COM-301 Computer Security
Exercise 5: Authentication

1. Why are both (user, password) required for authentication? Why can't we use just a password for authentication?

   Solution:
   Authentication is to verify the validity of a claimed identity. If there is no username, there is no claimed identity. One could have a valid password but, what permissions would be assigned?

   Note that sometimes, e.g., re-entering a password for your email client, you do not explicitly write the username, but the client already has it.

2. Servers should store hashed passwords instead of storing them as plaintexts. Would hashing the password in the client, before sending it to the server through the encrypted channel bring any security advantage? How would the server side of the authentication process change if we have hashing at the client? How would the changes affect security?

   Solution:
   It does not really add any advantage, and in fact introduces a problem with respect to only encrypting. How would we check at the server side? Two options:

   - Store the password in the clear so that when the hash comes one does: ReceivedHash = hash(StoredPassword). We are again storing cleartext passwords.
   - Store the hash at the server. Then the hash becomes the "cleartext" password. If the adversary gets hold of the password file, then she can just use those to log in as the users.

3. In class, we talked a lot about how to counter offline attacks against password cracking. Please discuss online attacks (i.e., attacks where the adversary tries to guess a password by interacting with the system). Are they easier to thwart than offline attacks? What are the typical defenses against them? What is the rationale behind these defenses?

   Solution:
   Online attacks are easier to thwart than offline attacks because one can try to reduce the capabilities of the adversary.

The most typical defenses are: (i) Limiting the number of trials that can be done, (ii) Using captchas to ensure that a human is introducing the password

Both methods, as mentioned above, have a goal of restricting the adversary: restrict the number of attempts, and avoid attacks which can be automatically performed by a machine.

Stealing a hashed password storage is an example of changing an online attack to an offline attack. If the attacker needs to connect to a remote server to check password acceptance then the server can limit the interaction, but if the attacker can check with the stolen storage locally then the server cannot limit attacker's capabilities.

4. Why is having a slow hash function that hinders offline attacks not a problem for the authentication process?

   Solution:
   It is not a problem because during authentication the slow hash function is run only once. Therefore it does not affect the usability of the system.

   In an offline attack, the adversary has to repeat the operation a large number of times. Therefore, the increase in time from a normal hash function to a slow hash function, which in one execution is not really important, becomes significant.

5. Argue whether having the verifier (Morty, in the slides) sending two challenges to increase the security of a challenge response protocol is a good idea.

   Solution:
   The challenge ensures that an adversary cannot use an old login request as long as the old challenge does not match the new one. Having two challenge requires the adversary to guess both challenges correctly. The server use a random challenge and having an accidental challenge match only depends on the size of the challenge rather than their number. Therefore, adding more challenges does not have any security impact beyond increasing the challenge size.

6. If you steal the biometric template from a database with server-side processing of biometrics, can you use it directly to fool authentication in another system with server-side processing?

   Solution:
   No, you cannot. When this server receives the template as input, it will process it as if it was a raw biometric, resulting in a new value which would make the comparison with the stored template fail.

7. If you were to implement a biometric authentication system for

   (a) Entering a military base

(b) Using your loyalty card at the supermarket

How would you select the parameters for the biometric algorithm? Justify.

Solution:

(a) The parameters should be set such that the rate of false positives (unauthorized users access the base) is very low. It is not crucial that the rate of false negatives (authorized users are denied access) is somewhat high. Given the security requirements of military resources, having to repeat the authentication is not a big deal.

(b) The parameters should be set such that the rate of false negatives (authorized users not being able to access the card) is very low, and false positives can be high. A supermarket needs fast checkout, and it is not the end of the world if sometimes the loyalty card is misused.

8. In token-based authentication, the token produces a value by applying a number of times a cryptographic function on a pre-agreed "seed" value.

(a) Can this cryptographic function be a hash function? If yes, what properties must this hash have? If no, what is the reason?

(b) Would the situation help if for each authentication the server could send a challenge?

Solution:

(a) No, it cannot be a hash function.

Go back to the slide. How the token works is as follows: At a time instant n, it creates a new value by applying a cryptographic function n times on the seed. For instance:

$n = 1 \rightarrow v1 = f(seed)$

$n = 2 \rightarrow v2 = f(f(seed))$

$n = 3 \rightarrow v3 = f(f(f(seed))) \ldots$

Every time the user authenticates, the adversary gets access to the value v sent in the network.

If the function is a hash, which does not require a key – i.e., anybody can compute it –, then given a captured v, the adversary can compute any posterior v by just applying the hash function on the observed v again and again (e.g., if I capture v3=h(h(h(seed))), I can easily compute v4=h(v3)=h(h(h(h(seed)))). Therefore, it is required that the device uses a keyed function (or uses more advanced protocols than the one explained in the class).

(b) No, the challenge is still sent on the wire and thus available to the adversary. Therefore, the adversary can still compute the chain $H(PreviousHash, Challenge)$.

9. EPFL is opening a new in-campus paying system. They task you with designing the authentication mechanism to reload the cards that hold the money. You decide that since this is money, it would be wise to have two-factor authentication. Argue whether each of the following combinations are good choices.

   (a) Ask users to present their Camipro to the loading machine and input their SCIPER.

   (b) Ask users to present their Camipro to the loading machine and input their Gastpar login

   (c) Ask users for Gastpar login/password

   (d) Ask users to present their Camipro and a code sent to their email

   Solution:

   (a) Not good. SCIPER is not something you know, it is something everyone knows, it is printed on the Card.

   (b) The Gastpar login is also something that is not only known to the user

   (c) Not good. This is only one factor.

   (d) Good. These are two factors well separated. You can only read an email if you know the password.