Define the following secure design principles and provide an example of each.

1. Separation of privilege

Definition

A protection mechanism that requires two keys to unlock it is more robust and flexible than one that allows access to the presenter of only a single key.

The reason is that, once the mechanism is locked, the two keys can be physically separated and distinct programs, organizations, or individuals made responsible for them.  From then on, no single accident, deception, or breach of trust is sufficient to compromise the protected information.

Example

The Unix operating system based on Berkeley-based versions is an example.

Users in this system are not allowed to change from their account to the root account unless two conditions - the root password and group ID, are met. Meeting either condition is not sufficient to acquire root access.

2. Least privilege

Definition

A idea the idea of giving a subject or process only the privileges it needs to complete a task. So every program and every user of the system should operate using the least set of privileges necessary to complete the job.

Example

This technique is applied to separate code into parts on which least privilege can be applied. A sandbox process in browser or PDF reader as I’ve learnt is an example. So our system will be protected from the abnormal behaviors(side-effects) of browsing.

Another example is the network time daemon. It (ntpd) runs with root privileges because it needs to modify the current time. If an attacker gets control of ntpd, it will be more difficult to exploit the program. Because the ntpd is needs only one capability, CAP\_SYS\_TIME.

3. Fail-safe defaults

Definition

Base access decisions on permission rather than exclusion

A mechanism that does not compromise security even if it fails.

Example

A firewall is an example.

If a packet does not match a specified set of rules, it must not be allowed. Administrators can check if there are any corner cases that allow malicious packets and pass through the firewall. Administrators should configure the firewall to only allow the packets it considers allowed, and deny all others.

4. Do you consider the following requirement a quality attribute or functional requirement? In a sentence or two explain why.

*As an application user, I want to securely reset my password, so that I can change my password for security reasons or if I forgot my previous password”*

Answer:

Changing password itself is considered as functional REQ, but this case is a quality attribute because the user (who's forgotten the password or wants to reset it) want to change in a secure way. Furthermore; in this process, using secure network and accessing secure DB storage must also be considered, so it is a QA attribute that entirely affects the login (id, password) system.

Thank you for your time