Task 1.1P Apply software security principles to answer the following questions.

1. A software company adds security features (eg. input validation, encryption etc.) only when they experience cyber-attacks and not by default. Explain briefly which software security principle is being violated and how to mitigate it.
    - Reactive Security Security that is encryption and input validation is done using vendrs and software companies are only alerted in the event that a certain security problem has occurred. The "Secure by Design" principle whereby software security, should be incorporated right from the design phase. growth. Such attacks can be targeted to the earlier versions, which are especially vulnerable after the updates with higheramientos Security measures have been taking due to an attack. That can be lessened if Security issues begin to be addressed right from the requirement gathering phase and remains such right up to the This calls the phase of software development called maintenance. Documentation, threat modeling, and compliance with coding best practices as well as engaging routine security audits should also help. Consequently, possible risks are also handled before the software solution gets to the deployment stage so that potential risks can be effectively managed.

2. An application uses credentials stored in database to authenticate users. But in the event of database failure the authentication gets bypassed and unauthorized users can get access to the application. Explain briefly which software security principle is violated and the method to mitigate it.
    - How to Avoid Authentication Due to an error in the database Here what happens is, instead of authentication being averted, those users who shouldn't have the privilege to gain access of the system, can do so—by an error in the database. The "Fail Safe" principle that dictates that a system within a design needs to fail safely is violated here. In this case, it becomes insecure since access is given without the need for identification of the user. Thus, it remains necessary to add redundancy, for example, the appearance of backup identity providers or cryptographic tokens necessary as the positive impact of tangible authentication is weakening at their presence. This will also contribute to neutralizing undesired intrusion into the system even if some of the components of the system are sick.

3. In the software development process, you set guidelines that a developer should not be able to mark the software ready for production/release without the approval of quality assurance team. What type of security principle are you applying here?
    - The other one is Quality Control, which employs an independent approval procedure referred to as "separation of duties" or "checks and balances" in the event a program requires QA approval before the release of the software. Since no one has the ability to release, this reduces the chances that either faults or Trojan code cannot be identified. This is because; to ensure quality is maintained and software is protected from unauthorized access, quality approval from vendor QA has to be achieved alongside accountability.

4. An "Application A" makes an API call to email handling "Application B" to generate a to-do list from user flagged emails. However, the developer also provides read access to the contact list which is not required by the "Application A". Which secure coding principle must be applied by developers to reduce the security risks posed by this unused access rights.
    - Incorrect Authorization with API Request The only emails that the contact list needs are the flagged ones, but the application A receives early authorizations in this case. Among these, the first one violated is the "Least Privilege" principle which means that in order to enable the maximal usage of something, only the minimum rights should be provided. This reduces the security threats and if there are any vulnerabilities in the application, these are limited since the application was programmed to work with only the rights required by this application.

5. The mitigation strategy for CSRF attacks requires applying security measures at various levels such as input validation, validating session cookies, verifying origin headers. What kind of secure coding principle is being applied here explain briefly.
    - Several measures have been taken to reduce the effect of CSRF attack such as the use of origin header checks, session cookie check and the input check. This corresponds nicely with the basic concept known as the "Defense in Depth" which presupposes the use of several safety levels. Subsequently, layers of protection are developed, it becomes even harder for the attacker to crack down into the system and the overall risk of such attack is greatly reduced.

Reference

Jackson, M. (2024) Secure Software Development Life Cycle (SSDLC). https://blog.gitguardian.com/securing-your-sdlc-software-development-life-cycle/

Software Security: Building security in (2006). https://ieeexplore.ieee.org/document/4021964.

Schneier on Security: Data and Goliath (no date). https://www.schneier.com/books/data-and-goliath/.