“Kyiv Vocation College of Communication”
Cyclical Commission of Computer Engineering

REPORT ON EXECUTION
LABORATORY WORK №9

on the discipline: "Operating Systems"

Topic: "System and user protection in Linux. Creating users and groups"

Performed by student
group RPZ-13a
Kateryna Hranat
Checked by teacher
V.S. Sushanova

Performed by student group RPZ-13a Kateryna Hranat

Kyiv 2024

Work objectives:

1. Acquiring practical skills in working with the Bash command shell.
2. Familiarization with basic actions for creating new users and new user groups.

Material Support for Classes:

1. IBM PC type computers.
2. Windows operating system and VirtualBox virtual machine (Oracle).
3. GNU/Linux operating system (any distribution).
4. Cisco Networking Academy website netacad.com and its online courses on Linux.

Performed by student group RPZ-13a Kateryna Hranat

Tasks for Preliminary Preparation:

1. Read the brief theoretical information for the laboratory work and create a small dictionary of basic English terms related to the purpose of commands and their parameters.

English terms	Ukrainian terms
Administrative Accounts	Адміністративні облікові записи
Non-administrative	Неадміністративні
User Accounts	Користувацькі облікові записи
System Accounts	Системні облікові записи
Group Accounts	Групові облікові записи
Modifying a Group	Зміна групи

2. Study the materials of the online course "NDG Linux Essentials" from Cisco:
    - Chapter 15 - System and User Security
    - Chapter 16 - Creating Users and Groups

Complete✔

3. Complete testing in the NDG Linux Essentials course on the following topics:
    - Chapter 15 Exam
    - Chapter 16 Exam

Complete✔

4. Based on the material covered, provide answers to the following questions:

4.1 Explain the concept of UPG. When is it appropriate to use them?

UPG stands for User Private Groups. These are groups created for each user when their account is created. UPGs provide a way to control access permissions to files and directories, particularly when multiple users need access to shared files without sharing them with everyone.

They are useful in situations where you want to restrict access to specific files or directories to only a select group of users.

4.2 *What commands can be used to create user groups? Provide examples.

The primary command used to create user groups in Unix-like operating systems is `groupadd`. Here's an example:

groupadd mygroup

This command creates a new group named "mygroup".

4.3 **What commands can be used to modify the settings of user groups? Provide examples.

The primary command used to modify user group settings in Unix-like operating systems is `groupmod`. Here's an example:

groupmod -g 1001 mygroup

This command changes the GID (Group ID) of the group "mygroup" to 1001.

5. Prepare an initial version of the report in electronic form:
   - Title page, topic, and purpose of the work
   - Glossary of terms
   - Answers to points 4.1 - 4.3 from the tasks for preliminary preparation

Complete✔

Performed by student group RPZ-13a Kateryna Hranat

Progress of Work:

1. Initial work in CLI mode in a Linux operating system of the Linux family:
   1.1. Start your Linux-based operating system (if you are using your own PC and have it installed) and open the terminal.
   Work through all the command examples provided in the lab assignments of the NDG Linux Essentials - Lab 15: System and User Security and Lab 16: Creating Users and Groups. Create a table to describe these commands.

Command	**Description**	**Example**
`groupadd`	Create a new group	`groupadd mygroup`
`groupmod`	Modify group attributes	`groupmod -g 1001 mygroup`
`useradd`	Create a new user	`useradd -m -s /bin/bash myuser`
`passwd`	Change user password	`passwd myuser`
`usermod`	Modify user attributes	`usermod -G group1,group2 myuser`
`chown`	Change file or directory owner and group	`chown user:group file.txt`
`chmod`	Change file permissions	`chmod 755 file.txt`
`su`	Switch user	`su - myuser`
`sudo`	Execute a command as another user (usually root)	`sudo command`

2. Execute practical tasks in the terminal (demonstrate screenshots):

- Practice the commands last, w, and who in the terminal. Compare the output of each command and identify the details missing in each compared to the others.

last, w, and who commands provide information about the users who are currently logged into the system or have previously logged in. They display details such as login time, duration of the session, and terminal used.

- *Create two new user groups - super_admins, noob_users, and good_students, defining their identifiers.

```
sysadmin@localhost:~$ sudo groupadd super_admins
[sudo] password for sysadmin:
sysadmin@localhost:~$ sudo groupadd noob_users
sysadmin@localhost:~$ sudo groupadd good_students
sysadmin@localhost:~$
```

- *Using the terminal, create a new user for each member of your team (or simply three arbitrary users if you're working alone), don't forget to set a password for each new user immediately after creation.

```
sysadmin@localhost:~$ sudo useradd -m -s /bin/bash user1
sysadmin@localhost:~$ sudo passwd user1
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
sysadmin@localhost:~$ sudo useradd -m -s /bin/bash user2
sysadmin@localhost:~$
sysadmin@localhost:~$ sudo passwd user2
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
sysadmin@localhost:~$ sudo useradd -m -s /bin/bash user3
sysadmin@localhost:~$
sysadmin@localhost:~$ sudo passwd user3
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
sysadmin@localhost:~$
```

- *Add the new users to the groups you created in such a way that there are 2 users in the super_admins and noob_users groups, one of which belongs to both groups. Add all three users to the good_students group.

```
sysadmin@localhost:~$ sudo usermod -aG super_admins,user2,user3 user1
sysadmin@localhost:~$
sysadmin@localhost:~$ sudo usermod -aG noob_users,user1 user2
sysadmin@localhost:~$
sysadmin@localhost:~$ sudo usermod -aG good_students user1
sysadmin@localhost:~$
sysadmin@localhost:~$ sudo usermod -aG good_students user2
sysadmin@localhost:~$
sysadmin@localhost:~$ sudo usermod -aG good_students user3
sysadmin@localhost:~$
sysadmin@localhost:~$
```

- **View information about the groups and which users belong to them, explaining what you see.

```
staff:x:50:
games:x:60:
users:x:100:
nogroup:x:65534:
systemd-journal:x:101:
systemd-network:x:102:
systemd-resolve:x:103:
mlocate:x:104:
uuidd:x:105:
input:x:106:
crontab:x:107:
syslog:x:108:
messagebus:x:109:
ssh:x:110:
bind:x:111:
sysadmin:x:1001:
super_admins:x:1002:user1
noob_users:x:1003:user2
good_students:x:1004:user1,user2,user3
user1:x:1005:user2
user2:x:1006:user1
user3:x:1007:user1
```

```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,sysadmin
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:
floppy:x:25:
tape:x:26:
sudo:x:27:sysadmin
```

- **Delete the first user you created and check if there is still information about them in the groups they were part of.

```
sysadmin@localhost:~$ sudo deluser user1
Removing user `user1' ...
userdel: group user1 not removed because it has other members.
Done.
```

```
sysadmin@localhost:~$ grep user1 /etc/group
user1:x:1005:user2
```

- **Delete the second user and check if there is still information about them in the groups they were part of.

```
sysadmin@localhost:~$ sudo deluser user2
Removing user `user2' ...
Warning: group `user2' has no more members.
Done.
```

```
sysadmin@localhost:~$ grep user2 /etc/group
```

- **Delete the third user and check if there is still information about them in the groups they were part of.

```
sysadmin@localhost:~$ sudo deluser user3
Removing user `user3' ...
Warning: group `user3' has no more members.
Done.
```

```
sysadmin@localhost:~$ grep user3 /etc/group
```

- **View information about the existing user groups.

```
sasl:x:45:
plugdev:x:46:
staff:x:50:
games:x:60:
users:x:100:
nogroup:x:65534:
systemd-journal:x:101:
systemd-network:x:102:
systemd-resolve:x:103:
mlocate:x:104:
uuidd:x:105:
input:x:106:
crontab:x:107:
syslog:x:108:
messagebus:x:109:
ssh:x:110:
bind:x:111:
sysadmin:x:1001:
super_admins:x:1002:
noob_users:x:1003:
good_students:x:1004:
user1:x:1005:
```

- **Delete the user groups you created.

```
sysadmin@localhost:~$ sudo groupdel super_admins
sysadmin@localhost:~$
sysadmin@localhost:~$ sudo groupdel noob_users
sysadmin@localhost:~$
sysadmin@localhost:~$ sudo groupdel good_students
```

- **View information about the existing user groups again.

```
shadow:x:42:
utmp:x:43:
video:x:44:
sasl:x:45:
plugdev:x:46:
staff:x:50:
games:x:60:
users:x:100:
nogroup:x:65534:
systemd-journal:x:101:
systemd-network:x:102:
systemd-resolve:x:103:
mlocate:x:104:
uuidd:x:105:
input:x:106:
crontab:x:107:
syslog:x:108:
messagebus:x:109:
ssh:x:110:
bind:x:111:
sysadmin:x:1001:
user1:x:1005:
```

Control questions

1. Why are passwords not stored in plain text in configuration files?

Passwords are not stored in plain text in configuration files for security reasons. Storing passwords in plain text makes them vulnerable to unauthorized access if the file is compromised. Instead, passwords are typically hashed or encrypted before storage to protect sensitive information.

2. Why is it not recommended to perform everyday operations using the root account?

It is not recommended to perform everyday operations using the root account because the root account has unrestricted access to system resources and commands. Accidental or malicious actions performed as root can have severe consequences, including system instability, data loss, or security breaches. Using a regular user account for everyday tasks helps minimize the risk of accidental damage to the system.

3. *What is the difference between the mechanisms for obtaining special privileges using su and sudo?

The main difference between su and sudo is in how they grant special privileges. su (switch user) allows a user to switch to another user account, typically the root account, after providing the root password. sudo (superuser do) allows a permitted user to execute a command with the privileges of another user, typically the root user, after authenticating themselves with their own password.

4. *Why is the root user's home directory not located in the /home directory?

The root user's home directory is typically not located in the /home directory for organizational and security reasons. Placing the root user's home directory in a separate location helps differentiate system files and directories from user-specific files. Additionally, separating the root user's home directory from regular user home directories reduces the likelihood of accidentally modifying or deleting critical system files.

5. *What is the purpose of the getent command?

The getent command is used to retrieve entries from various name service databases, including passwd, group, hosts, and others. It allows users to query information stored in these databases, such as user accounts, groups, and network configuration, without directly accessing the underlying files or databases.

6. *How can you change a user's password?

You can change a user's password using the passwd command followed by the username of the user whose password you want to change. For example:

```
passwd username
```

7. **How can existing user groups be deleted? Will there be any information about them left in the system?

Existing user groups can be deleted using the groupdel command followed by the group name. For example:

```
sudo groupdel groupname
```

Information about deleted groups may still exist in log files or other system records, but they will no longer be present in the /etc/group file or actively maintained by the system.

8. **What is the purpose of the chage command?

The chage command is used to change the aging information for a user account, such as password expiration date and password aging policies. It allows system administrators to manage user account security by configuring password expiration and other policies.

9. **Which parameters of the usermod command do you consider the most commonly used?

The most commonly used parameters of the usermod command include:

- `-aG`: Add user to supplementary groups.
- `-d`: Specify the user's home directory.
- `-s`: Specify the user's login shell.
- `-l`: Change the username.
- `-e`: Set the account expiration date.

Conclusions:

In conclusion, I have better understood the support of password security in the configuration files, the restriction of the use of the root account. Understanding privilege management mechanisms and implementing effective user account administration are critical to ensuring system integrity and security. In addition, strategic organizational choices and methods of preventive maintenance of the system further strengthen the overall protection of the system against potential threats.