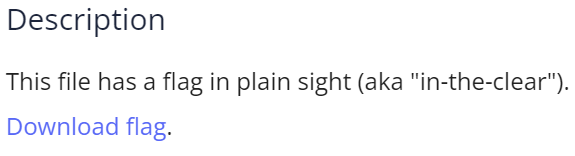
General Skills

1.Obident Cat

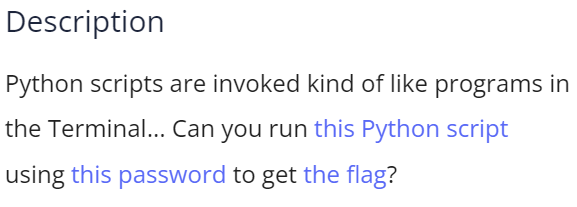


Solution: using cat and grep command.



Flag: picoCTF{s4n1ty\_v3r1f13d\_28e8376d}

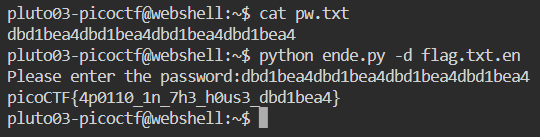
### 2. Python Wrangling



Solution: on inspecting the python script we see the code to decrypt.

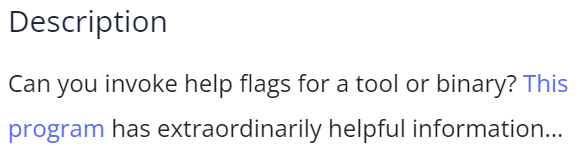


Using the password in pw.txt and the above command we get the flag.



Flag: picoCTF{4p0110\_1n\_7h3\_h0us3\_dbd1bea4}

### 3. Wave a flag



Solution: using strings and grep command.

Flag: picoCTF{b1scu1ts\_4nd\_gr4vy\_18788aaa}

### 4. Nice netcat...

### 

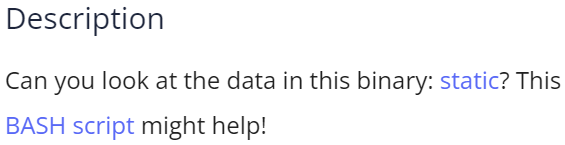
### 

### This gives a list of ASCII numbers. Using online ASCII to text converter.

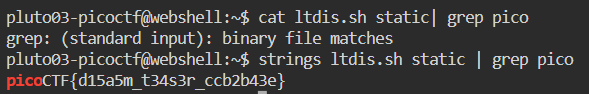
### 

### Flag: picoCTF{g00d\_k1tty!\_n1c3\_k1tty!\_d3dfd6df}

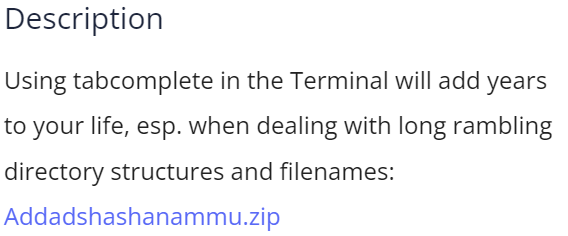
### 5. Static ain't always noise



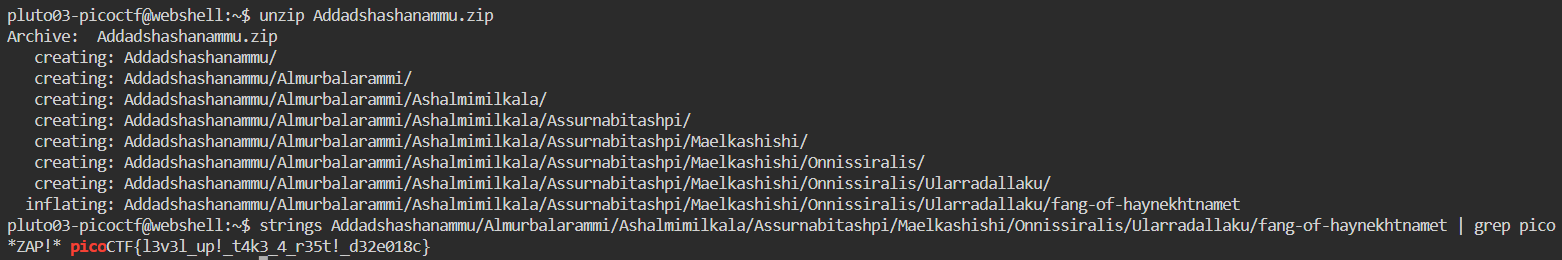
Solution: read the BASH script with static argument

Flag: picoCTF{d15a5m\_t34s3r\_ccb2b43e}

### 6. Tab, Tab, Attack



Solution: unzip the file. Follow the instructions and use tab.

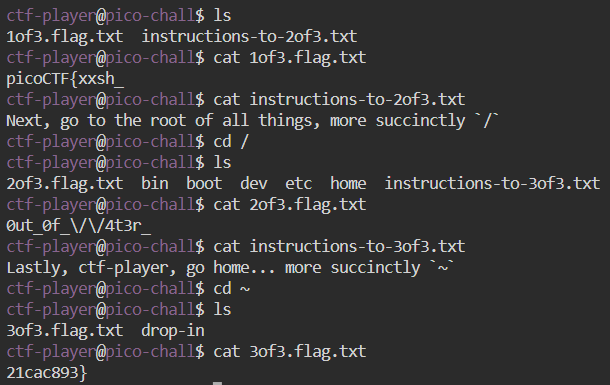


Flag: picoCTF{l3v3l\_up!\_t4k3\_4\_r35t!\_d32e018c}

### 7. Magikarp Ground Mission

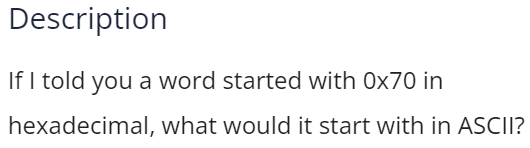
### 

### Solution:

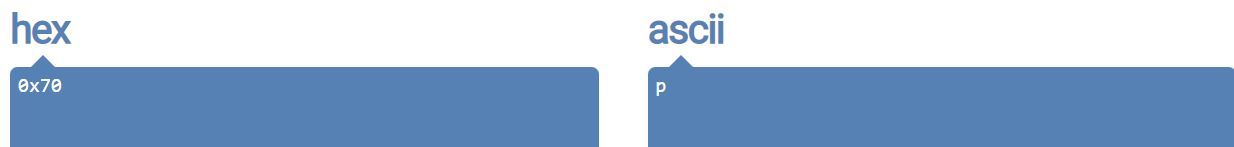


Flag: picoCTF{xxsh\_0ut\_0f\_\/\/4t3r\_21cac893}

8. Lets Warm Up

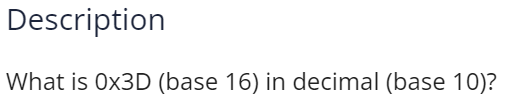


Solution: using online hex to ASCII converter

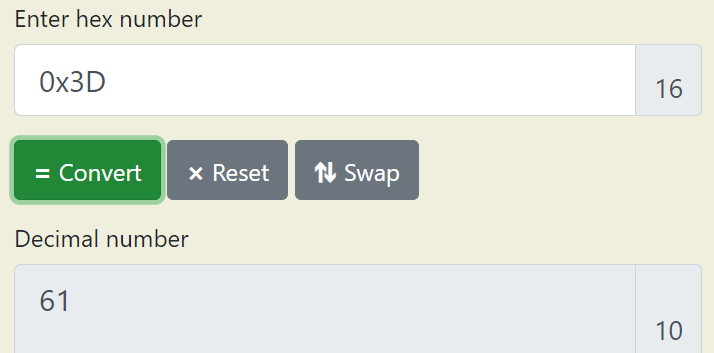


Flag: picoCTF{p}

### 9. Warmed Up

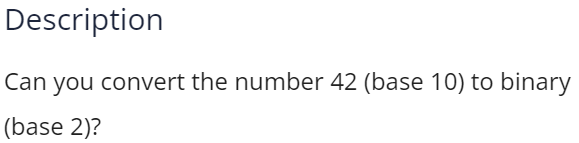


Solution: using online hex to decimal converter

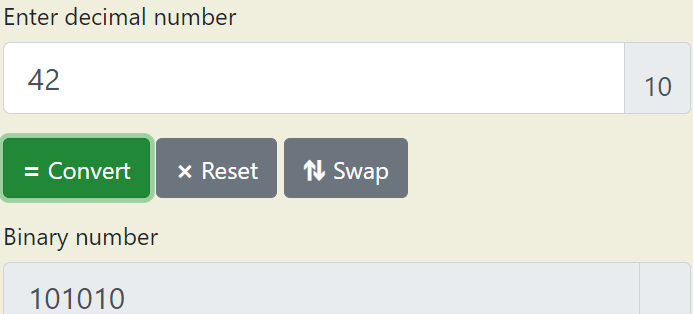


Flag: picoCTF{61}

### 10. 2Warm

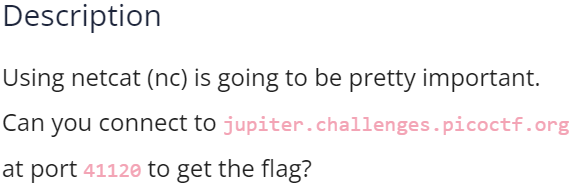


Solution: using online decimal to binary converter

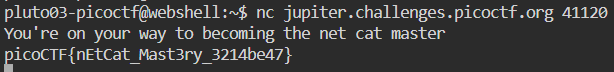


Flag: picoCTF{101010}

### 11. what's a net cat?

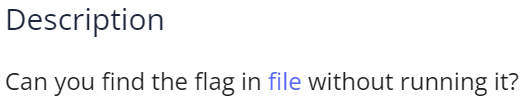


Solution: connect using netcat



Flag: picoCTF{nEtCat\_Mast3ry\_3214be47}

### 12. strings it

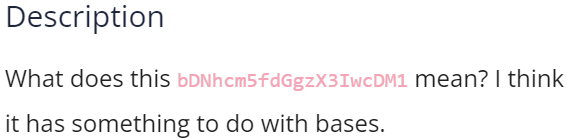


Solution: using strings command

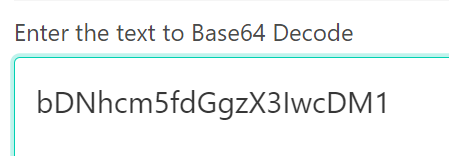
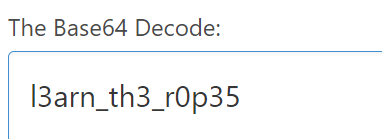


Flag: picoCTF{5tRIng5\_1T\_7f766a23}

### 13. Bases

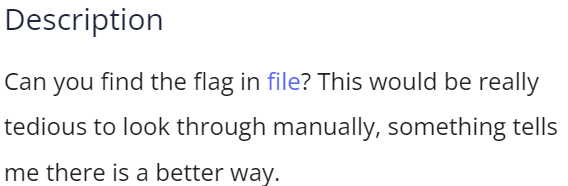


Solution: The data seems to be Base64 encoded. Using online decoder

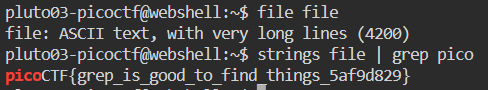
 

Flag: picoCTF{ l3arn\_th3\_r0p35}

### 14. First Grep

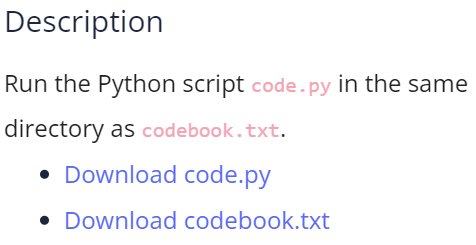


Solution: using grep command



Flag: picoCTF{grep\_is\_good\_to\_find\_things\_5af9d829}

### 15. Codebook

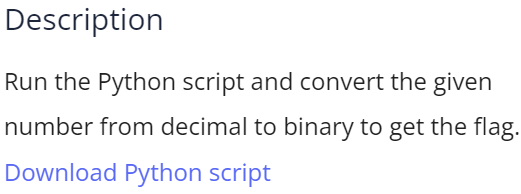


Solution: Download both and run.

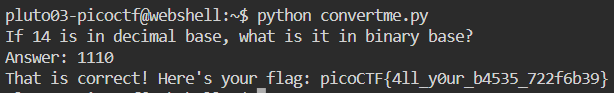


Flag: picoCTF{c0d3b00k\_455157\_d9aa2df2}

### 16. convertme.py

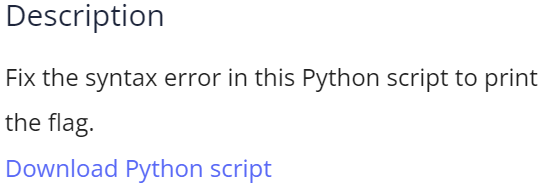


Solution: run the script and answer the question.

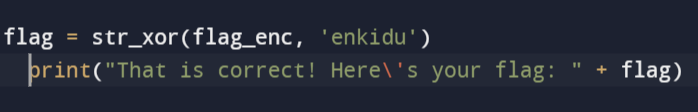


Flag: picoCTF{4ll\_y0ur\_b4535\_722f6b39}

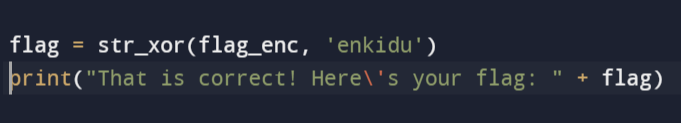
### 17. fixme1.py



Solution: On examining the code we can see the indentation error.



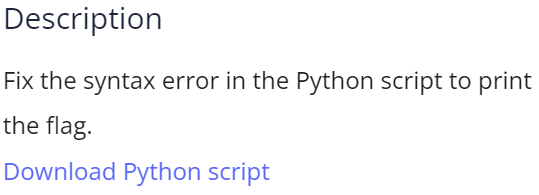
After correction execute the code.



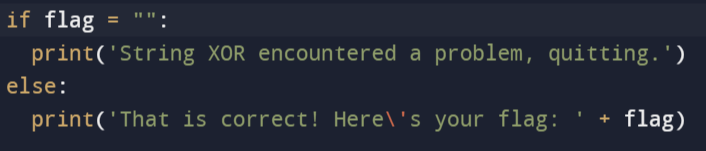


Flag: picoCTF{1nd3nt1ty\_cr1515\_6a476c8f}

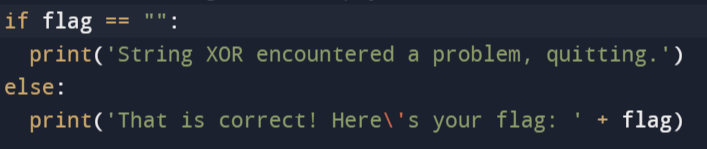
### 18. fixme2.py

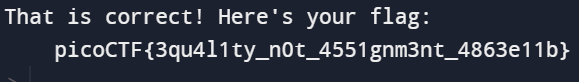


Solution: On examining the code, we can see the use of assignment operator instead of equality operator.



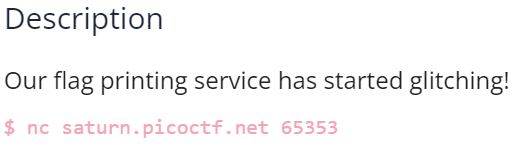
Execute after correction to get the flag.





Flag: picoCTF{3qu4l1ty\_n0t\_4551gnm3nt\_4863e11b}

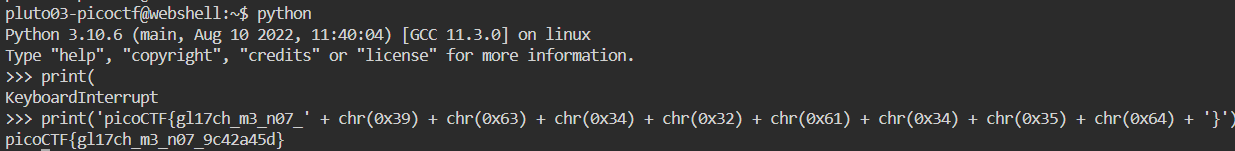
### 19. Glitch Cat



Solution: connect using netcat. We get the flag but some of the letters are in hex code.

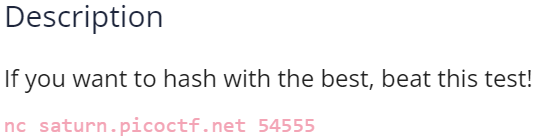


Write a python code to print the flag.

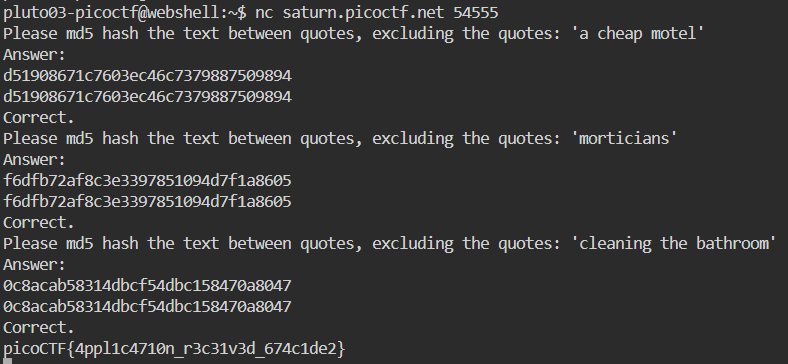


Flag: picoCTF{gl17ch\_m3\_n07\_9c42a45d}

### 20. HashingJobApp

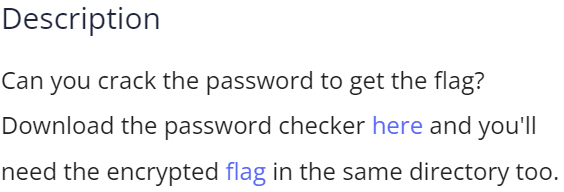


Solution: connect using netcat and answer the questions using online md5 hash generator.



Flag: picoCTF{4ppl1c4710n\_r3c31v3d\_674c1de2}

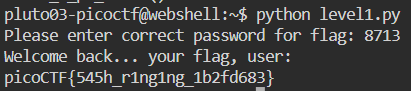
### 21. PW Crack 1



Solution: On inspecting the python code, we get the password.

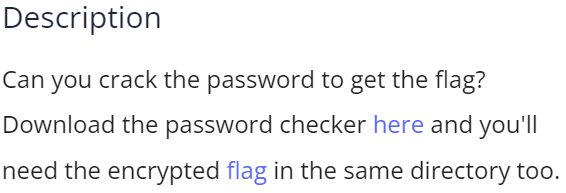


Run the python code using the found password.



Flag: picoCTF{545h\_r1ng1ng\_1b2fd683}

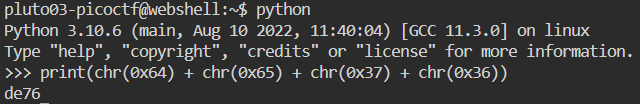
### 22. PW Crack 2



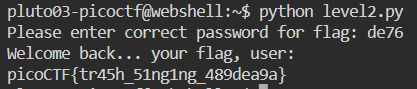
Solution: On inspecting the python code, we find the password in hex code.



Write a code to print it in words.

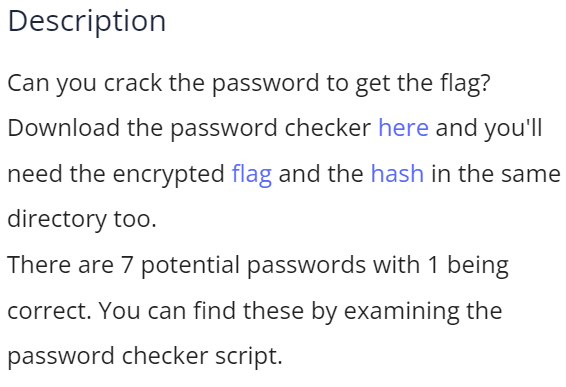


Now using the obtained password, execute the python code.

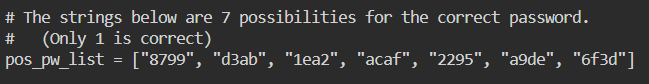


Flag: picoCTF{tr45h\_51ng1ng\_489dea9a}

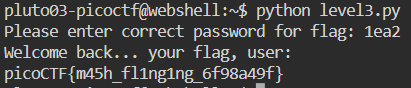
### 23. PW Crack 3



Solution: On inspecting the password checker code, we find 7 possible values.



Run the code and try the 7 passwords. 1 of them will give the flag



Flag: picoCTF{m45h\_fl1ng1ng\_6f98a49f}

### 24. runme.py

### 

### Solution: Download and run the python code

### 

### Flag: picoCTF{run\_s4n1ty\_run}

### 25. Serpentine

### 

### Solution: On inspecting the code we find that choice b does not print the flag.

### 

### Now change the code and execute it.

### 

### 

### Flag: picoCTF{7h3\_r04d\_l355\_7r4v3l3d\_8e47d128}

### 26.First Find

### 

### Solution: unzip the file. The require file is visible. Open it.

### 

### Flag: picoCTF{f1nd\_15\_f457\_ab443fd1}

### 27. Big Zip

### 

### Solution: unzip the file and grep the flag

### 

### Flag: picoCTF{gr3p\_15\_m4g1c\_ef8790dc}

### 28. Based

### 

### Solution: connect with netcat and answer the questions using a converter.

### 

### Flag: picoCTF{learning\_about\_converting\_values\_b375bb16}

### 29. plumbing

### 

### Solution: connect using netcat. Save the contents of this in another file. Search that file for the flag.

### 

### Flag: picoCTF{digital\_plumb3r\_5ea1fbd7}

### 30. mus1c

### 

### Solution: using online rockstar interpreter we get a sequence of ASCII numbers

### 

### Using online ASCII to text converter

### 

### Flag: picoCTF{rrrocknrn0113r}