Tutorial 9 Solutions

CS5285

Gerhard Hancke

Question 1

- (a) Suppose a 10-character password is chosen where each character can be one of the letters 'A' to 'Z', 'a' to 'z' or '0' to '9'. Determine the size of this password space.

- Each letter can be chosen among 2 times 26 + 10 = 62 characters, so the password space is of size 62^{10}.

Question 1

- (b) What is the strength of such a password in bits?
- $\log_2\{62^{10}\} \approx 60$ bits

Question 1

- (c) Compared to the suggested strength for symmetric key encryption, is this strong?

- No, suggested key strength for symmetric key encryption is at least 80 bits (128 bits recommended).

Question 1

- (d) In the real world, if you asked people to create and use passwords as described above, would those passwords actually be as strong as you calculated?
- No, passwords chosen by users will never be really random and the actual strength of the passwords will therefore be less than 60 bits.

Question 1

- (e) So how big a dictionary will an attacker need to build? How much effort is needed?
- It is not a key search problem, the dictionary contains all possible passwords
- So dictionary size is 62^{10}.

Question 1

- (e) How big would the dictionary be if we add a salt value (16-bit integer)
- Dictionary needs all combination of salt/pwd
- So dictionary size $62^{10} * 2^{16}$

Question 1

- (f) Sometimes terminology differs a bit - what are the possible differences between a dictionary, rainbow table and brute force attacks on passwords?
- **Dictionary**: Pre-compute hash of all possible password combinations (also could be called a look-up table, sometime refers to list of only known words/common passwords)
- **Brute force**: Try to search for all entries in file (cover all password combinations)
- **Rainbow table**: More efficient than Dictionary in terms of storage. You store hash chains (start and end point). If you get a hash X , keep rehashing it until you match an endpoint. Recreate the chain from the start point. The password in the chain entry preceding X.

Question 2

- Normally we will store a password record as (y, s) where y = h(password, s) and s is a salt. Determine which of the following two alternative methods for calculating y is insecure.

- [1] y = E(password, s) where E is a block cipher and y is the encryption of password,s using the password as the key.
  - Secure if the attacker cannot build a dictionary of size

    (size of s) x (size of password)
- [2] y = E(s, password) where E is a block cipher and y is the encryption of password,s using s as the key.
  - s is stored in plaintext in the password file (y,s) so we can just decrypt y.
  - Not secure!

Question 3

Authentication can be based on 1) something you know, 2) something you have or 3) something you are. In the following situations, which one(s) of the three categories are used?

- You enter the Run Run Shaw library by putting your student card on the reader.
- You log into online banking using your username, a password and a hardware token which generates a number when you press a button.
- You travel and go through immigration. First you hand your passport to the officer, then the officer asks you to place your hand on a fingerprint reader.
- You enter going through immigration using an automated system. First you insert your ID card or passport and enter a PIN, then place your thumb on a fingerprint reader.

Question 3

In each you use the following:

- Something you have (your student card)
- Something you know (password), something you have (hardware token).
- Something you have (passport), something you are (fingerprint)
- Something you have (passport/card), something you know (PIN) and something you are (fingerprint).

Question 4

Phishing has some technical complexity but can be argued to be an attack based more on social engineering. Educated users as to the risks of phishing is important but can we also implement technical authentication measure to mitigate the impact of phishing compromise?

- Phishing mostly target passwords, which is the "something user knows" authentication factor. We can therefore protect a user more if we use a 2nd factor, like "something a user has".
- At the same time we could use one-time login information in combination with something user has, e.g. sending a one time use PIN to user mobile, having OTP generators,etc..