Exercises, set 2

Computer Security

- You do not have to hand this in. You can completely ignore these exercises if you wish.
- These exercises are given for you to check your knowledge during the course.
- These exercises are based on material in Chapter 4-8. Some exercises may require you to look for information outside the textbook as well.
- Exam questions can be based on some or none of the exercises.
- The teachers are happy to help you out if you run into problems.

Exercise 2.1

Alice can read and write to the file x, read the file y and execute the file z. Bob can read x, read and write to y and has no access to z.
a) Write access control lists for this situation.
b) Write capability lists for this situation.
c) What is the difference between access control lists and capability lists in terms of revoking all access rights to a specific file and revoking all access rights for a specific person?

Exercise 2.2

Assuming that passwords have length 6 and all the (English) alphanumerical characters, upper and lower case, can be used in their construction. How long will a brute force attack take on average if
a) it takes one tenth of a second to check a password?
b) it takes a microsecond to check a password?
c) Compare your answers to the case when the passwords have length 8.

Exercise 2.3

Construct the lattice of security labels for the security levels public, confidential, strictly confidential, and for the categories ADMIN, LECTURERS, and STUDENTS. Which objects are visible to a subject with security label (confidential,STUDENTS) in a need-to-know policy? How many labels can be constructed from n security levels and m categories? For illustration, consider the values n=16 and m=64.

Exercise 2.4

You are given a security policy that uses the lattice of compartments as security labels. Access is granted only when the subject's label is a subset of the objects label.
a) With the categories ADMIN, LECTURERS and STUDENTS, which objects can be accesses by a subject with label STUDENTS?
b) Why is a subject with label ADMIN, STUDENTS more constrained than a subject with label STUDENTS?

c) Interpret the roles of the labels $\emptyset$ and ADMIN,LECTURERS, STUDENTS in this policy.

Note the difference between this exercise and exercise 2.3.

Exercise 2.5

a) In Unix, why are shadow password files used and how do they differ from the normal password files?

b) What is a salt and why is it used when a password is encrypted?

Exercise 2.6

Find the error in the following reasoning:

In a time-memory tradeoff attack using a rainbow table, there is only one table with mt rows to search, instead of t tables with m rows each as used in Hellman's original attack. Searching for the presence of a certain endpoint can be done with a logarithmic search provided that the endpoints are sorted. The number of endpoints are in total mt for both attacks. In the rainbow case, approximately $\log(mt)$ lookups are made, and in the Hellman case approximately $t \cdot \log(m)$ lookups are made for each candidate endpoint. Since the length of the chains are t in both cases the online time is about $t \cdot \log(mt)$ for the rainbow case, and $t \cdot t \cdot \log(m)$ for the Hellman case. Thus, we conclude that with the same amount of memory, the rainbow table needs approximately a factor t fewer table lookups and is therefore a factor t faster.

Exercise 2.7

Explain why a random salt protects against the time-memory tradeoff (or rainbow) attack.

Exercise 2.8

Explain the access permissions given to the program "run" when it is executed. To what extent are the permissions useful?

```
-rwsr--r-- 1 root admins 504 2010-02-01 05:43 run
```

Exercise 2.9

What effect does the command `umask 027` have on newly created files and programs?

Exercise 2.10

Assume that the file `Bill.txt` is owned by user Alice and the group Students. The file permissions in Unix gives the owner read access and the group write access (420). In windows there is one ACE giving Alice read access and one ACE giving the group Students write access. How is this seemingly similar situation handled in Unix and Windows respectively.

Exercise 2.11

When a user starts a process, the permissions of the user have to be transferred to the process which will be acting on behalf of the user. How is this implemented in Unix/Linux and Windows respectively?