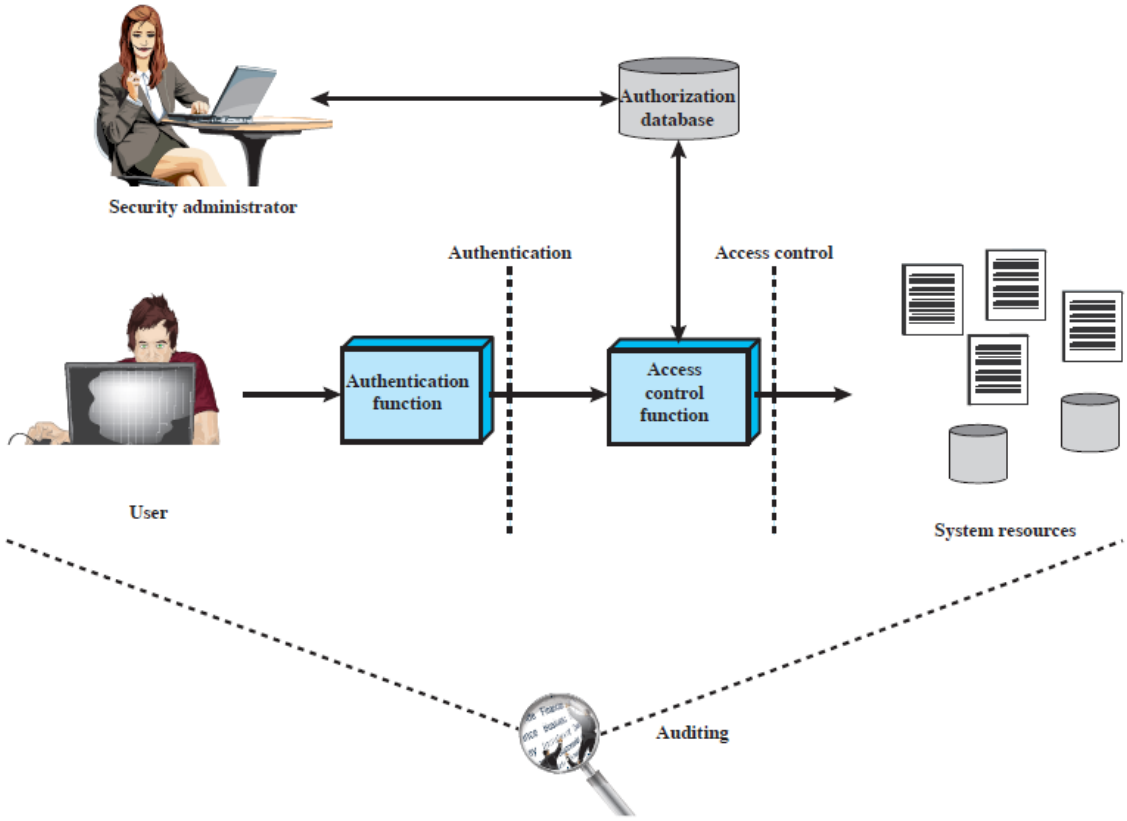
CAN302 W8

Overview of concepts

Access Control：

根据安全策略对系统资源的使用进行监管的过程，并且根据该策略，只有被授权实体（用户，程序，过程或其他系统）才允许使用该过程。

访问控制实现了一种安全策略，该策略指定谁或什么（例如，在进程的情况下）可以访问每个特定的系统资源以及每个实例中允许的访问类型。



Basic Elements of Access Control

- Subject：能够访问对象的实体，例如 learning mall 的用户
- Object：被控制访问的资源，object 是被用于包含或接收信息的实体，例如 learning mall 上的课程
- Access right：描述 subject 访问 object 的方式，例如 Read, Write, Execute, Delete, Create, and Search

Access Control Policies

访问控制策略规定了允许哪些类型的访问、在什么情况下以及由谁进行访问。

- Discretionary access control (DAC)
    - 根据请求者的身份控制访问，访问规则（授权）说明了允许（或不允许）请求者执行的操作
    - 一个实体可能具有访问权限，这些权限允许该实体根据自己的意愿使另一个实体能够访问某些资源

- Mandatory access control (MAC)
  - 通过对比具有安全许可的安全标签 (security labels with security clearances) 来控制访问
  - 集中控制 (Centrally controlled)，例如：用户无法授予访问权限
  - 用于军事信息安全
- Role-based access control (RBAC)
  - 根据用户在系统内的角色来控制访问权限，规则上声明允许给定角色中的用户进行哪些访问
- Attribute-based access control (ABAC)
  - 根据用户的属性、要访问的资源和当前环境条件控制访问

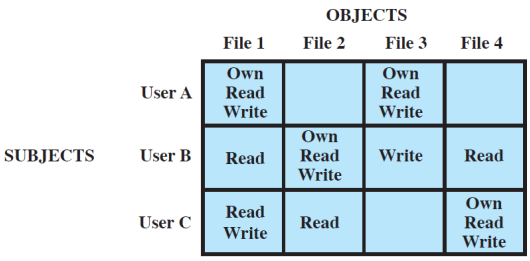
Access control policies and implementations

Discretionary access control

一个实体可能允许另一个实体访问某些资源的方案。

通常使用访问矩阵 (access matrix) 提供：

- 一个维度由可能尝试对资源进行数据访问的已识别 subject 组成，另一个维度列出可能被访问的 object
- 矩阵中的每个条目都表示特定 subject 对特定 object 的访问权限

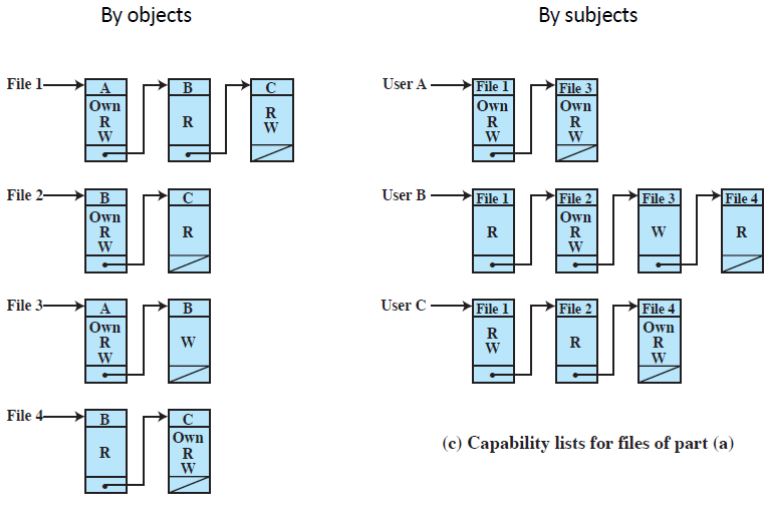


		OBJECTS			
		File 1	File 2	File 3	File 4
SUBJECTS	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix

Access Matrix and Other Access Control Data Structure：

- Access matrix
- Access control list：确定哪些使用者对特定资源具有哪些访问权限
- Capability list：确定特定用户可用的访问权限



(b) Access control lists for files of part (a)

(c) Capability lists for files of part (a)

- Authorization table

Subject	Access Mode	Object
A	Own	File 1
A	Read	File 1
A	Write	File 1
A	Own	File 3
A	Read	File 3
A	Write	File 3
B	Read	File 1
B	Own	File 2
B	Read	File 2
B	Write	File 2
B	Write	File 3
B	Read	File 4
C	Read	File 1
C	Write	File 1
C	Read	File 2
C	Own	File 4
C	Read	File 4
C	Write	File 4

Protection Domains

一组 object 以及对这些 object 的访问权限。就 access matrix 而言，一行定义了一个保护域。

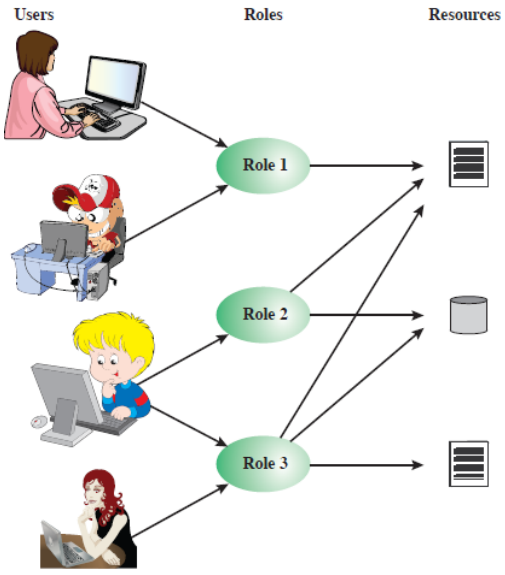
例如：UNIX。在用户模式下，某些内存区域受到保护，无法使用，某些指令可能无法执行；在内核模式下，可以执行特权指令，并且可以访问内存的受保护区域。

Role-based access control

根据用户在系统中的角色控制访问权限，以及规定给定角色中的用户允许哪些访问的规则。

相比之下，DAC是基于用户的身份。

将用户分配到角色/任务。用户与角色的关系是多对多的，角色与资源/对象的关系也是如此。用户到角色的分配可以是动态的。

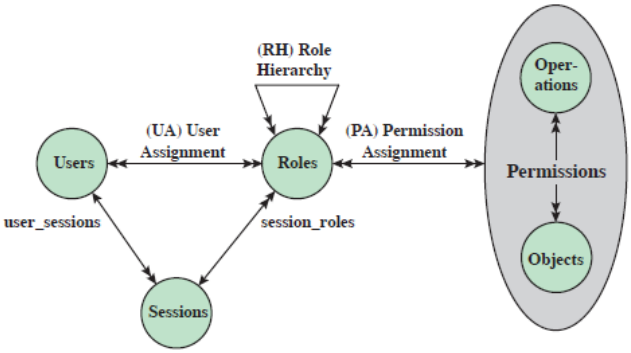


RBAC Reference Models

- $RBAC_0$：包含 RBAC 系统的最低功能
- $RBAC_1$：$RBAC_0$ + role hierarchies
- $RBAC_2$：$RBAC_0$ + constraints
- $RBAC_3$：$RBAC_1$ + $RBAC_2$

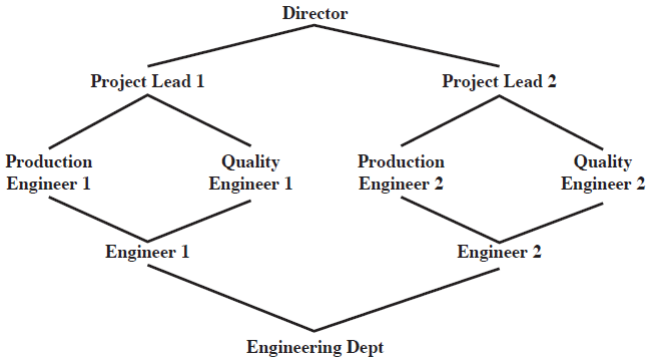
Base Model—$RBAC_0$

- User：有权访问此计算机系统的个人
- Role：给 user 分配 role 的系统
- Permission：批准对一个或多个 object 的特定访问模式
- Session：用户与用户分配到的角色集的已激活子集之间的映射 (比如用户有多个 role，建立用户和 role 子集的关联)



Role Hierarchies—$RBAC_1$

角色层次结构提供了一种反映组织中角色的层次结构的方法。



RBAC Constraints—$RBAC_2$

提供一种使 RBAC 适应组织管理和安全策略细节的方法。角色之间的已定义关系或与角色相关的条件。

Type：

- Mutually exclusive roles
  - 一个用户只能被分配到集合中的一个角色（在会话期间或静态），任何权限（访问权限）只能授予集合中的一个角色。
- Cardinality
  - 设置相对于角色的最大数量
- Prerequisite roles

- 规定仅当用户已分配给某个其他指定角色时，才能将用户分配给该角色

Attribute-based access control

一个相对较新的访问控制技术。可以定义表达资源和 subject 属性的条件的授权。优点在于它的灵活性和表现力。

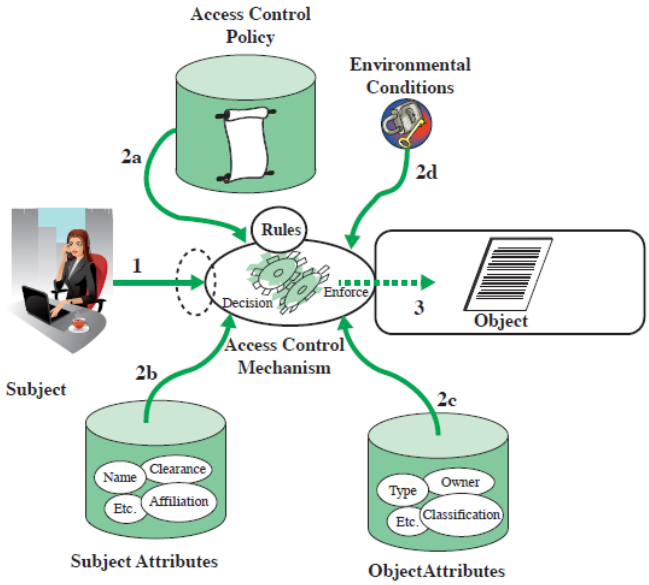
Three key elements to an ABAC model：

- attributes
- policy model
- architecture model

Attributes

- Subject attributes
  - 每个 subject 都有相关的属性，这些属性定义了 subject 的身份和特征
- Object attributes
  - Object 具有可用于做出访问控制决策的属性，例如 Word 的日期，作者等
- Environment attributes
  - 到目前为止，在大多数访问控制策略中基本上被忽略了

ABAC Logical Architecture



根据 object 和 subject attributes，以及 rule 判断 subject 是否可以访问 object。

ABAC Policies

策略是一组规则和关系，它们根据 subject 的权限以及在何种环境条件下如何保护资源或 object 来控制组织内允许的行为。

通常从需要保护的 object，和 subject 可用的权限的角度编写。

An ABAC policy model

- S, O, E stand for subjects, objects, and environments respectively
- $SA_k\ (1 \le k \le K), OA_m\ (1 \le m \le M), EA_n\ (1 \le n \le N)$ are pre-defined attributes for subjects, objects, and environments, respectively
- $ATTR(s), ATTR(o), ATTR(e)$ are attribute assignment relations for subject s, object o, and environment e, respectively.
  - Also used for the value assignment of individual attributes. For example,

$$Role(s) = \text{“}Module\ Leader\text{”}$$
$$ServiceOwner(o) = \text{“}xjtlu\text{”}$$
$$CurrentDate(e) = \text{“}04-15-2021\text{”}$$

- A Policy rule
  - decides on whether a subject s can access an object o in a particular environment e
  - is a Boolean function of the attributes of s, o, and e:

给 s、o 和 e 的所有属性赋值，如果函数的评估为 true，则授予对资源的访问权限；否则，访问将被拒绝。

Example：online entertainment store

该商店以固定的月费将电影传输给用户。商店必须根据用户的年龄和电影的内容分级强制实施以下访问控制策略。

Movie Rating	Users Allowed Access
R	Age 17 and older
PG-13	Age 13 and older
G	Everyone

- RBAC approach
  - Roles
    - Adult, Juvenile, or Child
  - Permissions
    - Can view R-rated movies, Can view PG-13-rated movies, and Can view G-rated movies.
- ABAC approach

  ABAC 方法不需要显式定义角色。相反，用户 u 是否可以访问或观看电影将通过评估策略规则来解决，如下所示：

```
R1:can_access(u, m, e) ←
    (Age(u) ≥ 17 ∧ Rating(m) ∈ {R, PG-13, G}) ∨
    (Age(u) ≥ 13 ∧ Age(u) < 17 ∧ Rating(m) ∈ {PG-13, G}) ∨
    (Age(u) < 13 ∧ Rating(m) ∈ {G})
```