P1L5 Access Control

Controlling Accesses to Resources

TCB - sees a request for a resource, how does it decide whether it should be granted?
Authentication establishes the source of a request
Authorization or access control answers the question if a certain source of a request is allowed to read the file.

Subject who owns a resource should be able to control access to it - this is not always true. There are some systems that will not allow sensitive data to be shared.

Controlling Accesses to Resources
Access control is basically about who is allowed access to resources.

There are two parts to access control:

1. Decide who should have access to certain resources (this is called an access control policy)
2. Enforcement - only accesses defined by the access control policy are granted.

Complete mediation - no one should be able to bypass access control and gain access to the TCB. *Complete mediation is essential for successful enforcement.*

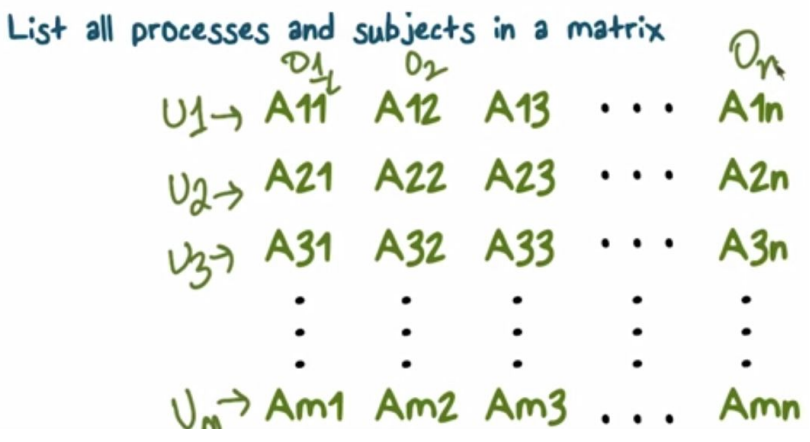
Access Control Matrix (ACM)

An access control matrix contains the information relevant to access control.
ACM:

row correspond to sources of the request : users/subjects/groups
columns correspond to resources that need to be protected

ACM[U,O] -- U=user, O=object, state captured is who has access to the resources of the system.

Implementing Access Control



A32 = ACM[U_3,O_2] … this describes the access rights of User#3 to Object#2

RWX - read, write, execute

Data Confidentiality Quiz
Confidentiality is about disclosure of data.
So the owner of a file should control who has read access to the file

Determining Access Quiz
An access control policy can define positive access: granting access to subjects.
If a user belongs to a specific group with access rights to an object, but the user is denied access rights to that object, then the negative access rights will take precedence. The user will be denied access.

Discretionary Access Control

In DAC access to a resource is at the discretion of its owner.
An example: Alice is owner of FOO. She grants read access to Bob but does not allow him to propagate this access.

This will not stop a third party from accessing the file. Bob can copy the file into another file and share that with the third party.

Implementing Access Control Using an ACM

The matrix is large, but sparse. Most resources are not usually shared amongst a lot of people.
We represent it in the system as:
Access Control List (ACL)

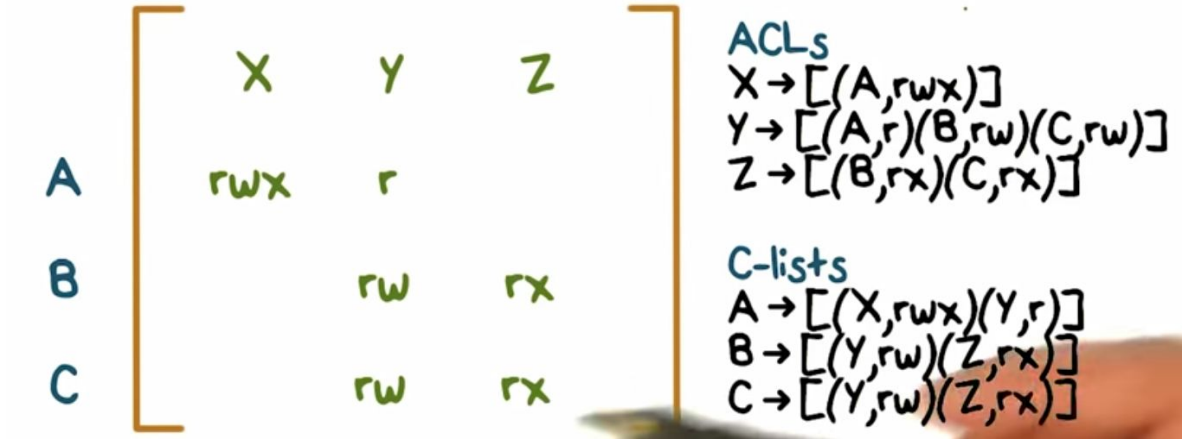
- Columns: for an object Oi ….[(ui1,rights1),(ui2, rights2), ...]

Capability List (C-List)

- Rows: for a user Ui ….[(Oi1, rights1),(Oi2,rights2), ...]

C-List- there will be one for each user.
ACL are for objects

Example Access Control Matrix



Users A,B,C
Objects X,Y,Z

To use ACLs -- look up by objects
To use C-Lists -- look up by user

ACL and C-List Implementation

Where should the ACL be stored?

- In the trusted part of the system
- Consists of access control entries (ACE)
- should be stored along with other object meta-data
- Checking requires traversal of the ACL

C-Lists Implementation

Where should C-List be stored?

- It is per user.
- A capability is an *unforgeable* reference/handle for a resource
- User catalogue of capabilities defines what a certain user can access
- Can be stored in objects/resources themselves (Hydra)

Sharing requires propagation of capabilities

- Need to add to the ACL
- cannot be forged

Possession of the capability means the user can access the resource.

ACL vs C-List
Most OS use ACLs

Efficiency - ACL are not as efficient as C-List when it comes to finding access rights.
Accountability - accountability can be found quite easily and in one place with ACL. With C-Lists each user's catalog must be checked to see if access to the object. ACLs are better at accountability.
Revocation - revoking access with ACL is easy, just remove it from the matrix. In C-List this is actually hard because one user does not have access to another user's catalog.

ACE Quiz

Alice goes to a movie and buys a ticket. She is allowed access because she has the ticket. This is an example of Capability because she has the ticket. The ticket taker does not need to know who Alice is, having the ticket is enough.

ACE Access Quiz
Some operating systems include deny or negative access rights. In this case, an access check procedure can terminate as soon as a negative or deny ACE is found, otherwise the whole ACL must be transversed.

Revocation of Rights Quiz
Revocation of access to certain access rights can be carried out easily in systems that use ACLs. ACLs require a traversal of the ACL, but C-Lists require every users catalogue must be checked.

Access Control Implementation in Unix-like Systems

- In Unix each resource looks like a file
- Each file has an owner, who has a unique user ID (UID). Access is possible for an owner, group, and world.
- Permissions are read, write, execute.
- The original ACL implementation had a fixed size representation (9 bits). The nine bits were a bit mask.
- Now full ACL support is available for many variants
- Setuid - this is for users to have write access during a specific period of time. For example, users playing a game may need write access to a specific file. When the user is done playing the game, they should not be able to access the file. It does this by temporarily changing the user id from the owner of the file to the player of the game.

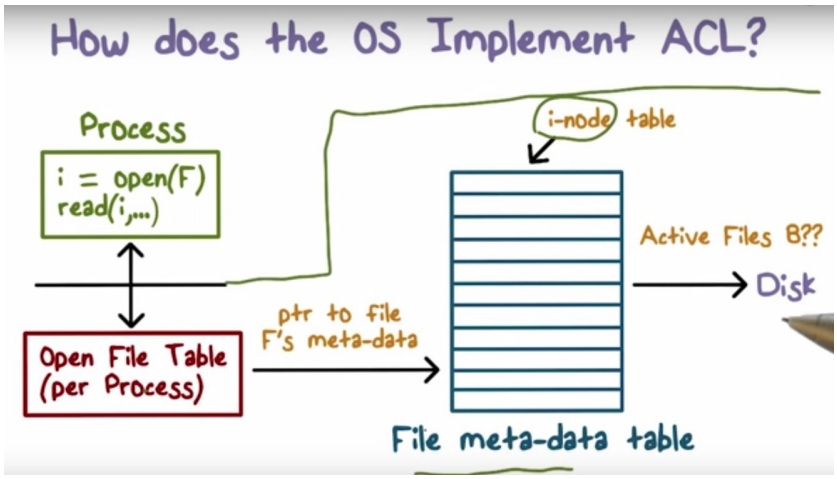
How are files used(system calls for accessing files?

- A resource must be created:
  Create(filename) /* there are several ways to do this */

The process who creates to the resource is the owner.

- To access the file, it needs to be opened.
  fd = open(filename, mode)
  mode → read, write, etc.
  the OS returns a small number that is called a descriptor
- to read the file:
  read (fd, buf, sizeof(buf))
  data will come from the file
- write(fd, buf, sizeof(buf))
- close(fd)
  will not need access to this resource anymore.

How Does the OS Implement ACL?



The line between the process and the Open File Table is the boundary between trusted and untrusted areas.

The OS keeps track of information about each file and its metadata, called an i-node. Open files are stored in the meta-data table. The file must be active.

The i is the index in the Open FIle Table, it is sent as a pointer to the metadata table.
Access control is done when the file is opened.
The ACL information is in the metadata table.

Time to Check vs TIme to Use Quiz (TOCTOU)
A vulnerability occurs when the file permission change after an open() call completes for the file and before it is closed.

Unix File Sharing Quiz
In Unix based systems a file cannot be shared by sharing its descriptor.

SetUID Bit Quiz
An executable file F1 has the setuid bit set and is owned by user U1.
When U2 executes F1 the UID of the process executing F1 is U1.

Role Based Access Control (RBAC)

In role based access control - the rights are associated by the roles.
-In an enterprise setting, access may be based on job function or role of a user.
- Users authenticate themselves to the system
-Users can then activate one or more roles

RBAC Benefits
-Policy need not be updated when a certain person leaves
-New employee should be able to activate the desired role
-Revisiting least privilege -- start with the minimum accesses available.
    -The user in RBAC can access only a subset of files because they are confined by their role.
-SELinux supports RBAC

RBAC Benefits:
In systems that do not support RBAC but allow user groups to be defined, benefits of RBAC can not be realized with groups. Groups are subject related, roles are resource related.

Fail-Safe Defaults implies that when an access control policy is silent about access to a certain user, that access must be DENIED when the user makes a request.