File System Access Control (ACL)

Due Date: 12/4/2022

Assignment 1 (100 Marks Total = 30% of overall grade)

Objectives

The learning objectives for the assignments are:

1. Be able to analyse an information system and identify its resources and components.
2. Analyse a security policy and identify relevant access control needs.
3. Implementing the policy using appropriate access control systems.

Although we will be focusing on the UNIX file system, what you learn here will also be applicable to the Windows operating system.

Case study

You are the system administrator working at WellingtonClinic and responsible for setting up the security controls to manage access for staff.

System overview

WellingtonClinic is system comprised of a set of files, directory structures, executable scripts and access control lists which facilitate the process of bookings between patients and their doctors primarily, while ensuring proper access control.

Data and directory structure:

WellingtonClinic uses files to store all information rather than the popular relational databases. This provides simplicity in design while ensuring all the requirements of the system are met.

The files and directory structure of WellingtonClinic is kept under a directory with the same name which is located under **/opt** directory. You'll need to be a super user to read and write to this folder.

There are multiples subdirectories under the main WellingtonClinic directories. These include

scripts: This subdirectory contains all the scripts (except **file-system-setup.sh**) which are used by the staff members to perform their day-to-day tasks such as registering a new patient, searching a patient's record and/or changing the basic personal information about the staff member (see tasks for a list of required scripts and their access rights). These scripts need to be accessed and executed by the right staff only.

staff: This directory contains information about the WellingtonClinic staff categorized by their role (i.e. administrators, doctors, nurses, receptionists). Each role is assigned a dedicated subdirectory.

Each member belonging to a particular role is also assigned a subdirectory of their own in their associated role's main directory. Each staff member's dedicated subdirectory follows the naming scheme of: *first two letters of firstname and first two letters of surname followed by the year they joined the clinic.*

- e.g., *Mary Teresa (doctor), joined 1997 -> mate1997*
- e.g., *Phil McGraw (nurse), joined 2008 -> phmc2008*

Each staff member's dedicated folder contains the following file:

sbasicinfo.log: This file contains the basic personal information in the following format. All fields are separated by comma (,). This file can be accessed by all clinic staff but can only be modified by the administrators.

- firstname,surname,date-of-birth,date-joined-the-clinic,physical address,email,phone number

- Example: username: *mate1997* *-> Doctor*
- *Mary Teresa,03/08/1953,1997,173 Rnd street Kelburn,m.theresa@yahoo.com,04528293*

- Example: username: *phmc2008* *-> Nurse*
- *Phil McGraw,15/04/1984,2008,45 Razyn Street,Petone,Lower Hutt,philmcg@gmail.com,02384756*

- Example: username: *ansm2002* *-> Receptionist*
- *Andy Smith ,13/09/1974,2002,58 Foster avenue Wellington,smith.andy74@gmail.com,07284756*

patients

There is a subdirectory for each patient which contains all the information about that particular patient. Each subdirectory's name follows a pattern of: *first letter of firstname and last letter of surname followed by the date of birth of the patient in numerical format.*

- e.g., Masood Mansoori, date of birth 23/12/2001 => *mi23122001*

- There are two files in the patient subdirectory:
  i. **pbasicinfo.log**: This file contains the basic personal information in the following format. All fields are separated by comma (,). This file can be viewed by all clinic staff but can only be modified by the receptionists.
     - e.g., patient: *mi23122001*
     - firstname,surname,date-of birth,gender,physical address,email,registered(i.e., assigned) doctor(s)
     - *masood mansoori,23/12/2001,male,130 Arostreet,masood.mansoori@mail.com,mate1997*
     - *Lucia Blakeley,11/09/1980,Female,935 Massachusetts Avenue Hamilton 4562,lucyblak@outlook.com, mate1997,brki2018*

ii. **pmedicalrecord.log**: This file contains information about the patient's medical history. Only the registered/assigned doctors can see and modify the content. The information is in the following format. The information is sorted by date.

- e.g., patient *mi23122001*
- date-of-visit,doctor-examined,healthissue,medication,dosage
- *11/2/2020,mate1997,belly itch ,scratchicilin,2 per day,*
- *08/5/2021**,**brki2018,headache,paracetamol,3 per day*

Roles

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

1. **Administrator** user belongs to groups "root" and "sudo" and sets up the directory structure by running the script **file-system-setup.sh** once**.** This means all major directories (WellingtonClinic, staff (administrators, nurses, receptionists, doctors), scripts and patients) are owned by this user. They create the roles and carry out account maintenance if needed (e.g., assigning staff to new roles, changing permissions, setting ACLs). Only administrators are able to add, remove or modify staff account and information.

   Only the administrator can delete subdirectories and files (including patients, doctors, nurses and other staff files and folders (i.e. entire records)).

   Receptionists can register a new patient, write, and modify the basic personal information about the patients and save them in the system (using **patients.sh**). Only the receptionists can register and/or delete a patient. When a new patient is registered, they are assigned to a doctor. A patient can be assigned to many doctors. Deleting a patient deletes the entire subdirectory and file dedicated to that particular patient.

   Receptionists can read everyone's (including staff and patients') basic personal information (i.e. **sbasicinfo.log, pbasicinfo.log**). Receptionists however do not have full access to patients' medical records (**pmedicalrecord.log**).

   The Clinic currently has one dedicated receptionist (in this case Andy). If Andy is not able to work for some reason, Lucia may take up the role and work as the receptionist until Andy is back at work. Therefore, selected nurses may work as receptionists and not all receptionists are nurses.

2. **Doctors** can read and change their own patients' medical record (i.e., pmedicalrecord.log) and can also write a new prescription (only for their registered patients or for those whom they have been currently or previously been booked/assigned (see task 10). Doctors are not allowed to change a patient's basic personal information. They are also not able to view or modify any information about other doctors' patients (i.e., patients whom they are not assigned/registered to and have not previously examined).

3. **Nurses** have no access to the patient's entire medical record content (i.e. **pmedicalrecord.log).** They can however read some of the medical records of patients only. This includes registered doctors and past and current prescribed medication and dosage(s) by the patient. This information is extracted from each patient's medical record file (i.e. **pmedicalrecord.log).**
   As mentioned earlier, selected nurses can be receptionists when required.

Staff members can be patients too. Doctors however cannot be their own assigned doctors and write their own prescription. If staff are registered as patients, their information is also saved in the patients' directory and follows the guidelines and format for patients (see below).

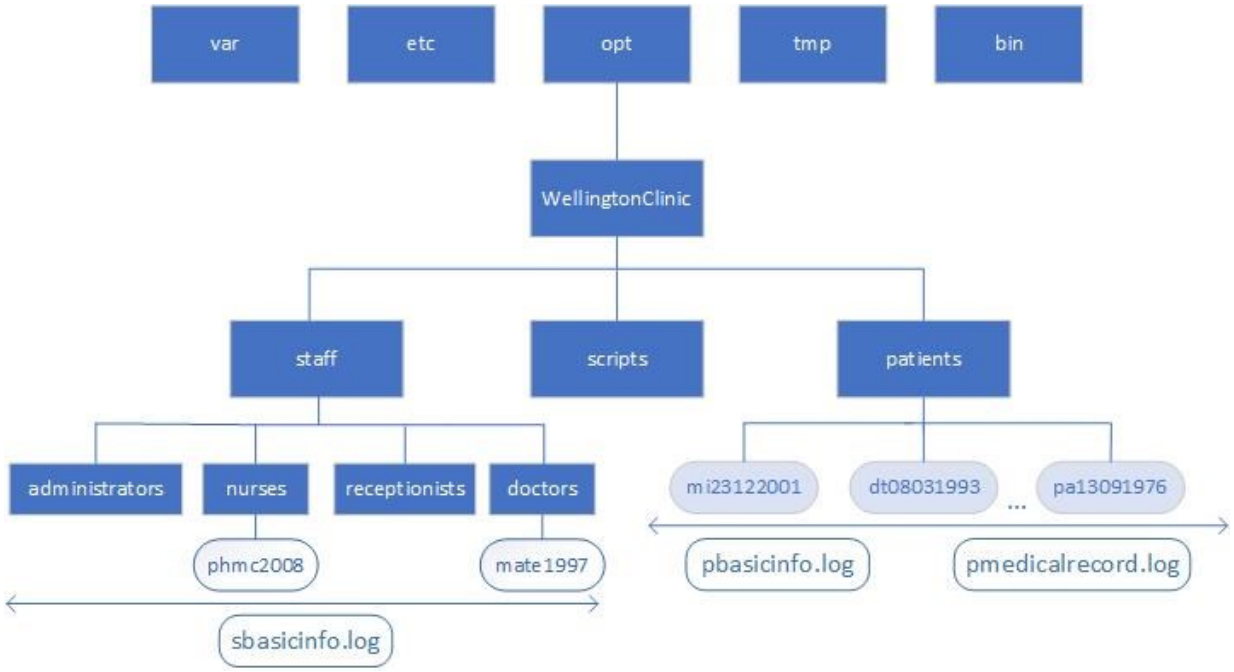

Figure 1 Directory and File Structure of WellingtonCLinic system

Files	sbasicinfo.log	pbasicinfo.log	pmedicalrecord.log	file-system-setup.sh	staff-create.sh	acls.sh	register-patient.sh	assign-doc.sh	search-patient.sh	search-doctor.sh	visit.sh	check-medication.sh	audit.sh
Owner	Adminstrator	Adminstrator	Adminstrator	Adminstrator	Adminstrator	Adminstrator	Adminstrator	Adminstrator	Adminstrator	Adminstrator	Adminstrator	Adminstrator	Adminstrator
administrator	rwx	rwx	rwx	rwx	rwx	rwx	rwx	rwx	rwx	rwx	rwx	rwx	rwx
registered doctor	r	r	rw	—	—	—	—	—	—	—	rx	r	—
un(non)registered doctor	r	r	—	—	—	—	—	—	—	—	—	—	—
nurse	r	r	r	—	—	—	—	—	—	—	—	rx	—
receptionist	r	rw	—	—	—	—	rx	rx	rx	rx	—	—	—

Figure 2 Access matrix for WellingtonClinic files

What you need to do (Tasks)

*****All scripts are saved in the scripts subdirectory, except file-system-setup.sh which must be kept outside the WellingtonClinic folder.**

1. (10 Marks) Create an access matrix of the directories shown in Figure 1 and created by the script **file-system-setup.sh (**see task 2**).** For each directory (group of directories) justify the permissions given (e.g., why did you have to give rwx permission to X directory for X staff)**. The permissions set on these directories must follow the principle of least privilege.**

2. (5 Marks) Write a Bash/Python script (**file-system-setup.sh**) to create the file system directory structure according to the information given in the case study. Only the administrator can execute this script. **This script must follow the principle of least privilege.**

3. (5 Marks) Write a Bash/Python script (**staff-create.sh**) to create the following staff and corresponding directories and files associated with each. The script should illustrate (include) object creation, assigning permissions. Only administrators are able to execute this script. This information must be embedded in the script and must not require user input.

   Doctors:
   1. *Mary Teresa, Date of Birth: 03/08/1953, Date Joined the clinic: 1997, Address: 173 Rnd street Kelburn, Email: m.theresa@yahoo.com, Phone: 04528293*
   2. *Breana Kipling, Date of Birth: 03/08/1991, Date Joined the clinic: 2018, Address: 4548 River Road HugoTown Auckland 5513, Email: szzh8@tempmail.net, Phone:071943668*
   3. *Mandy Dannel, Date of Birth: 15/12/1965, Date joined the clinic: 1993, Address: 343 Norma Avenue Dayton Napier 6731, Email: mandydl@gmail.com, Phone: 052637445*
   4. *Lance Bourne, Date of Birth: 07/04/1970, Date joined the clinic: 2002, Address: 25 Ferguson Street Franklin Greytown 8567, Email: lancb@outlook.com, Phone: 083736456*

   - **Nurses**:
     1. *Lucia Blakeley, Date of Birth: 11/09/1980, Date joined the clinic: 2004, Address: 935 Massachusetts Avenue Hamilton 4562, Email: lucyblak@outlook.com, Phone: 38347463*
     2. *Phil McGraw, Date of Birth: 15/04/1984, Date joined the clinic: 2008, Address: 45 Razyn Street Petone Lower Hutt 8435, Email: philmcg@gmail.com, Phone: 02384756*

   - **Receptionist(s)**:
     1. *Andy Smith, Date of Birth: 13/09/1974, Date joined the clinic: 2002, Address: 58 Foster avenue Wellington 5011, Email: smith.andy74@gmail.com, Phone: 07284756*
     2. *Lucia Blakeley, Date of Birth: 11/09/1980, Date joined the clinic: 2004, Address: 935 Massachusetts Avenue Hamilton 4562, Email: lucyblak@outlook.com, Phone: 38347463*
     3. *Phil McGraw, Date of Birth: 15/04/1984, Date joined the clinic: 2008, Address: 45 Razyn Street Petone Lower Hutt 8435, Email: philmcg@gmail.com, Phone: 02384756*

   - **Administrator(s)**:
     1. *Pauline Sanderson, Date-of-Birth: 08/03/1993, Date-joined-the-clinic: 1995, Address: 2452 Randolph Street Bedford Auckland 7752, Email: paulsand@admins.co.nz, Phone: 03747543*

4. (5 Marks) Explain in detail where the ACL information of an object is saved on a Linux system (EXT2, 3 and 4 file systems) and how your system keeps track of them.

5. (15 Marks) Write a Bash/Python script (**acls.sh**) to assign access rights using ACLs according to the policy rules and the provided and determined access matrixes (File and directory access matrixes). The script must illustrate (include) creation of ACLs according to the use case **and must follow the principle of least privilege.**

6. (5 Marks) Write **a** Bash/Python script (**register-patient.sh**) to register a new patient by creating the necessary folders and files. Only the receptionist (including the designated nurse) must be able to execute the script. The script must not allow a doctor to assign themselves to be the registered doctor. The script asks the receptionist to enter all the patient's basic personal information:

"Enter the following information about the patient:"
- *First name:*
- *Surname:*
- *Date of birth (23/12/2001):*
- *Gender:*
- *Physical address:*
- *Email:*
- *Phone number:*
- *Registered doctors (enter username separated by comma (,):*

And the patients are as following:

Patients:

1. *Masood,Mansoori,23/12/2001,male,130 Aro Street,masood.mansoori@mail.com,mate1997*
2. *David,Travert,08/03/1993,male,42 AZX ave. Thorndon Wellington,dtrt@gmail.com,mada1993*
3. *Peter,Garcia,13/09/1976,male,3 Kano street Kelburn Wellington,peterg@outlook.com,brki2018*
4. *Lance,Bourne,07/04/1970,male, 25 Ferguson Street Franklin Greytown 8567, lancb@outlook.com,083736456, mada1993,mate1997,brk2018*
5. *Lucia Blakeley,11/09/1980,Female, 935 Massachusetts Avenue Hamilton 4562,lucyblak@outlook.com,mate1997,brki2018*

7. (5 Marks) Write a script (**assign-doc.sh**) to add a doctor to the list of registered doctors for a patient. Only the receptionist (including the designated nurse) can execute this script. The information is written to pbasicinfo.log file. The script must ask the user to enter the name of the patient and the name or username of the newly registered/assigned doctor.

8. (5 Marks) Write a script (**searchpatient.sh**) to search and list all the patients ever examined by a particular doctor. This script can be executed by the receptionist only (excluding those who are also nurses). The script asks for the full name or username of a particular doctor. The script must display the information as follows:

Doctor	**Patients**
Mandy Dannel (mada1993)	David Travert, Lance Bourne

9. (5 Marks) Write a script (**searchdoctor.sh**) to search and list all the doctors who have ever been assigned to a patient. This script can be executed by the receptionist only (excluding those who are also nurses). The script must display the information as follows:

Patient	**Doctors**
Lance,Bourne	mada1993,Mary Teressa:mate1997,Breana Kipling:brk2018

10. (10 Marks) Write a Bash/Python script (**visit.sh**) to add information about a patient's visit. When a doctor executes this script and the doctor is in the list of registered doctors, it allows to add an entry to the records (please see **pmedicalrecord.log** for the information and the format.) therefore this script must only allow a doctor to write to a patients' pmedicalrecord.log file if the doctor executing it is in the list of the registered doctors for the patient.

11. (5 Marks) Write a Bash/Python script (**check-medication.sh**) for nurses to view a list of past and current medications and the registered doctor who prescribed them (may list the doctor's full name or just the username).

12. (10 Marks) A script file (**audit.sh**) which displays information on any changes in permission, object creation, deletion and modification on the WellingtonClinic directory, its subdirectories and files. The audit script should display the username performing the operation, the type of the operation, the object on which the operation takes place and the date associated with the access operation. This script is only run by the administrator(s) (See below).

Username	**Object**	**Operation**	**date**
mate1997	pmedicalrecord.log	write	date

13. (10 Marks) Nurses having r (read) access to the pmedicalrecord.log file is a serious vulnerability even though the access to information is filtered through the interface (i.e. check-medication.sh) to view the permitted information within the file only. Propose alternative methods to minimize the risk of confidentiality violation of patient's medical history information and justify the proposed method(s). Discuss in detail how your method would achieve that, implement your solution, and illustrate its effectiveness.

14. (5 Marks) – Writing, presentation, and Code readability (i.e. structured code, inclusion of #comments)

What to submit

Please submit one archive (zip, tar, gz) file containing the following items:

1. A document file (preferably .pdf file) which includes:
    - Cover page (including your name and student ID)
    - Access control matrix (Task 1)
    - Task 4
    - Task 13

2. Script files corresponding to each given task

Notes

- **Do not** make assumptions regarding roles, users, access rights and directory structures. Follow the case study strictly.
- You must strictly follow the principle of least privilege
- Using default ACLs is preferred (if possible)

- Please use shell scripting (preferred) or Python to implement the system. The default shell for ECS is "zsh" but bash could also be used. zsh is also the default bash for Mac OS.
- This is a good beginner's guide to writing scripts: http://tldp.org/LDP/Bash-Beginners-Guide/Bash-Beginners-Guide.pdf
- Netlab (accessible at netlab.ecs.vuw.ac.nz) has basic practical exercises on writing shell scripts!

Grading Criteria

The criteria for grading are:

- Completeness – Did you complete all the tasks and how comprehensively? There is no word limit to the report. Provide explanation where necessary.
- Accuracy - How well did you complete the tasks? Examples:
    - How detailed is your diagram in Task 3. Did you identify all subjects and objects and associated rights? (Only use the objects mentioned in this scenario)
    - Are access rights properly set? Can nurses view the patient's medical history?
- Scope - How thoroughly did you consider the problem? What design decisions did you have to You are encouraged to include diagrams with brief explanations where (if) necessary.
- Presentation - Did you use the right terminology? Please check for readability, we mark a lot of these and generally we look more favorably on well-structured and well-written ones.

Letter grades

A-range:

Complete, accurate, and well presented. Shows excellent knowledge and understanding of access control methods. Well-argued. Where required, contains good original input from the student.

- Example. Code is documented and well structured.

B-range:

Mostly complete, mostly accurate, and well presented. Shows a good knowledge and good understanding of the methods but either fails to complete some parts of the tasks or is unclear or is poorly argued.

C-range:

Satisfactory performance although some errors in accuracy and/or problems with presentation. Shows only some basic knowledge of the material or fails to understand some important parts of it, or does not provide solutions to a significant portion of the tasks.

D-range:

Poor performance overall, some evidence of learning but very problematic in all aspects mentioned above.

E-range:

Well below the required standard.