[2023 JBUCTF] crypto

random_primes

Write-Up

문제 개요

제공 파일 : random_primes.py

```python
from Crypto.Util.number import getPrime
from random import randint

flag = open('/flag', 'rb').read()
flag = int.from_bytes(flag, 'big')

def gen_prime_list(n):
    primes = []
    for i in range(n):
        primes.append(getPrime(512))
    return primes

primes = gen_prime_list(4)

e = 65537
p = primes[randint(0, len(primes) - 1)]
q = primes[randint(0, len(primes) - 1)]
n = p * q

c = pow(flag, e, n)
primes = ', '.join(map(str, primes))
print(f'enc_flag = {c}')
print(f'primes = [{primes}]')

```

gen_prime_list 함수로 소수들로 이루어진 리스트를 생성해서 그 리스트 중에서 랜덤으로 소수를 골라서 p, q 변수에 값을 저장한다.

그 후에 p, q를 RSA의 개인키로 이용하여 flag를 암호화하고, 그 암호화 한 값과, 생성한 리스트를 출력한다.

문제 풀이

RSA 암복호화 참고 : https://url.kr/56ngfw

소수 리스트의 크기가 총 4이고 이중에서 중복으로 2개의 소수를 선택하여 개인키 값을 정하기 때문에 4H2 = 5C2 = 10개의 조합으로 p, q 값을 정해서 RSA 복호화를 시도하면 10번의 시도중에서 flag를 획득할 수 있다.

exploit.py

```python
from pwn import *

p = remote('172.17.0.2', 10007)

p.recvuntil(b'enc_flag = ')
enc_flag = int(p.recvline()[:-1].decode())

p.recvuntil(b'primes = ')
primes = p.recvline()[:-1].decode()
primes = list(map(int, primes[1:-1].split(', ')))
e = 65537

for i in range(len(primes)):
    for j in range(i, len(primes)):
        n = primes[i] * primes[j]
        phi = (primes[i] - 1) * (primes[j] - 1)
        d = pow(e, -1, phi)
        dec_flag = pow(enc_flag, d, n)
        dec_flag = hex(dec_flag)[2:]

        if len(dec_flag) & 1:
            dec_flag = '0' + dec_flag
        dec_flag = bytes.fromhex(dec_flag)
        if dec_flag[0:6] == b'scpCTF':
            print(f'flag is : {dec_flag.decode()}')
            break
```

FLAG

scpCTF{78de570e092fd854ff182094744b00f0a494c67ed03a20274b52b7733c01ca141cead68519637556619c7271a6bb00ad}