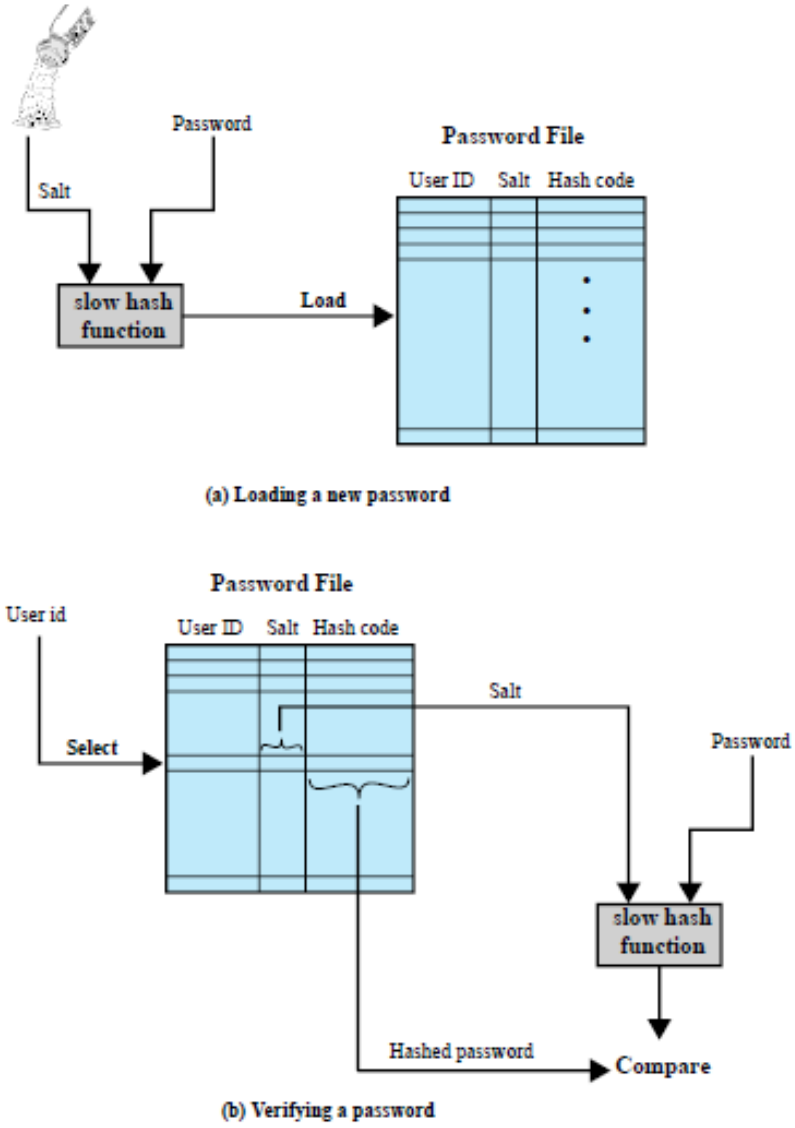
SOLUTION

Time: 60 Minutes. **Max Points**: 15

Question 1: [CLO # 1] **[1.5 + 1 = 2.5 Points]**

Illustrate the Unix password scheme with the help of a labelled diagram. Why slow-hash is important? Justify your answer.

Solution:



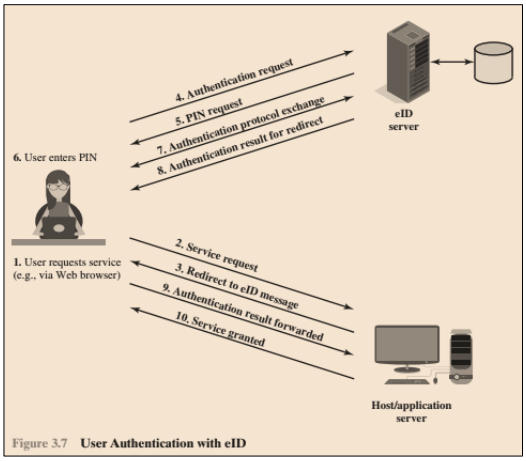
Slow hash is required to add computing delays for attackers trying to break the passwords. A rapidly computed algorithm could make brute-force attacks more feasible, especially with the rapidly evolving power of modern hardware.

Question 2: [CLO # 1]

[1.5 + 1 = 2.5 Points]

a) Illustrate a scenario where Electronic Identity cards (eID) hosted at NADRA eID Server is used by daraz.pk for the sale of small self-defense firearms.
NADRA eID server not only authenticate the user. Optional: It can even do a two factor authentication (without the knowledge of daraz.pk) using the citizens mobile phone number

b) Why biometric authentication is both strong and appropriate in your opinion?
NADRA stores the citizen biometric info. Firearm sale is risky and biometric authentication provides added security (two or multi-



factor). It is appropriate, as there is a shipment delay where any physical risk assessment can be performed and action taken.

Question 3: [CLO # 2] **[1.5 + 1 = 2.5 Points]**

Discuss the key similarities and differences between DAC, RBAC, and ABAC access control models / types. Also give two concrete examples of using some of those four access control types in your real applications or systems.

Solution:

All access control policies contain subjects (owners, groups, and world), objects, and access rights. DAC allows for one entity to grant another entity the same access to the same resources and is based on the identity of the user. All information regarding this scheme is contained in organized lists (e.g. Access Matrix, ACL, capability list). One example of this is the log-in system at Mines. Depending on your identity (username), you are granted certain capabilities. For example, I cannot download software on a school computer, but someone with an administrative ID can.

RBAC is not based on the user's identity but the role they play in the organization. Each role has specific access rights and those rights will likely not change frequently. The original RBAC model can be modified to include constraints (mutual exclusivity) or hierarchies. An example of a RBAC model is implemented in the Dresdner Bank, which grants access rights in order of job ranking. In a hierarchical setting, it is assumed the higher position will obtain all the access rights of the lower group.

The ABAC model uses attributes to define subjects, objects, and environment, and is capable of enforcing all the other models. The decision made by access control is dependent on four sources of information: subject/object attributes, AC policy, and environmental conditions.

Question 4: [CLO # 2] **[2.5 Points]**

Consider Bank Al-Falah is using an international banking software that has a database schema containing 1000 tables and 700 views to the internal web portal that has 2000 different web pages (use of DML and query) and 1200 reports. Specify (brief description + illustration) a role-based access control system for this scenario to avoid disclosure of unrelated information. Job functions are cashier, utility bill person, etc. and users are assigned different roles based on their job function(s).

Solution:

Job functions to web application interface mapping are created resulting in schema object (tables, views, etc.) permission. These permission are assigned to roles. The web application assigned will assign one or more roles to each user. For example, a cash/cheque counter operation is assigned cash_check_entry ROLE and cash_check_query ROLE. A valid answer should state grant of permission of database object (probably GRANT SQL DCL command) to different roles, Assignment of roles to uses and at least one job function with related roles.

Question 5 [CLO # 3] **[2.5 Points]**

a) Does this query have a SQL injection problem? Justify.

```
$sql = "SELECT * FROM employee
WHERE eid=’SHA2($eid, 256)’ and password=’SHA2($passwd, 256)’";
```

b) The following SQL statement is sent to the database to add a new user to the database, where the content of the $name and $passwd variables are provided by the user, but the EID and Salary field are set by the system. How can a malicious employee set his/her salary to a value higher than 80000?

```
$sql = "INSERT INTO employee (Name, EID, Password, Salary) VALUES ('$name', 'EID6000', '$passwd', 80000)";
```

Solution:

a) It still has a SQL injection problem. For example, we can let eid be "x, 256)' OR 1=1 #".

b) Let's assume the user wants username as john and password as iheartburgers. By setting $name to john and $passwd to iheartburgers', 2000000)# a malicious employee can set his/her salary to 2Million

Question 6: [CLO # 3] **[1 + 1.5 = 2.5 Points]**

a) An example of cascading authorizations phenomenon is shown in figure 1. Suppose at t =70 B revokes the rights from C, redraw the figure to show the effects and explain your answer.

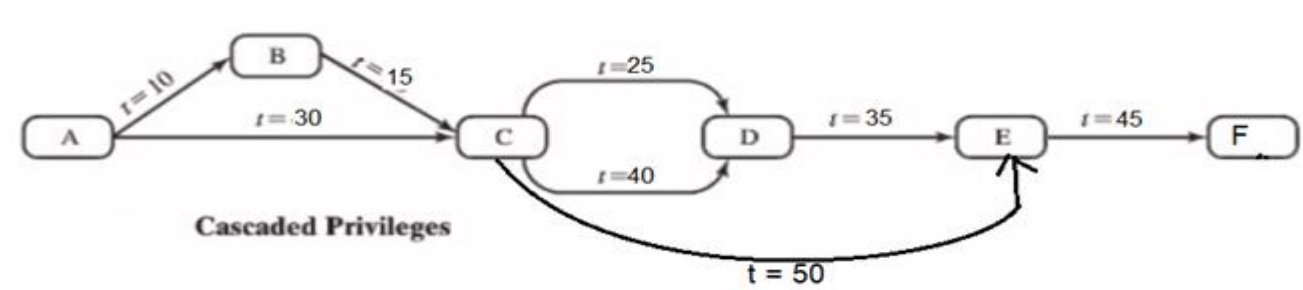
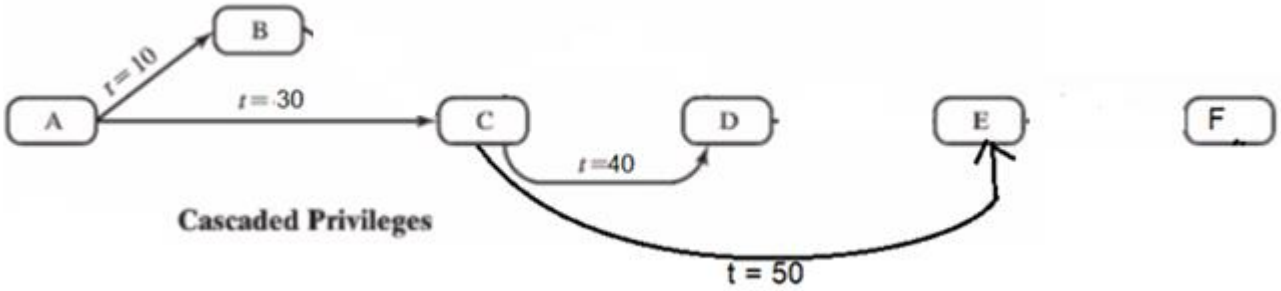


Figure 1 Cascading Authorizations

Solution:



The resulting diagram is shown above. Reason is time based which should be properly explained by the student for full marks.

b) Another example of cascading authorizations phenomenon is shown in figure 2. Suppose at t =75 Bob revokes the rights of Chris, redraw the figure to show the effects and explain your answer.

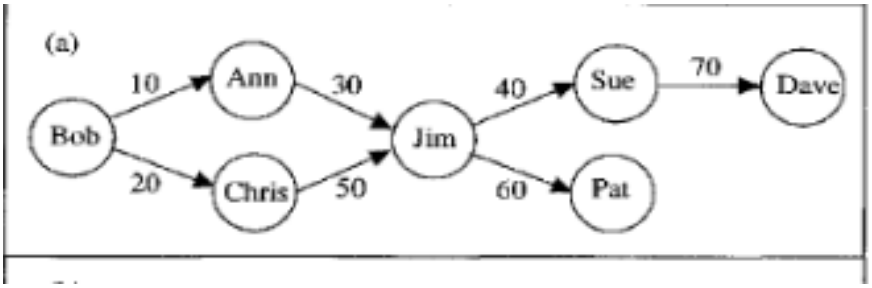
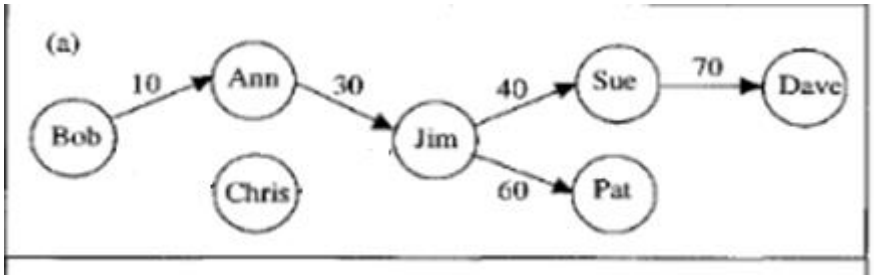


Figure 2 Grant Privileges

Solution:



The rights of Chris given to Jim will also be revoked. Reason is time based which should be properly explained by the student for full marks.