UNIX / LINUX - SYSTEM LOGGING

In this chapter, we will discuss in detail about system logging in Unix.

Unix systems have a very flexible and powerful logging system, which enables you to record almost anything you can imagine and then manipulate the logs to retrieve the information you require.

Many versions of Unix provide a general-purpose logging facility called **syslog**. Individual programs that need to have information logged, send the information to syslog.

Unix *syslog* is a host-configurable, uniform system logging facility. The system uses a centralized system logging process that runs the program **/etc/syslogd** or **/etc/syslog**.

The operation of the system logger is quite straightforward. Programs send their log entries to *syslogd*, which consults the configuration file **/etc/syslogd.conf** or **/etc/syslog** and, when a match is found, writes the log message to the desired log file.

There are four basic syslog terms that you should understand –

Sr.No.	Term & Description
1	**Facility** The identifier used to describe the application or process that submitted the log message. For example, mail, kernel, and ftp.
2	**Priority** An indicator of the importance of the message. Levels are defined within syslog as guidelines, from debugging information to critical events.
3	**Selector** A combination of one or more facilities and levels. When an incoming event matches a selector, an action is performed.
4	**Action** What happens to an incoming message that matches a selector — Actions can write the message to a log file, echo the message to a console or other device, write the message to a logged in user, or send the message along to another syslog server.

Syslog Facilities

We will now understand about the syslog facilities. Here are the available facilities for the selector. Not all facilities are present on all versions of Unix.

Facility	Description
1	**auth** Activity related to requesting name and password (getty, su, login)
2	**authpriv** Same as auth but logged to a file that can only be read by selected users
3	**console** Used to capture messages that are generally directed to the system console
4	**cron** Messages from the cron system scheduler
5	**daemon** System daemon catch-all
6	**ftp** Messages relating to the ftp daemon
7	**kern** Kernel messages
8	**local0.local7** Local facilities defined per site
9	**lpr** Messages from the line printing system
10	**mail** Messages relating to the mail system
11	**mark**

	Pseudo-event used to generate timestamps in log files
12	**news** Messages relating to network news protocol (nntp)
13	**ntp** Messages relating to network time protocol
14	**user** Regular user processes
15	**uucp** UUCP subsystem

Syslog Priorities

The syslog priorities are summarized in the following table −

Sr.No.	Priority & Description
1	**emerg** Emergency condition, such as an imminent system crash, usually broadcast to all users
2	**alert** Condition that should be corrected immediately, such as a corrupted system database
3	**crit** Critical condition, such as a hardware error
4	**err** Ordinary error
5	**Warning** Warning

6	**notice** Condition that is not an error, but possibly should be handled in a special way
7	**info** Informational message
8	**debug** Messages that are used when debugging programs
9	**none** Pseudo level used to specify not to log messages

The combination of facilities and levels enables you to be discerning about what is logged and where that information goes.

As each program sends its messages dutifully to the system logger, the logger makes decisions on what to keep track of and what to discard based on the levels defined in the selector.

When you specify a level, the system will keep track of everything at that level and higher.

The /etc/syslog.conf file

The **/etc/syslog.conf** file controls where messages are logged. A typical **syslog.conf** file might look like this −

```
*.err;kern.debug;auth.notice /dev/console
daemon,auth.notice           /var/log/messages
lpr.info                     /var/log/lpr.log
mail.*                       /var/log/mail.log
ftp.*                        /var/log/ftp.log
auth.*                       @prep.ai.mit.edu
auth.*                       root,amrood
netinfo.err                  /var/log/netinfo.log
install.*                    /var/log/install.log
*.emerg                      *
*.alert                      |program_name
mark.*                       /dev/console
```

Each line of the file contains two parts −

- A **message selector** that specifies which kind of messages to log. For example, all error messages or all debugging messages from the kernel.
- An **action field** that says what should be done with the message. For example, put it in a file or send the message to a user's terminal.

Following are the notable points for the above configuration −

- Message selectors have two parts: **a facility** and **a priority**. For example, *kern.debug* selects all debug messages (the priority) generated by the kernel (the facility).
- Message selector *kern.debug* selects all priorities that are greater than debug.

- An asterisk in place of either the facility or the priority indicates "all". For example, ***.debug** means all debug messages, while **kern.*** means all messages generated by the kernel.
- You can also use commas to specify multiple facilities. Two or more selectors can be grouped together by using a semicolon.

Logging Actions

The action field specifies one of five actions −

- Log message to a file or a device. For example, **/var/log/lpr.log** or **/dev/console**.
- Send a message to a user. You can specify multiple usernames by separating them with commas; for example, root, amrood.
- Send a message to all users. In this case, the action field consists of an asterisk; for example, *.
- Pipe the message to a program. In this case, the program is specified after the Unix pipe symbol (|).
- Send the message to the syslog on another host. In this case, the action field consists of a hostname, preceded by an at sign; for example, @tutorialspoint.com.

The logger Command

Unix provides the **logger** command, which is an extremely useful command to deal with system logging. The **logger** command sends logging messages to the syslogd daemon, and consequently provokes system logging.

This means we can check from the command line at any time the **syslogd** daemon and its configuration. The logger command provides a method for adding one-line entries to the system log file from the command line.

The format of the command is −

```
logger [-i] [-f file] [-p priority] [-t tag] [message]...
```

Here is the detail of the parameters −

Sr.No.	Option & Description
1	**-f filename** Uses the contents of file filename as the message to log.
2	**-i** Logs the process ID of the logger process with each line.
3	**-p priority** Enters the message with the specified priority (specified selector entry); the message priority can be specified numerically, or as a facility.priority pair. The default priority is user.notice.
4	**-t tag** Marks each line added to the log with the specified tag.

5	**message** The string arguments whose contents are concatenated together in the specified order, separated by the space.

You can use Manpage Help to check complete syntax for this command.

Log Rotation

Log files have the propensity to grow very fast and consume large amounts of disk space. To enable log rotations, most distributions use tools such as ***newsyslog*** or ***logrotate***.

These tools should be called on a frequent time interval using the **cron daemon**. Check the man pages for *newsyslog* or *logrotate* for more details.

Important Log Locations

All the system applications create their log files in ***/var/log*** and its sub-directories. Here are few important applications and their corresponding log directories –

Application	Directory
httpd	/var/log/httpd
samba	/var/log/samba
cron	/var/log/
mail	/var/log/
mysql	/var/log/