**Module**  **18**

## Linux server - Manage user and Groups and working with file systems

* Assignment Level Basic
  1. What is default uid for root user ?

In most Unix-like operating systems, including Linux, the default user ID (UID) for the root user is 0. The root user is also known as the superuser or administrator, and it has the highest level of privileges on the system. The UID 0 is reserved exclusively for the root user, and it grants full access and control over all files, directories, processes, and system resources.

* 1. What is default uid for system user ?

The default UID for system users can vary depending on the specific operating system and configuration. In general, system users are assigned UIDs that fall within a specific range reserved for system accounts.

On most Unix-like systems, including Linux, system users typically have UIDs in the range from 1 to 999. The actual UID assigned to a specific system user may vary based on the system's configuration and any existing user accounts.

It's worth noting that system users are typically used to run specific services or perform system-related tasks and are not intended for regular user interaction.

* 1. What is the uid for normal users ? 4. How to add comment in user file?

The UID for normal users can vary depending on the specific operating system and its configuration. In many Unix-like systems, including Linux, the UIDs for normal users typically start from 1000 and increment for each new user added to the system. The exact starting UID and increment value can be defined in the system's configuration files, such as /etc/login.defs or /etc/default/useradd. However, it's important to note that the specific UID ranges and configurations can differ between different distributions and setups.

To add a comment or description for a user in the user file, you can use the chfn command or directly edit the /etc/passwd file.

Using the chfn command:

rust

Copy code

chfn username

This command will open an editor where you can enter the desired comment or description for the user. Save and exit the editor to apply the changes.

Editing the /etc/passwd file directly:

bash

Copy code

sudo nano /etc/passwd

Locate the line corresponding to the user you want to add a comment for and edit the field containing the comment. The comment field is typically the "GECOS" field, which is the fifth field in the /etc/passwd file. After making the necessary changes, save the file.

It's important to exercise caution when modifying system files like /etc/passwd and to make backups or use appropriate tools to ensure the integrity of the file.

* 1. From “ /etc/passwd “ which information will we gather ?

The "/etc/passwd" file in Unix-like systems contains essential information about user accounts. Each line in the file represents a user account and is structured in a colon-separated format with several fields. The information you can gather from the "/etc/passwd" file includes:

Username: The first field specifies the username associated with the user account.

Password: The second field, usually represented as an "x", indicates the user's password is stored in the "/etc/shadow" file or an alternative authentication mechanism.

User ID (UID): The third field denotes the unique numerical identifier (UID) assigned to the user.

Group ID (GID): The fourth field represents the numerical identifier (GID) of the primary group associated with the user.

User Info/Comment: The fifth field, also known as the GECOS field, often contains additional user information or a comment/description.

Home Directory: The sixth field specifies the user's home directory path.

Shell: The seventh field indicates the user's default shell.

By examining the "/etc/passwd" file, you can gather these details about user accounts on the system. However, note that passwords are typically stored in a separate file, such as "/etc/shadow," for security reasons, and the entry in "/etc/passwd" usually shows an "x" in the password field.

* 1. From “ /etc/shadow “ which information will we gather ?

The "/etc/shadow" file is a crucial file in Unix-like systems that stores secure user account information, particularly the encrypted passwords. It is readable only by the root user to prevent unauthorized access. The information you can gather from the "/etc/shadow" file includes:

Username: The first field corresponds to the username associated with the user account.

Password: The second field contains the encrypted password or a placeholder (such as "!" or "\*") indicating that the account does not have a password set or is locked.

Last Password Change: The third field denotes the date of the last password change, represented as the number of days since January 1, 1970 (epoch time).

Minimum Password Age: The fourth field specifies the minimum number of days that must pass before the user can change their password.

Maximum Password Age: The fifth field indicates the maximum number of days after which the user must change their password.

Password Expiration Warning: The sixth field represents the number of days before the password expires when the user will start receiving expiration warnings.

Account Inactivity: The seventh field denotes the number of days of inactivity allowed before the account is automatically disabled.

Account Expiration: The eighth field indicates the expiration date of the account, represented as the number of days since January 1, 1970 (epoch time).

Reserved Field: The ninth field is reserved for future use and is typically empty.

The "/etc/shadow" file provides essential information related to password policies, password expiration, and account management for user accounts on the system.

* 1. From “ /etc/group “ which information will we gather ?

The "/etc/group" file in Unix-like systems contains information about user groups on the system. It provides details about group names, group IDs (GIDs), and the members belonging to each group. The information you can gather from the "/etc/group" file includes:

Group Name: The first field represents the name of the group.

Password: The second field typically contains an "x" or an encrypted password, but it is not commonly used and is often left empty.

Group ID (GID): The third field denotes the unique numerical identifier assigned to the group.

Group Members: The fourth field lists the usernames of users who are members of the group. Multiple usernames are separated by commas.

By examining the "/etc/group" file, you can gather information about the groups present on the system, their respective GIDs, and the users who belong to each group. This file is useful for managing group-based permissions and access control on Unix-like systems.

* 1. From “ /etc/gshadow “ which information will we gather ?

The "/etc/gshadow" file is a shadowed version of the "/etc/group" file in Unix-like systems. It is typically readable only by the root user and contains secure group account information. The "/etc/gshadow" file provides additional security for group-related information by storing the group passwords and other sensitive data. The information you can gather from the "/etc/gshadow" file includes:

Group Name: The first field represents the name of the group.

Password: The second field contains the encrypted password or a placeholder (such as "!" or "\*") indicating that the group does not have a password set.

Group Administrators: The third field lists the usernames of users who are designated as administrators or group managers.

Group Members: The fourth field lists the usernames of users who are members of the group.

Reserved Field: The fifth field is reserved for future use and is typically empty.

The "/etc/gshadow" file is intended to provide an additional layer of security for group-related information, particularly group passwords and administrator privileges. It helps protect sensitive group data from unauthorized access and is only accessible to the root user for management and administration purposes.

* 1. What is the meaning of + and – in file permission?

In Unix-like systems, file permissions are represented by a combination of letters and symbols that define the access rights for different categories of users. The symbols "+" and "-" have specific meanings in relation to file permissions:

"+": When the "+" symbol appears in file permissions, it indicates that an additional permission or access right is granted in addition to the base permissions. For example, if a file has the permission "-rw-r--r--+" (readable and writable by the owner, readable by the group and others, with an additional permission), the additional permission could be an extended ACL (Access Control List) or special permissions such as setuid, setgid, or sticky bit.

"-": The "-" symbol, when used in file permissions, signifies the absence or removal of a specific permission. For example, if a file has the permission "-rwxrwxrwx" (readable, writable, and executable by the owner, group, and others), removing the write permission for others would result in the permission "-rwxrwxr-x".

The "+" and "-" symbols are used to modify or indicate changes to the base permissions of a file or directory. They are used in conjunction with the letters "r" (read), "w" (write), and "x" (execute) to define the access permissions for the owner, group, and others.

* 1. What is “ r “ “ w ” ‘ x “ in file permission 11. What is “ 4 “ “ 2 “ “1” in files permission

In file permissions, the letters "r," "w," and "x" represent the different access rights or permissions that can be granted to different categories of users:

"r" stands for "read" permission. It allows the user to view the contents of a file or list the files in a directory.

"w" stands for "write" permission. It allows the user to modify or edit the contents of a file or create, delete, or rename files within a directory.

"x" stands for "execute" permission. It allows the user to execute or run a file if it is a program or script, or enter into a directory.

The combination and arrangement of these permissions determine what actions can be performed on a file or directory by the owner, group, and others.

In file permissions, the numbers 4, 2, and 1 represent the numeric values assigned to the "read" (r), "write" (w), and "execute" (x) permissions, respectively, when using numeric representation of file permissions.

The number 4 corresponds to the read permission (r).

The number 2 corresponds to the write permission (w).

The number 1 corresponds to the execute permission (x).

These numeric values are used in combination to represent file permissions using octal notation. For example, in the octal representation "644" for a file, the first digit "6" represents the owner's permissions (read and write), the second digit "4" represents the group's permissions (read only), and the third digit "4" represents the permissions for others (read only).

* 1. What is the use of umask?

The umask (user file-creation mode mask) is a command and a concept in Unix-like operating systems that determines the default permissions applied to newly created files and directories by a user or process. It acts as a mask or filter to modify the permissions that are automatically assigned when a file or directory is created.

The umask value consists of a three-digit octal number (e.g., 022 or 077) that represents the permissions to be masked or subtracted from the default permissions. Each digit in the umask value corresponds to a different set of permissions: user (owner), group, and others.

The umask works by subtracting the specified permission bits from the default permissions. For example, if the umask is set to 022, it subtracts the write (w) permission for group and others, resulting in default permissions of 644 (rw-r--r--) for newly created files and 755 (rwxr-xr-x) for newly created directories.

The primary purpose of the umask is to enhance security and control the default permissions for files and directories. It allows users and administrators to define the level of privacy and access restrictions they want to enforce on their system. By setting an appropriate umask value, unnecessary permissions can be restricted by default, ensuring that sensitive files and directories are not overly permissive when created.

* 1. What is default root permission for directory?

The default root permission for a directory in Unix-like systems, including Linux, is typically "rwx" (read, write, and execute) for the owner (root user) and "r-x" (read and execute) for the group and others.

When a directory is created by the root user, it is usually assigned the default permissions of "rwxr-xr-x" (or 755 in octal notation). This means that the root user has full access to the directory, including the ability to read its contents, create, delete, and modify files within it, as well as traverse into the directory. The group and other users have read and execute permissions, allowing them to view the contents of the directory and traverse into it, but not modify or create files within it.

These default permissions for the root user ensure that critical system directories and files are protected and accessible only to the superuser, while still allowing other users to access necessary files and directories as needed. It's worth noting that specific systems or configurations may have variations in default permissions, so it's always advisable to check the system documentation or consult the specific distribution's guidelines for the accurate default root permission settings.

* Assignment Level Intermediate
  1. How to assign another new home directory for new user? 2. Command to check group membership of any user

To assign a new home directory for a new user, you can follow these steps:

Create the new user using the useradd command, specifying the desired username and options:

bash

Copy code

sudo useradd -m -d /path/to/new\_home\_directory username

Replace "/path/to/new\_home\_directory" with the actual path where you want to set the new home directory and "username" with the desired username for the new user. The -m option ensures that the home directory is created.

Set the appropriate ownership and permissions for the new home directory:

bash

Copy code

sudo chown -R username:username /path/to/new\_home\_directory

Replace "username" with the actual username and "/path/to/new\_home\_directory" with the path of the new home directory.

Optionally, if you want to update the user's home directory path after it has been created, you can use the usermod command:

bash

Copy code

sudo usermod -d /path/to/new\_home\_directory username

Again, replace "/path/to/new\_home\_directory" with the desired path and "username" with the username of the user.

The command to check the group membership of any user is groups. To check the group membership of a specific user, simply run the following command:

bash

Copy code

groups username

Replace "username" with the actual username of the user you want to check. This command will display a list of groups to which the user belongs.

* 1. What happened if I use “ su – “ command ?

The "su -" command is used in Unix-like systems to switch to another user account, typically the root user, with a full login environment. When you use the "su -" command, the following actions occur:

Authentication: You will be prompted to enter the password for the target user account you want to switch to. In the case of switching to the root user, you will need to provide the root password.

Environment setup: Once authenticated, the "su -" command sets up a new shell session with an environment similar to a fresh login session for the target user. This includes loading the target user's shell configuration files (such as .bashrc, .profile, etc.) and setting up the user's environment variables.

Current working directory: The command also changes your current working directory to the target user's home directory. So, if you were in a different directory before running "su -", you will be switched to the target user's home directory.

Permissions and privileges: By default, "su -" gives you the privileges and permissions of the target user. In the case of switching to the root user, you gain administrative access and can perform system-wide actions that are typically restricted to the root user.

Overall, the "su -" command allows you to assume the identity and environment of another user, usually the root user, with full access rights and a fresh login environment. This is useful for performing administrative tasks or accessing resources that require higher privileges.

* 1. Which command is used to delete any user with its home directory?

The command used to delete a user account along with its home directory varies depending on the specific operating system or distribution you are using. Here are two commonly used commands:

userdel:

The "userdel" command is used to delete a user account in many Unix-like systems. By default, it does not remove the user's home directory. However, you can use the "-r" option to remove the home directory as well. Here's the syntax:

Copy code

sudo userdel -r username

Replace "username" with the username of the user you want to delete. The "-r" option ensures that the user's home directory and associated files are removed along with the account.

deluser:

In some distributions, such as Debian-based systems, the "deluser" command is available as a user account management tool. Similar to "userdel", it can be used to delete a user account along with its home directory. Here's the syntax:

arduino

Copy code

sudo deluser --remove-home username

Replace "username" with the username of the user you want to delete. The "--remove-home" option ensures that the user's home directory is removed.

Note: It's important to exercise caution when deleting user accounts and their associated files. Make sure you have a backup or have verified that the account and its files are no longer needed before proceeding with the deletion.

* 1. How to add new user without home directory ?

To add a new user without creating a home directory, you can use the useradd command with the --no-create-home or -M option. Here's the syntax:

bash

Copy code

sudo useradd --no-create-home username

Replace "username" with the desired username for the new user. The --no-create-home option instructs useradd not to create a home directory for the user.

After executing the command, the new user account will be created without a home directory. This is useful when you want to create a user account that does not require a personal home directory, such as system accounts or service accounts.

Note: Without a home directory, the user won't have a designated location to store personal files or configurations. If the user needs a specific directory for its use, you may need to manually create and assign the appropriate permissions to a directory as needed.

* 1. Command to assign account expiry to the user ?

To assign an account expiry to a user in Unix-like systems, you can use the chage command. The chage command allows you to modify the user account aging and expiry information, including setting an account expiry date. Here's the syntax:

bash

Copy code

sudo chage -E YYYY-MM-DD username

Replace "YYYY-MM-DD" with the desired account expiry date in the format Year-Month-Day, and "username" with the username of the user account you want to modify.

For example, to set an account expiry for the user "john" to July 31, 2023, you would run:

bash

Copy code

sudo chage -E 2023-07-31 john

After executing the command, the user account will have an expiry date set. Once the expiry date is reached, the user account will no longer be able to log in.

You can also use the chage command with other options to modify additional account aging information such as password expiration, minimum and maximum password age, warning period, and more. To learn more about the available options, you can refer to the chage command's manual page by running man chage.

* 1. Command to add a new group …

To add a new group in Unix-like systems, you can use the groupadd command. Here's the syntax:

bash

Copy code

sudo groupadd groupname

Replace "groupname" with the desired name for the new group.

For example, to create a group named "developers", you would run:

bash

Copy code

sudo groupadd developers

After executing the command, the new group will be created.

By default, the groupadd command assigns the next available Group ID (GID) to the new group. If you want to specify a specific GID for the group, you can use the -g option followed by the desired GID:

bash

Copy code

sudo groupadd -g GID groupname

Replace "GID" with the desired Group ID and "groupname" with the name of the group.

After creating the group, you can assign users to the group using the usermod command with the -G option:

bash

Copy code

sudo usermod -a -G groupname username

Replace "groupname" with the name of the group and "username" with the username of the user you want to add to the group. The -a option ensures that the user is appended to the group without removing them from their existing groups.

* 1. What is default root permission for file?

The default root permission for a file in Unix-like systems, including Linux, is typically "rw-r--r--" (readable and writable by the owner, readable by the group, and readable by others).

When a file is created by the root user, it is usually assigned the default permissions of "rw-r--r--" (or 644 in octal notation). This means that the root user has full read and write access to the file, allowing them to modify its contents, while the group and other users have only read access, allowing them to view the contents of the file.

These default permissions for the root user ensure that critical system files are protected and accessible only to the superuser, while still allowing other users to access necessary files as needed. However, specific systems or configurations may have variations in default permissions, so it's always advisable to check the system documentation or consult the specific distribution's guidelines for accurate default root permission settings.

* 1. What is the default umask for root?

The default umask for the root user in Unix-like systems, including Linux, is typically "022".

The umask value for the root user determines the default permissions assigned to newly created files and directories. In the case of the umask value "022", it subtracts the write (w) permission for group and others, resulting in default permissions of "644" (rw-r--r--) for newly created files and "755" (rwxr-xr-x) for newly created directories.

However, it's important to note that the default umask for the root user can be customized or modified based on system configurations or administrator preferences. Therefore, the actual default umask for the root user may vary depending on the specific operating system or distribution you are using. It's always recommended to consult the system documentation or check the specific distribution's guidelines for the accurate default umask value for the root user in your environment.

* 1. What is the default umask for student?

The default umask for a student or regular user in Unix-like systems, including Linux, is typically "002".

The umask value for a user determines the default permissions assigned to newly created files and directories. In the case of the umask value "002", it subtracts the write (w) permission for others, resulting in default permissions of "664" (rw-rw-r--) for newly created files and "775" (rwxrwxr-x) for newly created directories.

However, it's important to note that the default umask for a student or regular user can be customized or modified based on system configurations or administrator preferences. Therefore, the actual default umask for a student may vary depending on the specific operating system or distribution you are using. It's always recommended to consult the system documentation or check the specific distribution's guidelines for the accurate default umask value for regular users or students in your environment.

* 1. Which command is used to set user ownership?

The command used to set user ownership of a file or directory in Unix-like systems is chown (short for "change owner"). Here's the syntax:

bash

Copy code

sudo chown username filename/directoryname

Replace "username" with the desired username or user ID of the new owner, and "filename/directoryname" with the name of the file or directory you want to change ownership for.

For example, to set the user ownership of a file named "myfile.txt" to a user named "john", you would run:

bash

Copy code

sudo chown john myfile.txt

To set the user ownership of a directory named "mydirectory" to a user with the user ID 1001, you would run:

bash

Copy code

sudo chown 1001 mydirectory

The chown command can also accept the -R option to change ownership recursively, which means it will change ownership not only for the specified file or directory but also for all files and subdirectories within it:

bash

Copy code

sudo chown -R username directoryname

This is useful when you want to change ownership for a directory and all its contents.

Please note that the chown command requires administrative privileges, so you may need to use sudo or run the command as the root user.

* 1. Which command is used to set group ownership?

The command used to set group ownership of a file or directory in Unix-like systems is chown (short for "change owner"). Here's the syntax:

bash

Copy code

sudo chown :groupname filename/directoryname

Replace "groupname" with the desired group name or group ID of the new owner, and "filename/directoryname" with the name of the file or directory you want to change ownership for.

For example, to set the group ownership of a file named "myfile.txt" to a group named "staff", you would run:

bash

Copy code

sudo chown :staff myfile.txt

To set the group ownership of a directory named "mydirectory" to a group with the group ID 1001, you would run:

bash

Copy code

sudo chown :1001 mydirectory

The colon (":") before the groupname or group ID indicates that you are specifying the group ownership.

The chown command can also accept the -R option to change ownership recursively, which means it will change ownership not only for the specified file or directory but also for all files and subdirectories within it:

bash

Copy code

sudo chown -R :groupname directoryname

This is useful when you want to change ownership for a directory and all its contents.

Please note that the chown command requires administrative privileges, so you may need to use sudo or run the command as the root user.

* Assignment Level Advance
  1. I have on user with the name of KAMAL, Now, I want to add this user in the group name N which command will used?

add the user "KAMAL" to the group "N", you can use the usermod command with the -aG option. Here's the syntax:

bash

Copy code

sudo usermod -aG N KAMAL

The -aG option ensures that the user "KAMAL" is appended to the group "N" without removing them from their existing groups.

After executing the command, the user "KAMAL" will be added to the group "N". This allows the user to have the group's permissions and access resources associated with the group.

Please note that the command requires administrative privileges, so you may need to use sudo or run the command as the root user.

* 1. What is the difference between “ usermod -G “ and “ usermod -aG “

The usermod command is used to modify user account properties in Unix-like systems. The options -G and -aG are used with usermod to add a user to one or more groups. Here's the difference between them:

usermod -G groupname username:

The -G option replaces the existing group membership of the user with the specified group (groupname).

This means that the user will be removed from any other groups they were previously a member of and will only be a member of the specified group.

If you use this option without specifying any groups, the user will be removed from all other groups, effectively leaving them with only the specified group as their group membership.

It is important to note that if you don't specify the -a option, the user will be removed from all other groups, including their primary group, which may cause unexpected behavior.

usermod -aG groupname username:

The -aG option appends the specified group (groupname) to the user's existing group membership.

This means that the user will remain a member of their existing groups and will also be added as a member of the specified group.

This option allows you to add the user to multiple groups without removing them from their current group memberships.

It is important to use the -a option along with -G to ensure that the specified group is appended to the user's existing group memberships instead of replacing them.

In summary, the -G option replaces the user's existing group membership with the specified group, while the -aG option appends the specified group to the user's existing group memberships. The choice between them depends on whether you want to replace or append the group memberships for the user.

* 1. What is the meaning of “ -1 “ in password state information?

In password state information, the value "-1" typically represents a locked password field. It indicates that the user account has been locked and the user is not able to authenticate using a password.

In Unix-like systems, including Linux, the /etc/shadow file stores password-related information for user accounts. Each line in the /etc/shadow file represents a user account, and the second field of the line contains the password state information.

The password state field has various values to indicate different conditions of the password. The most common values are:

"!!" or "!" (two exclamation marks or an exclamation mark alone): It signifies that the password is locked or disabled, and the user cannot log in using a password.

"\*": It indicates that the user account is locked, and the user cannot log in using a password.

"": An empty field indicates that the account has no password set. This is often the case for accounts that use alternative authentication methods, such as SSH key-based authentication.

The value "-1" specifically represents a locked password field, indicating that the user account is locked and password-based authentication is disabled. This can be used to temporarily disable a user account or enforce other authentication methods, such as key-based authentication or single sign-on (SSO).

It's worth noting that the interpretation of password state information may vary slightly depending on the specific operating system or configuration. It's always recommended to consult the system documentation or specific distribution's guidelines for accurate information on password state codes.

* 1. Which command tis used to remove the password of any user?

To remove the password of a user in Unix-like systems, you can use the passwd command with the -d option. Here's the syntax:

bash

Copy code

sudo passwd -d username

Replace "username" with the username of the user whose password you want to remove.

For example, to remove the password of a user named "john", you would run:

bash

Copy code

sudo passwd -d john

After executing this command, the password for the specified user will be removed. This means that the user will no longer be required to enter a password to log in. However, it's important to note that removing the password for a user effectively allows anyone with physical or remote access to the system to log in as that user without authentication.

Please exercise caution when removing passwords, and ensure that you have a valid reason and appropriate security measures in place before taking such action. It's generally recommended to have passwords or alternative forms of authentication to secure user accounts.

* 1. What is the use of “ gpasswd “ ?

The gpasswd command is used to administer group passwords in Unix-like systems. It allows you to manage the group password and group membership. Here are the main uses of the gpasswd command:

Set or change the group password: You can use gpasswd to set or change the password for a group. This is done by the group owner or a user with appropriate permissions. The command syntax is:

bash

Copy code

sudo gpasswd groupname

After running this command, you will be prompted to enter the new password for the group. If the group already has a password set, running this command again will prompt you to enter the new password to replace the existing one.

Add or remove members from a group: With gpasswd, you can add or remove members from a group. This is typically done by the group owner or a user with appropriate permissions. The command syntax for adding a member is:

bash

Copy code

sudo gpasswd -a username groupname

Replace "username" with the username of the user you want to add, and "groupname" with the name of the group you want to add the user to.

To remove a member from a group, use the -d option:

bash

Copy code

sudo gpasswd -d username groupname

This will remove the specified user from the group.

List group members: You can use gpasswd to display a list of members in a group. The command syntax is:

bash

Copy code

gpasswd -M groupname

Replace "groupname" with the name of the group you want to list the members for. This will show you a list of usernames belonging to the specified group.

Please note that the gpasswd command requires administrative privileges, so you may need to use sudo or run the command as the root user to perform these actions.

* 1. Command to change password policy

The command to change the password policy varies depending on the specific operating system or distribution you are using. Here are a few common examples:

For systems using the passwd command:

You can use the passwd command with the -l option to modify the password policy.

The specific options and their meanings may vary depending on the system.

To view the available options and their descriptions, you can use the man passwd command to access the manual page for passwd.

For systems using the pam\_pwquality module:

The pam\_pwquality module is a common module used for enforcing password policies.

The configuration file for pam\_pwquality may be located at /etc/security/pwquality.conf.

You can modify the parameters in this configuration file to change the password policy.

After making changes, you may need to restart the relevant services or processes for the changes to take effect.

For systems using the pluggable authentication modules (PAM) framework:

The PAM framework provides a flexible way to configure and enforce password policies.

The configuration files for PAM may be located in the /etc/pam.d/ directory.

You can edit the relevant configuration file, such as common-password, to modify the password policy.

The specific parameters and their meanings may vary depending on the system and the modules configured.

Please note that modifying the password policy requires administrative privileges, so you may need to use sudo or run the commands as the root user. Additionally, it's important to be cautious when modifying the password policy and ensure that you understand the implications and requirements of your specific environment.

* 1. What is use of “ sudo “

The sudo command is used in Unix-like operating systems to execute commands with elevated privileges. It stands for "Superuser Do" and allows regular users to perform administrative tasks that are typically reserved for the root user.

Here are some key uses of the sudo command:

Administrative tasks: By using sudo, regular users can execute commands with administrative privileges, allowing them to perform tasks such as installing software, modifying system configurations, managing users and groups, and performing system maintenance.

Enhanced security: Instead of constantly logging in as the root user, which can be risky, sudo provides a more secure approach. It enables users to execute specific commands with elevated privileges while still operating within the confines of their regular user account. This helps mitigate the risks associated with unrestricted root access.

Access control: sudo allows system administrators to control which users or groups have permission to execute specific commands as root. The /etc/sudoers configuration file defines the rules and permissions for sudo usage, specifying which users or groups are allowed to execute commands and which commands they can execute.

Logging and auditing: sudo provides a logging mechanism that records user activities when executing commands with elevated privileges. This allows administrators to monitor and audit system changes made by users through sudo, aiding in system administration and security analysis.

To use sudo, simply prefix the desired command with sudo. For example, to install software using apt package manager, you would run:

bash

Copy code

sudo apt install <package-name>

You will be prompted to enter your password to authenticate and confirm your privilege escalation. After successful authentication, the command will be executed with the necessary administrative privileges.

Please note that sudo access is typically granted to trusted users

* 1. Command to reset virtual machine

[11:43 AM, 7/1/2023] Sarvesh: The sudo command is used in Unix-like operating systems to execute commands with elevated privileges. It stands for "Superuser Do" and allows regular users to perform administrative tasks that are typically reserved for the root user.

Here are some key uses of the sudo command:

Administrative tasks: By using sudo, regular users can execute commands with administrative privileges, allowing them to perform tasks such as installing software, modifying system configurations, managing users and groups, and performing system maintenance.

Enhanced security: Instead of constantly logging in as the root user, which can be risky, sudo provides a more secure approach. It enables users to execute specific commands with elevated privileges while still operating within the confines of their regular user account. This helps mitigate the risks associated with unrestricted root access.

Access control: sudo allows system administrators to control which users or groups have permission to execute specific commands as root. The /etc/sudoers configuration file defines the rules and permissions for sudo usage, specifying which users or groups are allowed to execute commands and which commands they can execute.

Logging and auditing: sudo provides a logging mechanism that records user activities when executing commands with elevated privileges. This allows administrators to monitor and audit system changes made by users through sudo, aiding in system administration and security analysis.

To use sudo, simply prefix the desired command with sudo. For example, to install software using apt package manager, you would run:

bash

Copy code

sudo apt install <package-name>

You will be prompted to enter your password to authenticate and confirm your privilege escalation. After successful authentication, the command will be executed with the necessary administrative privileges.

Please note that sudo access is typically granted to trusted users

[11:45 AM, 7/1/2023] Sarvesh: The sudo command is used in Unix-like operating systems to execute commands with elevated privileges. It stands for "Superuser Do" and allows regular users to perform administrative tasks that are typically reserved for the root user.

Here are some key uses of the sudo command:

Administrative tasks: By using sudo, regular users can execute commands with administrative privileges, allowing them to perform tasks such as installing software, modifying system configurations, managing users and groups, and performing system maintenance.

Enhanced security: Instead of constantly logging in as the root user, which can be risky, sudo provides a more secure approach. It enables users to execute specific commands with elevated privileges while still operating within the confines of their regular user account. This helps mitigate the risks associated with unrestricted root access.

Access control: sudo allows system administrators to control which users or groups have permission to execute specific commands as root. The /etc/sudoers configuration file defines the rules and permissions for sudo usage, specifying which users or groups are allowed to execute commands and which commands they can execute.

Logging and auditing: sudo provides a logging mechanism that records user activities when executing commands with elevated privileges. This allows administrators to monitor and audit system changes made by users through sudo, aiding in system administration and security analysis.

To use sudo, simply prefix the desired command with sudo. For example, to install software using apt package manager, you would run:

bash

Copy code

sudo apt install <package-name>

You will be prompted to enter your password to authenticate and confirm your privilege escalation. After successful authentication, the command will be executed with the necessary administrative privileges.

Please note that sudo access is typically granted to trusted users

User

8. Command to reset virtual machine

ChatGPT

The command to reset a virtual machine depends on the specific virtualization software or platform you are using. Here are some common commands to reset virtual machines in popular virtualization environments:

VMware:

For VMware Workstation or VMware Fusion, you can use the menu options to reset a virtual machine. Go to the "Virtual Machine" menu and choose "Power" or "Restart" options.

Alternatively, you can use the vmrun command-line utility to reset the virtual machine. The command syntax is:

bash

Copy code

vmrun reset <path-to-vmx-file>

Replace <path-to-vmx-file> with the full path to the virtual machine configuration file (VMX).

VirtualBox:

For VirtualBox, you can use the VBoxManage command-line tool to reset a virtual machine. The command syntax is:

bash

Copy code

VBoxManage controlvm <vm-name> reset

Replace <vm-name> with the name or UUID of the virtual machine.

Hyper-V:

For Hyper-V on Windows, you can use the PowerShell command Reset-VM to reset a virtual machine. The command syntax is:

powershell

Copy code

Reset-VM -Name <vm-name>

Replace <vm-name> with the name of the virtual machine.

KVM/QEMU:

For KVM/QEMU, you can use the virsh command-line tool to reset a virtual machine. The command syntax is:

bash

Copy code

virsh reset <domain-name>

Replace <domain-name> with the name or ID of the virtual machine domain.

These commands will reset the virtual machine, which typically means that it will be powered off and then started again. Please note that resetting a virtual machine may cause any unsaved data or running processes inside the virtual machine to be lost, so make sure to save your work before performing a reset.

* 1. How to change user and group ownership on same time

To change both user and group ownership of a file or directory at the same time, you can use the chown command with the following syntax:

bash

Copy code

sudo chown <user>:<group> <file or directory>

Replace <user> with the desired username or user ID, and <group> with the desired group name or group ID. <file or directory> should be replaced with the path to the file or directory you want to change ownership for.

For example, if you want to change the ownership of a file named "example.txt" to the user "john" and the group "staff", you would run the following command:

bash

Copy code

sudo chown john:staff example.txt

This command will change the user ownership to "john" and the group ownership to "staff" for the specified file. You can apply the same command to directories as well.

Please note that changing ownership of a file or directory typically requires administrative privileges, so you may need to use sudo or run the command as the root user.

* 1. Command to change user permission on directory

To change user permissions on a directory, you can use the chmod command in Unix-like systems. Here's the command syntax:

bash

Copy code

chmod <permissions> <directory>

Replace <permissions> with the desired permissions you want to assign to the user, and <directory> with the path to the directory you want to modify.

The permissions can be specified using the numeric or symbolic notation. Here are examples of both:

Numeric Notation:

Each permission is represented by a digit:

Read (r): 4

Write (w): 2

Execute (x): 1

The sum of the digits represents the desired permission value.

For example, to give the user read and write permissions (4 + 2 = 6), you would use the following command:

bash

Copy code

chmod 600 <directory>

Symbolic Notation:

The symbolic notation uses letters to represent the permissions.

The basic symbols are:

u: User (owner)

g: Group

o: Others (everyone else)

a: All (user, group, and others)

The available operations are:

+: Add permission

-: Remove permission

=: Set permission explicitly

For example, to give the user read and write permissions and remove all other permissions, you would use the following command:

bash

Copy code

chmod u=rw,go= <directory>

Remember to use sudo or run the command as the root user if you don't have sufficient permissions to change the directory's ownership.

* 1. List of special permission in Linux 7.0 is……

To change user permissions on a directory, you can use the chmod command in Unix-like systems. Here's the command syntax:

bash

Copy code

chmod <permissions> <directory>

Replace <permissions> with the desired permissions you want to assign to the user, and <directory> with the path to the directory you want to modify.

The permissions can be specified using the numeric or symbolic notation. Here are examples of both:

Numeric Notation:

Each permission is represented by a digit:

Read (r): 4

Write (w): 2

Execute (x): 1

The sum of the digits represents the desired permission value.

For example, to give the user read and write permissions (4 + 2 = 6), you would use the following command:

bash

Copy code

chmod 600 <directory>

Symbolic Notation:

The symbolic notation uses letters to represent the permissions.

The basic symbols are:

u: User (owner)

g: Group

o: Others (everyone else)

a: All (user, group, and others)

The available operations are:

+: Add permission

-: Remove permission

=: Set permission explicitly

For example, to give the user read and write permissions and remove all other permissions, you would use the following command:

bash

Copy code

chmod u=rw,go= <directory>

Remember to use sudo or run the command as the root user if you don't have sufficient permissions to change the directory's ownership.

* 1. What happened if i used this command…?[ #chmod u+s /user/bin/vim ]

If you use the command chmod u+s /usr/bin/vim, it sets the "setuid" permission on the Vim executable file located at /usr/bin/vim. Here's what happens when you set the "setuid" permission:

Ownership change: The ownership of the file remains the same, but the effective user ID (EUID) of the process executing the file will be temporarily changed to the owner of the file, in this case, the user who owns the Vim executable.

Elevated privileges: When a user executes the Vim binary with the "setuid" permission, the process runs with the privileges of the owner of the file (in this case, the owner of the Vim executable), rather than the privileges of the user who is executing it.

In the context of the command you provided, setting the "setuid" permission on the Vim executable can have the following implications:

Elevated privileges: When a regular user executes /usr/bin/vim, it will run with elevated privileges, potentially allowing the user to perform actions they wouldn't normally be able to do. This can be useful for certain system administration tasks where elevated privileges are required.

Security concerns: Setting the "setuid" permission should be done with caution, as it can introduce security risks. If there are any vulnerabilities or malicious code in the Vim executable or any plugins it loads, running Vim with elevated privileges could potentially lead to unauthorized access or system compromise. Therefore, it's important to ensure that the Vim executable is secure and up to date.

Best practices: It is generally recommended to be cautious when setting the "setuid" permission on executables and to only do so when absolutely necessary. Careful consideration should be given to the potential security implications and alternative solutions should be explored whenever possible.

Please note that modifying system files and setting the "setuid" permission often requires administrative privileges, so you may need to use sudo or run the command as the root user.

* 1. What happened if i used this command…. [ #chmod g+s /data ]

If you use the command chmod g+s /data, it sets the "setgid" permission on the /data directory. Here's what happens when you set the "setgid" permission:

Group ownership inheritance: When the "setgid" permission is set on a directory, newly created files and directories within that directory inherit the group ownership of the parent directory, rather than the group ownership of the user who created them.

Group collaboration: Setting the "setgid" permission on a directory can facilitate group collaboration by ensuring that all files and directories created within that directory belong to the same group. This can help maintain consistent group access permissions and simplify group-based file sharing.

In the context of the command you provided, setting the "setgid" permission on the /data directory can have the following implications:

Inherited group ownership: If a user creates a file or directory within the /data directory, it will automatically be assigned the group ownership of the /data directory, regardless of the user's own group membership.

Group collaboration: This can be useful in scenarios where multiple users or applications need to access and collaborate on files within the /data directory. By setting the "setgid" permission, you ensure that all new files and directories created within /data maintain the same group ownership, allowing users belonging to that group to access and modify them.

Subdirectory permissions: It's important to note that the "setgid" permission on a directory only affects newly created files and directories within that directory. It does not automatically change the permissions of existing files and directories within /data.

Please keep in mind that modifying system files and directories often requires administrative privileges, so you may need to use sudo or run the command as the root user. Additionally, it's important to carefully consider the implications of changing permissions and to ensure that it aligns with your intended use case and security requirements.

* Task: 1
  1. Find details about current logged-in user.

To find details about the current logged-in user in a Unix-like system, you can use the whoami and id commands. Here's how:

whoami: This command displays the username of the current user.

bash

Copy code

whoami

Example output:

Copy code

john

id: This command provides more detailed information about the current user, including user ID (UID), group ID (GID), and group membership.

bash

Copy code

id

Example output:

scss

Copy code

uid=1000(john) gid=1000(john) groups=1000(john),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),116(lpadmin),126(sambashare)

In the output, "uid" refers to the user ID, "gid" refers to the primary group ID, and "groups" lists the additional groups the user belongs to.

These commands provide basic information about the current logged-in user, including the username, user ID, group ID, and group membership.

* 1. Show all processes on terminal

To display all processes running on the terminal, you can use the ps command with various options. Here are a few commonly used options:

ps: This command, when used without any options, displays a snapshot of the currently running processes owned by the current user.

bash

Copy code

ps

ps -e: This option displays information about all processes on the system.

bash

Copy code

ps -e

ps -ef: This option provides a detailed view of all processes, including the process hierarchy (parent-child relationship) and the associated command-line arguments.

bash

Copy code

ps -ef

ps aux: This option is similar to ps -ef but provides a more concise output.

bash

Copy code

ps aux

These commands will list the processes running on the terminal along with their respective process IDs (PIDs), parent process IDs (PPIDs), CPU and memory usage, and other information. The exact output format may vary slightly depending on the operating system and the version of the ps command available.

You can further customize the output by combining ps with additional options or using tools like grep to filter specific processes based on criteria such as process names, user IDs, or other attributes.

* 1. Create primary group

To create a primary group in a Unix-like system, you can use the groupadd command. Here's the syntax:

bash

Copy code

sudo groupadd <groupname>

Replace <groupname> with the desired name for the primary group.

For example, to create a primary group named "staff", you would run the following command:

bash

Copy code

sudo groupadd staff

The groupadd command will create a new entry for the primary group in the system's group database, typically located in the /etc/group file. It will assign a unique group ID (GID) to the group and make the necessary system configurations.

Please note that creating a group typically requires administrative privileges, so you may need to use sudo or run the command as the root user.

* 1. Create supplementary group

To create a supplementary group (also known as a secondary group) in a Unix-like system, you can use the groupadd command with the -g option to specify the group ID (GID). Here's the syntax:

bash

Copy code

sudo groupadd -g <gid> <groupname>

Replace <gid> with the desired group ID and <groupname> with the name of the supplementary group.

For example, to create a supplementary group named "developers" with a GID of 1001, you would run the following command:

bash

Copy code

sudo groupadd -g 1001 developers

The groupadd command will create a new entry for the supplementary group in the system's group database, typically located in the /etc/group file. It will assign the specified GID to the group and make the necessary system configurations.

Please note that creating a group typically requires administrative privileges, so you may need to use sudo or run the command as the root user. Additionally, it's important to choose a unique GID that is not already in use by another group on your system.

* 1. Find groups details and list on terminal P6. Find user details and list on terminal.

To find group details and list them on the terminal, you can use the cat or less command to view the contents of the /etc/group file. Here's the command:

bash

Copy code

cat /etc/group

or

bash

Copy code

less /etc/group

This command will display the contents of the /etc/group file, which contains information about all the groups on the system. Each line represents a separate group and includes details such as the group name, group password (represented by "x"), group ID (GID), and a comma-separated list of users who are members of the group.

To find user details and list them on the terminal, you can use the cat or less command to view the contents of the /etc/passwd file. Here's the command:

bash

Copy code

cat /etc/passwd

or

bash

Copy code

less /etc/passwd

This command will display the contents of the /etc/passwd file, which contains information about all the users on the system. Each line represents a separate user and includes details such as the username, user password (represented by "x"), user ID (UID), primary group ID (GID), user's full name, home directory, and default shell.

Please note that viewing the contents of these system files typically requires administrative privileges, so you may need to use sudo or run the command as the root user. Additionally, be cautious when viewing or manipulating system files to avoid accidentally modifying critical configuration data.

* 1. Use “sudo”

The sudo command allows users with the appropriate permissions to execute commands as the root user or another user. It provides a way to perform administrative tasks or access restricted resources on a Unix-like system. Here are a few examples of how to use sudo:

Execute a single command with root privileges:

bash

Copy code

sudo command

Replace command with the actual command you want to run with elevated privileges. For example, sudo apt update will update the system's package repositories.

Open an interactive root shell:

bash

Copy code

sudo -i

This command will start an interactive shell session as the root user, allowing you to execute multiple commands with root privileges.

Run a specific command as another user:

bash

Copy code

sudo -u username command

Replace username with the desired username and command with the command you want to run as that user. For example, sudo -u john ls /home/john will list the contents of the home directory for the user "john."

When using sudo, you may be prompted to enter your password to verify your identity and authorization to perform the requested action. Make sure to use sudo responsibly and only execute commands that you fully understand and have a legitimate need for.

* 1. View the last 5 lines of the “ /var/log/messages “

To view the last 5 lines of the /var/log/messages file, you can use the tail command with the -n option. Here's the command:

bash

Copy code

tail -n 5 /var/log/messages

This command will display the last 5 lines of the /var/log/messages file, which is a common system log file that contains various system messages and events.

If you don't have sufficient permissions to view the log file, you may need to run the command with sudo:

bash

Copy code

sudo tail -n 5 /var/log/messages

Please note that the location and naming of log files can vary depending on the specific distribution and configuration of your system. Additionally, the file /var/log/messages may not exist on all systems. If the file doesn't exist, you can try viewing other log files such as /var/log/syslog or /var/log/messages.log depending on your system.

* 1. Add a new user with name “ NuPuR “

To add a new user with the name "NuPuR" in a Unix-like system, you can use the useradd command. Here's the syntax:

bash

Copy code

sudo useradd <username>

Replace <username> with the desired name for the new user. In this case, you would run the following command:

bash

Copy code

sudo useradd NuPuR

By default, the useradd command will create a new user with the specified username, assign a unique user ID (UID) to the user, and create a home directory with the same name as the username in the /home directory.

However, this basic command may not set up all the necessary user configurations, such as setting a password or creating a primary group. To create a user with additional options, such as specifying a home directory or assigning a primary group, you can use additional options with the useradd command.

For example, to create a user named "NuPuR" with a specific home directory and primary group, you can use the following command:

bash

Copy code

sudo useradd -d /path/to/home -g groupname NuPuR

Replace /path/to/home with the desired home directory path and groupname with the name of the primary group.

Please note that creating a user typically requires administrative privileges, so you may need to use sudo or run the command as the root user.

* + 1. Remove this user and user’s home directory
  1. Create new supplementary group name is “ whEEL “

To create a new supplementary group named "whEEL" in a Unix-like system, you can use the groupadd command. Here's the command:

bash

Copy code

sudo groupadd whEEL

This command will create a new entry for the supplementary group "whEEL" in the system's group database, typically located in the /etc/group file. The group will be assigned a unique group ID (GID) and will initially have no members.

Please note that creating a group typically requires administrative privileges, so you may need to use sudo or run the command as the root user. Additionally, it's important to choose a unique group name that is not already in use by another group on your system.

* 1. Create a new user with name “ ELviS “

To create a new user with the name "ELviS" in a Unix-like system, you can use the useradd command. Here's the syntax:

bash

Copy code

sudo useradd <username>

Replace <username> with the desired name for the new user. In this case, you would run the following command:

bash

Copy code

sudo useradd ELviS

By default, the useradd command will create a new user with the specified username, assign a unique user ID (UID) to the user, and create a home directory with the same name as the username in the /home directory.

However, this basic command may not set up all the necessary user configurations, such as setting a password or creating a primary group. To create a user with additional options, you can use additional command-line options with the useradd command.

For example, to create a user named "ELviS" with a specific home directory and primary group, you can use the following command:

bash

Copy code

sudo useradd -d /path/to/home -g groupname ELviS

Replace /path/to/home with the desired home directory path and groupname with the name of the primary group.

Please note that creating a user typically requires administrative privileges, so you may need to use sudo or run the command as the root user.

* 1. Add / Append a user to a supplementary group

To add or append a user to a supplementary group in a Unix-like system, you can use the usermod command with the -aG option. Here's the command:

bash

Copy code

sudo usermod -aG <groupname> <username>

Replace <groupname> with the name of the supplementary group and <username> with the username of the user you want to add to the group. For example, to add the user "ELviS" to the supplementary group "whEEL", you would run the following command:

bash

Copy code

sudo usermod -aG whEEL ELviS

The -aG option ensures that the user is appended to the specified supplementary group without removing them from their existing groups. This command will modify the user's group membership in the system's user database, typically located in the /etc/group file.

Please note that modifying user group membership typically requires administrative privileges, so you may need to use sudo or run the command as the root user. Additionally, the changes may take effect upon the user's next login or session initialization.

* 1. Restrict / Lock login access for “ ELviS “ user

To add or append a user to a supplementary group in a Unix-like system, you can use the usermod command with the -aG option. Here's the command:

bash

Copy code

sudo usermod -aG <groupname> <username>

Replace <groupname> with the name of the supplementary group and <username> with the username of the user you want to add to the group. For example, to add the user "ELviS" to the supplementary group "whEEL", you would run the following command:

bash

Copy code

sudo usermod -aG whEEL ELviS

The -aG option ensures that the user is appended to the specified supplementary group without removing them from their existing groups. This command will modify the user's group membership in the system's user database, typically located in the /etc/group file.

Please note that modifying user group membership typically requires administrative privileges, so you may need to use sudo or run the command as the root user. Additionally, the changes may take effect upon the user's next login or session initialization.

* 1. Create a new user name “ LiNuX without home directory

To create a new user named "LiNuX" without a home directory in a Unix-like system, you can use the useradd command with the -M option. Here's the command:

bash

Copy code

sudo useradd -M LiNuX

The -M option instructs the useradd command to not create a home directory for the user. This is useful when you want to create a user without providing them with a home directory.

By default, the useradd command will assign a unique user ID (UID) to the user and create an entry in the system's user database, typically located in the /etc/passwd file. However, since the home directory is not created, the user won't have a specific location to store files and configurations.

Please note that creating a user typically requires administrative privileges, so you may need to use sudo or run the command as the root user.

* 1. Create a new user name “ RedHat “ with new home directory “

To create a new user named "RedHat" with a new home directory in a Unix-like system, you can use the useradd command with the -d option. Here's the command:

bash

Copy code

sudo useradd -d /path/to/new/home RedHat

Replace /path/to/new/home with the desired path for the new home directory. For example, if you want the new home directory to be /home/RedHat, you would run the following command:

bash

Copy code

sudo useradd -d /home/RedHat RedHat

The -d option allows you to specify the new home directory for the user. By default, the useradd command will create the home directory if it doesn't already exist.

Please note that creating a user typically requires administrative privileges, so you may need to use sudo or run the command as the root user. Additionally, make sure to choose a valid and unused path for the new home directory.

* 1. /etc/HatRed

The path /etc/HatRed you mentioned, /etc/HatRed, does not correspond to a standard configuration file or directory in Unix-like systems. The typical convention for system configuration files in the /etc directory does not include a file or directory named "HatRed".

It's possible that you may have made a typo or provided a non-standard file path. Please double-check the path or provide more context if you are referring to a specific file or directory.

* 1. Create a new user with two(2) days expiry

To create a new user with an account expiration date set to two (2) days in the future in a Unix-like system, you can use the useradd command with the -e option. Here's the command:

bash

Copy code

sudo useradd -e $(date -d "+2 days" +%Y-%m-%d) <username>

Replace <username> with the desired name for the new user. For example, if you want to create a user named "John" with a two-day account expiration, you would run the following command:

bash

Copy code

sudo useradd -e $(date -d "+2 days" +%Y-%m-%d) John

The -e option allows you to specify the account expiration date for the user. The command $(date -d "+2 days" +%Y-%m-%d) calculates the date two days in the future and formats it as YYYY-MM-DD for the useradd command.

Please note that setting an account expiration date may require administrative privileges, so you may need to use sudo or run the command as the root user. Additionally, the actual account expiration behavior may depend on the system configuration and any automated processes that handle account expiration.

* 1. Remove password for “ ELviS “ user

To remove the password for the user "ELviS" in a Unix-like system, you can use the passwd command with the -d option. Here's the command:

bash

Copy code

sudo passwd -d ELviS

This command will remove the password for the specified user, allowing them to log in without entering a password. Please note that removing the password for a user is generally not recommended for security reasons unless there are specific requirements or use cases.

Please note that modifying user settings typically requires administrative privileges, so you may need to use sudo or run the command as the root user. Additionally, this command will only remove the user's password and will not disable their account or make any other changes..

* 1. Check user password policy for “ LiNuX “ user

To check the password policy for the user "LiNuX" in a Unix-like system, you can use the passwd command with the -S option followed by the username. Here's the command:

bash

Copy code

sudo passwd -S LiNuX

This command will display the password status and password aging information for the specified user. The output will include details such as the password status (e.g., "L" for locked or "P" for a valid password), the date of the last password change, the minimum password age, the maximum password age, and other related information.

Please note that checking user password policy typically requires administrative privileges, so you may need to use sudo or run the command as the root user. The specific output format and available information may vary depending on the operating system and its configuration.

* Task :2
  1. Login from “LiNuX” user

To log in as a specific Linux user, you can follow these steps:

Open a terminal or shell session on your Linux system. This can usually be done by searching for "Terminal" or "Shell" in the applications menu or by using a keyboard shortcut like Ctrl+Alt+T.

Once you have the terminal open, you can log in as a specific user using the su or sudo command, depending on your requirements.

To log in as a different user with their environment and settings intact, you can use the su command followed by the username. For example:

bash

Copy code

su <username>

Replace <username> with the username of the user you want to log in as. You will be prompted to enter the password for that user.

To execute a specific command or perform administrative tasks as another user, you can use the sudo command followed by the command you want to run. For example:

bash

Copy code

sudo <command>

Replace <command> with the command you want to execute as the specified user. You will be prompted to enter your own password, as long as you have sudo privileges, and the command will be executed with the permissions of the specified user.

Please note that logging in as a different user typically requires administrative privileges or knowledge of the password for that user. It's important to use these commands responsibly and only perform actions that you are authorized to do.

* 1. Create new directory on desktop name is "FoLdEr”

To log in as a specific Linux user, you can follow these steps:

Open a terminal or shell session on your Linux system. This can usually be done by searching for "Terminal" or "Shell" in the applications menu or by using a keyboard shortcut like Ctrl+Alt+T.

Once you have the terminal open, you can log in as a specific user using the su or sudo command, depending on your requirements.

To log in as a different user with their environment and settings intact, you can use the su command followed by the username. For example:

bash

Copy code

su <username>

Replace <username> with the username of the user you want to log in as. You will be prompted to enter the password for that user.

To execute a specific command or perform administrative tasks as another user, you can use the sudo command followed by the command you want to run. For example:

bash

Copy code

sudo <command>

Replace <command> with the command you want to execute as the specified user. You will be prompted to enter your own password, as long as you have sudo privileges, and the command will be executed with the permissions of the specified user.

Please note that logging in as a different user typically requires administrative privileges or knowledge of the password for that user. It's important to use these commands responsibly and only perform actions that you are authorized to do.

User

2. Create new directory on desktop name is "FoLdEr”

ChatGPT

To create a new directory named "FoLdEr" on the desktop in a Linux system, you can use the mkdir command. Here's the command:

bash

Copy code

mkdir ~/Desktop/FoLdEr

This command will create a new directory named "FoLdEr" within your user's home directory. The ~/Desktop portion of the command specifies the path to the desktop directory.

After running the command, you should see the "FoLdEr" directory appear on your desktop. You can then use this directory to store files or organize your data.

Please note that the actual path to the desktop directory may vary depending on your Linux distribution and desktop environment. The ~/Desktop path is a common location for the desktop directory, but it may be different in your specific setup.

* 1. Change group ownership from LiNuX to root on “FoLdEr” directory

To change the group ownership of the "FoLdEr" directory from "LiNuX" to "root" in a Linux system, you can use the chown command with the :groupname syntax. Here's the command:

bash

Copy code

sudo chown :root ~/Desktop/FoLdEr

This command will change the group ownership of the "FoLdEr" directory to the "root" group. The ~/Desktop/FoLdEr portion of the command specifies the path to the directory.

After running the command, the group ownership of the "FoLdEr" directory should be updated to "root". You can verify the ownership using the ls -l command, which displays detailed file and directory information.

Please note that changing group ownership typically requires administrative privileges, so you may need to use sudo or run the command as the root user. Additionally, make sure to provide the correct group name, which is "root" in this case.

* 1. Create new file on /etc/ with name “ FiLe”

Creating a new file named "FiLe" in the "/etc/" directory requires administrative privileges. Please ensure that you have the necessary permissions or use the appropriate commands with elevated privileges, such as sudo. Here's the command:

bash

Copy code

sudo touch /etc/FiLe

This command uses the touch command to create a new empty file named "FiLe" in the "/etc/" directory. The sudo command is used to run the command with root privileges, as modifying files in the "/etc/" directory typically requires administrative access.

Please exercise caution when modifying files in system directories like "/etc/", as improper changes can impact the functioning of your Linux system. Make sure to follow best practices and only make changes when necessary and with a clear understanding of the potential consequences.

* 1. Check permissions of above file

To check the permissions of a file in Linux, you can use the ls command with the -l option to display detailed file information. Here's the command to check the permissions of the "FiLe" file in the "/etc/" directory:

bash

Copy code

ls -l /etc/FiLe

The output will display information about the file, including the permissions. The permissions are represented by a series of characters that indicate read (r), write (w), and execute (x) permissions for the owner, group, and others. Here's an example of how the output may appear:

bash

Copy code

-rw-r--r-- 1 root root 0 Jul 1 10:00 /etc/FiLe

In the example above, the file permissions are -rw-r--r--, where the first character - represents a regular file, rw- represents the owner's permissions (read and write), r-- represents the group's permissions (read only), and r-- represents the permissions for others (read only).

Please note that checking permissions in certain system directories like "/etc/" may require administrative privileges, so you may need to use sudo or run the command as the root user.

* 1. Login from “student” user

To log in as the "student" user, you can use the su command followed by the username. Here's the command:

bash

Copy code

su student

After running the command, you will be prompted to enter the password for the "student" user. Once you enter the correct password, you will be logged in as the "student" user.

Please note that logging in as another user typically requires administrative privileges or knowledge of the user's password. If you don't have the necessary permissions or the correct password, you may not be able to log in as the "student" user.

* 1. Create new directory on students home with name “file1”

To create a new directory named "file1" in the home directory of the "student" user, you can use the following command:

bash

Copy code

mkdir /home/student/file1

This command creates a new directory named "file1" within the home directory of the "student" user, which is typically located at "/home/student". After running the command, the "file1" directory will be created inside the "student" user's home directory.

Please note that the exact location of the home directory may vary depending on your Linux distribution and user configuration. Make sure you have the necessary permissions to create directories in the specified location.

* 1. Remove read and write permission for group and other on above file “file1”

To remove the read and write permissions for the group and other users on the "file1" directory in the home directory of the "student" user, you can use the chmod command with the appropriate permission settings. Here's the command:

bash

Copy code

chmod go-rw /home/student/file1

This command removes the read and write permissions for the group (g) and other (o) users on the "file1" directory. The /home/student/file1 portion of the command specifies the path to the directory.

After running the command, the group and other users will no longer have read and write permissions on the "file1" directory. You can verify the permissions using the ls -l command, which displays detailed file and directory information.

Please note that changing permissions on directories typically requires administrative privileges, so you may need to use sudo or run the command as the root user. Additionally, be cautious when modifying permissions, as improper changes can impact the access and functionality of the directory.

* 1. Add execute permission for everyone on “file2”

To add execute permission for everyone on the file named "file2," you can use the chmod command with the +x option. Here's the command:

bash

Copy code

chmod +x /path/to/file2

Replace /path/to/file2 with the actual path to the file "file2" that you want to modify.

After running the command, the execute permission will be added for everyone on the specified file. You can verify the permissions using the ls -l command, which displays detailed file information.

Please note that changing permissions on files typically requires administrative privileges, so you may need to use sudo or run the command as the root user. Additionally, be cautious when modifying permissions, as incorrect changes can affect the security and functionality of the file.

* 1. Set Read,write,execute for USER

To set read, write, and execute permissions for the user on a file or directory, you can use the chmod command with the appropriate permission settings. Here's the command:

bash

Copy code

chmod u+rwx /path/to/file\_or\_directory

Replace /path/to/file\_or\_directory with the actual path to the file or directory that you want to modify.

After running the command, the user will have read, write, and execute permissions on the specified file or directory. You can verify the permissions using the ls -l command, which displays detailed file information.

Please note that changing permissions on files or directories typically requires administrative privileges, so you may need to use sudo or run the command as the root user. Additionally, be cautious when modifying permissions, as incorrect changes can affect the security and functionality of the file or directory.

* 1. Set Read and execute for GROUP

To set read and execute permissions for the group on a file or directory, you can use the chmod command with the appropriate permission settings. Here's the command:

bash

Copy code

chmod g+rx /path/to/file\_or\_directory

Replace /path/to/file\_or\_directory with the actual path to the file or directory that you want to modify.

After running the command, the group will have read and execute permissions on the specified file or directory. You can verify the permissions using the ls -l command, which displays detailed file information.

Please note that changing permissions on files or directories typically requires administrative privileges, so you may need to use sudo or run the command as the root user. Additionally, be cautious when modifying permissions, as incorrect changes can affect the security and functionality of the file or directory.

* 1. Set No permission for other on “Directory1”

To set no permissions for the "other" users on a directory named "Directory1", you can use the chmod command with the appropriate permission settings. Here's the command:

bash

Copy code

chmod o= /path/to/Directory1

Replace /path/to/Directory1 with the actual path to the directory that you want to modify.

After running the command, the "other" users will have no permissions on the specified directory. You can verify the permissions using the ls -l command, which displays detailed file and directory information.

Please note that changing permissions on directories typically requires administrative privileges, so you may need to use sudo or run the command as the root user. Additionally, be cautious when modifying permissions, as incorrect changes can affect the security and functionality of the directory.

* 1. Create new group name “ateam” , And add two new user in this group

“andy” and “alice”, set password is “password”

To create a new group named "ateam" and add two new users, "andy" and "alice", to this group with the password "password", you can follow these steps:

Create the new group "ateam" using the groupadd command:

bash

Copy code

sudo groupadd ateam

Create the new users "andy" and "alice" using the useradd command, specifying the group as "ateam" and setting the password for each user:

bash

Copy code

sudo useradd -G ateam -m andy

sudo useradd -G ateam -m alice

The -G option is used to specify the supplementary group(s) for the user. The -m option is used to create the home directory for each user.

Set the password "password" for each user using the passwd command:

bash

Copy code

sudo passwd andy

sudo passwd alice

You will be prompted to enter and confirm the new password for each user.

After following these steps, the "ateam" group will be created, and the users "andy" and "alice" will be added to this group with the specified password.

Please note that creating groups and users typically requires administrative privileges, so you may need to use sudo or run the commands as the root user. Additionally, it's important to choose secure and unique passwords for your users.

* 1. Login from root and root home directory

To log in as the root user and access the root user's home directory, you can follow these steps:

Open a terminal or command prompt.

Switch to the root user by running the following command and providing the root user's password when prompted:

bash

Copy code

sudo su

This command elevates your privileges to the root user.

After successfully logging in as the root user, you will be in the root user's home directory, typically located at /root.

Please note that logging in as the root user is typically reserved for administrative tasks and should be used with caution. It's recommended to only log in as the root user when necessary and to log out and return to a regular user account for day-to-day tasks.

* 1. Create a new directory in “/home” name is “ateam-text”

To create a new directory named "ateam-text" in the "/home" directory, you can use the following command:

bash

Copy code

sudo mkdir /home/ateam-text

This command will create a new directory called "ateam-text" within the "/home" directory. The sudo command is used to run the command with administrative privileges since creating a directory in the "/home" directory typically requires elevated permissions.

After running the command, you can verify that the directory was created by using the ls command:

bash

Copy code

ls /home

You should see the "ateam-text" directory listed among the other directories in "/home".

* 1. Change the group ownership of the ateam-text directory to “ateam”.

To change the group ownership of the "ateam-text" directory to the group "ateam", you can use the following command:

bash

Copy code

sudo chown :ateam /home/ateam-text

This command uses the chown command to change the group ownership of the directory. The : followed by the group name specifies the new group ownership. In this case, it sets the group ownership to "ateam". The directory path /home/ateam-text should be replaced with the actual path to the "ateam-text" directory.

After running the command, you can verify the group ownership of the directory using the ls -l command:

bash

Copy code

ls -l /home

The output should display the updated group ownership for the "ateam-text" directory as "ateam".

* 1. Ensure the permission of ateam-text allows group members to create

To ensure that group members have appropriate permissions on the "ateam-text" directory, you can set the group permissions to allow read, write, and execute access. Here's the command:

bash

Copy code

sudo chmod g+rwx /home/ateam-text

This command uses the chmod command to modify the permissions of the directory. The g+rwx part specifies that the group should have read, write, and execute permissions.

After running the command, you can verify the permissions of the directory using the ls -l command:

bash

Copy code

ls -l /home

The output should display the updated permissions for the "ateam-text" directory, with the group having read, write, and execute permissions.

Please note that changing permissions on directories in system directories like "/home" typically requires administrative privileges, hence the use of sudo.