPQ==',0
e db 'continue!',0 ; DATA XREF: sub_4009C6+1E5↑o
dMe db 'You found me!!!',0 ; DATA XREF: sub_4009C6+338↑o
db 'bye bye~',0 ; DATA XREF: sub_4009C6+342↑o
align 20h

```

CSDN @禾兮兮

利用工具翻译出看雪的网站，不是flag

首页 > 程序员开发工具 > base64解码

在线base64解码/编码工具

转换内容: aHR0cHM6Ly9iYnMucGVkaXkuY29tL3RocmVhZC0yNTQxNzluaHRl

Base64编码 Base64解码

转换结果: https://bbs.pediy.com/thread-254172.htm

Base64解码/编码说明

CSDN @禾兮兮

解密第一个函数（对数组进行i++的异或）得到提示Info:The first four chars are `flag`

在函数上下寻找，找到了sub\_400D35函数，使用"R"键，发现存在有flag的疑似字符

```

Instruction  Data  Unexplored  External symbol
IDA View-A  Pseudocode-B  Pseudocode-A  Strings window  Hex View-1  Structu
1 __int64 sub_400D35()
2 {
3     __int64 result; // rax
4     unsigned __int64 v1; // rt1
5     unsigned int v2; // [rsp+Ch] [rbp-24h]
6     signed int i; // [rsp+10h] [rbp-20h]
7     signed int j; // [rsp+14h] [rbp-1Ch]
8     unsigned int v5; // [rsp+24h] [rbp-Ch]
9     unsigned __int64 v6; // [rsp+28h] [rbp-8h]
10
11     v6 = __readfsqword(0x28u);
12     v2 = sub_43FD20(0LL) - qword_6CEE38;
13     for ( i = 0; i <= '\x040'; ++i )
14     {
15         sub_40F790(v2);
16         sub_40FE60(v2);
17         sub_40FE60(v2);
18         v2 = (unsigned __int64)sub_40FE60(v2) ^ 0x98765432;
19     }
20     v5 = v2;
21     if ( ((unsigned __int8)v2 ^ byte_6CC0A0[0]) == 'f' && (HIBYTE(v5) ^ a5V[2]) == 'g' )
22     {
23         for ( j = 0; j <= 24; ++j )
24             sub_410E90((unsigned __int8)(byte_6CC0A0[j] ^ *((_BYTE *)&v5 + j % 4)));
25     }
26     v1 = __readfsqword(0x28u);
27     result = v1 ^ v6;
28     if ( v1 != v6 )
29         sub_444020();
30     return result;
31 }

```

CSDN @禾兮兮

结合之前的提示，我们要找到4个与数组异或得到flag，脚本如下：

```

#include<stdio.h>
int main()
{
    int a[4]={102,108,97,103};
    int b[4];
    int i[25]={0x40, 0x35, 0x20, 0x56, 0x5D, 0x18, 0x22, 0x45, 0x17, 0x2F, 0x24, 0x6E, 0x62, 0x3C, 0x27, 0x5
    int k;
    for(k=0;k<4;k++)
    {
        b[k]=a[k]^i[k];
    }
    for(k=0;k<25;k++)
    {
        i[k]=i[k]^b[k%4];
        printf("%d ",i[k]);
    }
}

```

十进制串(空格隔开)例如97 98 99转成abc:

102 108 97 103 123 65 99 116 49 118 101 95 68 101 102 101 110 53 101 95 84 101 115 116 125

十进制转ASCII

ASCII转十进制

ASCII: flag{Active\_Defen5e\_Test}

十六进制串例如616263转成abc:

十六进制转ASCII

ASCII转十六进制