Ubuntu Linux User Management Security Lab
By Michael Ambeguia

Lab Intro: This lab is inspired by the topics covered in the book ***Mastering Linux Security and Hardening 3rd Edition*** by Donald Tevault. I highly recommend reading this book if you're interested in hardening your Linux machines since Donald explains the concepts in a detailed fashion and provides plenty of examples of their use in real world situations.

Lab Goals: The goal for this lab is to apply the topics in chapter 3 " Securing Normal User Accounts". I will be demonstrating my knowledge of the topics covered in the chapter and reinforcing my learning by doing hands-on tasks. ****Note I will not be following the labs in the book I will simply just apply the concepts (this is my own lab).****

Lab Sections:

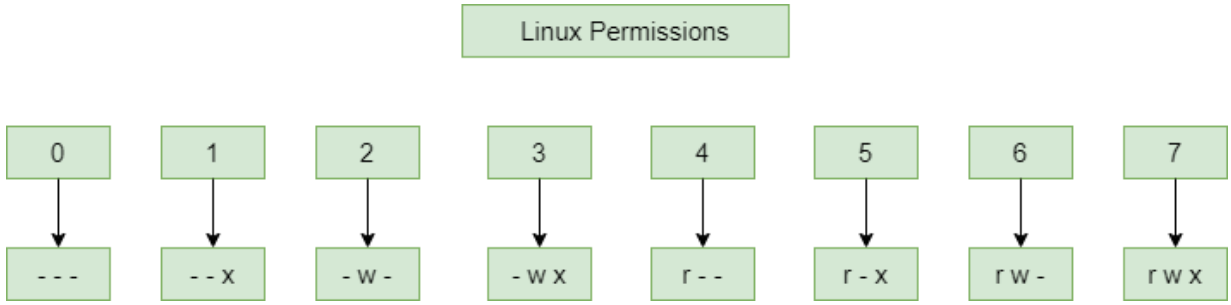
1. Locking down user home directories.
2. Enforcing strong password criteria
3. Setting/ enforcing password, account expiration.
4. Locking user accounts

Section #1: Locking down user home directories.

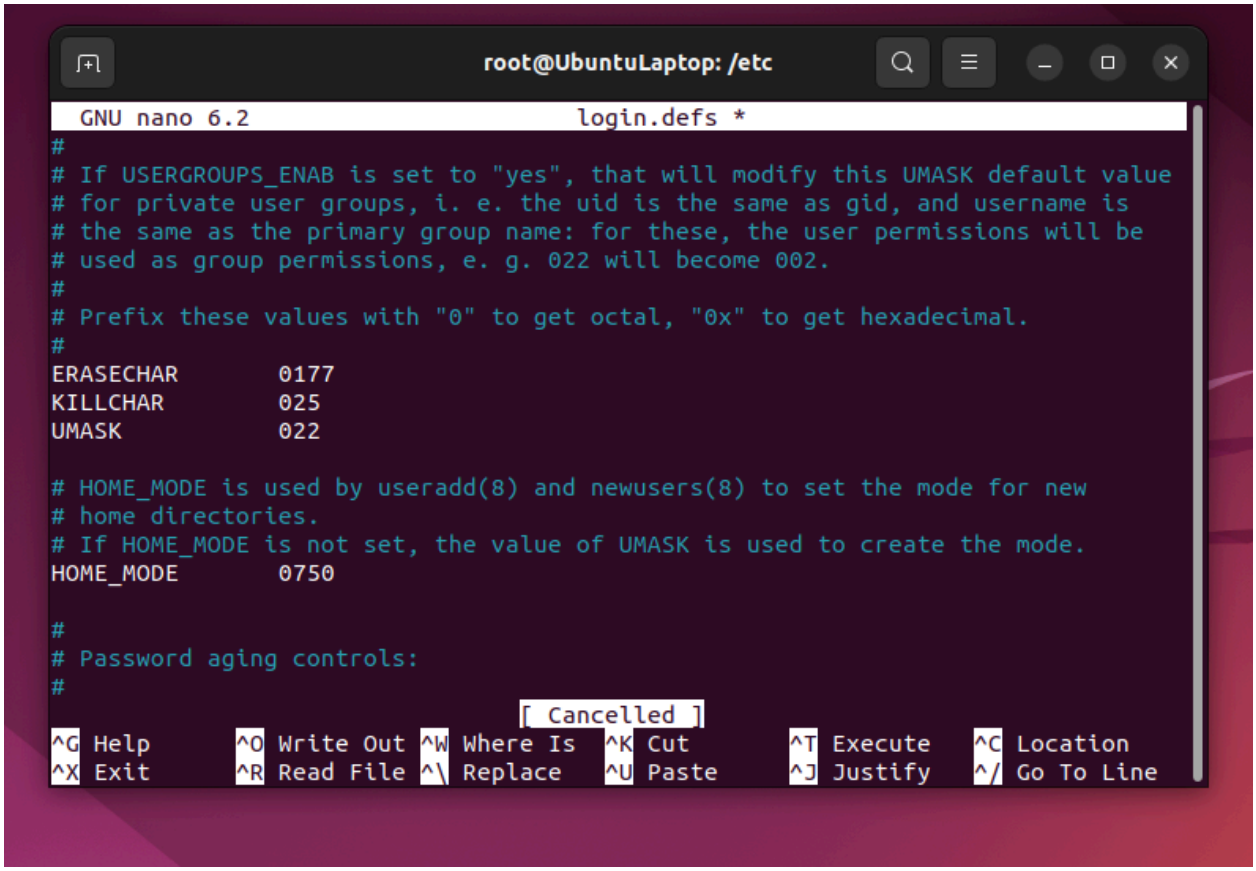
Issue: On Ubuntu machines, when you first create a new user their home directory is not protected using access permissions. That means that any other user on the machine can access that user's home directory and vice-versa. This is not proper since everyone should only have access to their own directory. If sharing is needed, then a shared directory can be created amongst the users on a machine.

Solution #1. Change the settings on the /etc/login.defs config (for changing the directory permissions based on the useradd command). Note that useradd does not automatically create a user's home directory, that is done manually. Also the proper setting might be in place by default on Ubuntu!

Background (Linux permissions):



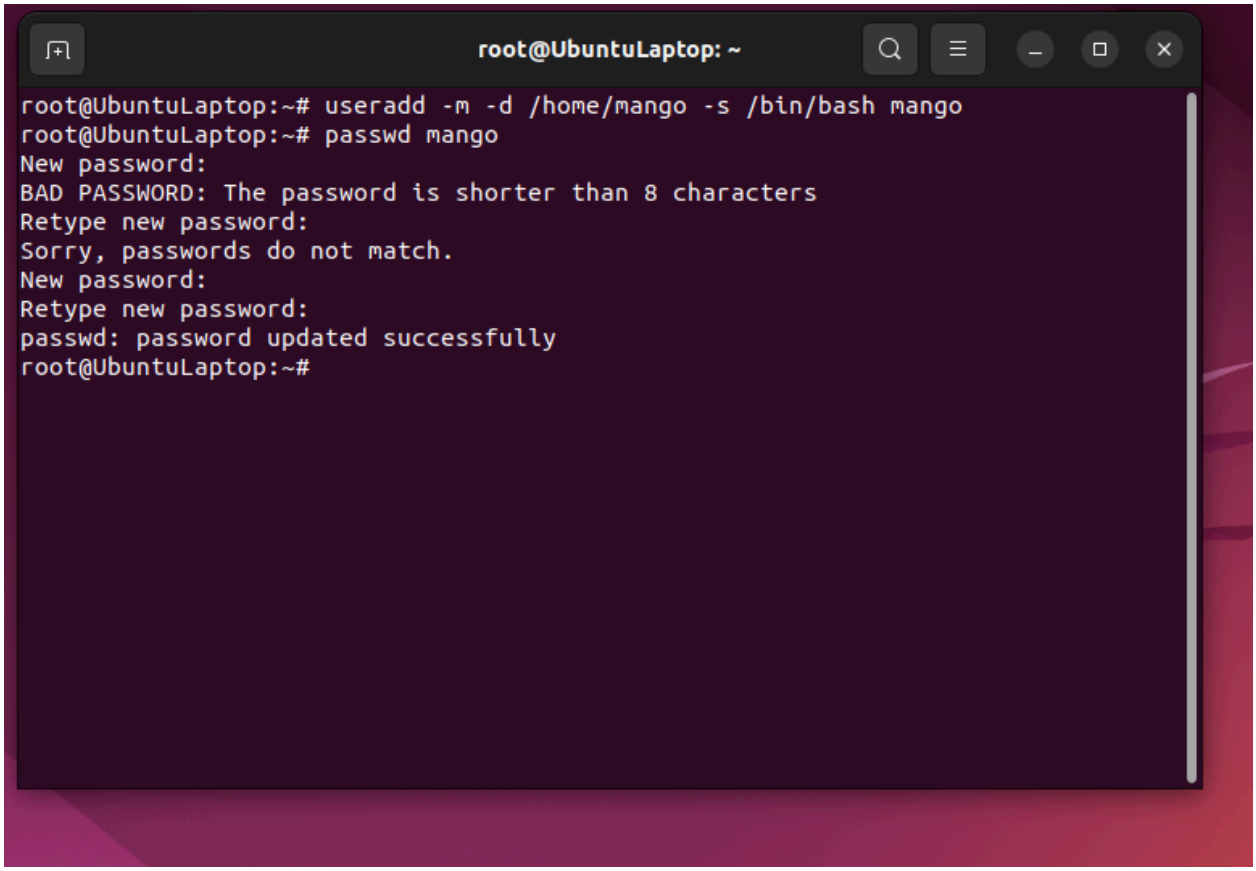
1. Check if the /etc/login.defs config file has HOME_MODE set to 0750.



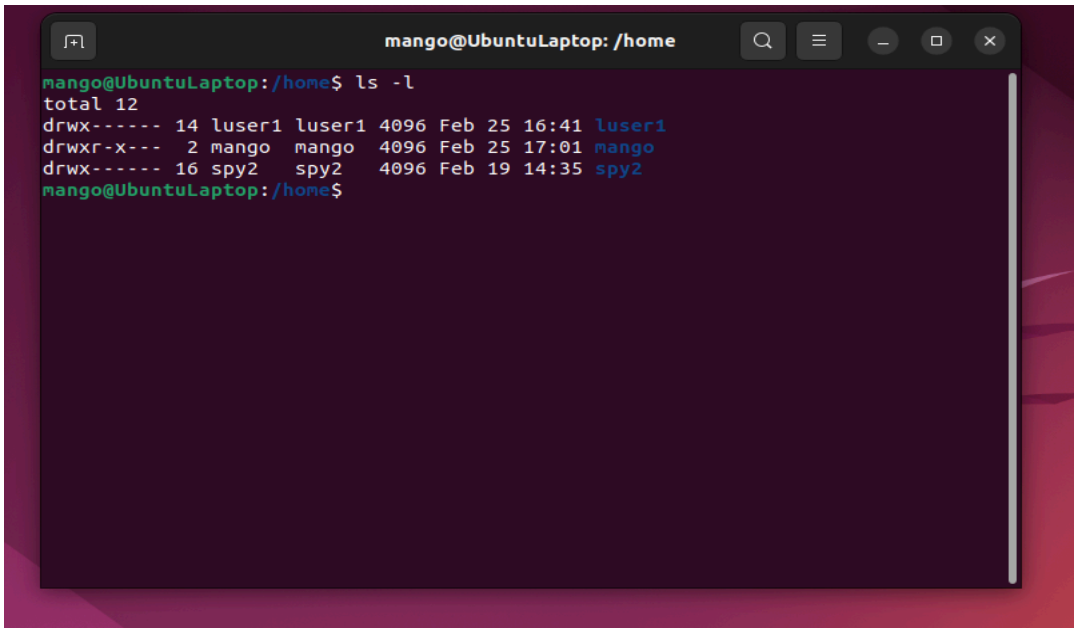
Note: Since it is already set to 0750, I don't have to do anything, new users already have a locked home directory upon creation by default.

2. Let me test this theory to be 100% sure. I will create a new user using useradd. I will then check if they have a protected directory. I created a new user, mango. I created his home directory, set his default shell to bash, and even gave him a password (HelloJello).

Create user Mango, his directory, and specify his shell. Set his Password as well.



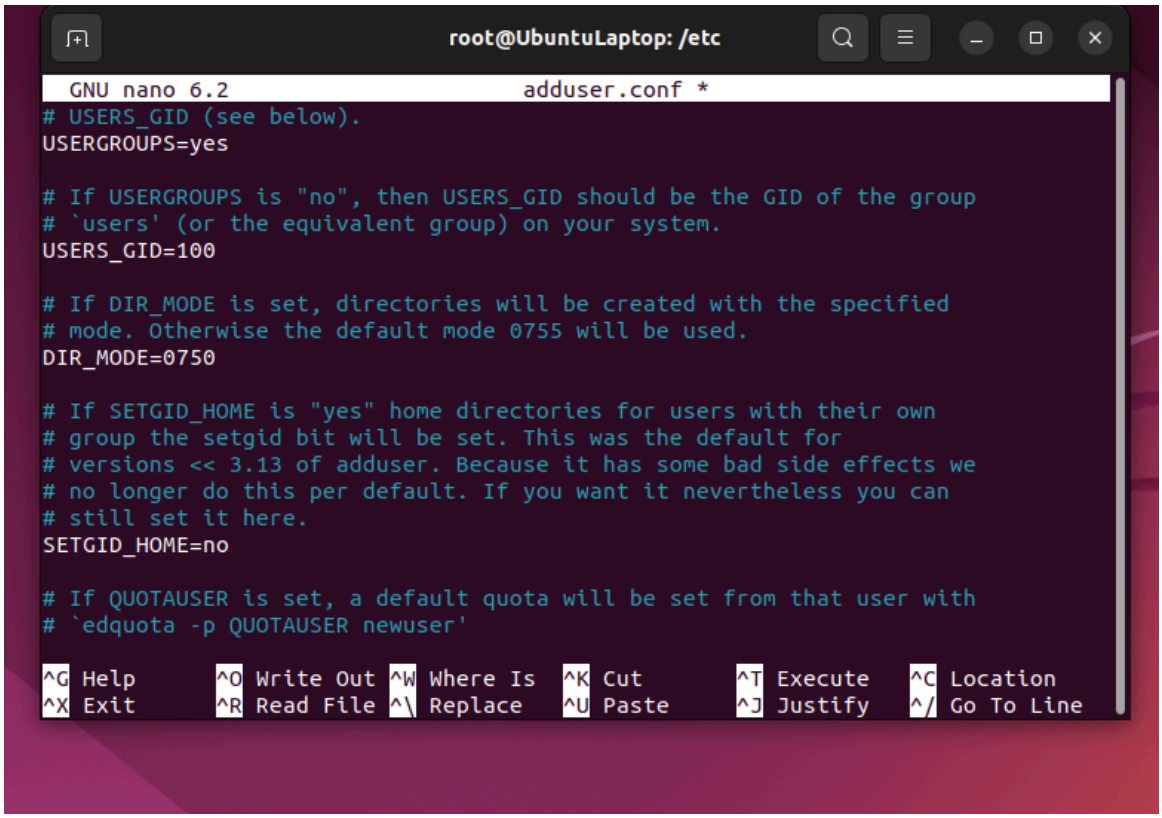
Verify that Mango has a protected directory. (He does, but his group has permission as well which is okay).



So, the HOME_MODE on the /etc/login.defs config file is working.

Solution #2. Change the settings on the /etc/adduser.conf file(for changing the directory permissions based on the adduser command). **Note adduser automatically creates a home directory for users. Again, the settings might be set properly already!**

1. Check if the /etc/adduser.conf file has proper directory protections set. It is already set! DIR_MODE = 0750 which means that only the user and their group can access the directory!



2. Double check this setting by using adduser to create a new user. Created a new user strawberry. Password set to ($h0rtC@k3)

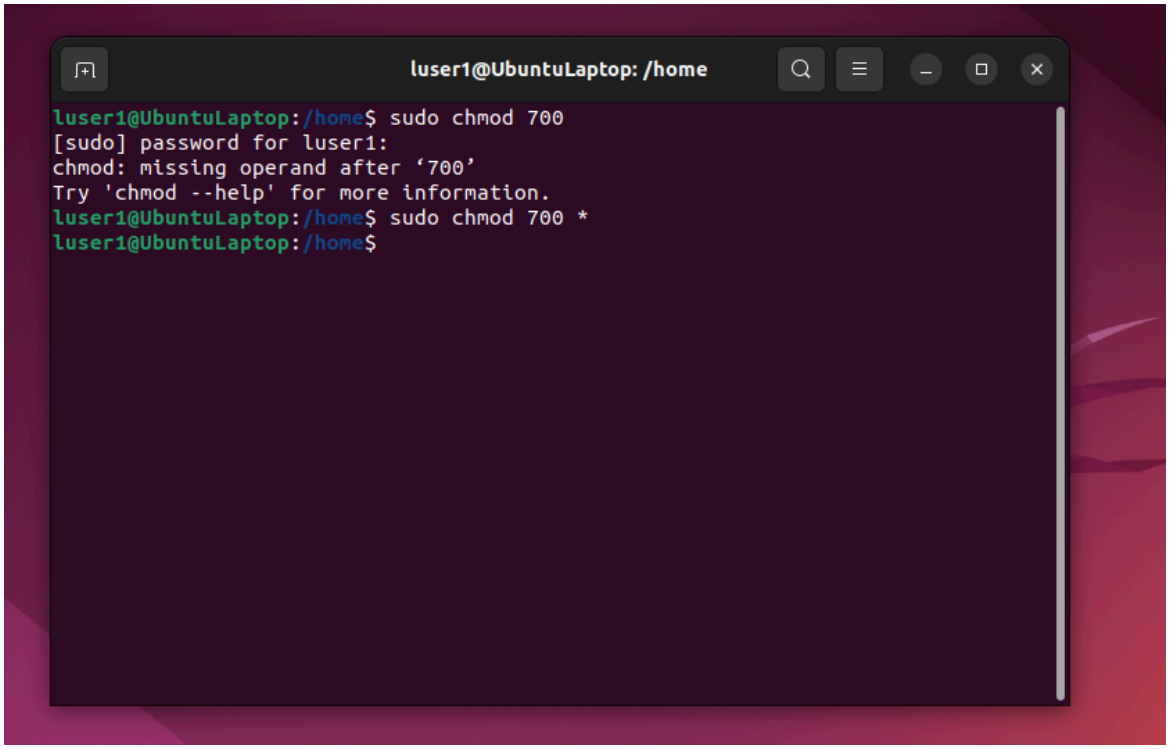
```
root@UbuntuLaptop: ~
root@UbuntuLaptop:~# adduser strawberry
Adding user `strawberry' ...
Adding new group `strawberry' (1003) ...
Adding new user `strawberry' (1003) with group `strawberry' ...
Creating home directory `/home/strawberry' ...
Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a diction
ary word
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for strawberry
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
root@UbuntuLaptop:~#
```

3. Check strawberry's home directory permissions. It worked! Strawberry has a protected directory similar to mango.

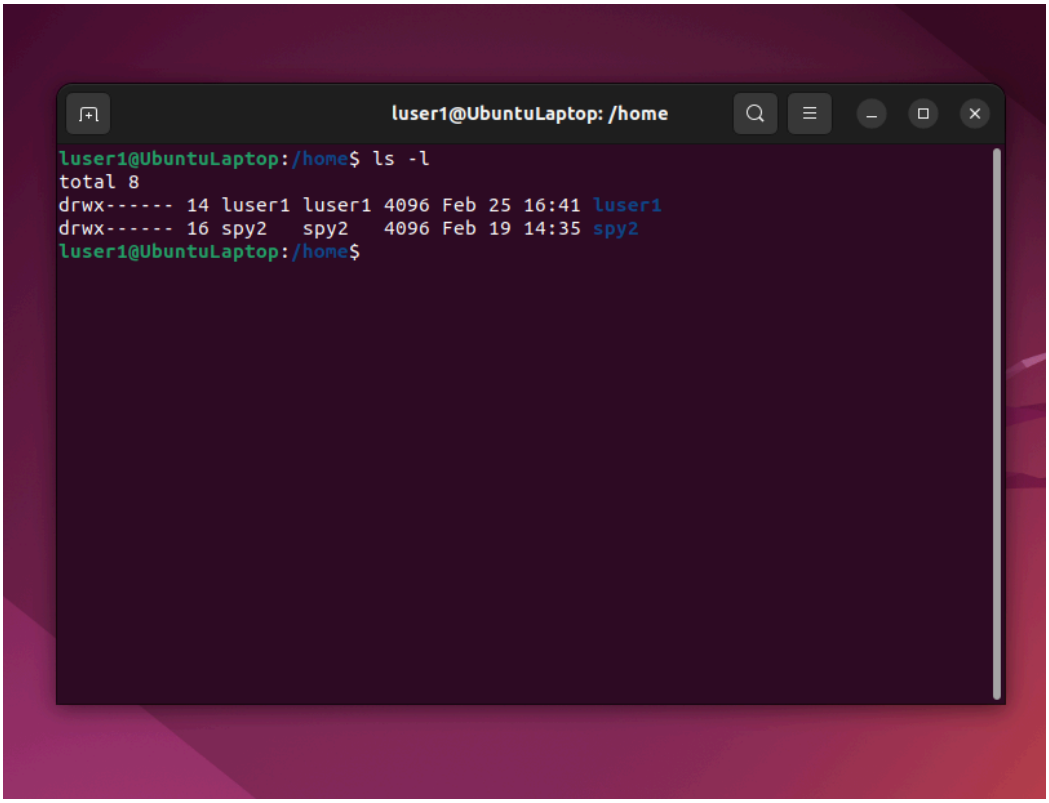
```
strawberry@UbuntuLaptop: /home
strawberry@UbuntuLaptop:/home$ ls -l
total 16
drwx------ 14 luser1     luser1     4096 Feb 25 16:41 luser1
drwxr-x---  2 mango      mango      4096 Feb 25 17:18 mango
drwx------ 16 spy2       spy2       4096 Feb 19 14:35 spy2
drwxr-x---  2 strawberry strawberry 4096 Feb 25 17:19 strawberry
strawberry@UbuntuLaptop:/home$
```

Solution #3. Manually change the directory permissions.

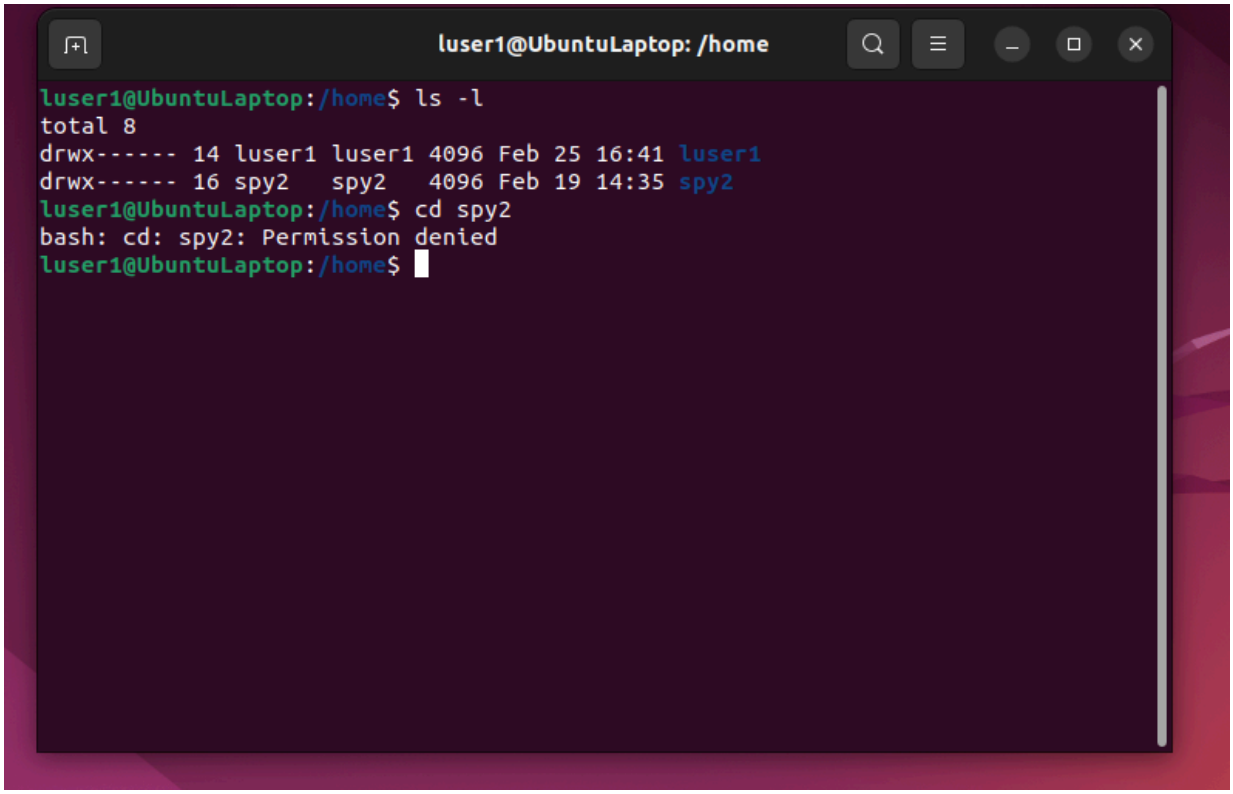
1. Use chmod and grant full permissions to the owners of the home directories only. Note that this method can only be used on users post creation! Also if you want really strict home directory protection.



2. Verify that the owners of the home directories are the only ones with permission to access them.



3. Test it further. I will try to access spy2's directory as luser1. This is what happens.



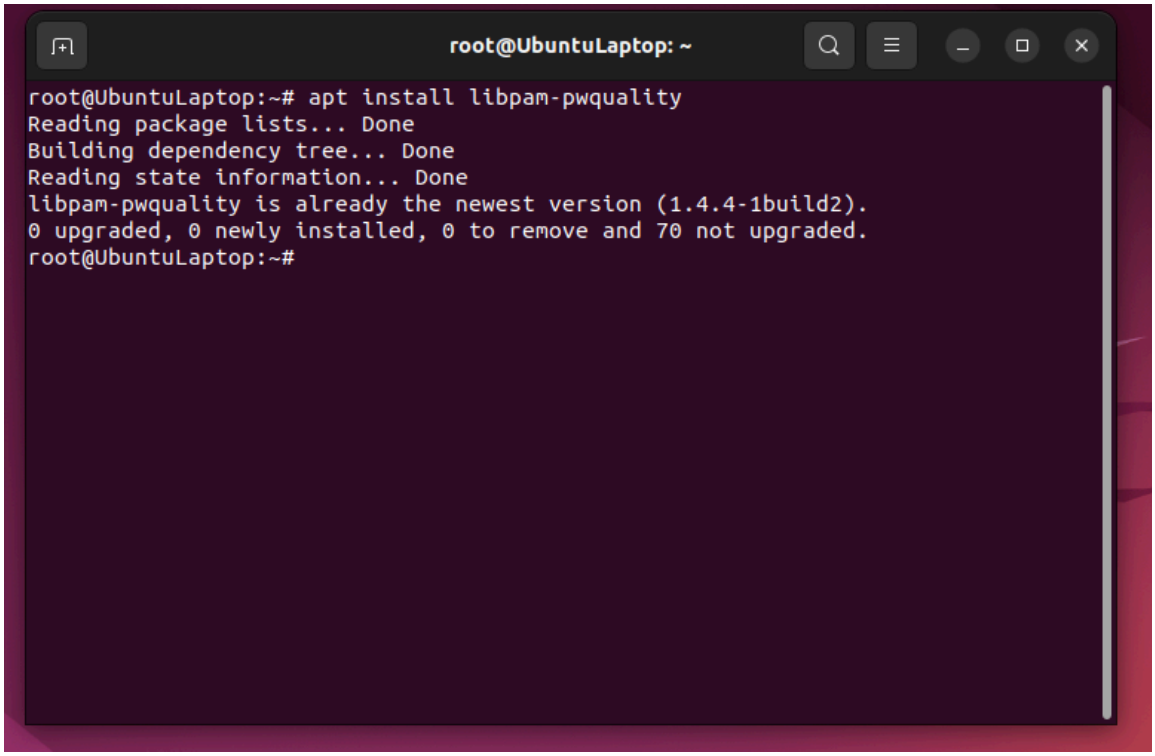
Note: Success! I cannot access spy2's directory as luser1, meaning that each user has a protected home directory.

Section #2: Enforcing strong password criteria.

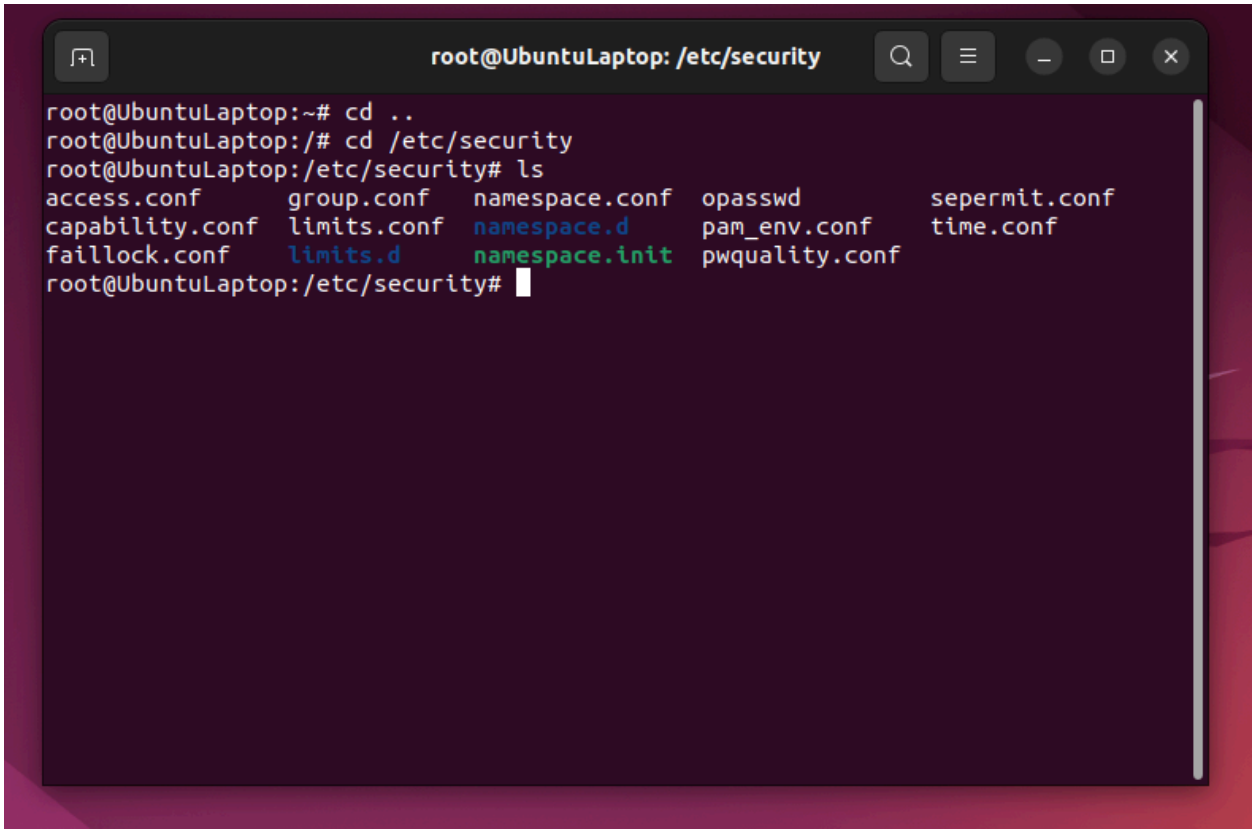
Issue: Not enforcing password criteria can put your Linux machines at risk of brute-force and various other attacks. Enforcing strong password criteria ensures that all users on the machine follow the same strength requirements, thus making sure that no one on the machine can be an easy target for attackers.

Solution: Install the libpam pwquality utility on your linux machine. Then set password criteria for all users on the machine.

1. I already had pw-quality installed. You can download it by using the following command:



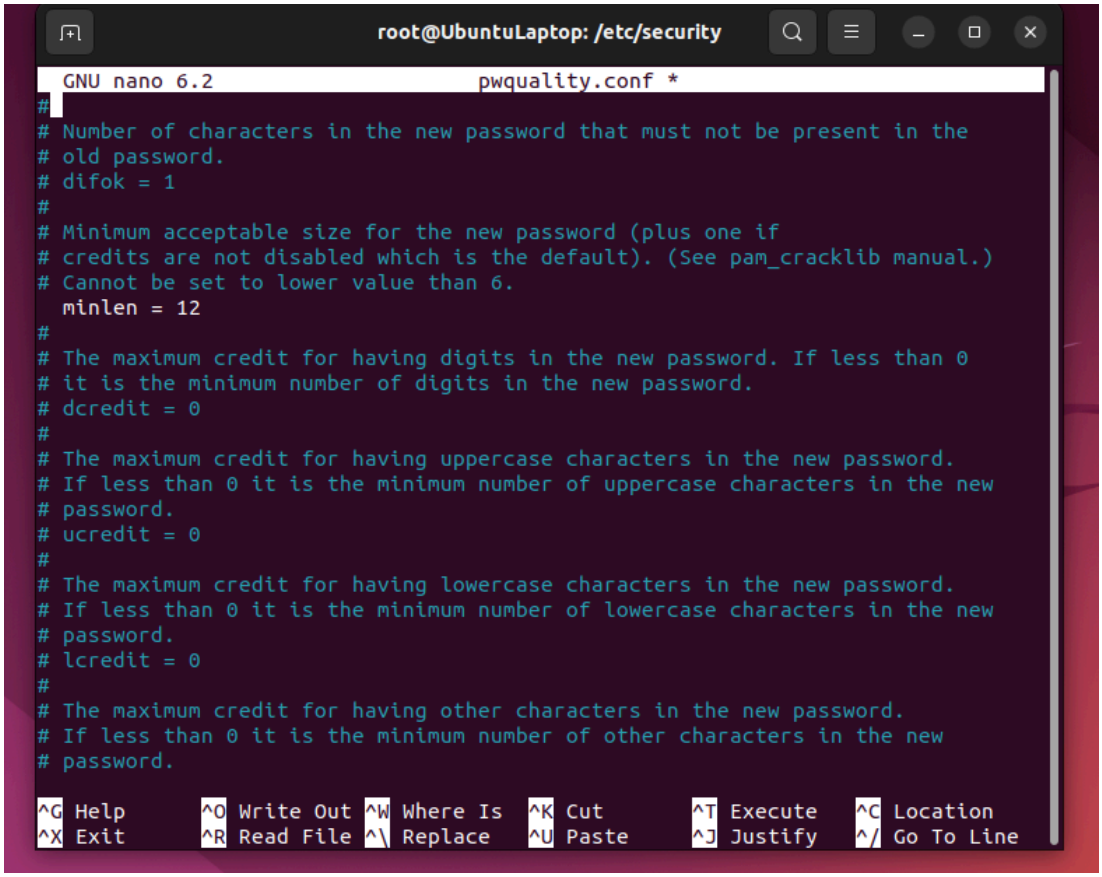
2 a. Now let us look at the pwquality.conf file and set stronger policies on my machine. It can be found in the /etc/security directory.



2 b. Open up the pwquality.conf file using your text editor of choice. I will change the settings so that my minimum password length for new users will be 12. I will also enforce other qualities such as # of retries before getting an error, and the number of different characters required for a password from each class. For everything to work, I must uncomment the parameter!

My changes:

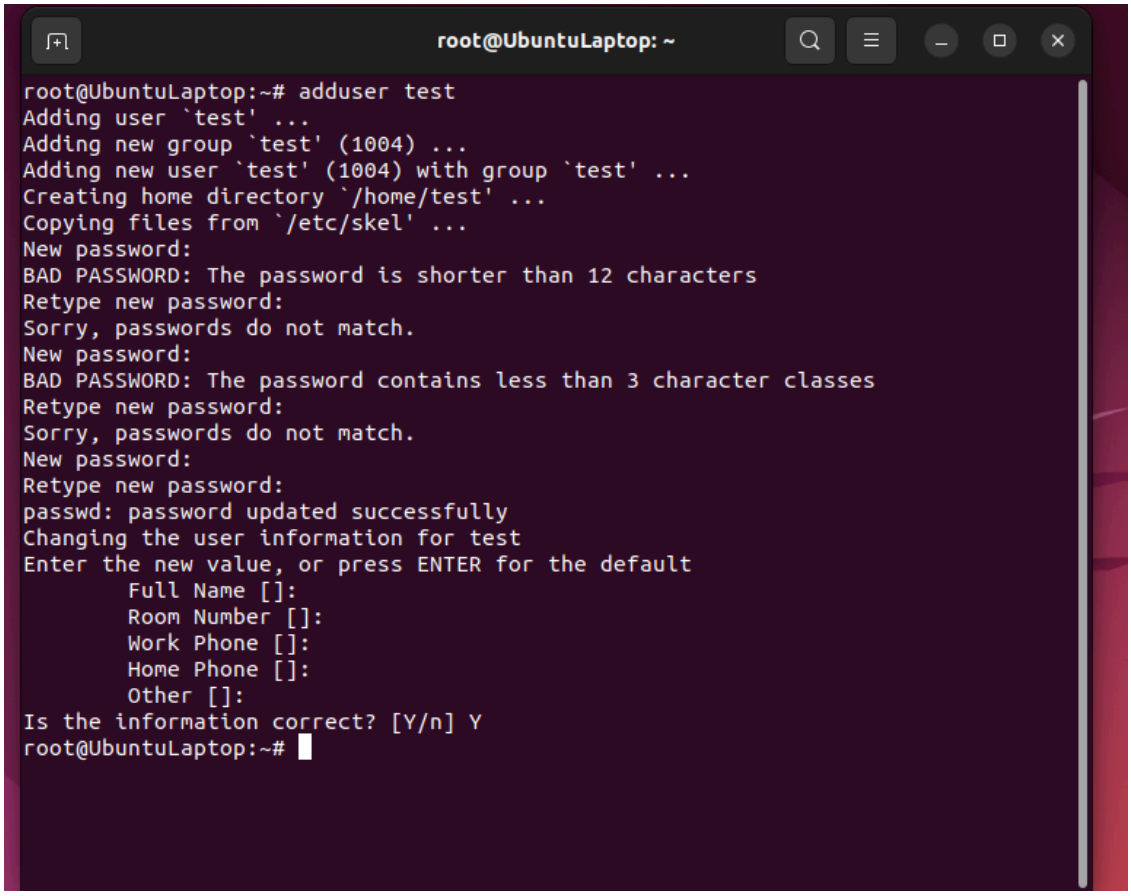
1. Minlen =12 (Passwords must be 12 characters or longer)
2. Retry = 2
3. Minclass = 3 (upper, lower, and digit required)



```
root@UbuntuLaptop: /etc/security
  GNU nano 6.2                    pwquality.conf *
# ucredit = 0
#
# The maximum credit for having lowercase characters in the new password.
# If less than 0 it is the minimum number of lowercase characters in the new
# password.
# lcredit = 0
#
# The maximum credit for having other characters in the new password.
# If less than 0 it is the minimum number of other characters in the new
# password.
# ocredit = 0
#
# The minimum number of required classes of characters for the new
# password (digits, uppercase, lowercase, others).
  minclass = 3
#
# The maximum number of allowed consecutive same characters in the new password.
# The check is disabled if the value is 0.
# maxrepeat = 0
#
# The maximum number of allowed consecutive characters of the same class in the
# new password.
# The check is disabled if the value is 0.
# maxclassrepeat = 0
#
# Whether to check for the words from the passwd entry GECOS string of the user.
# The check is enabled if the value is not 0.

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

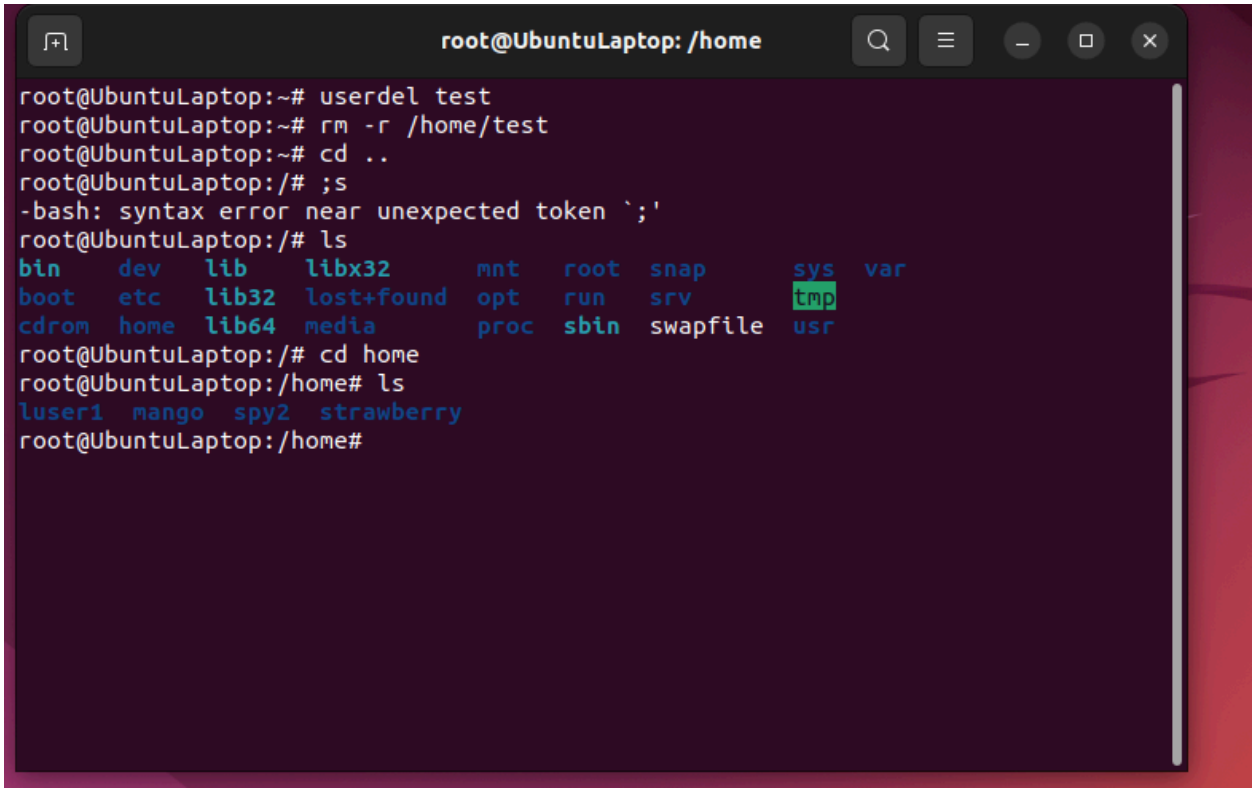
```
root@UbuntuLaptop: /etc/security
  GNU nano 6.2                    pwquality.conf *
# The check is enabled if the value is not 0.
# usercheck = 1
#
# Length of substrings from the username to check for in the password
# The check is enabled if the value is greater than 0 and usercheck is enabled.
# usersubstr = 0
#
# Whether the check is enforced by the PAM module and possibly other
# applications.
# The new password is rejected if it fails the check and the value is not 0.
# enforcing = 1
#
# Path to the cracklib dictionaries. Default is to use the cracklib default.
# dictpath =
#
# Prompt user at most N times before returning with error. The default is 1.
  retry = 2
#
# Enforces pwquality checks on the root user password.
# Enabled if the option is present.
# enforce_for_root
#
# Skip testing the password quality for users that are not present in the
# /etc/passwd file.
# Enabled if the option is present.
# local_users_only

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

3. Now I will save the file, then test my conditions on a new user.



Note: My first attempt was using helloworld. Second attempt was using helloworld1234 (only lower and digits). Last attempt was HelloWorld1234 (lower, upper, and digits so 3 classes).

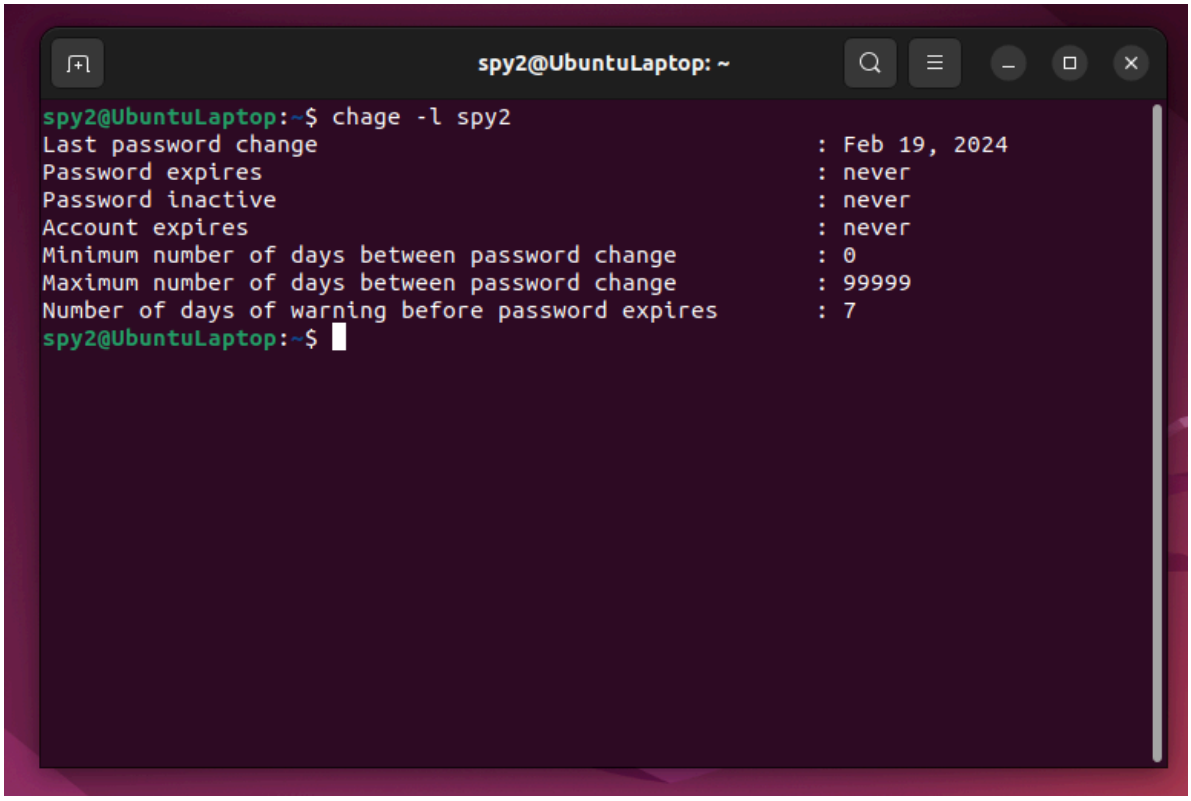
Bonus: Deleting a user on Linux (deleting test). Userdel User. Then get rid of the user's home directory by using rm -r. Then I verified that the user's home directory was deleted.



Section #3: Setting/ enforcing password expiration.

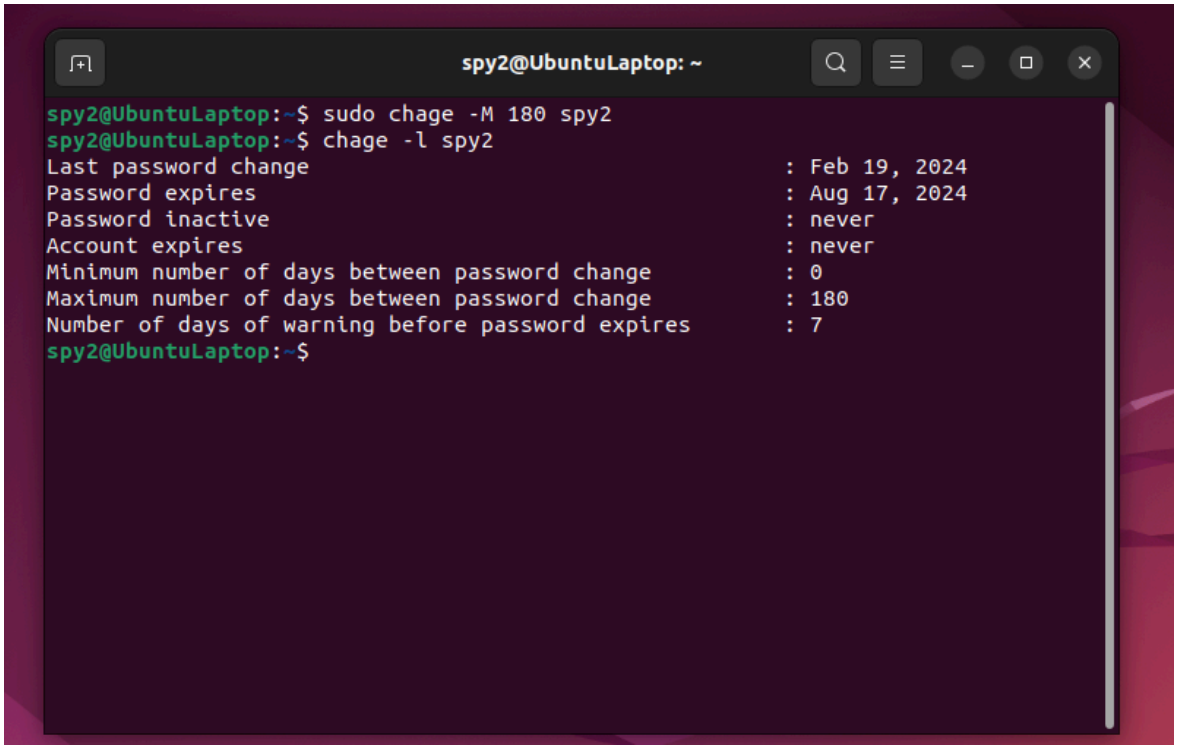
Issue: You shouldn't use the same password for many years since the likelihood of it being found and cracked increases. Instead, you should set a password expiration date so that your password would change every six months to a year. Account expiration is important as well, especially in a work setting. People come and go, and you shouldn't keep old users on an account indefinitely. Lastly, in the event that an attacker tries to brute force a user's account, locking that account would be useful. Unlocking accounts is also useful in situations where a user forgot their password and thus exceeded the attempts limit.

Solution #1: Check user account expiry data. Use chage -l username.

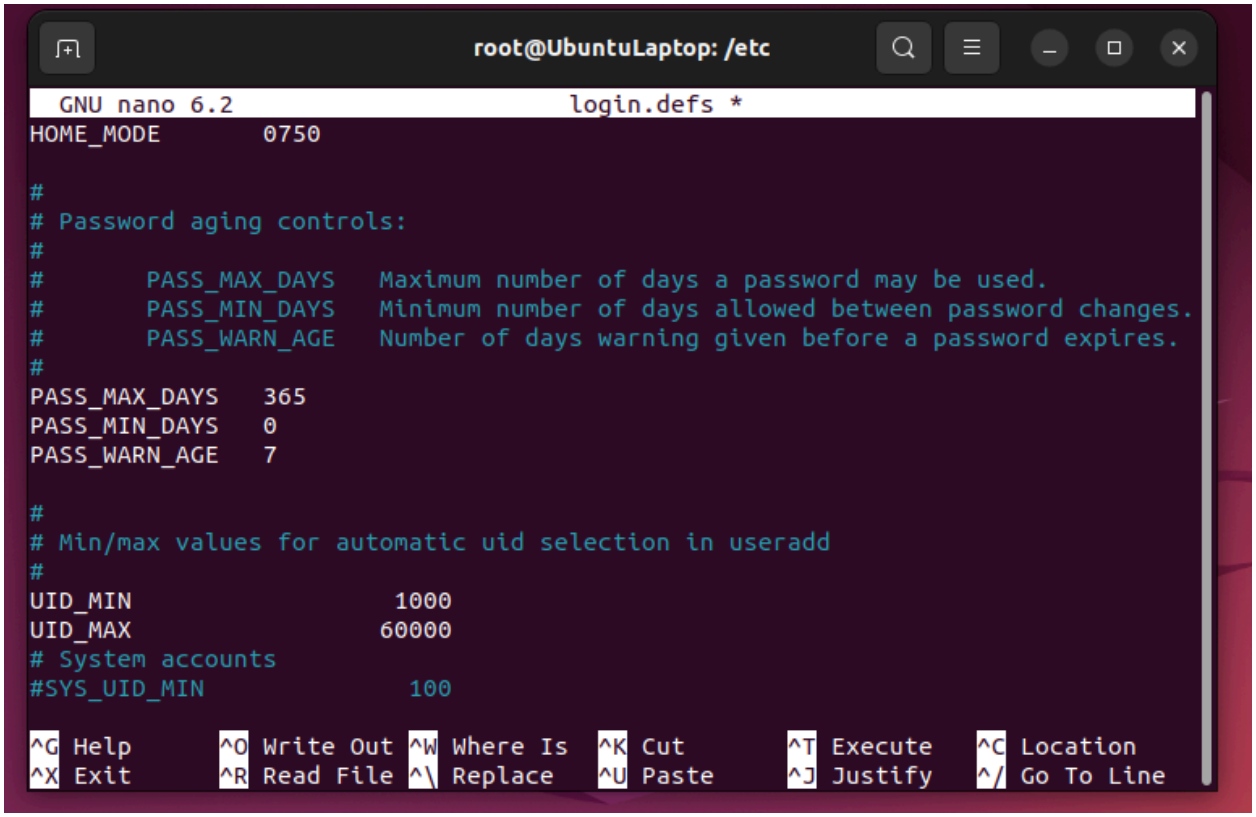


Spy2 does not have a password expiry date or account expiry date. Let us assume that since he is a spy he needs to change his password every six months (twice a year).

2. Set expiry date for spy2 password. Used chage -M 180 spy2. I assumed that 180 days would roughly equal six months.

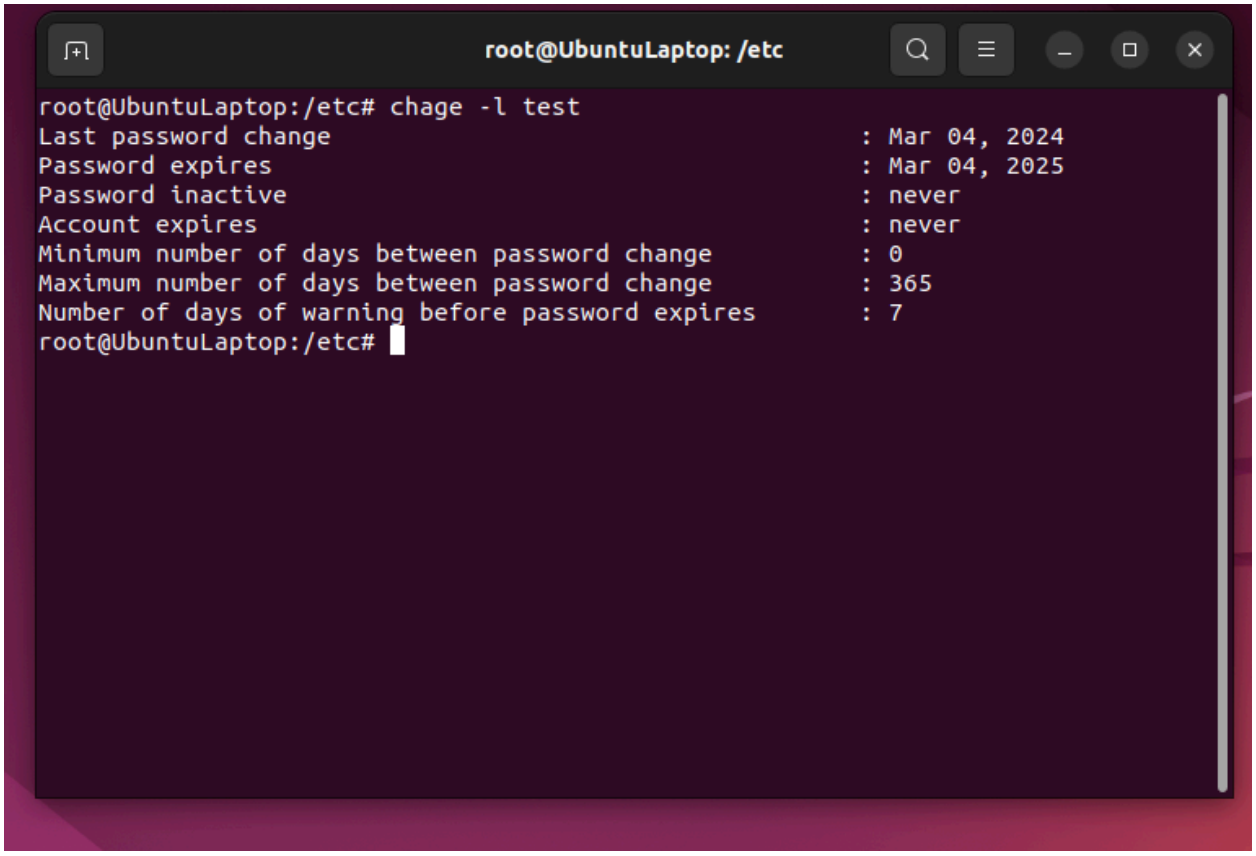


Solution #2: Set account wide password expiry data by modifying the /etc/login.defs file.

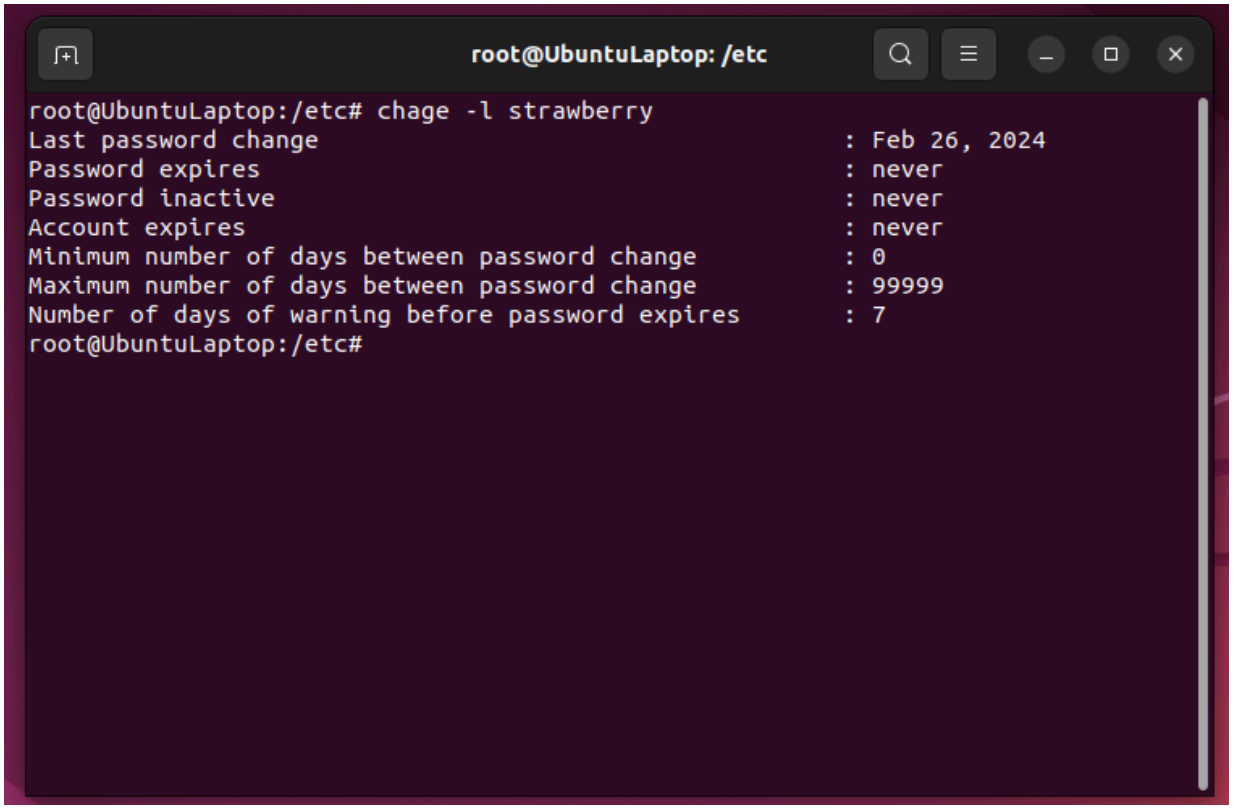


Now the default account password will be used for only a year (365 days) and users will need to change their passwords yearly. Now let us verify this by checking two user accounts.

Account #1: test



Account #2: strawberry



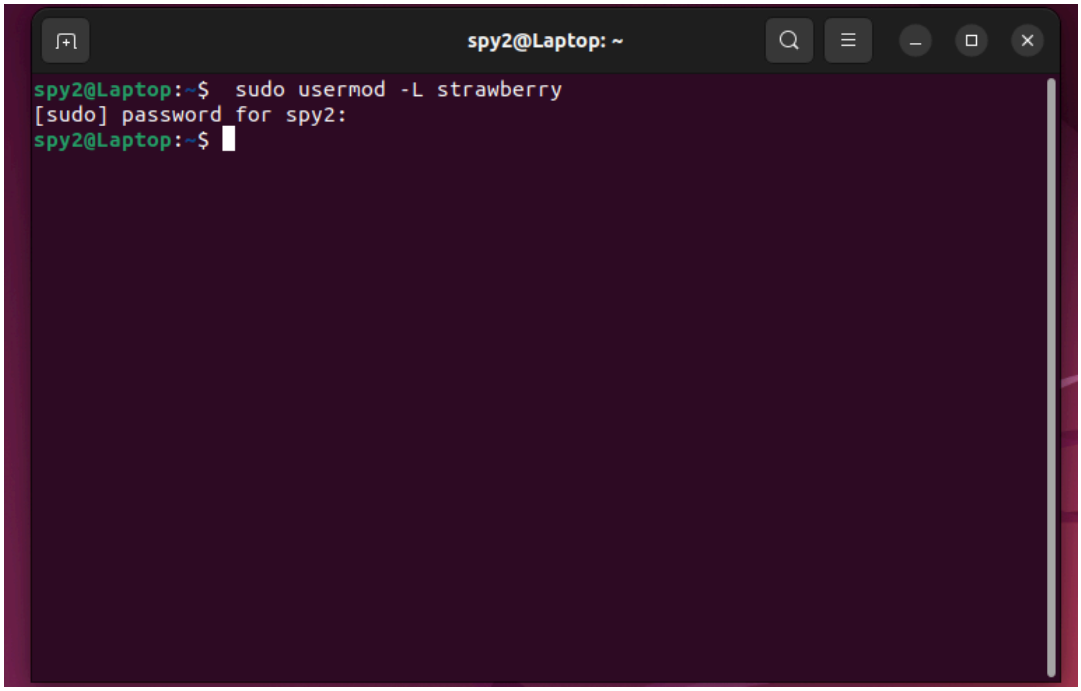
There is a difference between strawberry and test. Why? Well, strawberry was created before I changed the login.defs file. Test was created after, so the 1 year reset policy came into effect for him only.

Section #4: Locking/ Unlocking User Accounts

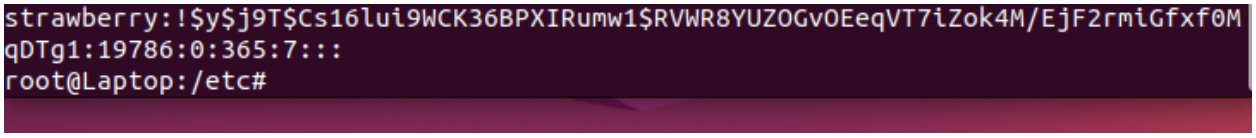
Issue: Sometimes user accounts should be locked down. If a user is on vacation and has sudo privileges, then their account should be locked so that any attackers or other curious employees can't attempt to use the elevated privilege. Or say a user was caught doing something suspicious and you don't want them to have access to the system. Either way, you do not want a user's account to be accessible. Expiring accounts is important as well. Say you hire a contractor for 1 year. You can automatically expire them after a year so that you don't forget to remove them from the system.

Solution #1: Locking a user account

Method #1. Use Usermod -L (must have root privileges)



Note that if you look in the /etc/shadow file (where user info like password hashes are) the locked user Strawberry's password will have a ! in front of their hashed password. This exclamation point essentially blocks the system from reading the hashed password effectively making it impossible to authenticate the user.



Method #2 Use passwd -l (must have root privileges as well)

```
root@Laptop: ~
root@Laptop:~# passwd -l mango
passwd: password expiry information changed.
root@Laptop:~#
```

Again, we can verify that this worked by looking in the /etc/shadow file contents:

```
mango:!$y$j9T$L/WkLLaF3N1/PszI3ZsOl0$BocoGjqMO4RMYE9.BgtIWcB0d7WjHudSSk0EsYwoqT1
:19792:0:99999:7:::
root@Laptop:/etc#
```

The ! is in front of his hashed password.

Solution #2 Unlocking Accounts

Method #1. Use usermod -U

```
root@Laptop: ~
root@Laptop:~# usermod -U strawberry
root@Laptop:~#
```

Let us verify that Strawberry is now able to login and be authenticated. We can do that by looking in the /etc/shadow file.

```
strawberry:$y$j9T$Cs16lui9WCK36BPXIRumw1$RVWR8YUZOGvOEeqVT7iZok4M/EjF2rmiGfxf0MqDTg1:19786:0:365:7:::
root@Laptop:/etc#
```

And yes, the ! is gone!

Method #2 Use passwd -u

```
root@Laptop: ~
root@Laptop:~# passwd -u mango
passwd: password expiry information changed.
root@Laptop:~#
```

Now let us confirm that Mango's account is unlocked in the /etc/shadow file:

```
mango:$y$j9T$L/WkLLaF3N1/PszI3ZsOl0$BocoGjqMO4RMYE9.BgtIWcB0d7WjHudSSk0EsYwoqT1:
19792:0:99999:7:::
root@Laptop:/etc#
```