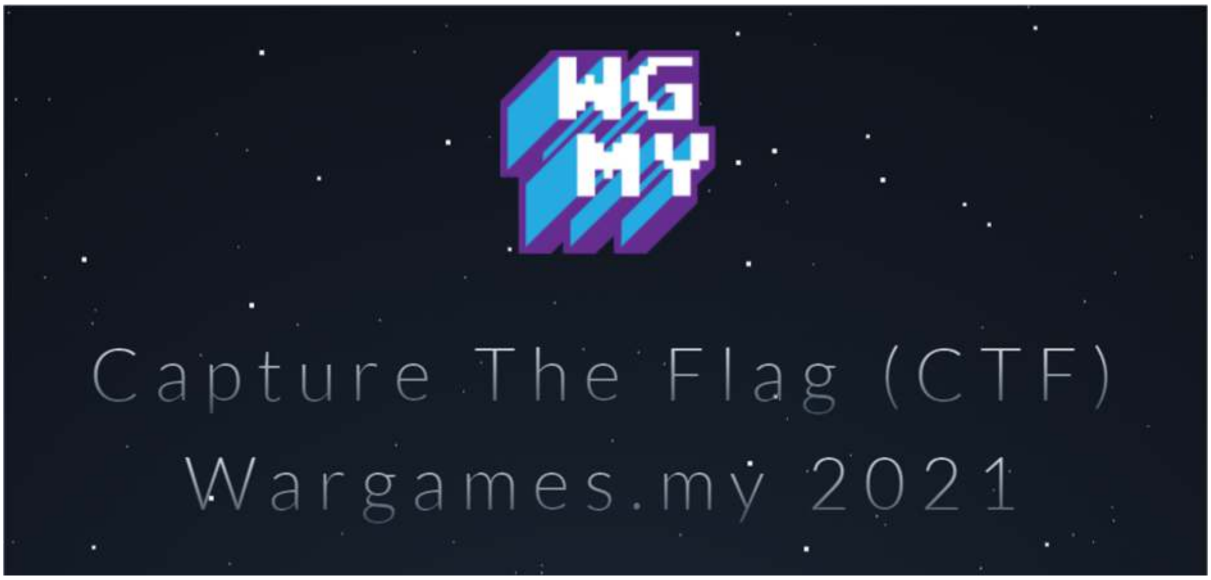
WGMY CTF 2021



Team Silver Dawn

Cryptography

Easyrsa

We have received a python file with the name chal.py

```python
#!/usr/bin/env python3
from Crypto.Util.number import *
from secret import flag

# Generate public key
p = getStrongPrime(1024)
q = getStrongPrime(1024)
n = p*q
e = 0x10001
# Encrypt the flag
m = bytes_to_long(flag)
c = pow(m, e, n)

print(f"n = {n}")
print(f"c = {c}")
print(f"hint = {p*q-p-q+1}")
# Output:
# n = 183043134996272788724973471067810887658449717529244949365811372943992515981220544919703526249978048915973685
# c = 326595170717224270972747273938687349470324991228550526537114639319603037241378180393016466314937222843933373
# hint = 1830431349962727887249734710678108876584497175292449493658113729439925159812205449197035262499780489159736
```

First of all, this is an RSA question, through the question we know $c \equiv m^e \bmod n$ then we can get $m \equiv c^d \bmod n$.

$$c \equiv m^e \bmod n$$

$$m \equiv c^d \bmod n.$$

To complete this equation, we need to get the value of d. d and e are two exponents that are modular and inverse to each other (exponent). So, we can know $d = e\ phin$. While $phi = (p-1) * (q-1)$, which equal $(pq - p - q + 1)$.

In the chal.py program, the hint already provides this value.

```
print(f"hint = {p*q-p-q+1}")
```

```
# hint = 1830431349962727887249734710678108876584497175292449493658113729439925159812205449197035262499780489159736
```

After we determine the idea of solving the problem, we create a python file called SilverDawn.py, and we import gmpy2 and binascii for decryption. Then paste the value provided in the question.

```
import gmpy2
import binascii

e = 0x10001
n = 1830431349962727887249734710678108876584497175292449493658113729439925159812205449197035262499780489159736857
c = 326595170717224270972747273938687349470324991228550526537114639319603037241378180393016466314937222843933373
hint = 1830431349962727887249734710678108876584497175292449493658113729439925159812205449197035262499780489159736
```

d = e phi, and the value of phi is equal to hint, so we import gmpy2.invert(e,hint)

```
d = gmpy2.invert(e,hint)
```

and m = c^d mod n, so we import gmpy2.powmod(c,d,n)

```
m = gmpy2.powmod(c,d,n)
```

Finally, use binascii to convert the value of m.

```
print(binascii.unhexlify(hex(m)[2:]))
```

The whole program of the file SilverDawnRSA.py

```
import gmpy2
import binascii

e = 0x10001
n = 1830431349962727887249734710678108876584497175292449493658113729439925159812205449197035262499780489159736857 2
c = 326595170717224270972747273938687349470324991228550526537114639319603037241378180393016466314937222843933373386
hint = 183043134996272788724973471067810887658449717529244949365811372943992515981220544919703526249978048915973685
d = gmpy2.invert(e,hint)
m = gmpy2.powmod(c,d,n)
print(binascii.unhexlify(hex(m)[2:]))
```

Output: