Module 13
Where Data is Stored

Exam Objective

4.3 Where Data is Stored

Objective Description

Where various types of information are stored on a Linux system.

Introduction

FSH and Processes

- A typical Linux system has thousands of files. The **Filesystem Hierarchy Standard (FHS)** provides a guideline for distributions on how to organize these files.

- The *Linux kernel* is the core of the GNU/Linux operating system. It is important to understand the role of the Linux kernel and how it both processes and provides information about the system

- Learn how to view running processes with the `ps`, `top` and other commands.

- discussion of how the system records or logs messages

Linux Kernel Processes

Kernel Processes

- A key function of the Linux kernel is to manage processes.
- The kernel accepts commands and manages processes that carry out those commands.
- The kernel gives commands access to devices like memory, disks, network interfaces, keyboards, mice, monitors and more.
- The kernel also provides access to information about active processes through a *pseudo filesystem* that is visible under the `/proc` directory. Other pseudo filesystems include `/dev` and `/sys`, which give information about hardware devices.

Pseudo filesystems are ones that appear to be real files on disk, but exist only in memory.

The /proc Directory

- The `/proc` directory not only contains information about processes (as name “proc” suggests), but also provides information about system hardware and current kernel configuration.

- The output shows a variety of named and numbered directories:

```
sysadmin@localhost:~$ ls /proc
1           cpuinfo      irq          modules      sys
128         crypto       kallsyms     mounts       sysrq-trigger
17          devices      kcore        mtrr         sysvipc
21          diskstats    key-users    net          thread-self
```

The /proc Directory

- Some of the commands that read from `/proc` include; `top`, `free`, `mount`, `unmount`.
- There are also important regular files in the `/proc` directory such as:
  - `/proc/cmdline` - Contains information passed to kernel during boot
  - `/proc/ meminfo` - Contains information about kernel memory usage
  - `/proc/modules` - Contains list of modules loaded into the kernel

Process Hierarchy

- When the kernel finishes loading during boot, it starts the *init* process and assigns it a PID of 1.
- This process then starts other system processes and assigns a PID in sequential order.
- When one process starts another process, the first process is called a *parent process*. The second process is called a *child process*.

Process Hierarchy

- Processes can be mapped into a “tree” which can be viewed with the `pstree` command.

```
sysadmin@localhost:~$ pstree
init-+-cron


     |-login---bash---pstree
     |-named---18*[{named}]
     |-rsyslogd---2*[{rsyslogd}]
     `-sshd
```

Viewing Process Snapshot

- Another way of viewing processes is with the `ps` command.
- By default, `ps` will only show running processes.
- The `ps` command can also be used with the `head` and `grep` commands to filter processes displayed:

```
sysadmin@localhost:~$ ps -e | grep firefox
 6090 pts/0    00:00:07 firefox
```

Viewing Processes in Real Time

- The `top` command has a dynamic, screen-based interface that will regularly update the output of running processes.

```
sysadmin@localhost:~$ top
```

```
top - 16:58:13 up 26 days, 19:15,  1 user,  load average: 0.60, 0.74, 0.60
Tasks:   8 total,   1 running,   7 sleeping,   0 stopped,   0 zombie
Cpu(s):  6.0%us,  2.5%sy,  0.0%ni, 90.2%id,  0.0%wa,  1.1%hi,  0.2%si,  0.0%st
Mem:  32953528k total, 28126272k used,  4827256k free,     4136k buffers
Swap:        0k total,        0k used,        0k free, 22941192k cached

  PID USER      PR   NI VIRT RES  SHR  S %CPU %MEM     TIME+ COMMAND
    1 root      20    0 17872 2892 2640 S    0  0.0   0:00.02 init
   17 syslog    20    0  171m 2768 2392 S    0  0.0   0:00.20 rsyslogd
```

Viewing Memory

- To view a snapshot of the memory used at that moment, use the `free` command:

```
sysadmin@localhost:~$ free
             total       used       free     shared    buffers     cached
Mem:      32953528   26171772    6781756          0       4136   22660364
-/+ buffers/cache:    3507272   29446256
Swap:            0          0          0
```

  - The output above explained:
    - `Mem:` is the statistics for physical memory on the system
    - `-/+ buffers/cache:` is the physical memory minus memory used by the kernel
    - `Swap:` is virtual memory

Log Files

- Processes running on a system produce output that describes what the process is doing.
- Some output goes to the terminal, however other output is not seen in the terminal and gets written to files as *log messages (*or *log data)* instead.
- Some processes log data by default, while others use a daemon to log data.
  - Examples of daemons include; `syslogd`, `klogd`, `rsyslogd`, `journald`
- Log files are placed under the `/var/log` directory.

Log Files

- To view log files:
  - Use `cat` or `less` command
  - Use `journalctl` command

- Log files are *rotated*, meaning older log files are renamed and replaced with newer log files.

- Most log files contain text, which can be viewed safely with many tools. Other files such as the `/var/log/btmp` and `/var/log/wtmp` files contain binary. Use the `file` command to view binary log files.

Kernel Messages

- Kernel messages can be found in the following files:
  - `/var/log/dmesg` - contains the kernel messages that were produced during system startup.
  - `/var/log/messages` - will contain kernel messages that are produced as the system is running.
- To view messages generated by the kernel, use the `dmesg` command. To filter the output, use a pipe with the `less` or `grep` command:

```
sysadmin@localhost:~$ dmesg | grep -i usb
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
```

Filesystem Hierarchy Standard

Filesystem Hierarchy Standard

- **Filesystem Hierarchy Standard (FHS)** is a set of standards supported by the Linux Foundation.
- FHS categorizes system directories as:
  - Shareable / Not shareable
  - Static / Variable
- The FHS standard defines four hierarchies of directories used in organizing the files of the filesystem:
  - Top-level hierarchy: `/`
  - Second-level hierarchy: `/usr`
  - Third-level hierarchy: `/usr/local`
  - Fourth-level hierarchy: `/var`

Organization Within the Filesystem Hierarchy

- **User and Home Directories**: The `/home` directory will typically have a directory underneath it for each user account.

- **Binary Directories**: Contains the programs that users and administrators execute to start processes or applications running on the system.
  - Includes `/bin`, `/usr/bin`, `/usr/local/bin` and other non-user specific directories.

- **Root Restricted Binaries**: the sbin directories are primarily intended to be used by the system administrator (the root user) and include:
  - `/sbin`, `/usr/sbin`, and `/usr/local/sbin`

Organization Within the Filesystem Hierarchy

- **Software Application Directories:**
  - Microsoft Windows - Applications files are installed in a single subdirectory under the `C:\Program Files` directory.
  - Linux - Applications may have files in multiple directories spread out throughout the Linux filesystem.
  - To view list of application files, use `dpkg -L packagename` (Debian) and `rpm -ql packagename` (Red Hat).

- **Library Directories:** Files which contain code that is shared between multiple programs.
  - Commonly use file extension of `.so`
  - Examples include: `/lib`, `/lib64`, `/usr/lib`, `/usr/lib64`, `/usr/local/lib`

Organization Within the Filesystem Hierarchy

- **Variable Data Directories:** The `/var` directory and many of its subdirectories can contain data that will change frequently.
    - Examples include: `/var/mail`, `/var/spool/mail`, `/var/spool/cups`