PICOCTF CHALLENGES

1. ***Challenge: "Let's Warm Up"***

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

Solution:
The problem is just about hexadecimal to decimal conversion. Here we need to convert 0X70 to decimal and then return ASCII character then wrap the solution in picoCTF{} and submit as a flag.
Conversion: Hexadecimal=70
Decimal value = $7*16^1 + 0*16^0 = 112$
ASCII value of 112 -> p

Results and Flag:
The flag obtained is: picoCTF{p}.

2. ***Challenge: "2Warm"***

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

Solution:
The Python function "bin" is built into Python3 and is the quickest way to convert decimal to binary without creating our own function.

Results and Flag:
The flag obtained is: picoCTF{101010}.

3. ***Challenge: "WarmedUp"***

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

Solution:
The hexadecimal number 0x3d is equal to 61 in decimal (base 10)..

Results and Flag:
The flag obtained is: picoCTF{61}.

4. ***Challenge: "Obedient Cat"***

Challenge Description:
This file has a flag in plain sight (aka "in-the-clear"). Download flag.

Solution:
As the description tells us, the flag is present in the file provided to us. We simply need to open the file and get the flag. To do this, we'll download the given file to our console and then use `cat` to open the file and get our flag.

Results and Flag:
The flag obtained is: picoCTF{s4n1ty_v3r1f13d_b5aeb3dd}.

5. ***Challenge: "Wave a Flag"***

Challenge Description:
Can you invoke help flags for a tool or binary? This program has extraordinarily helpful information…

Solution:
We'll download the provided file using wget. To run the file warm we'll need to first make it into an executable file. To do that, we'll be using the chmod +x warm command. Once the executable file is created we'll execute the file using ./warm.

Results and Flag:
The flag obtained is: picoCTF{b1scu1ts_4nd_gr4vy_755f3544}.

6. ***Challenge: "convertme.py"***

Challenge Description:
Run the Python script and convert the given number from decimal to binary to get the flag. Download Python script

Solution:
I download the file on my terminal with :wget, after that I execute the file.Then I converted the file by searching on google 81 to binary and I got the answers then the flag .

Results and Flag:
The flag obtained is: picoCTF{4ll_y0ur_b4535_9c3b7d4d}

7. ***Challenge: "what's a netcat?"***

Challenge Description:
Using netcat (nc) is going to be pretty important. Can you connect to jupiter.challenges.picoctf.org at port 25103 to get the flag?

Solution:
According to the description, all I had to do was to connect to the given nc using the port given to access the flag.

Results and Flag:
The flag obtained is: picoCTF{nEtCat_Mast3ry_d0c64587}

8. ***Challenge: "nice netcat?"***

Challenge Description:
There is a nice program that you can talk to by using this command in a shell: $ nc mercury.picoctf.net 49039, but it doesn't speak English…

Solution:
Connecting to the port given with the netcat command gives us an output of a set of several integers. Shell command: **nc mercury**.picoctf.net 49039

Results and Flag:
The flag obtained is: picoCTF{g00d_k1tty!_n1c3_k1tty!_9b3b7392}

9. ***Challenge: "Tab,Tab,Attack"***

Challenge Description:
Using tab complete in the Terminal will add years to your life, esp. when dealing with long rambling directory structures and filenames: Addadshashanammu.zip

Solution:
The challenge gives us a zip file to download which we download and unzip with the command.Assuming that the flag will be in the last directory we can use the tab autocomplete feature of the linux terminal to change our directory to the last one in the unzipped set of folders. Simply typing "cd A" and then pressing the tab key 7 times, the folder names will be autocomplete and we will end up in the "Ularradallaku" directory. This is because there is only one directory within each directory in the tree so there is only one possible option that can be autocomplete. After we change our current working directory to the last one we end up with a file named "fang-of-haynekhtnamet". Running the file command we see that it is a 64-bit ELF executable file and we execute it using "./" in the linux terminal to get the flag.

Results and Flag:
The flag obtained is: picoCTF{l3v3l_up!_t4k3_4_r35t!_76266e38}

10. ***Challenge: "Python Wrangling"***

Challenge Description:
Python scripts are invoked kind of like programs in the Terminal... Can you run this Python script using this password to get the flag?

Solution:
Running the script by itself we get the following usage:python3.8 ende.py
Usage: ende.py (-e/-d) [file]
So then I tried running it with the file:python3.8 ende.py -d flag.txt.en
Please enter the password:
I didn't feel like manually typing the password so I used cat and a pipe:cat pw.txt | python3.8 ende.py -d flag.txt.en

Results and Flag:
The flag obtained is: picoCTF{4p0110_1n_7h3_h0us3_aa821c16}

11. *Challenge: "Magikarp Ground Mission"*

Challenge Description:
Do you know how to move between directories and read files in the shell? Start the container, `ssh` to it, and then `ls` once connected to begin. Login via `ssh` as `ctf-player` with the password, `6dee9772`Additional details will be available after launching your challenge instance.

Solution:
The challenge gives us the option to start an instance by clicking "Start Instance" which gives us an ssh port to connect to as well as a user and a password for that user to use when connecting to the given port. Starting the instance and then sshing to it with the ssh command on linux gives us a bash shell to use to find the flag.

Results and Flag:
The flag obtained is: picoCTF{xxsh_0ut_0f_\/\/4t3r_c1754242}

12. *Challenge: "First Grep"*

Challenge Description:
Can you find the flag in file? This would be really tedious to look through manually, something tells me there is a better way.

Solution:
When we simply cat this file in the shell to find the flag, we are met with a lot of random noise. Here,we can use a grep command. What grep does is it filters for a specific expression in a plain-text. We know that picoCTF flags are all in the format of picoCTF{...}, so we can grep for the expression picoCTF. Specifically, we would do cat file | grep picoCTF.

Results and Flag:
The flag obtained is: picoCTF{grep_is_good_to_find_things_bf6aec61}

13. *Challenge: "First Find"*

Challenge Description:
Unzip this archive and find the file named 'uber-secret.txt' Download zip file

Solution:
Make the search case-insensitive with the "-i" option:grep -i 'searched-word' filename Search recursively in all files in a given directory with the "-r" option: grep -ir 'searched-word' '/directory'.Search whole words only with the "-w" option: grep -irw 'searched-word' '/directory'.Print the line numbers in which the searched word was found with the "-n" option:grep -irwn 'searched-word' '/directory'.Search for multiple words syntax:grep -ir 'word1|word2|word3' '/directory'But still there was no file named uber-secret.txt. As I was searching hidden files I found something. In a walkthrough I saw another approach, the previous one was the bruteforce one only.Use "locate" tool to search, we can install it using : "sudo apt install locate","/root/Downloads/PicoCTF/files/adequate_books/more_books/.secret/deeper_secrets/deepest_secrets/uber-secret.txt"

Results and Flag:
The flag obtained is: picoCTF{f1nd_15_f457_ab443fd1}

14. *Challenge: "Big Zip"*

Challenge Description:
Unzip this archive and find the flag.Download zip file

Solution:
To download and unzip the archive we will use wget a tool for downloading files online. Here is what I typed in my shell.$ wget https://artifacts.picoctf.net/c/504/big-zip-files.zip $ unzip big-zip-files. Now that we have downloaded and unzips the file we can go in and do some exploring using "ls" and "cd". I find that the "-r" parameter tells grep to search all the files in a directory. I run the following grep command. I'm telling grep to search all the files in "big-zip-files" for the pattern "pico" since the flags are structured as picoCTF{flag}

Results and Flag:
The flag obtained is: picoCTF{gr3p_15_m4g1c_ef8790dc}

15. *Challenge: "Static ain't always noise"*

Challenge Description:
Can you look at the data in this binary: static? This BASH script might help!

Solution:
The challenge gives us a link to download a shell script (ltdis.sh) and a binary executable file (static). we see that it performs disassembly of a binary by running the "objdump -d" and "strings" commands on that binary.Assuming that the flag may have been a static string in the binary we look at the static.ltdis.strings.txt and get the flag

Results and Flag:
The flag obtained is: picoCTF{d15a5m_t34s3r_f6c48608}

16. *Challenge: "Strings it"*

Challenge Description:
Can you find the flag in file without running it?

Solution:
First, I checked the Manual page for the Strings. I found it working and it is as follows : “strings <file_name>”Then a list of strings appeared that were too large to read one by one. I just managed to read the string I want by using the grep command : “strings strings | grep picoCTF”

Results and Flag:
The flag obtained is: picoCTF{5tRIng5_1T_827aee91}

17. *Challenge: "Plumbing"*

Challenge 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 14291.

Solution:
Opening the document using gedit and using find option and there you go flag is here.

Results and Flag:
The flag obtained is: picoCTF{digital_plumb3r_5ea1fbd7}

18. *Challenge: "Super SSH"*

Challenge Description:
Using a Secure Shell (SSH) is going to be pretty important. Additional details will be available after launching your challenge instance.

Solution:
I typed in the command : ssh -p 60525 ctf-player@titan.picoctf.net, and after connecting to the Host I got the flag.

Results and Flag:
The flag obtained is: picoCTF{s3cur3_c0nn3ct10n_65a7a106}

19. *Challenge: "Mod26"*

Challenge Description:
Cryptography can be easy, do you know what ROT13 is?
cvpbPGS{arkg_gvzr_V'yy_gel_2_ebhaqf_bs_ebg13_uJdSftmh}

Solution:
ROT13 is a cipher that rotates each character 13 letters over. The mod 26 is a hint about looping back around. I used an online decoder.

Results and Flag:
The flag obtained is: picoCTF{next_time_I'll_try_2_rounds_of_rot13_ZNMldSDw}

20. ***Challenge: "Bases"***

Challenge Description:

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

Solution:

I used dcode.fr/base-36-cipher to crack it but found a statement : 0604345231964440583950460607 5

Results and Flag:

The flag obtained is: picoCTF{l3arn_th3_r0p35}

21. ***Challenge: "insp3ctor"***

Challenge Description:

Kishor Balan tipped us off that the following code may need inspection: https://jupiter.challenges.picoctf.org/problem/44924/ (link) or http://jupiter.challenges.picoctf.org:44924

Solution:

Using a browser's developer tools, we can see the source code of the site. On Chrome, this is the Inspect Element option, which can be found by right clicking or Ctrl+Shift+C on Windows, Cmd+Shift+C on Linux/MacOS. We then go to the Source tab and view the index.html, mycss.css, and myjs.js files, each containing a part of the flag.

Results and Flag:

The flag obtained is: picoCTF{tru3_d3t3ct1ve_0r_ju5t_lucky?832b0699}

22. ***Challenge: "Enhance"***

Challenge Description:

Download this image file and find the flag. Download image file

Solution:

As we analyze the code and directly to see at the last CTF is given in code and merge all the valid lines.

Results and Flag:

The flag obtained is: picoCTF{3nh4nc3d_24374675}

23. Challenge: "interencdec"

Challenge Description:
Can you get the real meaning from this file.Download the file here.

Solution:
If we open the file, we get the following string: YidkM0JxZGtwQlRYdHFhR3g2YUhsZmF6TnFlVGwzWVROclgyMHdNakV5TnpVNGZRPT0nCg== This looks like a base64 string, so let's try and decode it. the result would be this:b'd3BqdkpBTXtqaGx6aHlfazNqeTl3YTNrX20wMjEyNzU4fQ==' However, the result still looks like a base64 encoded string. It seems like the data might have been encoded multiple times. so we decode it again. Note that I removed **b' '** from the string and decoded the text in the middle:wpjvJAM{jhlzhy_k3jy9wa3k_m0212758

Results and Flag:
The flag obtained is: picoCTF{caesar_d3cr9pt3d_f0212758}

24. Challenge: "Glory of Garden"

Challenge Description:
This garden contains more than it seems.

Solution:
So I download the file and it's just a photo of a garden… the hint talks about a hex editor so I open one in the browser, and load the image to it.

Results and Flag:
The flag obtained is: picoCTF{more_than_m33ts_the_3y3f20F5be9}

25. *Challenge: "Sleuthkit Intro"*

Challenge Description:
Download the disk image and use mmls on it to find the size of the Linux partition. Connect to the remote checker service to check your answer and get the flag. Note: if you are using the webshell, download and extract the disk image into /tmp not your home directory. Download disk image Additional details will be available after launching your challenge instance.

Solution:
Download disk image. Access checker program: nc saturn.picoctf.net 52279

Results and Flag:
The flag obtained is: picoCTF{mm15_f7w!}

26. *Challenge: "Disk, disk, Sleuth"*

Challenge Description:
Use `srch_strings` from the sleuthkit and some terminal-fu to find a flag in this disk image: dds1-alpine.flag.img.gz

Solution:
After downloading the image, I used gunzip to unzip it and then ran srch_strings and used grep command

Results and Flag:
The flag obtained is: picoCTF{f0r3ns1c4t0r_n30phyt3_267e38f6}

27. *Challenge: "Disk, disk, Sleuth ll"*

Challenge Description:
All we know is the file with the flag is named `down-at-the-bottom.txt`... Disk image: dds2-alpine.flag.img.gz

Solution:
Using the TSK Tool Overview website we can find that the fls command can list all files in a directory. We specify the -r, which means recursive so it will scan the entire disk image, and -p, so it prints the full path, flags.

Results and Flag:
The flag obtained is: picoCTF{f0r3ns1c4t0r_n0v1c3_0ba8d02d}

28. *Challenge: "St3go"*

Challenge Description:
Download this image and find the flag.Download image

Solution:
The challenge name suggests that the flag has been hidden in the image via steganography. Furthermore, St3g0 is almost exactly $t3g0, which is a commonly used delimetter for lsb encoding. We can make the educated guess that it is LSB encoded, and use a python script we find online to decode.

Results and Flag:
The flag obtained is: picoCTF{7h3r3_15_n0_5p00n_87ef5b0b}

29. *Challenge: "What lies Within"*

Challenge Description:
There's something in the building. Can you retrieve the flag?

Solution:
So as the hint says, there must be a decoder online, I search “image decoder online” of course…. And I found this site. Choose the decode tab and upload the image, and press “Decode”.

Results and Flag:
The flag obtained is: picoCTF{h1d1ng_1n_th3_b1t5}

30. *Challenge: "Wireshark doo dooo do doo.."*

Challenge Description:
Can you find the flag? Shark1.pcapng

Solution:
Wireshark is a network analyzer tool.Here we are given a file which has a dump of data captured over the network. From this dump we need to find something useful and in this case the flag. After a bit of searching around, using filters etc tried to follow TCP stream and tried finding something useful and after a bit of scrolling then in stream 5 we have something useful finally.The last line is something kind of like the flag but its encrypted version.Trying it in https://www.dcode.fr/cipher-identifier, it comes out to be ROT13 cipher and on decrypting it we have our flag.

Results and Flag:
The flag obtained is: picoCTF{p33kab00_1_s33_u_deadbeef}

31. *Challenge: "PW Crack1"*

Challenge Description:
Can you crack the password to get the flag? Download the password checker here and you'll need the encrypted flag in the same directory too.

Solution:
:I just saw the Python file and found the password that was to be entered on running the file.Then just type in the password and you will get the flag.

Results and Flag:
The flag obtained is: picoCTF{545h_r1ng1ng_56891419}

32. *Challenge: "PW Crack2"*

Challenge Description:
Can you crack the password to get the flag? Download the password checker here and you'll need the encrypted flag in the same directory too.

Solution:
Now here when I opened the code, I found some ASCIIs codes to get the password. I used the Table of ASCII to get the values and the password was "4ec9". Enter the password and you will get the flag

Results and Flag:
The flag obtained is: picoCTF{tr45h_51ng1ng_9701e681}

33. *Challenge: "PW Crack3"*

Challenge Description:
Can you crack the password to get the flag? Download the password checker here and you'll need the encrypted flag and the hash in the same directory too. There are 7 potential passwords with 1 being correct. You can find these by examining the password checker script.

Solution:
I saw the code and found :pos_pw_list = ["f09e", "4dcf", "87ab", "dba8", "752e", "3961", "f159"] I used every password and entered into the execution and got the flag.

Results and Flag:
The flag obtained is: picoCTF{m45h_fl1ng1ng_cd6ed2eb}

34. *Challenge: "PW Crack4"*

Challenge Description:
Can you crack the password to get the flag?Download the password checker here and you'll need the encrypted flag and the hash 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.

Solution:
I simply used a for loop and iterate over every element in the list and used it as a password and this let me get the password and provided me with the flag.

Results and Flag:
The flag obtained is: picoCTF{fl45h_5pr1ng1ng_d770d48c}

35. *Challenge: "PW Crack5"*

Challenge Description:
Can you crack the password to get the flag? Download the password checker here and you'll need the encrypted flag and the hash in the same directory too. Here's a dictionary with all possible passwords based on the password conventions we've seen so far.

Solution:
I just changed my approach from directly using the "dictionary.txt" and converted it into the list using the ".split()" function.These concepts helped me to build a new approach to it.

Results and Flag:
The flag obtained is: picoCTF{h45h_sl1ng1ng_36e992a6}

36. *Challenge: "Fixme1.py"*

Challenge Description:
Fix the syntax error in this Python script to print the flag. Download Python script

Solution:
The indentation on the last line for print command was wrong. Just delete the whitespaces you are good to go. Just, follow the instructions, by correcting the code and you will get the flag.

Results and Flag:
The flag obtained is: picoCTF{1nd3nt1ty_cr1515_09ee727a}

37. *Challenge: " Mind your P's and Q's"*

Challenge Description:
In RSA, a small e value can be problematic, but what about N? Can you decrypt this? values

Solution:
Use the tool RSADecoder.After installing the RsaCtfTool, get the flag by entering the following command: python3 RsaCtfTool.py -n <n-value> -e <e-value> --uncipher <c-value>

Results and Flag:
The flag obtained is: picoCTF{sma11_N_n0_g0od_23540368}

38. *Challenge: "Fixme2.py"*

Challenge Description:
Fix the syntax error in this Python script to print the flag. Download Python script

Solution:
Use the command, "python3 fixme2.py" to execute the python file. I used hints 1,2,3 to get the flag.

Results and Flag:
The flag obtained is: picoCTF{3qu4l1ty_n0t_4551gnm3nt_f6a5aefc}

39. *Challenge: "Mini RSA"*

Challenge Description:
Do you know how to move between directories and read files in the shell? Start the container, `ssh` to it, and then `ls` once connected to begin. Login via `ssh` as `ctf-player` with the password, `6dee9772`Additional details will be available after launching your challenge instance.

Solution:
So RSA is an asymmetric key encryption method , which has a public key and a private key to encrypt or decrypt any text. The encryption algorithm is as follows, “c = (M ^ e) % N ” , where c = cipher text, M = main text , e = encryption (public) key , N = part of both public and private key.

Results and Flag:
The flag obtained is: picoCTF{e_sh0u1d_b3_lArg3r_a166c1e3}

40. *Challenge: "Where are the robots?"*

Challenge Description:
Can you find the robots? https://jupiter.challenges.picoctf.org/problem/60915/ (link) or http://jupiter.challenges.picoctf.org:60915

Solution:
So I access the robots.txt file by adding /robots.txt to the URL of the site. Now my attention goes to the “Disallow” part of the file. It's an HTML extension which means it should lead to another webpage. I added “/8028f.html” to the site’s URL and voila! Here’s the flag.

Results and Flag:
The flag obtained is: picoCTF{ca1cu1a1ing_Mach1n3s_8028f}