Homework 04

Md Moniruzzaman Monir, # 50291708

October 16, 2018

Problem 01

Solution:

a) **Mask** limits the effective access rights granted to all **groups** and to **named users**. The file owner and others permissions are not affected by the effective rights mask. For example, suppose a user 'abc' has read and write permission (rw–) for a file, and the mask value of that file is 'rw–'. This mask value represents the upper limit of access rights. Now the system will check that upper limit is 'rw–' and user 'abc' wants 'rw–'. So the user will get the access rights. But if I change the mask value to 'r––', then the system will see that the upper limit of rights is 'r––' and user 'abc' wants 'rw–' which can't be given. So the system will assign the effective access right as 'r––' :

```
user :abc: rw–          # effective: r––
mask :: r––             // file's mask value
```

For each file, **getfacl** displays the file name, owner, the group, and the Access Control List (ACL). If a directory has a default ACL, getfacl also displays the default ACL. Non-directories cannot have default ACLs. If getfacl is used on a file system that does not support ACLs, getfacl displays the access permissions defined by the traditional file mode permission bits.

The output format of getfacl is as follows :

```
1:  file: somedir/ 2:  owner: abc
3:  group: staff
4:  flags: -s-
5: user::rwx
6: user:xyz:rwx # effective:r-x
7: group::rwx # effective:r-x
8: group:cool:r-x
9: mask::r-x
10: other::r-x
11: default:user::rwx
12: default:user:joe:rwx # effective:r-x
13: default:group::r-x
14: default:mask::r-x
15: default:other::—
```

Line 14 shows the default value of 'mask'. Lines 11–15 display the default ACL associated with this directory. Directories may have a default ACL. Regular files never have a default ACL.
The default value of the **mask** depends on the value of the **umask** of a system. When we set default ACL permission along with masking then there should not be any effect of masking bit. Mode of file and directory gets preference at kernel level. While we create any file, kernel passes mode 0666 to its **open** system call and it passes mode 0777 to **mkdir** system call during creating directory. Then based on the value of umask it sets permission of the file and directory. Effective permission is mapped to masking permission while we pass extended attributes to setfacl. In timberlake server, the umask value is 022. Any new files will, by default, have the permissions 644. Likewise, any new directories will, by default, be created with the permissions 755.

The default value of mask for a file can be **rw–** and for directory can be **rwx**.

b) i) After running the command :

   setfacl -m user::rw−, user:abc:rw−, group::r−−, mask:rwx, other:− − − **file**

   The owner's access rights on **file** : 'rw−'
   The (owners) group's access rights on **file** : 'r−−'
   Access rights of user 'abc' on **file** : 'rw−'
   Others access rights on **file** : '− − −'

   ii) After running the command :

   setfacl -m user::rwx, user:abc:r−x, group::r−x, mask:−−x, other:− − − **file**

   The owner's access rights on **file** : 'rwx'
   The (owners) group's access rights on **file** : '−−x'
   Access rights of user 'abc' on **file** : '−−x'
   Others access rights on **file** : '− − −'

c) setfacl -Rdm user:abc:rw− dir
   setfacl -Rm user:abc:rw− dir

   R is recursive, which means everything under that directory will have the rule applied to it. **d** is default, which means for all future items created under that directory, have these rules apply by default. **m** is needed to modify rules. The first command is for new items and the second command is for existing items under the folder.

Problem 02

Solution:

a) Total number of passwords : $127^6 + 127^7 + 127^8$
   Number of password of the shortest length : 127^6
   Time for searching entire password space with 10 million password guess per second : 6821230597 sec
   Time for shortest password : 419587.2915 sec

b) Total number of passwords : $10^6 + 10^7 + 10^8$
   Number of password of the shortest length : 10^6
   Time for searching entire password space with 10 million password guess per second : 11.1 sec
   Time for shortest password : 0.1 sec

c) Total number of passwords : $62^6 + 62^7 + 62^8$
   Number of password of the shortest length : 62^6
   Time for searching entire password space with 10 million password guess per second : 22191852.04 sec
   Time for shortest password : 5680.023558 sec

d) Total number of passwords : $(52 * 62^5 - 52^6) + (52 * 62^6 - 52^7) + (52 * 62^7 - 52^8)$
   Number of password of the shortest length : $(52 * 62^5 - 52^6)$
   Time for searching entire password space with 10 million password guess per second : 13161763.98 sec
   Time for shortest password : 2786.82976 sec

Problem 03

Solution:

a) The system has 2^{24} users, and each user is assigned a salt (12-bit) chosen uniformly at random. Total number of different salt value = 2^{12}.

Time Unit: The time required to perform a dictionary attack against a single password without the use of salt.

Without salt : **# hash computations per word in the dictionary** = 1

With salt : **# hash computations per word in the dictionary** = min (# users, # different salt value)

Expected time to find all users passwords using a dictionary attack : 2^{12} * **Time Unit**

b) As eight more characters were added to the password so the size would increase by 127^8 and so the time would increase by the same.
The time unit is $\frac{1}{127^{16}}$.

c) Expected time to find all users passwords using a dictionary attack : 2^{24} * **Time Unit**