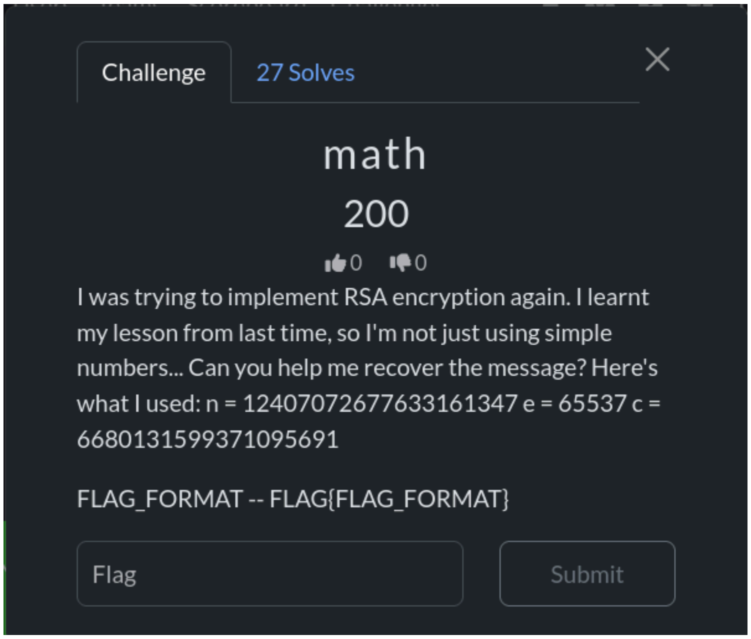
MATH

In this post i’ll be describing how i found a flag in a CTF organized by CyberSecure-X
So, this was a cryptography challenge and the challenge was called *MATH*

So, reading the description of the challenge we see its a RSA challenge basically we were given some value
The challenge read

> *“I was trying to implement RSA encryption again. I learnt my lesson from last time, so I'm not just using simple numbers... Can you help me recover the message? Here's what I used: n = 12407072677633161347 e = 65537 c = 6680131599371095691”*



math

200

👍0 👎0

I was trying to implement RSA encryption again. I learnt my lesson from last time, so I'm not just using simple numbers... Can you help me recover the message? Here's what I used: n = 12407072677633161347 e = 65537 c = 6680131599371095691

FLAG_FORMAT -- FLAG{FLAG_FORMAT}

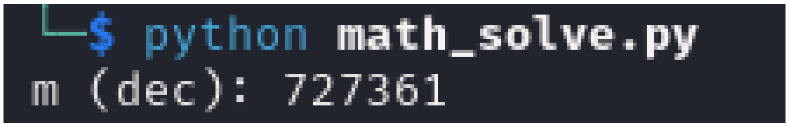
We are given three values
For calculating RSA we have to get
Two primes → p,q
Modulus → $n=p\times q$
Public exponent → e (commonly 65537)
Private exponent → d, where $d \equiv e^{-1}(\mathrm{mod}(p-1)(q-1))$

For this we could have used some online tool to calculate the remaining values
But using a terminal and some python was better for me
So i wrote a simple script to calculate the values

The script was
math_solve.py *

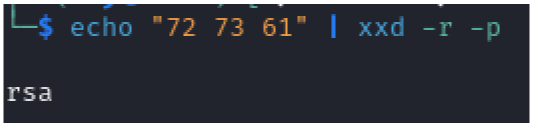
```
from sympy import factorint, mod_inverse

n = 1240707267763316134
e = 65537
c = 6680131599371095691

f = factorint(n)
p, q = list(f.keys())

phi = (p-1)*(q-1)
d = mod_inverse(e, phi)
m = pow(c, d, n)
print("m (dec):", m)
```

We needed the value of m which is the final rsa value and by running the script we got



```
└─$ python math_solve.py
m (dec): 727361
```

But wait what are these value?
Bas64? = n(doesn't look like any flag)
Base32? = Error: Invalid base32 characters(oh no..probably not this)
ASCII? = Nothing comes up
HEX? =



```
└─$ echo "72 73 61" | xxd -r -p

rsa
```

Why would the rsa literally say rsa after decoding?

Submitting the flag as flag{rsa}
-Correct
So the flag was - ✅ `flag{rsa}`