Pass Task 8.1. Security Concerns in Enterprise Application

1. Authentication: In software security, authentication is a process to verify if an user is a valid user and can access to the system. The most common method of authentication is using Keyword data such as Username & Password validation. Some other examples authentication method are using key or KeyCard to touch on a door gate, biometric authentication such as using finger print or eye scanner.

2. Authorization: After an user has been authenticated, authorization is the process to determine which resources can be made accessible to this user. Such information can be form in a set of policies, rules or permissions. For example, authorization in some residential building are provided in building KeyCard access. Each resident will be given a Card which can be used to open the entrance door but can only allow lift to access to the floor that the resident lives in.

3. Confidentiality: A security system will not be secured if there is user, who is able to pass the security, give up his/her access to other third parties. Therefore, confidentiality define a set of rules that make all users of the security system to make a promise on limit and restrict certain type of information out of public. In real world example, most commonly, the word "Confidential" appearing on internal documents denotes that they must be restricted to internal audience only, that is, the information should only be seen by the people who are authorized with access.

4. Data Integrity: This term is refer to the consistency and accuracy of information stored in a database system. This is to make sure that there is no leak in the data system and the information stored will not contradict each other. Firstly, they must be stored correctly in proper storage location and then they must be stored in full. For example, a text data should be stored in a column that can store alphabet characters. Database management system provides function to define constraints & rules that checking data consistency before it can be persisted.

5. Role-based Authentication: On an example of residential building security, the security model is hardly changed and it is quite easy to manage the access permission by using a key card. However, in an enterprise environment, user assignments is changing over time and is complex and hard to manage. Therefore, a role-based authentication is using to define a set of roles, permissions and role-permission relationship so that user assignment can be easy. Instead of assigning each user with a set of permissions, a role-based authentication model assign these permissions to a role, then later assign users to the proper role they stay on. For example, a role in IT department can have access to all equipment and computer system but not to Human Resources system. When a new IT staff is recruited to the team, this staff can be assigned to the IT role instead of assign each individual permission item to him.

6. Security Principals: This is to define practices to implement a security model in a system. Depending on the security policies of each different environment & company, the security principals could employ more or less practices to secure their systems. Each security model that being implemented should be adhere to these principals as a form of integrity of the security model itself. This set of practices sometimes can be referred to as Security Policy Framework.

7. Security Role: In a role-based authentication model, security role is a set of permissions that are related to each others that are closely map to the role model in real world. Users with the proper authentication can be assigned to a security role if he is granted with all the permissions available within that role.

8. Role Provider: In .Net, RoleProvider is a provider class that defines the contract that ASP.NET implements to provide role-management services using custom role providers. This RoleProvider define the methods related to role management such as CreateRole, DeleteRole, GetRoles as well as User Role assignment service such as: AddUsersToRoles, FindUsersInRoles, IsUserInRole…

9. Different types of Role Provider: In .Net, there are 3 built-in RoleProvider classes:
- *SqlRoleProvider*: a role provider that manages storage of the role membership information in a SQL Server database.
- *WindowsTokenRoleProvider*: a role provider that integrate with Windows group memberships which are usually defined in local Windows User Management & Security Policies,
- *AuthorizationStoreRoleProvider*: a role provider that manage storage of role-membership information in an authorization-manager policy store such as XML file or Active Directory server.

10. SqlRoleProvider: Is one of the RoleProvider that manage storages of the role membership information for an ASP.NET application in a SQL Server database. This provider can be used side-by-side with SqlMembershipProvider to provide a complete implementation of a Role-based authentication model for an ASP.Net Application in a Sql Server Database.