Cryptography

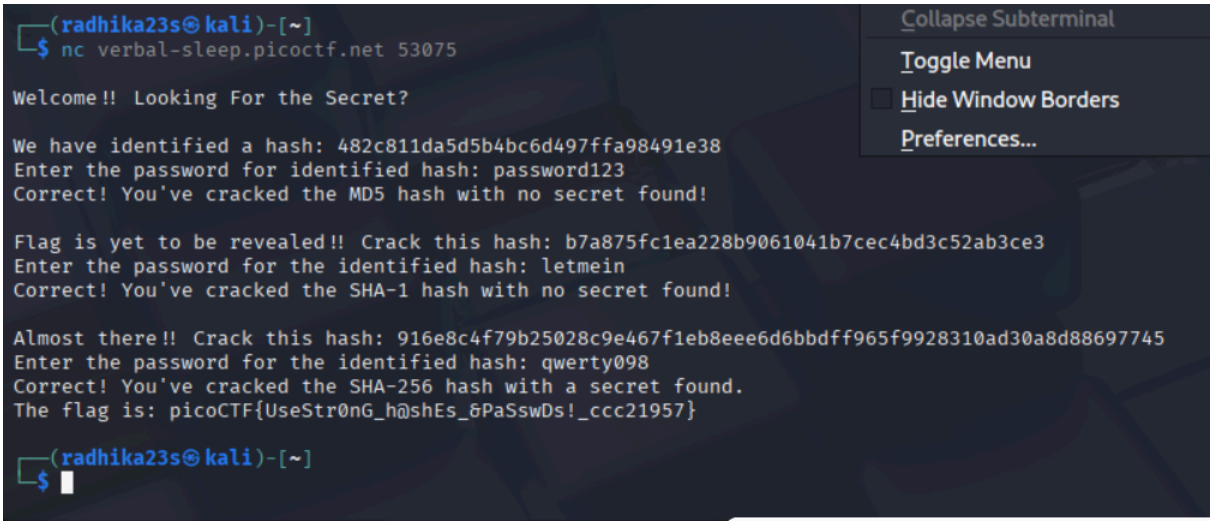
Hashcrack

Author: Nana Ama Atombo-Sackey

Description

A company stored a secret message on a server which got breached due to the admin using weakly hashed passwords. Can you gain access to the secret stored within the server?

Additional details will be available after launching your challenge instance.



Mod 26

Author: Pandu

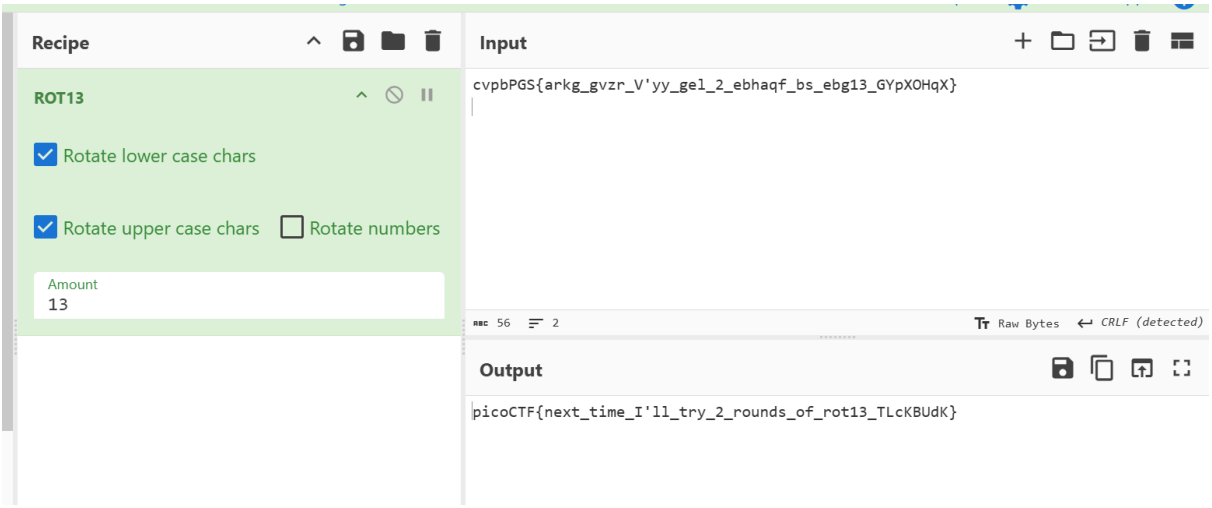
Description

Cryptography can be easy, do you know what ROT13 is? `cvpbPGS{arkg_gvzr_V'yy_gel_2_ebhaqf_bs_ebg13_GYpXOHqX}`

ROT13 is a simple letter substitution cipher that replaces a letter with the 13th letter after it in the Latin alphabet.

ROT13 is a special case of the Caesar cipher which was developed in ancient Rome, used by Julius Caesar in the 1st century BC.[1] An early entry on the Timeline of cryptography.

ROT13 can be referred by "Rotate13", "rotate by 13 places", hyphenated "ROT-13" or sometimes by its autonym "EBG13".



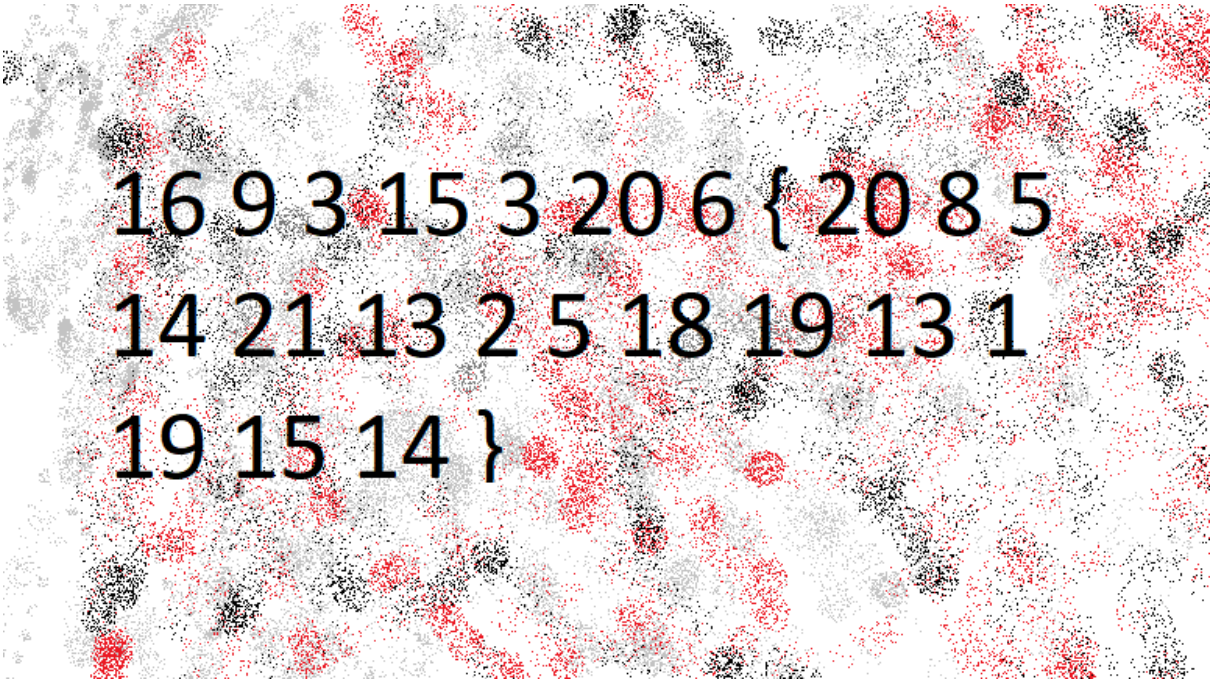
The Numbers

Author: Danny

Description

The numbers... what do they mean?

HINT PROVIDED:



This looks like a **number-to-letter cipher**, most likely **A=1, B=2, ..., Z=26**.

Let's decode:

- **16 9 3 15 3 20 6**
  - 16 = P
  - 9 = I
  - 3 = C
  - 15 = O
  - 3 = C
  - 20 = T
  - 6 = F

    → **PICOTF**
- **20 8 5 14 21 13 2 5 18 19 13 1 19 15 14**
  - 20 = T
  - 8 = H
  - 5 = E
  - 14 = N
  - 21 = U

- 13 = M
- 2 = B
- 5 = E
- 18 = R
- 19 = S
- 13 = M
- 1 = A
- 19 = S
- 15 = O
- 14 = N

  → **THENUMBERSMASON**

  So the whole thing reads:

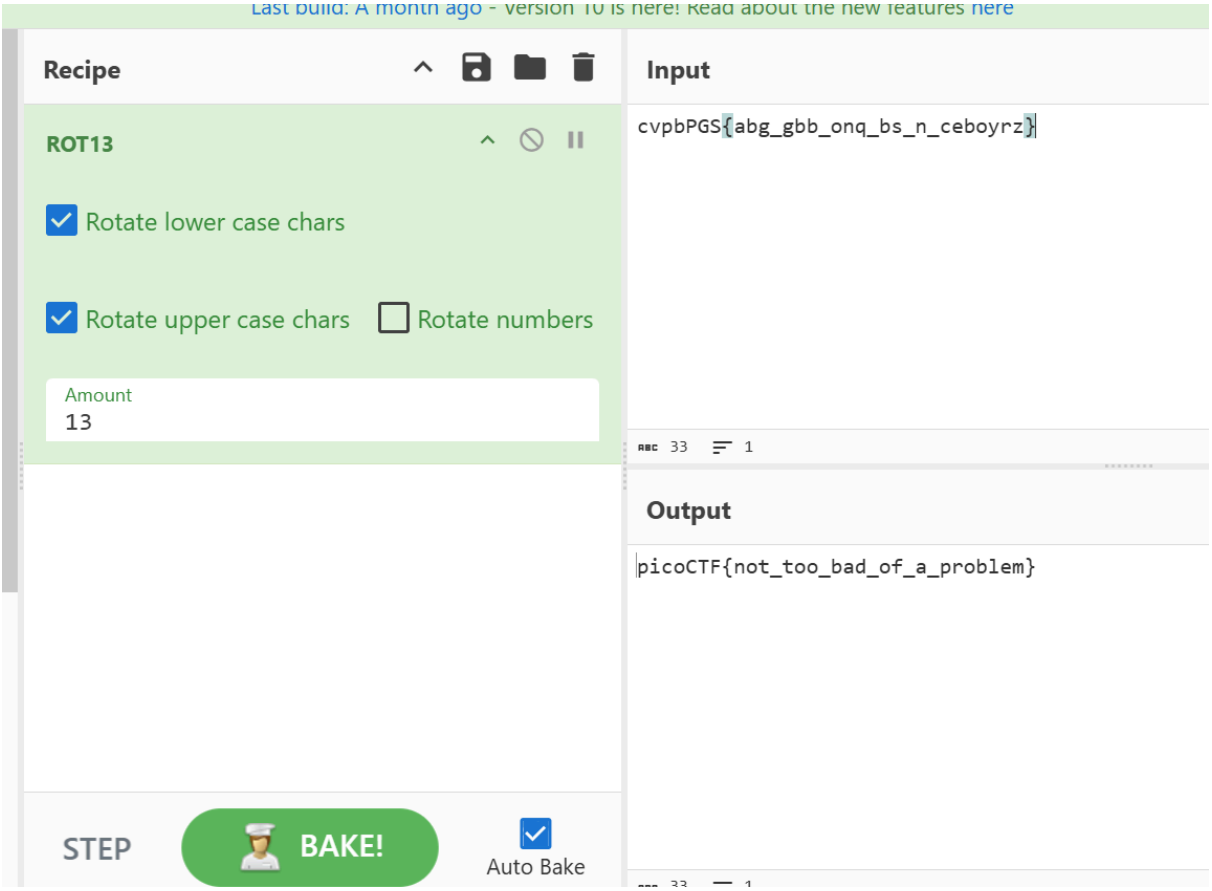
  picoctf{thenumbersmason}

13

Author: Alex Fulton/Daniel Tunitis

Description

Cryptography can be easy, do you know what ROT13 is? `cvpbPGS{abg_gbb_onq_bs_n_ceboyrz}`



interencdec

Author: NGIRIMANA Schadrack

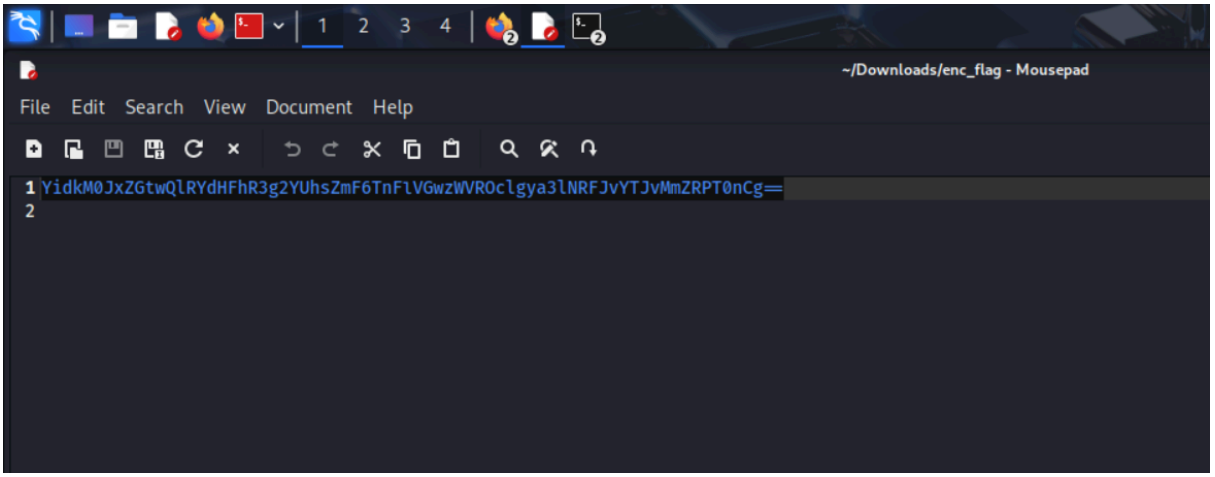
Description

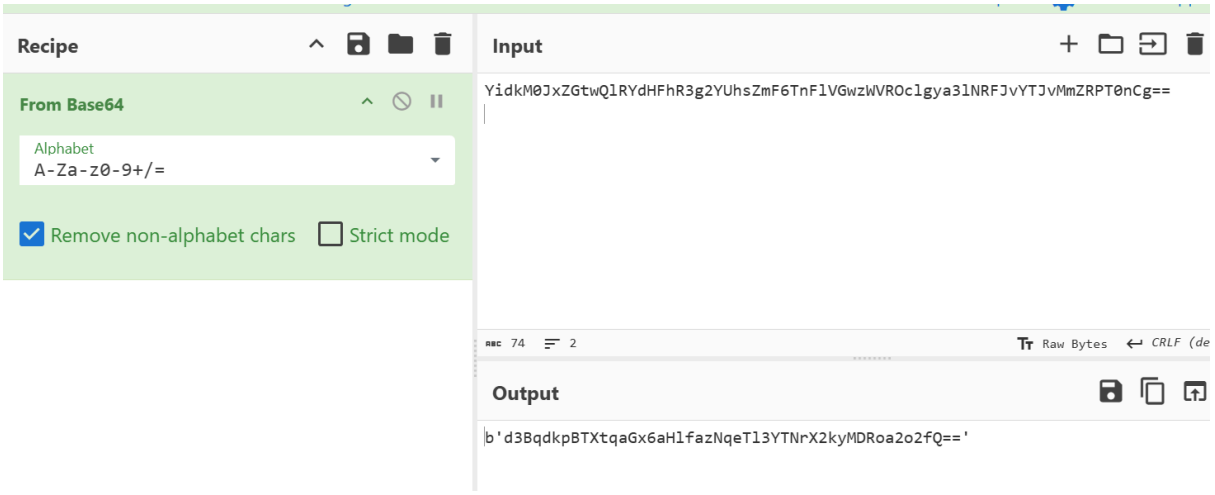
Can you get the real meaning from this file.Download the file here.

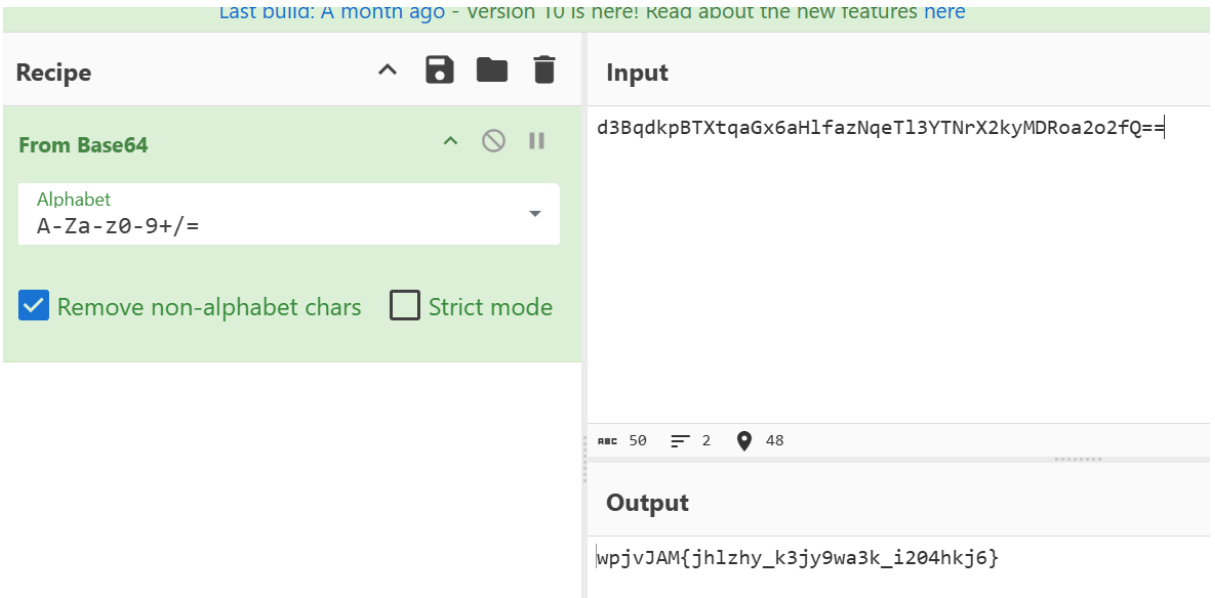
here

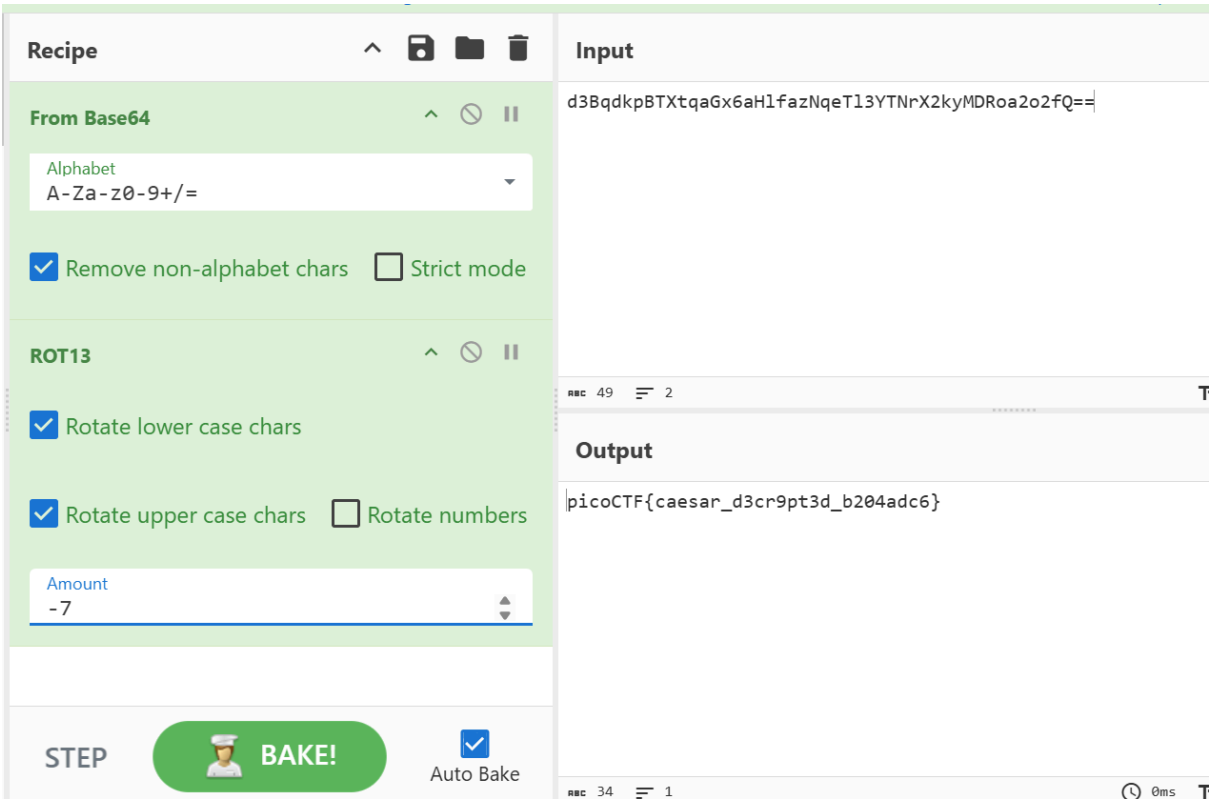
Hints:

Engaging in various decoding processes is of utmost importance


~/Downloads/enc_flag - Mousepad
File Edit Search View Document Help
1 YidkM0JxZGtwQlRYdHFhR3g2YUhsZmF6TnFlVGwzWVROclgya3lNRFJvYTJvMmZRPT0nCg==
2


Recipe
From Base64
Alphabet
A-Za-z0-9+/=
Remove non-alphabet chars
Strict mode
Input
YidkM0JxZGtwQlRYdHFhR3g2YUhsZmF6TnFlVGwzWVROclgya3lNRFJvYTJvMmZRPT0nCg==
74 2
Raw Bytes
CRLF (de
Output
b'd3BqdkpBTXtqaGx6aHlfazNqeTl3YTNrX2kyMDRoa2o2fQ=='


Last build: A month ago - Version 10 is here! Read about the new features here
Recipe
From Base64
Alphabet
A-Za-z0-9+/=
Remove non-alphabet chars
Strict mode
Input
d3BqdkpBTXtqaGx6aHlfazNqeTl3YTNrX2kyMDRoa2o2fQ==
50 2 48
Output
wpjvJAM{jhlzhy_k3jy9wa3k_i204hkj6}



- **From Base64** (twice pehle hi kar liya)
- **ROT13 / Caesar Cipher** → Operation: **ROT-N**
  - Shift: **7**
  - Direction: **Decode (left shift)**

Base64 is a notation for encoding arbitrary byte data using a restricted set of symbols that can be conveniently used by humans and processed by computers.

This operation decodes data

fr

om an ASCII Base64 string back into its raw format.

e.g.

```
aGVsbG8=
```

becomes

```
hello
```

EVEN RSA CAN BE BROKEN???

Author: Michael Crotty

Description

This service provides you an encrypted flag. Can you decrypt it with just N & e? Connect to the program with netcat: `$ nc verbal-sleep.picoctf.net 51569` The program's source code can be downloaded here.

here

from sys import exit
from Crypto.Util.number import bytes_to_long, inverse
from setup import get_primes

e = 65537

def gen_key(k):
"""
Generates RSA key with k bits
"""
p,q = get_primes(k//2)
N = p*q*
d = inverse(e, (p-1)(q-1))

```
return ((N,e), d)
```

def encrypt(pubkey, m):
N,e = pubkey
return pow(bytes_to_long(m.encode('utf-8')), e, N)

def main(flag):
pubkey, _privkey = gen_key(1024)
encrypted = encrypt(pubkey, flag)
return (pubkey[0], encrypted)

if **name** == "**main**":
flag = open('flag.txt', 'r').read()
flag = flag.strip()
N, cypher = main(flag)
print("N:", N)

```
print("e:", e)
print("cyphertext:", cypher)
exit()
```

What I did (short)

1. Factored NNN. It turned out to be even, so the factors are:

```
p = 2
q = 11857677260352208384117169986817086700026647292967486306289574907679320430566457558281627223560018290455854076615029016653405210492821320539247842887887409
```

1. Computed φ(N)=(p−1)(q−1)=q−1\varphi(N)=(p-1)(q-1)=q-1φ(N)=(p−1)(q−1)=q−1, then d=e−1 φ(N)d = e^{-1} \bmod \varphi(N)d=e−1modφ(N).
2. Decrypted m=cd Nm = c^d \bmod Nm=cdmodN and converted to bytes → got the flag above.

Python script you can run locally to reproduce the decryption:

```
from Crypto.Util.number import long_to_bytes, inverse

# Given values
N = 23715354520704416768234339973634173400053294585934972612579149815358640861132915116563254447120036580911708153230058033306810420985652641078495677185774818
e = 65537
c = 88699244209262762498034960277943352377944176919384117730052642521391162595511706944895802554964813994142192603910477879539371570030052761664158932932838 55

# Factors of N
p = 2
q = N // p

# Euler totient
phi = (p - 1) * (q - 1)
```

```
# Private exponent
d = inverse(e, phi)

# Decrypt
m = pow(c, d, N)

# Convert to bytes
flag = long_to_bytes(m)
print(flag.decode())
```

👉 This should output:

```
picoCTF{tw0_1$_pr!m3df98b648}
```

rotation

Author: Loic Shema

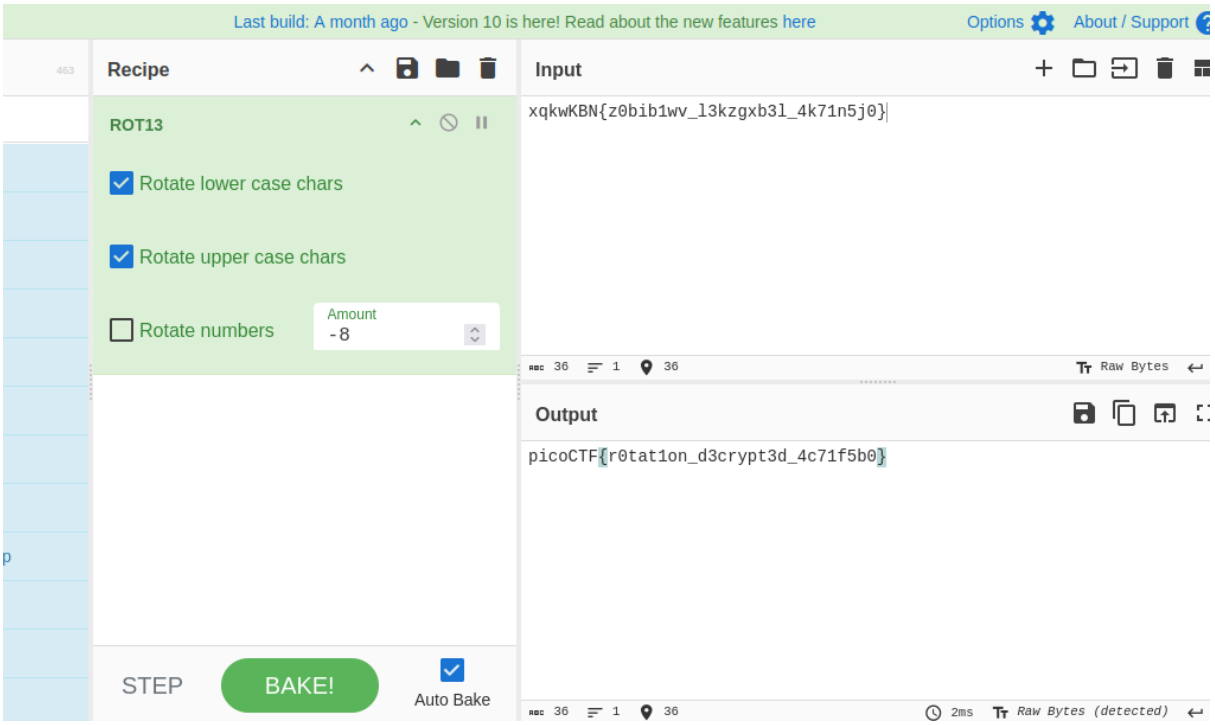
Description

You will find the flag after decrypting this file
Download the encrypted flag here.

here

Hints

Sometimes rotation is right



caesar

Author: Sanjay C/Daniel Tunitis

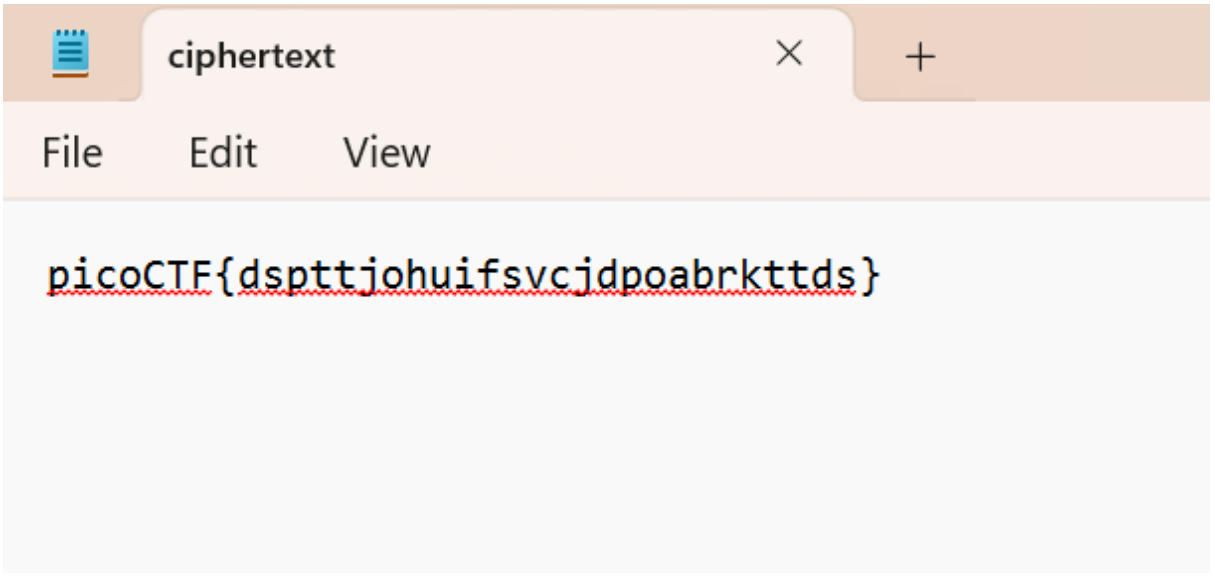
Description

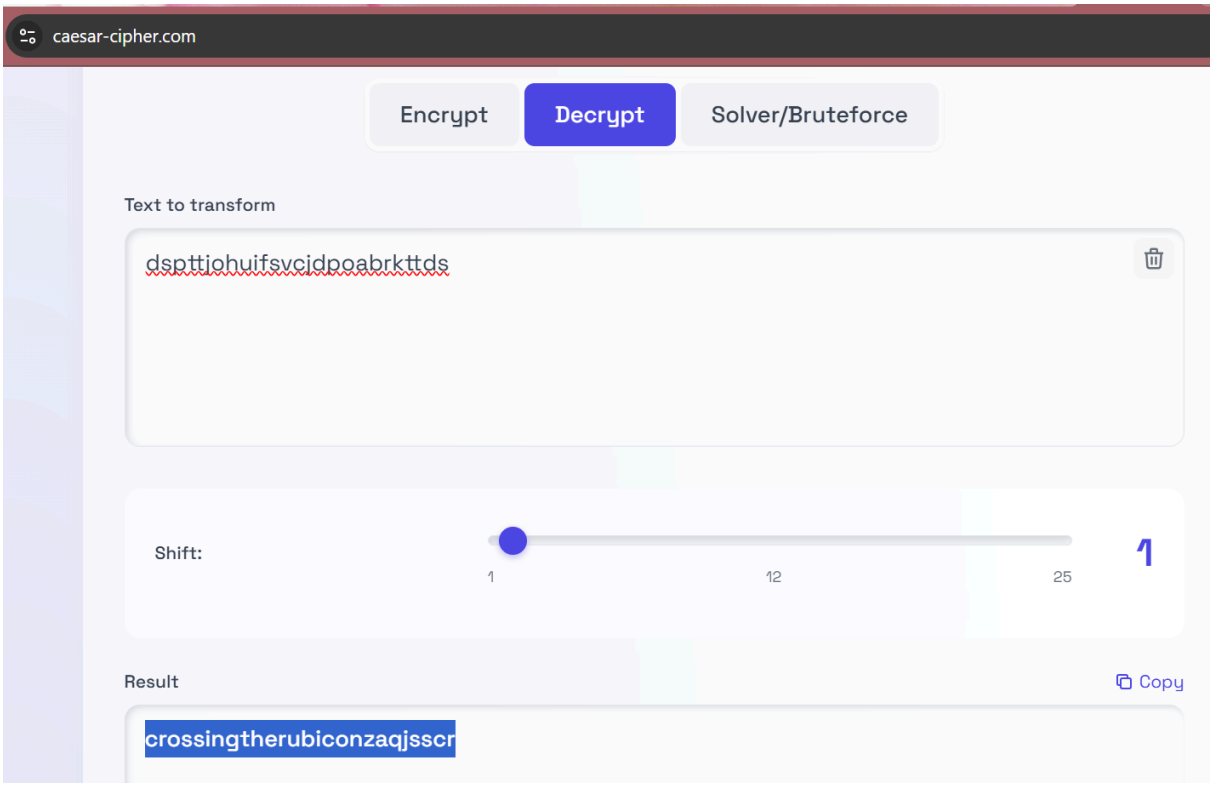
Decrypt this message.

debug info: [u:880571 e: p: c:64 i:153]

Hints

caesar cipher tutorial





so the flag is

picoCTF{**crossingtherubiconzaqjsscr}**