CSE 565 Fall 2018
Homework 4
Due on October 16, 2018, at 9:30am

1. (25 points) Read about and experiment with (Unix) commands `setfacl` and `getfacl`. (These commands worked for me with files created in the `/tmp` directory on timberlake, but not with files in my home directory, which are not served by the file system on timberlake.) Answer the following questions about their functionality:

   (a) What is the meaning of the mask, i.e., how does it influence effective access rights? What is the default value of the mask?

   (b) What access rights the owner, the (owner's) group, user `abc`, and others have on file `file` after executing each of the commands below. If necessary, you can assume that `abc`'s group is different from the owner's group.

      i. `setfacl -m user::rw-,user:abc:rw-,group::r--, mask:rwx,other:--- file`

      ii. `setfacl -m user::rwx,user:abc:r-x,group::r-x, mask:--x,other:--- file`

   (c) What command(s) should be executed to set special permissions that will be applied to all files created in a specific directory? You can answer this question on the example that enables user `abc` to have read and write access to all files (which will be) created in directory `dir`.

2. (35 points) This exercise is designed to help understand the effect of password rules on the password space. Suppose that a system allows a user to choose a password of length between six and eight characters, inclusive. The system also places constraints on what constitutes a valid password. For each category below, compute (1) the total number of passwords and (2) the number of passwords of the shortest length. Suppose that someone gets ahold of the stored (hashed) passwords and is able to test 10 million password guesses per second. Compute (3) how long it would take to exhaustively search the entire password space to recover all passwords with 100% probability and (4) how long it would take to exhaustively search the space of passwords of the shortest length for each category below.

   (a) Password characters may be any ASCII characters from 1 to 127, inclusive.

   (b) Password characters must be digits.

   (c) Password characters may be any alphanumeric characters ("A" through "Z," "a" through "z," and "0" through "9").

   (d) Password characters maybe any alphanumeric characters, but the first character must be a letter and there must be at least one digit in a password.

3. (30 points) This exercise is about effectiveness of using salts for stored passwords. Some time ago, the designers of the UNIX password algorithm used a 12-bit salt to make password guessing attacks less effective. Passwords were chosen to be up to 8 characters long. Suppose that a particular system has 2^{24} users and each user is assigned a salt chosen uniformly at random. The questions below ask about the complexity of mounting dictionary attacks. Assume that the time required to perform a dictionary attack against a single password without the use of salt is treated as a time unit. Also assume that the use of salt doesn't slow down the hashing algorithm, and the password hashes and salts are accessible.

(a) What is the expected time to find (or try to find) all users' passwords using a dictionary attack?

(b) Assume that eight more characters were added to the password (which makes passwords' length to be up to 16 characters) and the hashing algorithm takes that into account without noticeable performance degradation. What would be the expected time to mount the dictionary attack on each user's password? What are the units for measuring time in this case?

(c) Assume that passwords are eight characters long, but the salt length is increased to 24 bits. What would be the expected time to find all users' passwords using a dictionary attack?

Because the salts are chosen at random, what values they take will differ from one execution to another. For that reason, you can provide approximate numbers.