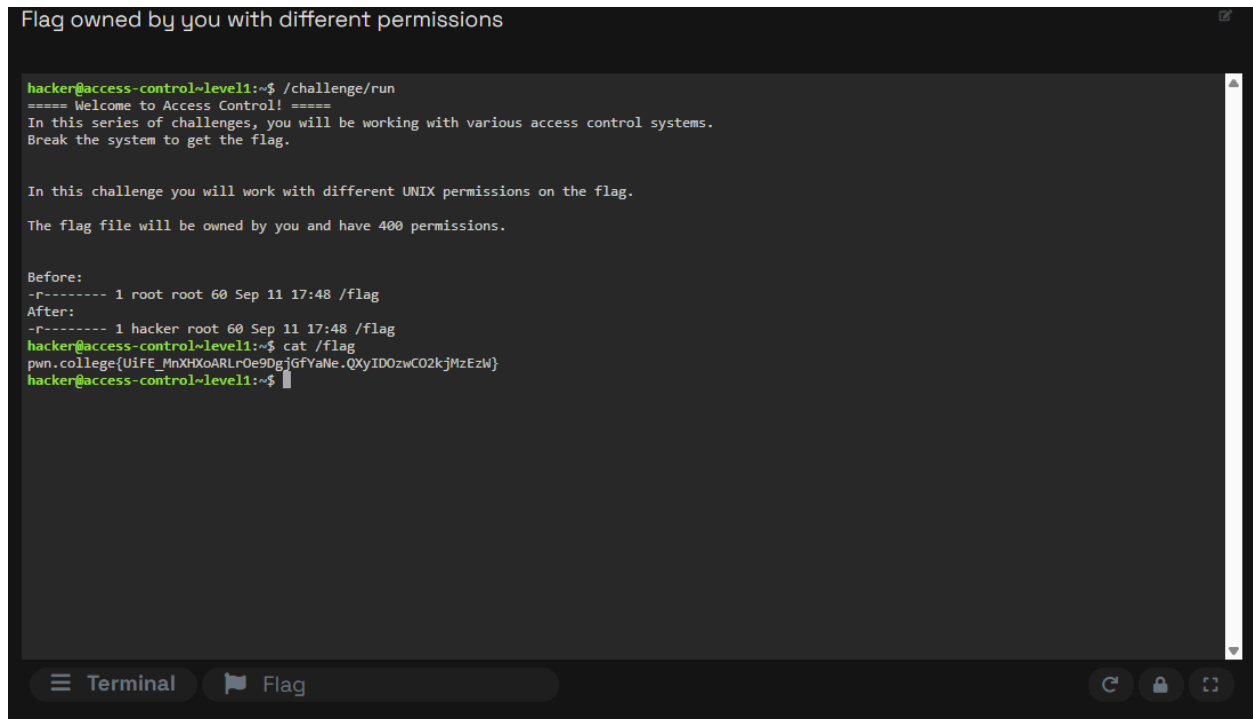


Jaleel Williamson  
jayw-713  
CSCI 400 Lab 5  
9/18/25

Intro to Cybersecurity: Access Control  
Unix Permissions

<https://pwn.college/intro-to-cybersecurity/access-control/>

- **Level 1**



```
Flag owned by you with different permissions

hacker@access-control~level1:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work with different UNIX permissions on the flag.
The flag file will be owned by you and have 400 permissions.

Before:
-r----- 1 root root 60 Sep 11 17:48 /flag
After:
-r----- 1 hacker root 60 Sep 11 17:48 /flag
hacker@access-control~level1:~$ cat /flag
pwn.college{UiFE_MnXHxOARLrOe9DgiGfYaNe.QXyID0ziwC02kjMzEziW}
hacker@access-control~level1:~$
```

Screenshot taken before successful capture.

```
Flag owned by you with different permissions

hacker@access-control~level1:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work with different UNIX permissions on the flag.

The flag file will be owned by you and have 400 permissions.

Before:
-r----- 1 root root 60 Sep 11 17:48 /flag
After:
-r----- 1 hacker root 60 Sep 11 17:48 /flag
hacker@access-control~level1:~$ cat /flag
pwn.college{UIFE_MnXtXoARLn0e9DgJGfYaNe.QXyID0zwC02kJmZEzW}
hacker@access-control~level1:~$

Successfully completed level1!
```

After running the challenge binary, I found that the **/flag file** was now owned by me with read-only permissions, so I could directly access it using **cat /flag**. It felt straightforward but reinforced my understanding of how UNIX file permissions work, especially how ownership grants immediate access without needing complex bypasses. Luckily in this challenge we had **400 permissions**.

• Level 2

```
hacker@access-control~level2:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work with different UNIX permissions on the flag.

The flag file will be owned by root, group as you, and have 040 permissions.

Before:
-r----- 1 root root 60 Sep 11 17:54 /flag
After:
-r----- 1 root hacker 60 Sep 11 17:54 /flag
hacker@access-control~level2:~$ cat /flag
pwn.college{g7kMTZyXtQXEuNTwikA0likokKaa.QXzID0zwC02kJmZEzW}
hacker@access-control~level2:~$
```

Screenshot taken before successful capture.

```
hacker@access-control~level2:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work with different UNIX permissions on the flag.

The flag file will be owned by root, group as you, and have 040 permissions.

Before:
-r----- 1 root root 60 Sep 11 17:54 /flag
After:
-r----- 1 root hacker 60 Sep 11 17:54 /flag
hacker@access-control~level2:~$ cat /flag
pwn.college{g7k4TzyX1QxEuNTwika011cxKaa.QXzID0zwC02kjMzEzW}
hacker@access-control~level2:~$
```

🎉 Successfully completed level2! 🎉

Terminal aa.QXzID0zwC02kjMzEzW}

After running the challenge binary, the **/flag** file had its **permissions** set to **040**, meaning only the group could read it, and since I was in the hacker group, I could directly access it using **cat /flag**. This highlighted how **group permissions** in UNIX can grant access even if the user isn't the owner.

- **Level 3**

```
hacker@access-control~level3:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work with different UNIX permissions on the flag.

The flag file will be owned by you and have 000 permissions.

Before:
-r----- 1 root root 60 Sep 11 17:58 /flag
After:
-r----- 1 hacker root 60 Sep 11 17:58 /flag
hacker@access-control~level3:~$ chmod 400 /flag
hacker@access-control~level3:~$ cat /flag
pwn.college{YB1qUX_utuvAfHpd0nly171edA.QX8ID0zwC02kjMzEzW}
hacker@access-control~level3:~$
```

Terminal Flag

Screenshot taken before successful capture.

```
hacker@access-control~level3:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work with different UNIX permissions on the flag.

The flag file will be owned by you and have 000 permissions.

Before:
-r----- 1 root root 60 Sep 11 17:58 /flag
After:
----- 1 hacker root 60 Sep 11 17:58 /flag
hacker@access-control~level3:~$ chmod 400 /flag
hacker@access-control~level3:~$ cat /flag
pwn.college{YBlqUX_utuvAfHpdDnLv171edA.QX0ID0zwC02kJmZEzW}
hacker@access-control~level3:~$
```

🎉 Successfully completed level3! 🎉

Terminal JA.QX0ID0zwC02kJmZEzW}

After running the challenge binary, the **/flag file** was owned by me but had **no permissions (000)**. Since I owned the file, I used **chmod** to add read permissions for myself with **chmod 400 /flag**. This allowed me to then read the flag directly using **cat /flag**. This demonstrated that ownership grants the ability to **modify** permissions, even when initial access is restricted.

- Level 4

```
hacker@access-control~level4:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work understand how the SETUID bit for UNIX permissions works.

What if /bin/cat had the SETUID bit set?

Before:
-rwxr-xr-x 1 root root 43416 Sep  5 2019 /bin/cat
After:
-rwsr-xr-x 1 root root 43416 Sep  5 2019 /bin/cat
hacker@access-control~level4:~$ /bin/cat /flag
pwn.college{YJpUMh5har8CFH3ENInrQYbBL0d.QX1ID0zwC02kJmZEzW}
hacker@access-control~level4:~$
```

Terminal Flag

Screenshot taken before successful capture.

```
hacker@access-control~level4:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work understand how the SETUID bit for UNIX permissions works.
What if /bin/cat had the SETUID bit set?

Before:
-rwxr-xr-x 1 root root 43416 Sep  5 2019 /bin/cat
After:
-rwsr-xr-x 1 root root 43416 Sep  5 2019 /bin/cat
hacker@access-control~level4:~$ /bin/cat /flag
pwn.college{YJpUMh5har8CFh3EN1NrQYbBL0d.QX1ID0zwC02kjMzEzW}
hacker@access-control~level4:~$
```

🎉 Successfully completed level4! 🎉

After running the challenge binary, **/bin/cat** had the **SETUID** bit set, meaning it would execute with root privileges. Since the flag file (likely **/flag**) is owned by root and has restricted permissions, I used **/bin/cat /flag** to read it. This worked because the **SETUID-enabled** cat ran as root, bypassing the permission checks and allowing me to access the flag directly. This demonstrated the power of **SETUID** in granting elevated privileges through specific binaries.

- Level 5

```
hacker@access-control~level5:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work understand how the SETUID bit for UNIX permissions works.

What if /bin/cp had the SETUID bit set?

Hint: Look into how cp will deal with different permissions.

Another Hint: check the man page for cp, any options in there that might help?

Before:
-rwxr-xr-x 1 root root 153976 Sep  5  2019 /bin/cp
After:
-rwsr-xr-x 1 root root 153976 Sep  5  2019 /bin/cp
hacker@access-control~level5:~$ /bin/cp --chmod=644 /flag /tmp/flag_copy
/bin/cp: unrecognized option '--chmod=644'
Try '/bin/cp --help' for more information.
hacker@access-control~level5:~$ /bin/cp /flag /dev/stdout > /tmp/flag_copy
hacker@access-control~level5:~$ cat /tmp/flag_copy
pwn.college{MvEiwrPpzSrygfqln_PLX8Uo_JA.QX2ID0zwC02kjMzEzW}
hacker@access-control~level5:~$
```

Screenshot taken before successful capture.

```
hacker@access-control~level5:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work understand how the SETUID bit for UNIX permissions works.

What if /bin/cp had the SETUID bit set?

Hint: Look into how cp will deal with different permissions.

Another Hint: check the man page for cp, any options in there that might help?

Before:
-rwxr-xr-x 1 root root 153976 Sep  5  2019 /bin/cp
After:
-rwsr-xr-x 1 root root 153976 Sep  5  2019 /bin/cp
hacker@access-control~level5:~$ /bin/cp --chmod=644 /flag /tmp/flag_copy
/bin/cp: unrecognized option '--chmod=644'
Try '/bin/cp --help' for more information.
hacker@access-control~level5:~$ /bin/cp /flag /dev/stdout > /tmp/flag_copy
hacker@access-control~level5:~$ cat /tmp/flag_copy
pwn.college{MvEiwrPpzSrygfqln_PLX8Uo_JA.QX2ID0zwC02kjMzEzW}
hacker@access-control~level5:~$
```

🎉 Successfully completed level5! 🎉

After running the challenge binary, **/bin/cp** had the **SETUID** bit set, allowing it to run with root privileges. To access the flag, I used **/bin/cp** to copy the contents of **/flag** to standard output and then redirected that output to a file in **/tmp**, which I could read. This worked because the **SETUID-enabled** cp could read the restricted **/flag** file as root, and the redirection created a file

owned by me with default permissions. This approach bypassed the permission issues since the copied file was created under my user account, making it accessible without needing to modify permissions directly.

- **Level 6**

```
hacker@access-control~level6:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work with different UNIX permissions on the flag.

The flag file is owned by root and a new group.

Hint: Search for how to join a group with a password.

Before:
-r----- 1 root root 60 Sep 11 18:23 /flag
After:
----r----- 1 root group_cejsmmrw 60 Sep 11 18:23 /flag
The password for group_cejsmmrw is: gbpedagk
hacker@access-control~level6:~$ newgrp group_cejsmmrw
Password:
hacker@access-control~level6:~$ cat /flag
pwn.college{QKxoHAsabMeqcVx0s60hV5nlzZE.QX3ID0zwC02kjMzEzW}
hacker@access-control~level6:~$
```

Screenshot taken before successful capture.

```
hacker@access-control~level6:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work with different UNIX permissions on the flag.

The flag file is owned by root and a new group.

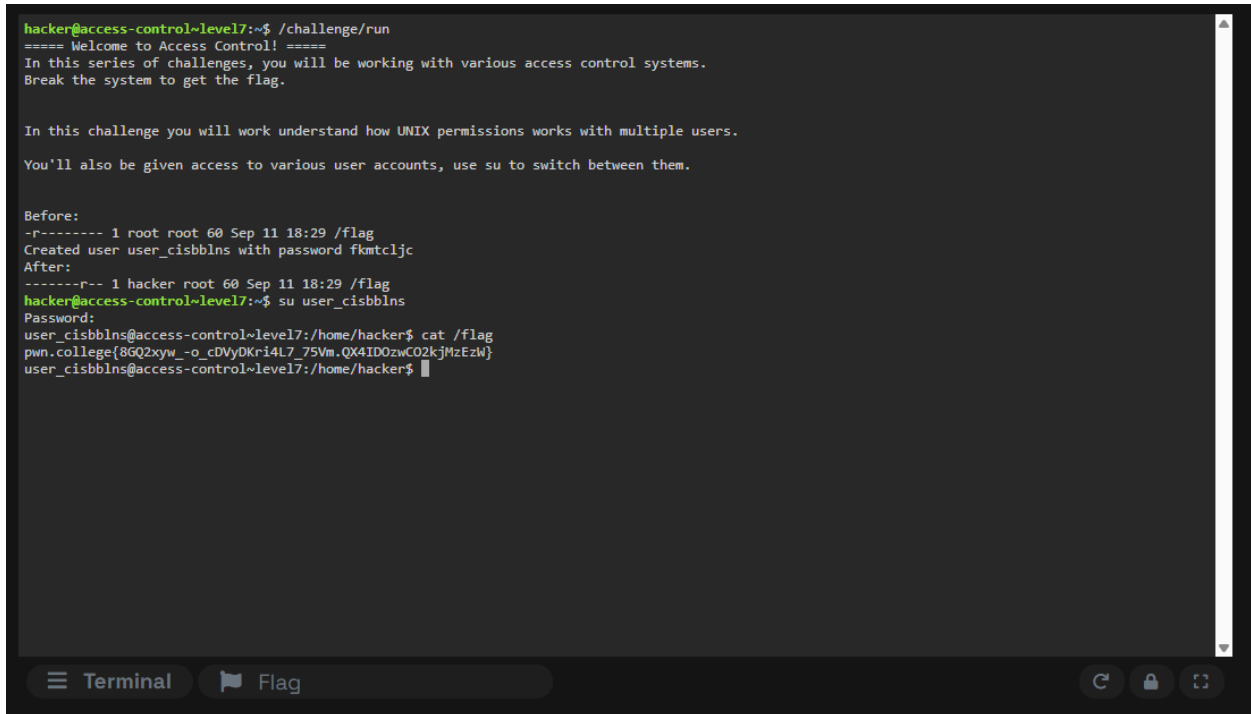
Hint: Search for how to join a group with a password.

Before:
-r----- 1 root root 60 Sep 11 18:23 /flag
After:
----r----- 1 root group_cejsmmrw 60 Sep 11 18:23 /flag
The password for group_cejsmmrw is: gbpedagk
hacker@access-control~level6:~$ newgrp group_cejsmmrw
Password:
hacker@access-control~level6:~$ cat /flag
pwn.college{QKxoHAsabMeqcVx0s60hV5nlzZE.QX3ID0zwC02kjMzEzW}
hacker@access-control~level6:~$
```

🎉 Successfully completed level6! 🎉

After running the challenge binary, the **/flag** file had its permissions set to allow only the group **group\_cejsmmrw** to read it. Since I was not initially a member of that group, I used the **newgrp** command with the provided password to switch to that group. After entering the password, I gained group membership and was able to read the flag directly using **cat /flag**. This demonstrated how group passwords can be used to gain access to restricted files without root intervention.

- Level 7



```
hacker@access-control~level7:~$ ./challenge/run
==== Welcome to Access Control! ====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work understand how UNIX permissions works with multiple users.
You'll also be given access to various user accounts, use su to switch between them.

Before:
-r----- 1 root root 60 Sep 11 18:29 /flag
Created user user_cisbblns with password fkmcljc
After:
-----r-- 1 hacker root 60 Sep 11 18:29 /flag
hacker@access-control~level7:~$ su user_cisbblns
Password:
user_cisbblns@access-control~level7:/home/hacker$ cat /flag
pwn.college{8GQ2xyw_-o_cDVyDKr-i4L7_75Vm.QX4ID0zwC02kjHzEzW}
user_cisbblns@access-control~level7:/home/hacker$
```

Screenshot taken before successful capture.



```
hacker@access-control~level7:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work understand how UNIX permissions works with multiple users.
You'll also be given access to various user accounts, use su to switch between them.

Before:
-r----- 1 root root 60 Sep 11 18:29 /flag
Created user user_cisbblns with password fkmtcljc
After:
-----r-- 1 hacker root 60 Sep 11 18:29 /flag
hacker@access-control~level7:~$ su user_cisbblns
Password:
user_cisbblns@access-control~level7:/home/hacker$ cat /flag
pwn.college{8GQ2xyw_o_cDVyDKri4L7_75Vm_QX4ID0zwC02kjMzEzW}
user_cisbblns@access-control~level7:/home/hacker$
```

🎉 Successfully completed level7! 🎉

Terminal 'm.QX4ID0zwC02kjMzEzW}'

After running the challenge binary, the **/flag** file had permissions set to **-----r--**, meaning only others could read it. Since I was the **owner (hacker)** but without read permissions, I needed to access it as another user. I used the provided credentials to switch to **user\_cisbblns** with the password **fkmtcljc** using **su user\_cisbblns**. Once switched, I was able to read the flag directly with **cat /flag** because the file granted read access to others. This demonstrated how UNIX permissions can allow access through other users when appropriate permissions are set.

• Level 8

```
hacker@access-control~level8:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work understand how UNIX permissions works with multiple users.
You'll also be given access to various user accounts, use su to switch between them.

Before:
-r----- 1 root root 60 Sep 11 18:33 /flag
Created user user_yeotlidd with password irxiwzhk
After:
-r----- 1 user_yeotlidd root 60 Sep 11 18:33 /flag
hacker@access-control~level8:~$ su user_yeotlidd
Password:
user_yeotlidd@access-control~level8:/home/hacker$ cat /flag
pwn.college{QFQmi#NmdCfXLFJ15kX750v-Fd-.QX5ID0zwC02kjMzEzW}
user_yeotlidd@access-control~level8:/home/hacker$
```

Terminal Flag

Screenshot taken before successful capture.

```
hacker@access-control~level8:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work understand how UNIX permissions works with multiple users.
You'll also be given access to various user accounts, use su to switch between them.

Before:
-r----- 1 root root 60 Sep 11 18:33 /flag
Created user user_yeotlidd with password irxiwzhk
After:
-r----- 1 user_yeotlidd root 60 Sep 11 18:33 /flag
hacker@access-control~level8:~$ su user_yeotlidd
Password:
user_yeotlidd@access-control~level8:/home/hacker$ cat /flag
pwn.college{0F0m#nomdCfXLfJlSkX750v-Fd-.QX5ID0zwC02kjmZEzW}
user_yeotlidd@access-control~level8:/home/hacker$
```

🎉 Successfully completed level8! 🎉

Terminal d-.QX5ID0zwC02kjmZEzW}

After running the challenge binary, the /flag file was owned by **user\_yeotlidd** with read-only permissions (400). Since I was logged in as hacker, I used **su user\_yeotlidd** with the provided password **irxiwzhk** to switch to that user. Once switched, I was able to read the flag directly with **cat /flag** because I became the **owner** of the file. This demonstrated how switching users can grant access to files based on ownership and permissions.

- Level 9

```
hacker@access-control~level9:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work understand how UNIX permissions works with multiple users.
You'll also be given access to various user accounts, use su to switch between them.

Before:
-r----- 1 root root 60 Sep 11 18:37 /flag
Created user user_uytzvixa with password lxqdcnxi
After:
-r----- 1 root user_uytzvixa 60 Sep 11 18:37 /flag
hacker@access-control~level9:~$ su user_uytzvixa
Password:
user_uytzvixa@access-control~level9:/home/hacker$ cat /flag
pwn.college{08hITG68z5s04iAzVyjPN3XScrT.QXw#D0zwC02kjmZEzW}
user_uytzvixa@access-control~level9:/home/hacker$
```

Terminal Flag

Screenshot taken before successful capture.

```
hacker@access-control~level9:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work understand how UNIX permissions works with multiple users.
You'll also be given access to various user accounts, use su to switch between them.

Before:
-r----- 1 root root 60 Sep 11 18:37 /flag
Created user user_uytzvixa with password kxqdcnxi
After:
-r----- 1 root user_uytzvixa 60 Sep 11 18:37 /flag
hacker@access-control~level9:~$ su user_uytzvixa
Password:
user_uytzvixa@access-control~level9:/home/hacker$ cat /flag
pwn.college{0BhITG6Bz5s04IAzVyrjPN3XScRT_0Xw#D0zwC02kjMzEzW}
user_uytzvixa@access-control~level9:/home/hacker$
```

🚩 Successfully completed level9! 🚩

Terminal r.QXwMD0zwC02kjMzEzWj

After running the challenge binary, the **/flag** file had its permissions set to allow only the group **user\_uytzvixa** to read it. Since I was logged in as hacker, I used **su user\_uytzvixa** with the provided password **kxqdcnxi** to switch to that user. Once switched, I was able to read the flag directly with **cat /flag** because **user\_uytzvixa** is a member of the group **user\_uytzvixa**, which had the necessary read permissions. This demonstrated how group-based permissions can be accessed by switching to a user account that belongs to the required group.

## • Level 10

```
hacker@access-control-level10:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work understand how UNIX permissions works with multiple users.
You'll also be given access to various user accounts, use su to switch between them.
Hint: How can you tell which user is in what group?

Before:
-r----- 1 root root 60 Sep 11 18:45 /flag
Created user user_pkhfiq with password lzgrtqcg
Created user user_gymwdf with password mxeqwp
Created user user_dlpqrk with password wblhajw
Created user user_xfrkqza with password duxlfqye
Created user user_uyrted with password hzjgmk
Created user user_midakke with password hjcexafv
Created user user_febpfje with password nccfwmk
Created user user_gboaxbe with password gjlvuaw
Created user user_dymawok with password btsokvsk
Created user user_lqblmr with password rjwzgo
After:
-r----- 1 root group_nvc 60 Sep 11 18:45 /flag
hacker@access-control-level10:~$ grep group_nvc /etc/group
group_nvc:x1801:user_gboaxbe
hacker@access-control-level10:~$ su user_dlpqrk
Password:
su: Authentication failure
hacker@access-control-level10:~$ su user_gboaxbe
Password:
user_gboaxbe@access-control-level10:/home/hacker$ cat /flag
pwn.college{0r0Qut-V1Mkv#Wj-Suk1U0h_0_QXw#D0zwC02kjMzEzW}
user_gboaxbe@access-control-level10:/home/hacker$
```

Terminal Flag

Screenshot taken before successful capture.

```
hacker@access-control-level10:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work understand how UNIX permissions works with multiple users.
You'll also be given access to various user accounts, use su to switch between them.
Hint: How can you tell which user is in what group?

Before:
-=-=-=-=- 1 root root 60 Sep 11 18:45 /flag
Created user user_gblrfag with password lgrtqog
Created user user_yymasbf with password mbqowfp
Created user user_dipwrfk with password whbajfw
Created user user_xfrkna with password duxifqe
Created user user_yyyvetgd with password bqzjgrnk
Created user user_milafke with password hcxasfv
Created user user_tebqjak with password noohemk
Created user user_gbhoaxe with password gjlviwxw
Created user user_gjpwew with password btsoheek
Created user user_lgblmor with password rljoeign
After:
-=-=-=-=- 1 root group_nvc 60 Sep 11 18:45 /flag
hacker@access-control-level10:~$ grep group_nvc /etc/group
group_nvc:x:101:101:user_gbhoaxe
hacker@access-control-level10:~$ su user_dipwrfk
Password:
su: Authentication failure
hacker@access-control-level10:~$ su user_gbhoaxe
Password:
user_gbhoaxe@access-control-level10:/home/hacker$ cat /flag
pan.college{0R0qun:viMcwWj}w4clt0h_0_0x0b0zco0k36tFag
user_gbhoaxe@access-control-level10:/home/hacker$
```

Successfully completed level10!

After running the challenge binary, the **/flag** file was set to be readable only by the group **group\_nvc**. To determine which user belongs to this group, I used **grep group\_nvc /etc/group**, which revealed that **user\_gbhoaxe** is a member of **group\_nvc**. I then switched to this user using the provided password (**gjlviwxw**) and read the flag with **cat /flag**. This demonstrated how to identify group membership and switch users to access files based on group permissions.

• Level 11

```
hacker@access-control-level11:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work understand how UNIX permissions for directories work with multiple users.
You'll be given access to various user accounts, use su to switch between them.

Created user user_rakstfeu with password wocsjuov
Created user user_knxbwkvf with password lzumtiva
A copy of the flag has been placed somewhere in /tmp:
total 36
drwxrwxrwt 1 root root 4096 Sep 13 12:27 .
drwxr-xr-x 1 root root 4096 Sep 13 12:27 ..
-rw-r--r-- 1 root root 55 Sep 7 01:34 .crates.toml
-rw-r--r-- 1 root root 423 Sep 7 01:34 .crates2.json
drwxr-xr-x 2 hacker hacker 4096 Sep 13 12:27 .dojo
drwxr-xr-x 2 root root 4096 Sep 7 01:34 bin
drwxr-xr-x 1 root root 4096 Sep 7 01:23 hspcrfddata_root
drwx----- 2 mysql mysql 4096 Sep 7 01:24 tmpIpsOPGOVKK
dr-xr-x--x 2 root user_rakstfeu 4096 Sep 13 12:27 tmp333dzt0c
hacker@access-control-level11:~$ su user_rakstfeu
Password:
user_rakstfeu@access-control-level11:/home/hacker$ ls -la /tmp/tmp333dzt0c
total 12
dr-xr-x--x 2 root user_rakstfeu 4096 Sep 13 12:27 .
drwxrwxrwt 1 root root 4096 Sep 13 12:27 ..
-r--r----- 1 root user_knxbwkvf 60 Sep 13 12:27 tmp_ue9qi4f
user_rakstfeu@access-control-level11:/home/hacker$ cat /tmp/tmp333dzt0c/flag
cat: /tmp/tmp333dzt0c/flag: No such file or directory
```

First screenshot continues into the next one.

```
In this challenge you will work understand how UNIX permissions for directories work with multiple users.
You'll be given access to various user accounts, use su to switch between them.

Created user user_rakstfeu with password wocsjuov
Created user user_knxbwkvf with password izumtiva
A copy of the flag has been placed somewhere in /tmp:
total 36
drwxrwxrwt 1 root root 4096 Sep 13 12:27 .
drwxr-xr-x 1 root root 4096 Sep 13 12:27 ..
-rw-r--r-- 1 root root 55 Sep 7 01:34 .crates.toml
-rw-r--r-- 1 root root 423 Sep 7 01:34 .crates2.json
drwxr-xr-x 2 hacker hacker 4096 Sep 13 12:27 .dojo
drwxr-xr-x 2 root root 4096 Sep 7 01:34 bin
drwxr-xr-x 1 root root 4096 Sep 7 01:23 hsperrdata_root
drwx----- 2 mysql mysql 4096 Sep 7 01:24 tmp.TpSOPGOVKK
dr-xr-x--x 2 root user_rakstfeu 4096 Sep 13 12:27 tmp333dzt0c
hacker@access-control-level11:~$ su user_rakstfeu
Password:
user_rakstfeu@access-control-level11:/home/hacker$ ls -la /tmp/tmp333dzt0c
total 12
dr-xr-x--x 2 root user_rakstfeu 4096 Sep 13 12:27 .
drwxrwxrwt 1 root root 4096 Sep 13 12:27 ..
-r--r----- 1 root user_knxbwkvf 60 Sep 13 12:27 tmp_ue9qi4f
user_rakstfeu@access-control-level11:/home/hacker$ cat /tmp/tmp333dzt0c/flag
cat: /tmp/tmp333dzt0c/flag: No such file or directory
user_rakstfeu@access-control-level11:/home/hacker$ su user_knxbwkvf
Password:
user_knxbwkvf@access-control-level11:/home/hacker$ cat /tmp/tmp333dzt0c/tmp_ue9qi4f
pwn.college{A0cPYFAFBu-b-2rx5I6RrWQ6XKX.QXyMD0zwC02kjMzEzW}
user_knxbwkvf@access-control-level11:/home/hacker$
```

Screenshot taken before successful capture.

```
In this challenge you will work understand how UNIX permissions for directories work with multiple users.
You'll be given access to various user accounts, use su to switch between them.

Created user user_rakstfeu with password wocsjuov
Created user user_knxbwkvf with password izumtiva
A copy of the flag has been placed somewhere in /tmp:
total 36
drwxrwxrwt 1 root root 4096 Sep 13 12:27 .
drwxr-xr-x 1 root root 4096 Sep 13 12:27 ..
-rw-r--r-- 1 root root 55 Sep 7 01:34 .crates.toml
-rw-r--r-- 1 root root 423 Sep 7 01:34 .crates2.json
drwxr-xr-x 2 hacker hacker 4096 Sep 13 12:27 .dojo
drwxr-xr-x 2 root root 4096 Sep 7 01:34 bin
drwxr-xr-x 1 root root 4096 Sep 7 01:23 hsperrdata_root
drwx----- 2 mysql mysql 4096 Sep 7 01:24 tmp.TpSOPGOVKK
dr-xr-x--x 2 root user_rakstfeu 4096 Sep 13 12:27 tmp333dzt0c
hacker@access-control-level11:~$ su user_rakstfeu
Password:
user_rakstfeu@access-control-level11:/home/hacker$ ls -la /tmp/tmp333dzt0c
total 12
dr-xr-x--x 2 root user_rakstfeu 4096 Sep 13 12:27 .
drwxrwxrwt 1 root root 4096 Sep 13 12:27 ..
-r--r----- 1 root user_knxbwkvf 60 Sep 13 12:27 tmp_ue9qi4f
user_rakstfeu@access-control-level11:/home/hacker$ cat /tmp/tmp333dzt0c/tmp_ue9qi4f
pwn.college{A0cPYFAFBu-b-2rx5I6RrWQ6XKX.QXyMD0zwC02kjMzEzW}
user_knxbwkvf@access-control-level11:/home/hacker$
```

After running the challenge, the flag was stored in the file `/tmp/tmp333dzt0c/tmp_ue9qi4f`. The file had permissions `-r--r-----`, meaning it was readable only by the **owner (root)** or the group (**user\_knxbwkvf**). I switched to **user\_knxbwkvf** using the password **izumtiva** and then read the file directly with `cat /tmp/tmp333dzt0c/tmp_ue9qi4f` to retrieve the flag. This demonstrated how to access files by switching to a user account that has group read permissions, even when the directory permissions allow only specific users to list contents.

## • Level 12

```
hacker@access-control-level12:~$ /challenge/run
===== Welcome to Access Control! =====
In this series of challenges, you will be working with various access control systems.
Break the system to get the flag.

In this challenge you will work understand how UNIX permissions for directories work with multiple users.
You'll be given access to various user accounts, use su to switch between them.

Created user user_xhznswfh with password hjyzyfic
Created user user_xchjjhbb with password lqgkzpzp
Created user user_bwmhvphx with password musvxyvn
A copy of the flag has been placed somewhere in /tmp:
total 36
drwxrwxrwt 1 root root 4096 Sep 13 12:36 .
drwxr-xr-x 1 root root 4096 Sep 13 12:35 ..
-rw-r--r-- 1 root root 55 Sep 7 01:34 .crates.toml
-rw-r--r-- 1 root root 423 Sep 7 01:34 .crates2.json
drwxr-xr-x 2 hacker hacker 4096 Sep 13 12:36 .dojo
drwxr-xr-x 2 root root 4096 Sep 7 01:34 bin
drwxr-xr-x 1 root root 4096 Sep 7 01:23 hspcrfdata_root
drwx----- 2 mysql mysql 4096 Sep 7 01:24 tmp.TpSOPGOVKK
dr-xr-x--x 3 root user_xchjjhbb 4096 Sep 13 12:36 tmphcgwx6pr
hacker@access-control-level12:~$ su user_xchjjhbb
Password:
su: Authentication failure
hacker@access-control-level12:~$
hacker@access-control-level12:~$ su user_xchjjhbb
Password:
user_xchjjhbb@access-control-level12:/home/hacker$ ls -la /tmp/tmphcgwx6pr
```

First screenshot continues into the next one.

```
-rw-r--r-- 1 root root 55 Sep 7 01:34 .crates.toml
-rw-r--r-- 1 root root 423 Sep 7 01:34 .crates2.json
drwxr-xr-x 2 hacker hacker 4096 Sep 13 12:36 .dojo
drwxr-xr-x 2 root root 4096 Sep 7 01:34 bin
drwxr-xr-x 1 root root 4096 Sep 7 01:23 hspcrfdata_root
drwx----- 2 mysql mysql 4096 Sep 7 01:24 tmp.TpSOPGOVKK
dr-xr-x--x 3 root user_xchjjhbb 4096 Sep 13 12:36 tmphcgwx6pr
hacker@access-control-level12:~$ su user_xchjjhbb
Password:
su: Authentication failure
hacker@access-control-level12:~$
hacker@access-control-level12:~$ su user_xchjjhbb
Password:
user_xchjjhbb@access-control-level12:/home/hacker$ ls -la /tmp/tmphcgwx6pr
total 12
dr-xr-x--x 3 root user_xchjjhbb 4096 Sep 13 12:36 .
drwxrwxrwt 1 root root 4096 Sep 13 12:36 ..
dr-xr-x--x 2 root user_bwmhvphx 4096 Sep 13 12:36 tmpksd0pea0
user_xchjjhbb@access-control-level12:/home/hacker$ su user_bwmhvphx
Password:
user_bwmhvphx@access-control-level12:/home/hacker$ cat /tmp/tmphcgwx6pr/<FLAGFILE>
bash: syntax error near unexpected token `newline'
user_bwmhvphx@access-control-level12:/home/hacker$ ls -la /tmp/tmphcgwx6pr/tmpksd0pea0
total 12
dr-xr-x--x 2 root user_bwmhvphx 4096 Sep 13 12:36 .
dr-xr-x--x 3 root user_xchjjhbb 4096 Sep 13 12:36 ..
-r--r----- 1 root user_xhznswfh 60 Sep 13 12:36 tmpm4txgs6q
user_bwmhvphx@access-control-level12:/home/hacker$ su user_xhznswfh
Password:
user_xhznswfh@access-control-level12:/home/hacker$ cat /tmp/tmphcgwx6pr/tmpksd0pea0/tmpm4txgs6q
pwn.college{wP7yMVot56tU8ob-AbthSKN3VyK.QXzMD0zwC02kjMzEzW}
user_xhznswfh@access-control-level12:/home/hacker$
```

Screenshot taken before successful capture.

```
-rw-r--r-- 1 root root      55 Sep  7 01:34 .crates.toml
-rw-r--r-- 1 root root     423 Sep  7 01:34 .crates2.json
drwxr-xr-x 2 hacker hacker   4096 Sep 13 12:36 .dojo
drwxr-xr-x 2 root  root      4096 Sep  7 01:34 bin
drwxr-xr-x 1 root  root      4096 Sep  7 01:23 hsperrdata_root
drwx----- 2 mysql mysql    4096 Sep  7 01:24 tmp.TpSOPG0VKK
dr-xr-x--x 3 root  user_xchjjhbb 4096 Sep 13 12:36 tmpcgcwx6pr
hacker@access-control-level12:~$ su user_xchjjhbb
Password:
su: Authentication failure
hacker@access-control-level12:~$
hacker@access-control-level12:~$ su user_xchjjhbb
Password:
user_xchjjhbb@access-control-level12:/home/hacker$ ls -la /tmp/tmpcgcwx6pr
total 12
dr-xr-x--x 3 root user_xchjjhbb 4096 Sep 13 12:36 .
drwxrwxrwt 1 root root          4096 Sep 13 12:36 ..
dr-xr-x--x 2 root user_bwmhvphx 4096 Sep 13 12:36 tmpksd0pea0
user_xchjjhbb@access-control-level12:/home/hacker$ su user_bwmhvphx
Password:
user_bwmhvphx@access-control-level12:/home/hacker$ cat /tmp/tmpcgcwx6pr/<FLAGFILE>
bash: syntax error near unexpected token 'newline'
user_bwmhvphx@access-control-level12:/home/hacker$ ls -la /tmp/tmpcgcwx6pr/tmpksd0pea0
total 12
dr-xr-x--x 2 root user_bwmhvphx 4096 Sep 13 12:36 .
dr-xr-x--x 3 root user_xchjjhbb 4096 Sep 13 12:36 ..
-r--r--r-- 1 root user_xchjjhbb 4096 Sep 13 12:36 tmpm4txgs6q
user_bwmhvphx@access-control-level12:/home/hacker$ su user_xhznswfh
Password:
user_xhznswfh@access-control-level12:/home/hacker$ cat /tmp/tmpcgcwx6pr/tmpksd0pea0/tmpm4txgs6q
gwn.college(wP7yMVoT56tU8ob-AbthSKN3vyk.QXzMD0zwCO2kzMzEzW)
user_xhznswfh@access-control-level12:/home/hacker$
```

Successfully completed level12! 🎉

Terminal .QXzMD0zwCO2kzMzEzW}

After running the challenge, the flag was located in a nested directory structure within **/tmp**. I first switched to **user\_xchjjhbb** (password: **lgqkzpu**) to access the directory **/tmp/tmpcgcwx6pr**. Then, I switched to **user\_bwmhvphx** (password: **musvxyyn**) to access the subdirectory **tmpksd0pea0**. Finally, I switched to **user\_xhznswfh** (password: **hxyzfyic**) to read the file **tmpm4txgs6q** because it was **group-readable** by **user\_xhznswfh**. The flag was retrieved by reading the file with **cat**. This process involved switching users multiple times to leverage group permissions on directories and files, ultimately allowing access to the flag.