Linux Access Control Lists (ACLs)

This lab was developed for the Labtainer framework by the Naval Postgraduate School, Center for Cybersecurity and Cyber Operations under National Science Foundation Award No. 1438893.

1 Overview

This exercise explores the use of Linux ACLs to provide access control over files, with more flexibility than the access control offered by traditional UNIX file permissions. It is assumed the student has received instruction, or independent study, in access control policies and ACLs. A description of Linux ACLs can be found at `https://wiki.archlinux.org/index.php/Access_Control_Lists`

2 Lab Environmnet

This lab runs in the Labtainer framework, available at http://my.nps.edu/web/c3o/labtainers. That site includes links to a pre-built virtual machine that has Labtainers installed, however Labtainers can be run on any Linux host that supports Docker containers.

From your labtainer-student directory start the lab using:

```
labtainer acl
```

Links to this lab manual will be displayed.

3 Setup

After starting the lab, three virtual terminals will be created, each with a login prompt. Login to these as three different users:

user	password
bob	password4bob
alice	password4alice
harry	password4harry

4 Lab Tasks

In this lab, you will use the `getfacl` and `setfacl` commands to view and modify ACLs on files. Use the -h option to learn about these commands, e.g., `getfacl -h`.

4.1 Review existing file permissions

In the ”alice” terminal, cd to the /shared_data directory and list the files:

```
cd /shared_data
ls -l
```

Observe the file permissions on the accounting.txt file and the two directories. Would you expect Alice to be able to view the content of accounting.txt? Try to cat the file.

Look again at the directory listing. Note how the accounting.txt entry has a permission setting of:

```
-rw-rw----+
```

That + symbol at the end indicates that this file has an ACL in addition to the standard UNIX permissions of ”rw” for user and group. You can view the ACL permissions on this file using:

```
getfacl accounting.txt
```

Note how one of the three users has permission to modify that file? Go to that user’s virtual terminal and append to the file using:

```
echo "more stuff" >> /shared_data/accounting.txt
```

Back in the Alice terminal, try to modify that file to confirm she lacks that access.

4.2 Task 2: Set an ACL on a single file

As the Bob user, use the setfacl command to allow Alice to read the /shared_data/bob/bobstuff.txt file. Then, as the Alice user, confirm her ability to read the file. And as the Harry user, confirm he lacks permission to read the file.

4.3 Task 2: Set an default ACL for a directory

As the Alice user, we want to define a default ACL such that, whenever Alice creates a new file in the /shared_data/alice directory, that new file will be readable by Bob, but not by users other than Bob and Alice. Consider doing this systematically:

- Create a file in /shared_data/alice and check its permissions
- Set the default ACL on the alice directory to permit bob to read newly created files.
- Create another new file in /shared_data/alice and check its permissions. Are they what you expect?
- Revise your default ACL on the alice directory if necessary
- Confirm permissions on a newly created file are as desired.

4.4 Task 3: Trojan Horses

Revisit the permissons on the /shared_data/accounting.txt file. Bob is unable to read this file, but he would very much like to know its content. Bob knows Alice is a fool for ascii art, and he created the /shared_data/bob/fun script. As Bob, modify that script so that if Alice (or Harry) run the script, it will make a copy of the accounting.txt file in a manner that allows Bob to see the content. Confirm that when Bob runs this script, it does not provide him access to the data. But when it is run by Alice, then Bob gets access to the information. Note the distinction beween Bob getting access to the file, and Bob getting access to the information.

5 Submission

After finishing the lab, go to the terminal on your Linux system that was used to start the lab and type:

```
stoplab acl
```

When you stop the lab, the system will display a path to the zipped lab results on your Linux system. Provide that file to your instructor, e.g., via the Sakai site.