Lab 3: Permissions
A542 - Technical Foundations of Cybersecurity
Spring 2024

1. UNIX ACL Concepts

Take a look at the following site that explains the UNIX permissions struc-
ture in more detail than in your textbook: http://www.zzee.com/solutions/
unix-permissions.shtml

(a) Consider a computer running a UNIX-like operating system that has three users, Alice, Bob, and Trudy. Alice and Bob are under the same group called friends. Below (page 2) is a directory listing of their home directories (i.e., the output of the ls -l command):

(i) Alice opened up aliceDetails.txt and made a few changes. Will she be allowed to save those changes? Explain.

- Yes, Alice will be able to save her changes to the aliceDetails.txt file. This is because she is the owner of the file and has user rw- permission.

(ii) Bob is trying to access his own file bobDetails.txt but discovers that he cannot read it. Why? How would Bob be able to gain read access to this file? What if he wants to modify the file?

- He cannot read the file because the user setting does not have permission to read it. It is set at - - -.
- In order for Bob to read and modify his file he needs to do the following on the Unix OS, “chmod u+rw bobDetails.txt”. This will allow Bob to read the file and then modify the file if he wants.

(iii) When Trudy tries to display the contents of aliceFolder under Alice’s home directory using the command ls -l, she is able to view the files and directories. Will she be able to cd into the directory? Why or why not?

- Yes, Trudy will be able to cd into the aliceFolder directory. This is because Alice has that directory set to r-x for group. The “x” allows the group to execute into the directory/applications.

(iv) Explain whether the UNIX permission structure is a column approach(ACL) or a row approach (CL)?

- The UNIX permission structure is a CL row approach. It is done in a 3x3 approach (user, group, other) (read, write, execute). When permissions are displayed on a UNIX OS it is shown in a single row like this,
- “-rwxrwxrwx”
- ACL would display multiple names under “user” and create many rows for each individual and columns for the names with a column to assign permissions.

(b) The following example shows the permissions of a file that is used by users
to change passwords on the system:
Notice an s in the permissions. Explain what the s means and its significance.
Why is it needed to change passwords?

```
−rwsr−xr−x 1 r o o t r o o t 41292 2009−07−31 0 9 : 5 5 / u s r / b i n / passwd
```

- When the “s” is placed on the executable file within user this means it is now SUID or setuid. This allows anyone to execute the file as the owner. It is used to change passwords because some tasks need elevated privileges in order to execute it.

(c) Under html-pub (in Alice’s home directory, see permissions above) there is
a file called index.html with the following permissions:

```
−rw−r−−r−− 2 a l i c e a l i c e 4096 2010−01−17 1 8 : 4 6 i n d e x . html
```

(i) Trudy hosts a webserver that is requested to display the contents of the
index.html file. Will the contents be displayed? Explain why or why not.

- Yes, the web server will be able to display the contents of the index.html file. This is because Alice has the permissions set to “r” for group and other. If Trudy is part of the group, then the server falls under group and can read. If Trudy is not part of the group, the server falls under other but can still read it.

(d) Explain the privileges of each user (Alice, Bob, and Trudy) with respect
to the folder Documents (under Alice).

- Alice has rwx permissions which gives her the permission to read, change/save, and execute the files/directory.
- Bob falls under group called “friends” so he has read and execute permission. If he was not part of the group he falls under other which has no permissions/privileges at all.
- Trudy falls under group called “friends” so she has read and execute permission. If he was not part of the group he falls under other which has no permissions/privileges at all.
- Bob and Trudy could both cd into the Documents directory and read the files within assuming they still fall under “friends” group and if she has them set to x.

2. Hands-on UNIX

We have prepared a UNIX computer (burrow/silo) in the lab where you experiment with
permissions on files and folders in your home directory. You need a
secured shell/terminal program (ssh) to connect to that UNIX computer.

Server name: burrow.sice.indiana.edu
Username: <your IU username>
Password: <your IU passphrase>

On a Windows computer, you can use an ssh client called PuTTY (https://putty.org). On a GNU/Linux or MacOS computer, you can use the ssh command on the terminal. To transfer files over ssh, you can use WinSCP (https://winscp.net).
On GNU/Linux or MacOS, you can simply use the scp command. Once you are connected to the server over ssh, and have a shell, you will also be using the following commands:

1. mkdir <dir name> to create a new directory named <dir name>
2. touch <file name> to create a new (empty) file named <dir name>
3. cp <source file> <destination file> to copy <source file> to <destination file>
4. mv <source file/folder> <destination file/folder> to rename or move <source file> to <destination file>
5. rm <file name> to delete a file named <file name>
6. rm -f <directory name> to delete a directory named <directory name> and all files/directories inside it.

To change ownership or group of files, you can use the chown or chgrp commands, and to change permissions, you can use the chmod command. You can read about these commands online, or you can read the manual (man) pages installed on the server. To lookup a man page, type man <command name>, for example, man chmod. A man page gets displayed using a pager program, where you can navigate using the arrow keys, and type q to quit.

(a) Ensure that your home directory and any subdirectories can never be listed using ls -l by their respective groups and others unless as specified below.

(b) Create an html-pub directory with a simple index.html webpage. Create permissions so that pages can be loaded through the webserver. The webserver should be able to browse your html-pub directory, and read the index.html file. You can test whether your page is accessible by visiting the following url: http://html.sice.indiana.edu/~username

(c) Create a directory named images within the html-pub directory with three images that can be loaded through an external browser. Note that this directory cannot be listed if you go to the following URL on your web browser, that is because the web server does not allow directory listings.
http://html.sice.indiana.edu/~username/images

However, if you type the complete URL to each image in the web browser, the image should be displayed. List these URLs in your report.
http://html.sice.indiana.edu/~username/images/image01.jpg

1. https://html.luddy.indiana.edu/~avelopez/images/image01.jpg
2. https://html.luddy.indiana.edu/~avelopez/images/image02.jpg
3. https://html.luddy.indiana.edu/~avelopez/images/image03.jpg

(d) Create a directory named class-project and create a file within the directory called project-design.txt. Make sure both the directory and file are accessible by only you and the burrow group. Ensure that anybody in the burrow group will be able to navigate your directory structure to enter this directory and access the file.

(e) Besides basic owner and group permissions, many UNIX file systems also provide custom access control lists (ACLs) on files and directories. The burrow server has ACLs enabled, which can be manipulated with the getfacl and setfacl commands. Lookup how to use these commands from their man pages.

(i) Create a file named secret.txt in your home directory. Use the chmod command to make it readable only to yourself, and no one else. Use the getfacl command to get the ACL for this file. What does it say?

- After using the getfacl command the ACL for this file retrieves as follows,

```
avelopez@silo:~$ getfacl secret.txt
# file: secret.txt
# owner: avelopez
# group: students
user::r--
group::---
other::---
```

ii Use the setfacl command to give the AI (saetukur) read access to the same file. Now run the getfacl command on the file again. What does it say?

- After adding saetukur with read access using setfacl I get the following,

```
avelopez@silo:~$ getfacl secret.txt
```

```
# file: secret.txt
# owner: avelopez
# group: students
user::r--
user:saetukur:r--
group::---
mask::r--
other::---
```