Access Control and Authentication

IMPORTANT NOTES: Study lecture materials at least 1 hour and prepare Question 1-4 prior to the tutorial session. Prepared questions will be discussed in the tutorial session.

1. In the context of access control, what is the difference between a subject and an object? What is an access right?

   A subject is an entity capable of accessing objects. Generally, the concept of subject equates with that of process. Any user or application actually gains access to an object by means of a process that represents that user or application.

   An object is anything to which access is controlled. Examples include files, portions of files, programs, and segments of memory.

   An access right describes the way in which a subject may access an object. For example, read access, write access, execute access.

2. In UNIX, both files and directories include a nine-bit access control permission mode. If care is not taken, this can create access control problems. For example, consider a file `file.txt` with permission mode `rw-r--r--` contained in a directory `dir` with permission mode `rwx-wx---`. How might the file be compromised in this case?

   Suppose that the directory `dir` and the file `file.txt` have the same owner Alice and group and that Alice intends to prevent anyone except herself from editing `file.txt`. Indeed, disregarding the administrator superuser `root`, no one besides the owner Alice of `file.txt` can change its contents, because Alice set the protection mode for `file.txt` so that only the owner Alice has write permission. However, Alice neglected setting the protection mode for the directory `dir` correctly. With the set protection mode, anyone in the owner Alice's group has write permission for `dir`, so that any such person can remove (delete) `file.txt` from `dir` and create a new version of `file.txt`, which for most purposes is the equivalent of being able to modify `file.txt`.

3. In general terms what are the three means of authenticating a user's identity?

   - Something you know (SYK): password, pin code
   - Something you have (SYH): smart-card, ATM, phone number
   - Something you are (SYA): fingerprint, iris
   - In some contexts, another form of authentication named "something you do" is defined. It refers to authentication using signature, voice recognition, pin pressing pattern (the speed and power when you press pin code into pin pad). In the lecture, we called this form of authentication *dynamic biometrics*, and considered it as another form of SYA.

4. In the context of biometric authentication, define the terms false match rate and false non-match rate, and explain the use of a threshold in relationship to these two rates.

   A false match occurs when an imposter's biometric data is declared by the system to be matched with the stored biometric data for a user.

   A false mismatch occurs when the system declares that the biometric data of a genuine user does not match the stored biometric data for that user.

   The rate refers to the probability of a false match or false mismatch.

   If the matching score of the presented value, S > T (a threshold) then a match is assumed and for S < T, a mismatch is assumed.

5. What is meant by a one-way hash function? Why is it useful for protecting passwords against attacks that expose the stored file contents of the password authentication server?

   A one-way hash function is a function f that is easy to calculate in one direction ($x \to f(x)$) and infeasible in the opposite direction ($f(x) \to x$).

   It can be used to process data into a format (hash value) that is infeasible to reverse back to the data (in theory). If the data is sufficiently unpredictable to make brute-force search attacks infeasible, then it is computationally infeasible (in practice) to derive the data from its hash value.

   Such one-way hash functions are useful for protecting passwords from being exposed if the password ('shadow') file on the authentication server is exposed by an attacker. Instead of storing each password pwd directly in the shadow file, the server instead hides the password by storing instead its hash value $y = f(\text{pwd})$ in the shadow file. To authenticate the user who logs in with a password pwd', the server computes its hash $y' = f(\text{pwd}')$ and compares y' with the stored password hash y for that user in the password shadow file. This hashing is computationally feasible, whereas due to the one-way property of f, it would be difficult for an attacker to compute the password pwd from the hash value y. To slow down brute-force search attacks, password hashing uses specially designed one-way hash functions (a.k.a. password-based key derivation functions) that are relatively slow to evaluate. An example of such a function is *bcrypt*, used for password hashing in Linux operating systems. In practice, a random salt is also hashed together with the password to slow down pre-computation attacks, and the salt is stored in the clear in the password shadow file (see lecture slides and optional exercise sheet for more details).

6. Let x to be a password that contains exactly 3 characters. The characters are chosen from a set of alphabet A={ a,b,c,d,e }. How many possible distinct x can be created in the following scenarios:

   (a) If x can contain repeated characters? e.g. aaa is a valid password

   (b) If x should not contain repeated characters? e.g. aab is not a valid password

   (a) $5^3 = 125$

   (b) $5 \times 4 \times 3 = 60$