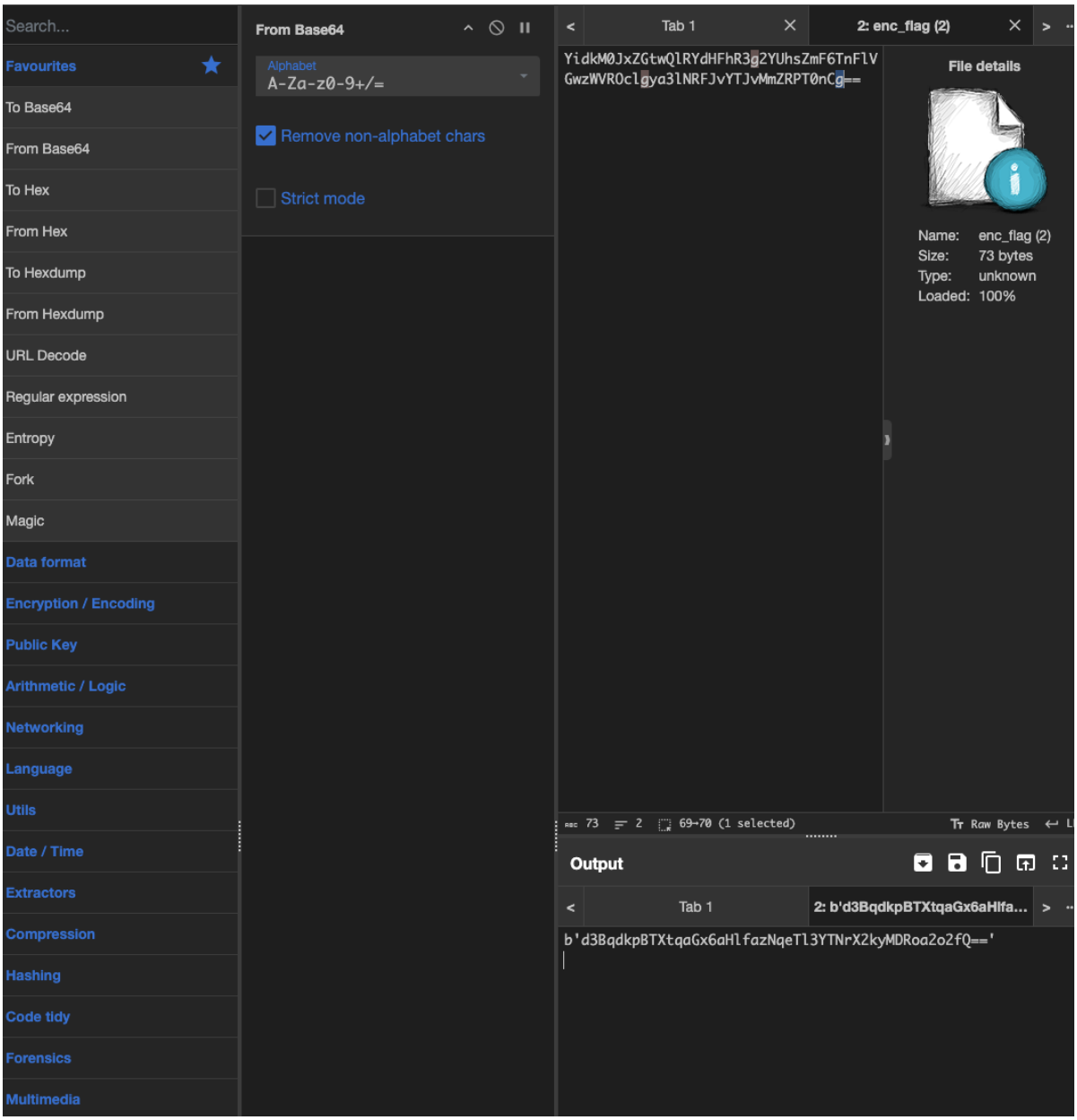
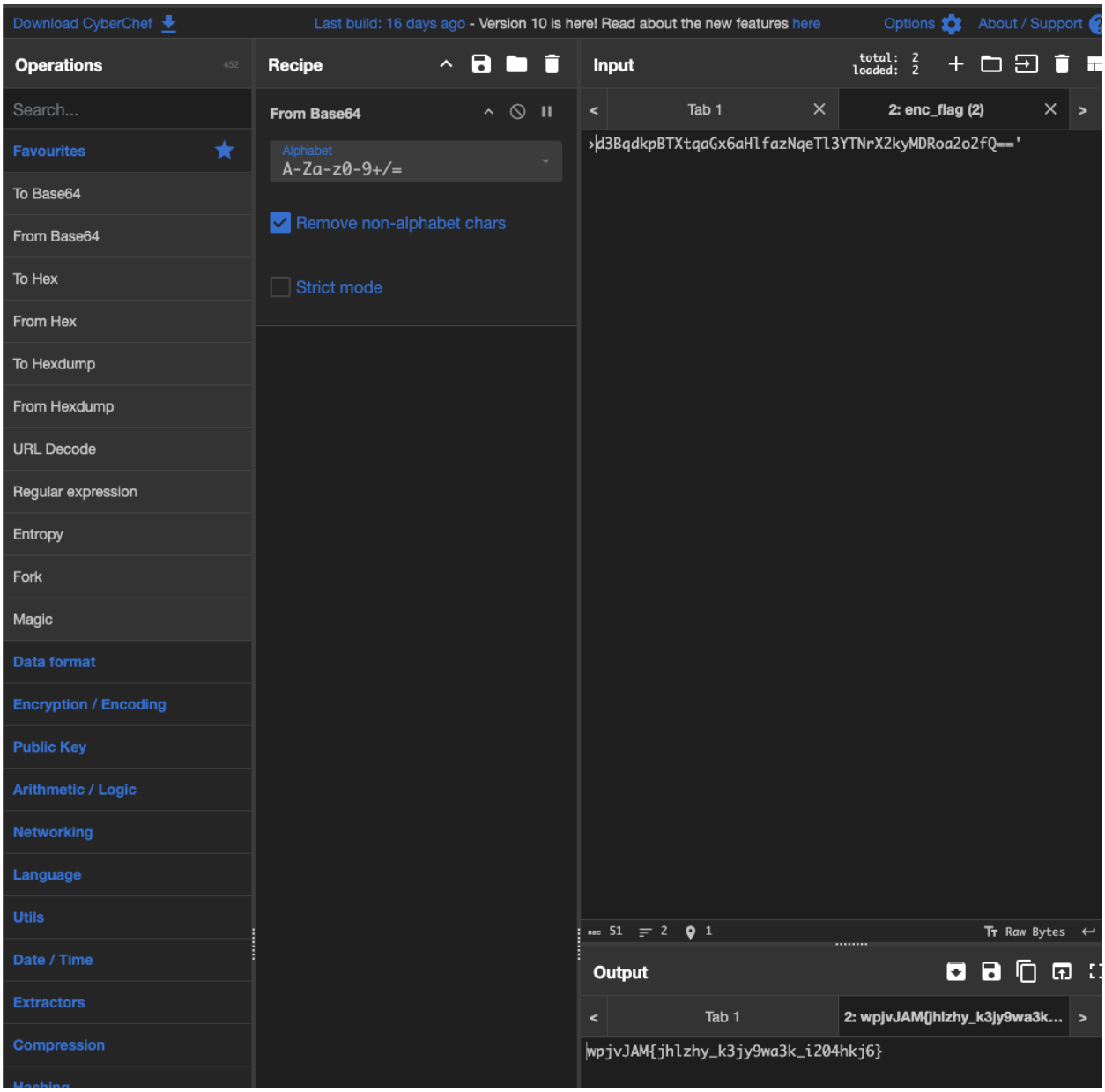
Exercise 1


Search...
Favourites
To Base64
From Base64
To Hex
From Hex
To Hexdump
From Hexdump
URL Decode
Regular expression
Entropy
Fork
Magic
Data format
Encryption / Encoding
Public Key
Arithmetic / Logic
Networking
Language
Utils
Date / Time
Extractors
Compression
Hashing
Code tidy
Forensics
Multimedia
From Base64
Alphabet
A-Za-z0-9+/=
Remove non-alphabet chars
Strict mode
Tab 1
2: enc_flag (2)
YidkM0JxZGtwQlRYdHFhR3g2YUhsZmF6TnFlV
GwzWVROclgya3lNRFJvYTJvMmZRPT0nCg==
File details
Name: enc_flag (2)
Size: 73 bytes
Type: unknown
Loaded: 100%
73
2
69-70 (1 selected)
Raw Bytes
Output
Tab 1
2: b'd3BqdkpBTXtqaGx6aHlfa...
b'd3BqdkpBTXtqaGx6aHlfazNqeTl3YTNrX2kyMDRoa2o2fQ==''


Download CyberChef
Last build: 16 days ago - Version 10 is here! Read about the new features here
Options
About / Support
Operations
Search...
Favourites
To Base64
From Base64
To Hex
From Hex
To Hexdump
From Hexdump
URL Decode
Regular expression
Entropy
Fork
Magic
Data format
Encryption / Encoding
Public Key
Arithmetic / Logic
Networking
Language
Utils
Date / Time
Extractors
Compression
Recipe
From Base64
Alphabet
A-Za-z0-9+/=
Remove non-alphabet chars
Strict mode
Input
total: 2
loaded: 2
Tab 1
2: enc_flag (2)
d3BqdkpBTXtqaGx6aHlfazNqeTl3YTNrX2kyMDRoa2o2fQ==
Raw Bytes
Output
Tab 1
2: wpjvJAM{jhlzhy_k3jy9wa3k...
wpjvJAM{jhlzhy_k3jy9wa3k_i204hkj6}

Operations

rot

ROT13

ROT47

ROT8000

Rotate left

Rotate Image

Rotate right

ROT13 Brute Force

ROT47 Brute Force

Parse ObjectID timestamp

Avro to JSON

From UNIX Timestamp

From Octal

Protobuf Decode

Protobuf Encode

Drop bytes

From Float

Remove Diacritics

Remove null bytes

Remove whitespace

Drop nth bytes

From HTML Entity

From Hex Content

Randomize Colour Palette

From Quoted Printable

Generate HOTP

Generate TOTP

Optical Character Recognition

Cartesian Product

From Case Insensitive Regex

Microsoft Script Decoder

Public Key from Certificate

Public Key from Private Key

Parse SSH Host Key

Translate DateTime Format

Convert Image Format

Generic Code Beautify

Convert to NATO alphabet

Recipe

ROT13 Brute Force

- [x] Rotate lower case chars
- [x] Rotate upper case chars
- [ ] Rotate numbers

Sample length: 100

Sample offset: 0

- [x] Print amount

Crib (known plaintext string)

Input

total: 2
loaded: 2

Tab 1 | 2: enc_flag (2)

```
wpjvJAM{jhlzhy_k3jy9wa3k_i204hkj6}
```

34 1 Raw Bytes

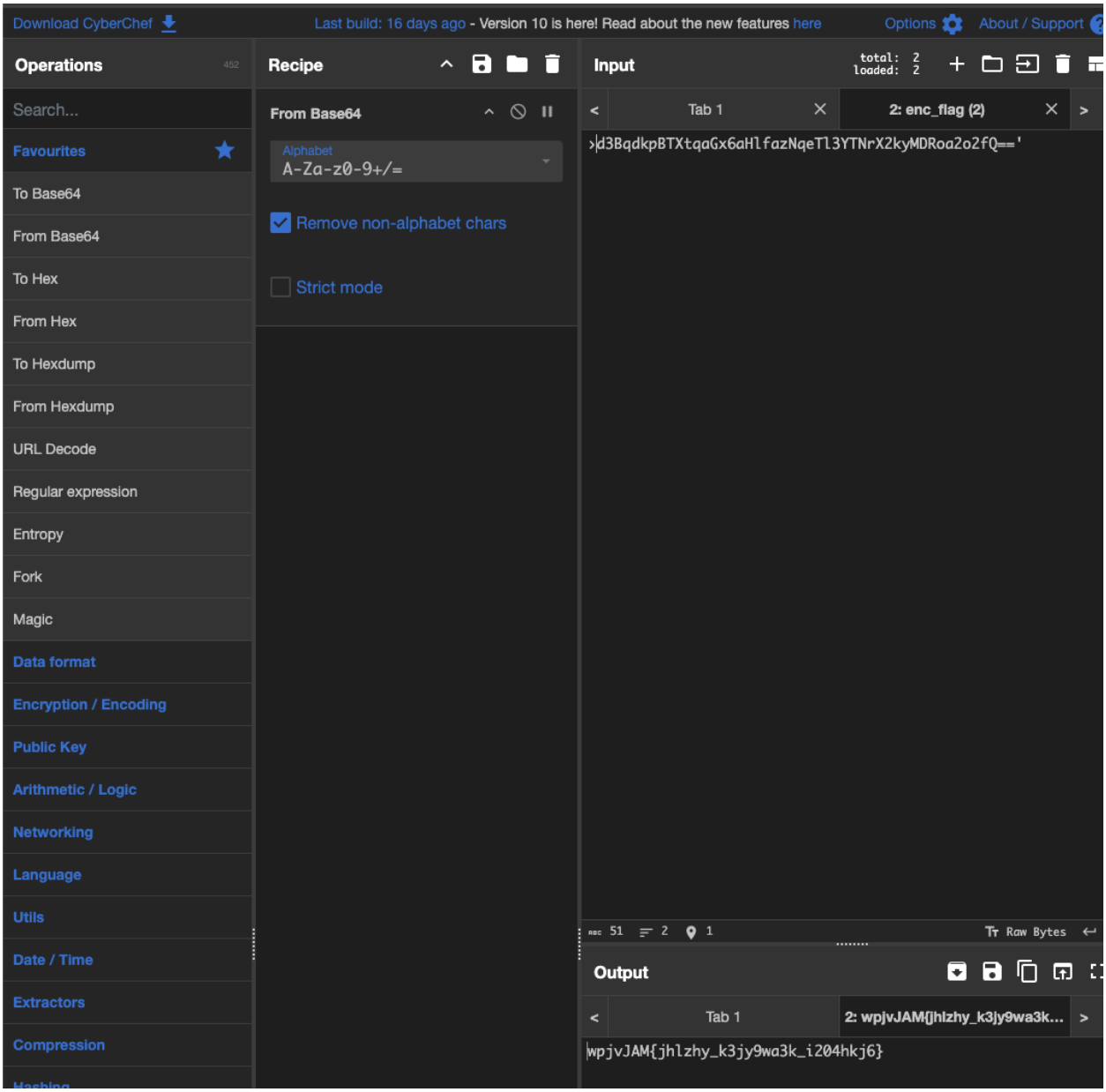
Output

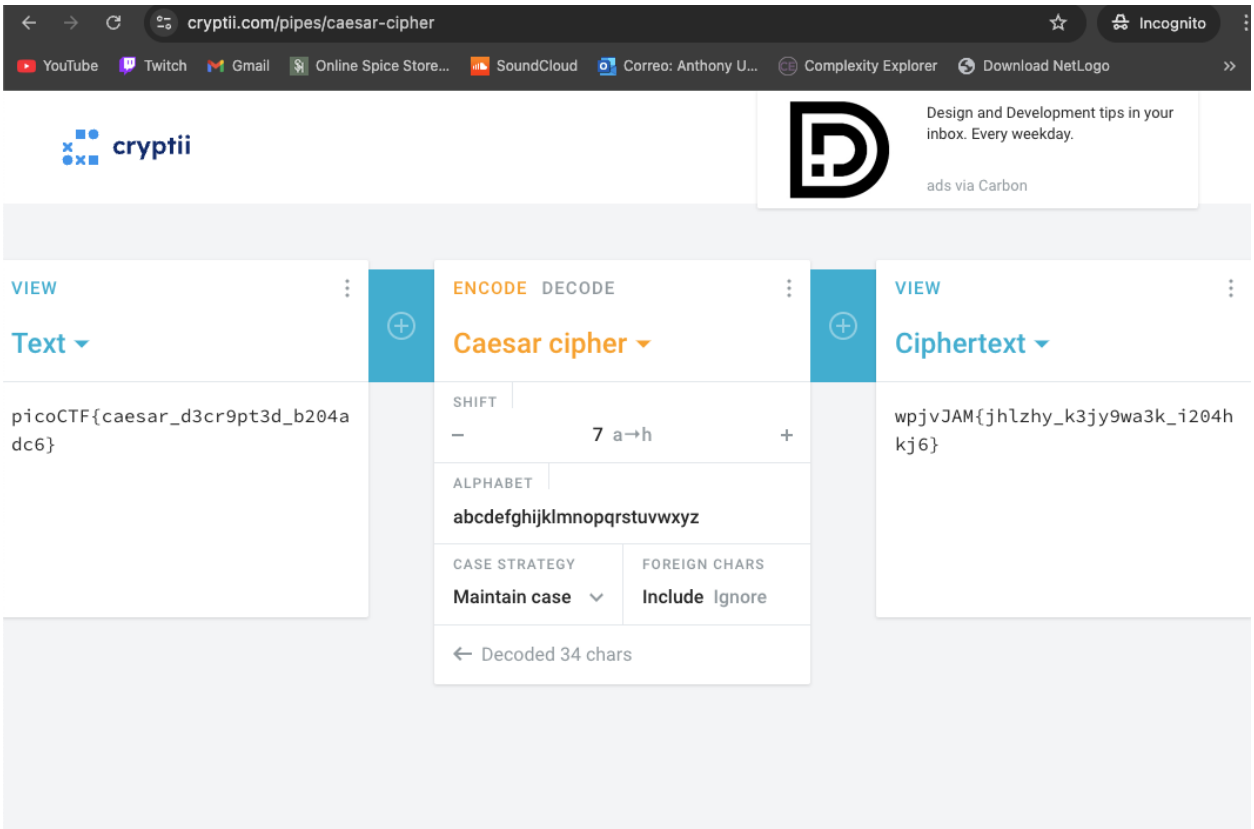
Tab 1 | 2: Amount = 1: xqkwKBN{kim...

```
Amount =  1: xqkwKBN{kimaiz_l3kz9xb3l_j204ilk6}
Amount =  2: yrlxLCO{ljnbja_m3la9yc3m_k204jml6}
Amount =  3: zsmyMDP{mkockb_n3mb9zd3n_l204knm6}
Amount =  4: atnzNEQ{nlpdlc_o3nc9ae3o_m204lon6}
Amount =  5: buoaOFR{omqemd_p3od9bf3p_n204mpo6}
Amount =  6: cvpbPGS{pnrfne_q3pe9cg3q_o204nqp6}
Amount =  7: dwqcQHT{qosgof_r3qf9dh3r_p204orq6}
Amount =  8: exrdRIU{rpthpg_s3rg9ei3s_q204psr6}
Amount =  9: fyseSJV{squiqh_t3sh9fj3t_r204qts6}
Amount = 10: gztfTKW{trvjri_u3ti9gk3u_s204rut6}
Amount = 11: haugULX{uswksj_v3uj9hl3v_t204svu6}
Amount = 12: ibvhVMY{vtxltk_w3vk9im3w_u204twv6}
Amount = 13: jcwiWNZ{wuymul_x3wl9jn3x_v204uxw6}
Amount = 14: kdxjXOA{xvznvm_y3xm9ko3y_w204vyx6}
Amount = 15: leykYPB{ywaown_z3yn9lp3z_x204wzy6}
Amount = 16: mfzlZQC{zxbpxo_a3zo9mq3a_y204xaz6}
Amount = 17: ngamARD{aycqyp_b3ap9nr3b_z204yba6}
Amount = 18: ohbnBSE{bzdrzq_c3bq9os3c_a204zcb6}
Amount = 19: picoCTF{caesar_d3cr9pt3d_b204adc6}
Amount = 20: qjdpDUG{dbftbs_e3ds9qu3e_c204bed6}
Amount = 21: rkeqEVH{ecguct_f3et9rv3f_d204cfe6}
Amount = 22: slfrFWI{fdhvdu_g3fu9sw3g_e204dgf6}
Amount = 23: tmgsGXJ{geiwev_h3gv9tx3h_f204ehg6}
Amount = 24: unhtHYK{hfjxfw_i3hw9uy3i_g204fih6}
Amount = 25: voiuIZL{igkygx_j3ix9vz3j_h204gji6}
```

picoCTF{caesar_d3cr9pt3d_b204adc6}

Second Solution




cryptii.com/pipes/caesar-cipher
Incognito
YouTube
Twitch
Gmail
Online Spice Store...
SoundCloud
Correo: Anthony U...
Complexity Explorer
Download NetLogo
cryptii
Design and Development tips in your inbox. Every weekday.
ads via Carbon
VIEW
Text
picoCTF{caesar_d3cr9pt3d_b204adc6}
ENCODE DECODE
Caesar cipher
SHIFT
7 a→h
ALPHABET
abcdefghijklmnopqrstuvwxyz
CASE STRATEGY
Maintain case
FOREIGN CHARS
Include Ignore
Decoded 34 chars
VIEW
Ciphertext
wpjvJAM{jhlzhy_k3jy9wa3k_i204hkj6}

Exercise 2

EXPLORADOR

ODIN_APP

- .dart_tool
- .idea
- android
- assets
- build
- ios
- lib
  - data
    - response_models.dart
    - services.dart
    - session_manager.dart
    - token_generator.dart
  - di
    - injector.dart
  - ui
    - common
      - bm_flat_button.dart
      - bm_outlined_button....
      - bm_outlined_rounde...
      - bm_text_button.dart
      - circular_button.dart
    - home
      - home_page.dart
      - sale_card.dart
    - line
      - line_details_page.dart
    - login
      - login_bloc.dart
      - login_page.dart
    - sale
      - edit_quantity_dialog....
      - line_card.dart
      - sale_bloc.dart
      - sale_page.dart
    - splash
      - animated_splash_pa...
      - loading_painter.dart
      - splash_bloc.dart
      - splash_customizer.dart
    - app_navigator.dart
    - ui_theme.dart
  - main.dart
- linux
  - flutter
  - .gitignore
  - CMakeLists.txt
  - main.cc
  - my_application.cc
  - my_application.h
- macos
- test
- web
- windows
- .flutter-plugins
- .flutter-plugins-depende...
- .gitattributes
- .gitignore
- .metadata
- analysis_options.yaml
- odin_app.iml
- pubspec.lock
- pubspec.yaml

bloc.dart | token_generator.dart | main.dart | Exercise2.py

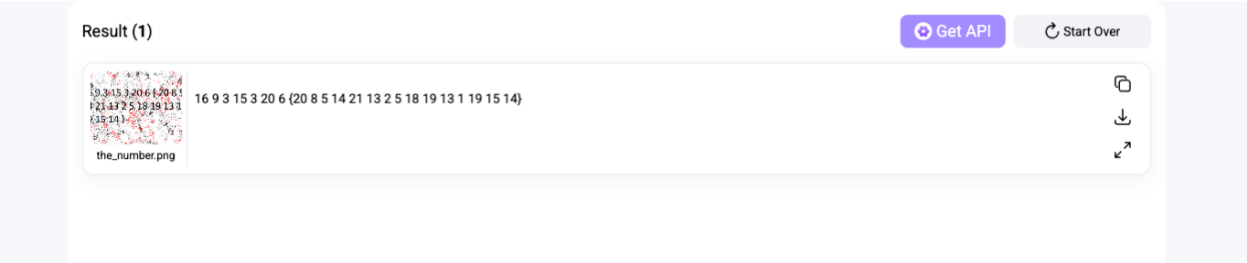
Drive-Condimensa > 9no semestre > Aplicaciones Distribuidas > Computer Security > Exercise2.py

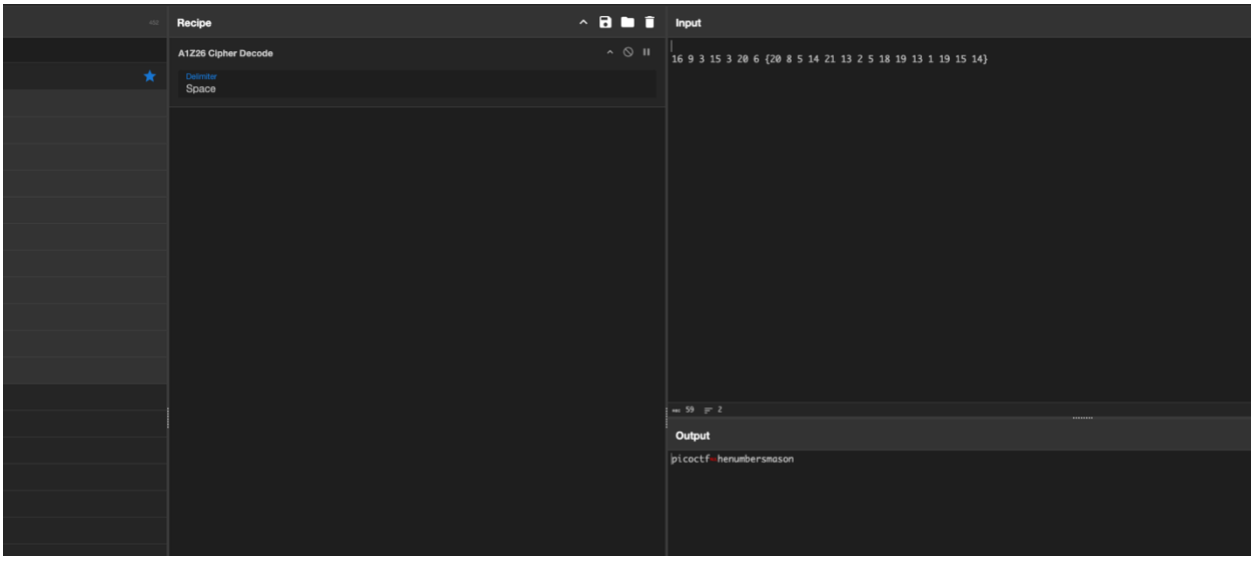
```python
def a1z26_to_text(numbers):
    """
    Converts a list of numbers into text using the A1Z26 cipher.
    """
    return ''.join(chr(num + 64) for num in numbers)

# Example list of numbers (you can change this list for another challe
numbers = [16, 9, 3, 15, 3, 20, 6, 20, 8, 5, 14, 21, 13, 2, 5, 18, 19,

# Decode the numbers
decoded_text = a1z26_to_text(numbers)

# Format the decoded text into the flag format
flag = f"picoCTF{{{decoded_text.lower()}}}"

# Print the found flag
print("Flag found:", flag)
```

PROBLEMAS 75 | SALIDA | TERMINAL | Python

```
ulisesfajardo@MacBook-Pro-3 odin_app % /opt/homebrew/bin/python3 "/Users/ulisesfajardo/Libra
ry/CloudStorage/OneDrive-Condimensa/9no semestre/Aplicaciones Distribuidas/Computer Security
/Exercise2.py"
Flag encontrada: picoCTF{picoctfthenumbersmason}
ulisesfajardo@MacBook-Pro-3 odin_app % /opt/homebrew/bin/python3 "/Users/ulisesfajardo/Libra
ry/CloudStorage/OneDrive-Condimensa/9no semestre/Aplicaciones Distribuidas/Computer Security
/Exercise2.py"
Flag found: picoCTF{picoctfthenumbersmason}
ulisesfajardo@MacBook-Pro-3 odin_app %
```

Second Solution





Exercise 3

...

ken_generator.dart | main.dart | Exercise2.py | convert.py

Users > ulisesfajardo > Downloads > convert.py > ...

```python
import sys
chars = ""
from fileinput import input
for line in input():
  chars += line

lookup1 = "\n \"#()*+/1:=[]abcdefghijklmnopqrstuvwxyz"
lookup2 = "ABCDEFGHIJKLMNOPQRSTabcdefghijklmnopqrst"

out = ""

prev = 0
for char in chars:
  cur = lookup1.index(char)
  out += lookup2[(cur - prev) % 40]
  prev = cur

sys.stdout.write(out)
```

odels.dart
t
nager.dart
rator.dart

utton.dart
ed_button....
ed_rounde...
utton.dart
utton.dart

e.dart
dart

odin_app

main.dart | Exercise2.py | convert.py | convert2.py

Users > ulisesfajardo > Downloads > convert2.py > ...

```python
import sys
chars = ""
from fileinput import input
for line in input():
    chars += line

lookup1 = "\n \"#()*+/1:=[]abcdefghijklmnopqrstuvwxyz"
lookup2 = "ABCDEFGHIJKLMNOPQRSTabcdefghijklmnopqrst"

out = ""

prev = 0
for char in chars:
    cur = lookup1.index(char)
    out += lookup2[(cur - prev) % 40]
    prev = cur

sys.stdout.write(out)
```

```
AndroidStudioProjects                    Proyecto Informe
Applications                             Public
Applications (Parallels)                 Qt
Cisco Packet Tracer 8.2.2                QtDesignStudio
Desktop                                  Sites
Development                              Virtual Machines.localized
Documents                                VirtualBox VMs
Downloads                                bdpuzzle.txt
GNS3                                     build-untitled-Qt_6_3_1_for_macOS-Debug
IdeaProjects                             client
Library                                  convert.py
LocaChangeResource                       discoclient.properties
Movies                                   dumps
Music                                    java_error_in_idea_9498.log
NetBeansJDKs                             node_modules
NetBeansProjects                         package-lock.json
OneDrive - Condimensa                    package.json
Parallels                                server
Pictures                                 tomcat-native-2.0.8-src
                                         untitled
ulisesfajardo@MacBook-Pro-3 ~ % cd Downloads
ulisesfajardo@MacBook-Pro-3 Downloads % cp convert.py convert2.py
ulisesfajardo@MacBook-Pro-3 Downloads %
```

```
ulisesfajardo@MacBook-Pro-3 Downloads % python3 convert.py < ciphertext
Texto descifrado:
#asciiorder
#fortychars
#selfinput
#pythontwo

chars = ""
from fileinput import input
for line in input():
    chars += line
b = 1 / 1

for i in range(len(chars)):
    if i == b * b * b:
        print chars[i] #prints
        b += 1 / 1
```

picoCTF{adlibs}

Second Solution

Storage > OneDrive-Condimensa > 9no semestre > Computer Security > Exercise 3 > exercise3

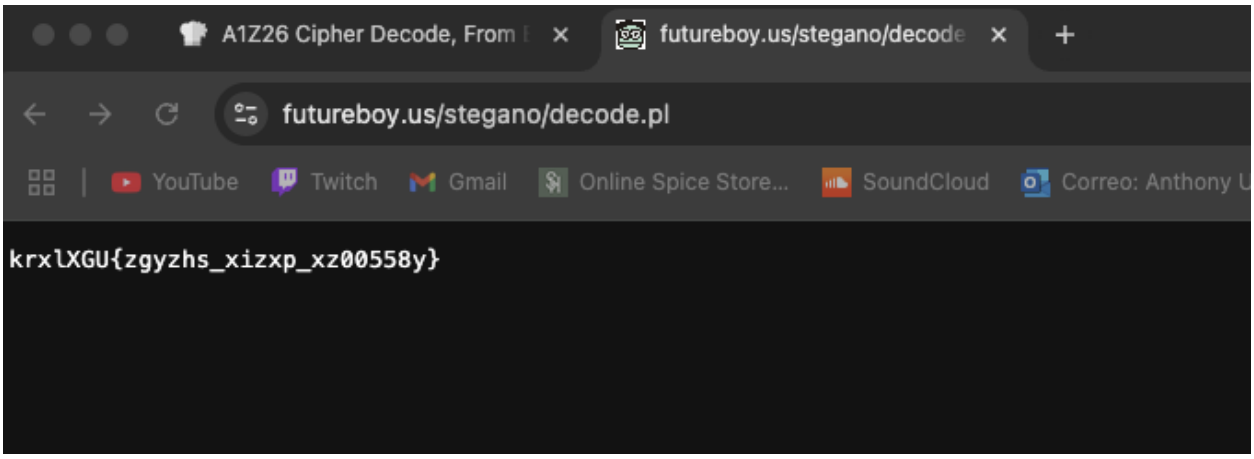
```python
import sys

# Tablas de conversión del cifrado
lookup1 = "\n \"#()*+/1:=[]abcdefghijklmnopqrstuvwxyz"
lookup2 = "ABCDEFGHIJKLMNOPQRSTabcdefghijklmnopqrst"

# Leer el texto cifrado desde el archivo
with open("ciphertext", "r") as file:
    chars = file.read()

# Variable para almacenar el texto descifrado
out = ""
prev = 0

# Proceso de descifrado (inverso al cifrado en convert.py)
for char in chars:
    cur = lookup2.index(char)  # Encuentra el índice en lookup2
    decoded_index = (cur + prev) % 40  # Inversa de la transformación
    out += lookup1[decoded_index]  # Obtener el carácter original
    prev = decoded_index

# Formatear la flag como picoCTF
flag = f"picoCTF{{{out.strip()}}}"
print("Flag encontrada:", flag)
```

oudStorage > OneDrive-Condimensa > 9no semestre > Computer Security > Exercise 3 > convert_

```python
# Conversión del código a Python 3
from fileinput import input

# Leer el contenido del archivo de entrada
chars = ""  # Inicializamos la variable
for line in input():
    chars += line

b = 1  # En Python 3, la división debe mantenerse como entero

# Extraer caracteres en posiciones de cubos perfectos
decoded_text = ""
for i in range(len(chars)):
    if i == b ** 3:  # Verificamos si el índice es un cubo perfecto
        decoded_text += chars[i]  # Agregamos el carácter a la salida
        b += 1  # Incrementamos b

print("Flag encontrada:", f"picoCTF{{{decoded_text.strip()}}}")
```

```
enc: Use -help for summary.
ulisesfajardo@MacBook-Pro-3 Downloads % openssl enc -aes-256-cbc -d -in secrect.
enc -k da099
Can't open secrect.enc for reading, No such file or directory
8595964416:error:02001002:system library:fopen:No such file or directory:crypto/
bio/bss_file.c:69:fopen('secrect.enc','rb')
8595964416:error:2006D080:BIO routines:BIO_new_file:no such file:crypto/bio/bss_
file.c:76:
ulisesfajardo@MacBook-Pro-3 Downloads % openssl enc -aes-256-cbc -d -in secret.e
nc -k da099
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
picoCTF{su((3ss_(r@ck1ng_r3@_da099d93}%
ulisesfajardo@MacBook-Pro-3 Downloads % cd ..
ulisesfajardo@MacBook-Pro-3 ~ % cd OneDrive\ -\ Condimensa
ulisesfajardo@MacBook-Pro-3 OneDrive - Condimensa % cd 9no\ semestre
ulisesfajardo@MacBook-Pro-3 9no semestre % cd Computer\ Security
ulisesfajardo@MacBook-Pro-3 Computer Security % cd Exercise
cd: no such file or directory: Exercise
ulisesfajardo@MacBook-Pro-3 Computer Security % cd Exercise\ 3
ulisesfajardo@MacBook-Pro-3 Exercise 3 % python3 convert_python3.py < ciphertext


Flag encontrada: picoCTF{LgHDPt}
ulisesfajardo@MacBook-Pro-3 Exercise 3 %
```

Exercise 4



Chrome File Edit View History Bookmarks Profiles Tab Window Help

A1Z26 Cipher Decode, From | futureboy.us/stegano/decode | Atbash Cipher - CyberChef

gchq.github.io/CyberChef/#recipe=Atbash_Cipher()&input=a3J4bFhHVXt6Z3l6aHNfeGl...

YouTube Twitch Gmail Online Spice Store... SoundCloud Correo: Anthony U... Complexity Explorer

Download CyberChef

Last build: 16 days ago - Version 10 is here! Read about the new features here

Operations 452

Search...

Favourites

To Base64

From Base64

To Hex

From Hex

To Hexdump

From Hexdump

URL Decode

Regular expression

Entropy

Fork

Magic

Data format

Encryption / Encoding

Public Key

Arithmetic / Logic

Networking

Language

Utils

Date / Time

Extractors

Compression

Hashing

Recipe

Atbash Cipher

Input

krxlXGU{zgyzhs_xizxp_xz00558y}

30 1

Output

picoCTF{atbash_crack_ca00558b}

Second Solution

```
--->  No broken files found.
--->  No broken ports found.
ulisesfajardo@MacBook-Pro-3 Downloads % steghide extract -sf atbash.jpg
Anotar salvoconducto:
anot⍰ los datos extra⍰dos e/"encrypted.txt".
ulisesfajardo@MacBook-Pro-3 Downloads % cat encrypted.txt
krxlXGU{zgyzhs_xizxp_xz00558y}
ulisesfajardo@MacBook-Pro-3 Downloads %
```

Exercise 5

rsa_oracle

Medium | Cryptography | picoCTF 2024 | browser_webshell_solvable

AUTHOR: GEOFFREY NJOGU

Congratulations! You've solved this challeng Hint 1
Assignment: HW1 (CMP-6006-SP2025)

Description

Can you abuse the oracle?
An attacker was able to intercept communications between a bank and a fintech company. They managed to get the message (ciphertext) and the password that was used to encrypt the message.
After some intensive reconassainance they found out that the bank has an oracle that was used to encrypt the password and can be found here `nc titan.picoctf.net 54618`. Decrypt the password and use it to decrypt the message. The oracle can decrypt anything except the password.

This challenge launches an instance on demand.
Its current status is: RUNNING
Instance Time Remaining: 28:43

Restart Instance

Hints

1 2 3 4

Crytography Threat models: chosen plaintext attack.

1,943 users solved

95% Liked

picoCTF{FLAG}

Submit Flag

rsa_oracle

Medium | Cryptography | picoCTF 2024 | browser_webshell_solvable

AUTHOR: GEOFFREY NJOGU

Congratulations! You've solved this challenge!
Assignment: HW1 (CMP-5006-SP2025)

Description

Can you abuse the oracle?
An attacker was able to intercept communications between a bank and a fintech company. They managed to get the message (ciphertext) and the password that was used to encrypt the message.
After some intensive reconassainance they found out that the bank has an oracle that was used to encrypt the password and can be found here `nc titan.picoctf.net 54618`. Decrypt the password and use it to decrypt the message. The oracle can decrypt anything except the password.

This challenge launches an instance on demand.
Its current status is: RUNNING
Instance Time Remaining: 29:53

Restart Instance

Hints

1 2 3 4

OpenSSL can be used to decrypt the message. e.g `openssl enc -aes-256-cbc -d ...`

1,943 users solved

95% Liked

picoCTF{FLAG}

Submit Flag

Forensics — Easy — Secret of the Polyglot

Web Exploitation — Easy — IntroToBurp

Cryptography — interencdec

rsa_oracle

Medium | Cryptography | picoCTF 2024 | browser_webshell_solvable

AUTHOR: GEOFFREY NJOGU

Congratulations! You've solved this challenge!
Assignment: HW1 (CMP-5006-SP2025)

Description

Can you abuse the oracle?
An attacker was able to intercept communications between a bank and a fintech company. They managed to get the message (ciphertext) and the password that was used to encrypt the message.
After some intensive reconassainance they found out that the bank has an oracle that was used to encrypt the password and can be found here `nc titan.picoctf.net 54618`. Decrypt the password and use it to decrypt the message. The oracle can decrypt anything except the password.

This challenge launches an instance on demand.
Its current status is: RUNNING
Instance Time Remaining: 28:00

Restart Instance

Hints

1 2 3 4

The key to getting the flag is by sending a custom message to the server by taking advantage of the RSA encryption algorithm.

1,943 users solved

95% Liked

picoCTF{FLAG}

Submit Flag

Forensics — Easy — Secret of the Polyglot

Web Exploitation — Easy — IntroToBurp

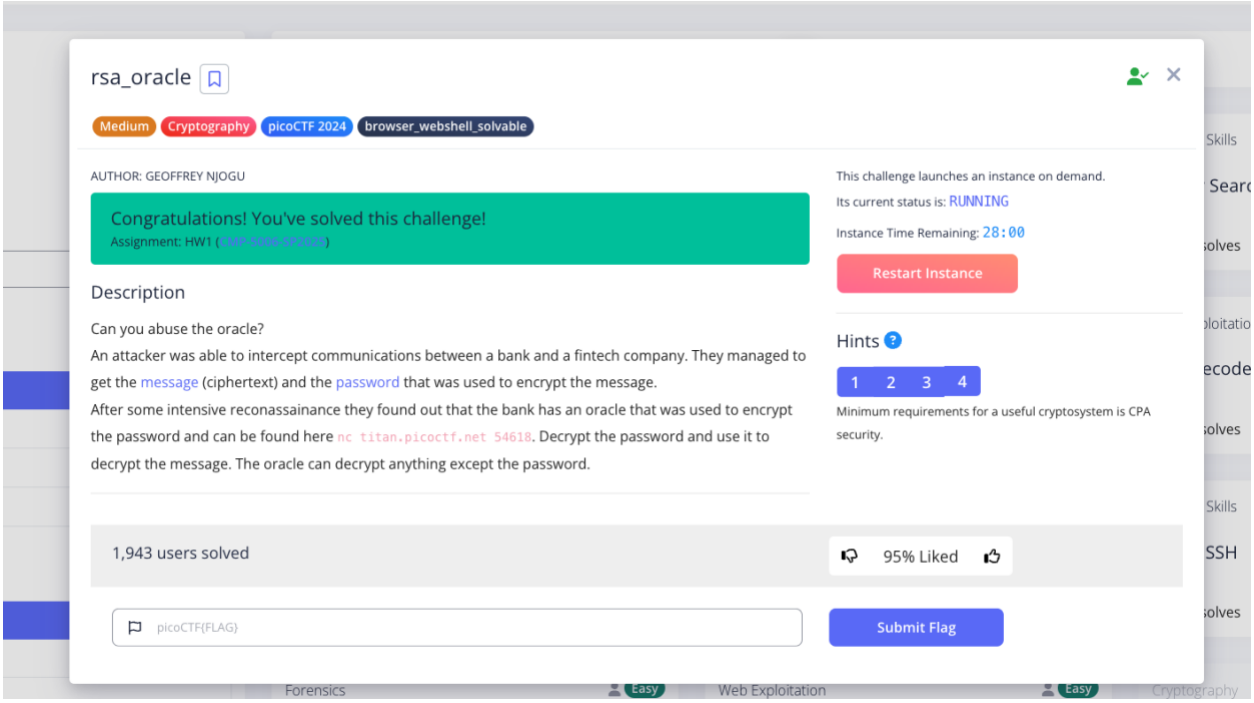
Cryptography — interencdec

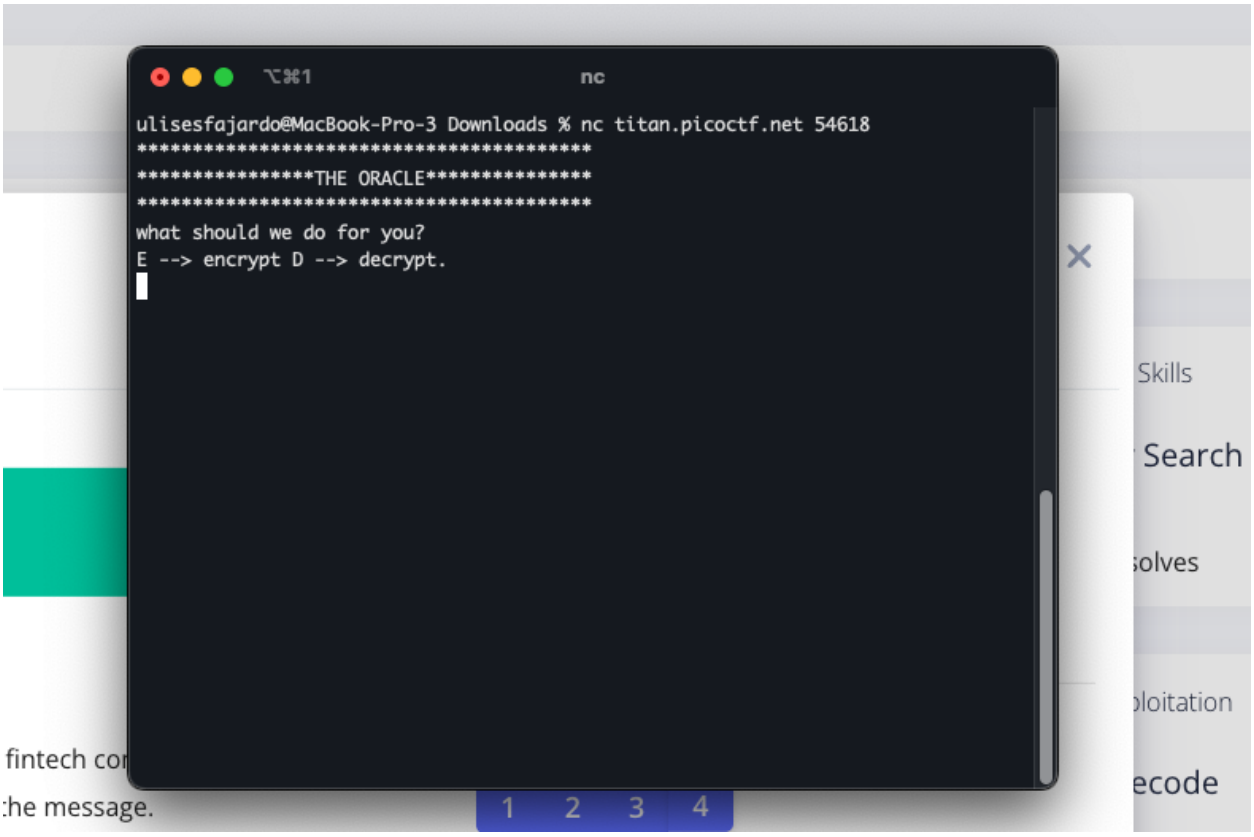
27,204 solves 95%

24,777 solves 47%

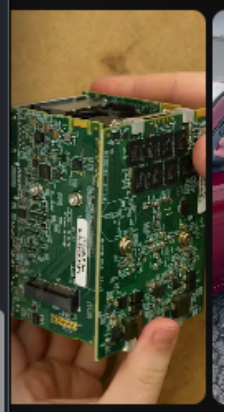
33,945 solves

« ‹ 1 2 3 4 5 6 7 › »


rsa_oracle
Medium
Cryptography
picoCTF 2024
browser_webshell_solvable
AUTHOR: GEOFFREY NJOGU
Congratulations! You've solved this challenge!
Assignment: HW1
Description
Can you abuse the oracle?
An attacker was able to intercept communications between a bank and a fintech company. They managed to get the message (ciphertext) and the password that was used to encrypt the message.
After some intensive reconassainance they found out that the bank has an oracle that was used to encrypt the password and can be found here nc titan.picoctf.net 54618. Decrypt the password and use it to decrypt the message. The oracle can decrypt anything except the password.
This challenge launches an instance on demand.
Its current status is: RUNNING
Instance Time Remaining: 28:00
Restart Instance
Hints
1 2 3 4
Minimum requirements for a useful cryptosystem is CPA security.
1,943 users solved
95% Liked
picoCTF{FLAG}
Submit Flag
Forensics
Easy
Web Exploitation
Easy
Cryptography


nc
ulisesfajardo@MacBook-Pro-3 Downloads % nc titan.picoctf.net 54618
****
**********THE ORACLE**********
****
what should we do for you?
E --> encrypt D --> decrypt.
Skills
Search
solves
ploitation
ecode
fintech co
the message.
1 2 3 4

```
what should we do for you?
E --> encrypt D --> decrypt.
d
Enter text to decrypt: 422827347115257099385775520904061114322733624519087584764
914280750184896084785197365823948557003083399978026945700009194878516437491594 24
71027917017922546%
ulisesfajardo@MacBook-Pro-3 Downloads % nc titan.picoctf.net 54618
****************************************
***************THE ORACLE***************
****************************************
what should we do for you?
E --> encrypt D --> decrypt.
d
Enter text to decrypt: 422827347115257099385775520904061114322733624519087584764
914280750184896084785197365823948557003083399978026945700009194878516437491594 24
71027917017922546
Lol, good try, can't decrypt that for you. Be creative and good luck

what should we do for you?
E --> encrypt D --> decrypt.
```



Exercise2.py | convert.py | convert2.py | Exercise5.py

Users > ulisesfajardo > Downloads > Exercise5.py > ...

```python
from pwn import *

# Connect to the remote server
conn = remote('titan.picoctf.net', 54618)

# Receive the welcome message
msg = conn.recvuntil('decrypt.')
print(msg.decode())

# Send the encryption option
conn.sendline(b'E')

# Receive the server's response
msg = conn.recvuntil('keysize):')
print(msg.decode())

# Send the number 2 for encryption
conn.sendline(b'\x02')
msg = conn.recvuntil('ciphertext (m ^ e mod n)')
msg = conn.recvline()

# Get the value of 2^e and multiply it by m^e from the password.enc fi
cipher_value = int(msg.decode()) * 422827347115257099385775520904061111

# Select the decryption option
msg = conn.recvuntil('decrypt.')
print(msg.decode())
conn.sendline(b'D')

# Send the value of 2^e * m^e for decryption
msg = conn.recvuntil('decrypt:')
print(msg.decode())
conn.sendline(str(cipher_value))

# Receive the decrypted response
msg = conn.recvuntil('hex (c ^ d mod n):')
print(msg.decode())
msg = conn.recvline()
print(msg.decode())

# Convert the response from hexadecimal to an integer and divide by 2
plaintext = int(msg, 16) // 2
print(hex(plaintext))

# Convert the result to ASCII
ascii_text = bytes.fromhex(hex(plaintext)[2:]).decode('ascii')
print(ascii_text)

# Close connection
conn.close()
```

PROBLEMAS 75 | SALIDA | TERMINAL | Python - Downloads

```
what should we do for you?
E --> encrypt D --> decrypt.
/Users/ulisesfajardo/Downloads/Exercise5.py:31: BytesWarning: Text is not bytes; assuming AS
CII, no guarantees. See https://docs.pwntools.com/#bytes
  msg = conn.recvuntil('decrypt:')

Enter text to decrypt:
/Users/ulisesfajardo/Downloads/Exercise5.py:33: BytesWarning: Text is not bytes; assuming AS
CII, no guarantees. See https://docs.pwntools.com/#bytes
  conn.sendline(str(cipher_value))
/Users/ulisesfajardo/Downloads/Exercise5.py:36: BytesWarning: Text is not bytes; assuming AS
CII, no guarantees. See https://docs.pwntools.com/#bytes
  msg = conn.recvuntil('hex (c ^ d mod n):')
 decrypted ciphertext as hex (c ^ d mod n):
 c8c2607272

0x6461303939
da099
[*] Closed connection to titan.picoctf.net port 54618
ulisesfajardo@MacBook-Pro-3 Downloads %
```

```
what should we do for you?
E --> encrypt D --> decrypt.
^C
ulisesfajardo@MacBook-Pro-3 Downloads % openssl enc -aes-256-abc -d -in secrect.
enc -k da099
enc: Unrecognized flag aes-256-abc
enc: Use -help for summary.
ulisesfajardo@MacBook-Pro-3 Downloads % openssl enc -aes-256-cbc -d in secrect.e
nc -k da099
Extra arguments given.
enc: Use -help for summary.
ulisesfajardo@MacBook-Pro-3 Downloads % openssl enc -aes-256-cbc -d -in secrect.
enc -k da099
Can't open secrect.enc for reading, No such file or directory
8595964416:error:02001002:system library:fopen:No such file or directory:crypto/
bio/bss_file.c:69:fopen('secrect.enc','rb')
8595964416:error:2006D080:BIO routines:BIO_new_file:no such file:crypto/bio/bss_
file.c:76:
ulisesfajardo@MacBook-Pro-3 Downloads % openssl enc -aes-256-cbc -d -in secret.e
nc -k da099
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
picoCTF{su((3ss_(r@ck1ng_r3@_da099d93}%
ulisesfajardo@MacBook-Pro-3 Downloads %
```