[2023 JBUCTF] crypto

distance

Write-Up

문제 개요

제공 파일 : distance.py

```python
from Crypto.Util.number import getPrime, isPrime
from random import randrange

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

r = randrange(2**15, 2**16)

p = getPrime(1024)
q = p + r * 2
while isPrime(q) != True:
    q += 2
e = 0x10001
n = p * q

c = pow(m, e, n)

print(f'{c=}')
print(f'{n=}')
print(f'{e=}')

```

Flag를 RSA로 암호화해서 공개키(n, e)와 암호문(c)을 출력하는 코드이다.

문제 풀이

RSA 암복호화 참고 : https://url.kr/56ngfw

```
    r = randrange(2**15, 2**16)

    p = getPrime(1024)
    q = p + r * 2
    while isPrime(q) != True:
        q += 2
```

위 문제 코드의 일부분을 보면 개인 키인 두 소수(p, q)의 차이가 2r 만큼의 차이가 나는 것을 볼 수 있다. 이를 이용해서 아래와 같이 n + r**2 = (p + r)**2 이라는 식을 도출할 수 있다.

$$
\begin{aligned}
&n = p \times q \quad (p < q, \quad p, q \in \mathbb{P}) \\
&q = p + 2r \\
&n = p \times q = p \times (p + 2r) = p^2 + 2pr \\
&n + r^2 = p^2 + 2pr + r^2 = (p + r)^2 \\
&\therefore n + r^2 = (p + r)^2
\end{aligned}
$$

r의 크기가 2**15에서 2**16 – 1 사이로 작은 숫자이다.

따라서 r값을 2**15에서 2**16 – 1 까지의 숫자로 무차별 대입을 해서 n + r**2을 계산한 후, n + r**2의 제곱근이 정수이면, n + r**2 = (p + r)**2를 만족하는 r값을 찾은 것이다.

따라서 n + r**2 = (p + r)**2에 n, r을 대입하여 p를 구하고 q = p + 2r도 구해서 RSA 개인키를 생성한 후, 그 개인키로 암호화된 flag를 복호화 하면 flag를 알게 된다.

exploit.py

```
from gmpy2 import is_square, isqrt
from pwn import *

p = remote('172.17.0.2', 10006)

c = int(p.recvline()[2:-1].decode())
n = int(p.recvline()[2:-1].decode())
e = int(p.recvline()[2:-1].decode())

for r in range(2**15, 2**16):
    if is_square(n + r*r):
        p = isqrt(n + r*r) - r
        q = p + 2 * r
        break

phi = (p-1) * (q-1)
d = pow(e, -1, phi)
m = pow(c, d, n)

print(f'flag : {bytes.fromhex(hex(m)[2:]).decode()}')

```

FLAG

scpCTF{e9c0b7e2127d5d0eeddd27e06f8dbbe838e8b67d9a7f}