File system hierarchy in a common Linux distribution, file and device

permissions, study of system configuration files in /etc, familiarizing log files

for system events, user activity, network events.

File System Hierarchy

The Linux File Hierarchy Structure or the Filesystem Hierarchy Standard (FHS)

defines the directory structure and directory contents in Unix-like operating

systems. It is maintained by the Linux Foundation.

● In the FHS, all files and directories appear under the root directory /even if they

are stored on different physical or virtual devices.

● Some of these directories only exist on a particular system if certain subsystems,

such as the X Window System, are installed.

● Most of these directories exist in all UNIX operating systems and are generally

used in much the same way; however, the descriptions here are those used

specifically for the FHS and are not considered authoritative for platforms other

than Linux.

1. / (Root): Primary hierarchy root and root directory of the entire file system

hierarchy.

● Every single file and directory start from the root directory

● The only root user has the right to write under this directory

● /root is the root user’s home directory, which is not the same as /

2. /bin: Essential command binaries that need to be available in single-user

mode; for all users, e.g., cat, ls, cp.

● Contains binary executables

● Common Linux commands you need to use in single-user modes are located under

this directory.

● Commands used by all the users of the system are located here e.g., ps, ls, ping,

grep, cp

3. /boot : Boot loader files, e.g., kernels, initrd.

● Kernel initrd, vmlinux, grub files are located under /boot

● Example: initrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic

4. /dev : Essential device files, e.g., /dev/null.

These include terminal devices, usb, or any device attached to the system.

● Example: /dev/tty1, /dev/usbmon0

5. /etc : Host-specific system-wide configuration files.

● Contains configuration files required by all programs.

● This also contains startup and shutdown shell scripts used to start/stop individual

programs.

● Example: /etc/resolv.conf, /etc/logrotate.conf.

6. /home : Users’ home directories, containing saved files, personal settings,

etc.

● Home directories for all users to store their personal files.

● example: /home/kishlay, /home/kv

7. /lib : Libraries essential for the binaries in /bin/ and /sbin/.

● Library filenames are either ld\* or lib\*.so.\*

● Example: ld-2.11.1.so, libncurses.so.5.7

8. /media : Mount points for removable media such as CD-ROMs (appeared in

FHS-2.3).

● Temporary mount directory for removable devices.

● Examples, /media/cdrom for CD-ROM; /media/floppy for floppy drives;

/media/cdrecorder for CD writer

9. /mnt : Temporarily mounted filesystems.

● Temporary mount directory where sysadmins can mount filesystems.

10. /opt : Optional application software packages.

● Contains add-on applications from individual vendors.

● Add-on applications should be installed under either /opt/ or /opt/ sub-directory.

11. /sbin : Essential system binaries, e.g., fsck, init, route.

● Just like /bin, /sbin also contains binary executables.

● The Linux commands located under this directory are used typically by system

administrator, for system maintenance purpose.

● Example: iptables, reboot, fdisk, ifconfig, swapon

12. /srv : Site-specific data served by this system, such as data and scripts for

web servers, data offered by FTP servers, and repositories for version

control systems.

● srv stands for service.

● Contains server specific services related data.

● Example, /srv/cvs contains CVS related data.

13. /tmp : Temporary files. Often not preserved between system reboots, and

may be severely size restricted.

● Directory that contains temporary files created by system and users.

● Files under this directory are deleted when system is rebooted.

14. /usr : Secondary hierarchy for read-only user data; contains the majority of

(multi-)user utilities and applications.

● Contains binaries, libraries, documentation, and source-code for second level

programs.

● /usr/bin contains binary files for user programs. If you can’t find a user binary

under /bin, look under /usr/bin. For example: at, awk, cc, less, scp

15. /proc : Virtual filesystem providing process and kernel information as files.

In Linux, corresponds to a procfs mount. Generally, automatically

generated and populated by the system, on the fly.

● Contains information about system process.

● This is a pseudo filesystem contains information about running process. For

example: /proc/{pid} directory contains information about the process with that

particular pid.

● This is a virtual filesystem with text information about system resources. For

example: /proc/uptime

File permissions in Linux

Every file and directory in Linux have the following three permissions for all the

three kinds of owners:

Permissions for files

● Read – Can view or copy file contents

● Write – Can modify file content

● Execute – Can run the file (if its executable)

Permissions for directories

● Read – Can list all files and copy the files from directory

● Write – Can add or delete files into directory (needs execute permission as well)

● Execute – Can enter the directory.

Each letter denotes a particular permission:

● r: Read permission

● w: Write permission

● x: Execute permission

● –: No permission set

Device Permissions on Linux

When using ykman on Linux, you may find that the tool is sometimes unable to

access your YubiKey for some of the commands. This is often due to USB device

permissions, and can be tested by running the same ykman command as root.

The YubiKey is accessed in several different ways, depending on which command

is invoked.

Smart Card Access

For smart card-based applications, or when accessing a YubiKey over NFC, the

access is done via pcscd, the PC/SC Smart Card Daemon. It’s usually enough to

have pcscd installed and running for this to work. Smart card access is required

for the piv, oath, and openpgp commands, as well as for any command issued over

NFC.

Keyboard Access

The Yubico OTP application is accessed via the USB keyboard interface.

Permission is typically granted using udev, via a rules file. You can find an

example udev rules file which grants access to the keyboard interface here.

Keyboard access is required for the otp command.

FIDO Access

The FIDO protocols are accessed via a USB HID interface. As with keyboard

access, permission is granted through udev. You can find an example udev rules

file which grants access to a large number (not just YubiKeys) of FIDO devices

here. FIDO access is required for the fido command