COM-301 Computer Security
Exercise sheet: Computer security principles

Note: Some of these exercises could have multiple acceptable answers. The answers we provide are jut one example. If you have another answer and would like to check, use the forum or send us an email or ask us during the exercise sessions or ask for student hours.

1. A company publishes the design of its security software product in a manual that accompanies the executable software. In what ways does this satisfy the principle of open design? In what ways does it not?

   Solution:
   Satisfies the principle: The publication of a manual explaining the design is indeed open, as it is possible for everyone to understand the functionality that the software is intended to provide.

   Does not satisfy the principle: However, the software itself is not open. Thus, one cannot fully apply the Linus law: one cannot look at the code, thus one cannot find bugs in the implementation. Therefore, it cannot take advantage of the positive aspects promoted by the principle of open design.

2. A common technique for inhibiting password guessing is to disable an account after three consecutive failed login attempts. Discuss how this technique might prevent legitimate users from accessing the system. Why is this situation a violation of the principle of least common mechanism?

   Solution:
   In order to prevent legitimate users from accessing the system an adversary can attempt to log in with the legitimate user's username but incorrect credentials so that this user gets blocked.

   This attack is enabled by the fact that legitimate users and potential adversaries share the same interface (common mechanism). This allow the potential adversaries to access the login mechanism and lock out a legitimate user.

   Note: This does not mean that systems should be built with users having different authentication interface. It only means that the fact that they share a common interface opens the door to attacks.

3. Explain, in terms of the security principles, why it is not a good idea to revert the execution of a program that writes in memory regions that was not allocated for this program to reserve more memory. What is a better option than reverting in terms of security ?

   Solution:
   No, it is not a good idea because it is very difficult to guarantee that everything in the system will roll back correctly (the more difficult as the complexity of the system increases). This violates the fail-safe default principle. The safe solution is what programs already do: stop and exit with an error to inform the user that something has gone wrong, and to avoid creating more damage in the system.

4. The IC Building elevator requires card key access to go to the 4th floor. The door connecting the stairs and the 4th floor opens without a card. Which of the following security principles is missing in this design?

   (a) Fail-safe defaults
   (b) Complete mediation
   (c) Separation of privilege

   Solution:
   The missing security principle is Complete mediation since the principle states that every access to every object must be checked for authority, and in this case, the access to 4th floor via stairs is not checked.

5. You are building a system and following the Compromised Recording principle you decide to create a log to register accesses to a very sensitive file `ImportantStuff` to detect if someone non authorized has accessed the file. Are the following good or bad ideas to make sure that this log fulfills its goal (justify your answer in terms of security principles):

   (a) Store the log in the same folder as the `ImportantStuff` file.
   (b) Keep two copies of the log on the same machine as `ImportantStuff` file. to the log.
   (c) If you detect an suspicious access, stop every user from accessing `ImportantStuff`
   (d) Store in this log information about other file `LessImportantThings`.

   Solution:

   (a) Bad idea: all your eggs are in the same basket. The log should be somewhere else.
      - Separation of privileges: If an attacker gains write access to the whole folder, it would be easy for them to remove their presence from the logs. Also, file access should not be associated with log access.

- Least common mechanism: It should not be easy to both modify the file and its log.

(b) Bad idea: all your eggs are in the same basket. The log should be somewhere else.

- Separation of privileges: One machine being compromised should not compromise the whole system.

(c) Good idea: if you don't know what happened with the file, better not do anything with it (fail-safe principle).

(d) Bad idea: now logs are mixed, more principals have access to the log, complexity increases.

- Least common mechanism: Two files are now logged into the same mechanism, which contradicts the principle.