

Linux System Administration

Core System Administration Concepts



Linux System
Administration

Superuser

- The Linux Superuser is defined as UID 0
- The Linux Superuser's name is root by default, but this is changeable.
 - Any user with user ID #0 would have superuser rights and privileges.
- Most system administration tasks require superuser privileges.



Superuser Rights and Privileges

- Some command can be executed only by superuser.
- Superuser can read and write to any file.
- Superuser can execute any file providing executable permissions are set for some user, group, or the world.



- Many administrative commands used primarily by superuser are in the /sbin and /usr/sbin directories.
- The /sbin and /usr/sbin directories appear in roots directory path, but not in average users' directory paths.



Attaining Superuser privileges

- There are several ways of attaining superuser rights and privileges.
 - Bring the system into Single User Mode.
 - run level 1
 - Log onto the system as root.
 - Execute the su command while logged in as a normal user permitted to su to root.
 - Use the sudo command to particular commands with superuser privileges.



Principle of Least Privilege

- NEVER do as superuser what you can do as a less-privileged user.
- Work as superuser when necessary, the revert to an ordinary user as soon as practically possible.
- When ordinary users make mistakes, it's seldom critical, but when superuser makes a mistake, it is often CATASTROPHIC!



suid programs

- suid programs are identified by an s in the execute permission position in ls -l
- -r-s--x--x 1 root root 17700 Jun 25 14:38 /sbin/passwd
- suid programs run with the full rights and privileges of the file's owner for the duration of the program.
- suid programs owned by root can be very dangerous.



- Some programs will ask for for the root password in order to perform certain functions, such as the gui system configuration utilities.
- These programs give up superuser privileges when the program completes execution.



Remote Access

- NEVER login remotely as root via an unencrypted protocol like telnet, rlogin, rsh, etc. unless all networks between you and your system are known to be extremely secure.
- Use ssh for remote access instead.



Secure Shell

- ssh is a better and more secure alternative to telnet, etc.
- Provides encryption, etc.
- Configurable to use RSA keys/passphrases or standard passwords.
- Can also be used to create encrypted “tunnels” for other protocols.



su command

- su can be used to create a shell as any valid user if you are superuser or know the target user's password.
- su *user* - if no user, root is assumed.
- Important options
 - -c *command*, --command=*command*
 - Duration of shell is execution of *command*
 - -, -l creates a login shell, inheriting the new users' environment.



Processes

- Every program, command and shell on a Linux system runs as a process.
- Think of a process as a job or task that the computer executes.
- The ps command displays information about processes running on the system.
- The top command displays information about the top running processes in terms of system resources.



Linux Processes

- Linux systems are multi-processing systems.
 - Many processes co-exist and run simultaneously.
- The Linux kernel arranges all waiting processes in a queue and provides each in turn with a “slice” of CPU execution time.
 - This is called scheduling.



Linux Processes

- Linux processes can be single-threaded or multi-threaded.
 - Single-threaded processes have one thread of control.
 - Multi-threaded processes have more than one thread of control, and therefore can multitask within the process.



Processes and Priority

- Linux process have priorities associated with them.
- Higher priority processes will be allocated more execution time by the scheduler than lower priority processes.
- Process priorities can be manipulated by the nice command.
- “Nice” processes run as a lower priority.
 - The higher the nice value, the lower the priority



kill command

- *kill* sends a signal to a running process.
- `kill -l` lists all valid signals
- Important signals
 - SIGHUP - Hangup - Useful for forcing processes to reread config files, etc.
 - SIGINT - Interrupt - Same effect as the CONTROL-c interrupt key.
 - SIGTERM - Terminate - Terminates process.
 - SIGKILL - Kill - Terminates the target process.



Trapping Signals

- Programs may “trap” and “handle” signals
- A signal that is trapped and handled by a process generally will not terminate the process, but will be handled in some way defined by the program code.
- The SIGKILL signal CANNOT be trapped, and so always terminates the process.



ps tree command

- ps tree displays running processes as a tree structure.
- Shows the parent/child relationships between processes.
- init is the first process to run when the system boots.
- All other processes are descended from init.



ps command

- The ps command displays information about running processes.
- ps has many options and formats.
 - ps -e shows all running processes.
 - ps -f displays process ID, parent process ID, process start time, process cumulative running time, associated terminal if any, etc.



killall and pidof commands

- killall kills all processes with a specified command name.
- pid of returns the process ID(s) of all processes with a specified command name.



Shutting Down the System

- `/sbin/shutdown` *option time*
- Options
 - `-r` reboots the system
 - `-h` shuts down (halts) the system.
- `halt` is an alternate way to shut down the system.
- `reboot` is an alternate way to reboot the system.



Linux Run Levels

- 0 Halt
- 1 Single User
- 3 Multiuser
- 5 Multiuser with X (Graphical Login)
- 6 Reboot
- The default system run level is controlled by the *initdefault* directive in `/etc/inittab`.
- `telinit` puts the system into a specific run level.



The Boot Process

- When the system boots up, the first script to run is `/etc/rc.d/sysinit`.
 - Performs basic system configuration
 - Sets system clock, hostname, sets up swap, mounts filesystems, etc.
- Next, the `/etc/rc.d/rc.nd` scripts for the appropriate run level are run.
- Init scripts are run in order by number.



Init Startup Script

- The scripts in the `/etc/rc.d/rcn.d` directories are symbolic links to the service init scripts in `/etc/rc.d/init.d`.
- These links can be created/deleted manually to activate and deactivate services upon boot.
- Or can better be managed with *chkconfig*
 - *S* scripts run on startup, *K* scripts on shutdown



chkconfig

- chkconfig - manages boot configuration of system services and daemons.
- Important options
 - --list - Lists services and their states
 - --levels - Configures services



Xinetd Super Server

- xinetd listens for network connections and launches the appropriate service in response.
- xinetd is configured by the `/etc/xinetd.conf` file and `/etc/xinet.d` directory.
- xinetd has a sophisticated set of service configuration and access options.



xinetd

- Global configuration directives go in xinetd's configuration file, `xinetd.conf`.
 - These apply to xinetd itself, or to all xinetd services.
- Individual service configuration directives go in the individual file for the service in xinetd's service directory, `/etc/inetd.d`



Rescue Mode

- Rescue mode is a special environment that can be used to repair a system that will not boot.
- To boot into rescue mode, boot from the Linux installation CD and type “linux rescue” at the “boot:” prompt.



nmap

- nmap is a network scanning tool that reports on services running and ports open on local or remote systems.
- nmap is very useful for letting you see what your system looks like to others on remote systems and remote networks.

