

## PicoCTF Challenges

### General Skills

#### 1. Super SSH

- You just need to ssh to the sever using the following command:

```
ssh ctf-player@titan.picoctf.net -p 55058
```

-And enter the given password then you get the following flag :

```
picoCTF{s3cur3_c0nn3ct10n_5d09a462}
```

-Congrats you solve this,Happy haunting Nepker's

#### 2. Binary search

- First launch the instance then you can download the zip or use ssh to access the lab

-ssh -p 59057 ctf-player@atlas.picoctf.net and password: 84b12bae

-you have to choose the number between 1 -1000

-when you give the right guess then you get the flag as following:

```
picoCTF{g00d_gu355_2e90d29b}
```

-Wow you solve it,Happy haunting Nepker's

#### 3.Time Machine

-Download the challenge.zip file

-Then unzip the zip file

-Go to drop-in folder than use 'git log' command which gives the following results and flag :

```
sudo git log
```

```
commit 89d296ef533525a1378529be66b22d6a2c01e530 (HEAD -> master)
```

Author: picoCTF <ops@picoctf.com>

Date: Tue Mar 12 00:07:22 2024 +0000

picoCTF{t1m3m@ch1n3\_186cd7d7}

-Wow you solve it ,Happy haunting Nepker's

#### 4. Commitment Issues

-Download the zip file

-Then unzip the zip file

-go to drop-in folder and use 'git log' command which gives the following result:

```
sudo git log
```

```
commit 3899edb7f3110d613c72ad40083fd8feef703d0 (HEAD -> master)
```

Author: picoCTF <ops@picoctf.com>

Date: Sat Mar 9 21:09:58 2024 +0000

```
remove sensitive info
```

```
commit 6603cb4ff0c4ea293798c03a32e0d78d5ab12ca2
```

Author: picoCTF <ops@picoctf.com>

Date: Sat Mar 9 21:09:58 2024 +0000

```
create flag
```

-Then use 'git show' command which gives the flag

```
sudo git show
```

```
commit 3899edb7f3110d613c72ad40083fd8feef703d0 (HEAD -> master)
```

Author: picoCTF <ops@picoctf.com>

Date: Sat Mar 9 21:09:58 2024 +0000

remove sensitive info

```
diff --git a/message.txt b/message.txt
```

```
index ed59373..d552d1e 100644
```

```
--- a/message.txt
```

```
+++ b/message.txt
```

```
@@ -1 +1 @@
```

```
-picoCTF{s@n1t1z3_9539be6b}
```

```
+TOP SECRET
```

## 5. Blame Game

-Download the zip file

-Then unzip the zip file

-go to drop-in folder and use 'git log | grep picoCTF{' command which gives the following result:

```
picoCTF{@sk_th3_1nt3rn_cfca95b2}
```

## 6. First Find

-Download the zip file and unzip the file

-Use 'find' command and look for the 'uber-secret.txt' in the list

- use the cat command as given below

```
cat adequate_books/more_books/.secret/deeper_secrets/deepest_secrets/uber-secret.txt
```

-Then you will get the flag :

```
picoCTF{f1nd_15_f457_ab443fd1}
```

-Wow congrats you find the flag, Happy Haunting Nepker'sls

## 7. Big Zip

-Download the zip file and unzip the file

-Then use the 'grep' command as given after going in to the big-zip-files

```
grep -r picoCTF{
```

-You will get the flag:

```
picoCTF{gr3p_15_m4g1c_ef8790dc}
```

-Wow congrats you find the flag, Happy Haunting Nepker's

## 8. Binhex

-First Launch the instance

-Then use the 'nc' command as given : nc titan.picoctf.net 57235

-It shows as the given below prompt then you need to calculate as prompt says:

```
nc titan.picoctf.net 57235
```

Welcome to the Binary Challenge!"

Your task is to perform the unique operations in the given order and find the final result in hexadecimal that yields the flag.

Binary Number 1: 10011001

Binary Number 2: 01010100

Question 1/6:

Operation 1: '|'

Perform the operation on Binary Number 1&2.

Enter the binary result: 11011101

Correct!

Question 2/6:

Operation 2: '\*'

Perform the operation on Binary Number 1&2.

Enter the binary result: 11001000110100

Correct!

Question 3/6:

Operation 3: '&'

Perform the operation on Binary Number 1&2.

Enter the binary result: 00010000

Correct!

Question 4/6:

Operation 4: '<<'

Perform a left shift of Binary Number 1 by 1 bits.

Enter the binary result: 100110010

Correct!

Question 5/6:

Operation 5: '>>'

Perform a right shift of Binary Number 2 by 1 bits .

Enter the binary result: 101010

Correct!

Question 6/6:

Operation 6: '+'

Perform the operation on Binary Number 1&2.

Enter the binary result: 11101101

Correct!

Enter the results of the last operation in hexadecimal: ED

Correct answer!

- Finally you got The flag is: picoCTF{b1tw^3se\_0p3eR@tl0n\_su33essFuL\_1367e2c6}

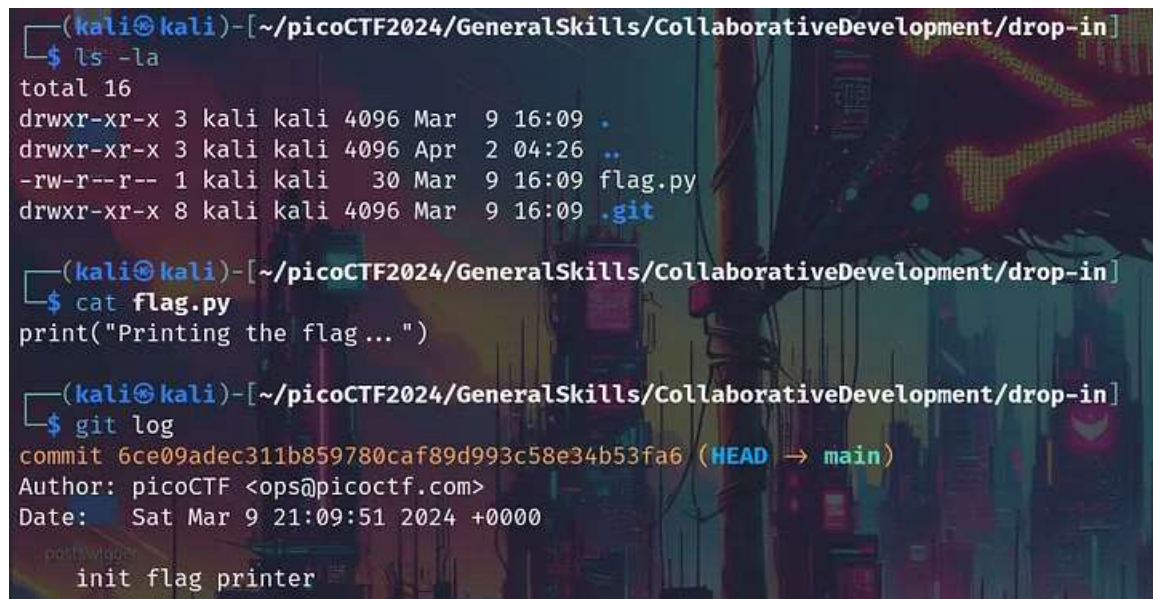
-You can use binary calculator : <https://www.rapidtables.com/calc/math/binary-calculator.html>

-Wow congrats you nailed it, Happy Haunting Nepker's

#### 9.Collaborative Development:

-First download the zip and unzip the zip file

-Then go to drip-in folder

A terminal window with a dark background and a colorful, abstract pattern on the right side. The terminal shows three commands and their outputs. The first command is 'ls -la' which lists files and their permissions, owner, group, size, and date. The second command is 'cat flag.py' which shows the content of the flag.py file. The third command is 'git log' which shows the commit history.

```
(kali㉿kali)-[~/picoCTF2024/GeneralSkills/CollaborativeDevelopment/drop-in]
$ ls -la
total 16
drwxr-xr-x 3 kali kali 4096 Mar  9 16:09 .
drwxr-xr-x 3 kali kali 4096 Apr  2 04:26 ..
-rw-r--r-- 1 kali kali  30 Mar  9 16:09 flag.py
drwxr-xr-x 8 kali kali 4096 Mar  9 16:09 .git

(kali㉿kali)-[~/picoCTF2024/GeneralSkills/CollaborativeDevelopment/drop-in]
$ cat flag.py
print("Printing the flag...")

(kali㉿kali)-[~/picoCTF2024/GeneralSkills/CollaborativeDevelopment/drop-in]
$ git log
commit 6ce09adec311b859780caf89d993c58e34b53fa6 (HEAD -> main)
Author: picoCTF <ops@picoctf.com>
Date: Sat Mar 9 21:09:51 2024 +0000

    init flag printer
```

But let's try new things :)

- git branch

```
(kali㉿kali)-[~/picoCTF2024/GeneralSkills/CollaborativeDevelopment/drop-in]
$ git branch
feature/part-1
feature/part-2
feature/part-3
* main
```

let's try:

- git checkout feature/part-1

```
(kali㉿kali)-[~/picoCTF2024/GeneralSkills/CollaborativeDevelopment/drop-in]
$ cat flag.py
print("Printing the flag... ")

(kali㉿kali)-[~/picoCTF2024/GeneralSkills/CollaborativeDevelopment/drop-in]
$ git checkout feature/part-1
Switched to branch 'feature/part-1'

(kali㉿kali)-[~/picoCTF2024/GeneralSkills/CollaborativeDevelopment/drop-in]
$ cat flag.py
print("Printing the flag... ")
print("picoCTF{t3@mwork_", end='')
```

Nice, we got the first part :)

Now Let's try part 2:

- git checkout feature/part-2

```
(kali㉿kali)-[~/picoCTF2024/GeneralSkills/CollaborativeDevelopment/drop-in]
$ git checkout feature/part-2
Switched to branch 'feature/part-2'

(kali㉿kali)-[~/picoCTF2024/GeneralSkills/CollaborativeDevelopment/drop-in]
$ cat flag.py
print("Printing the flag... ")

print("m@k3s_th3_dr3@m_", end='')
```

Finally, let's try the third:

- git checkout feature/part-3

```
(kali@kali)-[~/picoCTF2024/GeneralSkills/CollaborativeDevelopment/drop-in]
$ git checkout feature/part-3
Switched to branch 'feature/part-3'

(kali@kali)-[~/picoCTF2024/GeneralSkills/CollaborativeDevelopment/drop-in]
$ cat flag.py
print("Printing the flag... ")

print("w0rk_██████████")
```

We got it, now let's combine the three parts:

```
(kali@kali)-[~/picoCTF2024/GeneralSkills/CollaborativeDevelopment/drop-in]
$ python3 flag.py
Printing the flag...
picoCTF{t3@mW0rk_m@k3s_th3_dr3@m_w0rk_██████████}
```

-This is the flag: picoCTF{t3@mW0rk\_m@k3s\_th3\_dr3@m\_w0rk\_e4b79efb}

-WOw congrats you nailed it, Happy haunting Nepker's.

10. repetitions:

-Download the encoded file enc\_flag

-Then use the given command in the terminal :

```
sudo cat enc_flag | base64 -d | base64 -d | base64 -d | base64 -d | base64 -d | base64 -d
```

-Or you can use cyber chef to decode this message using this given link:

[https://cyberchef.io/#recipe=From\\_Base64\('A-Za-z0-9%2B/%3D',true\)From\\_Base64\('A-Za-z0-9%2B/%3D',true\)From\\_Base64\('A-Za-z0-9%2B/%3D',true\)From\\_Base64\('A-Za-z0-9%2B/%3D',true\)From\\_Base64\('A-Za-z0-9%2B/%3D',true\)&input=Vm1wR1UxRXISWGxVV0d4VFItEetWVIl3WkZOV2JHeHlWMjFHVjFKdGVEQlViRnBQWVd4S2RGVnNhRnBXVmxVeFdWWmFTMVpXV25WaApSbVJYWld0YWlxZFdXbXRTTWs1eVRsWldXQXBpVIZwVVZtMTBkMVZXWkZkVmEyUnBZbFphV0ZadE5WZFZaM0JwVTBWS2VsZFdVa05rCk1sWlhWbGhvV0dKWVFrOVZiRkpYVTBaa2NWUnVUbGRhTTBKWlZXcEdTMlZXV2tkYVNHUIhDazFzV25wV1YzaGhWbTFLUms1WE9WVlckVmtwRVZHeGFZVmRGTVZoU2JGcFNWMMFZLV1ZaR1ZtdE5SVFZlVj0V1UySllVbFZEYlVwWfYyNXNWV0pHY0haV2JHUKhaRWRXUmxcwphR2tLWWxScmVsWkVSbGRVTWtwelVXcFdUbEpZVGt4RFp6MDIDZz09Cg](https://cyberchef.io/#recipe=From_Base64('A-Za-z0-9%2B/%3D',true)From_Base64('A-Za-z0-9%2B/%3D',true)From_Base64('A-Za-z0-9%2B/%3D',true)From_Base64('A-Za-z0-9%2B/%3D',true)From_Base64('A-Za-z0-9%2B/%3D',true)&input=Vm1wR1UxRXISWGxVV0d4VFItEetWVIl3WkZOV2JHeHlWMjFHVjFKdGVEQlViRnBQWVd4S2RGVnNhRnBXVmxVeFdWWmFTMVpXV25WaApSbVJYWld0YWlxZFdXbXRTTWs1eVRsWldXQXBpVIZwVVZtMTBkMVZXWkZkVmEyUnBZbFphV0ZadE5WZFZaM0JwVTBWS2VsZFdVa05rCk1sWlhWbGhvV0dKWVFrOVZiRkpYVTBaa2NWUnVUbGRhTTBKWlZXcEdTMlZXV2tkYVNHUIhDazFzV25wV1YzaGhWbTFLUms1WE9WVlckVmtwRVZHeGFZVmRGTVZoU2JGcFNWMMFZLV1ZaR1ZtdE5SVFZlVj0V1UySllVbFZEYlVwWfYyNXNWV0pHY0haV2JHUKhaRWRXUmxcwphR2tLWWxScmVsWkVSbGRVTWtwelVXcFdUbEpZVGt4RFp6MDIDZz09Cg)

-Then you get the flag given below:



picoCTF{base64\_n3st3d\_dic0d!n8\_d0wnl04d3d\_9b59b35c}

-Wow congrats,You solve it. Happy Haunting Nepker's

## 11.Endianness

-Launch the instance

-Then netcat the server for ctf using given command:

```
nc titan.picoctf.net 58601
```

-Then do as given below:

```
nc titan.picoctf.net 58601
```

Welcome to the Endian CTF!

You need to find both the little endian and big endian representations of a word.

If you get both correct, you will receive the flag.

Word: puizh ( you just have to convert this word into hexadecimal)

-You can also use this link to convert: <https://www.rapidtables.com/convert/number/ascii-to-hex.html> (You also have to know about the endianness for Little and Big endianness letter)

Enter the Little Endian representation: 687A697570

Correct Little Endian representation!

Enter the Big Endian representation: 7075697A68

Correct Big Endian representation!

Congratulations! You found both endian representations correctly!

Your Flag is: picoCTF{3ndi4n\_sw4p\_su33ess\_25c5f083}

-Wow congrats you nailed it . Happy haunting Nepker's

## 12. Wave a flag:

-Downlaod the file using the given command:

```
sudo wget https://mercury.picoctf.net/static/b28b6021d6040b086c2226ebeb913bc2/warm
```

-Then use :

```
chmod +x warm
```

-Then use:

```
./warm -h
```

-You got the flag: picoCTF{b1scu1ts\_4nd\_gr4vy\_d6969390}

-Congrats you solve it,Happy Haunting Nepkers

### 13. runme.py

- Download runme.py file

-Then run the file using the given command:

```
python3 runme.py
```

-You get the flag : picoCTF{run\_s4n1ty\_run}

- Wow congrats you nailed it, Happy Haunting Nepker's

### 14.PW Crack 1

-Download the level.py and level1.flag.txt.enc file

-Look up for the password in level.py using any text editor

-Then run the python script and enter the password

-You will get the flag: picoCTF{545h\_r1ng1ng\_1b2fd683}

-Wow congrats, Happy Haunting Nepker's

### 15.PW Crack 2

-Download the level.py and level1.flag.txt.enc file

-Look up for the password in level.py using any text editor

-You can see the encoding there just decode that you get the password for this level

-You can decode using given commands:

```
python3
```

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

-you get the password 4ec9

-Then run the python script and enter the password

-you get the flag: picoCTF{tr45h\_51ng1ng\_9701e681}

-Congrats you got the flag, Happy Haunting Nepker's

## 16. HashingJobApp

-Firstly use the given command and give the answer for that you get it .

-you can use encrypt using command like : `echo -n chorus girls | md5sum`

- Or you can you the online tool for this like: <https://10015.io/tools/md5-encrypt-decrypt>

```
nc saturn.picoctf.net 54799
```

Please md5 hash the text between quotes, excluding the quotes: 'Cary Grant'

Answer:

```
eacef35ceac15db2ecaedd6b1b9280dd
```

```
eacef35ceac15db2ecaedd6b1b9280dd
```

Correct.

Please md5 hash the text between quotes, excluding the quotes: 'homeless shelters'

Answer:

```
ed96f5bcf2cb76f423a92190358b6881
```

```
ed96f5bcf2cb76f423a92190358b6881
```

Correct.

Please md5 hash the text between quotes, excluding the quotes: 'chorus girls'

Answer:

fdaf7298de6707185d68175ba4bd2f17

fdaf7298de6707185d68175ba4bd2f17

Correct.

picoCTF{4ppl1c4710n\_r3c31v3d\_3eb82b73}

## 17. Glitch Cat

-First launch the instance

-Then use the given 'nc' command

```
nc saturn.picoctf.net 52212
```

-where you get the flag message with encoding character's as given below:

```
picoCTF{gl17ch_m3_n07_' + chr(0x61) + chr(0x34) + chr(0x33) + chr(0x39) + chr(0x32) +  
chr(0x64) + chr(0x32) + chr(0x65) + '}
```

-I have write the python script to solve the challenge :

```
#!/usr/bin/python
```

```
flag_missing_part = chr(0x61) + chr(0x34) + chr(0x33) + chr(0x39) + chr(0x32) + chr(0x64) +  
chr(0x32) + chr(0x65)
```

```
print("picoCTF{gl17ch_m3_n07_" + flag_missing_part + "}")
```

-run using the command:

```
python3 script.py
```

-It gives you the flag: picoCTF{gl17ch\_m3\_n07\_a4392d2e}

-Wow congrats you got the flag

## 18. fixme1.py

-Download the file

-Then edit the fixme1.py using any editor and remove the indentation from the line 20

-Remove all the space that is before the print?

-Then run the script using :

```
python3 fixme1.py
```

-You got it,: picoCTF{1nd3nt1ty\_cr1515\_6a476c8f}

-Wow congrats and Happy Haunting Nepker's

#### 19. fixme2.py

-Download the file

-Then edit the fixme2.py using any editor and remove the indentation from the line 20

-Add == instead of = in line 22 of the code

-Then run the script using :

```
python3 fixme2.py
```

-You got it,: picoCTF{3qu4l1ty\_n0t\_4551gnm3nt\_f6a5aefc}

-Wow congrats and Happy Haunting Nepker's

#### 20. Convertme.py

-Download the file

-Then run the script and give the answer as it ask like given:

```
python convertme.py
```

If 18 is in decimal base, what is it in binary base?

Answer: 10010

That is correct! Here's your flag: picoCTF{4ll\_y0ur\_b4535\_9c3b7d4d}

-Wow congrats, You nailed it, Happy Haunting Nepker's

## 21. CodeBook:

-Download the code.py and codebook.txt in same directory

-Then check both files in same directory using 'ls' command

-Then run the code.py as given command:

```
python3 code.py
```

-Congrats you get the flag:picoCTF{c0d3b00k\_455157\_7d102d7a} and Happy Haunting Nepker's

## 22. Magikarp Ground Mission"

-First launch the instance

-then use 'ssh' command to the ctf server using following command:

```
ssh ctf-player@venus.picoctf.net -p 60600 and use the given password
```

-then do as given ..

```
cat 1of3.flag.txt
```

```
cat instruction-to-2of3.txt
```

```
cd /
```

```
cat 2of3.flag.txt
```

```
cat instruction-to-3to3.txt
```

```
cd ~/
```

```
cat 3of3.flag.txt
```

-Then concatenate all the flag text and you got that flag: picoCTF{xxsh\_0ut\_0f\_\\4t3r\_1118a9a4}

-Wow congrats you got it. Happy Haunting Nepker's

## 23. Tab, Tab, Attack:

-Download the zip file and unzip the file

-cd and enter tab until the last

-Then

chmod +x fang-of-haynekhtnamet

-Then

./fang-of-haynekhtnamet

-Congrats you got the flag:picoCTF{l3v3l\_up!\_t4k3\_4\_r35t!\_f3553887}

HAppy Haunting Nepker's

24. Obedient Cat:

-Download the file

-Then use the following command

cat flag

-Wow you got the flag : picoCTF{s4n1ty\_v3r1f13d\_f28ac910} and Happy Haunting Nepker's

25. 2Warm:

-As per the challenge i write the script that convert decimal to binary ,that script is given below:

```
#!/bin/bash
```

```
echo enter n
```

```
read n
```

```
c=$(echo "obase=2;$n" | bc)
```

```
echo binary $c
```

-You got the binary number and to have the flag you just have to cover the result with picoCTF{}

-wow congrats you done it. Happy Haunting Nepker's

## 26. First Grep

-Download the file and use the following command to get the flag

```
cat file | grep picoCTF{
```

-You got the flag: picoCTF{grep\_is\_good\_to\_find\_things\_dba08a45}

-Congrats and Happy Haunting Nepker's.

## 27. Warmed Up

-As the per challenge i write a script that convert the hexadecimal to decimal

```
#!/bin/bash
```

```
read n
```

```
echo $((($n))
```

-You just need to specify the number after running the script

-You got the result but you have to cover the result with picoCTF{} to make a flag.

-Then you nailed it. Happy Haunting Nepker's

## 28. Python Wrangling:

-Download all the files from the challenge

-Then use text editor and change all the 'pole.txt' to 'flag.txt.eng'

-Run the python script using command:

```
python3 ende.py -d flag.txt.e
```

-and enter the password from the pw.txt file

-Wow awesome you got the flag :picoCTF{4p0110\_1n\_7h3\_h0us3\_6008014f}

-Congrats and Happy Haunting Nepker's



## 29. Nice Netcat

-Use the netcat command given by the challenge as given below:

```
nc mercury.picoctf.net 22342
```

-copy the ascii number from the output of the netcat and

-You can use the online tool to decrypt the ascii code using the link given:

<https://www.duplichecker.com/ascii-to-text.php>

-Or you can use a python script to decrypt the ascii code to text as given below:

```
#!/usr/bin/python
```

```
# Input list
```

```
lst = [112 ,105 , 99, 111, 67, 84, 70, 123, 103, 48, 48, 100, 95, 107, 49, 116, 116, 121, 33, 95, 110, 49, 99, 51, 95, 107, 49, 116, 116, 121, 33, 95, 53, 102, 98, 53, 101, 53, 49, 100, 125, 10]
```

```
# Using chr() Method
```

```
res = ""
```

```
for i in lst:
```

```
    res = res + chr(i)
```

```
print (str(res))
```

-Then you get the flag : picoCTF{g00d\_k1tty!\_n1c3\_k1tty!\_5fb5e51d}

-Wow congrats you get it, Happy Haunting Nepker's

## 30. Static ain't make nouse

-Download the both script and static file from the challenge

-Look for the file type of both file using as given below:

```
file static
```

file ltdish.sh

-run the script with static with root privilege as given below:

```
sudo bash ltdish.sh static
```

-It give two files after running the script as given below:

```
static.ltdis.strings.txt
```

```
& static.ltdis.x86_64.txt
```

- Then use the cat and grep command to get flag as given below:

```
cat static.ltdis.strings.txt | grep picoCTF{
```

-Wow congrats you got the flag: picoCTF{d15a5m\_t34s3r\_98d35619} and Happy Haunting Nepker's

### 31. Bases

-First save the text given by challenge

-you can decode this using online tool given below:

<https://www.base64decode.org/>

-Or you can use the command line to decode as given below:

```
echo "bDNhcm5fdGgzX3lwcDM1" | base64 -d
```

- you just need to wrap the output with picoCTF{l3arn\_th3\_r0p35}

-Congrats you nailed it, Happy Haunting Nepker's

### 32. Strings it

-Downlaod the string file

-use the give command to get the flag without running it

```
strings strings | grep pico
```

-Wow congrats you got the flag: picoCTF{5tRIng5\_1T\_827aee91} and Happy Haunting Nepker's

33. What's a net cat?

-As per the challenge just netcat to the challenge server using given command:

```
nc jupiter.challenges.picoctf.org 64287
```

-Wow congrats you get the flag: picoCTF{nEtCat\_Mast3ry\_284be8f7} . Happy Haunting Nepker's

34. Lets warm up

-As per the challenge you just have to convert hex to ascii for this challenge

-So i have wrote a python script that convert hex to acsii

```
#!/usr/bin/python
```

```
import binascii
```

```
def hex_to_ascii(hex_str):
```

```
    hex_str = hex_str.replace(' ', '').replace('0x', '').replace('\t', '').replace('\n', '')
```

```
    ascii_str = binascii.unhexlify(hex_str)
```

```
    return ascii_str
```

```
hex_input = '0x70'
```

```
ascii_output = hex_to_ascii(hex_input)
```

```
print(format(ascii_output))
```

-Wow you just need to wrap the outp with picoCTF{p}

-Congrats and Happy Haunting Nepker's

35. don't-you-love-banners

-First use nc to the first sever which leaks the credentials as given below:

nc tethys.picoctf.net 54927

-Then nc to another server using given command

nc tethys.picoctf.net 56420

-Enter as per the question:

what is the password?

My\_Passw@rd\_@1234

What is the top cyber security conference in the world?

DEF CON

the first hacker ever was known for phreaking(making free phone calls), who was it?

JOHN

-Then look up for the banner and script file

-Then look at the root directory as given command below:

ls -al /root

-There you can find flag.txt and script.py and many more

-Now you just have to create a symbolic link but before delete the banner first using given command:

rm banner

-Then create link with /root/flag.txt

- ln -s /root/flag.txt /home/player/banner

-Then exit from the server and again use netcat command as given below:

nc tethys.picoctf.net 56420

-You see the flag there...: picoCTF{b4nn3r\_gr4bb1n9\_su((3sfu11y\_218ef5d6}

-Wow , Congrats you got the flag and Happy Haunting Nepker's

36. ASCII Numbers:

-Copy the ASCII Numbers in the text file and you can decode using the given link:

<https://onlinestringtools.com/convert-ascii-to-string>

-Or you can get it through the python script that i have wrote:

```
hx = "0x70 0x69 0x63 0x6f 0x43 0x54 0x46 0x7b 0x34 0x35 0x63 0x31 0x31 0x5f 0x6e 0x30 0x5f  
0x71 0x75 0x33 0x35 0x37 0x31 0x30 0x6e 0x35 0x5f 0x31 0x6c 0x6c 0x5f 0x74 0x33 0x31 0x31  
0x5f 0x79 0x33 0x5f 0x6e 0x30 0x5f 0x6c 0x31 0x33 0x35 0x5f 0x34 0x34 0x35 0x64 0x34 0x31  
0x38 0x30 0x7d "
```

```
str = hx.replace("0x", "")
```

```
str2 = str.replace(" ", "")
```

```
print(str2)
```

```
final = bytes.fromhex(str2).decode("utf-8")
```

```
print(final)
```

-Wow congrats you got the flag : picoCTF{45c11\_n0\_qu35710n5\_1ll\_t311\_y3\_n0\_l135\_445d4180} , Happy Haunting Nepker's

37.Specialer:

-I logged in through the given instances and tried to explore different possibility to get the flag. Initially, I tried some linux system commands. But, that didn't provide any results. It really seemed off. I guess the creator of this challenge has either deleted / moved these binaries from /bin directory.

```

Are you sure you want to continue connecting (yes/no)
Warning: Permanently added '[saturn.picoctf.net]:49
ctf-player@saturn.picoctf.net's password:
Specialer$ clear
-bash: clear: command not found
Specialer$ ls
-bash: ls: command not found
Specialer$ pwd
/home/ctf-player
Specialer$ ll
-bash: ll: command not found
Specialer$ cat *.*
-bash: cat: command not found
Specialer$ █

```

But luckily, there are commands like `pwd`, `cd` that provided intended response. So, I decided to autocomplete the command by pressing `<tab>` button and it yielded the following result:

```

Specialer$ █
!      bind      compopt  elif      fc        if        printf    shift     true      while
./     break     continue else      fg        in        pushd     shopt     type      {
:      builtin  coproc   enable    fi        jobs      pwd       source    typeset   }
[      caller   declare  esac      for       kill      read      suspend   ulimit
[[     case      dirs     eval      function  let       readarray test      umask
]]     cd        disown   exec      getopt    local     readonly then      unalias
alias  command   do       exit      hash      logout    return    time      unset
bash   compgen   done     export    help      mapfile   select    times     until
bg     complete  echo     false     history   popd      set       trap      wait
Specialer$ █

```

Fortunately, these commands were working fine. So, I decided to construct a shell program that would print the files & folders for the current directory.

Commands like `for`, `while`, `do`, `then`, `done`, `if`, `elif`, `echo` can be constructed as a program and it can be used to print the working directory contents. Quickly, I wrote a simple function that could perform the same.

```
for file in *
```

do

```
if [ -d "$file" ]; then
```

```
    echo "$file is a directory."
```

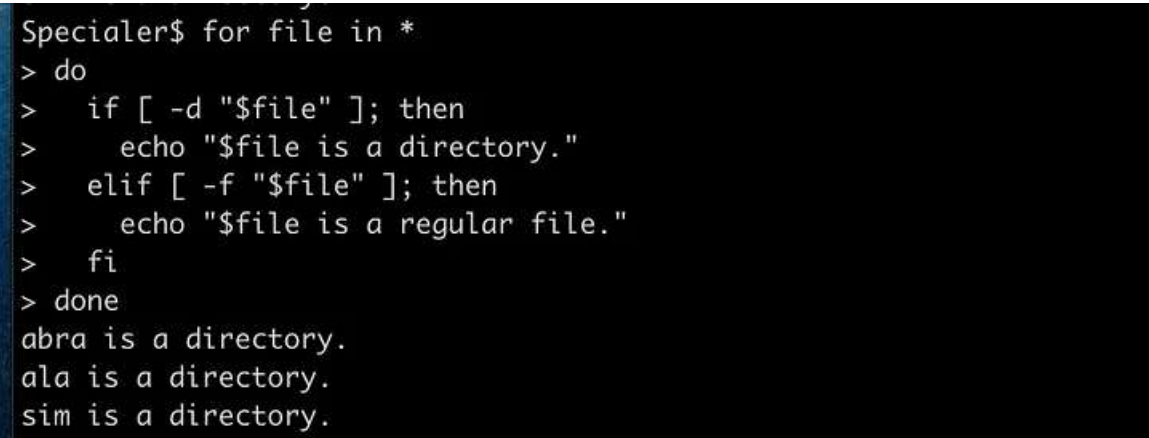
```
elif [ -f "$file" ]; then
```

```
    echo "$file is a file."
```

```
fi
```

done

This program would differentiate the files and folders and display it accordingly.



```
Specialer$ for file in *
> do
>   if [ -d "$file" ]; then
>     echo "$file is a directory."
>   elif [ -f "$file" ]; then
>     echo "$file is a regular file."
>   fi
> done
abra is a directory.
ala is a directory.
sim is a directory.
```

From the above result, we could see that there are three directories. Let's poke in and execute the same command.

Abra Directory:

```

Specialer$ cd abra
Specialer$ for file in *
> do
>   if [ -d "$file" ]; then
>     echo "$file: directory."
>   elif [ -f "$file" ]; then
>     echo "$file: file."
>   fi
> done
cadabra.txt: file.
cadaniel.txt: file.

```

Similarly, Ala and sim folder had the following results.

```

Specialer$ cd ../ala
Specialer$ for file in *
> do
>   if [ -d "$file" ]; then
>     echo "$file: directory."
>   elif [ -f "$file" ]; then
>     echo "$file: file."
>   fi
> done
kazam.txt: file.
mode.txt: file.
Specialer$ cd ../
.hushlogin .profile abra/ ala/ sim/
Specialer$ cd ../sim
Specialer$ for file in *
> do
>   if [ -d "$file" ]; then
>     echo "$file: directory."
>   elif [ -f "$file" ]; then
>     echo "$file: file."
>   fi
> done
city.txt: file.
salabim.txt: file.
Specialer$ █

```

Now, let's modify the code a bit, and print the content of each files from their folders:

for folder in abra ala sim



```
do
    cd "$folder"
    for file in *
    do
        if [ -d "$file" ]; then
            echo "$file: directory."
        elif [ -f "$file" ]; then
            echo "$folder/$file:"
            printf "%s " $(<$file) # input redirection; alternative to 'cat'
            printf "\n\n"
        fi
    done
    cd ..
done
```

```

Specialer$ for folder in abra ala sim
> do
>   cd "$folder"
>   for file in *
>   do
>     if [ -d "$file" ]; then
>       echo "$file: directory."
>     elif [ -f "$file" ]; then
>       echo "$folder/$file:"
>       printf "%s " $(<$file)
>       printf "\n\n"
>     fi
>   done
>   cd ..
> done
abra/cadabra.txt:
Nothing up my sleeve!

abra/cadaniel.txt:
Yes, I did it! I really did it! I'm a true wizard!

ala/kazam.txt:
return 0 picoCTF{y0u_d0n7_4ppr3c1473_wh47_w3r3_d01ng_h3r3_a8567b6f}

ala/mode.txt:
Yummy! Ice cream!

sim/city.txt:
05ed181c-4aa0-4d4a-8505-2fe6ca9097d3

sim/salabim.txt:
#He was so kind, such a gentleman tied to the oceanside#

```

-Wow congrats you got the flag. Happy Haunting Nepker's

### 38. SansAlpha

-Launch the instance

-ssh to the server as given credentials below:

ssh -p 57022 ctf-player@mimas.picoctf.net

passwor: 84b12bae

```
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 6.5.0-1014-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

SansAlpha$ ls
SansAlpha: Unknown character detected
SansAlpha$ █
```

As I tried to find commands without letters and \ to use in this challenge, I found wildcards which are special symbols to match one or more characters in file names, file paths, or command-line operations.

Wildcards (tldp.org)

Globbering (tldp.org)

Here are some Wildcard characters I will use later on:

? : match 1 character, ex. /??? -> /bin /dev /etc /lib

\* : match 0 or more characters, ex. /lib\* -> /lib /lib32 /lib64

[ ] : matches any single character within the specified range or set, ex. file[345].txt or file[3-5].txt -> file3.txt, file4.txt, file5.txt, file6.txt

[!] : matches any character that is not in the specified range or set, ex. file[!12].txt -> any files

from file0.txt-file9.txt, except file1.txt and file2.txt

I started by searching the current directory using `./*` to see all the files and noticed that, unlike our Kali terminal, this bash only returns 1 result at a time. I continued to use `*` and found a flag file, but don't have permission to read it. Therefore, I switched to using `'?'` to search more and found a couple of files in the current directory which also returned permission denied.

```
SansAlpha$ ./*  
bash: ./blargh: Is a directory
```

```
SansAlpha$ ./*/*  
bash: ./blargh/flag.txt: Permission denied
```

Having no usable commands, I had no other choice but to start searching from the root directory(`'/'`). My strategy was to keep adding `'?'` one by one to make sure I found as many as possible files or paths.

```

SansAlpha$ /?
bash: /?: No such file or directory

SansAlpha$ /??
bash: /?: No such file or directory

SansAlpha$ /???
bash: /bin: Is a directory

SansAlpha$ /????
bash: /boot: Is a directory

SansAlpha$ /?????
bash: /lib32: Is a directory

SansAlpha$ /??????
bash: /libx32: Is a directory

SansAlpha$ /???????
bash: /????????: No such file or directory

```

Thanks to this method, I found that the full path to the flag file which is /home/ctf-player/blargh/flag.txt, meaning that we're currently on /home/ctf-player. Nevertheless, as we expected, calling the full path doesn't help us with the permission denied issue. I paid extra attention to /bin/ as it contains essential command binaries, such as ls, cat, and chmod. Most of the commands found seem not to help with the challenge except /bin/base64 which somehow also matches with the hint given for this challenge. I tried to call "/bin/base64 /home/ctf-player/blargh/flag.txt" hoping it would return the base64 encrypted version of the flag.

```

SansAlpha$ /???/??????
/bin/base32: extra operand '/bin/base64'
Try '/bin/base32 --help' for more information.

SansAlpha$ /???/?????? /???/????????????/??????/????????
/bin/base32: extra operand '/bin/base64'
Try '/bin/base32 --help' for more information.

```

The problem is that the system seems to be confused with the command. From my point of view, I think the server, in fact, matches all the commands just like our Kali terminal but returns to us only the first command it found. In this case, It could look like this on the server side:

```
(root@kali)-[~]
# /bin/base32 /bin/base58 /bin/base64
/bin/base32: extra operand '/bin/base64'
Try '/bin/base32 --help' for more information.
```

That's why it tried to base32 the file after /bin/base32, then there's also /bin/base64 which exceeds the expected operand. For this reason, I guess we have to be more specific about the command we glob. I tried use /???/???64 to find only the file that's in /??? path, the name has any 4 characters followed by '64'.

```
SansAlpha$ /???/???64 /???/????????????/??????/????????
/bin/base64: extra operand '/bin/x86_64'
Try '/bin/base64 --help' for more information.
```

But then /bin/x86\_64 also matches the pattern, so I added [!\_] at the 4th character to exclude any file that has '\_' at the 4th character. After all this mess, we finally got base64 of the flag which we can easily decode with any method of our choice.

```
SansAlpha$ /???/???[!_]64 /???/????????????/??????/????????
cmV0dXJuIDAgcGljb0NURns3aDE1X211MTcxdjNyNTNfMTVfbTRkbjM1NV8xNDUyNTZlY30=
```

```
(root@kali)-[/home/kali/Downloads/pico2024]
# echo "cmV0dXJuIDAgcGljb0NURns3aDE1X211MTcxdjNyNTNfMTVfbTRkbjM1NV8xNDUyNTZlY30=" | base64 -d
return 0 picoCTF{7h15_mu171v3r53_15_m4dn355_b0d5e855}
```

-Wow congrats you agot the flag :picoCTF{7h15\_mu171v3r53\_15\_m4dn355\_b0d5e855}

-Happy Haunting Nepker's

39.Useless

-Launch the instance

-Login to the challenge server using ssh as given below:

```
ssh picoplayer@saturn.picoctf.net -p 62618
```

```
password: password
```

-Then use ls command to see the list of content in the server

-Then i run the useless script using following command:

```
./useless
```

-Then it gives me some hint about manual page

-I just use the man command to see the manual page as given below command and :

```
man useless
```

-Wow you got the flag: picoCTF{us3l3ss\_ch4ll3ng3\_3xploit3d\_3823}

-Congrats and Happy Haunting Nepker's

40. Special:

-Launch the instance for the challenge

-Use ssh command to reach the challenge server as given below command and credentials:

```
ssh -p 63369 ctfplayer@saturn.picoctf.net
```

```
password: d137d16e
```

-I have used ls command but it's says me to use \${paramter} so i used as given hint from the server .

-And i used all the command within the \${parameter} where i got the flag

```
Special$ ls
```

```
ls
```

```
sh: 1: ls: not found
```

```
Special$ ./
```

```
I
```

sh: 1: !: not found

Special\$ \${parameter?!s}

\${parameter?!s}

sh: 1: parameter: !s

Special\$ \${:!s}

\${:!s}

sh: 1: Bad substitution

Special\$ \${parameter=!s}

\${parameter=!s}

blargh

Special\$ \${parameter=cd blargh}

\${parameter=cd blargh}

Special\$ \${parameter=ls blargj}

\${parameter=ls blargj}

ls: cannot access 'blargj': No such file or directory

Special\$ \${parameter=ls blargh}

\${parameter=ls blargh}

flag.txt

Special\$ \${parameter=cat blargh/flag.txt}

\${parameter=cat blargh/flag.txt}

picoCTF{5p311ch3ck\_15\_7h3\_w0r57\_3befb794}

-wow you got the flag: picoCTF{5p311ch3ck\_15\_7h3\_w0r57\_3befb794} and Happy Haunting  
Nepker's.



#### 41. Permissions:

-Launch the instance for challenge

-Then use ssh to get into the server as given below:

```
ssh picoplayer@saturn.picoctf.net -p 59272
```

```
password: yX-YQgX-vS
```

-I have check the permission for the user with the root privilege and use that command to escalate the privilege to the root user as given below command:

```
picoplayer@challenge:/etc$ whoami
```

```
picoplayer
```

```
picoplayer@challenge:/etc$ sudo -l
```

```
[sudo] password for picoplayer:
```

Matching Defaults entries for picoplayer on challenge:

```
env_reset, mail_badpass,
```

```
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin
```

User picoplayer may run the following commands on challenge:

```
(ALL) /usr/bin/vi
```

```
picoplayer@challenge:/etc$ sudo vi test
```

```
root@challenge:/etc# whoami
```

```
root
```

```
root@challenge:/home/picooplayer# cd /root/
```

```
root@challenge:~# ls -al
```

```
total 12
```

```
drwx----- 1 root root  23 Aug  4 2023 .
```

```
drwxr-xr-x 1 root root 62 Sep 13 13:01 ..
```

```
-rw-r--r-- 1 root root 3106 Dec 5 2019 .bashrc
```

```
-rw-r--r-- 1 root root 35 Aug 4 2023 .flag.txt
```

```
-rw-r--r-- 1 root root 161 Dec 5 2019 .profile
```

```
root@challenge:~# cat .flag.txt
```

```
picoCTF{uS1ng_v1m_3dit0r_55878b51}
```

```
root@challenge:~#
```

```
-wow yo nailed the flag: picoCTF{uS1ng_v1m_3dit0r_55878b51} ,Happy Haunting Nepker's.
```

#### 42. Chrono:

```
-Launch the instance for the challenge server
```

```
-ssh to the server using the following commands:
```

```
ssh picoplayer@saturn.picoctf.net -p 62057
```

```
password: ENAFb6zfzn
```

```
-In this challenge you need to know about the crontab with it's directory file and usages
```

```
-You can look the crontab jobs using the command: crontab -l
```

```
-Then you need to check the crontab using the following command:
```

```
cat /etc/crontab
```

```
-You got the flag: picoCTF{Sch3DUL7NG_T45K3_L1NUX_1d781160}
```

```
-Congrats and Happy Haunting Nepker's
```

#### 43. Serprntine:

```
-Download the python script from the challenge server
```

```
-Run the code and see the result what you see
```

```
-To get the flag you must comment out the code or remove the code and add the print function
```

for flag in the code :

code should be look like this:

```
import random
```

```
import sysdef 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 = chr(0x15) + chr(0x07) + chr(0x08) + chr(0x06) + chr(0x27) +
chr(0x21) + chr(0x23) + chr(0x15) + chr(0x5c) + chr(0x01) + chr(0x57) + chr(0x2a) + chr(0x17) +
chr(0x5e) + chr(0x5f) + chr(0x0d) + chr(0x3b) + chr(0x19) + chr(0x56) + chr(0x5b) + chr(0x5e) +
chr(0x36) + chr(0x53) + chr(0x07) + chr(0x51) + chr(0x18) + chr(0x58) + chr(0x05) + chr(0x57) +
chr(0x11) + chr(0x3a) + chr(0x0f) + chr(0x0e) + chr(0x59) + chr(0x06) + chr(0x4d) + chr(0x55) +
chr(0x0c) + chr(0x0f) + chr(0x14)def print_flag():
```

```
    flag = str_xor(flag_enc, 'enkidu')
```

```
    print(flag)def print_encouragement():
```

```
    encouragements = ['You can do it!', 'Keep it up!',
```

```
        'Look how far you \ve come!']
```

```
    choice = random.choice(range(0, len(encouragements)))
```

```
    print('\n-----')
```

```
    print(encouragements[choice])
```

```
    print('-----\n\n')print_flag()
```

```
# def main():# print(
```

```

# ""
# Y
# .-^-.
# /  \ \  .-~~-.
# ()  () /  _ _  \  _ _ _
# \\ _/ / /  \ \  \  .~ _ _ ~.
# || / /  \ \  \ \  .!~  ~-. \.
# || / /  ) )  / /  \ \.
# \\\\ _/ /  / /  / /  "
# \\ _ _ .!  / /  ( (
#      / /  \ \ \ \ \
#      / /  \ \ \ \ \
#      / /  ) )
#      ( (  / /
#      \ \.  ! /
#      \. ~ _ _ _ _ ~ .!
#      ~. _ _ _ _ .~
# ""
# )
# print('Welcome to the serpentine encourager!\\n\\n')

# while True:
#     print('a) Print encouragement')
#     print('b) Print flag')

```

```

# print('c) Quit\\n')

# choice = input('What would you like to do? (a/b/c) ')


# if choice == 'a':

#     print_encouragement()


# elif choice == 'b':

#     print('\\nOops! I must have misplaced the print_flag function! Check my source code!\\n\\n')

# elif choice == 'c':

#     sys.exit(0)


# else:

#     print('\\nI did not understand "' + choice + '", input only "a", "b" or "c"\\n\\n')# if
__name__ == "__main__":

# main()

```

-After that run the script again, You will get the flag: picoCTF{7h3\_r04d\_l355\_7r4v3l3d\_aa2340b2}

-Wow congrats you got it, Happy Haunting Nepker's

#### 44.PW Crack 3

-Download all the files given in the challenge

-Run the python script using following command:

```
python3 level3.py
```

-It ask for password but we don;t know where is password:

-So i check other files using cat but i can't find

-After that i just check the script and it's crazy there is 7 password as given below:

```
Pos_pw_list: 8799", "d3ab", "1ea2", "acaf", "2295", "a9de", "6f3d"
```

-Run the script and enter the password one by one finally i found the flag :picoCTF{m45h\_fl1ng1ng\_6f98a49f}

-Wow congrats you nailed it, Happy Haunting Nepker's

#### 45. Pw Crack 4

-Download all the files given in the challenge

-Run the python script using following command:

```
python3 level4.py
```

-It ask for password but we don;t know where is password:

-So i check other files using cat but i can't find

-After that i just check the script and it's crazy there is 7 password as given below:

```
pos_pw_list = ["158f", "1655", "d21e", "4966", "ed69", "1010", "dded", "844c", "40ab", "a948",  
"156c", "ab7f""4a5f", "e38c", "ba12", "f7fd", "d780", "4f4d", "5ba1", "96c5", "55b9", "8a67",  
"d32b", "aa7a", "514b", "e4e1""1230", "cd19", "d6dd", "b01f", "fd2f", "7587", "86c2", "d7b8",  
"55a2", "b77c", "7ffe", "4420", "e0ee", "d8fb", "d748", "b0fe", "2a37", "a638", "52db", "51b7",  
"5526", "40ed", "5356", "6ad4", "2ddd", "177d", "84ae", "cf88", "97a3", "17ad", "7124", "eff2",  
"e373", "c974", "7689", "b8b2", "e899", "d042", "47d9", "cca9""ab2a", "de77", "4654", "9ecb",  
"ab6e", "bb8e", "b76b", "d661", "63f8", "7095", "567e", "b837", "2b80", "ad4f""c514", "ffa4",  
"fc37", "7254", "b48b", "d38b", "a02b", "ec6c", "eacc", "8b70", "b03e", "1b36", "81ff""77e4",  
"dbe6", "59d9", "fd6a", "5653", "8b95", "d0e5"]
```

-Let's modify the level\_4\_pw\_check() function to automate this process using a for loop. We'll check all the passwords and exit when the correct one is found. Of course, it should also return the flag! code should look like this:

```
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('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()
```

```
#level_4_pw_check()
```

```
# The strings below are 100 possibilities for the correct password.
```

```
# (Only 1 is correct)
```

```
pos_pw_list = ["158f", "1655", "d21e", "4966", "ed69", "1010", "dded", "844c", "40ab", "a948",  
"156c", "ab7f", "4a5f", "e38c", "ba12", "f7fd", "d780", "4f4d", "5ba1", "96c5", "55b9", "8a67",  
"d32b", "aa7a", "514b", "e4e1", "1230", "cd19", "d6dd", "b01f", "fd2f", "7587", "86c2", "d7b8",  
"55a2", "b77c", "7ffe", "4420", "e0ee", "d8fb", "d748", "b0fe", "2a37", "a638", "52db", "51b7",  
"5526", "40ed", "5356", "6ad4", "2ddd", "177d", "84ae", "cf88", "97a3", "17ad", "7124", "eff2",  
"e373", "c974", "7689", "b8b2", "e899", "d042", "47d9", "cca9", "ab2a", "de77", "4654",  
"9ecb", "ab6e", "bb8e", "b76b", "d661", "63f8", "7095", "567e", "b837", "2b80", "ad4f", "c514",  
"ffa4", "fc37", "7254", "b48b", "d38b", "a02b", "ec6c", "eacc", "8b70", "b03e", "1b36", "81ff",  
"77e4", "dbe6", "59d9", "fd6a", "5653", "8b95", "d0e5"]
```

```
def level_4_pw_check():
```

```
    for pw in pos_pw_list:
```

```
        pw_hash = hash_pw(pw)
```

```
        if( pw_hash == correct_pw_hash ):
```

```
            decryption = str_xor(flag_enc.decode(), pw)
```

```
            print(decryption)
```

```
level_4_pw_check()
```

-You got the flag after running the script :picoCTF{fl45h\_5pr1ng1ng\_cf341ff1}

-Congrats and Happy Haunting Nepker's



#### 46. Pw Crack 5:

-Download all the files given in the challenge server

-run the script,it ask for password but there is alot of password so i just rewrite the code to run the password list in the script as follows:

```
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('level5.flag.txt.enc', 'rb').read()
```

```
correct_pw_hash = open('level5.hash.bin', 'rb').read()
```

```
dictionary = open( 'dictionary.txt', 'r').read() # I have just opened the file
```

```
dictionary_as_a_list = dictionary.split("\n") # Here i have just converted it into a list
```

```

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(dictionary_as_a_list):

    for i in range(1,65537):

        user_pw = dictionary_as_a_list[i] # Here I passed the list as the password

        user_pw_hash = hash_pw(user_pw)


    #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

    continue

```

```
#print("That password is incorrect")
```

# Syntax of strip : <variable>.strip()

#This will remove the whitespaces from the hash

```
level_5_pw_check(dictionary_as_a_list)
```

-You got the flag: picoCTF{h45h\_sl1ng1ng\_ffcda23}

-Congrats and Happy Haunting Nepker's

47. 1\_wanna\_b3\_a\_r0ck5tar:

-Download the lyrics.txt

-Download thr rockstar.py in your terminal using following command:

```
pip install rockstar.py
```

-Then run the rockstar which create the script for python

```
rockstar.py -i lyrics.txt -o rockstar.py
```

-Then it gibes the number after the many things, You just have to use the number and decode that number from ASCII to text using the following link:

<https://www.browserling.com/tools/ascii-to-text>

-After convert you got the string and just warp this text with picoCTF{BONJVI} (NOOT RIGHT) sorry NEpker's

48.Plumbing:

-Just connect to the server using netcat as following command:

```
nc jupiter.challenges.picoctf.org 7480
```

-There is lot of stuff so we used the grep command to get the flag as given below:

```
nc jupiter.challenges.picoctf.org 7480 | grep picoCTF{
```

-You got the flag: picoCTF{digital\_plumb3r\_06e9d954}

-Congrats and Happy Haunting Nepker's

49. Based:

-Use the netcat command as given by the challenge as follows

nc jupiter.challenges.picoctf.org 29221

-You just have to answer the question with converting the binary, octal and hex number to word/string

-You can use the online tools as follows:

For binary to text: <https://www.duplichecker.com/binary-to-text.php>

For hex to text: <https://www.duplichecker.com/hex-to-text.php>

For Octal to text: <https://v2.cryptii.com/octal/text>

- you got the flag: picoCTF{learning\_about\_converting\_values\_00a975ff}

-Congrats and Happy Haunting Nepker's

50. flag\_shop

-Just use the netcat command as given below:

nc jupiter.challenges.picoctf.org 4906

-Do as i do below:

Welcome to the flag exchange

We sell flags

1. Check Account Balance

2. Buy Flags

3. Exit

Enter a menu selection

2

Currently for sale

1. Definitely not the flag Flag

2. 1337 Flag

1

These knockoff Flags cost 900 each, enter desired quantity

400000000

The final cost is: -777252864

Your current balance after transaction: 777254684

Welcome to the flag exchange

We sell flags

1. Check Account Balance

2. Buy Flags

3. Exit

Enter a menu selection

1

Balance: 777254684

Welcome to the flag exchange

We sell flags

1. Check Account Balance

2. Buy Flags

3. Exit

Enter a menu selection

2

Currently for sale

1. Definitely not the flag Flag

2. 1337 Flag

2

1337 flags cost 100000 dollars, and we only have 1 in stock

Enter 1 to buy one1

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

Welcome to the flag exchange

We sell flags

-You got the flag: picoCTF{m0n3y\_bag5\_9c5fac9b}

-Wow congrats and Happy Haunting Nepker's