Consider the following example:

```
# cat /etc/rsyslog.d/myConfig
local7.* /var/log/local7
# cd /etc/init.d
# ./syslogd restart
# logger -p local7.info A line to be placed in /var/log/local7
```

3. The `-f` option will log the lines from another file:

```
$ logger -f /var/log/source.log
```

See also

- The *Using head and tail for printing the last or first 10 lines* recipe of `Chapter 3`, *File In, File Out*, explains the head and tail commands

Managing log files with logrotate

Log files keep track of events on the system. They are essential for debugging problems and monitoring live machines. Log files grow as time passes and more events are recorded. Since the older data is less useful than the current data, log files are renamed when they reach a size limit and the oldest files are deleted.

Getting ready

The `logrotate` command can restrict the size of the log file. The system logger facility appends information to the end of a log file without deleting earlier data. Thus a log file will grow larger over time. The `logrotate` command scans log files defined in the configuration file. It will keep the last 100 kilobytes (for example, specified S*IZE = 100 k*) from the log file and move the rest of the data (older log data) to a new file `logfile_name.1`. When the old-data file (`logfile_name.1`) exceeds `SIZE`, `logrotate` renames that file to `logfile_name.2` and starts a new `logfile_name.1`. The `logrotate` command can compress the older logs as `logfile_name.1.gz`, `logfile_name.2.gz`, and so on.

How to do it...

The system's `logrotate` configuration files are held in `/etc/logrotate.d`. Most Linux distributions have many files in this folder.

We can create a custom configuration for a log file (say `/var/log/program.log`):

```
$ cat /etc/logrotate.d/program
/var/log/program.log {
missingok
notifempty
size 30k
  compress
weekly
  rotate 5
create 0600 root root
}
```

This is a complete configuration. The `/var/log/program.log` string specifies the log file path. Logrotate will archive old logs in the same directory.

How it works...

The `logrotate` command supports these options in the configuration file:

Parameter	**Description**
`missingok`	This ignores if the log file is missing and return without rotating the log.
`notifempty`	This only rotates the log if the source log file is not empty.
`size 30k`	This limits the size of the log file for which the rotation is to be made. It can be 1 M for 1 MB.
`compress`	This enables compression with gzip for older logs.
`weekly`	This specifies the interval at which the rotation is to be performed. It can be weekly, yearly, or daily.
`rotate 5`	This is the number of older copies of log file archives to be kept. Since 5 is specified, there will be `program.log.1.gz`, `program.log.2.gz`, and so on up to `program.log.5.gz`.

`create 0600 root root`	This specifies the mode, user, and the group of the log file archive to be created.

The options in the table are examples of what can be specified. More options can be defined in the `logrotate` configuration file. Refer to the man page at `http://linux.die.net/man/8/logrotate`, for more information.

Monitoring user logins to find intruders

Log files can be used to gather details about the state of the system and attacks on the system.

Suppose we have a system connected to the Internet with SSH enabled. Many attackers are trying to log in to the system. We need to design an intrusion detection system to identify users who fail their login attempts. Such attempts may be of a hacker using a dictionary attack. The script should generate a report with the following details:

- User that failed to log in
- Number of attempts
- IP address of the attacker
- Host mapping for the IP address
- Time when login attempts occurred

Getting ready

A shell script can scan the log files and gather the required information. Login details are recorded in `/var/log/auth.log` or `/var/log/secure`. The script scans the log file for failed login attempts and analyzes the data. It uses the `host` command to map the host from the IP address.