***CPRE 431***

***M03 HW***

**Assignments will be submitted in PDF format via Canvas.**

Please submit your homework online through Canvas. Late homework will not be accepted.

Please ensure that you support all your answers with the correct screenshots showing your solutions.

1. In this “lab” problem, you will be working on a Linux Server Virtual Machine (VM). An image of this VM is available on Canvas (attached with the HW). The VM is having an administrator and 5 users, as shown in the figure below. You don’t have access to any of the users of the Server, to be able to access the Server, you will need to perform password cracking! You were given a line of password hash from (/etc/shadow) for the administrator (admin1) of the Server (attached with the HW).
   1. Determine the used hash type of the password.

*A screenshot of a computer

Description automatically generated*

*From this website, we find out that the first part is the username, the first number between $ sign is the algorithm, followed by the salt, which is followed by the password hash.*

*In our case, we have;*

**

*So we know from* ***$6$****, that the type of hash is* ***SHA-512****.*

* 1. Determine the salt value of the password.

*Since the format for /etc/shadow is $id$salt$hash, in our case, the salt would be* ***xgLS35S6.***

* 1. Crack the passwords of all users using OpenSSL tool.

*So the way to crack the password is to generate a hash using the salt from the password contents of a shadow file. To do this we write a bash script that checks each line and finds the correct sequence by brute force.*

*This is the bash script I wrote:*

*A screenshot of a computer program

Description automatically generated*

*A screen shot of a computer screen

Description automatically generatedThis is a snippet of the code running and inspecting the hash of each line.*

*I had been running it for more than 10 minutes, so I decided to use the GPU servers (cpre587). This ended up giving me a password after matching the hash to the line from the text file ‘100-most-used-password-NIST.txt’.*

*A screen shot of a computer

Description automatically generated*

*Above is the output that I got after partially running my script.*

*The password is:* ***P@ssw0rd***

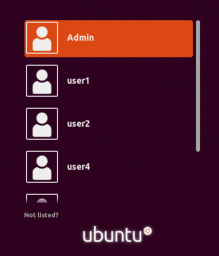
*A screenshot of a computer

Description automatically generated*

*This is a screenshot of the unlocked admin1 profile.*

Hints:

* It would be useful if you search for Linux shadow file password format.
* You must use the latest OpenSSL version 1.1.1x for this problem. It would be helpful to read about creating Linux password hashes using OpenSSL.
* You can use a password list for your cracking. There is a password list of most used 100K passwords, according to NIST attached with the HW.
* You will need to write some code to iterate through the password list (feel free to use any language you prefer).
* After cracking the password, ensure that you can access the Server.



1. The given Server users are managed as follows:

* There are 5 users: user1, user2, …, user5. The first four users are staff members, but user5 is an external consultant.
* User1 and user2 are programmers only, user4 is a manager only, and user3 is both programmer and manager.
  1. You are now the administrator of the server, and you are responsible for managing users and groups. Create 3 groups named “allstaff”, “prog” (short for programmers) and “mgmt” (short for management). Add users to the corresponding groups. List the users and groups to ensure the correct previous setup of system users and groups. Explain with screenshots how you applied this.

*To make new group, we call the groupadd function.*

A close up of words

Description automatically generated

*Below is a snipped of /etc/group which shows new added groups*

A screenshot of a computer code

Description automatically generated

*We then add users to existing groups by using the usermod -aG command.*

A screenshot of a computer program

Description automatically generated

*Finally after adding all users into their respective groups, we analayze the /etc/group file, and this is the result we get*

*A black background with white text

Description automatically generated*

* Inside their home directory, each programmer has a directory called code and a directory called documentation. Inside the code directory, there is a file called source\_code.txt, as well as one application called myapp.exe. Inside the documentation directory, there is a file called notes.txt.
* Each manager has a directory called finance (for financial information) including a confidential business.txt file.
* Each user also has a file called schedule.txt in their home directory.
  1. Configure file access controls so that it explicitly applies only the following:
* All users can view each other's schedules, but not other files in their home directory (except for as stated in the following).
* All staff can view files in the documentation directory of other staff.
* Programmers can view and edit each other's source code files, create new files in any code directory, as well as run each other's myapp.exe files.
* Financial information (in the finance directory) is only viewable by the manager that owns the files, not by any other user.

Hint

* Take screenshots of setting the access and listing it using (ls -l). Also, take screenshots of testing that the access control works by logging in as each user and checking they can(not) access the specified files/directories.
* Use only the basic Linux permissions. Do NOT use advanced permissions such as setfacl or getfacl.
* Use the [“Introduction to Linux” Page](https://canvas.iastate.edu/courses/64200/pages/introduction-to-linux) to help you in the needed commands for this Homework.

*Below are the hash for each user found in the shadow file,* ***“/etc/shadow****”.*

A screenshot of a computer code

Description automatically generated

***password output with new hash and new salt:P@***

*user1:*

*$6$2Ff.cblr$QwoNFOAme5xy5/anjAsZVIDrZdBKZ.hZ6UIIdLU9D4DDEs3O.CbRsICaVxxdQOTG2TOHYSHDwfdsrG0WGsXVB/:18169:0:99999:7:::*

**

*user2: $6$zZKc4nOX$Rac9mB17TLeFgE/TOH0gTgRnAmaNk67ezuZo4fQAOSkulEKrrW6sum0uElLvmAeBqhf0k/jCYn2dddJCC0QzI1:18169:0:99999:7:::*

**

*user3:*

*$6$dCPTizMy$b8Fiueet0w08JR66mI3UPg.U7ertejWDHTDCAyqbVSjhhLgTu8N2/51R496408q356m.gmJSjG.u89L.3K8HH.:18169:0:99999:7:::*

**

*user4:*

*$6$0Ptm7uW6$9cYOHvx3S6dJBgK4ZhVq.bPHJlaMH.KV/59bsX2UYVSBp6RUit6KKFLnuoKz5L5yHMH75YZymLcil9uBhV4XW.:18169:0:99999:7:::*

**

*user5:*

*$6$QpU0v3n/$Z5BKWAKu6SsZMI4KStZmlR/IZuhE9Ts.cezqBca3iApKmbT/GSBC1GUHf0I0mmytOdmqzclHkT47idGnpmHoe0:18169:0:99999:7:::*

**

***Note:******Common******Testing can be done using User3 and User5, since user3 is a part of all groups and user5 is a part of no groups.***

**USER 1:**

*sudo chmod +r schedule.txt*

*sudo chgrp allstaff documentation*

*sudo g+rx documentation*

*sudo chgrp allstaff documentation/notes.txt*

*sudo g+r documentation/notes.txt*

*sudo chgrp prog code*

*sudo chmod g+rwx code*

*sudo chgrp prog code/source\_code.txt*

*sudo chmod g+rw code/source\_code.txt*

*sudo chgrp prog code/myapp.exe*

*sudo chmod 750 code/myapp.exe*

A screen shot of a computer

Description automatically generated

*According to permission set, notes.txt of user1 under documentation directory can only be read by users in group allstaff, which means that user5 won’t be able to read it.*

A screenshot of a computer program

Description automatically generated

A computer screen shot of a program

Description automatically generated

I set read permissions also for myapp.exe since it wasn’t executing without it.

A computer screen shot of a computer code

Description automatically generated

*Testing:*

*A computer screen shot of a computer code

Description automatically generated*

*A computer screen with white text

Description automatically generated*

*From the snipped of code, we see that the commands ran from user3 for specific files with group permission, run without any issue. Meanwhile, since user5 is not part of allstaff, the only file accessible is schedule.txt*

**USER 2:**

*sudo chmod +r schedule.txt*

*sudo chgrp allstaff documentation*

*sudo g+rx documentation*

*sudo chgrp allstaff documentation/notes.txt*

*sudo g+r documentation/notes.txt*

*sudo chgrp prog code*

*sudo chmod g+rwx code*

*sudo chgrp prog code/source\_code.txt*

*sudo chmod g+rw code/source\_code.txt*

*sudo chgrp prog code/myapp.exe*

*sudo chmod 750 code/myapp.exe*

*A screenshot of a computer program

Description automatically generated*

***Testing:***

***A computer screen shot of a computer program

Description automatically generated***

***A computer screen shot of a computer code

Description automatically generated***

*Similar to what we saw in User1’s testing, this has the same permissions and are part of the same groups. Above is the execution as we expected.*

**USER 3:**

*sudo chmod +r schedule.txt*

*sudo chgrp allstaff documentation*

*sudo g+rx documentation*

*sudo chgrp allstaff documentation/notes.txt*

*sudo g+r documentation/notes.txt*

*sudo chgrp prog code*

*sudo chmod g+rwx code*

*sudo chgrp prog code/source\_code.txt*

*sudo chmod g+rw code/source\_code.txt*

*sudo chgrp prog code/myapp.exe*

*sudo chmod 750 code/myapp.exe*

*sudo chmod 700 finance*

*sudo chmod 400 finanance/business.txt*

**A screenshot of a computer program

Description automatically generated**

***Testing:***

*A computer screen shot of a computer code

Description automatically generated*

*Above we see that user4, who is in the mgmt group trying to access files. Documentation works. Code and finance directories are denied of permission even though user4 is in the mgmt group.*

*We go further by changing users and testing with a programmers account, we see that the programmer has access to the code directory and can read write and execute in the code directory. The permission was set this way so for cd, ls and writing new files.*

**USER 4:**

*sudo chmod +r schedule.txt*

*sudo chmod 700 finance*

*sudo chmod 400 finanance/business.txt*

*A computer screen shot of a program

Description automatically generated*

****

*Above we see that, the finance directory can only be opened by the user and not even anyone from the same group (user4).*

**USER 5:**

**A screenshot of a computer

Description automatically generated**

**A computer screen shot of a computer program

Description automatically generated**

*Above, we see that, user5 only has permission to see schedule.txt and no other file or directory.*

**Additional Tests:**

*We test users of the prog group being able to write and edit files in the code directory.*

*A screenshot of a computer

Description automatically generated*

*As we see, only source\_code.txt is editbale by users of the same group. This means that myapp.exe can only be executed, but not edited.*

*A computer screen shot of a black screen

Description automatically generated*

*A screenshot of a computer code

Description automatically generated*

*Above we see that source\_code.txt was editable by user1.*

*Another condition was that users of prog group would be able to create new files only in the code directory and no where else. Below shows the implementation that proves the correctness.*

*A screenshot of a computer program

Description automatically generated*

*Below is test\_file.txt that was attempted to be created in the documentation directory.*

*A computer screen shot of a black screen

Description automatically generated*

**Conclusion:**

*All the tests and screenshots proves that everything works as expected.*