攻防世界SignIn

原创

Outsider 于 2021-01-11 12:00:03 发布 1076 收藏 1

分类专栏： 攻防世界逆向

本文链接：https://blog.csdn.net/qq_48274326/article/details/112464774

ida进入到主函数

```
__int64 __fastcall main(int a1, char **a2, char **a3)
{
  char v4[16]; // [rsp+0h] [rbp-4A0h] BYREF
  char v5[16]; // [rsp+10h] [rbp-490h] BYREF
  char v6[16]; // [rsp+20h] [rbp-480h] BYREF
  char v7[16]; // [rsp+30h] [rbp-470h] BYREF
  char v8[112]; // [rsp+40h] [rbp-460h] BYREF
  char v9[1000]; // [rsp+B0h] [rbp-3F0h] BYREF
  unsigned __int64 v10; // [rsp+498h] [rbp-8h]

  v10 = __readfsqword(0x28u);
  puts("[sign in]");
  printf("[input your flag]: ");
  __isoc99_scanf("%99s", v8);
  sub_96A(v8, v9);
  __gmpz_init_set_str(v7, "ad939ff59f6e70bcbfad406f2494993757eee98b91bc244184a377520d06fc35", 16LL);
  __gmpz_init_set_str(v6, v9, 16LL);
  __gmpz_init_set_str(v4, "10346103590081691412139010129904904441395040517371217043416168653987816098454 9", 10LL
);
  __gmpz_init_set_str(v5, "65537", 10LL);
  __gmpz_powm(v6, v6, v5, v4);
  if ( (unsigned int)__gmpz_cmp(v6, v7) )
    puts("GG!");
  else
    puts("TTTTTTTTTql!");
  return 0LL;
}
```

很明显的rsa加密

第一步分解大数N 10346103590081691412139010129904904441395040517371217043416168653987816098454 9

有请yafu来得到p和q

```
D:\>\桌面\常用工具\网络安全工具（全）\Tools\Crypto\yafu\yafu-x64.exe
factor(10346103590081691412139010129904904441395040517371217043416168653987816098454 9)


fac: factoring 10346103590081691412139010129904904441395040517371217043416168653987816098454 9
fac: using pretesting plan: normal
fac: no tune info: using qs/gnfs crossover of 95 digits

starting SIQS on c78: 10346103590081691412139010129904904441395040517371217043416168653987816098454 9

==== sieving in progress (1 thread):   36224 relations needed ====
====           Press ctrl-c to abort and save state           ====


SIQS elapsed time = 1.5426 seconds.
Total factoring time = 1.5715 seconds


***factors found***

P39 = 366669102002966856876605669837014229419
P39 = 282164587459512124844245113950593348271

ans = 1
```


这里我们写个python脚本

```python
p = 366669102002966856876605669837014229419
q = 282164587459512124844245113950593348271
N = 10346103590081691412139010129904904441395040517371217043416168653987816098454 9
c = 0xad939ff59f6e70bcbfad406f2494993757eee98b91bc244184a377520d06fc35
e = 65537



def ext_euclid(a, b):
    old_s,s=1,0
    old_t,t=0,1
    old_r,r=a,b
    if b == 0:
        return 1, 0, a
    else:
        while(r!=0):
            q=old_r//r
            old_r,r=r,old_r-q*r
            old_s,s=s,old_s-q*s
            old_t,t=t,old_t-q*t
    return old_s, old_t, old_r
ol=(p-1)*(q-1)
d=ext_euclid(ol,e)[1]
while d<0:
    d+=ol
m = pow(c, d, N)
print(bytes.fromhex(hex(m)[2:]))

b'suctf{Pwn_@_hundred_years}'
```