Configure and Use System Log Files

Information

These notes were originally written in the year 2000 as part of a set of LPI Exam 101 training materials. The LPI training course at Bromley College was subsequently discontinued and some of the sections of the notes modified and incorporated into our one-day System Administration Courses. The remainder of the notes have now been made publicly available on the linuxtraining.org.uk website.

If you are a beginner please do not be put off of training courses by these notes, as they are rather technical. On the other hand if you are a more experienced Linux user we hope you find the coverage of this topic refreshingly clear.

For full details of our current Linux training please visit the site:
http://ce.bromley.ac.uk/linux

If you have reached this page from a search engine and wish to see the full contents list for the published notes please visit the site:
http://www.linuxtraining.org.uk

We hope you find these notes useful, but please remember that they apply to the 2.2 kernel. I will update them when I have the time.

Clive Gould - 21st December 2004

Configure and Use System Log Files

Objective 3

Configure and use system log files to meet administrative and security needs: Configure the type and level of information logged, manually scan log files for notable activity, arrange for automatic rotation and archiving of logs, track down problems noted in logs. Involves editing /etc/syslog.conf

Understanding Log Files

Many processes record an event log automatically. These logs can be invaluable in tracing system problems and security violations. Log files are typically stored in the directory **/var/log** and its associated subdirectories. When a log file reaches its maximum size it is renamed and the original file is overwritten with fresh log data. Log files can be distribution specific. Some of the main log files you may encounter are listed below:

Filename	***Description***
/var/log/boot.log	Contains information on all the processes started and stopped when the system starts up or shuts down. (A copy of what you see on the screen during boot or reboot). The contents of this log are controlled by the **syslogd** process.
/var/log/cron	Contains status messages from **cron**, a process which automatically runs scheduled jobs on a timed basis.
/var/log/dmesg	Contains messages recorded by the kernel during boot.
/var/log/httpd	This directory contains the access and error log files maintained by the Apache server
/var/log/lastlog	This file is similar to wtmp and contains login times for users. Can be accessed using the **lastlog** command. Used by finger to determine when a user was last logged on.
/var/log/maillog	Logs all mail messages in one place. The contents of this log are controlled by the syslogd process.
/var/log/messages	A general purpose log file to which many programs record messages. The contents of this log are controlled by the syslogd process.

Configure and Use System Log Files

Filename	***Description***
/var/log/news	This directory contains log files associated with the news server. Some of these log files are controlled by the syslogd process.
/var/log/secure	Records the date and time of local and remote logins and sessions. The contents of this log are controlled by the syslogd process.
/var/log/samba	This directory contains log files associated with the samba server, including machine, user and nmb logs.
/var/log/spooler	Contains mail and news errors of level err and higher. The contents of this log are controlled by the syslogd process.
/var/run/utmp	This is a binary file containing information on currently logged on users. Used by the **who**, **w** and **finger** commands. The format of this file may vary from one system to another.
/var/log/wtmp	This is a binary file containing log times and durations for each user. Can be accessed using the **last** command. The format of this file may vary from one system to another.

The commands **less**, **tail** and **grep** are very useful with log files

The use of the grep command with the secure log file is illustrated below:

```
[root@redhat log]# grep student1 secure.*
secure.4:Aug  2 10:14:43 redhat login: LOGIN ON 3 BY student1
FROM redhat
```

The use of the grep command with the messages log file is illustrated below:

```
[root@redhat log]# grep SCSI messages
Aug 29 10:26:47 redhat kernel: (scsi0) <Adaptec AHA-294X Ultra SCSI host adapter> found at PCI 9/0
Aug 29 10:26:47 redhat kernel: (scsi0) Wide Channel, SCSI ID=7, 16/255 SCBs
Aug 29 10:26:47 redhat kernel: (scsi0) in the Adaptec SCSI BIOS by hitting CTRL-A when prompted
Aug 29 10:26:47 redhat kernel: scsi0 : Adaptec AHA274x/284x/294x (EISA/VLB/PCI-Fast SCSI) 5.1.15/3.2.4
Aug 29 10:26:47 redhat kernel:        <Adaptec AHA-294X Ultra SCSI host adapter>
Aug 29 10:26:47 redhat kernel:   Type:   Direct-Access                    ANSI SCSI revision: 02
Aug 29 10:26:47 redhat kernel: SCSI device sda: hdwr sector= 512 bytes. Sectors= 17824700 [8703 MB] [8.7
GB]
```

Configure and Use System Log Files

The use of the tail command with the samba log file for machine rm113_10 is illustrated below:

```
[root@redhat samba]# tail log.rm113_10
[2000/07/06 12:07:48, 1] smbd/service.c:close_cnum(514)
  rm113_10 (172.16.32.119) closed connection to service netlogon
[2000/07/06 12:10:16, 1] smbd/service.c:close_cnum(514)
  rm113_10 (172.16.32.119) closed connection to service common
[2000/07/06 12:10:16, 1] smbd/service.c:close_cnum(514)
  rm113_10 (172.16.32.119) closed connection to service student10
[2000/07/06 13:31:48, 1] smbd/password.c:pass_check_smb(528)
  smb_password_check failed. Invalid password given for user 'student10'
[2000/07/06 13:47:50, 1] smbd/password.c:pass_check_smb(528)
  smb_password_check failed. Invalid password given for user 'alfredd'
```

Linux System Logging Utilities - syslogd

The **s**yslogd process logs selected kinds of system activity, such as error messages from the news server and warnings printed by the kernel. Syslogd runs as a daemon and is started in one of the rc files at boot time. Syslogd can also log the activity of remote hosts over a network.

The syntax for the syslogd process is shown below:

`syslogd` ***`option(s)`***

Common syslogd options are:

Option	*Explanation*
-f *CONFIG_FILE*	Specify an alternative *CONFIG_FILE* instead of /etc/syslog.conf, which is the default and is read at startup.
-m *INTERVAL*	The syslogd logs a mark timestamp regularly. The default interval between two marked lines is 20 minutes. This *INTERVAL* can be changed with this option. The interval is often set to to zero which turns this feature off entirely.

Note: There are a number of additional options available with syslogd, but these are concerned with remote logging and are beyond the scope of this course.

Configure and Use System Log Files

The file **/etc/syslog.conf** is used to control where syslogd records its information. Such a file might look like the following:

```
[root@redhat /root]# cat /etc/syslog.conf
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                         /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;news.none;authpriv.none    /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                      /var/log/secure

# Log all the mail messages in one place.
mail.*                                          /var/log/maillog

# Everybody gets emergency messages, plus log them on another
# machine.
*.emerg                                         *

# Save mail and news errors of level err and higher in a
# special file.
uucp,news.crit                      /var/log/spooler

# Save boot messages also to boot.log
local7.*                                        /var/log/boot.log

# INN

news.=crit                        /var/log/news/news.crit
news.=err                         /var/log/news/news.err
news.notice                       /var/log/news/news.notice
```

The first field of each line lists the kind of messages that should be logged, and the second field indicates where they should be logged to. The first field is of the format:

`facility.level;`*`facility.level....`*

Where facility is the system application or facility generating the message, and level is the level of severity. e.g. facility can be **mail** for the mail process, **kern** for the kernel, or **authpriv** for authentication programs such as login.
Listed in order of increasing severity, level can be : **debug, info, notice, warning, err, crit, alert or emerg.**

Configure and Use System Log Files

In the above example we can see that:

- All levels (*) of kernel messages are logged to the screen.
- All messages of level info and higher (.info), apart from those generated by mail, news and authentication (.none) are logged to /var/log messages.
- All levels of login authentication messages are logged to the file /var/log secure. For security this file is read and write only by the root.
- All levels of mail messages are logged to the file /var/log/maillog.
- All messages of level emerg are sent to all users (* in the second field).
- All news messages of just level crit (=crit) are logged to the file news.err

By default all messages of the specified level and higher are logged. The = is used to log only messages of the specified level and != can be used to exclude messages of a particular level from being logged.

The messages logged by syslogd usually contain the date, an indication of what process or facility delivered the message, and the message itself, all on a single line. An example of this taken from the /var/log/secure file is illustrated below:

```
[root@redhat log]# tail -2  /var/log/secure
Aug 31 13:52:27 redhat login: LOGIN ON tty1 BY clive
Aug 31 13:53:11 redhat login: ROOT LOGIN ON tty1
```

Log files can be important in tacking down system problems. If a log file grows too large you can delete it using rm. It will automatically be recreated by syslogd.

Your distribution probably comes equipped with a running syslogd and a properly configured /etc/syslog.conf file. However it is important to know where your log files are and what programs they represent. For example, you might want to log debug level messages from the kernel, which can be very verbose, by adding an appropriate line in /etc/syslog.conf. After you have done this you will need to send the syslogd process the -HUP signal to tell it to re-read its configuration file.

Configure and Use System Log Files

Rotates, Compresses, and Mails System Logs - logrotate

The logrotate command is designed to ease administration of systems that generate large numbers of log files. It allows automatic rotation, compression, removal, and mailing of log files. Each log file may be handled daily, weekly, monthly, or when it grows too large.

Normally, logrotate is run as a daily cron job and thus reads its configuration file(s) daily. Any number of configuration files may be specified on the command line. Later configuration files may override the options given in earlier files, so the order in which the logrotate configuration files are listed in is important.

An example of an **/etc/logrotate.conf** configuration file is illustrated below:

```
[root@redhat samba]# cat /etc/logrotate.conf
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# send errors to root
errors root

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own lastlog or wtmp -- we'll rotate them here
/var/log/wtmp {
    monthly
    create 0664 root utmp
    rotate 1

/var/log/lastlog {
    monthly
    rotate 1

# system-specific logs may be configured here
```

Configure and Use System Log Files

Both global options, which apply to all log files, and local options, which apply to specific log files, can be set in the /etc/logrotate.conf file. Any local options override global ones. Directives used in the above logrotate.conf file are explained in the table below:

Directive	*Explanation*
weekly	This global option ensures that all log files are rotated once a week.
rotate *COUNT* rotate 4	This global option ensures that all logs are rotated *COUNT* times before being removed or mailed. If count is 0, old versions are removed rather then rotated.
errors *ADDRESS* errors root	This global option ensures that any errors that occur during log file processing are mailed to the given *ADDRESS*
create *MODE OWNER GROUP* create	This global option controls the permissions, owner and group which will be applied to all new log files. If these are omitted the log file will assume the attributes of the original log file.
compress #compress	If this global option is enabled it causes all old versions of log files to be compressed with gzip.
include include /etc/logrotate.d	Includes additional logrotate configuration files containing local settings. This is commonly used to specify logrotate directives for individual applications.

The options associated with the sections of the file dealing with /var/log/wtmp and /var/log/lastlog are local rather than global. Both of these are rotated monthly rather than weekly and only one old log file is kept.

Below you can see a couple of log files maintained on a typical system:

```
-rw-------   1 root  root        828 Aug 31 15:42 secure
-rw-------   1 root  root          0 Aug 20 04:02 secure.1
-rw-------   1 root  root        299 Aug 18 17:47 secure.2
-rw-------   1 root  root        253 Aug 11 09:29 secure.3
-rw-------   1 root  root        266 Aug  2 10:15 secure.4
-rw-rw-r--   1 root  utmp       1152 Sep  1 10:16 wtmp
-rw-rw-r--   1 root  utmp     246144 Aug 31 16:03 wtmp.1
```

You can see that the secure log file is rotated four times, on a weekly basis, before being removed, whereas the wtmp log file is only rotated once, on a monthly basis, before removal.

Configure and Use System Log Files

Archiving Logs - Prerotate and Postrotate

You need to decide how long you want to keep old log files for. In some instances, such as the log files for the Apache server, you might want to keep six months worth of log files, or your organisation might want to keep records of when users log in or log out for several years.

One way to keep logs for long periods of time would just be to increase the count. However this could easily fill up your disk, especially if you want to keep several years worth of logs. This can be solved by moving log files to nearline or offline storage.

When logrotate rotates a particular log it automatically discards the oldest log and renames all the existing log files. By placing lines containing appropriate shell command(s) between the prerotate and endscript directives in your logrotate.conf configuration file, you can move your oldest log to somewhere safe before the logs are rotated and the log is overwritten and it contents lost.

By placing a shell command(s) between the postrotate and endscript directives in your logrotate.conf configuration file, you can execute the shell command after the logs have been rotated. This is typically used to tell a process, such as syslogd or httpd, to re-read its configuration file after the logs have been rotated. An example of use of postrotate is illustrated below:

```
[clive@redhat clive]$ cd /etc/logrotate.d
[clive@redhat logrotate.d]$ cat cron
/var/log/cron {
    missingok   #Go on to next log file is this one is empty
    notifempty  #Do not compress an empty log file
    postrotate
            /usr/bin/killall -HUP crond
    endscript
```