Secure Files/Directories using ACLs (Access Control Lists) in Linux

by Kuldeep Sharma | Published: April 22, 2014 | Last Updated: January 7, 2015

Linux Certifications - RHCSA / RHCE Certification | Ansible Automation Certification | LFCS / LFCE Certification

As a **System Admin**, our first priority will be to protect and secure data from unauthorized access. We all are aware of the permissions that we set using some helpful Linux commands like **chmod**, **chown**, **chgrp**… etc. However, these default permission sets have some limitation and sometimes may not work as per our needs. For example, we cannot set up different permission sets for different users on same directory or file. Thus, **Access Control Lists** (**ACLs**) were implemented.

Linux Access Control Lists

Let's say, you have three users, '**tecmint1**', '**tecmint2**' and '**tecmint3**'. Each having common group say '**acl**'. User '**tecmint1**' want that only '**tecmint2**' user can **read** and **access** files owned by '**tecmint1**' and no one else should have any access on that.

ACLs (**Access Control Lists**) allows us doing the same trick. These ACLs allow us to grant permissions for a **user**, **group** and any group of any users which are not in the group list of a user.

Note: As per Redhat Product Documentation, it provides ACL support for ext3 file system and NFS exported file systems.

How to Check ACL Support in Linux Systems

Before moving ahead you should have support for ACLs on current Kernel and mounted file systems.

1. Check Kernel for ACL Support

Run the following command to check ACL Support for file system and **POSIX_ACL=Y** option (if there is **N** instead of **Y**, then it means Kernel doesn't support ACL and need to be recompiled).

```
[root@linux ~]# grep -i acl /boot/config*

CONFIG_EXT4_FS_POSIX_ACL=y
CONFIG_REISERFS_FS_POSIX_ACL=y
```

```
CONFIG_FS_POSIX_ACL=y
CONFIG_GENERIC_ACL=y
CONFIG_TMPFS_POSIX_ACL=y
CONFIG_NFS_V3_ACL=y
CONFIG_NFSD_V2_ACL=y
CONFIG_NFSD_V3_ACL=y
CONFIG_NFS_ACL_SUPPORT=m
CONFIG_CIFS_ACL=y
CONFIG_9P_FS_POSIX_ACL=y
```

2. Check Required Packages

Before starting playing with ACLs make sure that you have required packages installed. Below are the required packages that needs to be installed using **yum** or **apt-get**.

```
[root@linux ~]# yum install nfs4-acl-tools acl libacl          [on RedHat bas
```

```
[tecmint@linux ~]$ sudo apt-get install nfs4-acl-tools acl     [on Debian bas
```

3. Check Mounted File System for ACLs Support

Now, check the mounted file system that whether it is mounted with ACL option or not. We can use '**mount**' command for checking the the same as shown below.

```
[root@linux ~]# mount  | grep -i root

/dev/mapper/fedora-root on / type ext4 (rw,relatime,data=ordered)
```

But in our case its not showing acl by default. So, next we have option to remount the mounted partition again using acl option. But, before moving ahead, we have another option to make sure that partition is mounted with acl option or not, because for recent system it may be integrated with default mount option.

```
[root@linux ~]# tune2fs -l /dev/mapper/fedora-root | grep acl

Default mount options:    user_xattr acl
```

```
[root@linux ~]# mount -o remount,acl /
```

Next, add the below entry to '/etc/fstab' file to make it permanent.

```
/dev/mapper/fedora-root /       ext4    defaults,acl 1 1
```

Again, remount the partition.

```
[root@linux ~]# mount -o remount  /
```

4. For NFS Server

On NFS server, if file system which is exported by NSF server supports ACL and ACLs can be read by NFS Clients, then ACLs are utilized by client System.

For disabling ACLs on NFS share, you have to add option "**no_acl**" in '**/etc/exportfs**' file on NFS Server. To disable it on NSF client side again use "**no_acl**" option during mount time.

How to Implement ACL Support in Linux Systems

There are two types of **ACLs**:

- **Access ACLs**: Access ACLs are used for granting permissions on any file or directory.
- **Default ACLs**: Default ACLs are used for granting/setting access control list on a specific directory only.

Difference between Access ACL and Default ACL:

- Default ACL can be used on directory level only.
- Any sub directory or file created within that directory will inherit the ACLs from its parent directory. On the other hand a file inherits the default ACLs as its access ACLs.
- We make use of "**–d**" for setting default ACLs and Default ACLs are optionals.

Before Setting Default ACLs

To determine the default ACLs for a specific file or directory, use the '**getfacl**' command. In the example below, the **getfacl** is used to get the default ACLs for a folder '**Music**'.

```
[root@linux ~]# getfacl Music/
```

```
# group: root
user::rwx
group::r-x
other::r-x
default:user::rwx
default:group::r-x
default:other::rw-
```

After Setting Default ACLs

To set the default ACLs for a specific file or directory, use the '**setfacl**' command. In the example below, the **setfacl** command will set a new ACLs (**read** and **execute**) on a folder '**Music**'.

```
[root@linux ~]# setfacl -m d:o:rx Music/
[root@linux ~]# getfacl Music/
# file: Music/
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
default:user::rwx
default:group::r-x
default:other::r-x
```

How to Set New ACLs

Use the '**setfacl**' command for setting or modifying on any file or directory. For example, to give **read** and **write** permissions to user '**tecmint1**'.

```
# setfacl -m u:tecmint1:rw /tecmint1/example
```

How to View ACLs

Use the '**getfacl**' command for viewing ACL on any file or directory. For example, to view ACL on '**/tecmint1/example**' use below command.

```
# owner: tecmint1
# group: tecmint1
user::rwx
user:tecmint1:rwx
user:tecmint2:r--
group::rwx
mask::rwx
other::---
```

How to Remove ACLs

For removing ACL from any file/directory, we use **x** and **b** options as shown below.

```
# setfacl -x ACL file/directory         # remove only specified ACL from file/

# setfacl -b  file/directory            #removing all ACL from file/direcoty
```

Let's implement ACL's on following scenario's.

Two Users (**tecmint1** and **tecmint2**), both having common secondary group named '**acl**'. We will create one directory owned by '**tecmint1**' and will provide the **read** and **execute** permission on that directory to user '**tecmint2**'.

Step 1: Create two users and remove password from both

```
[root@linux ~]# for user in tecmint1 tecmint2

> do
> useradd $user
> passwd -d $user
> done
Removing password for user tecmint1.
passwd: Success
Removing password for user tecmint2.
passwd: Success
```

Step 2: Create a Group and Users to Secondary Group.

Step 3: Create a Directory /tecmint and change ownership to tecmint1.

```
[root@linux ~]# mkdir /tecmint1
[root@linux ~]# chown tecmint1 /tecmint1/
```

```
[root@linux ~]# ls -ld /tecmint1/

drwxr-xr-x 2 tecmint1 root 4096 Apr 17 14:46 /tecmint1/
```

```
[root@linux ~]# getfacl /tecmint1

getfacl: Removing leading '/' from absolute path names
# file: tecmint1
# owner: tecmint1
# group: root
user::rwx
group::r-x
other::r-x
```

Step 4: Login with tecmint1 and create a Directory in /tecmint folder.

```
[tecmint@linux ~]$ su - tecmint1

Last login: Thu Apr 17 14:49:16 IST 2014 on pts/4
```

```
[tecmint1@linux ~]$ cd /tecmint1/
[tecmint1@linux tecmint1]$ mkdir example
```

```
[tecmint1@linux tecmint1]$ ll

total 4
drwxrwxr-x 2 tecmint1 tecmint1 4096 Apr 17 14:50 example
```

Step 5: Now set ACL using '**setfacl**', so that '**tecmint1**' will have all **rwx** permissions, '**tecmint2**' will have only **read** permission on '**example**' folder and other will have no permissions.

```
$ setfacl -m u:tecmint1:rwx example/
$ setfacl -m u:tecmint2:r-- example/
$ setfacl -m  other:--- example/
$ getfacl example/

# file: example
# owner: tecmint1
# group: tecmint1
user::rwx
user:tecmint1:rwx
user:tecmint2:r--
group::r-x
mask::rwx
other::---
```

Step 6: Now login with other user i.e. '**tecmint2**' on another terminal and change directory to '**/tecmint1**'. Now try to view the contents using '**ls**' command and then try to change directory and see the difference as below.

```
[tecmint@linux ~]$ su - tecmint2

Last login: Thu Apr 17 15:03:31 IST 2014 on pts/5
```

```
[tecmint2@linux ~]$ cd /tecmint1/
[tecmint2@linux tecmint1]$ ls -lR example/
example/:
total 0
```

```
[tecmint2@linux tecmint1]$ cd example/

-bash: cd: example/: Permission denied
```

```
[tecmint2@linux tecmint1]$ getfacl example/
```

```
# group: tecmint1
user::rwx
user:tecmint1:rwx
user:tecmint2:r--
group::rwx
mask::rwx
other::---
```

Step 7: Now give '**execute**' permission to '**tecmint2**' on '**example**' folder and then use '**cd**' command to see the effect. Now '**tecmint2**' have the permissions to view and change directory, but don't have permissions for writing anything.

```
[tecmint1@linux tecmint1]$ setfacl -m u:tecmint2:r-x example/
[tecmint1@linux tecmint1]$ getfacl example/

# file: example
# owner: tecmint1
# group: tecmint1
user::rwx
user:tecmint1:rwx
user:tecmint2:r-x
group::rwx
mask::rwx
other::---
```

```
[tecmint@linux ~]$ su - tecmint2

Last login: Thu Apr 17 15:09:49 IST 2014 on pts/5
```

```
[tecmint2@linux ~]$ cd /tecmint1/
[tecmint2@linux tecmint1]$ cd example/
[tecmint2@linux example]$ getfacl .
```

```
[tecmint2@linux example]$ mkdir test

mkdir: cannot create directory 'test': Permission denied
```

```
touch: cannot touch 'test': Permission denied
```

Note: After implementing ACL, you will see a extra '+' sign for 'ls –l' output as below.

```
[root@linux tecmint1]# ll

total 4
drwxrwx---+ 2 tecmint1 tecmint1 4096 Apr 17 17:01 example
```

Reference Links

ACL's Documentation

Tags: acls

Kuldeep Sharma View all Posts

Currently, Working in Middleware(Jboss/Apache Tomcat) And POSIX related technologies. Having more than 5 years of experience and love to do R&D on different open source tools/technologies.

Your name can also be listed here. Got a tip? Submit it here to become an TecMint author.

NEXT STORY

Ubuntu 14.04 Server Installation Guide and Setup LAMP (Linux, Apache, MySQL, PHP)

PREVIOUS STORY

Nmon: Analyze and Monitor Linux System Performance

YOU MAY ALSO LIKE...

How to Setup HTTPS (SSL Certificates) to Secure PhpMyAdmin Login

10 SEP, 2016

How to Secure Network Services Using TCP Wrappers in Linux

18 OCT, 2016

6 RESPONSES

Comments 6 Pingbacks 0

Kirankumar Badiger November 11, 2019 at 4:53 pm

Dear Tecmint team,

The explanation was very good, I have suggestion for you instead of creating **tecmint1** and **tecmint 2** users please use some different names so no one will confuse.

Reply

Angga November 6, 2019 at 9:09 am

Hi,

I tried to setACL "**setfacl -m u: user_dua: rw- / home / user_dua /**" can only read and write can not delete files.

after I try the WinSCP application at login user_dua can still delete files.

beg for your help and enlightenment

thank you

Reply

Hariharasudhan March 15, 2019 at 7:18 pm

Hi ,
cool explanation on ACL, I Have a question, what if My file permission is 700 and this is owned by some user and i have a acl for the group does the group can view write and execute in this file? or at least there should be a read permission to able to read write execute this file?

Thanks

Reply

arun natarajan June 20, 2016 at 5:45 pm
hi dude,

nice explanation.

i have a question for you.

Is ACL can be implemented between normal users alone ? is it possible to implement between root and normal user ?
like root user owned files can be accessed, executed by normal user bysetting up ACL permission ??

Reply

Ravi Saive June 21, 2016 at 11:08 am
@Arun,

Yes, I think it can be implemented using ACL, here are the guides on how to use ACL's in Linux.

https://www.tecmint.com/set-access-control-lists-acls-and-disk-quotas-for-users-groups/
https://www.tecmint.com/rhcsa-exam-configure-acls-and-mount-nfs-samba-shares/

Reply

GOT SOMETHING TO SAY? JOIN THE DISCUSSION.

Comment

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.

Notify me of followup comments via e-mail. You can also subscribe without commenting.

LINUX MONITORING TOOLS

How to Monitor Linux Server Security with Osquery

LibreNMS – A Fully Featured Network Monitoring Tool for Linux

Nmon: Analyze and Monitor Linux System Performance

20 Useful Commands of 'Sysstat' Utilities (mpstat, pidstat, iostat and sar) for Linux Performance Monitoring

Monitor Server Resources with Collectd-web and Apache CGI in Linux

LINUX INTERVIEW QUESTIONS

10 Linux Interview Questions and Answers for Linux Beginners – Part 3

10 Useful Interview Questions and Answers on Linux Commands

10 Useful 'Interview Questions and Answers' on Linux Shell Scripting

Nishita Agarwal Shares Her Interview Experience on Linux 'iptables' Firewall

15 Interview Questions on Linux "ls" Command – Part 1

OPEN SOURCE TOOLS

11 Best Graphical Git Clients and Git Repository Viewers for Linux

10 Best Media Server Software for Linux in 2019

10 Best Markdown Editors for Linux

My Favorite Command Line Editors for Linux – What's Your Editor?