IK2206 – Internet Security and Privacy
Chapter 10 – Authentication of People

10 Overview

- The term "high-quality key" is introduced. It could be considered as a kind of cryptographic key infeasible to break by exhaustive search.
- What three main techniques can be used to verify your identity?
  - What you know: passwords
  - What you have: physical keys or ATM cards
  - What you are: voice recognition or fingerprint

10.1 Passwords

- How may an attacker gain access to your password?

  - "Shoulder surfing"
  - Locally stored password files
  - Social engineering
  - Different kind of attacks: Offline attacks (directory attacks), brute force attacks

10.2 On-Line Password Guessing

- This section contains many examples of bad password selection. What methods can you use to make your passwords both secure and easy to remember?

  Usually the best combination of memorability and difficulty of guessing is a "pass-phrase" with intentional misspelling or punctuation and odd capitalization, like GoneFi$hing

- How can one defeat on-line attacks? Can you find any drawbacks with these methods? Are there any potential denial-of-service attacks?

  To keep track of the number of consecutive incorrect passwords for an account and when the number exceeds a threshold, lock the account and refuse access, even with a correct password.

  The downside is that with the aid of a computer it is possible to guess five bad passwords against all the accounts on a system and lock the all up.

- What is an "audit log"?

  ? The program that lets users set passwords should check for easy-to-guess passwords and disallow them. It might, for example, run them through a spell-checking dictionary and reject them if they are spelled correctly.

10.3 Off-Line Password Guessing

- Why would a passwords subject to off-line attacks need to be stronger than passwords subject to on-line attacks? (see also 10.4)

  The attacker can perform password guessing without anyone knowing it. To keep track if the number of consecutive incorrect passwords has no influence on a online-attack.

- Although passwords in password files may be hashed for improved security, why is it still a bad idea to make the password file world-readable?

  It is possible to guess a password and verify whether you got it right by hashing it and comparing to the stored value.

- With hashed passwords, what happens if Alice forgets her password? Can the system admin look it up and inform her?

  The user must choose a new password and the system administrator must install it.

- What is the difference between a "dictionary attack" and a "brute-force attack"?

  A dictionary attack uses a targeted technique of successively trying all the words in an exhaustive list called a dictionary. In contrast with a brute force attack, where a large proportion key space is searched systematically, a dictionary attack tries only those possibilities which are most likely to succeed, typically derived from a list of words for example a dictionary

- What is a salt and why would it be effective against a dictionary attack?

  When a user chooses a password, the system chooses a random number, known as salt. It stores both the salt and a hash of the combination of the salt and the password. During the authentication, the system computes the hash of the combination of the stored salt and the supplied password, and checks the computes hash against the stored hash.

  The salt makes it impossible to perform a single cryptographic hash operation and see whether a password is valid.

- Would the authentication message exchange differ if a salt is used?

  ?? I think there is no difference.

- Would users have to remember the salt value?

  No, users don’t have to remember the salt value. The salt is stored, together with the hash of the combination (salt, password), in the database.

10.4 How Big Should a Secret Be?

- How long sequence of random bits should your password correspond to according to the "rule of thumb" given in the book. Note, we believe the length of the sequence be related to

the strength of other security protocols in your system (such as session key length, hash output size etc.) rather than giving a "one size fits all" number.

A secret needs about 64 bits of randomness, since it is considered computationally infeasible to search 2^64 possibilities.

- Consider how long passwords are you able (and willing) to memorize?

- What is likely to happen if users are forced memorize long random sequences of characters, e.g., a string eleven character long?

  The user writes the password on a sticker attached to the device.

- Is it really feasible to achieve a password based authentication system which can handle off-line attacks?

  A person is not willing to memorize and type a secret which is as good as a 64-bit random number, and therefore passwords will be open to off-line password-guessing attacks.

10.5 Eavesdropping

- Eavesdropping passwords sent in clear-text is easy if you have access the data network between Alice and Bob. But eavesdropping data links is not the only threat; if an attacker is able to watch a person entering their password (cameras, keyboard loggers, ...) it won't matter how secure your network protocol is.
  - How should one address this?
  - Special gadgets generating one-time passwords?

    The user and the system have a list of valid passwords, but each one in only valid once.

10.6 Passwords and Careless Users (BRIEFLY)

10.6.1 Using a Password in Multiple Places

- What are the pros and cons of using the same password at multiple places? (In later chapters you can read about "single sign-on" services such as Kerberos, which takes a different approach to the problem).

  Pros:
  - User won't start to write down passwords

  Cons:
  -

10.6.2 Requiring Frequent Password Changes

- What are the pros and cons of enforcing frequent password changes?

  Pros:

  - If someone does learn your password it will only be useful until it next changes.

  Cons:

  - Users are more likely to write passwords down
  - Users like to choose weak passwords -> observable and guessable passwords
  - Users change usually only a part of a password.

- Why is it necessary to educate users about the importance of security?

  The inconvenience won't be worth the trouble of the restrictions. (Restrictions: don't allow similar passwords, keep track of the last time password, keep track of the last time the password was changed and does not allow another password change for some number of days)

10.6.3 A Login Trojan Horse to Capture Passwords

- Keyboard loggers could achieve the same goal as login trojans.

10.6.4 Non-Login Use of Passwords

10.7 Initial Password Distribution (BRIEFLY)

- Less relevant for this course, but an amuzing story! What do you think of the way security was administrated in this example?

10.8 Authentication Tokens

- What's the difference between a smart card as opposed to simpler authentication tokens such as credit cards with magnetic strips?

  Simple authentication tokens offer little or no protection against communications eavesdropping. Whatever information is sent "over the wire" can be collected just like a password and replayed later.

  Smart card is a device about the size of a credit card but with an embedded CPU and memory. Smart card is a better form of authentication tokens.

- Smart cards are divided into into 3 categories: "PIN protected memory card", "Cryptographic challenge/response cards", and "Cryptographic calculators"/"Readerless smart cards".
  - What are the differences?

  - Does the secret ever leave a smart card?
  - Is a special smart card reader required?

10.9 Physical Access (BRIEFLY)

10.10 Biometrics

- What are the pros and cons of using biometric devices for authentication?

  + You can't "loan out" nor can anything be stolen
  + Not anymore necessary for a human to memorize a secret
  - Not all of them are secure
  - Difficult to revoke

- Can you list some types of biometric devices? How do they differ in accuracy?
  - Fingerprints -> not secure
  - Iris scanner
  - Voiceprints
  - Handprint readers -> not so accurate