first 攻防世界

原创

北风~ 于 2020-05-16 18:40:00 发布 816 收藏

分类专栏： 逆向与保护

本文链接：https://blog.csdn.net/weixin_45055269/article/details/106157485

工具：

IDA

思路展开：

无壳64位，linux下运行，输错有错误提示`Badluck! There is no flag`

IDA启动，

根据用户输入时间随机生成6个数

```
v3 = useconds;
v4 = time(0LL);
srand(v4);
do
{
  ++v3;
  *(v3 - 1) = 100 * (rand() % 1000);          // 倒序
}
while ( v3 != (__useconds_t *)&unk_602208 );  // 产生的数和某数组比较
```

下面一堆的操作，里面的变量和后面没关系，所以不用管，接下来输入经过操作生成v11。

```
v11 = 0;
while ( v9 != v10 )
{
  v12 = *((_BYTE *)input + v10) + v10;
  ++v10;
  v11 ^= v12;
}
```

接下来，程序开启6个线程

```
v13 = (void (**)(void *))&newthread;
v14 = 0LL;
v15 = &newthread;
do
{
  if ( pthread_create(v15, 0LL, (void *(*)(void *))start_routine, v14) )
  {
    perror("pthread_create");
    exit(-1);
  }
  ++v14;
  ++v15;
}
while ( v14 != (char *)6 );                     // 6个线程
```


看线程的调用函数 `start_routine`

```
__int64 v4; // rbx
int v5; // eax
__int64 v6; // rdx
__int64 v8; // [rsp+0h] [rbp-38h]
unsigned __int64 v9; // [rsp+18h] [rbp-20h]

v1 = (signed int)a1;
v2 = (signed int)a1;
v3 = useconds[(signed int)a1];
v9 = __readfsqword(0x28u);
v4 = v2;
usleep(v3);
pthread_mutex_lock(&mutex);
sub_400E10(&input[v4], 4uLL, (__int64)&v8);
v5 = dword_6021E8;
v6 = dword_6021E8;
if ( v8 == byte_602120[v1] )
  dword_602220[v6] = input[v4];
else
  dword_602220[v6] = 0;
dword_6021E8 = v5 + 1;
pthread_mutex_unlock(&mutex);
return __readfsqword(0x28u) ^ v9;
}
```


红框中sleep函数，每次取输入的四个再往里传时会阻塞不同时间，这影响到传入的顺序，所以求出的输入顺序需要再变化。（这是个坑！！！）

绿框里找到md5的四个标志数组，所以函数是将输入的四个（怎么看出输入是四个为一组：绿框函数尺寸为4）为一组md5加密，与黄框中比较，若相等则传入 `602220`，（ `602220` 在main函数中提示与flag有关）

脚本爆破

```
import hashlib
check="4746bbbd02bb590fbeac2821ece8fc5cad749265ca7503ef4386b38fc12c4227b03ecc45a7ec2da7be3c5ffe121734e8"
for w in range(0,6):
 for i in range(48,123):
  for j in range(48,123):
   for m in range(48,123):
    for n in range(48,123):
     temp=chr(i)+chr(j)+chr(m)+chr(n)
     hashvalue=hashlib.md5(temp.encode()).hexdigest()
     if hashvalue[0:16]==check[w*16:w*16+16]:
      print(w,temp)
```

包括md5在内的一众算法的python写法

爆破生成6个数组，真实输入的顺序不确定。所以就是试，最终确定输入是`juhuhfenlapsdunuhjifiuer`这个顺序才可。

```
input1='juhuhfenlapsiuerhjifdunu'
check=[0xfe,0xe9,0xf4,0xe2,0xf1,0xfa,0xf4,0xe4,0xf0,0xe7,0xe4,0xe5,0xe3,0xf2,0xf5,0xef,0xe8,0xff,0xf6,0xf4,0xfd,
0xb4,0xa5,0xb2]
len=24
i=0
v11=0
while(i!=len):
 v12=ord(input1[i])+i
 v11=v11^v12
 i=i+1

input2='juhuhfenlapsdunuhjifiuer'
flag=''
for i in range(24):
 temp=ord(input2[i])^v11^check[i]
 flag+=chr(temp)
print(flag)
```

goodjobyougetthisflag233