VICTORIA UNIVERSITY OF WELLINGTON
TE HERENGA WAKA

CYBR371: System and Network Security

2024–T1: Arman Khouzani, Mohammad Nekooei

Assignment 1 (15%): Access Control

Submission Deadline: ***23:59:00 (NZST) on Sunday, 14 April 2024***

- There are two case studies for a total of 150 points. Since this assignment has 15% contribution, think of each 10 points contributing 1% to your overall grade.
- you should upload **a single compressed file** (zip, tar, etc), containing:
  - a **single pdf file** (8 pages max in total, excluding references) containing your answers to Q1 (1 page), Q3 (3 pages max), Q7 (1 page) and Q7 (3 pages max).
  - Script files for Q2, Q4, Q5, and Q6, named according to the what is specified in the task (e.g. **`setup.sh`** for Q2.)
- The answer to each question must start on a new page, properly marked showing which question. If you decide to skip a question, you must still label it on your document and leave the answer blank. **Do not include the question statement or the description of the case study in your report.**
- Do not use a phone to snap a shot, use **`PrintScreen`**!
- Since **`NetLab`** does not allow saving, and is time-limited, it is recommended that you run a Linux VM on your own machine. Just beware: if you are using the lab machines, move your image file to **`/local/scratch/[yourname]`** directory, which saves it to that local machine. This is because you have a limited disk quota on your home directory, and also your VM run faster.
- Recall that you can ask for help from tutors during lab times. However, never, under any circumstances, share your answers. **There is zero tolerance toward plagiarism.** If you are in doubt what constitutes plagiarism, ask me!

Question:	1	2	3	4	5	6	7	8	Total
Points:	15	15	30	15	15	15	15	30	150
Score:									

Case Study 1

You are the system administrator working at Bestington University. You are in charge of setting up the filesystem and access controls for the course **`CYBR371`**. The course has:

- two *lecturers*: **`arman`** and **`mohammad`**.
- three *tutors*: **`ilona`**, **`esther`**, **`immaculata`**.
- 94 *students*: for simplicity, assume their usernames are **`student000`** to **`student093`**.

  All of these are already registered users in our Linux system (they can log in with the above usernames). Currently, each of them is only a member of a group with the same name as their username.

The course has 9 assessment pieces:

- **`lab1`**, . . . , **`lab5`**, **`assignment1`**, **`assignment2`**, **`midterm`**, **`final`**.

For each assessment piece, there is a **`questions.pdf`** file and a solution **`solutions.pdf`** file. The lectures should be able to modify these files. The tutors should be able to view these files. The students should only be able view the "questions" files (naturally!). Each student should be able to upload their **`answers.docx`** file for each of the assessment pieces. The students should not be able to view or modify (or delete!) the answers of other students. The lecturers and tutors should be able to view them. The tutors and lecturers should be able to modify the single file **`grades.xlsx`** (so that they can mark the students and enter their grades). You may want to use the directory and file structure depicted in Fig. 1.

Like our department, they also use a networked filesystem: so no matter which machine a user log in from, the same filesystem is mounted for them.

The **`opt`** directory is owned by the root (you!). You create the shell script file **`setup-cybr371.sh`**, whose role is to create the sub-directories (and perhaps the files), and sets the proper access controls on them. The idea is that you (as the root), run this only once at the beginning of the trimester.

Tasks

Q.1 [15 points] Complete the following (partial) access control matrix, accordingly. You must follow the principle of least privileges as much as possible. The only available rights are **`r`**, **`w`**, and **`x`**.

	`setup.h`	`grades.xlsx`	`final/questions.pdf`	`final/solutions.pdf`	`final/student000/answers.docx`	`final/student001/answers.docx`
arman						
ilona						
student000						
student001						
student002						

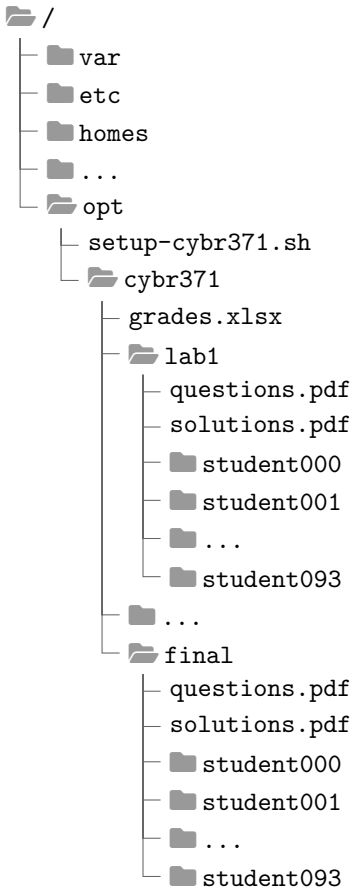

Figure 1: Recommended directory structure related to Case Study 1.

Q.2 [15 points] Provide the **`setup.sh`**. Recall that this is a shell script owned by the **`root`** user (you), supposed to be run only once at the beginning of the trimester, that sets up the sub-directories/files. It should of course also include the commands that create any necessary groups, adjusts group memberships, assigns the proper group membership, set ownership and permissions on directories and files, and potentially, any extended access control list that may needed to be applied.

Note:

- Try as much as possible to only rely on the Linux basic access control (owner, group, other). Use any extended access control (through **`setfacl`**) or special permissions if absolutely necessary.
- Also, when testing your script, try it with few number of students, say only 3 of them: `student000`,...,`student002`. Once you check that your script achieves the goals, change it slightly to work for 94 students before your submit.

Q.3 [30 points] Provide a detailed explanation of how your script achieves the requirements. You should also include the (screenshot) outputs of **`ls -al`** and potentially **`getfacl`** on some representative files and directories to establish your points (1000 words and 3 page maximum). You should also show the value of **`umask`**. Note that **`umask`** is a system-wide parameter (affecting all directories), so it does not matter where you issue it.

Case Study 2

You are the system administrator working at **`WellingtonClinic`**.

System overview

`WellingtonClinic` is system comprised of a set of files, directory structures, executable scripts and access control lists while ensuring proper access control. **`WellingtonClinic`** uses files to store all information rather than the popular (and honestly, the preferred more sensible solution of!) relational databases.

Roles

The following roles are defined in relation to the hospital environment:

Administrator user belongs to "**`sudo`**" (and others if needed) and sets up the directory structure. They create the roles, groups and carry out account maintenance if needed. Only administrators are able to add, remove or modify staff account and information. Use your account with the **`sudo`** permission as an administrator.

Doctors are users that can register a new patient. The doctor registering a new patient becomes the *primary doctor* of the patient. At registration of a patient, other doctors may also be specified as their *secondary* doctors. A patient may have none or multiple secondary doctors. Primary doctor and secondary doctors of a patient can read and write new prescription (only for their assigned patients). They should however not be able to view or modify any information about other doctors' patients (i.e., patients whom they have not registered (i.e. primary) nor are secondary doctors for). Only registered (i.e. primary and secondary) doctors must be able to change a patient's basic personal information. Doctors cannot be patients at the clinic.

Nurses are also users of the system. Nurses should not have access to the patient's entire medical records. They should only be able to access part of the patient's record extracted from the patient's file through a script (as detailed in the "tasks" section later). Nurses must also not be able to modify any information about patients. Nurses cannot be patients at the clinic.

Here are the staff at the clinic:

Name	username	Role
Dr Lou Ngevity	drloun	Doctor
Dr Stethos Cope	drstethosc	Doctor
Dr Bea Shure	drbeas	Nurse
Phil Paine	philp	Nurse

Patients

Patients are not users of the system. Their information is however saved on the system. There is a file associated with each patient which contains all the information about that particular patient. Each file's name follows a pattern of: *firstname and last name followed by the year the patient was born in*, e.g.:

- Rick, Sanchez, 1950 ⇒ RickSanchez1950
- Summer, Smith, 2007 ⇒ SummerSmith2007

The information in such files is in the following format.

```
FirstN,LastN,YearofBirth,RegDate,~PrimaryDr,#SecondaryDr(s)

date-of-visit,doctor-examined,diagnosis,medication,dosage
```

e.g., *RickSanchez1950*:

```
Rick,Sanches,1950,1/1/2024,~drstethosc

11/2/2024,drstethosc,existential dread,chillaxomine,bid hs
```

or, *SummerSmith2007*:

```
Summer,Smith,2007,1/1/2024,~drloun,#drstethosc

7/1/2024,drloun,boredom,adventurafil,qd
1/2/2024,drstethosc,itchy head,scratchacine,prn
10/2/2024,drloun,Wi-Fi Withdrawal Syndrome,placebazole,pc
```

The files and directory structure of **`WellingtonClinic`** is kept under **`/opt`** directory (Fig. 2). You'll need to have super user permissions to read and write to this directory. Root subdirectories (**`opt`**, **`var`**, **`etc`**, **`tmp`**, **`bin`**, **`usr`**, ...) are system folders so please make sure not to delete them! There is a subdirectory under the main folder of **`WellingtonClinic`** which contains the files associated with each patient.

Tasks

Q.4 [15 points] As an administrator, write a shell script (**`register-patient.sh`**) which allows doctors to execute the script and register a new patient by creating the necessary patient file. The script asks the registering doctor to enter the patient's basic personal information. The registering doctor becomes the primary doctor for the patient. Note: For the sake of simplicity, we assume there are no patients with duplicate information such as exact first name, last name and year of birth. Also you need to make sure nurses cannot register any new patients.

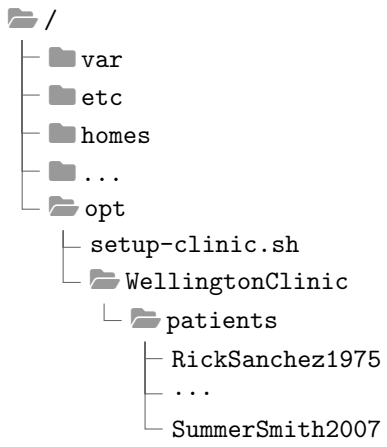

Figure 2: Recommended directory structure for Case Study 2.

```
Enter the following information about the patient.
First name:
Last name:
Year of birth:
```

Q.5 [15 points] First create the two files *RickSanchez1950* and *SummerSmith2007* as described in the case study. You will need these to check your script! Then write a shell script (**`check-medication.sh`**) that allows nurses to display a list of past and current medications and the date they were prescribed for a particular patient. The patient information (i.e. First Name, Last Name, and Year of Birth) is entered by the nurse. The information is displayed in the format shown in Fig. 3.

Patient	**Primary Doctor**	**Secondary Doctor(s)**	
Summer Smith	Dr Lou Ngevity	Dr Stethos Cope	
Date of Visit	**Attended Doctor**	**Medication**	**Dosage**
7/1/2024	Dr Lou Ngevity	adventurafil	qd
1/2/2024	Dr Stethos Cope	scratchacine	prn
10/2/2024	Dr Lou Ngevity	placebazole	pc

Figure 3: Example output of the **`check-medication.sh`** for a particular patient.

Q.6 [15 points] As an administrator, write a bash script **`setup-clinic.sh`** that does the following:

- add the staff (doctors and nurses) as new users according to the information given in the case study.
- create any groups that are needed and set group memberships.
- create the directory structure of the clinic.
- adjust the ownership, permission, and potentially ACLs on files/directories.

Q.7 [15 points] After studying the case study well, fill out the following partial access control matrix.

	`register-patient.sh`	`check-medication.sh`	`patients`	`patients/RickSanchez1950`	`patients/SummerSmith200`
All Doctors					
drloun					
drlstathosc					
All nurses					
others					

Q.8 [30 points] Provide a detailed explanation of how you achieved the requirements of this case study (1000 words and 3 page maximum). You should also include the (screenshot) outputs of **`ls -al`** and potentially **`getfacl`** on some representative files and directories to establish your points including the following resources:

```
check-medication.sh
patients (directory)
RickSanchez1950
SummerSmith2007
```

Also show the value of **`umask`**.