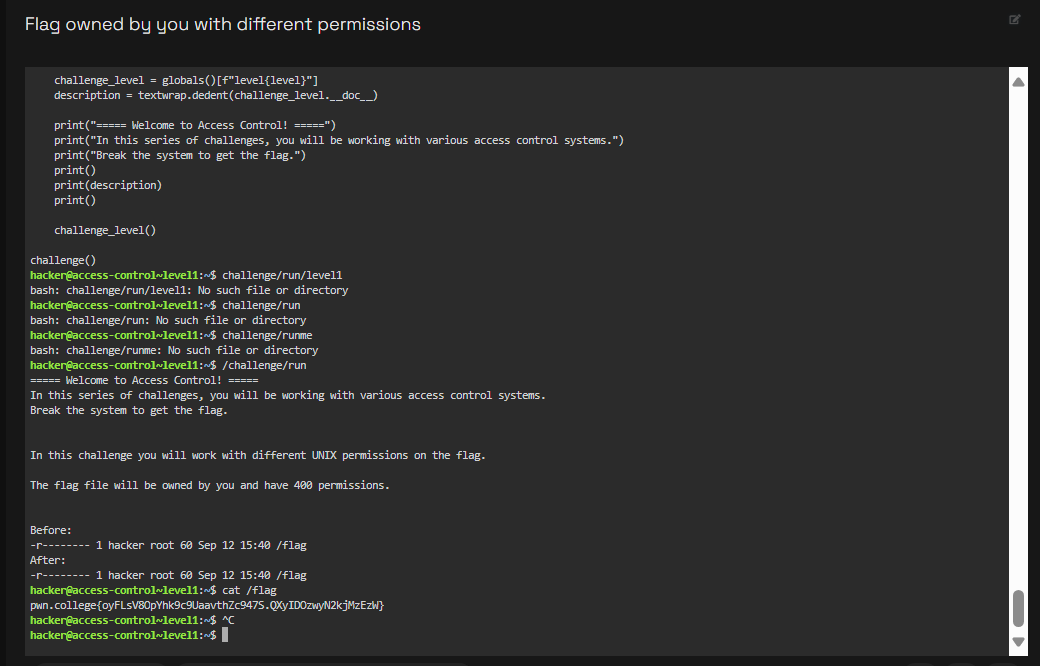
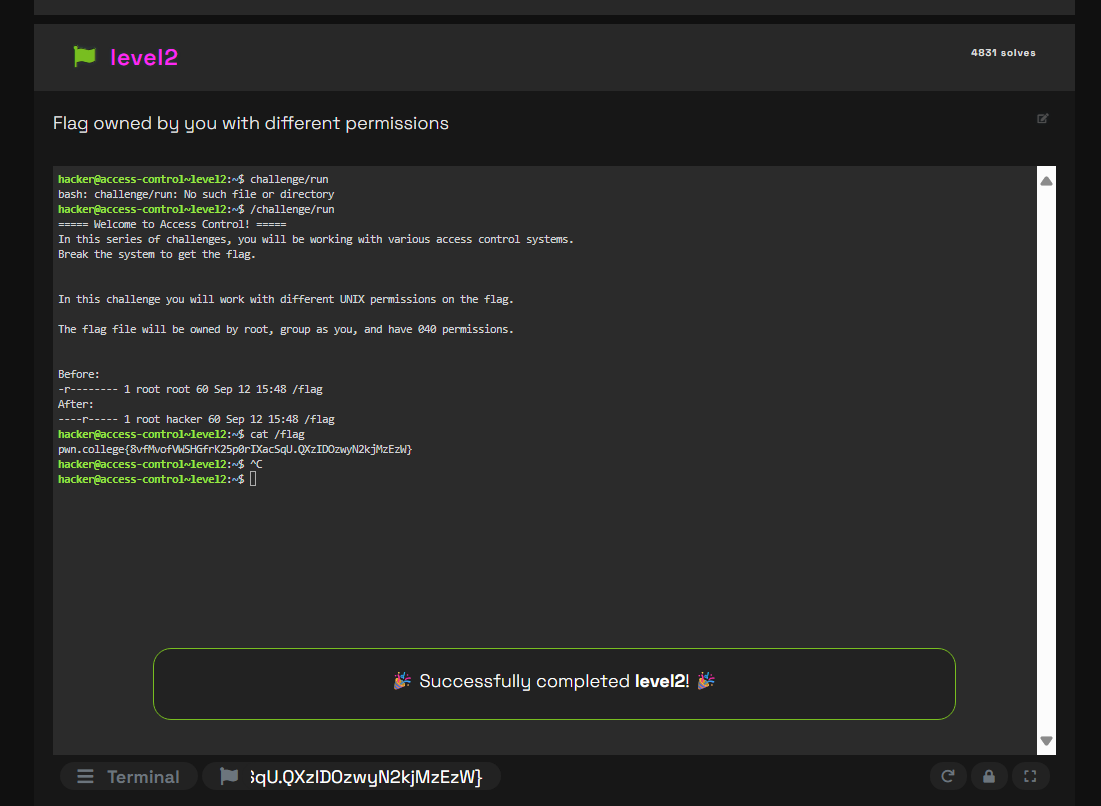
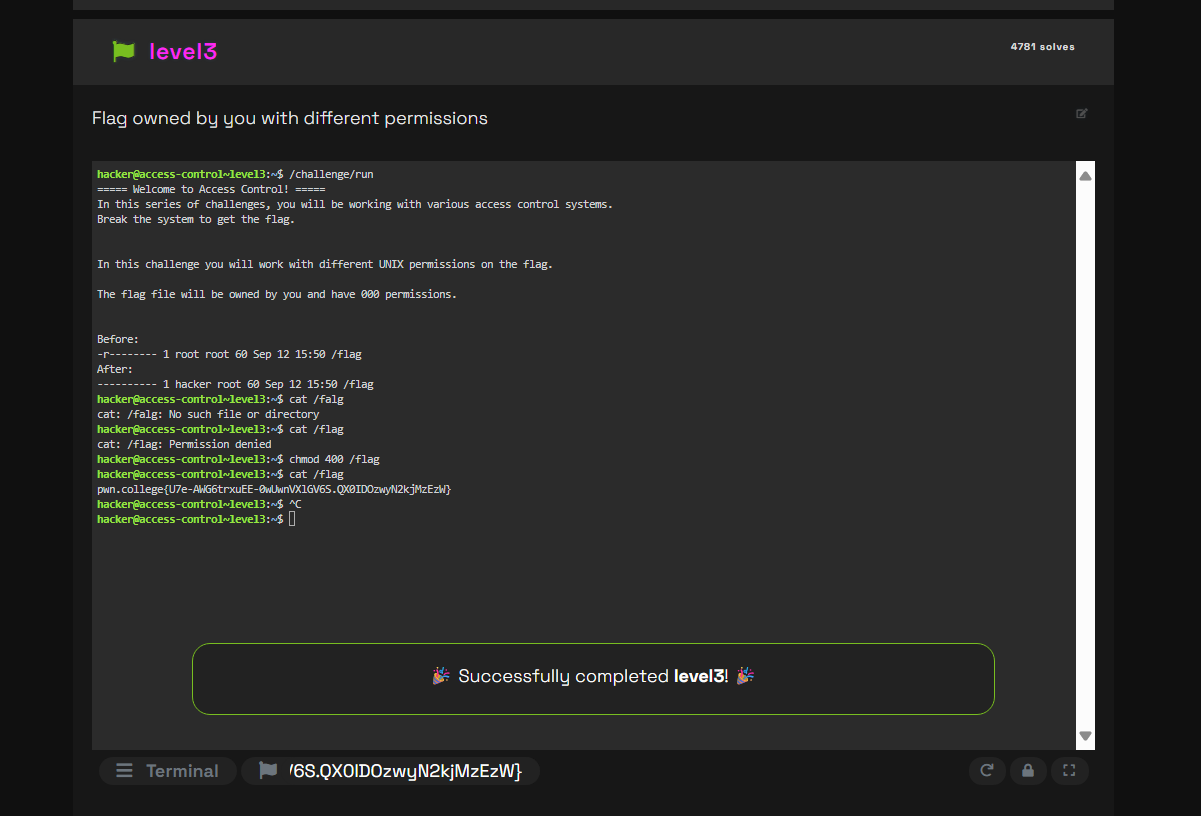
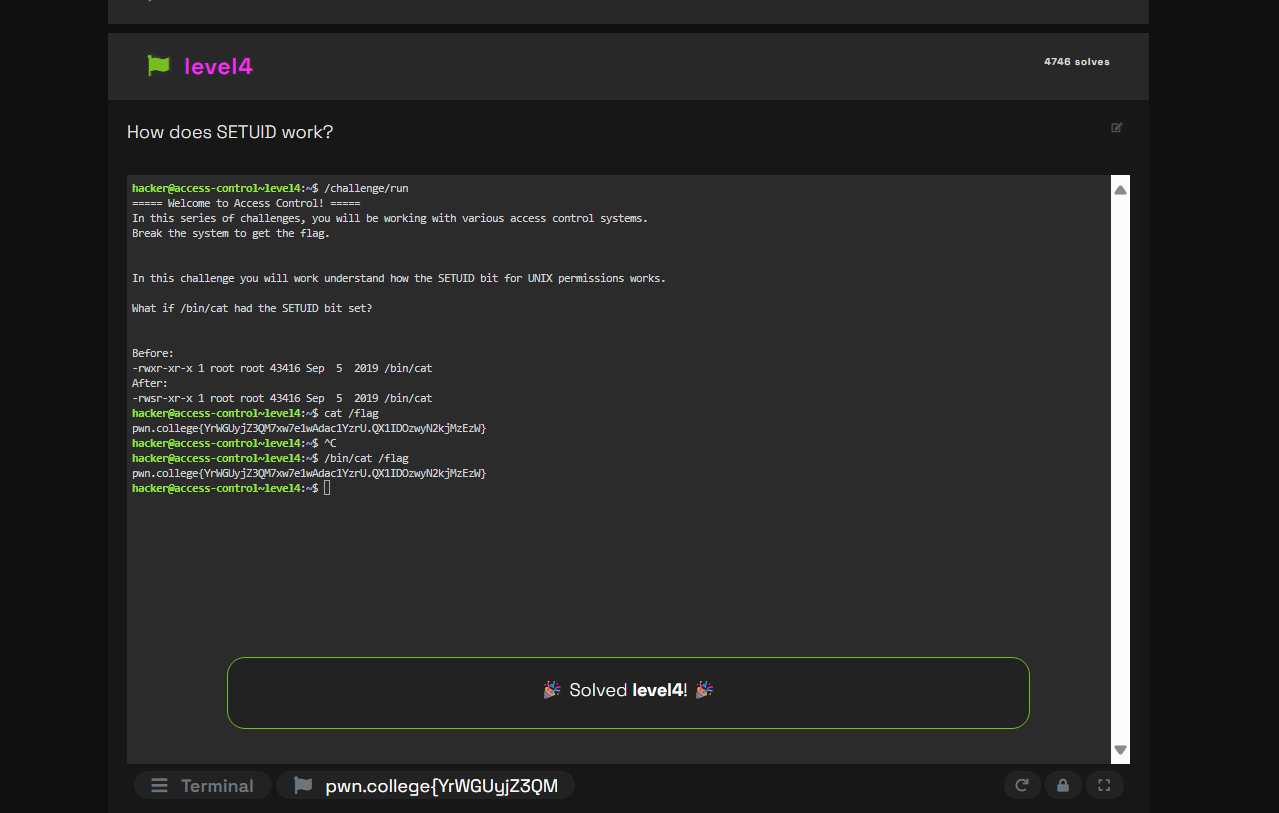
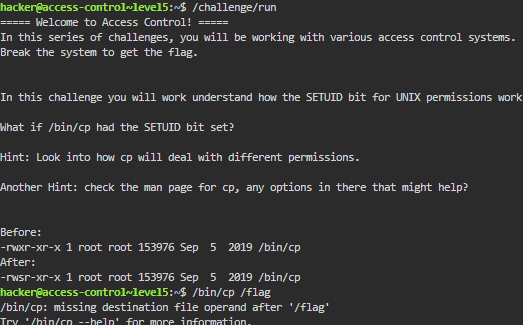
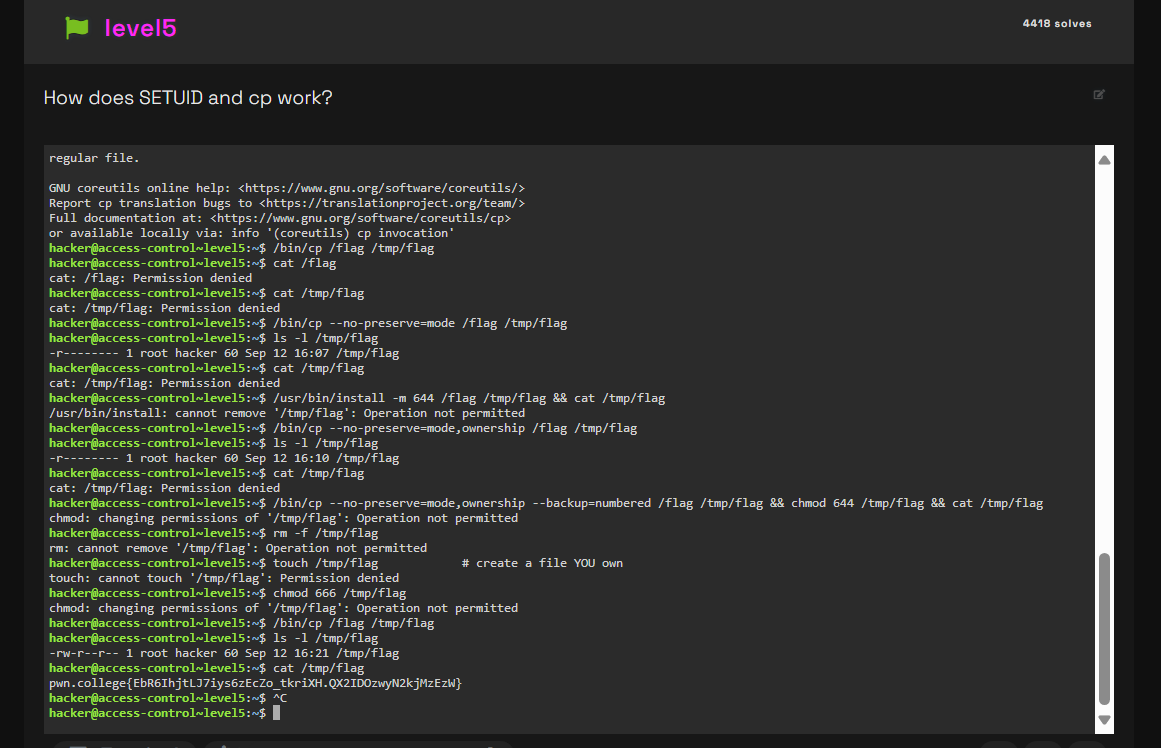
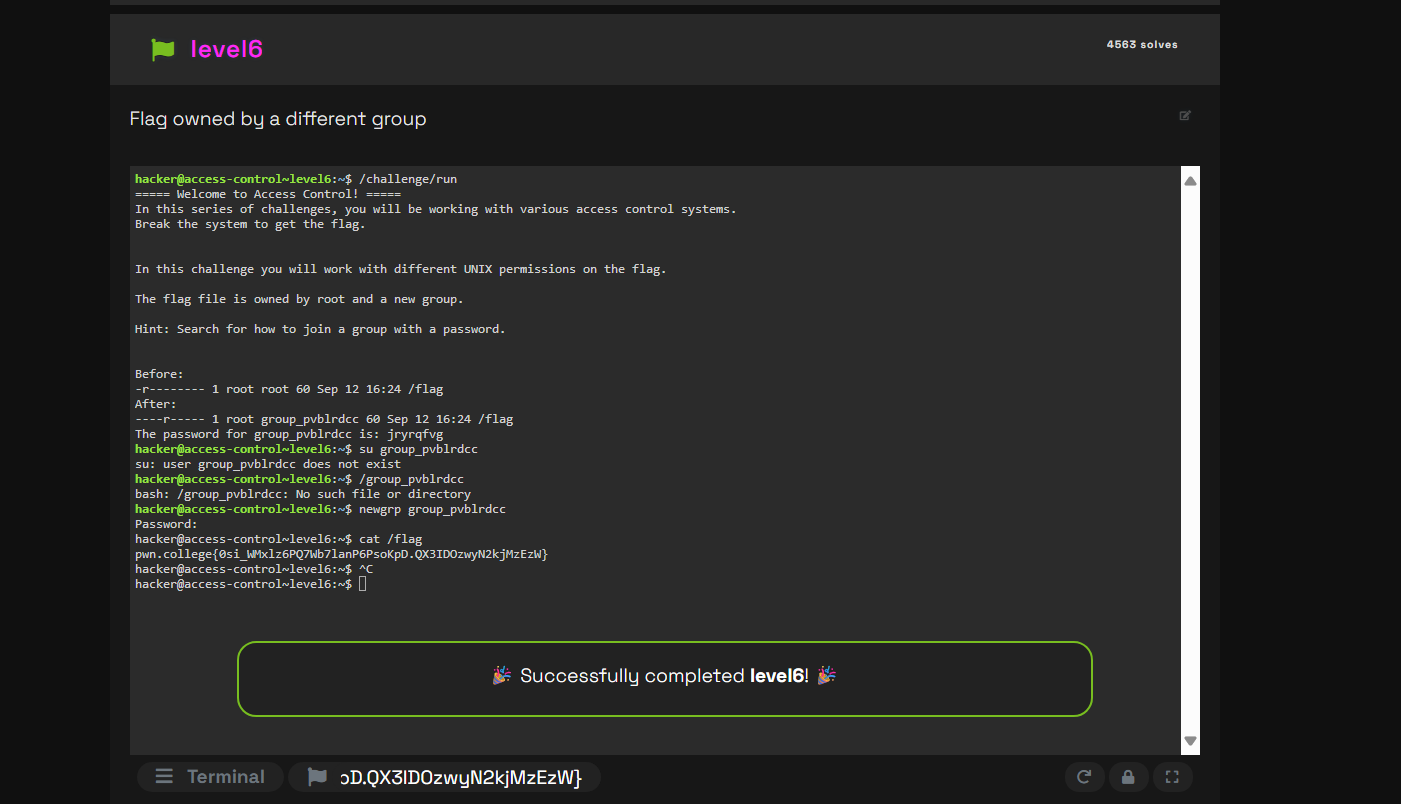
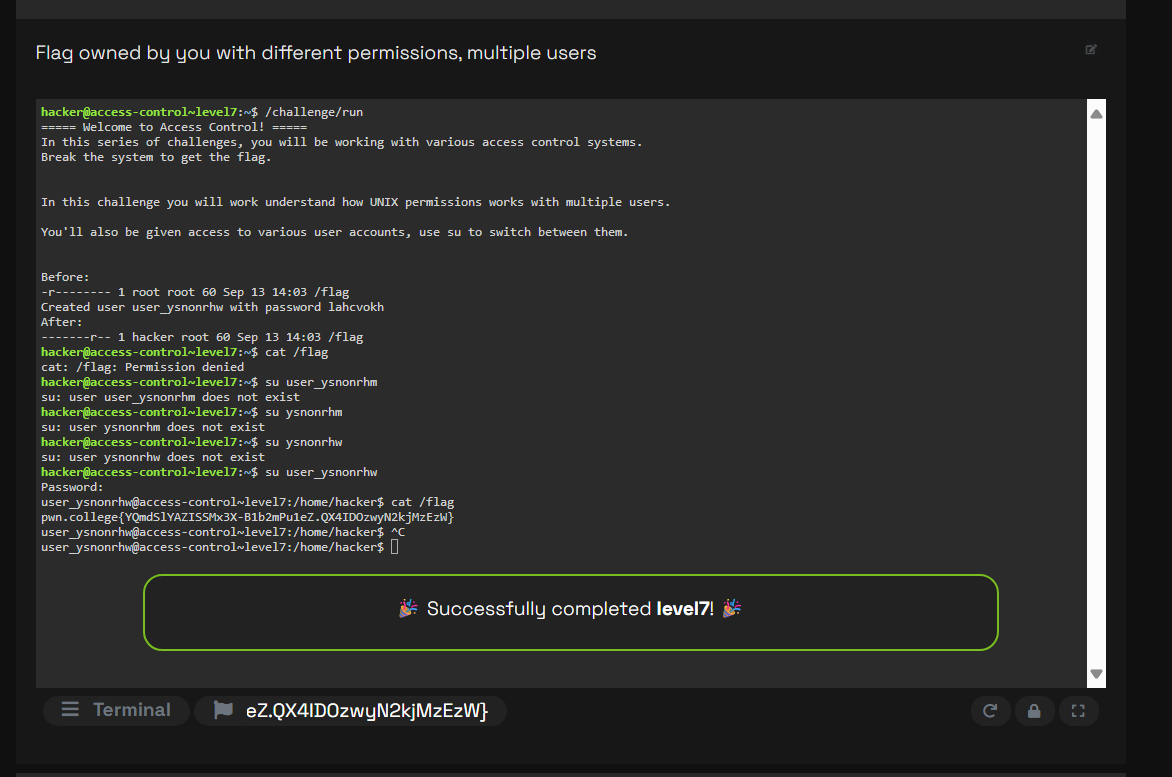
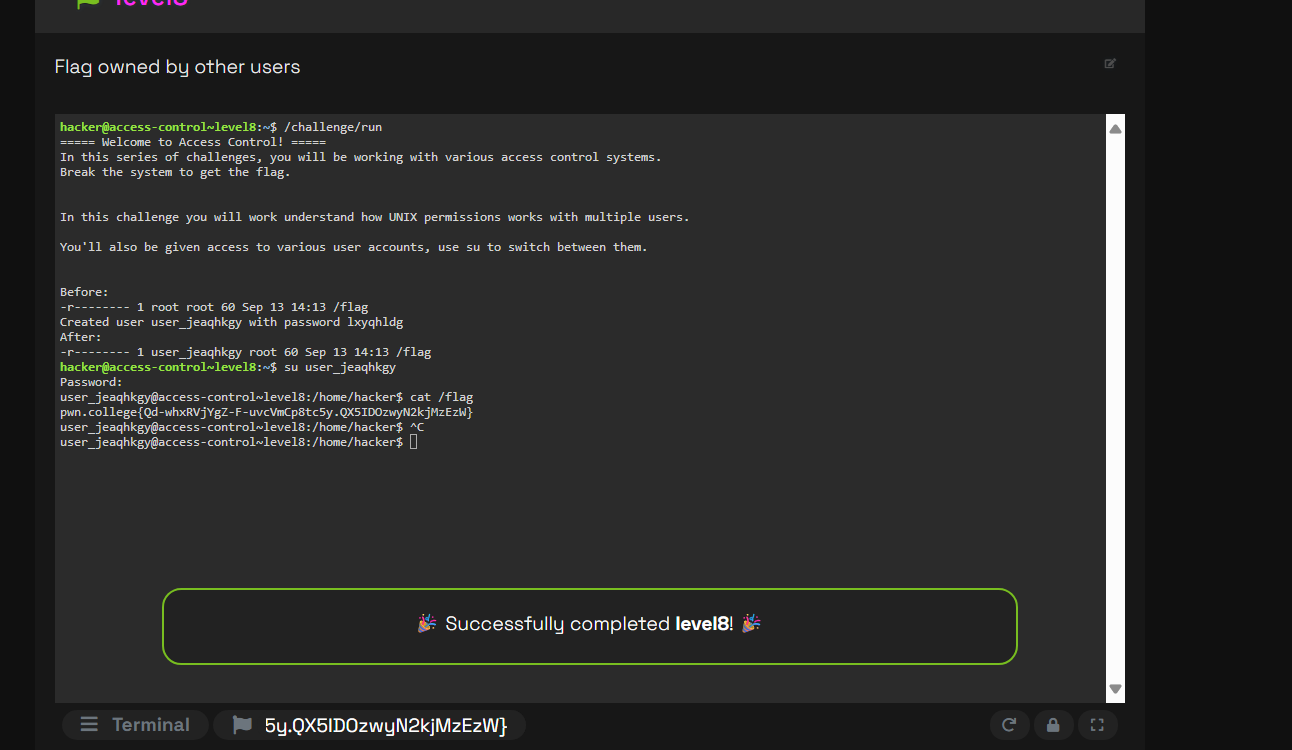
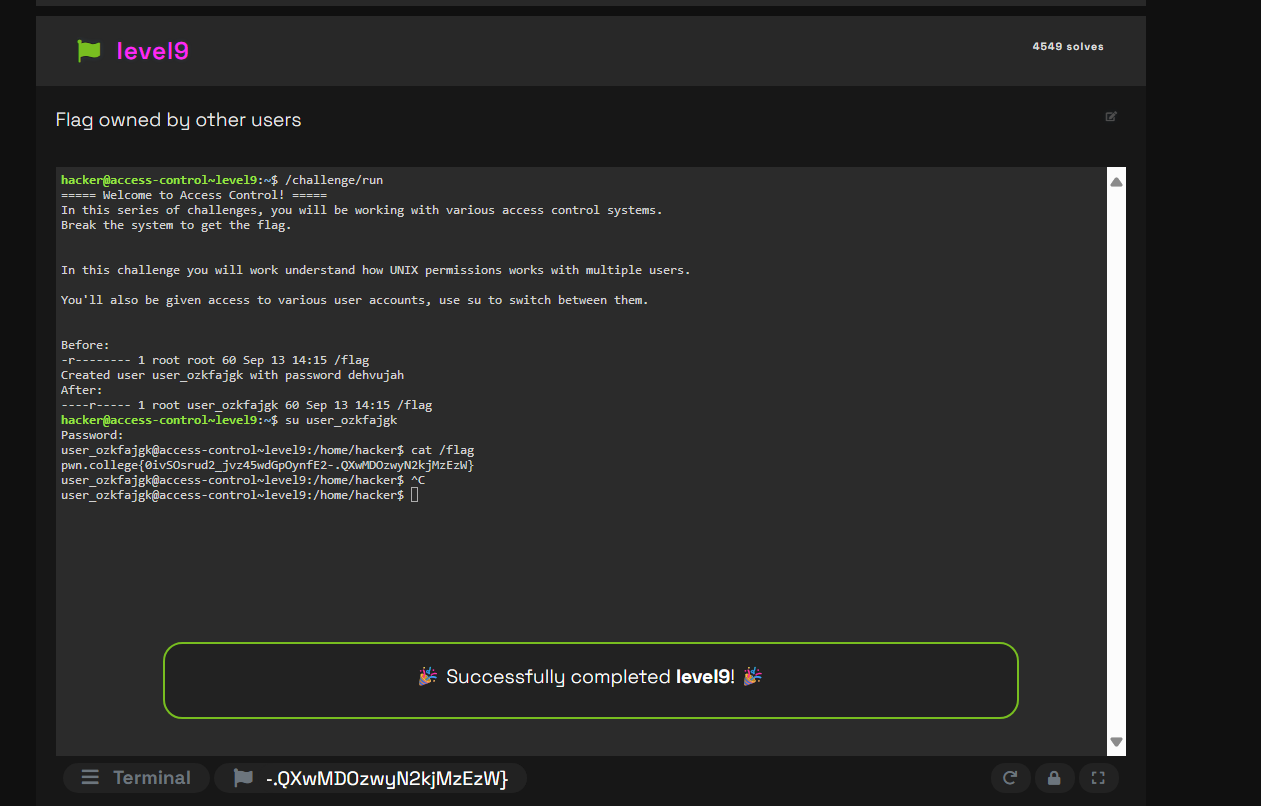
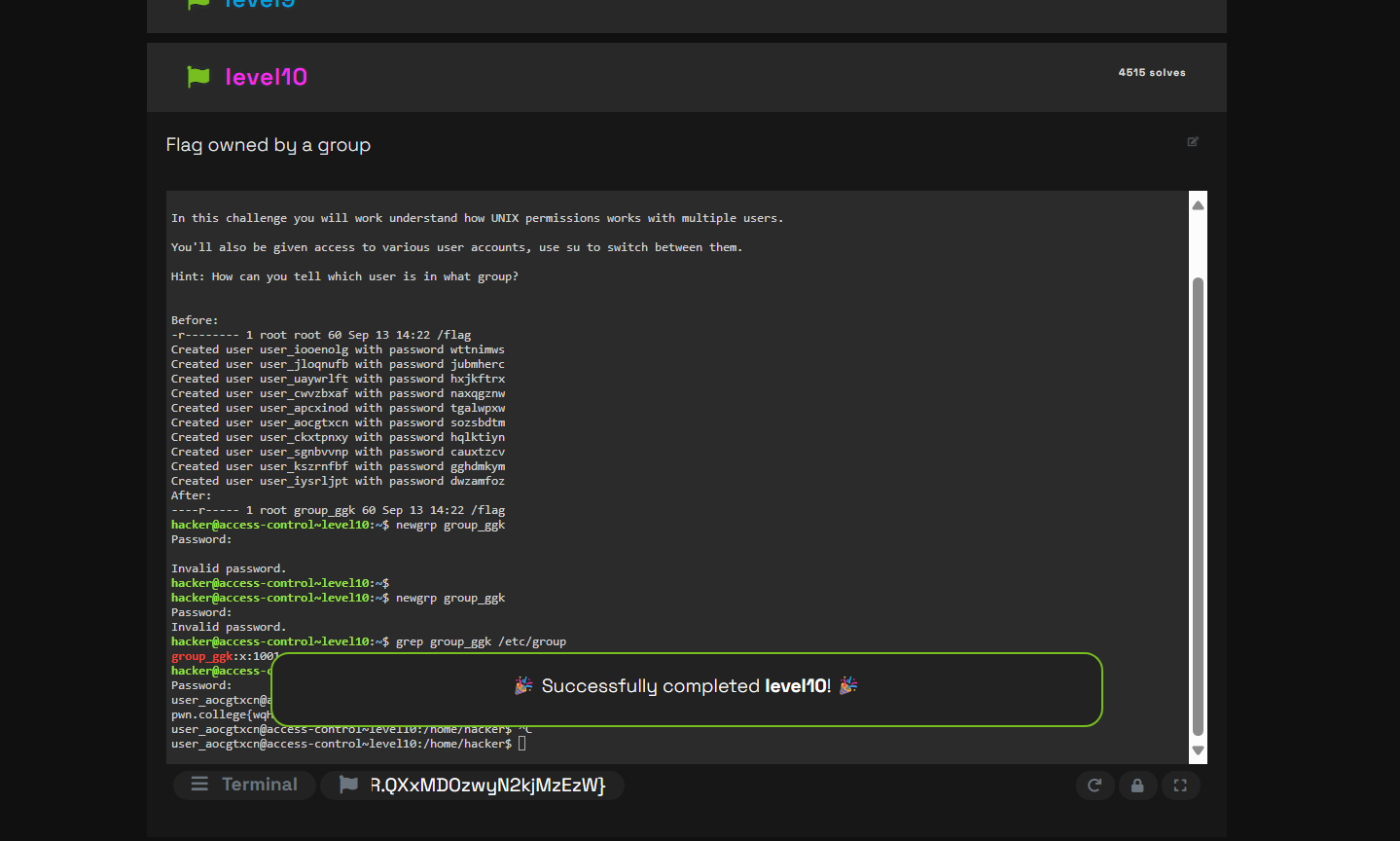
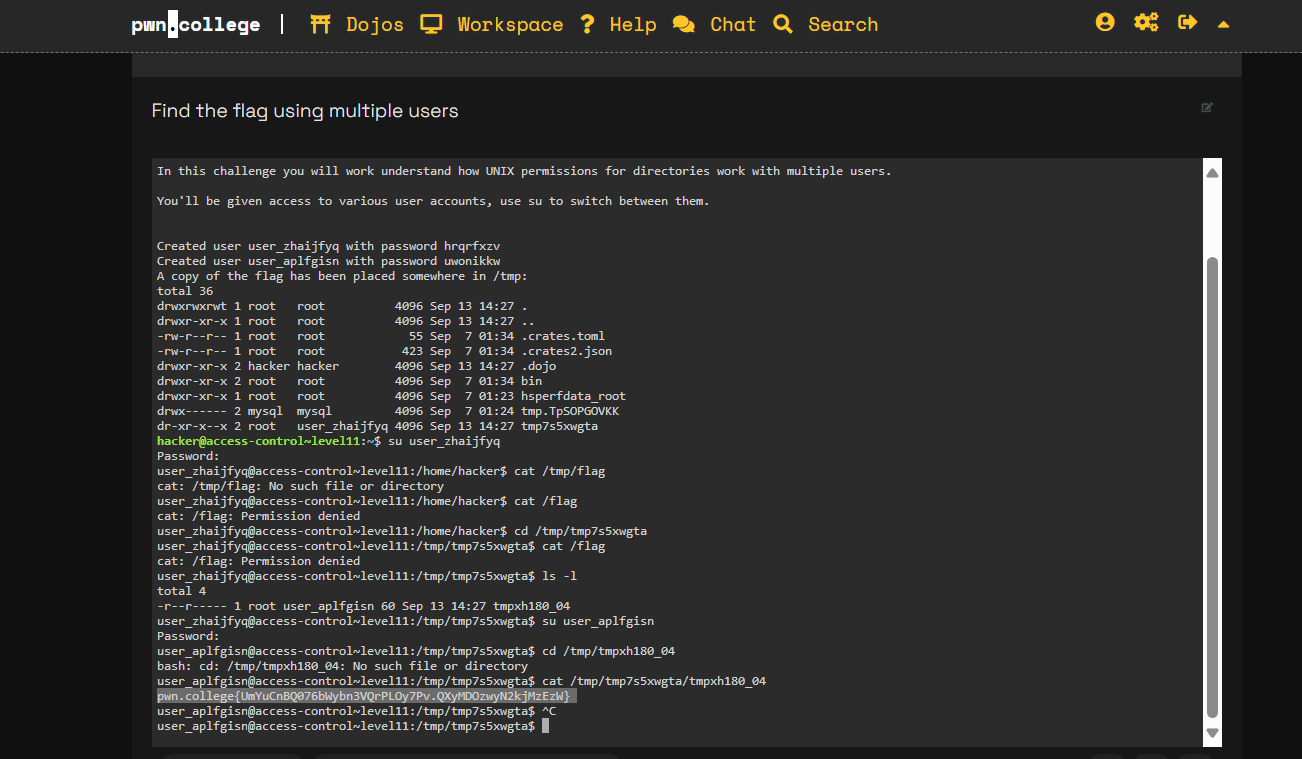
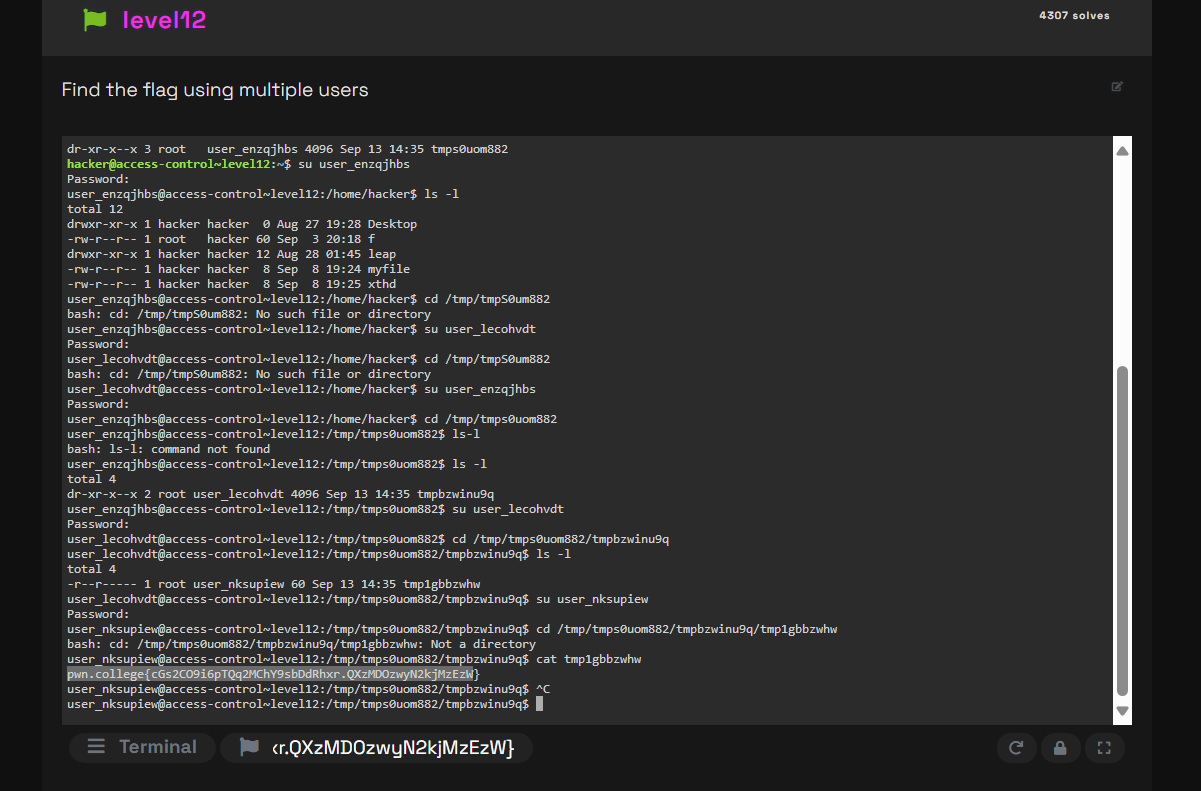
Carlos Rodriguez

Pwn. college username: Carlos

CSCI 400-01

9/13/25

Lab 5

1. Level 1
   1. 
   2. Using the /flag file, I investigated UNIX file permissions in this challenge. The file was set to 400 by the system, denying all other permissions and granting read access exclusively to the owner. I could examine the contents and finish the level with a straightforward command like cat /flag once I had verified that I was the owner.
2. Level 2
   1. 
   2. In this task, I worked with the /flag file to investigate UNIX file permissions. The file was set to 400 by the system, meaning that all other permissions are blocked and only the owner can read it. Once I had verified my ownership, I could read the contents and finish the level with a straightforward command like cat /flag.
3. Level 3
   1. 
   2. Because the flag file in this challenge was created with 000 permissions, no one—not even the owner—could read, write, or execute it. I was able to get around this restriction because I was the file's owner by using chmod 400 to change the permissions. I was able to successfully use cat /flag to make the contents of the flag visible after doing this.
4. Level 4
   1. 
   2. In this challenge, I learned how the SETUID bit changes the way a program runs. By setting the SETUID bit on /bin/cat, the program executed with root’s privileges instead of my own. This allowed me to use /bin/cat /flag and successfully read the flag file.
5. Level 5
   1. 
   2. 
   3. This exercise taught me how the cp command is impacted by the SETUID bit. Running /bin/cp /flag /tmp/flag gave the command root rights, enabling it to copy the contents of the file despite its tight permissions. I successfully read the flag after copying the file to my own location.
6. Level 6
   1. 
   2. The flag in this challenge was secured by being allocated to a unique group that had a password of its own. I changed my primary group to group\_pvblrdcc by using the newgrp command and the password that was supplied. I was able to successfully execute cat /flag and read the contents after I joined that group.
7. Level 7
   1. 
   2. I switched into the new user account that the challenge generated using the su command. Because the /flag file was world-readable, I could access it once I was inside that account. This demonstrated to me how several users can interact with the same file in different ways because of UNIX permissions.
8. Level 8
   1. 
   2. The task established a new user account, which I switched into using the su command. I was able to view the /flag file once I was inside that account because it was world-readable. I learned how different users can interact with the same file in different ways, thanks to UNIX permissions
9. Level 9
   1. 
   2. When the challenge produced a new user account, I switched into it using the su command. The /flag file was world-readable, so once I was inside that account, I could access it. I learned how UNIX permissions allow several users to work with the same file in different ways from this.
10. Level 10
    1. 
    2. The user account that belonged to the group that held the /flag file was the one I had to identify. I used the given password to switch to the right user after verifying the groups. Since the group has read access, I was able to successfully read the flag from there.
11. Level11
    1. 
    2. In order to gain access to restricted folders, I alternated between the many user identities that were generated by the challenge. I found the hidden flag file at /tmp/tmp7s5xwgta as user\_zhaijfyq, but I was unable to access it since it belonged to someone else. I was able to read the flag after changing to user\_aplfqisn since that account possessed the appropriate rights.
12. Level 12
    1. 
    2. I had to switch into the appropriate user account at each stage in order to access a series of restricted directories in /tmp. I accessed the first directory as user\_enzqjhbs, then the next as user\_lecohvdt, and finally the secret file as user\_nkuspjew. The flag was discovered after reading that file, demonstrating how sensitive material is protected by tiered UNIX rights across several users.