File System Access Control (ACL)

Due Date: 4/4/2021

Assignment 1 (100 Marks = 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 or Sharepoint.

Case study

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

System overview

Data and directory structure:

Assume that access to all resources is via the file system. Access to the following information needs to be controlled:

- Staff's basic personal information ("firstname,surname,date-of-birth,date-joined-the-clinic,physical address,email,phone number")
- Patients' basic personal information ("firstname,surname,date-of-birth,gender,physical address, email,phone number,registered doctor")
- Patients' medical record
- Patients' current medication

There is a subdirectory under the WellingtonClinic main directory called "scripts" (see image). This subdirectory contains all the scripts which are used by the staff members to perform their day to day tasks such as registering a new patient, searching for a patient's medical 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 only accessed and executed by the right staff.

The system keeps the information about a patient's information access rights in the ACLs set on each patient's subdirectory and file. This eliminates the need to keep track of each patient and doctor relationship. First time when a patient is registered, they are assigned to a doctor. The doctor's username is then added to the ACL of the patient's information (files and directories where applicable). If a booking is made for the patient with another doctor, the new doctor's name is also added to the ACL to have appropriate access rights according to the policy. This ensures that **not** all doctors get default access to all patients but only if they have (or are scheduled to) examine(d) the patient.

Each user must login into their own account to perform their designated activities. The system checks the logged user account to determine access rights. The logged user information is also used to keep track of access records and auditing purposes.

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

❖ Staff

All staff members have their own subdirectory under the main directory (i.e., staff) identified by their roles (e.g., doctors, nurses, receptionists, administrators). Each staff member's dedicated subdirectory follows their username 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*

All staff have a file in their dedicated directory which includes the basic personal information about the staff member.

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*

Staff members can be patients too. Doctors however cannot be their own 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).

1. **Administrators** are the only ones who have full control over the system. They initially setup the directory structure, create roles and carry out account maintenance if needed (e.g., changing initial permissions, setting ACLs and resetting passwords). Only administrators are able to add, remove or modify staff information. Administrators log in as a super user account on the system.

   No subdirectories and files can be deleted by staff. Only the administrator can delete items (including patients, doctors, nurses and other staff files and folders (i.e. entire records)).

2. **Receptionists** can register a new patient, make an appointment, write and modify the basic personal information about the patients and save them in the system (**patients.sh**). Only the receptionists can register a new patient. Receptionists can read everyone's (including staff and patients') basic personal information. Receptionists however do not have access to patients' medical records.

   When a new patient is registered, they are assigned to a doctor. If that particular doctor is not available, the patient can see other doctors.

   When an appointment is made, the patient's basic personal information is retrieved and displayed for the receptionist. If the registered doctor is not available, the receptionist can assign the patient to another doctor for that particular booking, in which case the new doctor will be able to read, write new prescription, and modify the patient's medical history (i.e. pmedicalinfo.log). The system does not keep track of doctor's availability, it assumes that other doctors could potentially be available.

3. **Doctors** can read and change their 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 booked or previously booked/examined). 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).

4. **Nurses** can read all patients' visit dates, medications and dosage. This information is extracted from each patient's medical record file (i.e. **pmedicalrecord.log).** Nurses have no access to the patient's entire medical record content.

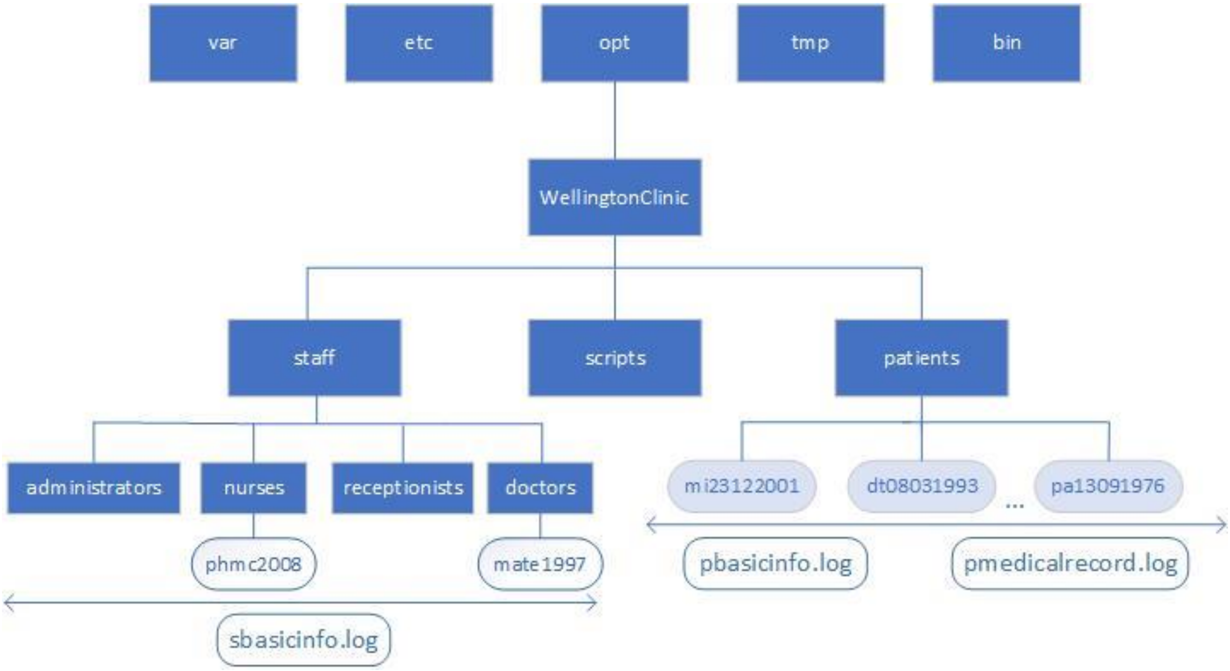
❖ Patients

Patients are not the users of the system therefore, they do not have any access rights. Their information is however saved on the system. Patients can be registered by any receptionists and examined by any doctors. Once a patient is booked and examined by a doctor, that particular doctor will have read and write access to the patient's medical records.

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 date of birth in numerical format.*

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

- There are two files in the patient subdirectory:
  1. **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,phone number,registered(i.e., assigned) doctor
     - *masood mansoori,23/12/2001,male,130 Aro Street,masood.mansoori@mail.com,08965193,mate1997*
  2. **pmedicalrecord.log**: This file contains information about the patient's medical history. Only the registered/assigned doctors and those who have been currently or previously booked with the patient (i.e. examined the patient) 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*



What you need to do (Tasks)

*****All scripts are saved in the scripts subdirectory (please see image).**

1. (5 Marks) Represent the policy using an access matrix. Associated rights are based on UNIX access rights and include: Read, Write and Execute. This must include identification of all subjects/objects and appropriate privileges/permissions.

2. (5 Marks) Write a Bash/Python script (**file-system.sh**) to create the file system directory structure according to the information given in the case study. Only the administrators are able to execute this script.

3. (10 Marks) Write a Bash/Python script (**staff-and-acls.sh**) to create the following staff and corresponding files associated with each. The script should illustrate (include) object creation and creation and/or modification of ownership. Only administrators are able to execute this script.

   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

   - **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) List the permissions and ACLs for all subjects/objects in the system. Explain the followings:

   - Do these ACLs match the access matrix? List any deviations from the access matrix
   - Explain any design decisions you had to make and the reason behind it.

5. [10] Explain in detail where the ACL information of an object is saved on a Linux system and how your system keeps track of them.
6. (20 Marks) Write a Bash/Python script (**acls.sh**) to assign access rights using ACLs according to the policy rules and your access matrix. The script must illustrate (include) creation of ACLs and setting proper permissions according to the use case.
7. (10 Marks) Write a Bash/Python script (**patients.sh**) which lists the following three options corresponding to each task and allows the user to select and execute them.

   - ➢ Register a new patient
   - ➢ Make an appointment (task 8)
   - ➢ Search for a patient (task 10)

   Register a new patient by creating the necessary folders and files. Only the receptionists (and root) must be able to execute the script. 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 doctor:*

And the patients are as following:

Patients:

1. *Masood,Mansoori,23/12/2001,male,130 Aro Street,masood.mansoori@mail.com,08965193,mate1997*
2. *David,Travert,08/03/1993,male,42 AZX ave. Thorndon Wellington,dtrt@gmail.com,04838372,mada1993*
3. *Peter,Garcia,13/09/1976,male,3 Kano street Kelburn Wellington,peterg@outlook.com,0575938,brki2018*
4. *Lance,Bourne,07/04/1970,male, 25 Ferguson Street Franklin Greytown 8567, lancb@outlook.com,083736456,mada1993*

8. (10 Marks) Write a Bash/Python script (**patients.sh**) to make an appointment for a patient. The script asks for patient's firstname, surname and date of birth. It then retrieves and lists the date and name of

all doctors visited in previous bookings by the patient. “If” a new doctor is scheduled to see the patient, proper access rights on patient files (and directories) must be granted to the new doctor.

9. (5 Marks) Write a Bash/Python script (**visit.sh**) to add information about a patient’s visit. This information is added by the patient’s doctor (pmedicalrecord.log).

10. (10 Marks) Write a script (**searchpatient.sh**) to ask for a doctor’s username and search and list all the patients ever examined by a particular doctor. The script can also ask for a patient’s first name, last name and date of birth and list all the doctors who have examined the patient. This script can be executed by all staff (not all users of the system but the clinic staff only).

11. (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 and the object on which the operation takes place. This file is only run by the administrator(s).

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 5
2. Script files corresponding to each given task (Tasks 2, 3, 6, 7, 8, 9, 10, 11)

Notes

- **Do not** make assumptions regarding roles, users, access rights and directory structures. Follow the case study strictly. Design decisions contradicting the use case MUST be explicitly mentioned in Task 4, explained and justified.
- 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.