

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:

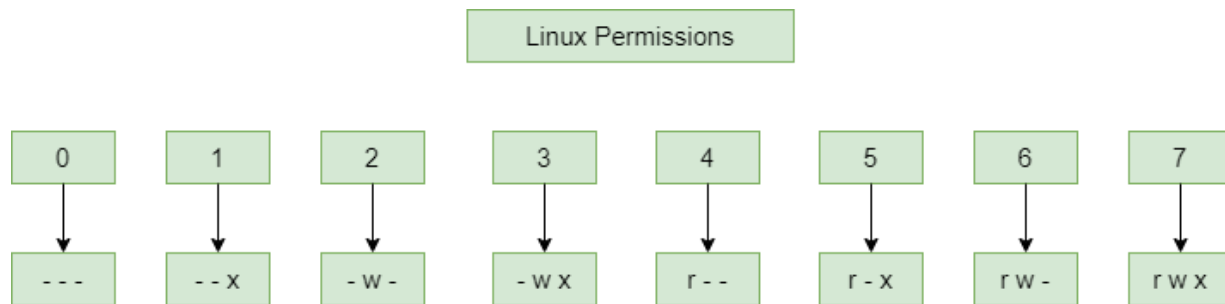
1. Locking down user home directories.
2. Enforcing strong password criteria
3. Setting/ enforcing password, account expiration.
4. Locking user accounts

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.

```
root@UbuntuLaptop: /etc
GNU nano 6.2 login.defs *
#
# If USERGROUPS_ENAB is set to "yes", that will modify this UMASK default value
# for private user groups, i. e. the uid is the same as gid, and username is
# the same as the primary group name: for these, the user permissions will be
# used as group permissions, e. g. 022 will become 002.
#
# Prefix these values with "0" to get octal, "0x" to get hexadecimal.
#
ERASECHAR      0177
KILLCHAR       025
UMASK          022

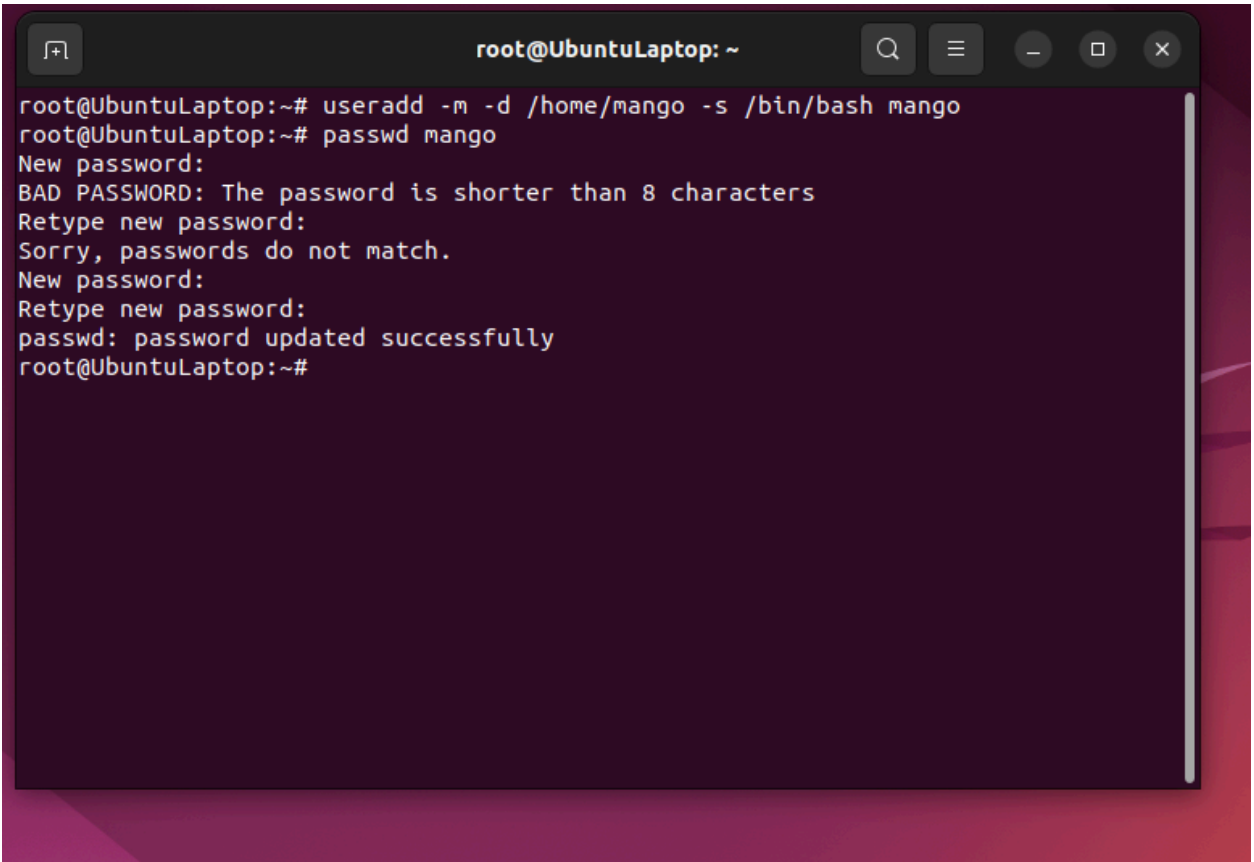
# HOME_MODE is used by useradd(8) and newusers(8) to set the mode for new
# home directories.
# If HOME_MODE is not set, the value of UMASK is used to create the mode.
HOME_MODE      0750

#
# Password aging controls:
#
[Cancelled]
^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
```

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.

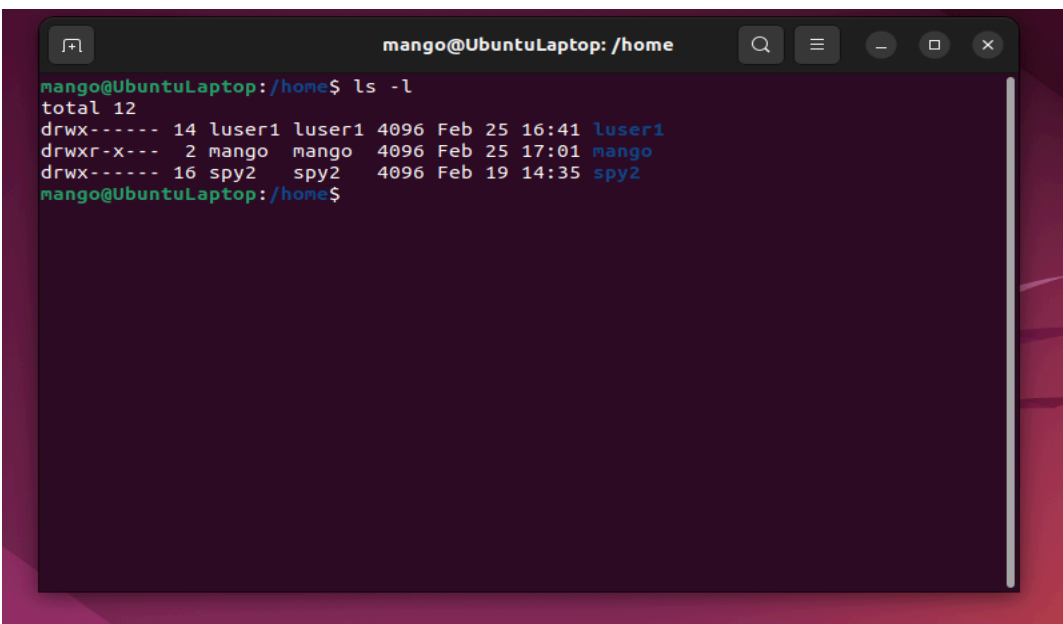
2. Let me test this theory to be 100% sure. I will create a new user using useradd. I will then check if they have a protected directory. I created a new user, mango. I created his home directory, set his default shell to bash, and even gave him a password (HelloJello).

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

A terminal window titled 'root@UbuntuLaptop: ~' with standard Ubuntu window controls. The terminal shows the execution of 'useradd -m -d /home/mango -s /bin/bash mango' followed by 'passwd mango'. The password setting process includes prompts for 'New password:', an error for 'BAD PASSWORD: The password is shorter than 8 characters', a 'Retype new password:' prompt, a mismatch error 'Sorry, passwords do not match.', another 'Retype new password:' prompt, and finally 'passwd: password updated successfully'.

```
root@UbuntuLaptop:~# useradd -m -d /home/mango -s /bin/bash mango
root@UbuntuLaptop:~# passwd mango
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: password updated successfully
root@UbuntuLaptop:~#
```

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

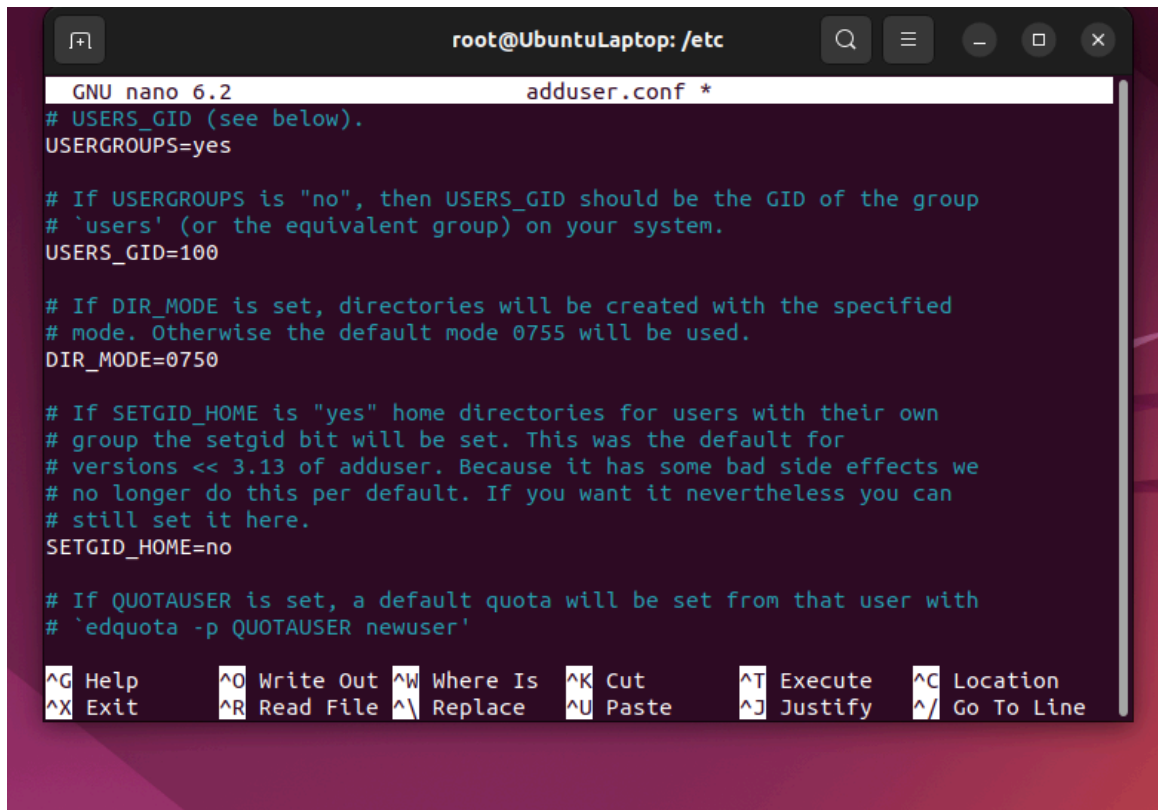
A terminal window titled 'mango@UbuntuLaptop: /home' with standard Ubuntu window controls. The terminal shows the command 'ls -l' being executed in the /home directory. The output lists three directories: 'luser1' owned by 'luser1', 'mango' owned by 'mango', and 'spy2' owned by 'spy2'. All three directories have permissions 'drwxr-xr-x' (755).

```
mango@UbuntuLaptop:/home$ ls -l
total 12
drwxr-xr-x 14 luser1 luser1 4096 Feb 25 16:41 luser1
drwxr-xr-x  2 mango  mango  4096 Feb 25 17:01 mango
drwxr-xr-x 16 spy2   spy2   4096 Feb 19 14:35 spy2
mango@UbuntuLaptop:/home$
```

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!**



The screenshot shows a terminal window titled 'root@UbuntuLaptop: /etc' with a nano 6.2 editor open to the file 'adduser.conf'. The file contains configuration settings for the adduser command. The visible content is as follows:

```
GNU nano 6.2 adduser.conf *
# USERS_GID (see below).
USERGROUPS=yes

# If USERGROUPS is "no", then USERS_GID should be the GID of the group
# 'users' (or the equivalent group) on your system.
USERS_GID=100

# If DIR_MODE is set, directories will be created with the specified
# mode. Otherwise the default mode 0755 will be used.
DIR_MODE=0750

# If SETGID_HOME is "yes" home directories for users with their own
# group the setgid bit will be set. This was the default for
# versions < 3.13 of adduser. Because it has some bad side effects we
# no longer do this per default. If you want it nevertheless you can
# still set it here.
SETGID_HOME=no

# If QUOTAUSER is set, a default quota will be set from that user with
# 'edquota -p QUOTAUSER newuser'
```

At the bottom of the terminal, there is a status bar with various keyboard shortcuts: ^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.

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 dictionary 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:~#
```

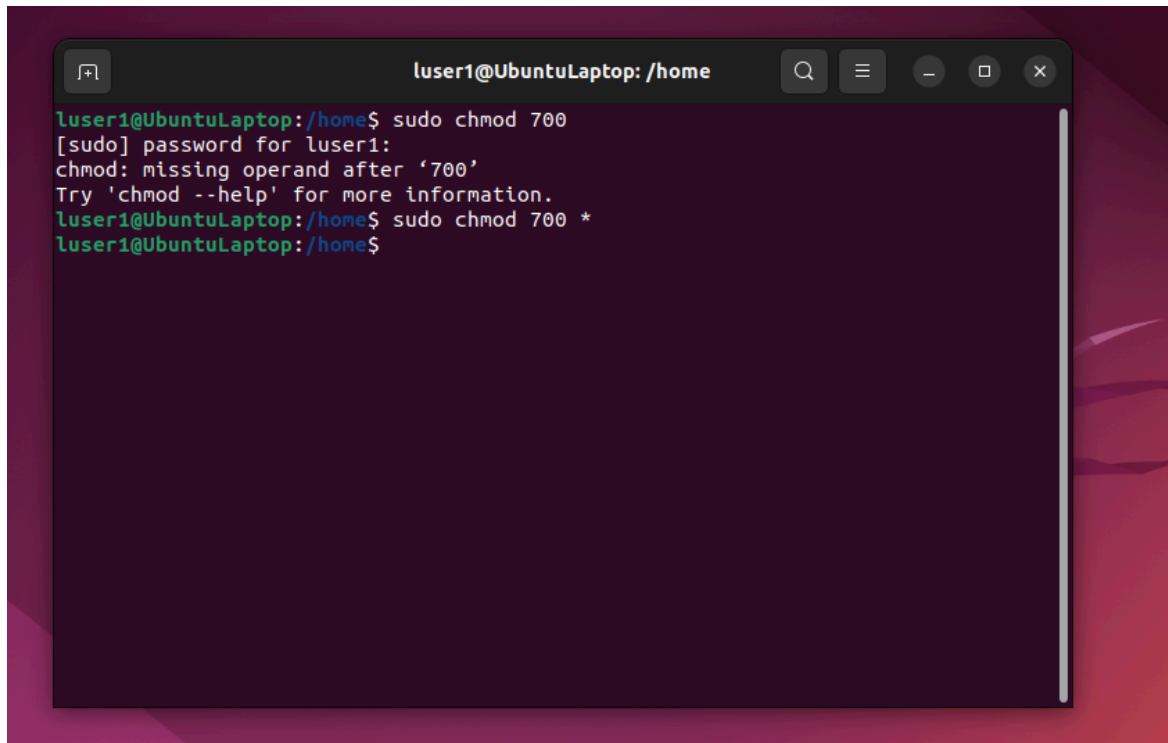
3. Check strawberry's home directory permissions. It worked! Strawberry has a protected directory similar to mango.

```
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.

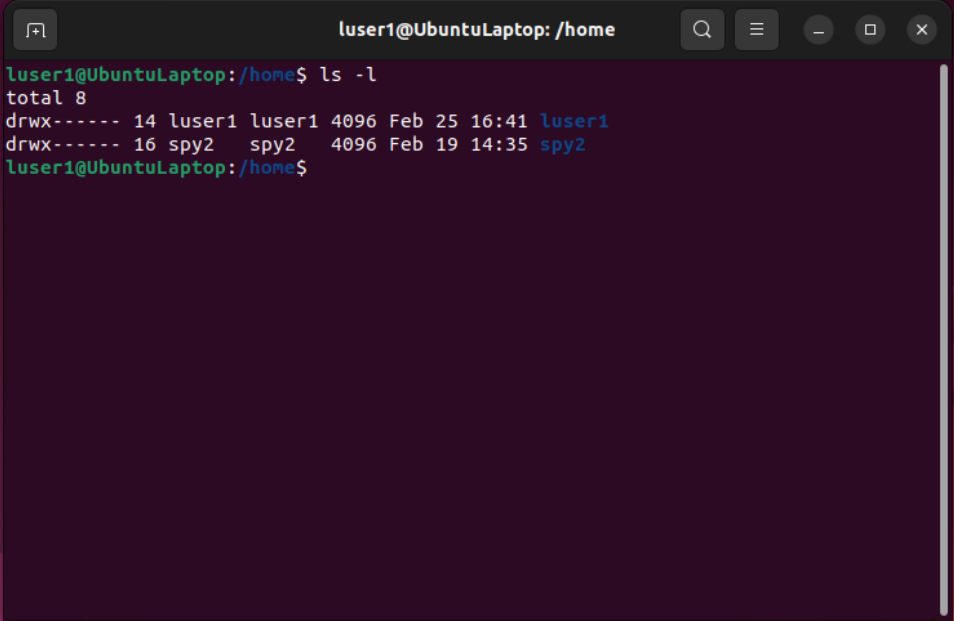
Note that this method can only be used on users post creation! Also if you want really strict home directory protection.



```
luser1@UbuntuLaptop: /home
luser1@UbuntuLaptop:/home$ sudo chmod 700
[sudo] password for luser1:
chmod: missing operand after '700'
Try 'chmod --help' for more information.
luser1@UbuntuLaptop:/home$ sudo chmod 700 *
luser1@UbuntuLaptop:/home$
```

The image shows a terminal window titled 'luser1@UbuntuLaptop: /home'. The user enters the command 'sudo chmod 700'. The system prompts for the password for 'luser1'. After the password is entered, an error message is displayed: 'chmod: missing operand after '700''. The user then enters 'Try 'chmod --help' for more information.' followed by the corrected command 'sudo chmod 700 *'. The command is executed successfully, and the prompt returns to 'luser1@UbuntuLaptop:/home\$'.

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

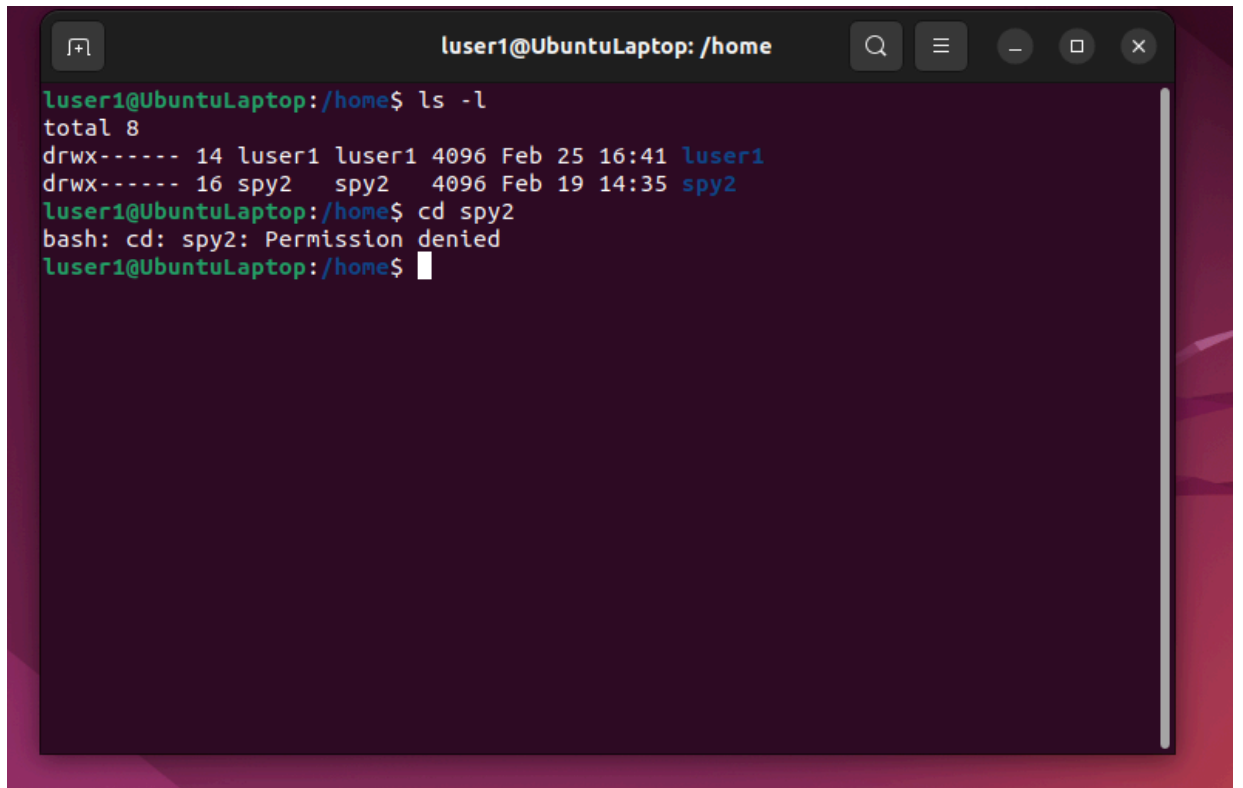


A terminal window titled 'luser1@UbuntuLaptop: /home' is shown. The user 'luser1' has executed the command 'ls -l'. The output shows two directories: 'luser1' and 'spy2'. The 'luser1' directory is owned by 'luser1' and has permissions 'drwx-----'. The 'spy2' directory is owned by 'spy2' and has permissions 'drwx-----'. The terminal window has a dark background and standard window controls.

```
luser1@UbuntuLaptop:/home$ ls -l
total 8
drwx----- 14 luser1 luser1 4096 Feb 25 16:41 luser1
drwx----- 16 spy2    spy2   4096 Feb 19 14:35 spy2
luser1@UbuntuLaptop:/home$
```

them.

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

A terminal window titled 'luser1@UbuntuLaptop: /home' with standard Ubuntu window controls. The terminal shows the command 'ls -l' being executed, displaying the permissions for two directories: 'luser1' and 'spy2'. Both are owned by their respective users and have permissions 'drwx-----'. The user 'luser1' then attempts to run 'cd spy2', which results in the error 'bash: cd: spy2: Permission denied'.

```
luser1@UbuntuLaptop:/home$ ls -l
total 8
drwx----- 14 luser1 luser1 4096 Feb 25 16:41 luser1
drwx----- 16 spy2    spy2   4096 Feb 19 14:35 spy2
luser1@UbuntuLaptop:/home$ cd spy2
bash: cd: spy2: Permission denied
luser1@UbuntuLaptop:/home$
```

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:


```
root@UbuntuLaptop: ~  
root@UbuntuLaptop:~# apt install libpam-pwquality  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
libpam-pwquality is already the newest version (1.4.4-1build2).  
0 upgraded, 0 newly installed, 0 to remove and 70 not upgraded.  
root@UbuntuLaptop:~#
```

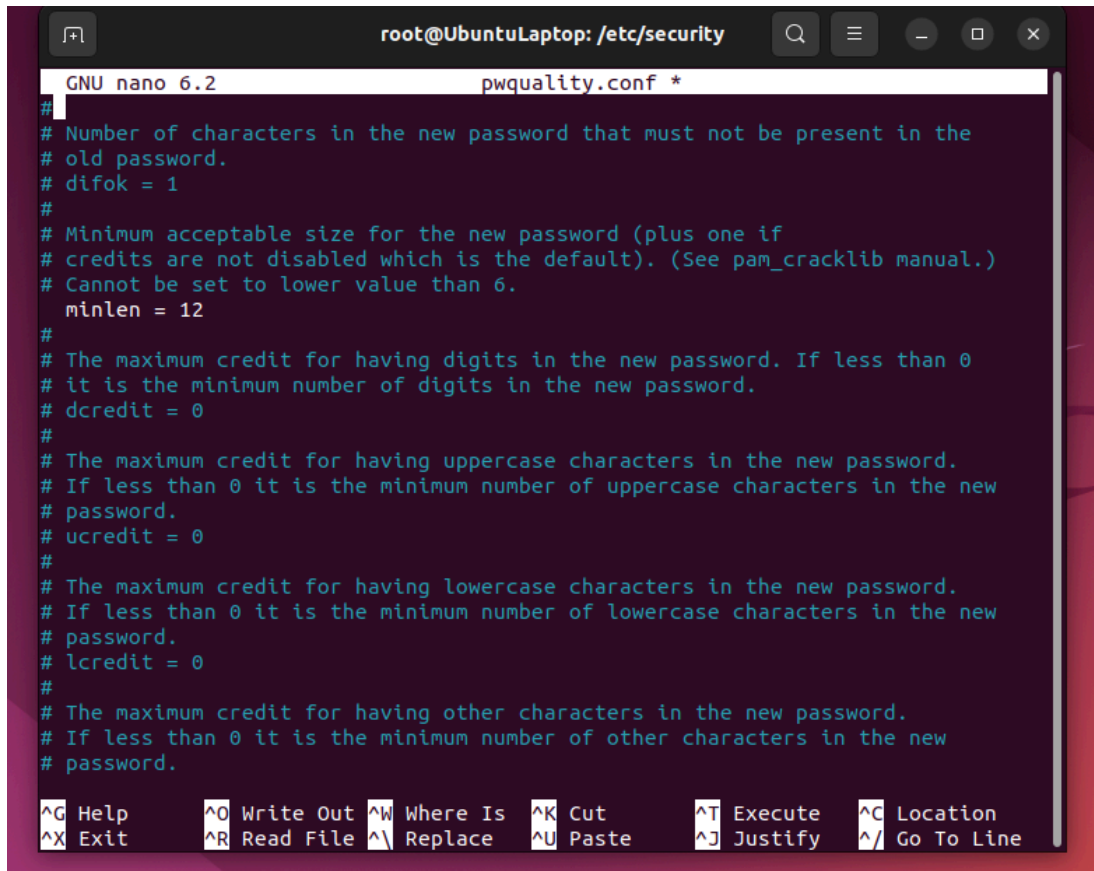
- 2 a. Now let us look at the pwquality.conf file and set stronger policies on my machine.
It can be found in the /etc/security directory.

```
root@UbuntuLaptop: /etc/security  
root@UbuntuLaptop:~# cd ../  
root@UbuntuLaptop:/# cd /etc/security  
root@UbuntuLaptop:/etc/security# ls  
access.conf      group.conf      namespace.conf  opasswd         sepermit.conf  
capability.conf  limits.conf     namespace.d     pam_env.conf    time.conf  
faillock.conf    limits.d        namespace.init  pwquality.conf  
root@UbuntuLaptop:/etc/security#
```

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)



The screenshot shows a terminal window with the title bar 'root@UbuntuLaptop: /etc/security'. The window contains the 'pwquality.conf' file being edited with 'GNU nano 6.2'. The file content is as follows:

```
#
# Number of characters in the new password that must not be present in the
# old password.
# difok = 1
#
# Minimum acceptable size for the new password (plus one if
# credits are not disabled which is the default). (See pam_cracklib manual.)
# Cannot be set to lower value than 6.
minlen = 12
#
# The maximum credit for having digits in the new password. If less than 0
# it is the minimum number of digits in the new password.
# dcredit = 0
#
# The maximum credit for having uppercase characters in the new password.
# If less than 0 it is the minimum number of uppercase characters in the new
# password.
# 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.
```

At the bottom of the terminal, there is a status bar with various keyboard shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^\ Replace, ^U Paste, ^J Justify, and ^_ Go To Line.

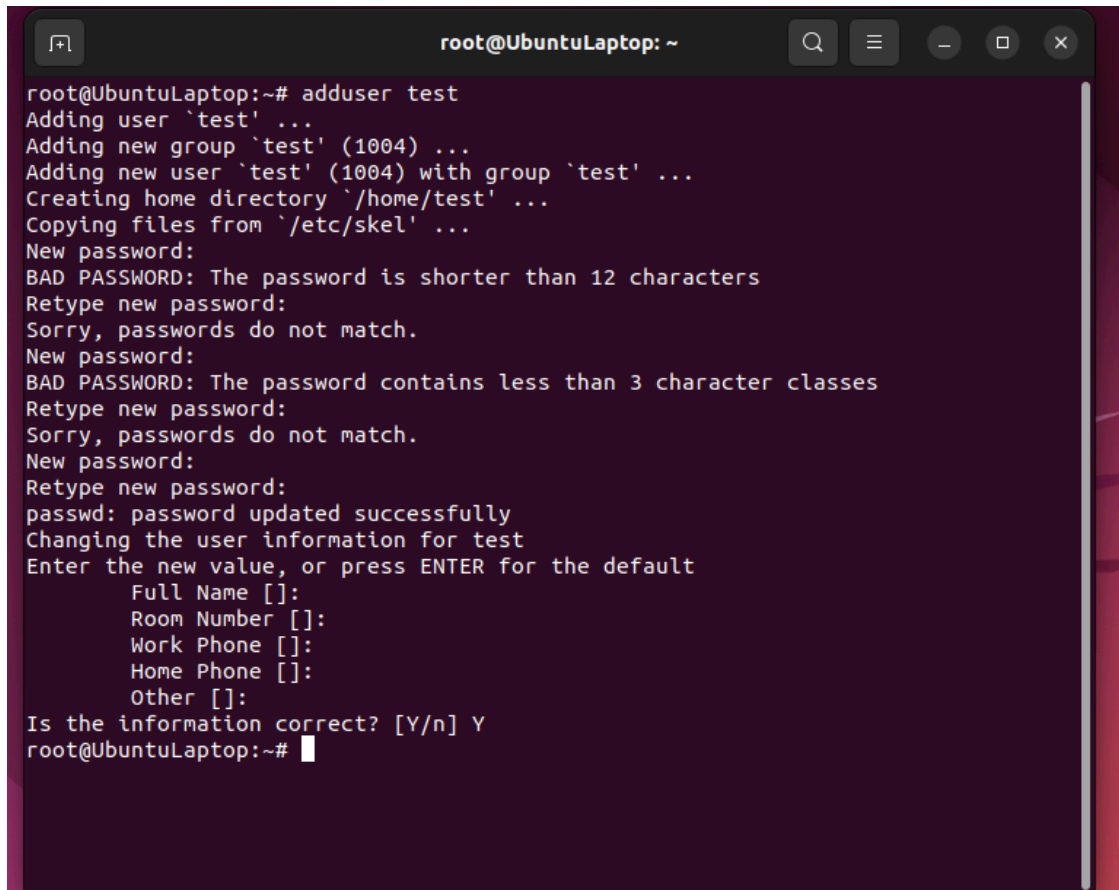
```
root@UbuntuLaptop: /etc/security
GNU nano 6.2 pwquality.conf *
# ucredit = 0
#
# The maximum credit for having lowercase characters in the new password.
# If less than 0 it is the minimum number of lowercase characters in the new
# password.
# lcredit = 0
#
# The maximum credit for having other characters 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.



```
root@UbuntuLaptop: ~  
root@UbuntuLaptop:~# adduser test  
Adding user `test' ...  
Adding new group `test' (1004) ...  
Adding new user `test' (1004) with group `test' ...  
Creating home directory `/home/test' ...  
Copying files from `/etc/skel' ...  
New password:  
BAD PASSWORD: The password is shorter than 12 characters  
Retype new password:  
Sorry, passwords do not match.  
New password:  
BAD PASSWORD: The password contains less than 3 character classes  
Retype new password:  
Sorry, passwords do not match.  
New password:  
Retype new password:  
passwd: password updated successfully  
Changing the user information for test  
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:~#
```

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).

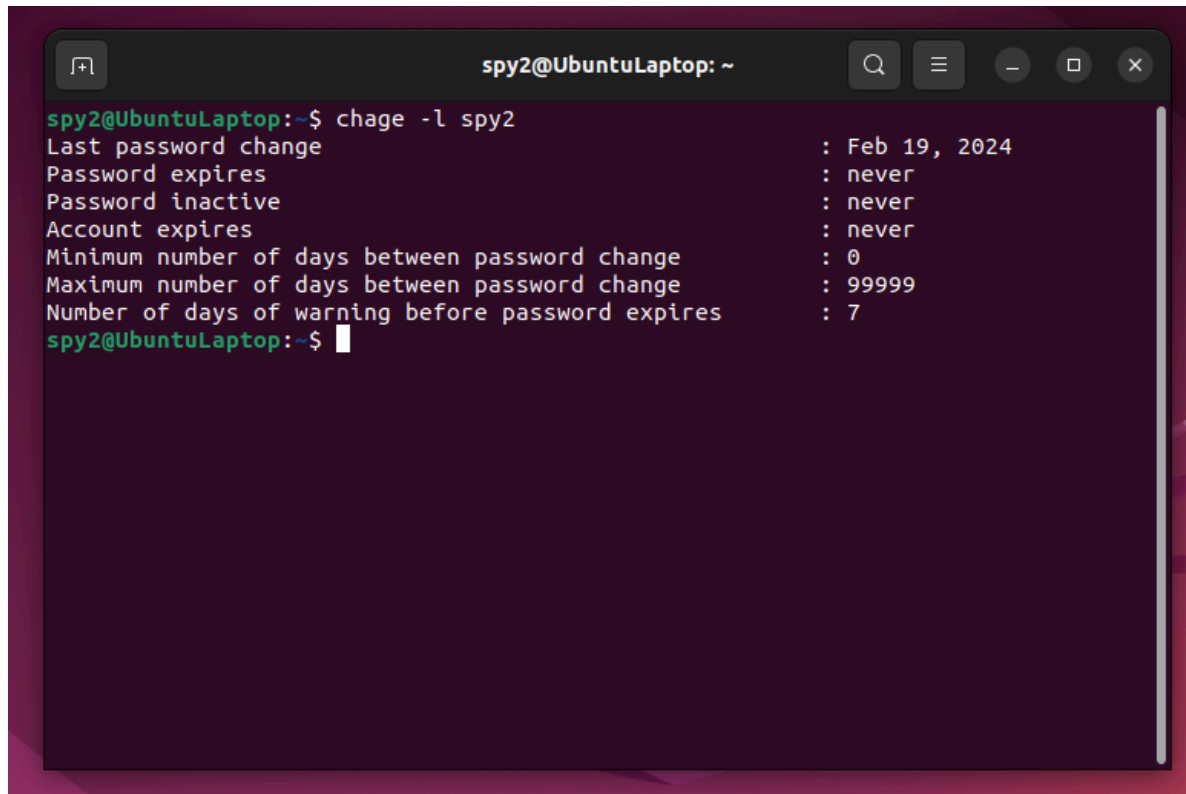
Bonus: Deleting a user on Linux (deleting test). Userdel User. Then get rid of the user's home directory by using rm -r. Then I verified that the user's home directory was deleted.

```
root@UbuntuLaptop: /home
root@UbuntuLaptop:~# userdel test
root@UbuntuLaptop:~# rm -r /home/test
root@UbuntuLaptop:~# cd ..
root@UbuntuLaptop:/# ;s
-bash: syntax error near unexpected token `;'
root@UbuntuLaptop:/# ls
bin    dev    lib    libx32  mnt    root    snap    sys    var
boot   etc    lib32  lost+found  opt    run    srv    tmp
cdrom  home  lib64  media   proc   sbin   swapfile  usr
root@UbuntuLaptop:/# cd home
root@UbuntuLaptop:/home# ls
luser1  mango  spy2  strawberry
root@UbuntuLaptop:/home#
```

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.

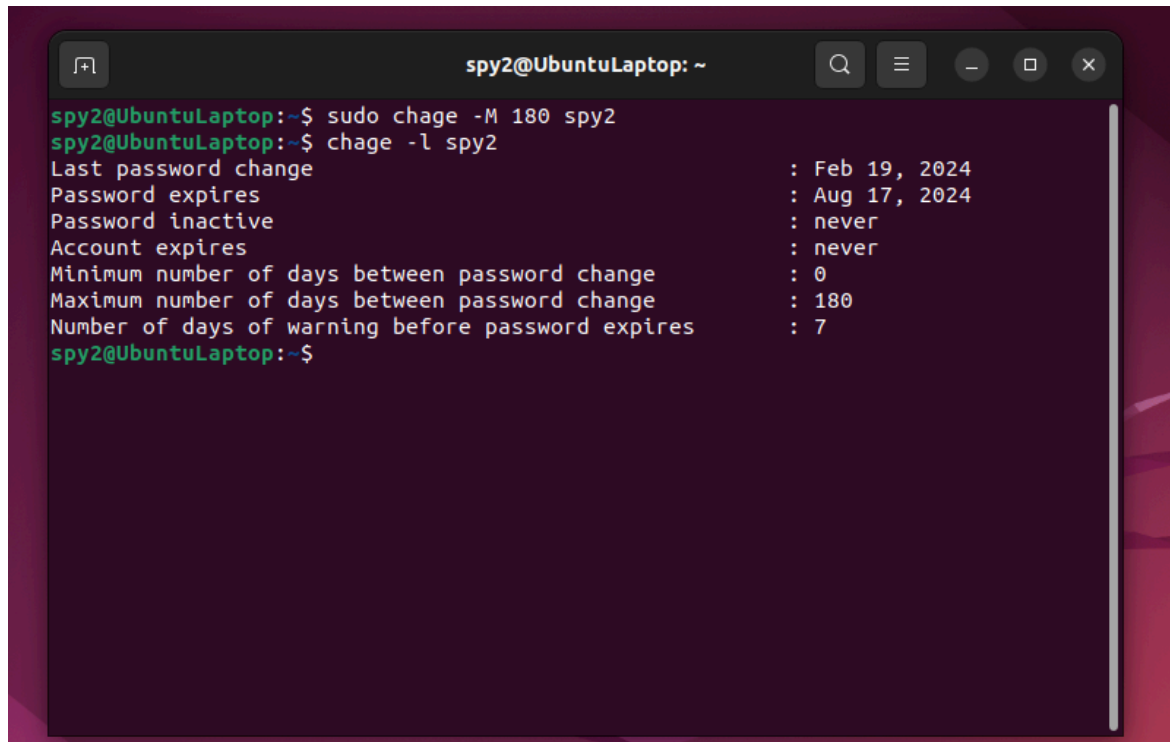
Solution #1: Check user account expiry data. Use `chage -l username`.

A terminal window titled 'spy2@UbuntuLaptop: ~' with standard Ubuntu window controls. The terminal shows the command 'chage -l spy2' and its output, which lists account and password settings for the user 'spy2'.

```
spy2@UbuntuLaptop:~$ chage -l spy2
Last password change                : Feb 19, 2024
Password expires                    : never
Password inactive                   : never
Account expires                     : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
spy2@UbuntuLaptop:~$
```

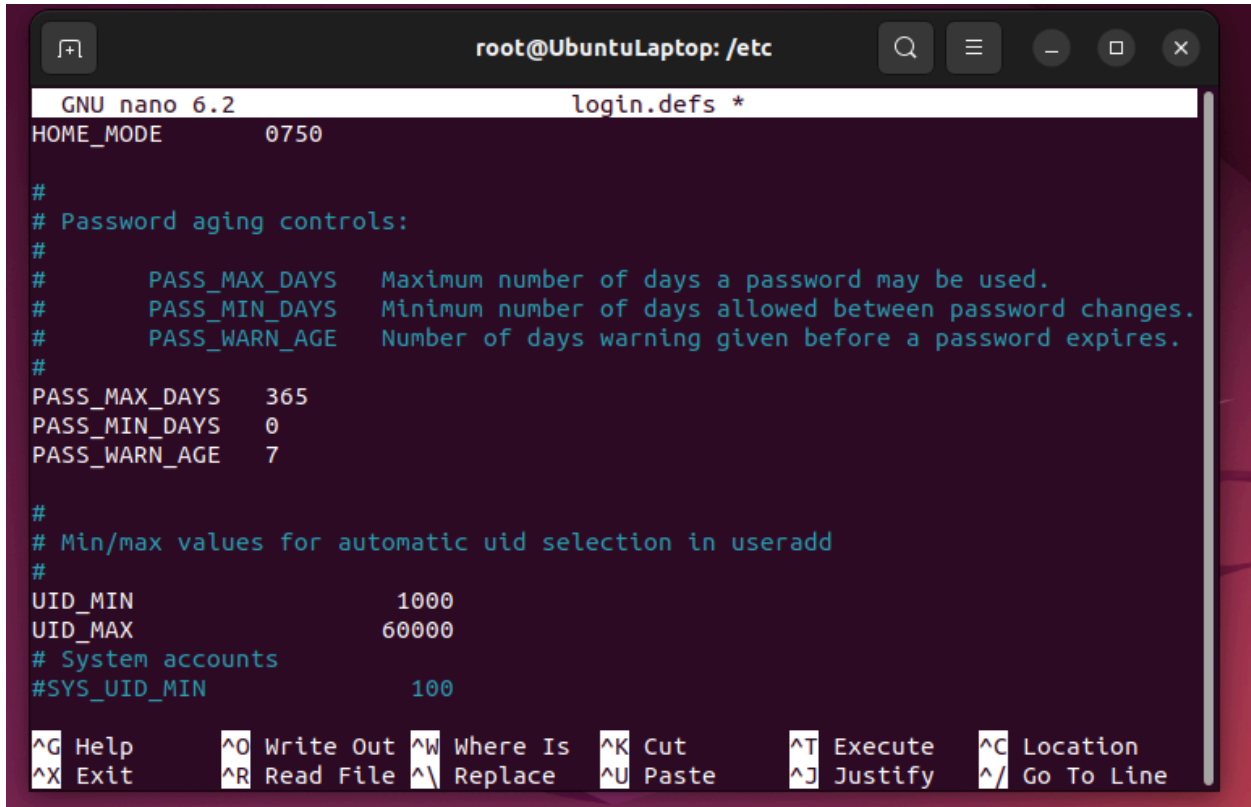
Spy2 does not have a password expiry date or account expiry date. Let us assume that since he is a spy he needs to change his password every six months (twice a year).

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

A terminal window titled 'spy2@UbuntuLaptop: ~' with standard Ubuntu window controls. The terminal shows the execution of 'sudo chage -M 180 spy2' followed by 'chage -l spy2'. The output lists password policy details for user spy2, including the last change date, expiration date (Aug 17, 2024), and the 180-day maximum interval.

```
spy2@UbuntuLaptop:~$ sudo chage -M 180 spy2
spy2@UbuntuLaptop:~$ chage -l spy2
Last password change           : Feb 19, 2024
Password expires               : Aug 17, 2024
Password inactive              : never
Account expires                : never
Minimum number of days between password change : 0
Maximum number of days between password change : 180
Number of days of warning before password expires : 7
spy2@UbuntuLaptop:~$
```

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

A screenshot of a terminal window on a Ubuntu system. The window title is 'root@UbuntuLaptop: /etc'. The terminal shows the nano 6.2 editor editing the file 'login.defs *'. The content of the file is as follows:

```
HOME_MODE      0750

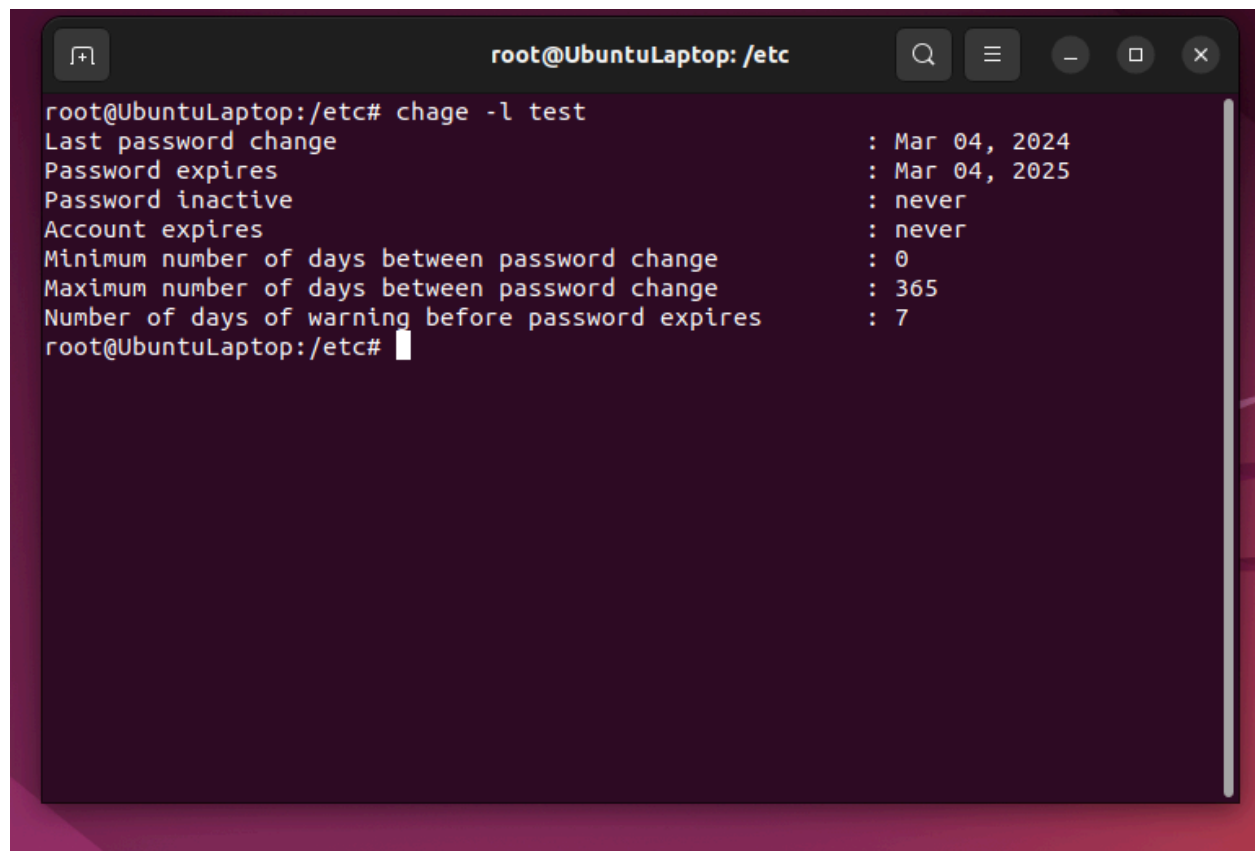
#
# Password aging controls:
#
#      PASS_MAX_DAYS   Maximum number of days a password may be used.
#      PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#      PASS_WARN_AGE   Number of days warning given before a password expires.
#
PASS_MAX_DAYS   365
PASS_MIN_DAYS   0
PASS_WARN_AGE   7

#
# Min/max values for automatic uid selection in useradd
#
UID_MIN          1000
UID_MAX          60000
# System accounts
#SYS_UID_MIN      100
```

At the bottom of the terminal, there is a status bar with various keyboard shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^\ Replace, ^U Paste, ^J Justify, and ^_ Go To Line.

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.

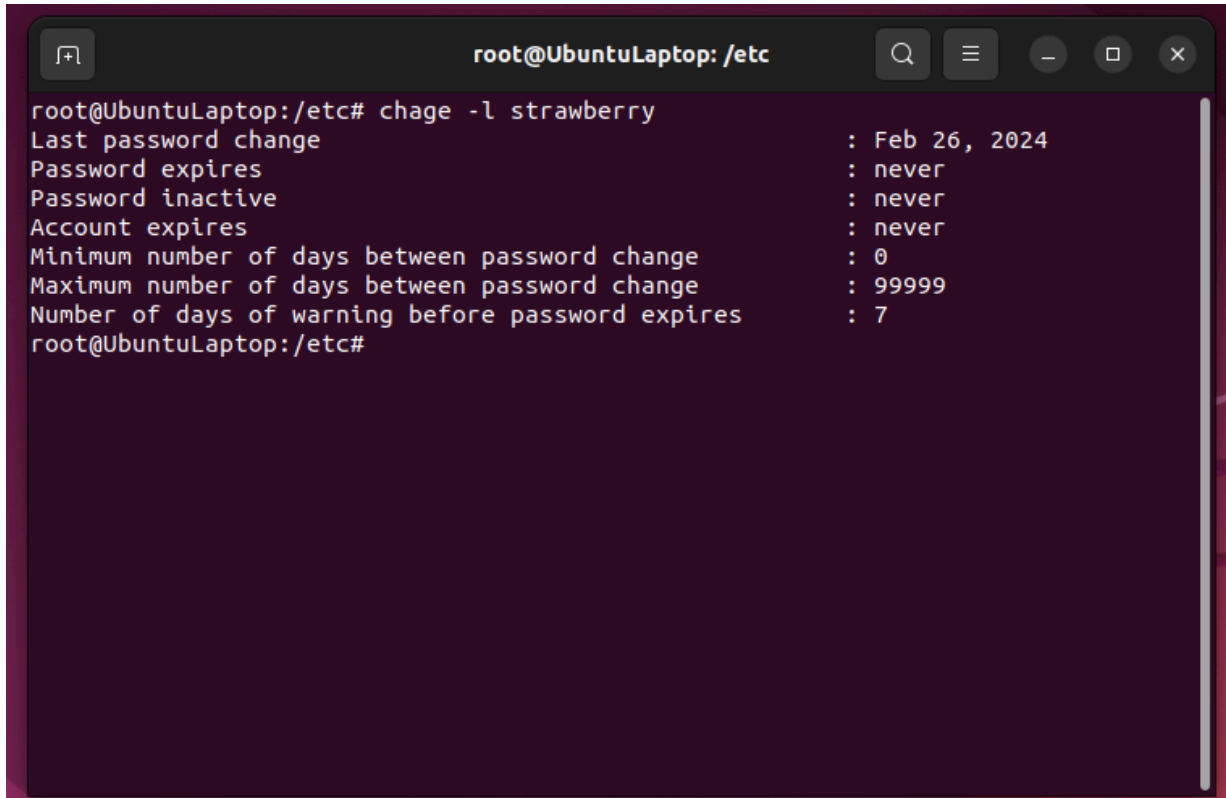
Account #1: test



A terminal window titled 'root@UbuntuLaptop: /etc' with standard window controls. The terminal displays the output of the command 'chage -l test'. The output lists various password and account settings for the 'test' user, including the last password change date, expiration date, inactive period, and warning days before expiration.

```
root@UbuntuLaptop:/etc# chage -l test
Last password change                : Mar 04, 2024
Password expires                    : Mar 04, 2025
Password inactive                   : never
Account expires                    : never
Minimum number of days between password change : 0
Maximum number of days between password change : 365
Number of days of warning before password expires : 7
root@UbuntuLaptop:/etc#
```

Account #2: strawberry

A terminal window titled 'root@UbuntuLaptop: /etc' with standard window controls. The command 'chage -l strawberry' has been executed, displaying the following password policy details for the 'strawberry' user:

```
root@UbuntuLaptop:/etc# chage -l strawberry
Last password change           : Feb 26, 2024
Password expires                : never
Password inactive              : never
Account expires                : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
root@UbuntuLaptop:/etc#
```

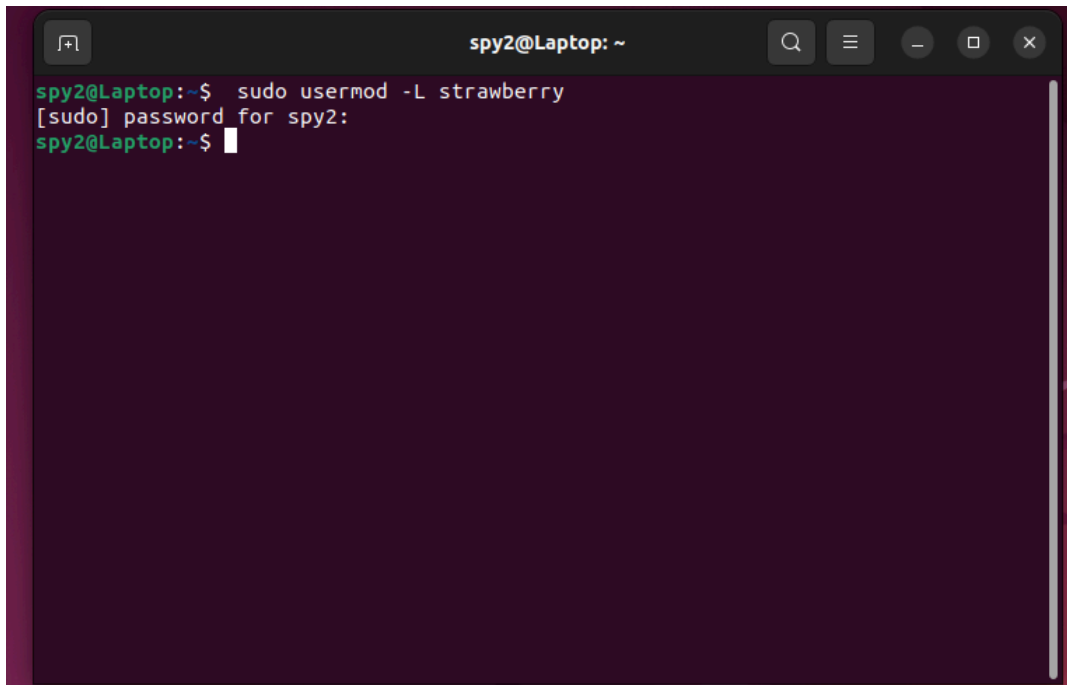
There is a difference between strawberry and test. Why? Well, strawberry was created before I changed the login.defs file. Test was created after, so the 1 year reset policy came into effect for him only.

Section #4: Locking/ Unlocking User Accounts

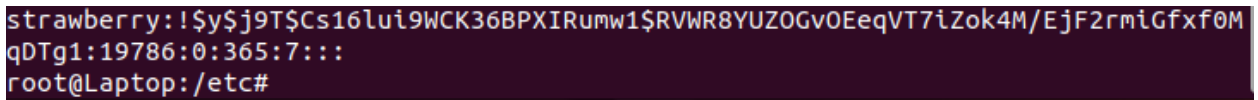
Issue: Sometimes user accounts should be locked down. If a user is on vacation and has sudo privileges, then their account should be locked so that any attackers or other curious employees can't attempt to use the elevated privilege. Or say a user was caught doing something suspicious and you don't want 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)

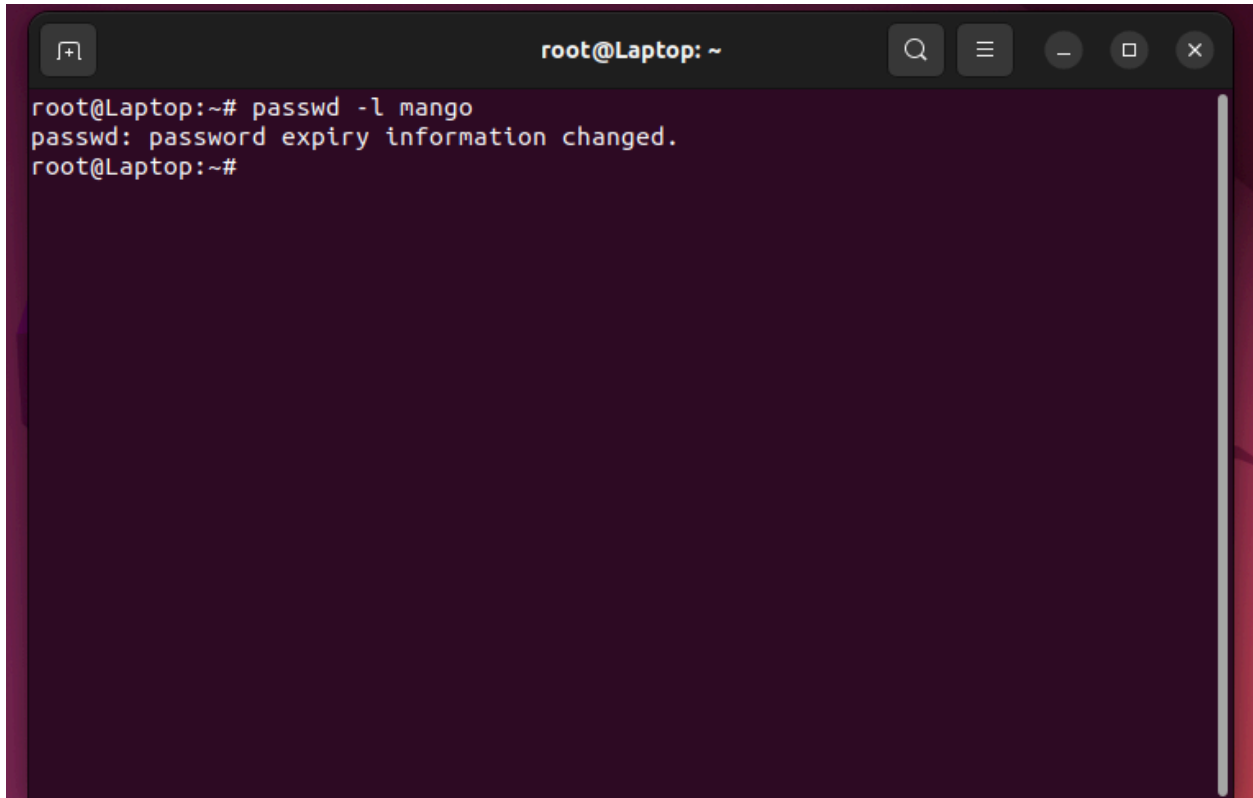
A terminal window titled 'spy2@Laptop: ~' with standard window controls. The prompt is 'spy2@Laptop:~\$'. The user enters 'sudo usermod -L strawberry'. The prompt changes to '[sudo] password for spy2:'. The user enters a password (represented by a white block). The prompt returns to 'spy2@Laptop:~\$'.

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.

A terminal window showing the output of a command to view the /etc/shadow file. The prompt is 'root@Laptop:/etc#'. The output shows the entry for 'strawberry' with a locked password (indicated by an exclamation mark) and other fields: 'strawberry:!!\$y\$j9T\$Cs16lul9WCK36BPXIRumw1\$RVWR8YUZOGv0EeqVT7iZok4M/EjF2rmiGfx0MqDTg1:19786:0:365:7:::'.

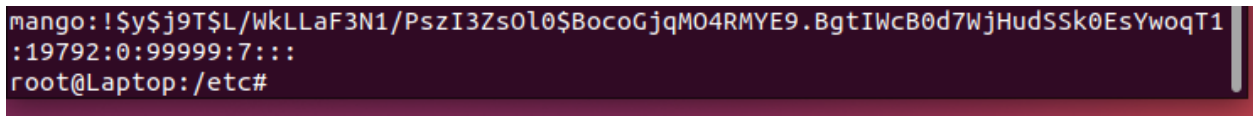
```
strawberry:!!$y$j9T$Cs16lul9WCK36BPXIRumw1$RVWR8YUZOGv0EeqVT7iZok4M/EjF2rmiGfx0MqDTg1:19786:0:365:7:::
root@Laptop:/etc#
```

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

A terminal window titled 'root@Laptop: ~' with standard window controls. The prompt is 'root@Laptop:~#'. The command 'passwd -l mango' has been entered. The output is 'passwd: password expiry information changed.' followed by the prompt '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:

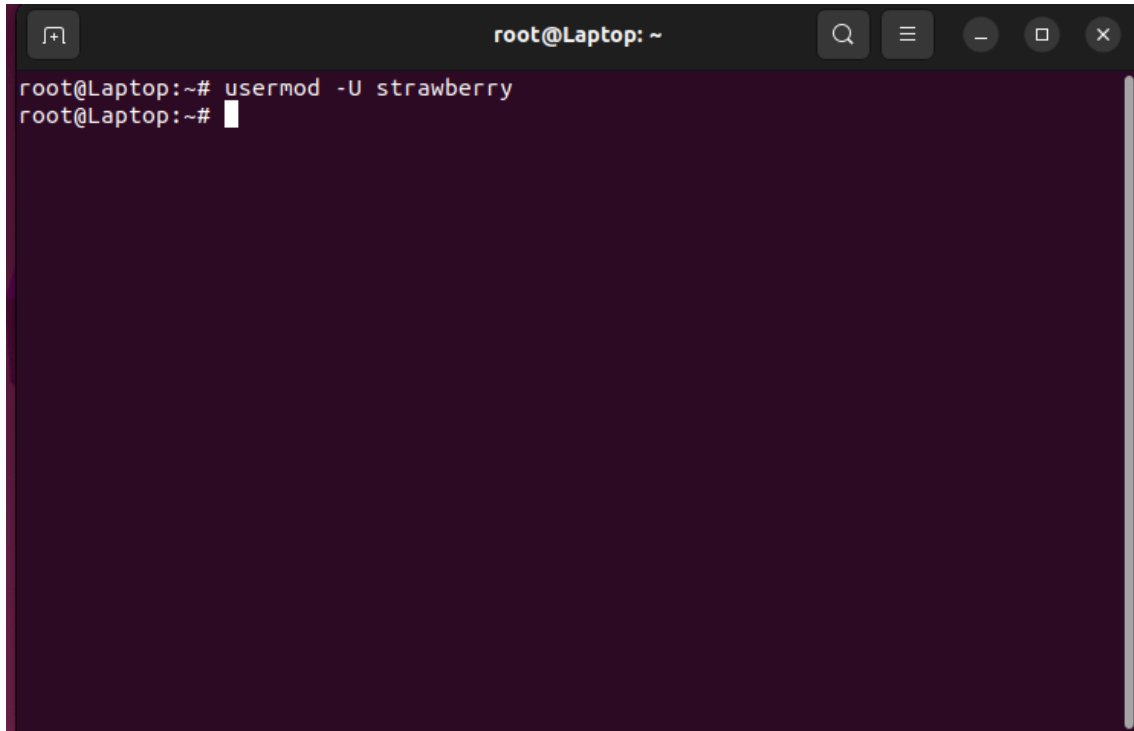
A terminal window showing the command 'cat /etc/shadow' being executed. The output shows the entry for 'mango' with a hashed password, an expiration date of 19792, and a lock status of 7. The prompt is 'root@Laptop:/etc#'.

```
mango: !$y$j9T$L/wkLLaF3N1/PszI3Zs0l0$BocoGjqM04RMYE9.BgtIWcB0d7WjHudSSk0EsYwoqT1
:19792:0:99999:7:::
root@Laptop:/etc#
```

The ! is in front of his hashed password.

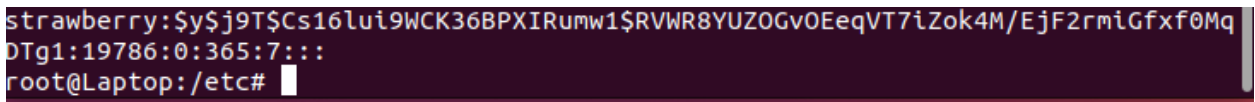
Solution #2 Unlocking Accounts

Method #1. Use usermod -U

A terminal window titled 'root@Laptop: ~' with standard window controls. The prompt is 'root@Laptop:~#'. The command 'usermod -U strawberry' has been entered and executed. The prompt is now 'root@Laptop:~#' with a cursor. The terminal background is dark purple.

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

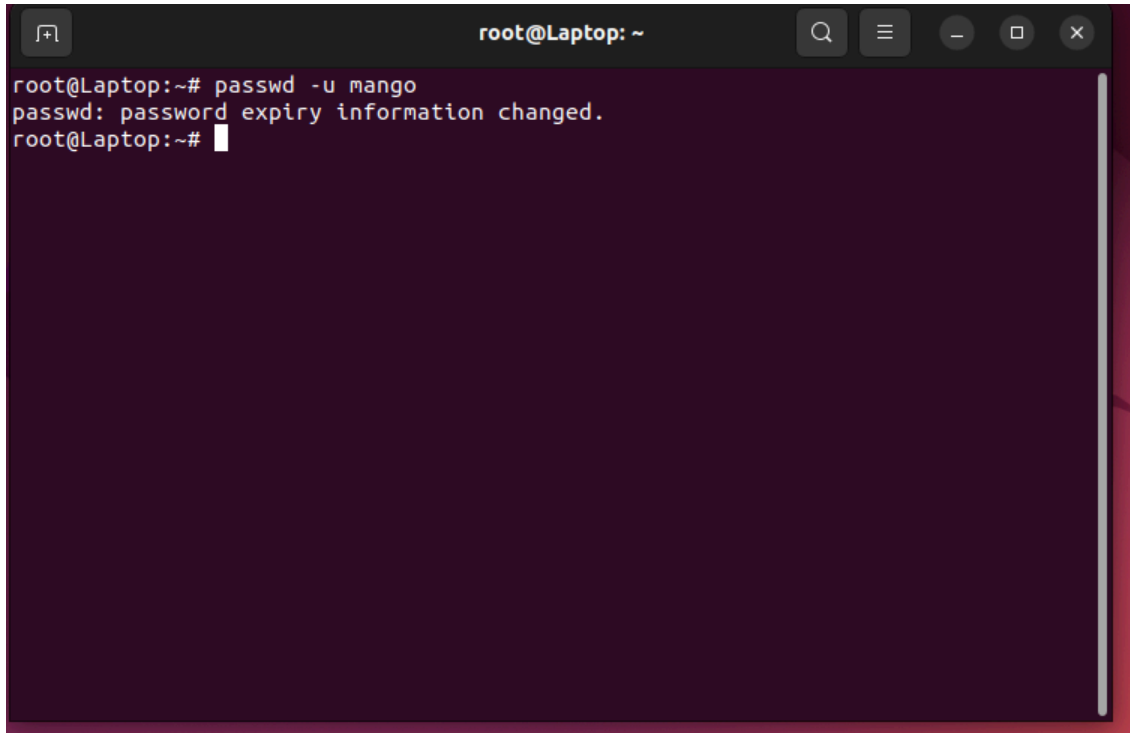
Let us verify that Strawberry is now able to login and be authenticated. We can do that by looking in the /etc/shadow file.

A terminal window showing the contents of the /etc/shadow file. The prompt is 'root@Laptop:/etc#'. The output shows the entry for 'strawberry' with a hashed password and other fields. The terminal background is dark purple.

```
strawberry:$y$j9T$Cs16lui9WCK36BPXIRumw1$RVWR8YUZ0Gv0EeqVT7iZok4M/EjF2rmiGxf0Mq
DTg1:19786:0:365:7:::
root@Laptop:/etc#
```

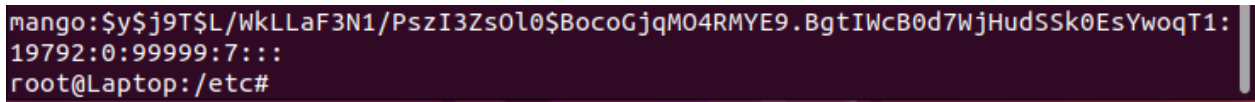
And yes, the ! is gone!

Method #2 Use passwd -u

A terminal window titled 'root@Laptop: ~' with standard window controls. The terminal shows the command 'passwd -u mango' being executed. The output is 'passwd: password expiry information changed.' followed by a new prompt '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:

A terminal window showing the contents of the /etc/shadow file for the mango user. The output is 'mango:\$y\$j9T\$SL/WkLLaF3N1/PszI3Zs0l0\$BocoGjqM04RMYE9.BgtIWcB0d7WjHudSSk0EsYwoqT1:19792:0:99999:7:::'. The prompt is 'root@Laptop:/etc#'.

```
mango:$y$j9T$SL/WkLLaF3N1/PszI3Zs0l0$BocoGjqM04RMYE9.BgtIWcB0d7WjHudSSk0EsYwoqT1:
19792:0:99999:7:::
root@Laptop:/etc#
```