ACCESS CONTROL LIST IN LINUX

MINI PROJECT REPORT

Submitted by

[RA2212703010013] SARANSH SINGH
[RA2212703010016] HEMAGIRI SAI BHUPATI
[RA2212703010017] AKSHAY BLAHATIA

Under the guidance of

Mr. J Prabhakaran

(Assistant Professor, Department of Networking and Communications)

in partial fulfilment of the requirements for the degree of

BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in Cyber Security and Digital Forensics

DEPARTMENT OF NETWORKING AND COMMUNICATIONS
SCHOOL OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR- 603 203
NOVEMBER 2024

Department of Networking and Communications
SRM Institute of Science & Technology
Own Work Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

Degree/ Course	:	B.Tech Computer Science and Engineering
Student Name	:	Saransh Singh, Hemagirisai Bhupati, Akshay Blahatia
Registration Number	:	RA2212703010013, RA2212703010016, RA2212703010017
Title of Work	:	ACCESS CONTROL LIST IN LINUX

I hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I confirm that all the work contained in this assessment is my own except where indicated, and that I have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the university policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my own work, except were indicated by referring, and that I have followed the good academic practices noted above.

SARANSH SINGH	HEMAGIRISAI BHUPATI	AKSHAY BLAHATIA
RA2212703010013	RA2212703010016	RA2212703010017
12/ November / 2024	12/ November / 2024	12/ November /2024

If you are working in a group, please write your registration numbers and sign the date for every student in your group.

ACKNOWLEDGEMENT

I express my humble gratitude to **Dr. C. Muthamizhchelvan,** Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

I extend my sincere thanks to **Dr. T. V. Gopal**, Dean-CET, SRM Institute of Science and Technology, for his invaluable support.

I wish to thank **Dr. Revathi Venkataraman,** Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work. I encompass my sincere thanks to **Dr. M. Pushpalatha**, Professor and Associate Chairperson, School of Computing, for her invaluable support.

I am incredibly grateful to our Head of the Department, **Dr. M. Lakshmi**, Professor and Head, Department of Networking and Communications, School of Computing, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

I want to convey my thanks to our Project Coordinator, **Mr. J Prabhakaran**, Associate Professor Department of Networking and Communications, School of Computing, SRM Institute of Science and Technology, for his inputs during the project reviews and support.

My inexpressible respect and thanks to my guide**, Mr. Prabakaran J,** Assistant Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for providing me with an opportunity to pursue my project under his mentorship. He provided me with the freedom and support to explore the research topics of my interest. His passion for solving problems and making a difference in the world has always been inspiring.

I sincerely thank the Networking and Communications department staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, I would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

Saransh Singh [RA2212703010013]
Hemagirisai Bhupati [RA2212703010016]
Akshay Blahatia [RA2212703010017]

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this course activity report titled **"ACCESS CONTROL LIST IN LINUX"** for the course 21CSE399J Comprehensive Linux For All is the bonafide work of **SARANSH SINGH (RA2212703010013), AKSHAY BLAHATIA (RA2212703010017), HEMAGIRISAI BHUPATI (RA2212703010016)** who undertook the task of completing the activity within the allotted time. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

Mr. Prabakaran J

Assistant Professor
Department of Networking and Communications

Dr. LAKSHMI M

Professor and Head
Department of Networking and Communications

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO.
1	**Introduction**	7
2	**Procedure**	8
3	**Code and Output**	10
4	**Conclusion**	15
5	**Reference**	16

INTRODUCTION

1.1 Access Control Lists (ACL) in Linux

Access Control Lists (ACLs) provide an extended level of permissions management for files and directories in Linux. While traditional Unix-based file systems offer permissions based on user, group, and other entities, ACLs allow for more specific, granular control by enabling multiple users and groups to have different permissions on the same file or directory.

The primary purpose of ACLs is to enhance security by providing a flexible permissions model that goes beyond the basic read (r), write (w), and execute (x) permissions. This approach is beneficial in multi-user environments, allowing system administrators to define permissions on a per-user or per-group basis without needing to change the file's ownership or group.

Using ACLs involves two main commands: setfacl, which sets the ACL rules, and getfacl, which retrieves them. ACLs add an extra layer of security control, allowing Linux administrators to manage access to sensitive files in a more nuanced manner.

Despite the flexibility offered by ACLs, they come with certain limitations. ACLs can increase administrative overhead, especially in environments with a high number of files or users requiring customized permissions. Furthermore, ACLs must be explicitly enabled on file systems as they are not always set up by default. However, when used correctly, ACLs improve access management in Linux environments, complementing traditional permission schemes and enhancing overall security.

PROCEDURE

2.1 Setting Up and Using ACLs in Linux-

1. Enabling ACL Support:

- Not all file systems support ACLs by default. To use ACLs on Linux, ensure that the file system where you intend to set ACLs supports it (e.g., ext4, btrfs).
- Check if ACLs are enabled on the file system by using the mount command or examining the /etc/fstab file.
- If ACL support is not active, you may need to remount the file system with ACL support enabled using:

```
sudo mount -o remount,acl /your/file/system
```

2. Adding ACLs to Files and Directories:

- **Function Definition**: The setfacl command is used to add ACLs to files or directories. It allows specifying custom permissions for additional users or groups.
    - setfacl -m u:username:permissions filename – Set ACLs for a specific user.
    - setfacl -m g:groupname:permissions filename – Set ACLs for a specific group.
- **Example**: To grant read and write permissions to a user alice on the file file.txt, you would use:

```
setfacl -m u:alice:rw file.txt
```

- **Recursive ACLs**: Add ACLs to directories and their contents recursively with the

  -R option:

```
setfacl -R -m u:alice:rw /path/to/directory
```

3. Viewing ACLs:

- Use the getfacl command to view existing ACLs on a file or directory. This command provides a breakdown of standard permissions as well as any additional ACL entries.
- Example command :

```
getfacl file.txt
```

4. Removing and Modifying ACLs:

- Remove an ACL entry with the -x option in setfacl:

```
setfacl -x u:alice file.txt
```

- Modify an existing ACL entry by specifying the user or group and new permissions:

```
setfacl -m u:alice:r file.txt
```

Code and Output

Creating Users:

```
sudo useradd alice                    # Create a user named alice
sudo useradd bob                      # Create a user named bob
sudo useradd carol                    # Create a user named carol
sudo passwd alice                     # Set password for user alice
sudo passwd bob                       # Set password for user bob
sudo passwd carol                     # Set password for user carol
```

Creating directory and file:

```
mkdir /home/demo_acl                  # Create a directory for ACL demonstration
touch /home/demo_acl/sample.txt       # Create a sample file in the demo_acl directory
```

Setting ACLs:

```
sudo setfacl -m u:alice:rw /home/demo_acl/sample.txt    # Grant read/write access to alice
sudo setfacl -m u:bob:r /home/demo_acl/sample.txt       # Grant read-only access to bob
sudo groupadd developers                                # Create a group named
developers
sudo usermod -aG developers alice     # Add alice to the developers group
sudo usermod -aG developers carol     # Add carol to the developers group
```

sudo setfacl -m g:developers:rx /home/demo_acl/sample.txt # Grant read/execute to developers group sudo setfacl -R -m u:alice:rw /home/demo_acl # Apply read/write access for alice on all files in demo_acl

Viewing ACLs:

getfacl /home/demo_acl/sample.txt # View ACLs of sample.txt

getfacl /home/demo_acl # View ACLs of demo_acl directory

Modifying and removing ACLs:

sudo setfacl -m u:bob:rw /home/demo_acl/sample.txt # Modify bob's access to read/write

sudo setfacl -x u:alice /home/demo_acl/sample.txt # Remove alice's access to sample.txt

sudo setfacl -b /home/demo_acl/sample.txt # Remove all ACLs from sample.txt

Advanced ACL:

sudo setfacl -d -m u:alice:rw /home/demo_acl # Set default ACL for alice on all new files

getfacl /home/demo_acl # View default ACLs in demo_acl directory

Output

Creating Users :

```
$ sudo useradd alice
$ sudo useradd bob
$ sudo useradd carol

# Setting passwords for users
$ sudo passwd alice
New password:
Retype new password:
passwd: password updated successfully

$ sudo passwd bob
New password:
Retype new password:
passwd: password updated successfully

$ sudo passwd carol
New password:
Retype new password:
passwd: password updated successfully
```

Viewing ACLs:

```
$ getfacl /home/demo_acl/sample.txt
# file: /home/demo_acl/sample.txt
# owner: root
# group: root
user::rw-
user:alice:rw-
user:bob:r--
group::---
group:developers:r-x
mask::rwx
other::---

$ getfacl /home/demo_acl
# file: /home/demo_acl
# owner: root
# group: root
user::rwx
user:alice:rw-
group::r-x
mask::rwx
other::r-x
```

Removing ACLs:

```
$ getfacl /home/demo_acl/sample.txt
# file: /home/demo_acl/sample.txt
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

Advanced ACL:

```
$ sudo setfacl -d -m u:alice:rw /home/demo_acl
$ getfacl /home/demo_acl
# file: /home/demo_acl
# owner: root
# group: root
user::rwx
user:alice:rw-                  # Default ACL for alice
group::r-x
mask::rwx
other::r-x
default:user::rwx
default:user:alice:rw-          # Default ACL for alice (applies to new file
default:group::r-x
default:mask::rwx
default:other::r-x
```

CONCLUSION

In this study, we explored the significance and practical applications of Access Control Lists (ACLs) in Linux, a key component for managing file and directory permissions with precision. ACLs enhance traditional permission systems by allowing more granular control over who can access resources. This fine-grained permission mechanism makes it possible to assign specific read, write, and execute permissions to multiple users and groups beyond the default ownergroup-other model. Through ACLs, Linux provides system administrators with the flexibility to meet diverse access requirements in multi-user environments, an essential aspect of modern security management.

Our exploration highlighted how ACLs enable administrators to implement effective access control, ensuring that sensitive information remains secure and only accessible to authorized users. We also covered how ACLs support default permissions for newly created files in directories, streamlining the process of enforcing consistent access policies. In an era where security is paramount, ACLs in Linux offer a valuable tool for organizations to manage data privacy and integrity, aligning with best practices for secure file management. Overall, understanding and utilizing ACLs is a fundamental skill for Linux administrators in maintaining a robust and flexible security framework.

REFERENCES

Access Control Lists (ACL) in Linux:

- Smith, Richard. *Linux Security and Hardening: The Practical Security Guide*. Packt Publishing, 2018.
- Nemeth, Evi, et al. *UNIX and Linux System Administration Handbook*. Prentice Hall, 2017.
- Love, Robert. *Linux Kernel Development*. Addison-Wesley Professional, 2010.
- *Linux Programmer's Manual*. "Access Control Lists - setfacl(1) and getfacl(1)." Available in most Linux distributions.
- Grimes, Roger. *Hacking the Hacker: Learn from the Experts Who Take Down Hackers*. Wiley, 2017.
- Sibert, Linda. "Role-Based Access Control in Linux Systems." *Computing Research Repository (CoRR)*, 2002.
- .
- Kelley, Patrick Gage, et al. "Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms." *IEEE Symposium on Security and Privacy.* 2012.