Sam Chen
Breesy Reyes

CECS 378 Assignment 7 - User Authentication

1. **What are the four means of authenticating a user's identity?**
   The four means of authenticating a user's identity are:
   - Something they know like a password or pin.
   - Something they possess like a key card.
   - Something they are like face or fingerprint recognition.
   - Something they do like a pattern.

2. **What are two common techniques used to protect a password file? Briefly explain them.**
   One common technique used to protect a password file is with hashed passwords and a salt value. To load a new password into the system, the user selects or is assigned a password. This password is combined with a fixed-length salt value.The password and salt serve as inputs to a hashing algorithm to produce a fixed-length hash code. The other common technique is password file access control. The hashed password portion of the file is accessible only by a privileged user so the opponent cannot read it without already knowing the password of a privileged user.

3. **List and briefly describe four common techniques for selecting or assigning passwords**
   We have user education where the user is advised to create hard and difficult passwords to provide guidelines on making password. Computer generated passwords, where characters are randomly generated combinations to make a password. This is unreliable. Reactive password checking, this means the system will try to hack into the users password, and if successful, it would deny the password then prompt the user again for another password. Complex password policy, if the user enters a password that is unfit the computer will reject it if it doesn't meet the requirements.

4. **List and briefly describe the principal physical characteristics used for biometric identification**

   The first is facial characteristics for people. The shape of facial features, eyes, nose, chin. Fingerprints, the same features for all humans on your fingers. Hand geometry identify features of the hand's shape and length width of the fingers. Retinal pattern is the veins under your eyes of a individual human. Signature handwriting, every person has a certain way of writing their name. Voice patterns, every individual has a unique tone, pitch, and speech.

5. **Assume that passwords are selected from four-character combinations of 26 alphabetic char-acters. Assume that an adversary is able to attempt passwords at a rate of one per second.**
   a. **Assuming no feedback to the adversary until each attempt has been completed, what is the expected time to discover the correct password?**

      We calculate 26^4 = 456,976 combinations possible for the password. So the expected is half this amount, which means 228,488 seconds
   b. **Assuming feedback to the adversary flagging an error as each incorrect character is entered, what is the expected time to discover the correct password?**

      The possible combinations are reduced to 26 every input, we get 26 * 4 = 104 attempts. It takes around 52 seconds to find the correct password.

6. **Assume that passwords are limited to the use of the 95 printable ASCII characters and that all passwords are 10 characters in length. Assume a password cracker with an encryption rate of 6.4 million encryptions per second. How long will it take to test exhaustively all possible passwords on a UNIX system?**

   We calculate 95^10 = 5.987e+19. So, 6.4 million. Then, divide e+19 by 6 million would be 1.52e+20 seconds required to go through all the combinations.

7. **It was stated that the inclusion of the salt in the UNIX password scheme increases the difficulty of guessing by a factor of 4096. But the salt is stored in plaintext in the same entry as the corresponding ciphertext password. Therefore, those two characters are known to the attacker and need not be guessed. Why is it asserted that the salt increases security? Explain**
   By implementing SALT, it will prevent hackers to use their hash table to crack a password. They won't be able to find the same password in their data to attack other people. People don't realize this because they use similar passwords throughout all their accounts.
8. **Using the previous question as a point of reference, wouldn't it be possible to completely thwart all password crackers by dramatically increasing the salt size to, say, 24 or 48 bits? Explain.**
   No, because SALT solves the problem of many people having similar passwords. If 100 people have the same password such as abc, and its compromised, it will not be an advantage for the hackers at all. If all the combinations are broken its blamed on them being weak instead of low SALT.

9. **Because of the known risks of the UNIX password system, the SunOS-4.0 documentation recommends that the password file be removed and replaced with a publicly readable file called /etc/publickey. An entry in the file for user A consists of a user's identifier ID A, the user's public key, PU a, and the corresponding private keyPRa. This private key is encrypted using DES with a key derived from the user's login passwordPa. WhenAlogs in, the system decrypts E(Pa,PRa) to obtain PR**
   a. **The system then verifies that Pa was correctly supplied. How?**
      Pa would be the inverse of PRA which makes it checkable. First take a portion of the key, then encrypt it with PUa and finally use Pa to decrypt it. It will be correct if its matching.
   b. **How can an opponent attack this system?**
      When the Pa is being checked during the encryption/decryption phase.

10. Consider the Bloom filter discussed in Section 3.3 in your textbook (Password-Based Authen-tication). Define k = number of hash functions; N = number of bits in hash table; and D = number of words in dictionary.

a. Show that the expected number of bits in the hash table that are equal to zero is expressed as φ= (1−k/N)^D

If there is only one hash function (k = 1), which produces one of N possible hash values, and there is only one word in the dictionary, then the probability that an arbitrary bit bi is set to 1 is just 1/N. If there are k hash functions, let us assume for simplicity that they produce k distinct hash functions for a given word. This assumption only introduces a small margin of error. Then, the probability that an arbitrary bit bi is set to 1 is k/N. Therefore, the probability that bi is equal to 0 is 1 – k/N. The probability that a bit is left unset after D dictionary words are processed is just the probability that each of the D transformations set other bits: Pr[bi = 0] = (1−k/N)^D This can also be interpreted as the expected fraction of bits that are equal to 0.

b. Show that the probability that an input word, not in the dictionary, will be falsely accepted as being in the dictionary is P= (1−φ)^k

A word not in the dictionary will be falsely accepted if all k bits tested are equal to 1. Now, from part (a), we can say that the expected fraction of bits in the hash table that are equal to one is 1 – φ. The probability that a random word will be mapped by a single hash function onto a bit that is already set is the probability that the bit generated by the hash function is in the set of bits equal to one, which is just 1 – φ. Therefore, the probability that the k hash functions applied to the word will produce k bits all of which are in the set of bits equal to one is (1 – φ)^k

c. Show that the preceding expression can be approximated as P≈(1−e^(−kD/N))^k

We use the approximation (1 – x) ≈ e^-x.