INTRODUCTION TO

CAPTURE THE FLAG {CTF}

AND BASICS OF CYBER-SECUIRITY

CYBER-SECURITY AND HACKING

UNDERSTANDING THREATS, DEFENSES, AND ETHICAL CONSIDERATIONS IN SAFEGUARDING DIGITAL SYSTEMS AGAINST MALICIOUS ATTACKS AND UNAUTHORIZED ACCESS.

WHAT IS PERSONAL INFORMATION?

A)

INFORMATION ABOUT YOUR AGE AND SCHOOL

B)

INFORMATION THAT CAN IDENTIFY YOU, LIKE YOUR NAME OR ADDRESS

C)

INFORMATION ABOUT YOUR FAMILY

D)

ALL OF THESE

WHAT IS PERSONAL INFORMATION?

D)

ALL OF THESE

CONCEPT OF HACKING

CYBERSECURITY ENTAILS PROTECTING DIGITAL SYSTEMS FROM UNAUTHORIZED ACCESS, DATA BREACHES, AND ATTACKS. HACKING INVOLVES EXPLOITING VULNERABILITIES IN SYSTEMS TO GAIN UNAUTHORIZED ACCESS OR MANIPULATE DATA FOR VARIOUS PURPOSES, INCLUDING THEFT, DISRUPTION, OR ESPIONAGE.

WHICH OF THE FOLLOWING IS A STRONG PASSWORD?

A) PASSWORD123

B) QWERTY

C) ILOVECATS

D) B$1Y%T3@K

WHICH OF THE FOLLOWING IS A STRONG PASSWORD?

D) B$1Y%T3@K

TYPES OF CYBER-THREATS

- MALWARE
- PHISHING
- RANSOMWARE
- DDOS ATTACKS

THEIR EFFECTS

- MALWARE : COMPROMISES SYSTEMS
- PHISHING :TARGETS SENSITIVE INFORMATION THROUGH FRAUDULENT EMAILS
- RANSOMWARE : ENCRYPTS DATA FOR EXTORTION
- DDOS ATTACKS : DISRUPT SERVICES

Their Effects

These threats can lead to financial loss, reputational damage, and operational disruptions for organizations.

SO...WHAT ARE CTF(S)

OR MORE SPECIFICALLY

CTF COMPETITONS

CTF (Capture The Flag) competitions are cybersecurity challenges where participants solve puzzles, crack codes, and exploit vulnerabilities in simulated environments. Teams or individuals compete to "capture flags," which are digital markers hidden within systems.

CHALLENGES FACED IN CTF

QUESTION

1. CRYPTOGRAPHY PUZZLES
2. WEB APPLICATION VULNERABILITIES
3. REVERSE ENGINEERING TASKS
4. BINARY EXPLOITATION
5. STEGANOGRAPHY
6. FORENSIC CHALLENGES

Let`s dive deeper into the reverse engineering type

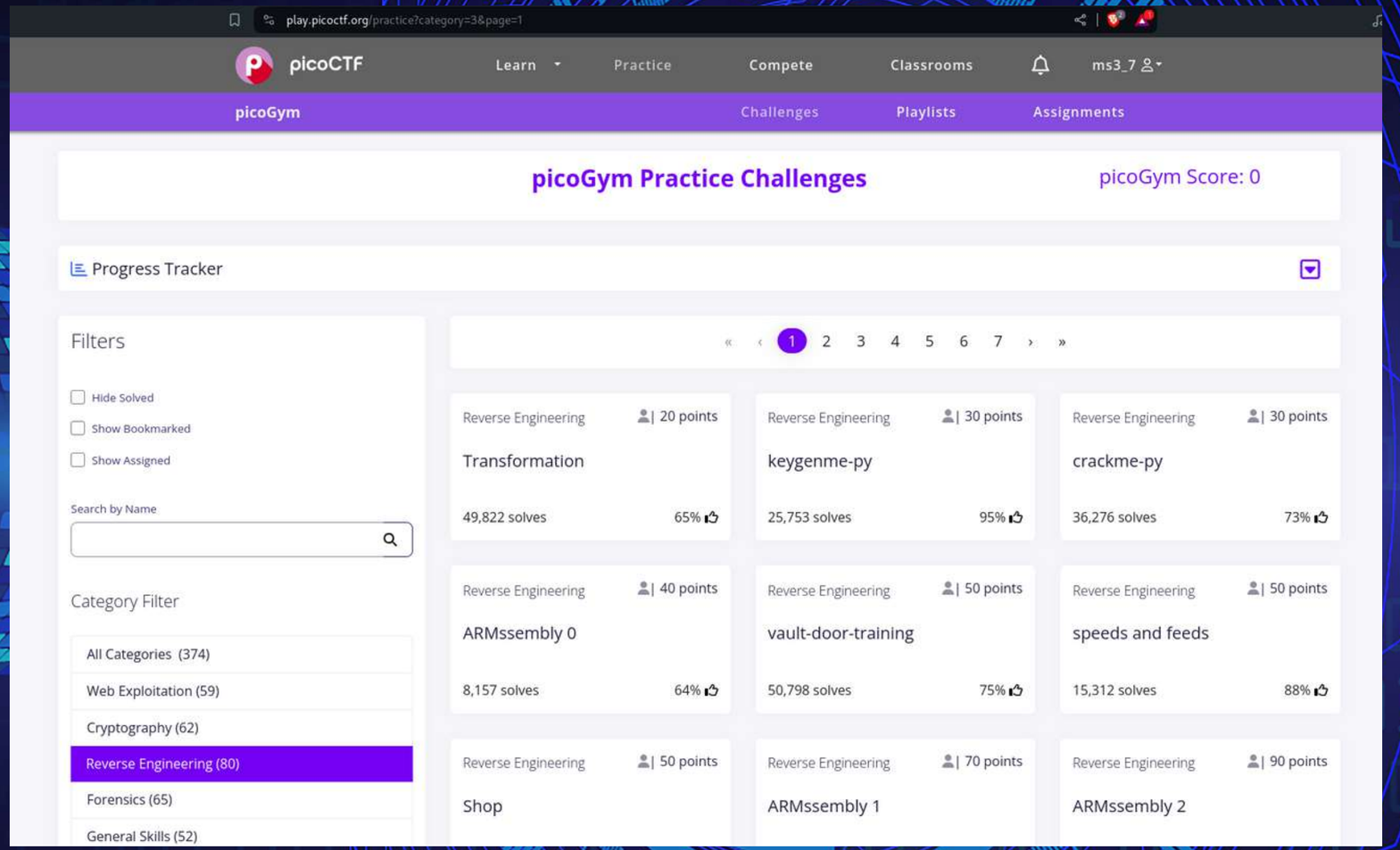
Reverse engineering challenges in CTF competitions involve dissecting software or hardware to understand its functionality, often with limited or no access to its source code or documentation. Participants typically encounter compiled binaries, firmware, or executables and are tasked with uncovering hidden functionalities, identifying vulnerabilities, or extracting sensitive information.

These challenges require a combination of analytical thinking, understanding of assembly language, debugging skills, and knowledge of common software vulnerabilities. Participants may use tools such as disassemblers, debuggers, and decompilers to analyze the binary code and understand its behavior.

COMMON TOOL AND TECHNIQUES USED IN CTF

- Disassemblers and Decompilers: Tools like IDA Pro, Ghidra, and radare2 .
-
- Debuggers: Tools like GDB (GNU Debugger) and WinDbg.
- Packet Sniffers: Tools like Wireshark and tcpdump are used to capture.
- Exploitation Frameworks: Frameworks like Metasploit and ExploitDB.
- Cryptographic Analysis Tools: Tools like OpenSSL, Cryptool, and John the Ripper.
- Web Vulnerability Scanners: Tools like OWASP ZAP, Burp Suite, and Nikto
- Forensic Tools: Tools like Autopsy, Sleuth Kit, and Volatility
- Steganography Tools: Tools like Steghide, OpenStego, and StegCracker

LETS SOLVE SOME REVERSE ENGINEERING CHALLENGES



PROBLEM 1

PATCHME.PY

```
┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ ll
total 98552
-rw-r--r-- 1 ag-031r13 ag-031r13     2207 Mar 29 15:13  1.c
-rw-r--r-- 1 ag-031r13 ag-031r13   133000 Mar 29 15:03  PDSL-Spr24-Wk10-Asg10.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13    99584 Mar 29 14:39  PDSL-Spr24-Wk5-Asg5.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13   167351 Mar 29 14:47  PDSL-Spr24-Wk6-Asg6.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13   153483 Mar 29 14:53  PDSL-Spr24-Wk7-Asg7.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13   117377 Mar 29 15:03  PDSL-Spr24-Wk8-Asg8.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13   266629 Mar 29 15:03  PDSL-Spr24-Wk9-Asg9.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13     1000 Apr  3 00:38  VaultDoorTraining.java
-rw-r--r-- 1 ag-031r13 ag-031r13 99932898 Mar 24 22:31  code_1.87.2-1709912201_amd64.deb
-rw-r--r-- 1 ag-031r13 ag-031r13     1463 Apr  3 00:06  crackme.py
-rw-r--r-- 1 ag-031r13 ag-031r13       36 Apr  2 18:07  flag.txt.enc
-rw-r--r-- 1 ag-031r13 ag-031r13      980 Apr  2 18:04  patchme.flag.py
-rw-r--r-- 1 ag-031r13 ag-031r13      489 Apr  1 20:19 'source code'
-rw-r--r-- 1 ag-031r13 ag-031r13      549 Apr  2 17:48  unpackme.flag.py

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ python patchme.flag.py
Please enter correct password for flag: 23e3e
That password is incorrect
```

Applications Places Apr 3 00

ag-031r13@kali: ~


```
┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ ll
total 98552
-rw-r--r-- 1 ag-031r13 ag-031r13     2207 Mar 29 15:13  1.c
-rw-r--r-- 1 ag-031r13 ag-031r13   133000 Mar 29 15:03  PDSL-Spr24-Wk10-Asg10.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13    99584 Mar 29 14:39  PDSL-Spr24-Wk5-Asg5.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13   167351 Mar 29 14:47  PDSL-Spr24-Wk6-Asg6.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13   153483 Mar 29 14:53  PDSL-Spr24-Wk7-Asg7.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13   117377 Mar 29 15:03  PDSL-Spr24-Wk8-Asg8.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13   266629 Mar 29 15:03  PDSL-Spr24-Wk9-Asg9.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13     1000 Apr  3 00:38  VaultDoorTraining.java
-rw-r--r-- 1 ag-031r13 ag-031r13 99932898 Mar 24 22:31  code_1.87.2-1709912201_amd64.deb
-rw-r--r-- 1 ag-031r13 ag-031r13     1463 Apr  3 00:06  crackme.py
-rw-r--r-- 1 ag-031r13 ag-031r13       36 Apr  2 18:07  flag.txt.enc
-rw-r--r-- 1 ag-031r13 ag-031r13      980 Apr  2 18:04  patchme.flag.py
-rw-r--r-- 1 ag-031r13 ag-031r13      489 Apr  1 20:19 'source code'
-rw-r--r-- 1 ag-031r13 ag-031r13      549 Apr  2 17:48  unpackme.flag.py

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ python patchme.flag.py
Please enter correct password for flag: 23e3e
That password is incorrect
```


```
┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ cat patchme.flag.py
### THIS FUNCTION WILL NOT HELP YOU FIND THE FLAG --LT ########################
def str_xor(secret, key):
    #extend key to secret length
    new_key = key
    i = 0
    while len(new_key) < len(secret):
        new_key = new_key + key[i]
        i = (i + 1) % len(key)
    return "".join([chr(ord(secret_c) ^ ord(new_key_c)) for (secret_c,new_key_c) in zip(secret,new_key)])
###############################################################################


flag_enc = open('flag.txt.enc', 'rb').read()



def level_1_pw_check():
    user_pw = input("Please enter correct password for flag: ")
    if( user_pw == "ak98" + \
                   "-=90" + \
                   "adfjhgj321" + \
                   "sleuth9000"):
        print("Welcome back... your flag, user:")
        decryption = str_xor(flag_enc.decode(), "utilitarian")
        print(decryption)
        return
    print("That password is incorrect")
```

```
-rw-r--r-- 1 ag-031r13 ag-031r13    349 Apr  2 17:48 unpackme.flag.py

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ python patchme.flag.py
Please enter correct password for flag: 23e3e
That password is incorrect
```

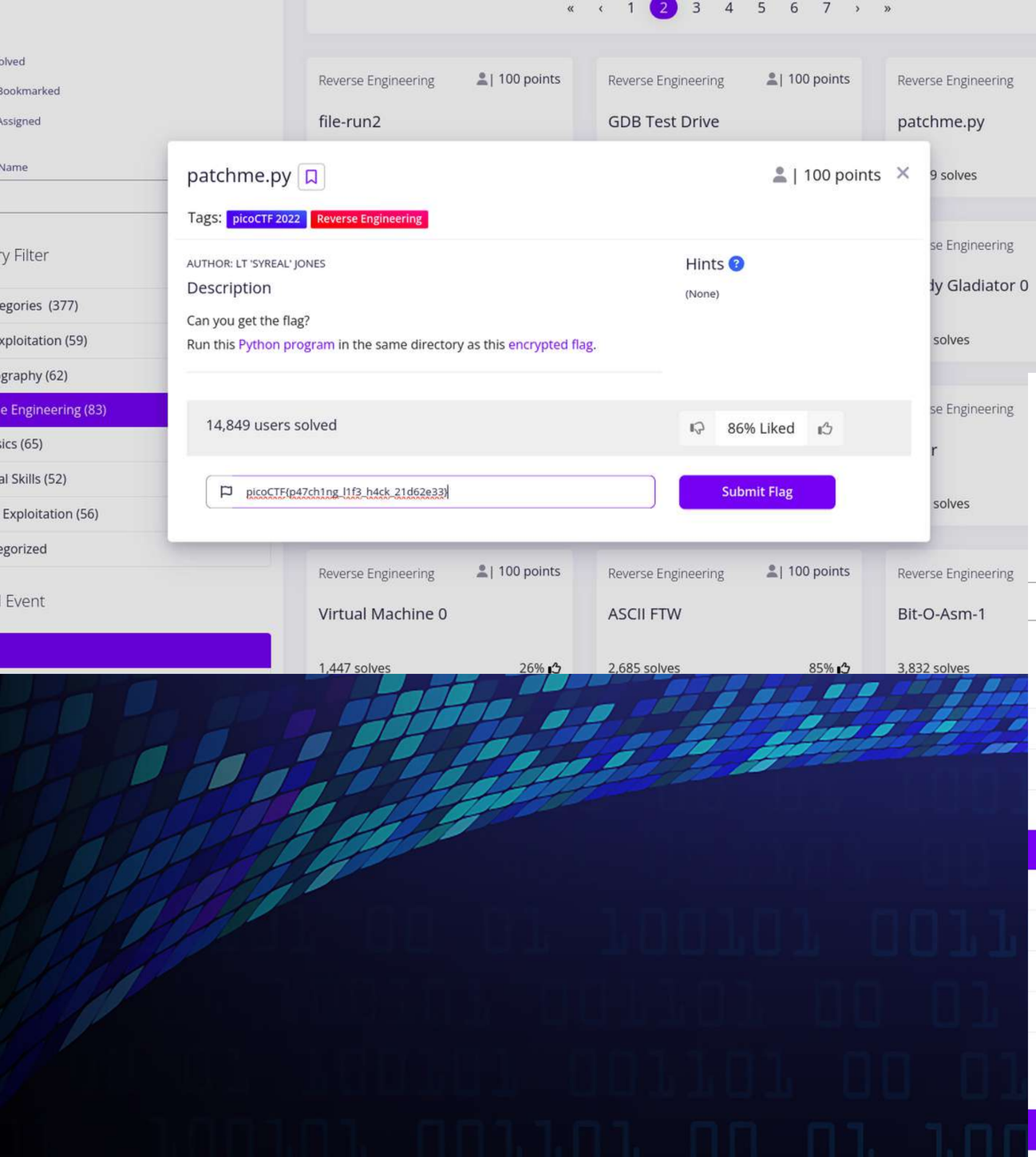
```
┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ cat patchme.flag.py
### THIS FUNCTION WILL NOT HELP YOU FIND THE FLAG --LT ########################
def str_xor(secret, key):
    #extend key to secret length
    new_key = key
    i = 0
    while len(new_key) < len(secret):
        new_key = new_key + key[i]
        i = (i + 1) % len(key)
    return "".join([chr(ord(secret_c) ^ ord(new_key_c)) for (secret_c,new_key_c) in zip(secret,new_key)])
###############################################################################


flag_enc = open('flag.txt.enc', 'rb').read()



def level_1_pw_check():
    user_pw = input("Please enter correct password for flag: ")
    if( user_pw == "ak98" + \
                   "-=90" + \
                   "adfjhgj321" + \
                   "sleuth9000"):
        print("Welcome back... your flag, user:")
        decryption = str_xor(flag_enc.decode(), "utilitarian")
        print(decryption)
        return
    print("That password is incorrect")



level_1_pw_check()
```

Screenshot captured
You can paste the image from the clipboard.

```
-rw-r--r-- 1 ag-031r13 ag-031r13     1000 Apr  3 00:38  VaultDoorTraining.java
-rw-r--r-- 1 ag-031r13 ag-031r13 99932898 Mar 24 22:31  code_1.87.2-1709912201_amd64.deb
-rw-r--r-- 1 ag-031r13 ag-031r13     1463 Apr  3 00:06  crackme.py
-rw-r--r-- 1 ag-031r13 ag-031r13       36 Apr  2 18:07  flag.txt.enc
-rw-r--r-- 1 ag-031r13 ag-031r13      980 Apr  2 18:04  patchme.flag.py
-rw-r--r-- 1 ag-031r13 ag-031r13      489 Apr  1 20:19 'source code'
-rw-r--r-- 1 ag-031r13 ag-031r13      549 Apr  2 17:48  unpackme.flag.py

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ python patchme.flag.py
Please enter correct password for flag: 23e3e
That password is incorrect

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ cat patchme.flag.py
### THIS FUNCTION WILL NOT HELP YOU FIND THE FLAG --LT ########################
def str_xor(secret, key):
    #extend key to secret length
    new_key = key
    i = 0
    while len(new_key) < len(secret):
        new_key = new_key + key[i]
        i = (i + 1) % len(key)
    return "".join([chr(ord(secret_c) ^ ord(new_key_c)) for (secret_c,new_key_c) in zip(secret,new_key)])
###############################################################################


flag_enc = open('flag.txt.enc', 'rb').read()


def level_1_pw_check():
    user_pw = input("Please enter correct password for flag: ")
    if( user_pw == "ak98" + \
                   "-=90" + \
                   "adfjhgj321" + \
                   "sleuth9000"):
        print("Welcome back... your flag, user:")
        decryption = str_xor(flag_enc.decode(), "utilitarian")
        print(decryption)
        return
    print("That password is incorrect")


level_1_pw_check()

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ python patchme.flag.py
Please enter correct password for flag: ak98-=90adfjhgj321sleuth9000
Welcome back... your flag, user:
picoCTF{p47ch1ng_l1f3_h4ck_21d62e33}

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$
```

level_1_pw_check()

```
┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ python patchme.flag.py
Please enter correct password for flag: ak98-=90adfjhgj321sleuth9000
Welcome back... your flag, user:
picoCTF{p47ch1ng_l1f3_h4ck_21d62e33}

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$
```



« ‹ 1 2 3 4 5 6 7 › »

Hurray! You earned 100 points.

Reverse Engineering | 100 points
file-run2
17,849 solves 78%

Reverse Engineering | 100 points
GDB Test Drive
12,668 solves 87%

Reverse Engineering
patchme.py
14,850 solves

Reverse Engineering | 100 points
Safe Opener
14,842 solves 93%

Reverse Engineering | 100 points
unpackme.py
11,632 solves 89%

Reverse Engineering
Ready Gladiator 0
5,310 solves

Reverse Engineering | 100 points
Reverse
10,539 solves 81%

Reverse Engineering | 100 points
Safe Opener 2
9,346 solves 70%

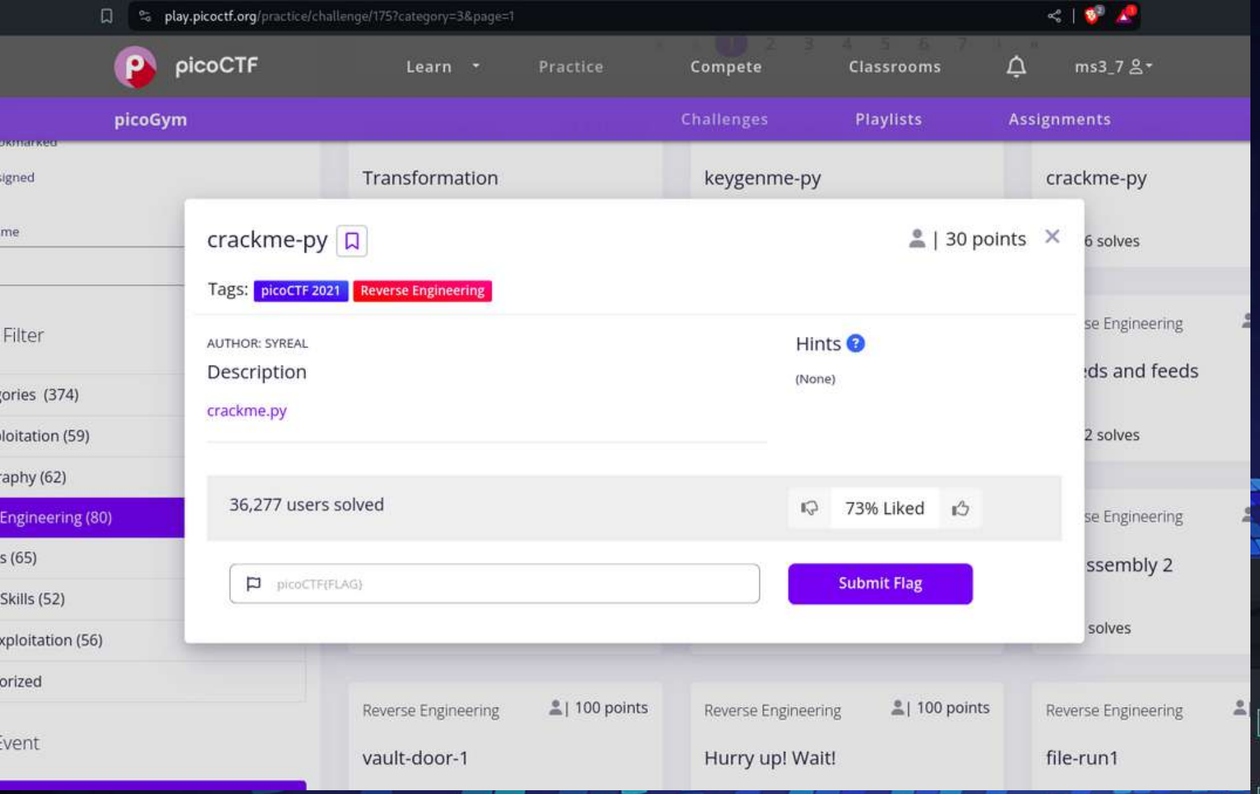
Reverse Engineering
timer
5,901 solves

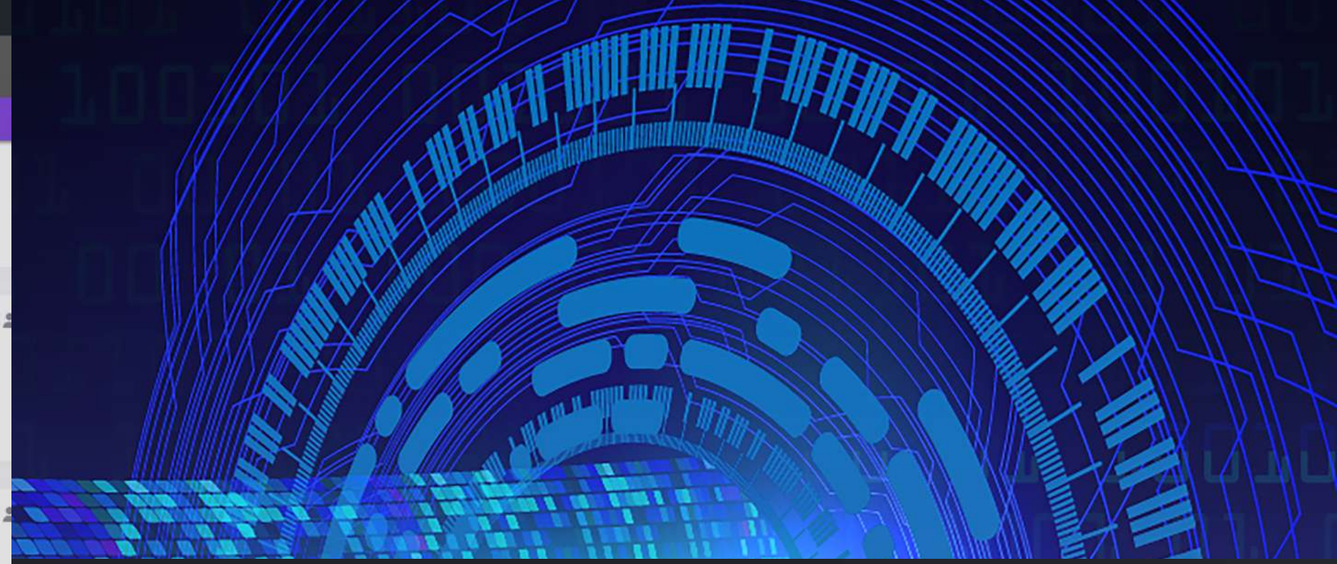
Reverse Engineering | 100 points
Virtual Machine 0
1,447 solves 26%

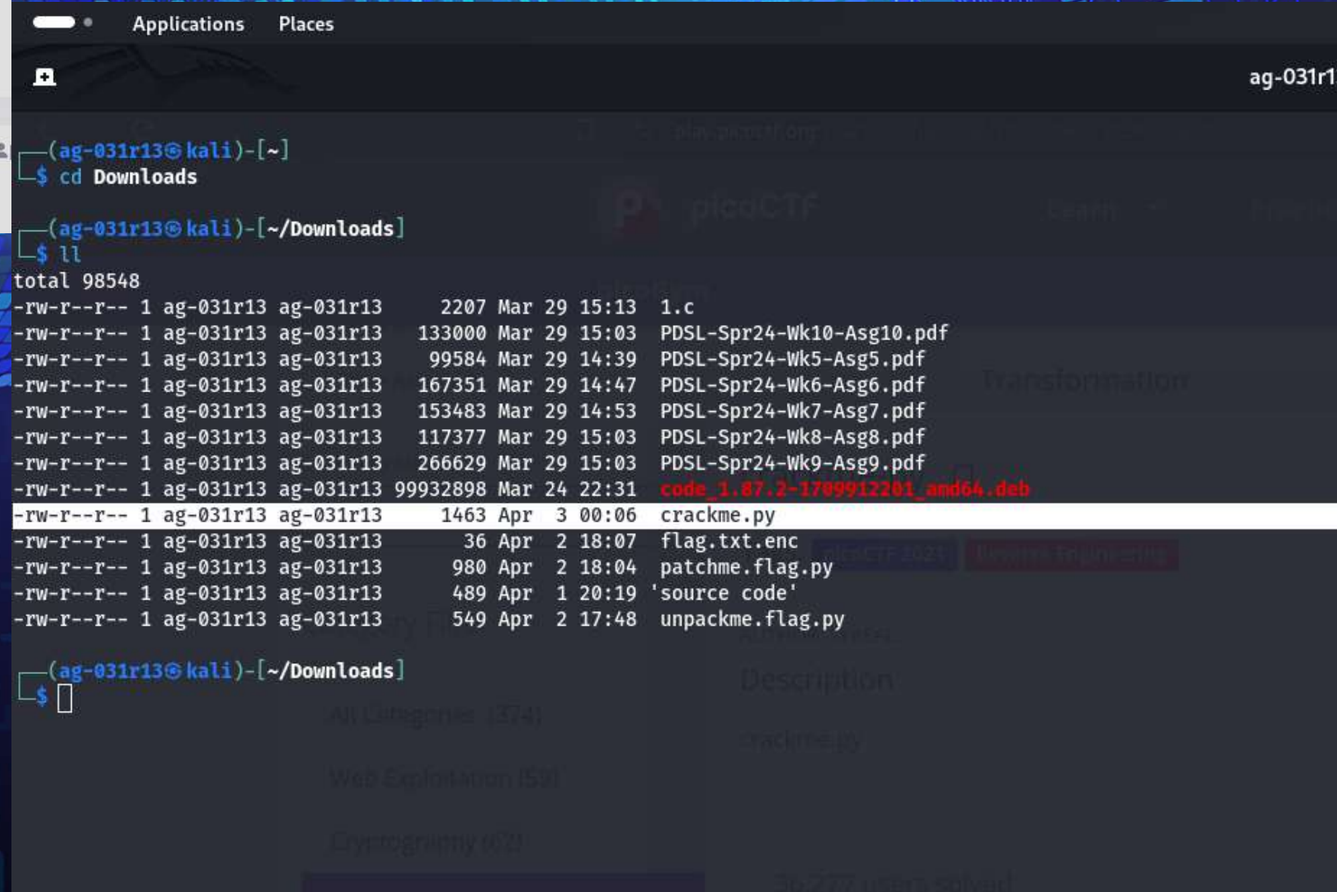
Reverse Engineering | 100 points
ASCII FTW
2,685 solves 85%

Reverse Engineering
Bit-O-Asm-1
3,832 solves

PROBLEM 2
CRACKME.PY


play.picoctf.org/practice/challenge/175?category=3&page=1
picoCTF
Learn
Practice
Compete
Classrooms
ms3_7
picoGym
Challenges
Playlists
Assignments
Transformation
keygenme-py
crackme-py
crackme-py
30 points
Tags:
picoCTF 2021
Reverse Engineering
AUTHOR: SYREAL
Description
crackme.py
Hints
(None)
36,277 users solved
73% Liked
picoCTF{FLAG}
Submit Flag
Filter
Reverse Engineering
100 points
vault-door-1
Reverse Engineering
100 points
Hurry up! Wait!
Reverse Engineering
file-run1




Applications
Places
ag-031r1
(ag-031r13㉿kali)-[~]
$ cd Downloads
(ag-031r13㉿kali)-[~/Downloads]
$ ll
total 98548
-rw-r--r-- 1 ag-031r13 ag-031r13 2207 Mar 29 15:13 1.c
-rw-r--r-- 1 ag-031r13 ag-031r13 133000 Mar 29 15:03 PDSL-Spr24-Wk10-Asg10.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13 99584 Mar 29 14:39 PDSL-Spr24-Wk5-Asg5.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13 167351 Mar 29 14:47 PDSL-Spr24-Wk6-Asg6.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13 153483 Mar 29 14:53 PDSL-Spr24-Wk7-Asg7.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13 117377 Mar 29 15:03 PDSL-Spr24-Wk8-Asg8.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13 266629 Mar 29 15:03 PDSL-Spr24-Wk9-Asg9.pdf
-rw-r--r-- 1 ag-031r13 ag-031r13 99932898 Mar 24 22:31 code_1.87.2-1709912201_amd64.deb
-rw-r--r-- 1 ag-031r13 ag-031r13 1463 Apr 3 00:06 crackme.py
-rw-r--r-- 1 ag-031r13 ag-031r13 36 Apr 2 18:07 flag.txt.enc
-rw-r--r-- 1 ag-031r13 ag-031r13 980 Apr 2 18:04 patchme.flag.py
-rw-r--r-- 1 ag-031r13 ag-031r13 489 Apr 1 20:19 'source code'
-rw-r--r-- 1 ag-031r13 ag-031r13 549 Apr 2 17:48 unpackme.flag.py
(ag-031r13㉿kali)-[~/Downloads]
$

```
┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ cat crackme.py
# Hiding this really important number in an obscure piece of code is brilliant!
# AND it's encrypted!
# We want our biggest client to know his information is safe with us.
bezos_cc_secret = "A:4@r%uL`M-^M0c0AbcM-MFE055a4ce`eN"

# Reference alphabet
alphabet = "!\"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ"+ \
           "[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~"



def decode_secret(secret):
    """ROT47 decode

    NOTE: encode and decode are the same operation in the ROT cipher family.
    """

    # Encryption key
    rotate_const = 47

    # Storage for decoded secret
    decoded = ""

    # decode loop
    for c in secret:
        index = alphabet.find(c)
        original_index = (index + rotate_const) % len(alphabet)
        decoded = decoded + alphabet[original_index]

    print(decoded)



def choose_greatest():
    """Echo the largest of the two numbers given by the user to the program

    Warning: this function was written quickly and needs proper error handling
    """

    user_value_1 = input("What's your first number? ")
    user_value_2 = input("What's your second number? ")
    greatest_value = user_value_1 # need a value to return if 1 & 2 are equal

    if user_value_1 > user_value_2:
        greatest_value = user_value_1
    elif user_value_1 < user_value_2:
        greatest_value = user_value_2

    print( "The number with largest positive magnitude is "
        + str(greatest_value) )
```

```
└─$ nano crackme.py

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ python crackme.py
  File "/home/ag-031r13/Downloads/crackme.py", line 20
    print (rotate_const)
TabError: inconsistent use of tabs and spaces in indentation

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ nano crackme.py

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ python crackme.py
What's your first number? 4567890
What's your second number? A:4@r%uL`M-^M0c0AbcM-MFE055a4ce`eN
The number with largest positive magnitude is A:4@r%uL`M-^M0c0AbcM-MFE055a4ce`eN

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ cat crackme.py
# Hiding this really important number in an obscure piece of code is brilliant!
# AND it's encrypted!
# We want our biggest client to know his information is safe with us.
bezos_cc_secret = "A:4@r%uL`M-^M0c0AbcM-MFE055a4ce`eN"

# Reference alphabet
alphabet = "!\"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ"+ \
           "[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~"



def decode_secret(secret):
    """ROT47 decode

    NOTE: encode and decode are the same operation in the ROT cipher family.
    """

    # Encryption key
    rotate_const = 47

    # Storage for decoded secret
    decoded = ""

    # decode loop
    for c in secret:
        index = alphabet.find(c)
        original_index = (index + rotate_const) % len(alphabet)
        decoded = decoded + alphabet[original_index]

    print (decoded)
```

Applications Places Apr 3 02:33

Screenshot captured
You can paste the image from the clipboard

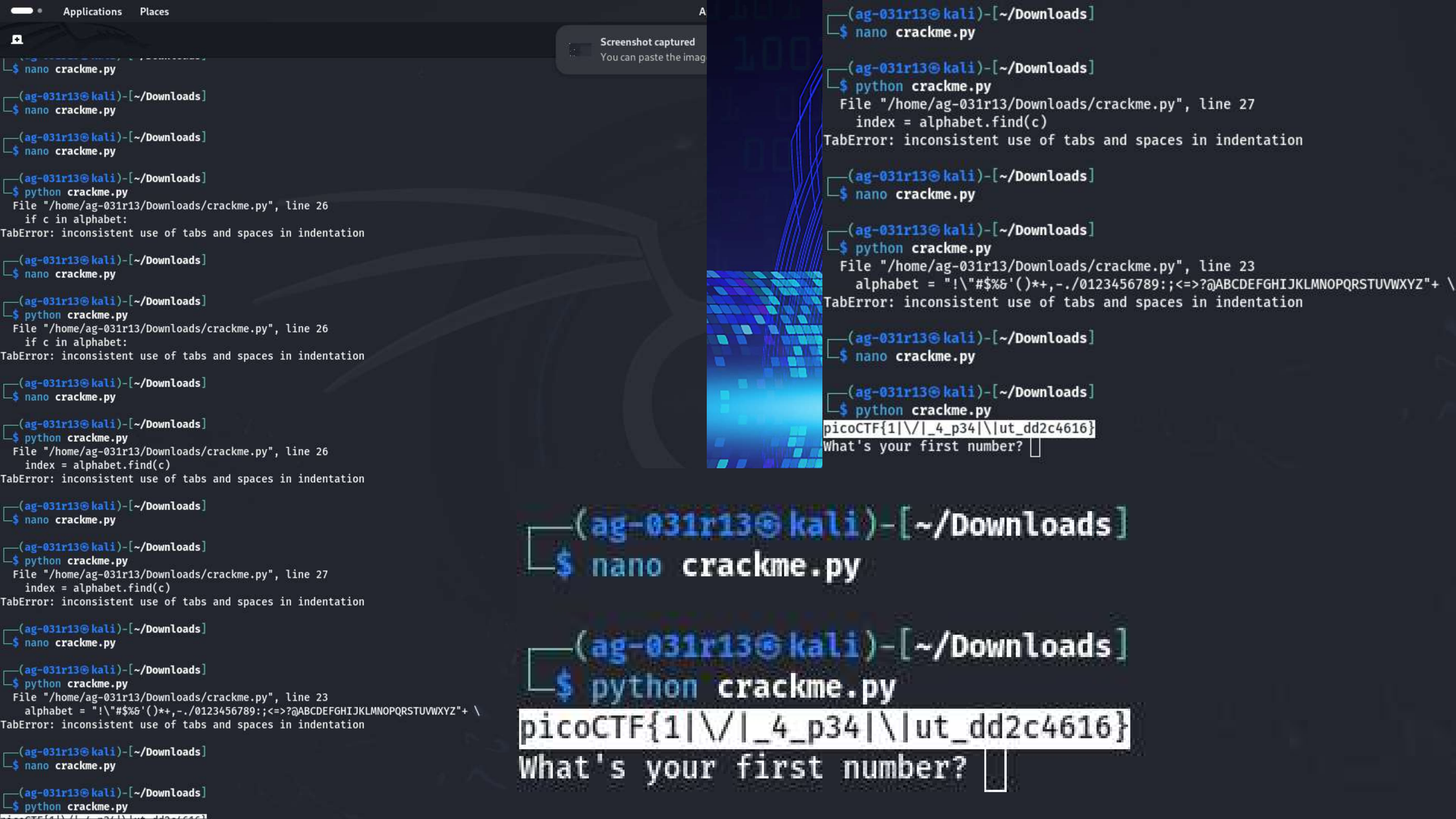
```
def choose_greatest():
    """Echo the largest of the two numbers given by the user to the program

    Warning: this function was written quickly and needs proper error handling
    """

    user_value_1 = input("What's your first number? ")
    user_value_2 = input("What's your second number? ")
    greatest_value = user_value_1 # need a value to return if 1 & 2 are equal

    if user_value_1 > user_value_2:
        greatest_value = user_value_1
    elif user_value_1 < user_value_2:
        greatest_value = user_value_2

    print( "The number with largest positive magnitude is "
        + str(greatest_value) )



choose_greatest()

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ python crackme.py
What's your first number? 90
What's your second number? 90
The number with largest positive magnitude is 90

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ nano crackme.py

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ python crackme.py
  File "/home/ag-031r13/Downloads/crackme.py", line 20
    print (rotate_const)
TabError: inconsistent use of tabs and spaces in indentation

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ nano crackme.py

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ python crackme.py
What's your first number? 4567890
What's your second number? A:4@r%uL`M-^M0c0AbcM-MFE055a4ce`eN
The number with largest positive magnitude is A:4@r%uL`M-^M0c0AbcM-MFE055a4ce`eN

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ cat crackme.py
# Hiding this really important number in an obscure piece of code is brilliant!
# AND it's encrypted!
# We want our biggest client to know his information is safe with us.
bezos_cc_secret = "A:4@r%uL`M-^M0c0AbcM-MFE055a4ce`eN"
```

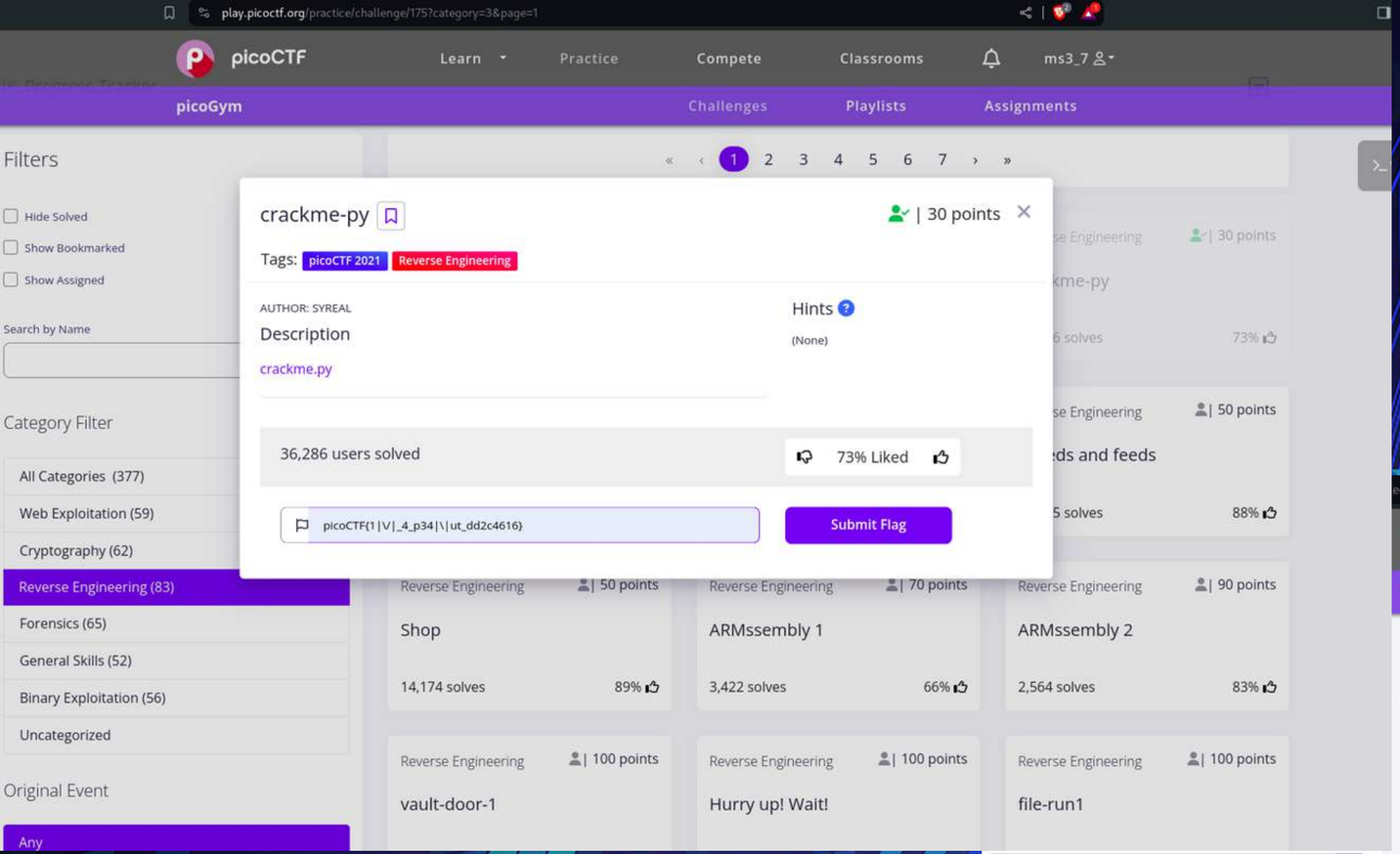
```
    NOTE: encode and decode are the same operation in the ROT cipher family.
    """

    # Encryption key
    rotate_const = 47

    # Storage for decoded secret
    decoded = ""

    # decode loop
    for c in secret:
        index = alphabet.find(c)
        original_index = (index + rotate_const) % len(alphabet)
        decoded = decoded + alphabet[original_index]

    print (decoded)



def choose_greatest():
    """Echo the largest of the two numbers given by the user to the program

    Warning: this function was written quickly and needs proper error handling
    """

    user_value_1 = input("What's your first number? ")
    user_value_2 = input("What's your second number? ")
    greatest_value = user_value_1 # need a value to return if 1 & 2 are equal

    if user_value_1 > user_value_2:
        greatest_value = user_value_1
    elif user_value_1 < user_value_2:
        greatest_value = user_value_2

    print( "The number with largest positive magnitude is "
        + str(greatest_value) )



choose_greatest()
```

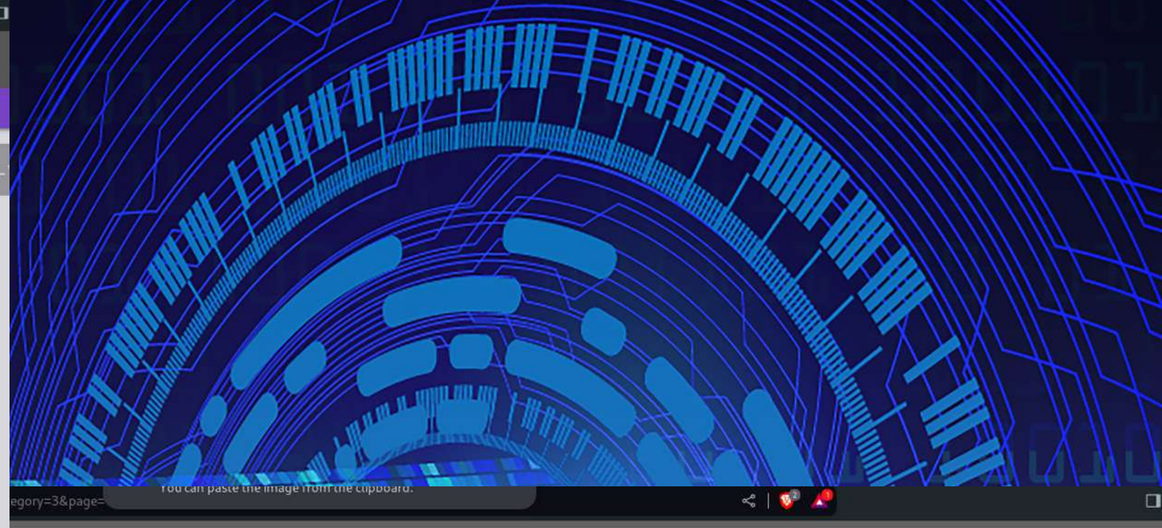
```
┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ nano crackme.py

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ nano crackme.py

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ nano crackme.py

┌──(ag-031r13㉿kali)-[~/Downloads]
└─$ python crackme.py
  File "/home/ag-031r13/Downloads/crackme.py", line 26
    if c in alphabet:
TabError: inconsistent use of tabs and spaces in indentation
```

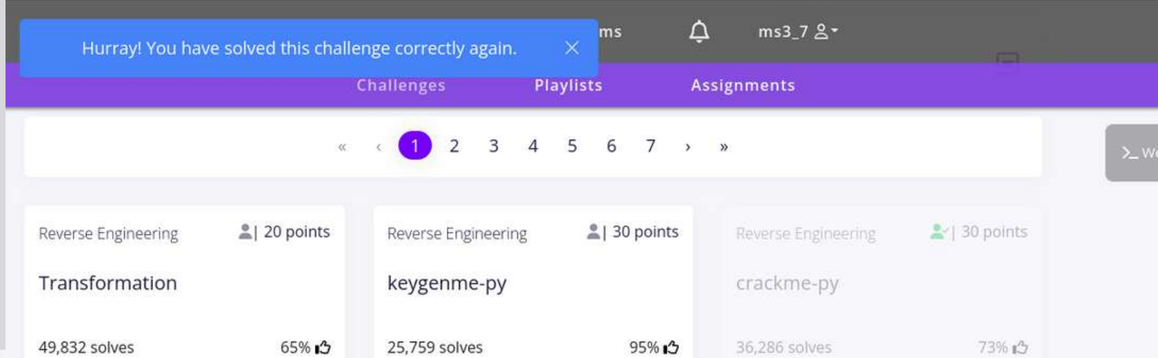
Applications Places Apr 3
ag-031r13@kal

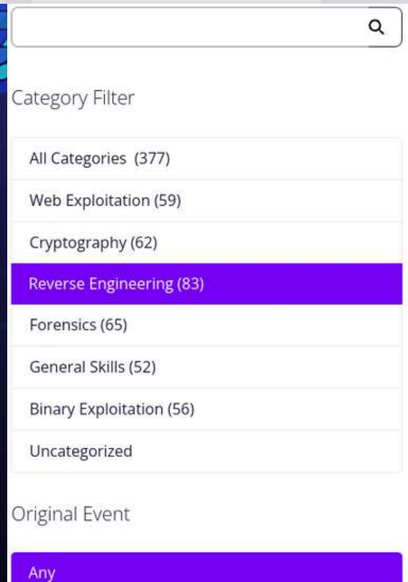
```
 GNU nano 7.2                                                    crac
# Hiding this really important number in an obscure piece of code is brilliant!
# AND it's encrypted!
# We want our biggest client to know his information is safe with us.
bezos_cc_secret = "A:4@r%uL`M-^M0c0AbcM-MFE055a4ce`eN"

# Reference alphabet
alphabet = "!\"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ"+ \
           "[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~"



def decode_secret(secret):
    """ROT47 decode

    NOTE: encode and decode are the same operation in the ROT cipher family.
    """

    # Encryption key
    rotate_const = 47

    # Storage for decoded secret
    decoded = ""

    # decode loop
    for c in secret:
        index = alphabet.find(c)
        original_index = (index + rotate_const) % len(alphabet)
        decoded = decoded + alphabet[original_index]

    print (decoded)



def choose_greatest():
    """Echo the largest of the two numbers given by the user to the program

    Warning: this function was written quickly and needs proper error handling
    """

    user_value_1 = input("What's your first number? ")
    user_value_2 = input("What's your second number? ")
    greatest_value = user_value_1 # need a value to return if 1 & 2 are equal

    if user_value_1 > user_value_2:
        greatest_value = user_value_1
    elif user_value_1 < user_value_2:
        greatest_value = user_value_2

    print( "The number with largest positive magnitude is "
        + str(greatest_value) )

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo
```


Applications
Places
Screenshot captured
You can paste the imag
(ag-031r13㉿kali)-[~/Downloads]
$ nano crackme.py
(ag-031r13㉿kali)-[~/Downloads]
$ nano crackme.py
(ag-031r13㉿kali)-[~/Downloads]
$ nano crackme.py
(ag-031r13㉿kali)-[~/Downloads]
$ python crackme.py
File "/home/ag-031r13/Downloads/crackme.py", line 26
if c in alphabet:
TabError: inconsistent use of tabs and spaces in indentation
(ag-031r13㉿kali)-[~/Downloads]
$ nano crackme.py
(ag-031r13㉿kali)-[~/Downloads]
$ python crackme.py
File "/home/ag-031r13/Downloads/crackme.py", line 26
if c in alphabet:
TabError: inconsistent use of tabs and spaces in indentation
(ag-031r13㉿kali)-[~/Downloads]
$ nano crackme.py
(ag-031r13㉿kali)-[~/Downloads]
$ python crackme.py
File "/home/ag-031r13/Downloads/crackme.py", line 26
index = alphabet.find(c)
TabError: inconsistent use of tabs and spaces in indentation
(ag-031r13㉿kali)-[~/Downloads]
$ nano crackme.py
(ag-031r13㉿kali)-[~/Downloads]
$ python crackme.py
File "/home/ag-031r13/Downloads/crackme.py", line 27
index = alphabet.find(c)
TabError: inconsistent use of tabs and spaces in indentation
(ag-031r13㉿kali)-[~/Downloads]
$ nano crackme.py
(ag-031r13㉿kali)-[~/Downloads]
$ python crackme.py
File "/home/ag-031r13/Downloads/crackme.py", line 23
alphabet = "!\"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ"+ \
TabError: inconsistent use of tabs and spaces in indentation
(ag-031r13㉿kali)-[~/Downloads]
$ nano crackme.py
(ag-031r13㉿kali)-[~/Downloads]
$ python crackme.py
(ag-031r13㉿kali)-[~/Downloads]
$ nano crackme.py
(ag-031r13㉿kali)-[~/Downloads]
$ python crackme.py
File "/home/ag-031r13/Downloads/crackme.py", line 27
index = alphabet.find(c)
TabError: inconsistent use of tabs and spaces in indentation
(ag-031r13㉿kali)-[~/Downloads]
$ nano crackme.py
(ag-031r13㉿kali)-[~/Downloads]
$ python crackme.py
File "/home/ag-031r13/Downloads/crackme.py", line 23
alphabet = "!\"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ"+ \
TabError: inconsistent use of tabs and spaces in indentation
(ag-031r13㉿kali)-[~/Downloads]
$ nano crackme.py
(ag-031r13㉿kali)-[~/Downloads]
$ python crackme.py
picoCTF{1|\/|_4_p34|\|ut_dd2c4616}
What's your first number?
(ag-031r13㉿kali)-[~/Downloads]
$ nano crackme.py
(ag-031r13㉿kali)-[~/Downloads]
$ python crackme.py
picoCTF{1|\/|_4_p34|\|ut_dd2c4616}
What's your first number?


play.picoctf.org/practice/challenge/175?category=3&page=1
picoCTF
Learn
Practice
Compete
Classrooms
ms3_7
picoGym
Challenges
Playlists
Assignments
Filters
Hide Solved
Show Bookmarked
Show Assigned
Search by Name
Category Filter
All Categories (377)
Web Exploitation (59)
Cryptography (62)
Reverse Engineering (83)
Forensics (65)
General Skills (52)
Binary Exploitation (56)
Uncategorized
Original Event
Any
« ‹ 1 2 3 4 5 6 7 › »
crackme-py
| 30 points
Tags: picoCTF 2021 Reverse Engineering
AUTHOR: SYREAL
Description
crackme.py
Hints
(None)
36,286 users solved
73% Liked
picoCTF{1|V|_4_p34|\|ut_dd2c4616}
Submit Flag
Reverse Engineering | 50 points
Shop
14,174 solves 89%
Reverse Engineering | 70 points
ARMssembly 1
3,422 solves 66%
Reverse Engineering | 90 points
ARMssembly 2
2,564 solves 83%
Reverse Engineering | 100 points
vault-door-1
Reverse Engineering | 100 points
Hurry up! Wait!
Reverse Engineering | 100 points
file-run1


You can paste the image from the clipboard.


Hurray! You have solved this challenge correctly again.
ms3_7
Challenges
Playlists
Assignments
« ‹ 1 2 3 4 5 6 7 › »
Reverse Engineering | 20 points
Transformation
49,832 solves 65%
Reverse Engineering | 30 points
keygenme-py
25,759 solves 95%
Reverse Engineering | 30 points
crackme-py
36,286 solves 73%
Reverse Engineering | 40 points
ARMssembly 0
8,163 solves 64%
Reverse Engineering | 50 points
vault-door-training
50,805 solves 75%
Reverse Engineering | 50 points
speeds and feeds
15,315 solves 88%
Reverse Engineering | 50 points
Shop
14,174 solves 89%
Reverse Engineering | 70 points
ARMssembly 1
3,422 solves 66%
Reverse Engineering | 90 points
ARMssembly 2
2,564 solves 83%
Reverse Engineering | 100 points
vault-door-1
Reverse Engineering | 100 points
Hurry up! Wait!
Reverse Engineering | 100 points
file-run1


Category Filter
All Categories (377)
Web Exploitation (59)
Cryptography (62)
Reverse Engineering (83)
Forensics (65)
General Skills (52)
Binary Exploitation (56)
Uncategorized
Original Event
Any

GNU nano 7.2

```
# Reference alphabet
alphabet = "!\"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ"+ \
           "[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~"



def decode_secret(secret):
    """ROT47 decode

    NOTE: encode and decode are the same operation in the ROT cipher family.
    """
    # Encryption key
    rotate_const = 47

    # Storage for decoded secret
    decoded = ""

    # decode loop
    for c in secret:
        index = alphabet.find(c)
        original_index = (index + rotate_const) % len(alphabet)
        decoded = decoded + alphabet[original_index]

    print (decoded)



def choose_greatest():
    """Echo the largest of the two numbers given by the user to the program

    Warning: this function was written quickly and needs proper error handling
    """

    user_value_1 = input("What's your first number? ")
    user_value_2 = input("What's your second number? ")
    greatest_value = user_value_1 # need a value to return if 1 & 2 are equal

    if user_value_1 > user_value_2:
        greatest_value = user_value_1
    elif user_value_1 < user_value_2:
        greatest_value = user_value_2

    print( "The number with largest positive magnitude is "
        + str(greatest_value) )


choose_greatest()
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-^G Help M-^O Write Out M-^W Where Is M-^K Cut M-^T Execute M-^C Location M-U Undo M-A
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-^X Exit M-^R Read File M-^\ Replace M-^U Paste M-^J Justify M-^/ Go To Line M-E Redo M-6

GNU nano 7.2 crackme.py

```
#Ming this really important number in an obscure piece of code is brilliant!
# AND it's encrypted!
# We want our biggest client to know his information is safe with us.
bezos_cc_secret = "A:4@r%uL`M-^M0c0AbcM-MFE055a4ce`eN"

# Reference alphabet
alphabet = "!\"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ"+ \
           "[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~"


def decode_secret(secret):
    rotate_const = 47
    decoded = ""
    for c in secret:
        if c in alphabet:
            index = alphabet.find(c)
            original_index = (index + rotate_const) % len(alphabet)
            decoded += alphabet[original_index]
        else:
            decoded += c
    return decoded

bezos_cc_secret = "A:4@r%uL`M-^M0c0AbcM-MFE055a4ce`eN"
print(decode_secret(bezos_cc_secret))


def choose_greatest():
    """Echo the largest of the two numbers given by the user to the program

    Warning: this function was written quickly and needs proper error handling
    """

    user_value_1 = input("What's your first number? ")
    user_value_2 = input("What's your second number? ")
    greatest_value = user_value_1 # need a value to return if 1 & 2 are equal

    if user_value_1 > user_value_2:
        greatest_value = user_value_1
    elif user_value_1 < user_value_2:
        greatest_value = user_value_2

    print( "The number with largest positive magnitude is "
        + str(greatest_value) )


choose_greatest()
```

[Read 47 lines]

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-^G Help M-^O Write Out M-^W Where Is M-^K Cut M-^T Execute M-^C Location M-U Undo M-A
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-^X Exit M-^R Read File M-^\ Replace M-^U Paste M-^J Justify M-^/ Go To Line M-E Redo M-6

THANK YOU