CSE 565 - Homework 4

Amlan Gupta (#50288686)

October 2018

1 Question

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?

The mode mask contains the permission bits that should not be set on a newly created file, hence it is the logical complement of the permission bits set on a newly created file. If some bit in the mask is set to 1, the corresponding permission for the newly created file will be disabled. Hence the mask acts as a filter to strip away permission bits and helps with setting default access to files.

Octal	Binary	Meaning
0	000	no permissions
1	001	execute only
2	010	write only
3	011	write and execute
4	100	read only
5	101	read and execute
6	110	read and write
7	111	read, write and execute

In UNIX, the default file creation value is 666. As understood from the above table 6 means the permission for read and write access. Three 6s are permission indicators for User, Group and Others. Using mask we can choose to retain or block some of the default permissions from being applied on the file. If the default permission is 666 and we set mask value as 022. The final permission given to the file would be 644.

The default value of mask is 022.

(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

After executing the above command the permission table will be:

Owner	User abc	Owner's Group	Others
- - -	- - -	- - -	- - -

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

After executing the above command the permission table will be:

Owner	User abc	Owner's Group	Others
rw -	r - -	r - -	- - -

(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.

setfacl -R -m user:abc:rw- dir
By using -R argument the operation can be applied to all files and directories recursively.

2 Question

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.

1. The below table demonstrates total number of passwords for each length, and total number of passwords following the conditions a,b,c, and d.

Conditions	Length 6	Length 7	Length 8	Total
a	4195872914689	532875860165503	67675234241018900	68212305974099100
b	1000000	10000000	100000000	111000000
c	56800235584	3521614606208	218340105584896	221918520426688
d	27868297600	1925540547840	129664230991360	131617639836800

For example, the number of passwords of length 6 consisting of any ASCII characters from 1 to 127 would be 127^6.
If the passwords are only generated with digits, total number could be 10^6.
For passwords containing alphanumeric characters, the total combinations could be 62^6.
For passwords that starts with a letter, there could be 52 different choices for first character. If we subtract passwords which have no digits in last 5 places (52^5) from rest of the options(62^5), we will have have 52 x (62^5 - 52^5).

The same logic can be followed for passwords of other lengths.

The time required to test all password available in password is listed below:

Conditions	Total	Time (Sec)
a	68212305974099100	6821230597
b	111000000	11.1
c	221918520426688	22191852.04
d	131617639836800	13161763.98

The time required to search the space of password of the shortest length for each category will be:

Conditions	Length 6	Time (Sec)
a	4195872914689	419587.2915
b	1000000	0.1
c	56800235584	5680.023558
d	27868297600	2786.82976

3 Question

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?

Let, we have a dictionary available with passwords-hash pair of length up to 8 characters. If the time required to perform a dictionary attack against a single password without the use of salt is treated as unit time, the required time to query passwords for all users will be 2^{24} units. With a 12-bit salt incorporated to the hashing function, 2^{12} attempts will be required for a single password. So total time needed will be $2^{24} * 2^{12} = 2^{36}$

(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?

If eight characters are added with the password, the dictionary should contains hashed versions of password of length up to 16. The expected time to mount the dictionary attack on each user's password will be same since there are no changes in salt length.

As the number of password in the dictionary table will increase significantly, there should be some changes in unit, as with that big a table time to query for a password match should take longer time.

(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?

With a salt of 24 bit, the time required to find all users' password will be increased to $2^{24} * 2^{24} = 2^{48}$ units.