Linux Assignment -02

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

In the Linux Filesystem Hierarchy Standard (FHS), the / (root) directory is the top-level directory in the file system hierarchy. It contains all other directories and files in the system. The root directory is the starting point for all file system paths and is represented by a forward slash (/).

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

/bin, /sbin, /usr/bin and /usr/sbin

/etc

/home

/var

/tmp

• /bin: Essential command binaries that need to be available in single-user mode, including to bring up the system or repair it, such as cat, ls, cp, and rm.

• /sbin: Essential system binaries that are necessary for booting, restoring, and repairing the system, such as fsck, ifconfig, and route.

• /usr/bin: Non-essential command binaries that are used by system users, such as awk, sed, and grep.

• /usr/sbin: Non-essential system binaries that are used by system administrators, such as useradd, userdel, and usermod.

• /etc: System configuration files that are used by system services and applications, such as passwd, group, and fstab.

• /home: Home directories for regular users.

• /var: Variable data files that are expected to grow in size, such as log files, spool files, and temporary files.

• /tmp: Temporary files that are deleted when the system is rebooted.

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

The /tmp directory is special when compared to other directories because its contents are deleted when the system restarts. This directory is used for storing temporary files that are created by various applications and processes running on the system. The /tmp directory is typically located in the root directory (/) of the file system hierarchy.

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

In Linux, the /proc directory contains a hierarchy of special files that represent the current state of the kernel and the system. The /proc directory is a virtual file system that provides a way for applications and users to peer into the kernel's view of the system.

Some of the information that can be found in the /proc directory includes:

• System information such as CPU and memory usage, uptime, and load average.

• Information about running processes, including their process ID (PID), memory usage, and open files.

• Information about system hardware, such as CPU and memory specifications.

• Information about system configuration, such as kernel parameters and network settings.

5.What makes /proc different from other filesystems?

The /proc filesystem is different from other filesystems in Linux because it is a virtual filesystem that does not exist on disk. It is a special filesystem that contains a hierarchy of special files that represent the current state of the kernel and the system. The /proc filesystem provides a way for applications and users to peer into the kernel's view of the system, allowing them to access information about the current state of the kernel and the system.

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

False. The /proc directory is a virtual filesystem that contains a hierarchy of special files which represent the current state of the kernel and the system. It is not a real filesystem, and the files in the /proc directory are generated on the fly by the kernel when they are accessed. The files in the /proc directory are called virtual files, and they contain information about various aspects of the system, such as system hardware, system configuration, running processes, and kernel parameters.

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

The /proc/cmdline file is useful for troubleshooting boot problems and for verifying that kernel parameters have been set correctly. The contents of the /proc/cmdline file can be viewed using the cat command or any other text editor.

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

In Linux, the system devices such as block storage can be found in the /dev directory. The /dev directory contains file system entries that represent devices that are attached to the system. These files are essential for the system to function properly.

Permissions

9.How to change the permissions of a file?

To change the permissions of a file in Linux, you can use the chmod command. The chmod command is used to change the permissions of a file or directory. There are two ways to specify the permissions: symbolic mode and numeric mode.

In symbolic mode, you can use the + and - signs to add or remove permissions, respectively. For example, to add read, write, and execute permissions to a file, you can use the command chmod +rwx filename. To remove write permission from a file, you can use the command chmod -w filename.

In numeric mode, you can use a three-digit octal number to specify the permissions. The first digit specifies the permissions for the owner of the file, the second digit specifies the permissions for the group that owns the file, and the third digit specifies the permissions for everyone else.

10.What does the following permissions mean?:

777-owner,groupmembers,others have read,write,execute permissions.

644-owner have read and write permisssion while groupmembers and ohers have only read permission.

750-owner have read write and execute permission,group have read and execute permission,others have no permissionas

11.What this command does? chmod +x some_file

The command chmod +x some_file makes the file some_file executable. The chmod command is used to change the permissions of a file or directory in Linux. The +x option adds the execute permission to the file, which allows the file to be executed as a program or script.

12.Explain what is setgid and setuid

The setuid permission, represented by the s bit in the file permissions, allows a user to run an executable file with the permissions of the file owner, instead of the permissions of the user who launched it. This is useful for programs that need to perform privileged operations, such as changing system settings or accessing sensitive data.

The setgid permission, represented by the s bit in the group permissions, allows a user to create files and directories with the group ownership of the parent directory, instead of the user's default group ownership. This is useful for shared directories where multiple users need to access and modify files.

13.What is the purpose of sticky bit?

The purpose of the sticky bit is to restrict file deletion in a directory. When the sticky bit is set on a directory, only the owner of a file or directory or the root user can delete or rename the file or directory, even if other users have write permissions on the file or directory. This is useful for shared directories where multiple users need to create and modify files, but only the owner should be able to delete them.

14.What the following commands do?

chmod-chmod is used to change the permissions of a file or directory. It can be used with either symbolic mode or numeric mode to change file permissions.

chown-chown is used to change the owner of a file or directory. It can be used to change the owner to a specific user or group.

Chgrp-chgrp is used to change the group ownership of a file or directory. It can be used to change the group ownership to a specific group.

15.What is sudo? How do you set it up?

sudo is a command in Linux that allows users to run commands with administrative or root privileges. It stands for "superuser do" and is used to perform tasks that require elevated permissions, such as installing software or modifying system files.

To set up sudo, you need to add your user account to the sudoers file. The sudoers file is located at /etc/sudoers and requires root permissions to edit. You can use the visudo command to edit the sudoers file and add your user account to the list of users who are allowed to use sudo.

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 be the root user or use the sudo command.

17.Explain what are ACLs. For what use cases would you recommend to use them?

In computer security, an Access Control List (ACL) is a list of permissions associated with a system resource or object. An ACL specifies which users or system processes are granted access to objects, as well as what operations are allowed on given objects. Each entry in a typical ACL specifies a subject and an operation.

18.You try to create a file but it fails. Name at least three different reason as to why it could happen1. 1.Permission issues: The user may not have the necessary permissions to create a file in the directory where they are trying to create it. This could be due to file permissions or ownership issues. For example, if the user does not have write permissions on the directory, they will not be able to create a file in it.

2. Disk space issues: The user may not have enough disk space to create the file. If the disk is full or nearly full, the user will not be able to create any new files until some space is freed up.

3. File already exists: The user may be trying to create a file with a name that already exists in the directory. In this case, the system will not allow the user to create a file with the same name. The user will need to choose a different name for the file.

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

1. Use the setfacl command to modify the ACL of the chmod binary file for the user with read and execute permissions: setfacl -m u::rx /usr/bin/chmod

2. Restore the execute permission to the chmod binary: chmod +x $(which chmod)

3. Remove all ACL entries of chmod (optional): setfacl -b /usr/bin/chmod

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. Here are a few methods:

1. Use rsync: If you have SSH access to the remote machine and the remote machine has rsync installed, you can use the following command to copy a file to the remote location: rsync -avz /path/to/local/file username@host:/destination/path

2. Use scp: scp is a command-line tool used to copy files securely between hosts. To copy a file from a local system to a remote system using scp, run the following command: scp file.txt remote_username@10.10.0.2:/remote/directory

3. Use sftp: sftp is a secure file transfer protocol that allows you to transfer files between hosts securely. To copy a file from a local system to a remote system using sftp, run the following command: sftp remote_username@remote_host:/remote/directory and then use the put command to upload the file.

4. Use ssh: If you have SSH access to the remote machine, you can use the ssh command to copy a file to the remote location. First, navigate to the directory where the file is located and then run the following command: ssh username@host "cat > /destination/path/file" < /path/to/local/file

21.How to generate a random string?

Using the $RANDOM variable: You can use the $RANDOM variable in a Bash script to generate a random string. Here's an example:

```
#!/bin/bash

charset="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"

length=10

result=""

for i in $(seq 1 $length); do
```

```
    result="$result${charset:RANDOM%${#charset}:1}"
done
echo $result
```

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

Using $RANDOM variable: You can use the $RANDOM variable in a Bash script to generate a random string of 7 characters. Here's an example:

```
#!/bin/bash
charset="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"
length=7
result=""
for i in $(seq 1 $length); do
    result="$result${charset:RANDOM%${#charset}:1}"
done
echo $result
```

Systemd

23.What is systemd?

systemd is a software suite that provides an array of system components for Linux operating systems. Its primary component is a "system and service manager" that is used to bootstrap user space and manage user processes. systemd is designed to unify service configuration and behavior across Linux distributions.

24.How to start or stop a service?

1. To start a service:

```
sudo systemctl start <service_name>
```

2.To stop a service:

```
sudo systemctl stop <service_name>
```

25.How to check the status of a service?

1. To check whether a service is running:

```
sudo systemctl is-active <service_name>
```

2. To check whether a service is enabled:

```
sudo systemctl is-enabled <service_name>
```

3. To view the status of all services:

```
sudo systemctl status
```

4.To view the status of a specific service:

```
sudo systemctl status <service_name>
```

26.On a system which uses systemd, how would you display the logs?

To display the logs on a system that uses systemd, you can use the journalctl command. Here are some examples:

1. To display all logs:

```
sudo journalctl
```

2. To display logs for a specific service:

```
sudo journalctl -u <service_name>
```

27.Describe how to make a certain process/app a service

1. Create a new service unit file in the /etc/systemd/system/ directory. You can use any text editor to create the file. For example:

```
sudo nano /etc/systemd/system/myapp.service
```

2. In the service unit file, define the service by specifying the service name, description, and the command to start the process or application. For example:

```
[Unit]

Description=My App Service


[Service]

ExecStart=/usr/bin/myapp


[Install]

WantedBy=multi-user.target
```

3. Save the service unit file and exit the text editor.

4. Reload the systemd daemon to read the new service unit file:

```
sudo systemctl daemon-reload
```

5. Start the service:

```
sudo systemctl start myapp.service
```

6. Enable the service to start automatically at boot time:

```
sudo systemctl enable myapp.service
```

7. Check the status of the service:

```
sudo systemctl status myapp.service
```

28.Troubleshooting and Debugging

Troubleshooting and debugging are two related but distinct processes. Troubleshooting is the process of identifying and resolving problems in a system or application, while debugging is the process of finding and fixing errors in computer code.

29.Where system logs are located?

System logs in Linux are located in the /var/log directory. The logs are stored as plain text and are easy to read.we use any GUI or CLI based text editor to read these files.

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 option. Here are the steps:

1. Open a terminal window.

2. Navigate to the directory where the file is located.

3. Type the following command:

```
tail -f <file_name>
```

For example, to follow the content of a file named example.log, type:

```
tail -f example.log
```

4. Press `Ctrl + C` to stop following the file.

This will display the content of the file as it is being appended in real-time. The -f option tells tail to keep the file open and display new lines as they are added to the file.

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

There are several tools that can be used for troubleshooting and debugging network issues. Some of the commonly used tools are:

1. Ping: This tool is used to test the connectivity between two devices on a network by sending ICMP packets and receiving responses.

2. Traceroute: This tool is used to trace the path that packets take from one device to another on a network.

3. Ipconfig/Ifconfig: This tool is used to view and configure network interfaces on a device.

4. NSlookup: This tool is used to query DNS servers to resolve domain names to IP addresses.

5. Wireshark: This tool is used to capture and analyze network traffic to troubleshoot issues.

6. Netstat: This tool is used to view active network connections and their status.

7. Nmap: This tool is used to scan networks and identify open ports and services.

8. Tcpdump: This tool is used to capture and analyze network traffic in real-time.

9. MTR: This tool is used to combine the functionality of Ping and Traceroute to provide more detailed information about network issues.

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

There are several tools that can be used for troubleshooting and debugging disk and file system issues. Some of the commonly used tools are:

1. Check Disk (CHKDSK): This tool is used to scan and repair disk errors on Windows systems.

2. Disk Utility: This tool is used to manage and repair disks and file systems on macOS systems.

3. fsck: This tool is used to check and repair file systems on Linux and Unix systems.

4. SMART: This tool is used to monitor and diagnose the health of hard drives and solid-state drives.

5. Disk Management: This tool is used to manage and troubleshoot disks and volumes on Windows systems.

6. Diskpart: This tool is used to manage disks and volumes on Windows systems.

7. TestDisk: This tool is used to recover lost partitions and files on Windows, Linux, and Unix systems.

8. GParted: This tool is used to manage and resize partitions on Linux and Unix systems.

9. HFSExplorer: This tool is used to access and recover files from HFS+ formatted disks on Windows systems.

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

There are several tools that can be used for troubleshooting and debugging process issues. Some of the commonly used tools are:

1. Task Manager: This tool is used to view and manage running processes on Windows systems.

2. Process Explorer: This tool is used to view detailed information about running processes on Windows systems.

3. GDB: This tool is used to debug C and C++ programs on Linux and Unix systems.

4. strace: This tool is used to trace system calls and signals made by a process on Linux and Unix systems.

5. lsof: This tool is used to view open files and network connections associated with a process on Linux and Unix systems.

6. ps: This tool is used to view information about running processes on Linux and Unix systems.

7. top: This tool is used to view real-time information about system resource usage and running processes on Linux and Unix systems.

8. tcpdump: This tool is used to capture and analyze network traffic on Linux and Unix systems.

9. itrace: This tool is used to trace system calls made by a process on macOS systems.

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

There are several tools that can be used for debugging CPU related issues. Some of the commonly used tools are:

1. Performance Profiler: This tool is used to analyze CPU usage without debugging in Visual Studio.

2. Perf: This tool is used to profile and trace system performance on Linux and Unix systems.

3. top: This tool is used to view real-time information about system resource usage and running processes on Linux and Unix systems.

4. htop: This tool is used to view real-time information about system resource usage and running processes on Linux and Unix systems.

5. strace: This tool is used to trace system calls and signals made by a process on Linux and Unix systems.

6. ltrace: This tool is used to trace library calls made by a process on Linux and Unix systems.

7. GDB: This tool is used to debug C and C++ programs on Linux and Unix systems.

8. DTrace: This tool is used to trace and debug system performance on macOS and some Linux and Unix systems.

DTrace is a dynamic tracing framework that allows an admin or developer to get a real-time look into a system either in user or kernel mode. It is used to trace and debug system performance on macOS and some Linux and Unix systems. DTrace provides dynamic instrumentation of both user/kernel functions, the ability to script using the D-language, speculative tracing, and displays system information and events.

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

1. Check for malware and viruses by running a full system scan with an antivirus program.

2. Check for available updates for the operating system and device drivers.

3. Check for low disk space and free up space if necessary.

4. Adjust the appearance and performance of Windows.

5. Pause OneDrive syncing.

6. Restart the system and open only the necessary apps.

7. Use ReadyBoost to help improve performance.

36.Explain iostat output

The iostat command is used for monitoring system input/output device loading by observing the time the devices are active in relation to their average transfer rates. It generates reports that can be used to change system configuration to better balance the input/output load between physical disks. The iostat command can display both basic and extended metrics. The basic metrics include:

1. Device: The name of the device being monitored.

2. tps: The number of transfers per second that were issued to the device.

3. kB_read/s: The number of kilobytes per second that were read from the device.

4. kB_wrtn/s: The number of kilobytes per second that were written to the device.

5. kB_read: The total number of kilobytes that were read from the device.

6. kB_wrtn: The total number of kilobytes that were written to the device.

The extended metrics include:

1. rrqm/s: The number of read requests that were merged per second.

2. wrqm/s: The number of write requests that were merged per second.

3. r/s: The number of read requests that were issued per second.

4. w/s: The number of

37.How to debug binaries?

Debugging binaries can be done using various tools and techniques. Some of the commonly used methods are:

1. Debugging symbols: If the original binary was written in C/C++ and you have a matching PDB (Program DataBase) file from that build, then you can debug the release with function names visible just by loading the exe into Visual Studio and single-stepping into it.

2. GDB: GDB is a popular debugger that can be used to debug binaries on Linux and Unix systems. It can be used to set breakpoints, examine memory, and step through code.

3. Debugging Tools for Windows: Debugging Tools for Windows includes several tools in addition to the debugging engine and debugging environments. The tools are in the installation directory of Debugging Tools for Windows.

4. Binary analysis tools: There are several binary analysis tools available on Linux and Unix systems that can be used to analyze and debug binaries. Some of the commonly used tools are objdump, readelf, and ltrace.

5. Dynamic analysis tools: Dynamic analysis tools like strace and ltrace can be used to trace system calls and library calls made by a binary.

38.What is the difference between CPU load and utilization?

CPU load and utilization are related to the performance of the CPU, but they are not the same thing. CPU utilization refers to the percentage of time that the CPU is busy processing instructions. It is a measure of how much work the CPU is doing at a given time. CPU load, on the other hand, is a measure of the number of processes that are using or waiting for the CPU at a given time. It is a measure of how much demand there is for the CPU's processing power.

CPU utilization is typically measured as a percentage of the total CPU capacity, while CPU load is measured as a count of the number of processes using or waiting for the CPU. CPU utilization can be affected by the speed of the CPU, the number of cores, and the workload being processed. CPU load can be affected by the number of processes running on the system, the priority of those processes, and the amount of I/O activity.

39.How you measure time execution of a program?

1. Manual stopwatch: The simplest and most intuitive way to measure time is to use a stopwatch manually from the moment you start a program. This method is called wall-clock time.

2. C/C++ clock() function: The clock() function in C/C++ can be used to measure the CPU time used by a program.

3. C/C++ gettimeofday() function: The gettimeofday() function in C/C++ can be used to measure the wall-clock time used by a program.

4. Linux time command: The time command in Linux can be used to measure the wall-clock time, user CPU time, and system CPU time used by a program.

5. Profiling tools: Profiling tools like gprof, perf, and valgrind can be used to measure the execution time of a program and identify performance bottlenecks.

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 the process writing to a file, you can use the lsof command followed by the path of the file. This will list all the processes that have the file open. Once you have identified the process, you can use the kill command followed by the process ID to terminate the process. The steps are:

1. Open a terminal window.

2. Type lsof followed by the path of the file. For example, lsof /path/to/file.

3. Look for the process ID (PID) of the process that has the file open.

4. Type kill followed by the PID of the process. For example, kill 1234.

5. Verify that the process has been terminated by running lsof again and checking that the file is no longer open.

Kernel

41.What is a kernel, and what does it do?

A kernel is a fundamental component of an operating system that acts as a bridge between the software and hardware. It is the core of the operating system and controls all the important functions of hardware. The kernel is responsible for managing system resources such as memory, CPU, and input/output devices. It provides a layer of abstraction between the hardware and software, allowing applications to interact with the hardware without needing to know the details of the underlying hardware.

42.How do you find out which Kernel version your system is using?

1. Open a terminal window.

2. Type uname -r and press Enter.

3. The output will show the kernel version of your system.

43.What is a Linux kernel module and how do you load a new module?

A Linux kernel module is a program that can be loaded into or unloaded from the kernel on demand, without the need to recompile the kernel or reboot the system. It is intended to enhance the functionality of the kernel and act as a translator between devices and the Linux kernel.

To load a new kernel module, you can use the modprobe command followed by the name of the module. The modprobe command is intelligent enough to resolve module dependencies, so when you load a module, its dependencies are also loaded. Here are the steps to load a new kernel module:

1. Open a terminal window.

2. Type sudo modprobe <module-name> and press Enter. Replace <module-name> with the name of the module you want to load.

3. If the module is successfully loaded, there will be no output. If there is an error, the output will indicate the reason for the error.

44.Explain user space vs. kernel space

In a Linux operating system, there are two distinct memory spaces: user space and kernel space. User space is the memory space where user applications run, while kernel space is the memory space where the operating system and kernel-level drivers run.

User space is the area of memory that non-kernel applications run in. User space processes run in user mode, which is the non-privileged execution mode that the process' instructions are executed with. Each user space process normally runs in its own virtual memory space and cannot access the memory of other processes unless explicitly allowed. This is the basis for memory protection in today's mainstream operating systems and a building block for privilege separation.

Kernel space, on the other hand, is the memory space where the operating system and kernel-level drivers run. The kernel space is privileged and has access to all system resources, including hardware devices. The kernel space provides services such as process management, file management, signal handling, memory management, and thread management.

45.In what phases of kernel lifecycle, can you change its configuration?

The kernel configuration can be changed during the development phase and after the release of the kernel. During the development phase, developers can change the kernel configuration to add or remove features, optimize performance, or fix bugs. The kernel configuration can be changed using tools like make menuconfig, make gconfig, or make xconfig. Once the kernel is released, the configuration can still be changed by updating the kernel configuration file and rebuilding the kernel.

46.Where can you find kernel's configuration?

The kernel configuration file can be found in different locations depending on the distribution and version of Linux. Here are some common locations where the kernel configuration file can be found:

1. /boot/config-*: This is the location of the kernel configuration file for the currently running kernel. The exact file name may vary depending on the distribution and version of Linux.

2. /proc/config.gz: This is a compressed file that contains the kernel configuration for the currently running kernel. This file is only available if the kernel was compiled with the CONFIG_IKCONFIG option.

3. /.config: This is the default name of the kernel configuration file that is used during the kernel compilation process. If you have previously compiled the kernel, the configuration file may be located in the kernel source directory.

4. make oldconfig: This command can be used to update an existing kernel configuration file with any new options that have been added to the kernel since the last release.

47.Where can you find the file that contains the command passed to the boot loader to run the kernel?

The file that contains the command passed to the boot loader to run the kernel can be found in the boot loader configuration file. The location of the boot loader configuration file depends on the boot loader being used.

For example, if the GRUB boot loader is being used, the configuration file is located at /boot/grub/grub.cfg or /boot/grub2/grub.cfg. The command passed to the boot loader to run the kernel can be found in the linux line of the configuration file.

48.How to list kernel's runtime parameters?

To list the kernel's runtime parameters, you can use the cat command to display the contents of the /proc/cmdline file. This file contains the command line parameters that were passed to the kernel at boot time. Here are the steps:

1. Open a terminal window.

2. Type cat /proc/cmdline and press Enter.

3. The output will show the kernel's runtime parameters.

49.Will running sysctl -a as a regular user vs. root, produce different result?

Running sysctl -a as a regular user vs. root will produce different results. The sysctl command is used to view and modify kernel parameters at runtime. Some kernel parameters are only accessible to the root user, while others can be accessed by regular users.

When running sysctl -a as a regular user, only the kernel parameters that are accessible to regular users will be displayed. This may be a subset of the total number of kernel parameters available on the system.

When running sysctl -a as the root user, all kernel parameters will be displayed, including those that are only accessible to the root user.

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 following steps:

1. Open a terminal window.

2. Type sudo sysctl net.ipv4.ip_forward=1 and press Enter. This will enable IPv4 forwarding in the kernel.

3. To make the change persistent across reboots, edit the /etc/sysctl.conf file and add the following line: net.ipv4.ip_forward=1.

4. Save the file and exit.

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

When you run the sysctl command to modify kernel parameters at runtime, the changes are applied immediately to the kernel's runtime parameters. The sysctl command writes the new value to the corresponding file in the /proc/sys/ directory, which is a virtual file system that provides an interface to kernel parameters.

The kernel reads the new value from the file and applies it to the corresponding parameter. This means that the changes take effect immediately and do not require a system reboot or any other action to be applied.

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

To make changes to kernel runtime parameters persist even after a system reboot, you need to modify the configuration files that are read during the boot process. There are several ways to achieve this:

1. /etc/sysctl.conf: This file contains a list of kernel parameters and their values. You can add new parameters or modify existing ones in this file. The changes will be applied during the boot process.

2. /etc/sysctl.d/: This directory contains configuration files that are read during the boot process. Yo

u can create a new file in this directory with the extension .conf and add new parameters or modify existing ones in the file. The changes will be applied during the boot process.

3. /etc/rc.local: This file contains commands that are executed during the boot process. You can add a command to set a kernel parameter using the sysctl command in this file. The changes will be applied during the boot process.

4. /etc/modprobe.d/: This directory contains configuration files for kernel modules. You can create a new file in this directory with the extension .conf and add a line to set a kernel parameter for a specific module. The changes will be applied during the boot process.

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?

The changes made to kernel parameters in a container do not affect the kernel parameters of the host on which the container runs. Containers are isolated from the host system and have their own set of kernel parameters.

When a container is started, it uses the kernel of the host system, but the kernel parameters are specific to the container and do not affect the host system.

SSH

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

To check if a Linux server is running SSH, you can use one of the following methods:

1. Use the ps command to list all the processes and filter the output using grep to check if the SSH process is running. The command is ps aux | grep sshd.

2. Use the systemctl command to check the status of the SSH service. The command is systemctl status sshd.

3. Check if the process sshd is running by using the command service sshd status.

4. Check if the process sshd is listening on port 22 by using the command netstat -tln | grep 22.

5. Use the telnet command to connect to the SSH port (port 22 by default) of the server. If the connection is successful, SSH is running on the server.

55.Why SSH is considered better than telnet?

1. Security: Telnet sends data in plain text, which makes it highly vulnerable to eavesdropping and interception. SSH, on the other hand, uses encrypted format for data transmission and also uses a secure channel, making it much more secure.

2. Authentication: Telnet does not use any authentication mechanisms for establishing a connection, while SSH uses public-key encryption as the most common and secure authentication method.

3. Portability: Telnet was designed with private networks in mind, while SSH was designed to cope with public networks and the need to maintain privacy and security when transferring data and making remote connections.

4. Overhead: SSH adds a bit more overhead to the bandwidth compared to Telnet, but the added security and authentication make it worth the extra overhead.

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

The ~/.ssh/known_hosts file is a file that stores the public key of all the servers that you have connected to using SSH. This file is used to verify the identity of servers in the future. When you connect to a server using SSH, the server sends its public key to your computer, and your computer stores the key in the known_hosts file.

The next time you connect to the same server, SSH checks the public key stored in the known_hosts file against the public key sent by the server. If the keys match, the connection is established. If the keys do not match, SSH warns you that the server's identity cannot be verified and asks you to confirm whether you want to continue connecting.

The known_hosts file is located in the ~/.ssh/ directory in the user's home directory. Each line in the file contains the public key of a server, along with the server's hostname or IP address.

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

The error message "host key verification failed" occurs when the server's host key does not match the key that was expected. This can happen when the server's key has been changed, or when the key has been compromised.

To fix the "host key verification failed" error, you can do one of the following:

1. Remove the old key entry from the known_hosts file and try connecting again.

2. Verify that the server's key has not been compromised and that the new key is legitimate, and then add the new key to the known_hosts file.

3. Disable host key checking by adding the -o StrictHostKeyChecking=no option to the SSH command. This is not recommended as it can compromise security.

58.What is the difference between SSH and SSL?

SSH (Secure Shell) and SSL (Secure Sockets Layer) are both protocols used for secure communication over a network, but they have different applications and methods of operation.

The main difference between SSH and SSL is their application. SSH is used for creating a secure tunnel to access network devices and servers over the internet, while SSL is mostly used for establishing a secure connection between websites and clients.

Another difference is in the method they both operate. SSH uses a three-step process: server verification, session key generation, and client authentication. It has a username and password authentication system. On the other hand, SSL uses digital certificates and public key infrastructure, and authentication only happens on the server-side.

59.What ssh-keygen is used for?

ssh-keygen is a tool used to generate, manage, and convert authentication keys for SSH (Secure Shell). It is used to create new authentication key pairs for SSH, which are used for automating logins, single sign-on, and for authenticating hosts.

The ssh-keygen command is a component of most SSH implementations used to generate a public key pair for use when authenticating with a remote server. Users generate a new public key and then copy their public key to the server using SSH and their login credentials for the remote server.

ssh-keygen is able to generate a key using one of three different digital signature algorithms. With the help of the ssh-keygen tool, a user can create passphrase keys for any of these key types. It can also be used to generate groups for use in Diffie-Hellman group exchange (DH-GEX), generate and update Key Revocation Lists, and to test whether given keys have been revoked.

60.What is SSH port forwarding?

SSH port forwarding, also known as SSH tunneling, is a technique used to forward network traffic from one host to another over an encrypted SSH connection. It allows you to forward otherwise insecure TCP traffic inside a secure SSH tunnel from a local machine to a destination server.

With SSH port forwarding, you can securely access services on a remote server that are not directly accessible from your local machine. For example, you can use SSH port forwarding to access a web server running on a remote machine that is not directly accessible from the internet.

There are two types of SSH port forwarding: local port forwarding and remote port forwarding. Local port forwarding forwards traffic from a local port on your machine to a remote port on a remote machine. Remote port forwarding forwards traffic from a remote port on a remote machine to a local port on your machine.