CIS 4360 Introduction to Computer Security

Home Assignment 5, Fall 2011 — with answers

Due: Tue 9:30pm, 10/11/2011

This concerns access control structures.
Examples taken from: Dieter Gollmann, Computer Security, John Wiley.

1. You are given two bits to capture access operations on a directory.

   - How would you use four operations available to you?
     Answer. *An obvious solution is to assign to each operation one of the four values the two bits can take. So read would be 00, write (create or delete) 01, append 01 and execute 11.*
   - How would you control the creation and deletion of files?
     Answer. You could use different values of the pair of bits. For example, create can be written 00, and delete can be grouped with write.
   - How would you implement the concept of "hidden" files with these access operations? (Hidden files are only visible to authorized subjects.)
     Answer. A solution for hidden files would be to set a (new) bit in a file descriptor and have the directory operations check that the caller of the file is authorized; in turn, this requires to define what "is authorized" means and one could, for example, state that a file should be hidden from everyone but its owner.

2. Consider a system with the four access operations read, write, grant, and revoke. You can use grant not only to give other subjects read and write access, you can also grant them the right to 'grant' access to the objects you own. Which data structure and algorithm would you use to implement the grant and revoke operation so that you can revoke all access to an object you own? (*Hint.* Delegation trees?)
   Answer: A 'delegation tree' is an obvious solution. Each user keeps track of the user rights that have been granted; to maintain the tree structure, one has to make sure that no delegation cycles occur. When a user revokes a right, this right has to be removed from all users in the relevant subtree.

3. Explain why the partial ordering of abilities as defined in Section 5.8.2 of the 3rd Edition (or Section 4.7.1 of the 2nd Edition) of the textbook does not constitute a lattice. Try to convert the partial ordering into a lattice by adding any further elements you need to the set of abilities.
   (*Hint.* In a lattice any two elements must have an upper bound. This fails here.)
   Answer. In general, two elements need not have an upper bound, take for example the abilities: .1 and .2.
   To get upper bounds, we may introduce a new symbol, e.g. the wildcard symbol $*$, and declare that $A \leq *$ holds for each ability A; so $.1 \leq *$ and $.2 \leq *$.

4. Construct the lattice of security labels for the security levels 'public', 'confidential', and 'strictly confidential', and for the categories ADMIN, LECTURERS, and STUDENTS.
   Answer. There are three security labels and 8 category subsets (of the set {ADMIN, LECTURERS, STUDENTS}. So the lattice has $24 = 3 * 8$ elements, 8 at each security level.

   - Which objects are visible to a subject with security label (confidential,{ STUDENTS }) in a need-to-know policy?
     Answer. A subject with security label (confidential,{ STUDENTS }) can access (e.g., read) objects with labels $(\text{public}, \{\})$, $(\text{public}, \{\text{STUDENTS}\})$, $(\text{confidential}, \{\})$, $(\text{confidential}, \{\text{STUDENTS}\})$.
   - How many labels can be constructed from n security levels and m categories? For illustration, consider the values $n = 16$ and $m = 64$.
     Answer. For the values given we get $16 * 2^{64} = 2^4 * 2^{64} = 2^{68}$ labels. In general, the number of labels is $n2^m$.

Mike Burmester