Information and Network Security

Access Control

Key Concepts in Access Control

- Identification
- Authentication
- Authorization
- Data/message integrity
- Accountability
- Non-repudiation

Identification

- It is the act of Identifying an entity from many.
- **Recognizing one from many**

Authentication

- It is the act of Identity Verification
- **Bob needs to prove he is who he claims to be.**
- Three General Ways to authenticate and verify identity:
  - Something you ***know (i.e., Passwords)***
  - Something you ***have (i.e., Cards)***
  - Something you ***are (i.e., Biometrics)***

Something you *KNOW*

- Example: Passwords
  - Advantages:
    - Password schemes are simple to implement compared to other authentication mechanisms, such as biometrics
    - Password schemes are Simple for users to understand

Something you *KNOW*

- **Example: Passwords**
  - **Disadvantages:**
  - **First Disadvantage: (Cracking easy passwords)**
    - Most users do not choose strong passwords, which are hard for attackers to guess
    - **Solution: Strong passwords including combinations of Alphabets, numbers and Special characters.**
  - **Second Disadvantage: (Password Stealing)**
    - **One Time Passwords**

      Problem: The major problem with this system is that no user will be able to remember all these password.
      Solution: A device could be used that keeps track of all the different passwords the user would need to use each time she logs in.

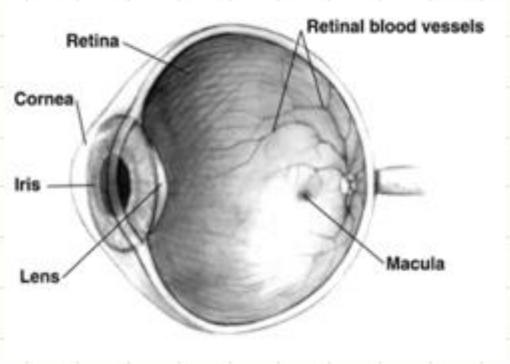
Something you *HAVE*

- **OTP Cards :** The OTP card is a One Time Password generator. When the code button is pushed a new dynamic password is displayed on the card. One such product, offered by RSA Security, is the SecurID card (other companies have different names for such cards). The SecurID card is a device that flashes a new password to the user periodically (every 60 seconds or so).
- The server knows the algorithm that the SecurID card uses to generate passwords, and can verify the password that the user enters.

Smart Cards: Another mechanism that can authenticate users based on something that they have is a smart card. A smart card is tamper-resistant, which means that if a bad guy tries to open the card or gain access to the information stored on it, the card will self-destruct.
ATM Card : You know what it is ☺

Something you *ARE*

- Biometrics
  - Palm Scan
  - Iris scan
  - Retina Scan
  - Fingerprinting
  - Voice Identification
  - Facial Recognition
  - Signature Dynamics



Disadvantages- Something You Are

- The key **disadvantages** to these biometric authentication techniques are the number of false positives and negatives generated
- A **false positive** occurs when a user is indeed an authentic user of the system, but the biometric authentication device rejects the user.
- A **false negative**, on the other hand, occurs when an impersonator successfully impersonates a user.
- **Social Acceptance** also plays an important factor.

Two Factor Authentication

- Authentication based upon two methods.
- ATM cards are an example of two-factor authentication at work.

Authentication

- People vs People Authentications
- People vs Computer Authentications
- Computer vs Computer Authentications
  - Client authentication
  - Server authentication
  - Mutual authentication
- Example : Kerberos

Authorization

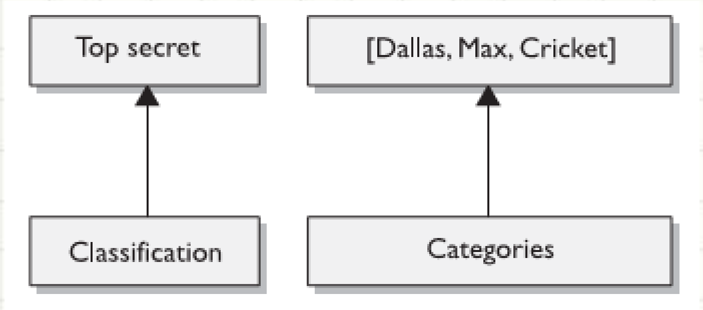
- **Permission to conduct some action**.
  - **Authentication is about verifying identity**,
  - **authorization is verifying a user’s authority.**
- Example : Use of ATM to withdraw Money you don’t have.

Authorization

- **Access Control Models**
- **Mandatory access control (MAC) model**
- In the MAC model, the computer system decides exactly who has access to which resources in the system.
- In the MAC model, if Alice creates a new document, the system can decide that no one but Alice is allowed to access that document.
- Alice herself does not have the right to decide who else is allowed to access the file that she authored.
- Even if she wants to share the document she authored with her friend Bob, she is not authorized to make that decision.

Authorization

- **Access Control Models**
- **Mandatory access control (MAC) model**
- Security Labels
  - Classification
  - Clearance
- Categories
  - Need to Know
- In a MAC model, only the computer system determines who is authorized to access documents that Alice creates.



Authorization

- **Access Control Models**
- **Discretionary Access Control (DAC)**
- The DAC model is different from the MAC model in that users are authorized to determine which other users can access files or other resources that they create, use, or own.
- In a discretionary access system, Alice could let Bob access a file at her discretion by issuing a command to the system, and then Bob would be given access to that file.

Authorization

- **Access Control Models**
- **Discretionary Access Control (DAC)**
- For instance, in UNIX, which uses a DAC model, Alice could issue the command to allow all users on the system to read the file.
- The ACL that results from such a command is shown in Table 1-4, in which the third row specifies that every user (denoted by *) has read privileges for the file /home/Alice/product_specs.txt.

```
chmod a+r /home/Alice/product_specs.txt
```

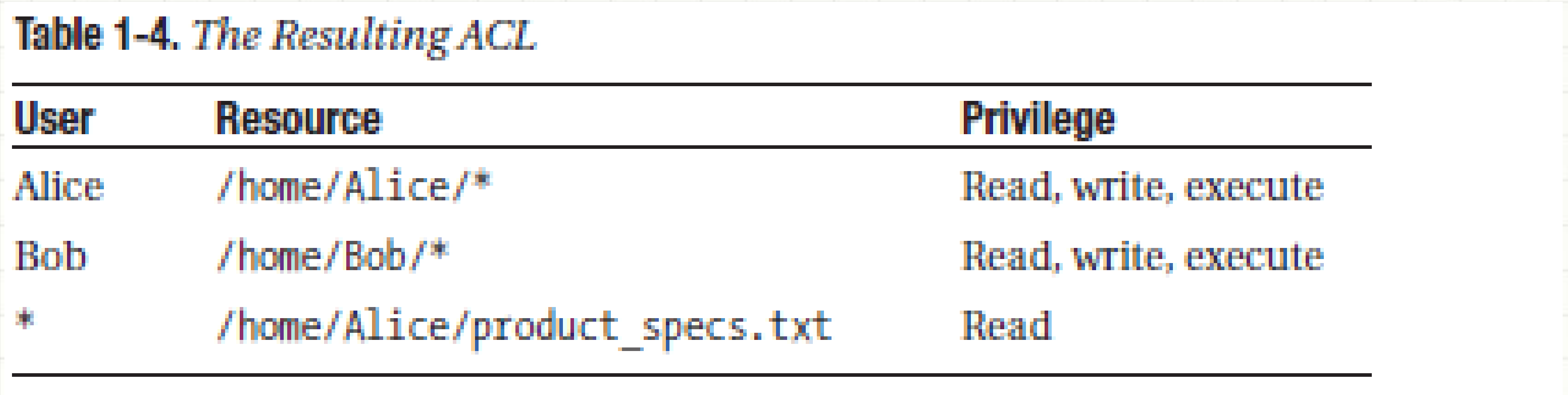


Table 1-4. *The Resulting ACL*

User	Resource	Privilege
Alice	/home/Alice/*	Read, write, execute
Bob	/home/Bob/*	Read, write, execute
*	/home/Alice/product_specs.txt	Read

Authorization

- **Access Control Models**
- **Role-Based Access Control (RBAC)**
- The third access control model is the RBAC model, which is similar to the MAC model in the sense that the system decides exactly which users are allowed to access which resources—but the system does this in a special way.
- A RBAC system will incorporate the user’s role into its access decision.

Authorization

- **Access Control Models**
- **Role-Based Access Control (RBAC)**
- As per the role-based ACL shown in Table 1-3, a backup operator is allowed to read data from all user home directories (/home/*) so that the data can be archived.

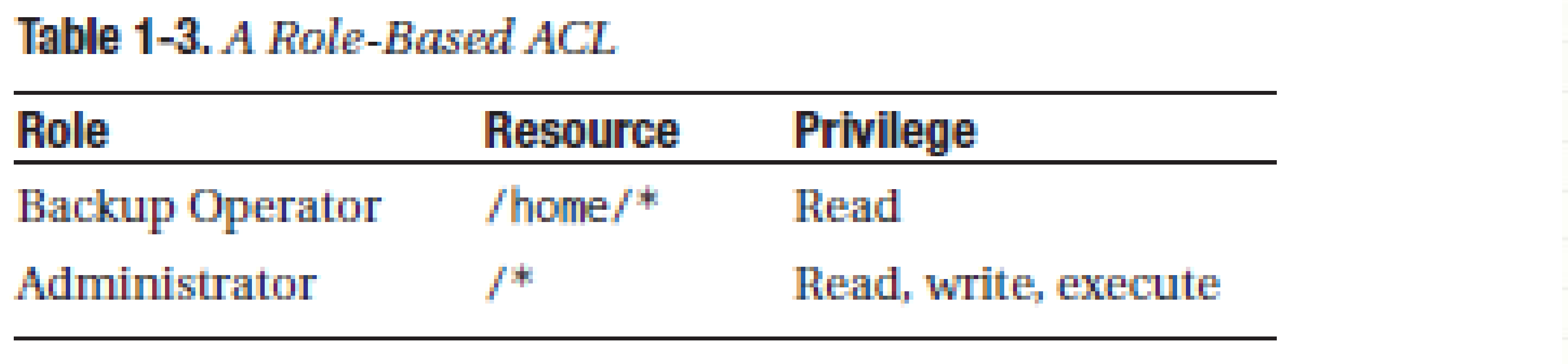


Table 1-3. *A Role-Based ACL*

Role	Resource	Privilege
Backup Operator	/home/*	Read
Administrator	/*	Read, write, execute

Authorization- Implementation

- **Access Control Lists (ACLs)**
- Minimally, an ACL is a set of users and a corresponding set of resources they are allowed to access.
- An entity (or a process) that is capable of being authenticated is often referred to as a principal.
- For example, Alice may have access to all the files in her home directory, but may not have access to Bob's files.
- Suppose Alice's home directory is /home/Alice, and Bob's home directory is /home/Bob. An ACL that models this is shown in Table 1-1.

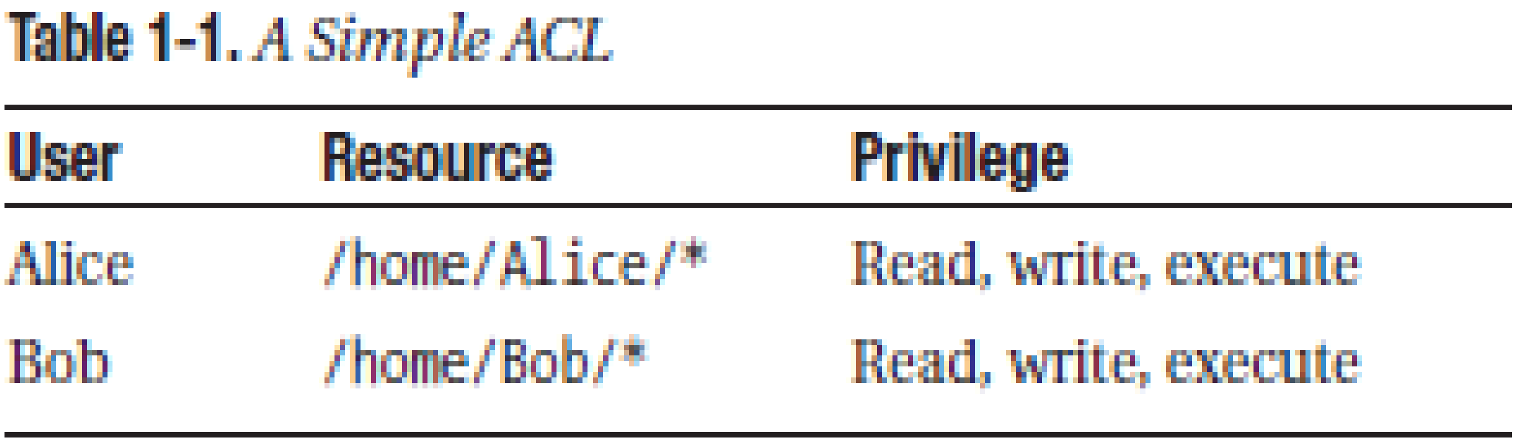


Table 1-1. *A Simple ACL*

User	Resource	Privilege
Alice	/home/Alice/*	Read, write, execute
Bob	/home/Bob/*	Read, write, execute

Authorization

- **Access Control Lists (ACLs)**
- In some more sophisticated ACL schemes, another piece of information called a *role* is added, which enables a user or principal to access particular resources.
- Table 1-2 shows an example mapping of users to roles, and
- Table 1-3 shows a role-based ACL

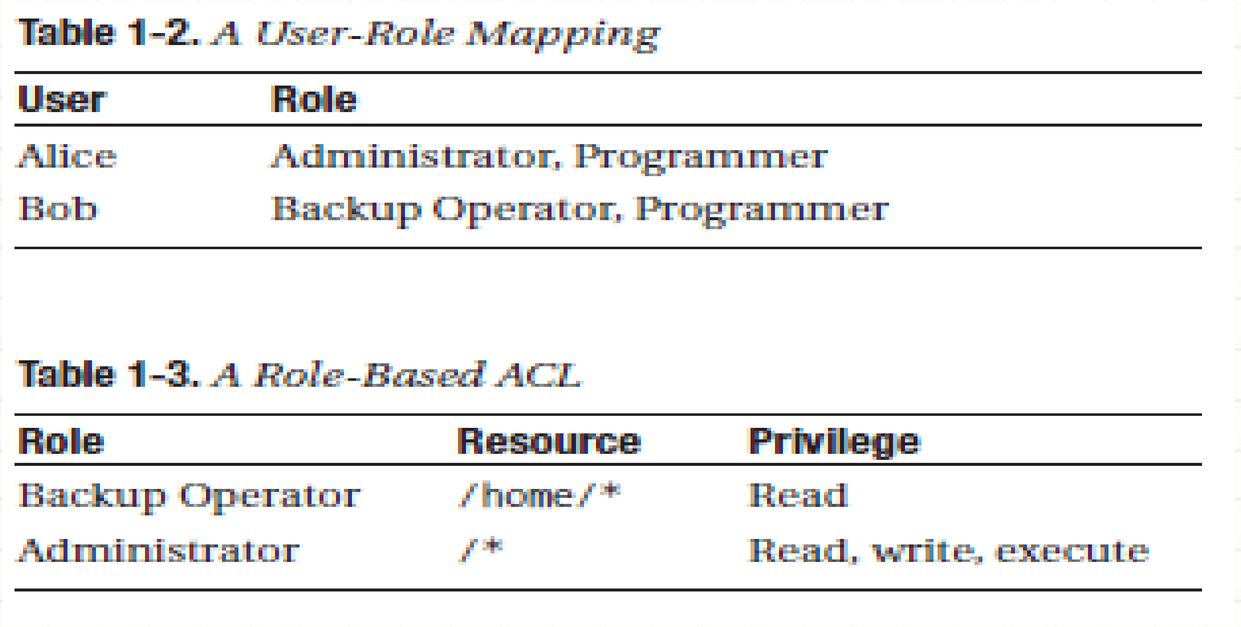


Table 1-2. *A User-Role Mapping*

User	Role
Alice	Administrator, Programmer
Bob	Backup Operator, Programmer

Table 1-3. *A Role-Based ACL*

Role	Resource	Privilege
Backup Operator	/home/*	Read
Administrator	/*	Read, write, execute

Accountability

- The goal of accountability is to ensure that you are able to determine who the attacker or principal is in the **case that something goes wrong or an erroneous transaction is identified**.
- In the case of a malicious incident, you want to be able **to prosecute and prove that the attacker conducted illegitimate actions.**
- In the case of an erroneous transaction, you want to identify which principal made the mistake.

Accountability

- Most computer systems **achieve** accountability through authentication and the **use of logging and audit trails**.
- It is also crucial to make sure that when the logging is done and audit trails are kept, the logs cannot be deleted or modified after the fact.

Accountability

- **To prevent logs from being deleted or altered**, they could immediately be transferred to another system that hopefully an attacker would not be able to access as easily.
- **MACs (message authentication codes)** can be used to construct integrity check tokens that can either be added to each entry of a log or associated with an entire log file to allow you to detect any potential modifications to the system log.
- You can also use **write once, read many (WORM) media** to store system logs, since once written, these logs may be hard (or even physically impossible) to modify—short of destroying the media completely.

Non-repudiation

- **The goal of non-repudiation is to ensure undeniability of a transaction by any of the parties involved. A trusted third party, such as Trent, can be used to accomplish this.**
- For example, let us say Alice interacted with Bob at some point, and she does not want Bob to deny that she interacted with him. Alice wants to prove to some trusted third party (i.e., Trent) that she did communicate with Bob.

Non-repudiation

- **To summarize, trusted third parties can help conduct non-repudiable transactions.**
- In general, non-repudiation protocols in the world of security are used to ensure that two parties cannot deny that they interacted with each other.
- In most non-repudiation protocols, as Alice and Bob interact, various sets of evidence, such as receipts, are generated.
- The receipts can be digitally signed statements that can be shown to Trent to prove that a transaction took place.

Non-repudiation

- Unfortunately, while non-repudiation protocols sound desirable in theory, they end up being very expensive to implement, and are not used often in practice.

Questions?