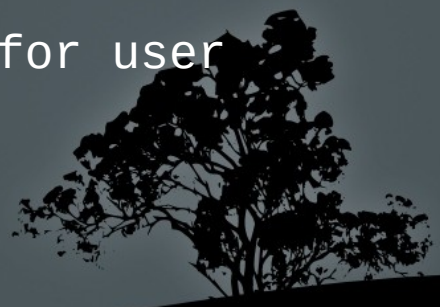
LPIC-1 101-400 – Lesson 21 – Lab

*** Enter into your Lab environment as root**

- **# cd Lab 21** # change into the Lab21 directory
- **# cat > test.txt <<EOF** # create a new file
  echo "I am executed!"
  EOF
- **# ls -l test.txt** # check the permissions
- **# umask** # see the default **umask**
- **# chmod 444 test.txt** # make the file read only
- **# ls -l test.txt** # verify the permissions
- **# vi test.txt** # edit the file
  o # create a new line at the end
  echo "Me too!" # write this line
  :wq # save the file. Did it work?
  :wq! # force save the file. How about now?
- **# chmod ug+w,o-r test.txt** # allow write by user and group but not others
- **# ls -l test.txt** # verify

Lesson 21 – Lab

- **# cat test.txt** # check the file contents
- **# ./test.txt** # try run the file as executable
- **# chmod 755 test.txt** # give execute permissions to the file
- **# ls -l test.txt** # verify
- **# ./test.txt** # run the file. Did it work?
- **# chmod a-x test.txt** # remove the execute bit
- **# ls -l test.txt** # verify
- **# su - user1** # change into a regular user
- **$ mkdir dir** # create a new directory
- **$ touch dir/test.file** # create an empty file in the directory
- **$ ls -la dir ; ls -ld dir** # list directory content and directory itself
- **$ chmod 400 dir** # leave only the read bit for user
- **$ ls -ld dir** # verify



Lesson 21 – Lab

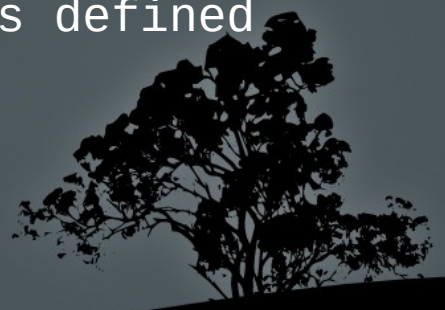
- **$ ls -la dir** # list the contents of **dir**. Does it work?
- **$ cd dir** # change into **dir**. Does it work?
- **$ chmod 100 dir** # enable the execute bit but disable read
- **$ ls -ld dir** # verify
- **$ ls -la dir** # list the contents of **dir**. Does it work?
- **$ cd dir** # change into **dir**. Does it work?
- **$ ls -la** # attempt another listing. Does it work?
- **$ exit** # exit back to root
- **# cp $(which vi) /usr/local/bin/vi-suid** # create a copy of **vi** somewhere inside the **$PATH**
- **# chmod u+s /usr/local/bin/vi-suid** # give the new binary the suid bit (this is a security risk and an example to avoid!)
- **# ls -l /usr/local/bin/vi-suid** # verify

Lesson 21 – Lab

- # **vi-suid** # verify that the new program is in the **$PATH**
  :q
- # **su - user1** # change into **user1**
- **$ vi-suid** # run the binary in one terminal
- **$ ps aux | grep vi-suid** # verify the process permissions in another terminal. What are the implications?
- # exit vi, close the two screen sessions and exit back to root
- # **rm /usr/local/bin/vi-suid** # remove the insecure binary
- # **find / -perm 400 -ls** # find files with **400** permissions
- # **find / -perm 640 -ls** # find files with **640** permissions
- # **find / -perm -4000 -ls** # find files with **SUID**
- # **find / -perm -6000 -ls** # find files with **SUID** and **SGID**
- # **find / -perm /6000 -ls** # find files with **SUID** or **SGID**
- # **find / -perm -2000 -ls** # find files with **SGID**
- # **find / -perm -1000 -ls** # find files with **Sticky**

Lesson 21 – Lab

- **# umask** # check the current umask
- **# touch test123.txt** # create an empty file
- **# ls -l test123.txt** # verify that permissions match the umask
- **# mkdir dir123** # create a new directory
- **# ls -ld dir123** # # verify that permissions match the umask
- **# umask 0027** # change the umask for this session
- **# umask** # verify the change
- **# touch test321.txt**
- **# ls -l test*** # verify the permissions
- **# mkdir dir321** # make another directory
- **# ls -ld dir*** # verify the permissions
- **# grep umask -r /etc** # check where umask is defined
- **# umask 0022** # reset umask back to default



Lesson 21 – Lab

- **# echo "Test ownership" > \ /home/user1/test.own** # create a **root** owned file under the **user1** homedir
- **# ls -la /home/user/test.own** # check ownership
- **# su - user1** # switch to **user1**
- **$ ls -l test.own** # verify ownership
- **$ echo "Append from user" >> test.own** # try add some text to the file. Did it work?
- **$ chown user1 test.own** # try changing the ownership of the file
- **$ rm -f test.own** # try removing the file
- **$ touch test.own** # try changing the timestamp of the file
- **$ ls -l test.own** # any change?
- **$ chgrp user1 test.own** # try changing the group ownership
- **$ exit** # exit back to **root**

Lesson 21 – Lab

- **# chown user1:user1 /home/user/test.own** # change the user/group ownership
- **# ls -l /home/user/test.own** # verify
- **# su - user1** # switch to **user1**
- **$ ls -l test.own** # verify the permissions
- **$ touch test.own** # try changing the timestamp of the file
- **$ ls -l test.own** # any change?
- **$ exit** # switch to **root**
- **# touch /tmp/test.user1** # create a temporary file
- **# chown user1 /tmp/test.user1** # change the ownership to **user1**
- **# su 1 user1** # change to **user1**
- **$ ls -l /tmp/test.user1** # verify group ownership
- **$ chgrp user /tmp/test.user1** # change group to **user1**
- **$ ls -l /tmp/test.user1** # any change?

Lesson 21 – Lab

- **$ echo "Test" > /tmp/test.user** # try writing to the file
- **$ cat /tmp/test.user** # check the file’s contents
- **$ rm /tmp/test.user** # remove the file
- **$ ls -l /tmp/test.user** # verify

License