❖ **PicoCTF Challenge**

1. Obedient Cat

- Lab: This file has a flag in plain sight (aka "in-the-clear"). Download flag.
- Solution: picoCTF{s4n1ty_v3r1f13d_b5aeb3dd}
- Exploitation: To do this, we'll use wget to download the given file to our console and then use cat to open the file and get our flag.

2. SuperSSH

- Lab: Using a Secure Shell (SSH) is going to be pretty important.Additional details will be available after launching your challenge instance.
- Solution: picoCTF{s3cur3_c0nn3ct10n_65a7a106}
- Exploitation: Try logging in 'as' someone with <user>@titan.picoctf.net

3. What's a netcat?

- Lab: 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: picoCTF{nEtCat_Mast3ry_b1d25ece}}
- Exploitation: Simply connect to the host and port with netcat to get the flag

4. Mod 26

- Lab: Cryptography can be easy, do you know what ROT13 is? cvpbPGS{arkg_gvzr_V'yy_gel_2_ebhaqf_bs_ebg13_uJdSftmh}
- Solution: picoCTF{next_time_I'll_try_2_rounds_of_rot13_hWqFsgzu}
- Exploitation: ROT13 is a simple letter substitution cipher that replaces a letter with the 13th letter after it in the alphabet.Hence ROT13 may be the cipher used for encryption.

5. Warmed Up

- Lab: What is 0x3D (base 16) in decimal (base 10)?
- Solution: picoCTF{61}
- Exploitation: In Python3, you can simply type the literal hex value with the 0x indicator (0X3D)

6. 2Warm

- Lab: Can you convert the number 42 (base 10) to binary (base 2)?
- Solution: picoCTF{101010}
- Exploitation: The Python function "bin" is built into Python3 and is the quickest way to convert decimal to binary without creating our own function.

7. Bases

- Lab:What does this bDNhcm5fdGgzX3IwcDM1 mean? I think it has something to do with bases.
- Solution: picoCTF{l3arn_th3_r0p35}
- Exploitation: On changing its base form 36 to 10 (DECIMAL) I got this : 42253276001340547821074237951 77.But this came out to be wrong.3AL652A86KJDQHP4M5PV9 → Base 32, 3554C512906A4DBA8E4962E7E9 -> Base 16, 652514242244065115565071113056375 1 → Base 8. At each base it is coming out to be different, and that is obvious but we in the end know it's a BASE 36 hash.

8. Wave a Flag

- Lab:Can you invoke help flags for a tool or binary? This program has extraordinarily helpful information...
- Solution: picoCTF{b1scu1ts_4nd_gr4vy_6635aa47}
- Exploitation: file warm - shows it as an executable file. When we try to run using ./warm It gives errors. One of the reasons might be that it does not have execution permission. To give permission enter - chmod +x warm running again gives Hello user! Pass me a -h to learn what I can do! Finally running - /warm -h returns the flag.

9. Tab, Tab, Attack

- Lab:Using tabcomplete in the Terminal will add years to your life, esp. when dealing with long rambling directory structures and filenames: Addadshashanammu.zip
- Solution: picoCTF{l3v3l_up!_t4k3_4_r35t!_a00cae70}
- Exploitation: On opening the file flag.txt using gedit and searching for picoCTF(We have flag format like picoCTF{} hence we should search for this string) boom we have our flag.

10. Insp3ct0r

- Lab: 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:picoCTF{tru3_d3t3ct1ve_0r_ju5t_lucky?832b0699}
- Exploitation: Access the given URL in browser and capture request/response using Burp Suite tool. A message displayed as “I used these to make this site: HTML, CSS, JS (JavaScript)”.Hence inspect element in the application or open view page source to view the code content and notice “<! — Html is neat. Anyways have 1/3 of the flag: picoCTF{tru3_d3 →” as comment.Now open css file which will reveal 2nd part of the flag value in comment “/* You need CSS to make pretty pages. Here’s part 2/3 of the flag: t3ct1ve_0r_ju5t */”. At last view the content of js file which will reveal the final part of the Flag value /* Javascript sure is neat. Anyways part 3/3 of the flag: _lucky?832b0699} */. After combining all these 3 parts, we got the final Flag value

11. Strings it

- Lab: Can you find the flag in file without running it?
- Solution:picoCTF{5tRIng5_1T_827aee91}
- Exploitation: 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”

12. Enhance

- Lab: Download this image file and find the flag. Download image file
- Solution:picoCTF{3nh4nc3d_24374675}
- Exploitation: If opening this in a normal image viewer is not useful we use a text editor or browser and view the source code of the image. As we analyze the code and directly see at the last CTF is given in code and merge all the valid lines. After completing this we have a flag.

13. interencdec

- Lab: Can you get the real meaning from this file.Download the file here.
- Solution:picoCTF{caesar_d3cr9pt3d_f0212758}
- Exploitation: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} Now this format looks like our flag, so we’re close! Let's try decoding it using Caesar cipher. I used this site which generates all the possible outputs.

14. caesar

- Lab: We found a brand new type of encryption, can you break the secret code? (Wrap with picoCTF{}) apbopjbobpnjpjnmnnnmnlnbamnpnononpnaaaamnlnkapndnkncamnpapncnbannaapncn dnlnpna new_caesar.py
- Solution:picoCTF{et_tu?_1ac5f3d7920a85610afeb2572831daa8}
- Exploitation: Let's reverse the new_caesar.py program.See comments in the solution script for a detailed explanation. We reverse the encoding mechanism, then try the possible offsets, and print the possible flags.

15. information

- Lab:Files can always be changed in a secret way. Can you find the flag? cat.jpg
- Solution:picoCTF{the_m3tadata_1s_modified}
- Exploitation: After downloading it we can run the commands file, hexdump and binwakto actually verify that it is a jpg image.

16. Glory of Garden

- Lab:This garden contains more than it seems.
- Solution:picoCTF{more_than_m33ts_the_3y3f20F5be9}
- Exploitation: 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.

17. Sleuthkit Intro

- Lab: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:picoCTF{mm15_f7w!}
- Exploitation: Download disk image. Access checker program: nc saturn.picoctf.net 52279

18. Nice netcat

- Lab: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: picoCTF{g00d_k1tty!_n1c3_k1tty!_3d84edc8}
- Exploitation: First copying and running the netcat command mentioned in the question.After running the command as an output we get a series of numbers. 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, 51, 100 ,56, 52, 101, 100, 99, 56, 125 ,10.These number could represent anything in this it looks like they are ASCII encoded.Copying them to ASCII to text converter tool.

19. Python Wrangling

- Lab: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: picoCTF{4p0110_1n_7h3_h0us3_67c6cc96}
- Exploitation: Looking at the extension of "flag.txt.en" we can deduce that it is an encoded version of the flag which we need to decode. As the challenge description says we can probably run the "ende.py" python script to decode it. We cat out the password.txt file as it may be necessary and then try to run the python script.

20. Magikarp Ground Mission

- Lab: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: picoCTF{xxsh_0ut_0f_\/\/4t3r_540e4e79}
- Exploitation:After launching the instance and entering the password we get the shell. On entering the ls command, we get,on opening the text file 1of3.flag.txt, we have our first part of the flag.On changing the directory and moving down we find another part of the flag.Now we need just the last part of the flag i.e. 2of3.flag.txt on moving further down, we find our third part

21. First Grep

- Lab:Can you find the flag in the file? This would be really tedious to look through manually, something tells me there is a better way.
- Solution: picoCTF{grep_is_good_to_find_things_bf6aec61}
- Exploitation:You can also find the file in /problems/first-grep_2_04dbf496b78e6c37c0097cdfef734d88 on the shell server.$ grep -oE "picoCTF{.*}" file

22. Where are the Robots?

- Lab:Can you find the robots? https://jupiter.challenges.picoctf.org/problem/60915/ (link) or http://jupiter.challenges.picoctf.org:60915
- Solution: picoCTF{ca1cu1a1ing_Mach1n3s_8028f}
- Exploitation: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.

23. PW crack1

- Lab: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: picoCTF{545h_r1ng1ng_56891419}
- Exploitation: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.

24. PW Crack2

- Lab: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: picoCTF{tr45h_51ng1ng_9701e681}
- Exploitation: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

25. Vault-door training

- Lab:Your mission is to enter Dr. Evil’s laboratory and retrieve the blueprints for his Doomsday Project. The laboratory is protected by a series of locked vault doors. Each door is controlled by a computer and requires a password to open. Unfortunately, our undercover agents have not been able to obtain the secret passwords for the vault doors, but one of our junior agents obtained the source code for each vault’s computer! You will need to read the source code for each level to figure out what the password is for that vault door. As a warmup, we have created a replica vault in our training facility. The source code for the training vault is here: VaultDoorTraining.java
- Solution: picoCTF{w4rm1ng_Up_w1tH_jAv4_3808d338b46}
- Exploitation:On viewing the source code we have, Here we have the password: w4rm1ng_Up_w1tH_jAv4_3808d338b46. Entering just the string won’t give access but wrapping it in picoCTF{w4rm1ng_Up_w1tH_jAv4_3808d338b46} gives access which is the flag.

26. Pw Crack3

- Lab: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: picoCTF{m45h_fl1ng1ng_cd6ed2eb}
- Exploitation: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

27. Pw Crack4

- Lab: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: picoCTF{fl45h_5pr1ng1ng_d770d48c}
- Exploitation: 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.

28. Pw Crack5

- Lab: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: picoCTF{h45h_sl1ng1ng_36e992a6}
- Exploitation: 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.

29. Mind your P's and Q's

- Lab:In RSA, a small e value can be problematic, but what about N? Can you decrypt this? values
- Solution: picoCTF{sma11_N_n0_g0od_23540368}
- Exploitation: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>

30. Mini RSA

- Lab:What happens if you have a small exponent? There is a twist though, we padded the plaintext so that (M ** e) is just barely larger than N. Let's decrypt this: ciphertext
- Solution: picoCTF{e_sh0u1d_b3_lArg3r_a166c1e3}
- Exploitation: 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.So if N>e then (M**e) % N == (M**e) but if N < e then we cannot just root inverse it to decrypt, so I reverse the algorithm and find out a way to get the plain text , M = {(N*i) + c} ^ (1/3) // where , i = any positive integer

31. fixme1.py

- Lab:Fix the syntax error in this Python script to print the flag. Download Python script
- Solution: picoCTF{1nd3nt1ty_cr1515_09ee727a}
- Exploitation: 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.

32. fixme2.py

- Lab:Fix the syntax error in this Python script to print the flag. Download Python script
- Solution: picoCTF{3qu4l1ty_n0t_4551gnm3nt_f6a5aefc}
- Exploitation: Use the command, “python3 fixme2.py” to execute the python file. I used hints 1,2,3 to get the flag.

33. Verify

- Lab:People keep trying to trick my players with imitation flags. I want to make sure they get the real thing! I'm going to provide the SHA-256 hash and a decrypt script to help you know that my flags are legitimate.You can download the challenge files here: challenge.zip
- Solution: picoCTF{trust_but_verify_c6c8b911}
- Exploitation:Check the checksum.txt. From this, we need to check the files in “files” directory to see which one is matched. Use sha256sum to cross check each files
  $ sha256sum files/* | grep "467a10447deb3d4e17634cacc2a68ba6c2bb62a6637dad9145ea673bf0be5e02"c6c8b911 matches the checksum. See the contents of the targeted file $ file files/c6c8b911. Utilize the decrypt script to decrypt the file $ ./decrypt.sh files/c6c8b911

34. Disk, disk, sleuth!

- Lab: Use `srch_strings` from the sleuthkit and some terminal-fu to find a flag in this disk image: dds1-alpine.flag.img.gz
- Solution: picoCTF{f0r3ns1c4t0r_n30phyt3_267e38f6}
- Exploitation: After downloading the image, I used gunzip to unzip it and then ran srch_strings and used grep command

35. Disk, disk, sleuth! II

- Lab: All we know is the file with the flag is named `down-at-the-bottom.txt`... Disk image: dds2-alpine.flag.img.gz
- Solution: picoCTF{f0r3ns1c4t0r_n0v1c3_0ba8d02d}
- Exploitation: 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.

36. Extensions

- Lab: This is a really weird text file TXT? Can you find the flag?
- Solution: picoCTF{now_you_know_about_extensions}
- Exploitation: There is nothing I can see in this junk, tried to Ctrl+F and search for the pico format flag but still nothing. but as we can see, at the first line we can see PNG, so maybe it's just an image? (It's called the header of the file for more reading) In order to open the file as an image, we need to rename the file extension, and now we can open the file as an image.

37. St3go

- Lab: Download this image and find the flag.Download image
- Solution: picoCTF{7h3r3_15_n0_5p00n_87ef5b0b}
- Exploitation: 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.

38. What Lies Within

- Lab:There's something in the building. Can you retrieve the flag?
- Solution: picoCTF{h1d1ng_1n_th3_b1t5}
- Exploitation: 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".

39. Wireshark doo dooo do doo...

- Lab:Can you find the flag? Shark1.pcapng.
- Solution: picoCTF{p33kab00_1_s33_u_deadbeef}
- Exploitation: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.

40. Trivial Flag Transfer Protocol

- Lab:Figure out how they moved the flag.
- Solution: picoCTF{h1dd3n_1n_pLa1n_51GHT_18375919}
- Exploitation:We can see the files were sent in this order: “instructions.txt”, “plan”, “picture1.bmp”, “picture2.bmp”, “picture3.bmp”. Now let’s export all of them to see what they contain by clicking File -> Export Object -> …TFTP.Then we can click Save All → Choose Folder to save → Click OK. Now we have got all the files we need, let’s dig in the same order all the files were sent. The first one is “instruction.txt”.cat ./instructions.txt
  And we get
  GSGCQBRFAGRAPELCGBHEGENSSVPFBJRZHFGQVFTHVFRBHESYNTGENAFSR
  E.SVTHERBHGNJNLGBUVQRGURSYNTNAQVJVYYPURPXONPXSBEGURCYNA
  After doing a Rot13 decode (you can go to https://rot13.com to decode it), we get
  TFTPDOESNTENCRYPTOURTRAFFICSOWEMUSTDISGUISEOURFLAGTRANSFER.
  FIGUREOUTAWAYTOHIDETHEFLAGANDIWILLCHECKBACKFORTHEPLAN
  The Console output wrote extracted data to “flag.txt”. It must be the flag here :). Let’s cat it. cat ./flag.txt. And the console output is:
  picoCTF{h1dd3n_1n_pLa1n_51GHT_18375919}