Assignment-2 Linux

1.In Linux FHS (Filesystem Hierarchy Standard) what is the /?

It is the root directory of the file system. All other directories and files in the system are organized under the root directory

2. What is stored in each of the following paths?

/bin,/sbin, /usr/bin and /usr/sbin
/etc
/home
/var
/tmp

/bin - This directory contains executable files (binaries) that are essential to the system's operation. These files are usually required during the boot process, and are accessible to all users of the system.
/sbin - This directory contains executable files (binaries) that are also essential to the system's operation, but are usually used only by the system administrator or during system maintenance tasks. These files are usually not accessible to regular users of the system.
/usr/bin - This directory contains executable files (binaries) that are not essential to the system's operation, but are used by most users. These files are usually installed as part of user applications or packages, and are accessible to all users of the system.
/usr/sbin - This directory contains executable files (binaries) that are similar to those in /sbin, but are not essential to the system's operation. These files are usually used by the system administrator or during system maintenance tasks, and are not accessible to regular users of the system.
/etc - This directory contains configuration files for the system and applications. These files are usually editable by the system administrator only.
/home - This directory contains home directories for each user on the system. Each user has their own subdirectory in /home, where they can store their personal files and settings.
/var - This directory contains variable files that can change in size or content as the system is running. This includes log files, temporary files, and other files that are generated by applications or the system.
/tmp - This directory contains temporary files that are created by applications or the system. The contents of this directory are deleted when the system is rebooted, and it is accessible to all users of the system.

3. What is special about the /tmp directory when compared to other directories?

The "/tmp" directory is a temporary directory that is typically used for storing temporary files by various programs and processes. One thing that is special about the "/tmp" directory when compared to other directories is that its contents are usually not preserved across system reboots. In other words, any files or directories created in the "/tmp" directory are typically deleted automatically when the system is restarted or shut down

4. What kind of information one can find in /proc?

The "/proc" directory in Linux is a special virtual file system that provides information about various system resources and processes such as Process information, System information, Kernel information and Hardware information.

5. What makes /proc different from other filesystems?

/proc is a virtual filesystem created in memory by the kernel .It provides a way for user-level programs to access and modify kernel data structures. Most files in /proc are read-only to prevent system instability. The majority of files in /proc are text files that contain information about the system or kernel.

6. True or False? only root can create files in /proc

False

7. What can be found in /proc/cmdline?

The "/proc/cmdline" file contains the command line arguments that were passed to the Linux kernel at boot time, including boot loader parameters, kernel parameters, and debugging information. It is a text file that can be read by any user on the system.

8. In which path can you find the system devices (e.g. block storage)?

The system devices such as block storage devices in Linux can be found in the "/dev" directory. This directory contains special files that represent various hardware devices and system resources. Block storage devices are typically represented by files in the "/dev" directory that begin with "sd". The files in the "/dev" directory are used by various system utilities and applications to access and manipulate hardware devices. The "/dev" directory is an important part of the Linux filesystem that provides access to hardware devices and system resources

Permissions

9. How to change the permissions of a file?

In Linux, you can change the permissions of a file using the "chmod" command in the terminal. To start, open a terminal window and navigate to the directory where the file is located. Once you are in the correct directory,

you can use the "ls" command to check the current permissions of the file you want to modify. The "chmod" command can then be used to set the desired permissions for the file. This command requires two arguments: the permissions you want to set, and the name of the file you want to modify. For example, you can use the command "chmod 755 filename" to give the owner of the file read, write, and execute permissions, and give the group and all other users read and execute permissions. In this case, the number "755" represents the permissions for the owner, group, and others in octal format. After running the "chmod" command, use the "ls" command again to check that the permissions of the file have been updated. In summary, the "chmod" command is an essential tool for managing file security and access control in Linux.

10. What does the following permissions mean?:
777
644
750
The following permissions represent the access rights for different types of users on a file or directory:

- 777: This permission set grants full access to the file or directory for all users, including the owner, group, and others. Specifically, it grants read, write, and execute permissions for all three types of users.
- 644: This permission set grants read and write permissions to the owner of the file, and read-only permissions to the group and others. Specifically, the owner has read and write permissions, while the group and others have read-only permissions.
- 750: This permission set grants read, write, and execute permissions to the owner of the file, read and execute permissions to the group, and no permissions to others. Specifically, the owner has full access, the group can read and execute the file, and others have no permissions to the file.

Overall, understanding file permissions in Linux is important for managing file security and access control.

11.What this command does? chmod +x some_file
The command "chmod +x some_file" sets the executable permission on the file named "some_file". The "+x" option is used to add the executable permission for the owner of the file. This allows the owner to run the file as a program or script. By making a file executable, you can execute it directly from the command line or from a script without having to specify the interpreter. It is important to note that this command only modifies the permissions for the owner of the file, and does not affect the group or other users. If you want to add executable permission for other users, you would need to specify the appropriate permission code or use the symbolic method to modify the permissions. Overall, understanding how to modify file permissions is an important aspect of managing file security and access control in Linux.

12. Explain what is setgid and setuid

Setgid and Setuid are special permissions in Linux that allow users to run programs with elevated permissions. Setgid, or "set group ID", sets the group ownership of a file or directory to the group that owns the parent directory, making it easier for multiple users to collaborate on shared files. Setuid, or "set user ID", allows a user to run a program with the permissions of the file's owner, typically used for programs that require elevated privileges to run. However, both setgid and setuid can pose a security risk if not used properly, and should be used with caution. It is important to understand how to properly set and manage these permissions to maintain system security.

13. What is the purpose of sticky bit?

The purpose of the sticky bit is to restrict the deletion of files within a directory. When the sticky bit is set on a directory, only the owner of a file, the owner of the directory, or the root user can delete the file. This can be useful in shared environments where multiple users have write access to the same directory, to prevent accidental deletion of files by other users.
The sticky bit is set using the chmodcommand with the +toption, like this:
chmod +t directory_nam

14. What the following commands do?

chmod
chown
Chgrp
The following commands are used for managing file permissions and ownership in Linux:

- chmod: Short for "change mode", chmod is used to change the permissions of a file or directory. It allows the user to add or remove read, write, and execute permissions for the owner, group, and others. The command is typically used in combination with octal or symbolic permissions to specify the new permissions for the file or directory.
- chown: Short for "change owner", chown is used to change the owner of a file or directory. It allows the user to specify a new user or group as the owner of the file or directory. The command is typically used in combination with the -R option to recursively change the ownership of all files and directories within a directory.
- chgrp: Short for "change group", chgrp is used to change the group ownership of a file or directory. It allows the user to specify a new group as the owner of the file or directory. The command is typically used in combination with the -R option to recursively change the group ownership of all files and directories within a directory.

Overall, these commands are essential for managing file permissions and ownership in Linux, and are often used by system administrators and users to maintain system security and control access to files and directories

15. What is sudo? How do you set it up?
Sudo is a command-line utility in Linux that allows users to run commands with administrative or root privileges without logging in as the root user. This is an important tool for managing system security and control access to sensitive files and commands. To set up sudo, you need to install it using a package manager like apt-get and add the user to the sudoers file, which is located at "/etc/sudoers". This file controls which users can use sudo and what commands they can execute with it. Once sudo is set up, users can run commands with elevated privileges by typing "sudo" before the command. For example, "sudo apt-get install <package_name>" would install software with elevated privileges. Overall, sudo is an essential tool in Linux for managing system security and ensuring that users have access to the tools they need to perform their tasks without compromising system stability.

16. True or False? In order to install packages on the system one must be the root user or use the sudo command
True. In order to install packages on the system, one must either be logged in as the root user or use the sudocommand to execute the installation command with superuser privileges

17. Explain what are ACLs. For what use cases would you recommend to use them?
ACLs, or Access Control Lists, are a way of defining more granular permissions for files and directories in Linux. They allow for permissions to be set for specific users and groups on a file or directory, while still preserving the traditional read, write, and execute permissions for the owner, group, and others. This can be helpful in scenarios where you need to grant or restrict access to specific files or directories for certain users or groups, without affecting the permissions of others. ACLs can add complexity to the permission system, so it's important to use them judiciously and only in cases where they're really needed.

18. You try to create a file but it fails. Name at least three different reason as to why it could happen
There could be several reasons why creating a file could fail. Here are three possible reasons:

- Permission issues: The user attempting to create the file may not have the necessary permissions to create files in the directory or location they are attempting to create it in. This could be due to file permissions, ownership issues, or access control lists (ACLs) that restrict access.
- Disk space issues: If there is not enough free space on the disk, attempts to create a file may fail. This can happen if the disk is full or if quotas are in place that limit the amount of space available to a particular user or group.

- File already exists: If a file with the same name already exists in the directory, attempts to create a new file with the same name may fail. In this case, the user may need to choose a different name or delete the existing file first.

There may be other reasons for file creation failures as well, such as disk errors or file system corruption, but the above reasons are some of the most common.

19. A user accidentally executed the following chmod -x $(which chmod).How to fix it?

When a user accidentally executes "chmod -x $(which chmod)" command, it removes the execute permission from the chmod binary file, making it impossible to execute any chmod command in the future. This is a serious issue because chmod is an essential tool for managing file permissions on Linux systems. To fix this issue, one can reboot the system into a single-user mode or use a live CD to access the root file system in read-write mode. Then, navigate to the directory where the chmod binary is located and grant execute permissions to the chmod binary using the command "chmod +x chmod". This will restore the execute permission to the chmod binary and allow users to execute chmod commands again.

Alternatively, if there is another user with sudo privileges, they can use the "sudo chmod +x /bin/chmod" command to grant execute permissions to the chmod binary file. It is important to be careful when modifying file permissions as it can have unintended consequences on the system. In general, it is recommended to create a backup of important system files before making any changes to file permissions or ownership.

Scenarios

20. You would like to copy a file to a remote Linux host. How would you do?

There are several ways to copy a file to a remote Linux host. One common way is to use the scp(secure copy) command, which uses SSH to securely copyfiles between hosts. Here's an example command to copy a file called file.txtfrom the local host to a remote host at IP address 192.0.2.1:scp file.txt user@192.0.2.1:/path/to/destinationIn this command, useris the username you use to log into the remote host, and /path/to/destinationis the path on the remote host where you want to copy the file to. You will be prompted to enter the password for the remote user account to complete the copy operation.21.

21.How to generate a random string?

In Linux, you can use the openssl command to generate a random string. The openssl rand command can be used to generate random data. By default,

openssl rand generates binary data, but you can use the -base64 option to encode the data in base64 format, which produces a string that is easier to read and use.
To generate a random string of a specific length, you can use the -base64 and -rand options together with the head command to extract the desired number of characters. For example, the following command generates a random string of length 10:openssl rand -base64 15 | head -c 10 ; echo
The head -c 10 command extracts the first 10 characters of the output of openssl rand -base64 15. The echo command adds a newline character to the end of the output, to make the output look cleaner. You can adjust the length of the output string by changing the value passed to head -c.

22. How to generate a random string of 7 characters?

In Linux, you can use the openssl rand command to generate a random string of a specified length. To generate a random string of 7 characters, you can use the following command:

openssl rand -base64 4 | cut -c1-7

This command generates 4 random bytes in Base64 format, then selects the first 7 characters using the cut command. The resulting output is a random string of 7 characters.

You can also use other methods to generate random strings in Linux, such as using the head command to select random lines from a dictionary file or using the shuf command to shuffle and select random characters from a set of characters.

Systemd

23. What is systemd?
Systemd is a system and service manager for Linux operating systems. It is responsible for controlling the startup and shutdown processes of the system, managing system services and daemons, and providing advanced logging and monitoring capabilities.

24. How to start or stop a service?
To start a service in Systemd,
sudo systemctl start <service-name>
To stop a service in Systemd,
sudo systemctl stop <service-name>

25. How to check the status of a service?
sudo service <service-name> status

26. On a system which uses systemd, how would you display the logs?
sudo journalctl -u <service-name>

27. Describe how to make a certain process/app a service
To make a process or application a service in Linux, you need to create a systemd unit file that describes the service and its configuration options. The unit file should be created in the **/etc/systemd/system/** directory with a name that ends with the **.service** extension. The unit file should define the service properties in the **[Unit]** section, the executable that starts the service in the **[Service]** section, and the dependencies and installation options in the **[Install]** section. After creating the unit file, you can start, stop, enable, and disable the service using the **systemctl** command. When writing a unit file, it's important to be clear and concise while providing all the necessary information to ensure that the service is properly managed by systemd.

28. Troubleshooting and Debugging
Troubleshooting and debugging are essential skills for Linux system administrators. To troubleshoot and debug issues on a Linux system, you should read the system logs, check the system resources, test network connectivity, use tools such as **strace** and **gdb**, check file permissions, and check for software updates. By being systematic and thorough in your approach and documenting your steps and findings, you can identify and resolve issues quickly and effectively, minimizing downtime and maximizing system performance.

29. Where system logs are located?
System logs in Linux are typically located in the **/var/log/** directory. This directory contains log files generated by various system services and applications, such as the syslog, kernel, Apache web server, and others. The log files are usually named according to the service or application that generates them, and they contain information about events, errors, and activities that occur on the system.

30. How to follow file's content as it being appended without opening the file every time?
To follow a file's content as it is being appended without opening the file every time, you can use the **tail** command with the **-f** (follow) option. The **tail** command displays the last few lines of a file, and with the **-f** option, it will continue to display any new lines that are appended to the file in real-time, you can use the following command:

tail -f /var/log/syslog

31. What are you using for troubleshooting and debugging network issues?

- Ping: to test network connectivity and check for packet loss
- Traceroute: to identify the path of network packets and locate network issues
- Netstat: to display network connections and their status
- Tcpdump: to capture and analyze network traffic
- Nmap: to scan network hosts and identify open ports and services
- Wireshark: a graphical tool to capture and analyze network traffic in detail

32. What are you using for troubleshooting and debugging disk & file system issues?

Some examples of tools used for troubleshooting and debugging disk and file system issues include:

- df and du commands: to check disk usage and identify which files and directories are taking up the most space.
- fsck: a file system checking utility that scans and repairs file system errors.
- mount and umount: commands used to mount and unmount file systems.
- lsblk: to display information about block devices and file systems.
- dmesg: to view system messages, including disk-related errors.
- smartctl: to monitor and analyze the health of a hard drive or solid-state drive (SSD)

33. What are you using for troubleshooting and debugging process issues?

Some examples of tools used for troubleshooting and debugging process issues include:

- top: to display system resource usage and information about running processes.
- ps: to display information about running processes.
- kill: to terminate a running process.
- strace: to trace system calls made by a process and identify issues with the process.
- lsof: to display information about files and network connections opened by a process.
- gdb: a debugger used to analyze and debug issues with a process

34. What are you using for debugging CPU related issues?

Some examples of tools used for debugging CPU-related issues include:

- top: to display system resource usage and information about processes, including CPU usage.

- htop: a more advanced version of top that provides more detailed information about system resource usage and processes.
- vmstat: to display information about system resource usage, including CPU usage.
- iostat: to display information about system resource usage, including CPU usage and disk I/O statistics.
- sar: to collect and report system resource usage over time, including CPU usage.

35. You get a call from someone claiming "my system is SLOW". What do you do?

- Check CPU and memory usage using top or htop
- Check disk usage and I/O using df and iostat
- Check network connectivity using ping and netstat
- Check system logs in /var/log for any error messages or warnings
- Check running processes and their resource usage using ps

36. Explain iostat output

- Device: The name of the disk or partition being monitored.
- tps: Transactions per second, the number of I/O operations per second.
- kB_read/s: The number of kilobytes read from the device per second.
- kB_wrtn/s: The number of kilobytes written to the device per second.
- kB_read: The total number of kilobytes read from the device.
- kB_wrtn: The total number of kilobytes written to the device.
- %util: The percentage of time the device was busy processing I/O operations.

37. How to debug binaries?

- Compile with debug information: When compiling the binary, include debug information using the appropriate compiler flags. This will allow for more detailed information to be included in error messages and allow for better debugging.
- Use a debugger: Debuggers like GDB (GNU Debugger) allow for interactive debugging of binary code. They allow for setting breakpoints, stepping through code, and examining variables and memory at run-time.
- Use static and dynamic analysis tools: Tools like strace, ltrace, and valgrind can be used to monitor system calls, library calls, and memory usage of a binary, respectively. This can help identify issues related to system calls or memory management.
- Review log files: Check log files related to the binary, including system logs and application logs, for any errors or warnings that may help identify the issue.

- Reproduce the issue: Try to reproduce the issue to better understand the root cause. This may involve creating test cases or using debugging tools to narrow down the issue.
- Use code profilers: Code profilers can be used to identify performance bottlenecks in the binary code. This can help identify areas of the code that may be causing issues.

38. What is the difference between CPU load and utilization?
CPU load and CPU utilization are two different metrics used to measure the performance of a CPU. CPU load refers to the number of processes that are waiting to be executed or are currently executing on the CPU. It is typically measured as the number of processes in the system's run queue. A high CPU load indicates that the CPU is being heavily utilized and may be struggling to keep up with the demand from running processes.
CPU utilization, on the other hand, refers to the percentage of time that the CPU is busy executing processes. It is typically measured as the percentage of time that the CPU spends in user mode, kernel mode, or idle. A high CPU utilization indicates that the CPU is being heavily used and may be struggling to keep up with the demand from running processes.
While both metrics provide insight into the performance of a CPU, they measure different aspects of CPU performance. CPU load provides information about how many processes are waiting to be executed, while CPU utilization provides information about how busy the CPU is. By understanding the difference between CPU load and utilization, system administrators can better diagnose performance issues and optimize the performance of their systems**.**

39. How you measure time execution of a program?

To measure the time execution of a program, the time command can be used. Simply prefix the command with "time"

Scenarios

40. You have a process writing to a file. You don't know which process exactly, you just know the path of the file. You would like to kill the process as it's no longer needed. How would you achieve it?
To find and kill a process that is writing to a specific file, you can use the "lsof" command to list all open files and the processes that have them open. First, identify the path of the file that is being written to. Then, use the "lsof" command with the grep utility to search for the file by its path. The output of the command will list all the processes that have the file open, along with their PID and other information. Look for the process that is writing to the file by checking the "FD" column for the "w" flag. Once you have identified the PID of the process, you can kill it using the "kill" command with the appropriate

signal. Use the SIGTERM signal first to request a graceful termination, and if the process does not respond, use the SIGKILL signal to force the process to terminate. However, forcefully terminating a process should be used as a last resort, as it may result in data loss or other issues**.**
For example, if the PID is 12345, you can run the following command:
kill 12345

Kernel

41. What is a kernel, and what does it do?
The kernel is the central part of an operating system that acts as a bridge between software applications and the hardware components of a computer. It is a program that manages and controls the computer's resources, including the CPU, memory, input/output devices, and network connections.

42. How do you find out which Kernel version your system is using?
You can find out which kernel version your system is using by running the "uname" command with the "-r" option. This will display the kernel release number, which includes the version number and other information about the kernel. To run this command, open a terminal window and type:
uname -r

43. What is a Linux kernel module and how do you load a new module?
A Linux kernel module is a piece of code that can be dynamically loaded into the kernel at runtime to add new functionality or support for new hardware devices. Kernel modules are used to extend the functionality of the kernel without requiring a complete recompilation and reboot of the operating system.
To load a new module, you need to have administrative privileges (root access) on the system. Here are the steps to load a new kernel module:

- Identify the name of the module you want to load: You can search for the module name in the Linux kernel source code or online resources.
- Check if the module is already loaded: You can use the "lsmod" command to list all the currently loaded modules and check if the module you want to load is already present.
- Load the module: You can use the "modprobe" command to load a new module. For example, to load the "usb-storage" module, you can run the following command as root:
    1. sudo modprobe usb-storage
- Verify the module is loaded: You can use the "lsmod" command again to verify that the new module has been loaded.

44. Explain user space vs. kernel space
The user space is the area of memory where user-level applications and programs run. These are programs that are executed by users and do not require administrative privileges to run. User space programs interact with the

kernel space through system calls, which allow them to access kernel-level services such as input/output operations, memory management, and process management.
On the other hand, the kernel space is the area of memory that is reserved for the kernel and its modules. This is where the operating system's core functions are executed, including hardware drivers, system calls, and interrupt handling. Programs running in kernel space have direct access to the system's hardware and resources and have higher privileges than those running in user space.

45. In what phases of kernel lifecycle, can you change its configuration?
Configuration changes can be made to the kernel during three phases of its lifecycle: compile time, boot time, and runtime. Compile-time configuration is done by modifying the kernel source code before it is compiled. Boot-time configuration is done by passing parameters to the bootloader or kernel at boot time. Runtime configuration is done by modifying kernel parameters while the system is running.

46. Where can you find kernel's configuration?
The kernel configuration is stored in a file called .config in the root directory of the kernel source tree. This file contains a list of configuration options and settings that were selected during the kernel compilation process.

47. Where can you find the file that contains the command passed to the boot loader to run the kernel?
On Linux, boot loader configuration files are typically located in the /boot directory and named grub.cfg or menu.lst. This file contains boot loader configuration information, such as command line options and parameters passed to the kernel at boot time.

48. How to list kernel's runtime parameters?

```
sysctl -a
```

49. Will running sysctl -a as a regular user vs. root, produce different result?
Running sysctl -a as a normal user will only show kernel parameters that are readable by non-privileged users. Many kernel parameters require root privileges to read or modify. So running sysctl -a as a regular user may not show all parameters or their current values.

50. You would like to enable IPv4 forwarding in the kernel, how would you do it?
To enable IPv4 forwarding in the kernel, you can use the sysctl command to modify the net.ipv4.ip_forward parameter.

Here are the steps to enable IPv4 forwarding in the kernel:
Open a terminal or command prompt as root or with sudo privileges.
Check the current value of the net.ipv4.ip_forward parameter by running sysctl net.ipv4.ip_forward
To enable IPv4 forwarding, set the value of the net.ipv4.ip_forward parameter to 1 by running the following command sysctl -w net.ipv4.ip_forward=1
Check if the value of the net.ipv4.ip_forward parameter was changed to 1 by running the following command
sysctl net.ipv4.ip_forward

51. How sysctl applies the changes to kernel's runtime parameters the moment you run sysctl command?

When you run sysctl commands to change kernel parameters, the changes take effect immediately and affect the real-time behavior of your system.
The sysctl command uses the sysctl() system call to communicate with the kernel and change the value of specified kernel parameters. The kernel then updates the appropriate data structures and applies the new settings immediately.

52. How changes to kernel runtime parameters persist? (applied even after reboot to the system for example)

•Identify the kernel parameter that you want to modify and its current value. You can use the sysctl command to view the current values of kernel parameters.
•Decide on the new value that you want to set for the parameter.
•Modify the appropriate configuration file in the /etc/sysctl.d directory or add the parameter to the /etc/sysctl.conf file. To modify an existing file, you can use a text editor such as vi or nano. To create a new file, you can use the touch command. For example, to set the net.ipv4.tcp_syncookies parameter to 0, you can create a new file called /etc/sysctl.d/99-sysctl.conf
•Save the changes to the file(s).
•To apply the changes immediately, run the sysctl command with the -p
•To verify that the changes have been applied, you can run the sysctl
•Finally, reboot the system to ensure that the changes persist across reboots. After the reboot, you can use the sysctl command to verify that the changes are still in effect.

53. Are the changes you make to kernel parameters in a container, affects also the kernel parameters of the host on which the container runs?

No, changes made to kernel settings inside the container do not affect the kernel settings of the server on which the container is running.
Containers run on the host OS and share the same kernel with the host. However, the kernel container view is isolated from the host view, and changes to the kernel settings inside the container only affect the kernel container view.

SSH

54. What is SSH? How to check if a Linux server is running SSH?

SSH (Secure Shell) is a network protocol that allows secure remote login and remote execution of commands on a computer over an unsecured network. It is commonly used by system administrators and developers to access and manage remote servers.

To check if a Linux server is running SSH,

systemctl status sshd

55. Why SSH is considered better than telnet?

SSH is considered better than Telnet for several reasons such as Security, Authentication, Portability and Functionality.

56. What is stored in ~/.ssh/known_hosts?

The ~/.ssh/known_hosts file stores the host keys of remote hosts that the user has previously connected to using SSH. It is used to authenticate the remote host during subsequent connections and prevent man-in-the-middle attacks. When a user connects to a remote host for the first time, SSH prompts the user to verify the host's key fingerprint and stores it in the known_hosts file. On subsequent connections, SSH compares the stored key with the key presented by the remote host during the connection process to ensure that the host has not been compromised or changed.

57. You try to ssh to a server and you get "Host key verification failed". What does it mean?

When you see the error message "Host key verification failed" while trying to ssh to a server, it means that the SSH client has detected a mismatch between the key of the remote host and the key stored in the ~/.ssh/known_hosts file on the client machine. This could be due to a few reasons:

- The remote host's key has changed since you last connected to it, possibly indicating a security breach or an intentional change.
- You are connecting to a different remote host with the same IP address or hostname as the one you previously connected to, and the key has changed.
- The known_hosts file on the client machine has been tampered with or corrupted.

58. What is the difference between SSH and SSL?

SSH (Secure Shell) and SSL (Secure Sockets Layer) are both security protocols used to establish secure connections between two systems over a network. However, there are some differences between them:

- Functionality: SSH is primarily used for remote command-line access and file transfers, while SSL is primarily used for secure web browsing, email communication, and data transfer.
- Port number: SSH typically uses port 22, while SSL typically uses port 443.
- Encryption: Both protocols use encryption to secure the data being transferred, but they use different encryption methods. SSH uses symmetric encryption for data transfer, while SSL uses asymmetric encryption for key exchange and symmetric encryption for data transfer.
- Authentication: SSH uses public key cryptography for authentication, while SSL uses a combination of public key cryptography and digital certificates for authentication.
- Protocol versions: SSH has several versions, including SSH1 and SSH2, while SSL has several versions, including SSLv2, SSLv3, and TLS (Transport Layer Security).

59. What ssh-keygen is used for?
ssh-keygen is a command-line utility used for generating, managing and converting authentication keys for SSH. The tool generates two keys: a public key and a private key. The public key is used for authentication and it's placed on the remote server you want to access, while the private key is kept on your local computer. The ssh-keygen tool is primarily used for creating a new SSH key pair, but it can also be used for various other tasks such as changing the passphrase, converting the key format, or changing the comment associated with the key. Once you have generated a key pair, you can copy the public key to the remote server's authorized_keys file, allowing you to log in without a password. Overall, ssh-keygen is a powerful tool for managing SSH keys and is a vital component of secure remote server access.

60. What is SSH port forwarding?
SSH port forwarding, also known as SSH tunneling, is a technique used to securely forward network connections from one computer to another through an encrypted SSH connection. It allows users to create an encrypted tunnel between their local computer and a remote server, forwarding network traffic from a specified local port to a specific remote port. This can be useful for a variety of purposes, such as accessing a service on a remote server that is not directly accessible from the local network, or for encrypting communications between two servers. SSH port forwarding can be configured in a variety of ways, such as local port forwarding, remote port forwarding, and dynamic port forwarding.