Homework 2 Solutions
Due: 19th December(Friday), in class

1. Consider a computer system with three users: Alice, Bob, and Cyndy. Alice owns the file *alicerc*, and Bob and Cyndy can read it. Cyndy can read and write the file *bobrc*, which Bob owns, but Alice can only read it. Only Cyndy can read and write the file *cyndyrc*, which she owns. Assume that the owner of each of these files can execute it.

a. Create the corresponding access matrix.

b. Cyndy gives Alice permission to read *cyndyrc*, and Alice removes Bob's ability to read *alicerc*. Show the new access control matrix.

Answers:

a. and b.

	alicerc	*bobrc*	*cyndyrc*
Alice	ox	r	
Bob	r	ox	
Cyndy	r	rw	orwx

	alicerc	*bobrc*	*cyndyrc*
Alice	ox	r	r
Bob		ox	
Cyndy	r	rw	orwx

2. In the in-class exercise on day 6, we considered the set of rights *{read, write, execute, append, list, modify, own}*. Using the syntax of the example in the slides (for *create-file),* you wrote a *command* ***delete_all_rights(p,q,s).*** This command causes *p* to delete all rights the subject *q* has over an object *s*. Here's an implementation of the command:

```
command delete_all_rights(p, q, s)
```

```
      delete r in A[q, s];
      delete w in A[q, s];
      delete x in A[q, s];
      delete a in A[q, s];
      delete l in A[q, s];
      delete m in A[q, s];
      delete o in A[q, s];
end;
```

a. Modify your command so that the deletion can occur only if p has modify rights over s.
   Answer: Here, we must condition the command on the presence of rights that *p* has over *s*:

```
command delete_all_rights(p, q, s)
      if m ∈ A[p, s] then
            delete r in A[q, s];
            delete w in A[q, s];
            delete x in A[q, s];
            delete a in A[q, s];
            delete l in A[q, s];
            delete m in A[q, s];
            delete o in A[q, s];
end;
```

b. Modify your command so that deletion can occur only if p has modify rights over s and q does not have own rights over s.
   Answer: This one is trickier. We cannot test for the *absense* of rights directly, so we build a surrogate object *z*. The idea is that *A*[*q*,*z*] will contain the right *o* if *q* does not have *o* rights over *s*, and will contain the right *m* if *q* has *o* rights over *s*. So, we need some auxiliary commands:

```
command make_aux_object(p, q, z)
      create object z;
      enter o in A[q,z];
end;
command fixup_aux_object(p, q, s, z)
      if o ∈ A[q, s] then
            delete o in A[q, z];
            enter m in A[q,z];
end;
```

Now we write the command to delete the rights if *p* has *m* rights over *s* and *q* does *not* have *o* rights over *s*. The last econdition is logically equivalent to *q* having *o* rights over *z*:

```
command prelim_delete_all_rights(p, q, s, z)
      if m ∈ A[p, s] and o ∈ A[q,z] then
            delete r in A[q, s];
            delete w in A[q, s];
```

```
            delete x in A[q, s];
            delete a in A[q, s];
            delete l in A[q, s];
            delete m in A[q, s];
            delete o in A[q, s];
end;
```

Finally, we create the actual delete command:

```
command delete_all_rights(p, q, s, z)
      make_aux_object(p, q, z);
      fixup_aux_object(p, q, s, z);
      prelim_delete_all_rights(p, q, s, z);
      destroy object z;
end;
```

3. This exercise asks you to consider the consequences of not applying the principle of attenuation of privilege to a computer system.
    a. What are the consequences of not applying the principle at all? In particular, what is the maximal set of rights that subjects within the system can acquire (possibly with the cooperation of other subjects)?
    b. Suppose attenuation of privilege applies only to access rights such as read and write, but not to rights such as own and grant rights. Would this ameliorate the situation discussed in part (a)? Why or why not?

Answer:

a. If any subject can give any rights to any other subject, then the maximal set of rights subjects within the system can acquire is all rights over all other entities in the system. If some rights can be given under specific conditions, the maximal set of rights depends upon the nature of the conditions themselves.
b. Applying attenuation of privileges to access rights only would not ameliorate the above situation, because the control rights only control the spread of rights throughout the system. If the owner can grant rights it does not have, it can grant the requisite access rights as desired, thus enabling information to flow about the system.

4. In the paper titled "Access Control: Principles and Practice," by R.S. Sandhu and P. Samarati, (see Sandu_Samarati_Access_Control.pdf on the class website under Homework) or http://www.profsandhu.com/journals/commun/i94ac(org).pdf the authors discuss how access control matrices are actually implemented. Answer the following questions after reading the paper:
    a. For the following ACM, determine (i) the Access Control list (ii) the capability list and (iii) the table of authorization relations.

	matt	*holly*	*heidi*
recipes	read	read, write	read, write
overpass	read	read, write	read, write
.shellrct	read	read	read
oven.dev			temp_ctl

    b. In about 500 words or less, differentiate between discretionary policies, mandatory policies, and role-based policies.