Linux logs

Article URL
Linux logs

Author
SsecurityHome.eu

Published: 26 May 2008

Last updated on 31 July 2008.

Â

Linux has a dedicated directory for logs: /var/log
The files and directories located there might differ depending on which distribution of linux, and what programs you have installed.

The logging is provided by 2 daemons:
syslogd and klogd

The following files are the most common ones.
auth.log

This file logs all authentication info. Failed logins and the successfull logins.

daemon.log

Most daemons(services) should log to this.

dmesg

This file is written durring boot. It contains messages form the kernel raised during boot time.

messages

This is the standard system log file, which contains messages from all your system software.
-- MARK -- is written to this file every 20 min, so you can see when the system went down

You can read the with any text-reader (less, more, ... even vi if you want to).
But most of the time you'll only need the last few lines.
This can be done with tail.
You can use follow, to get the messages scrolling in your console.

tail -f /var/log/messages
(ctrl+C to quit)

grep is also a handy tool.

Rotating logs.
After a while your logs can get pretty full.
So it is best to rotate them, so the logs files don't get to big.

Depending on wich flavor your linux is based, you have several ways to rotate them.
Debian based systems (Ubunto, gentoo, ...)
Sys
rotate

Redhat based systems (Fedora, Centos, ...)
rotate for everything

syslog

What files are handled by syslog? We can find out what are those files simply by inspecting the syslog configuration file (/etc/syslog.conf)

```
#  /etc/syslog.conf     Configuration file for syslogd.
#
#                    For more information see syslog.conf(5)
#                    manpage.

#
# First some standard logfiles.  Log by facility.
#

auth,authpriv.*                 -/var/log/auth.log
*.*;auth,authpriv.none          -/var/log/syslog
#cron.*                         -/var/log/cron.log
daemon.*                        -/var/log/daemon.log
kern.*                          -/var/log/kern.log
lpr.*                           -/var/log/lpr.log
mail.*                          -/var/log/mail.log
user.*                          -/var/log/user.log
uucp.*                          -/var/log/uucp.log

#
# Logging for the mail system.  Split it up so that
# it is easy to write scripts to parse these files.
#
mail.info                       -/var/log/mail.info
mail.warn                       -/var/log/mail.warn
mail.err                        -/var/log/mail.err

# Logging for INN news system
#
news.crit                       -/var/log/news/news.crit
news.err                        -/var/log/news/news.err
news.notice                     -/var/log/news/news.notice

#
```

```
# Some `catch-all' logfiles.
#
*.=debug;
        auth,authpriv.none;
        news.none;mail.none     -/var/log/debug
*.=info;*.=notice;*.=warn;
        auth,authpriv.none;
        cron,daemon.none;
        mail,news.none          -/var/log/messages

#
# Emergencies are sent to everybody logged in.
#
*.emerg                         *

#
# I like to have messages displayed on the console, but only on a virtual
# console I usually leave idle.
#
#daemon,mail.*;
#       news.=crit;news.=err;news.=notice;
#       *.=debug;*.=info;
#       *.=notice;*.=warn       /dev/tty8

# The named pipe /dev/xconsole is for the `xconsole' utility.  To use it,
# you must invoke `xconsole' with the `-file' option:
#
#    $ xconsole -file /dev/xconsole [...]
#
# NOTE: adjust the list below, or you'll go crazy if you have a reasonably
#       busy site..
#
daemon.*;mail.*;
        news.crit;news.err;news.notice;
        *.=debug;*.=info;
        *.=notice;*.=warn       |/dev/xconsole
```

You can change this add or remove log-files as you whish.

The rotating is done by *syslog*

and *savelog*

on a daily and weekly base.
The daily rotation is done by */etc/cron.daily/sysklogd*

* any files that contains the *.* facility in the syslog configuration are rotated daily. The reason for this, is that they will log all the information regardless of the facility, and can become quite quickly very big.

* if we will look inside the daily syslog cron we will see that it finds the logs it need to rotate by launching the file syslogd-listfiles:
/usr/sbin/syslogd-listfiles
/var/log/syslog /dev/null

So we can see here that by default my debian system will keep 7 archives of previous logs (7 days). If I would want to change this, then all I have to do is to replace the -c 7 parameter with what I need.

The weekly rotation is done by */etc/cron.weekly/sysklogd*

logrotate

Logrotate is an application used to rotate all other log files not handled by syslog itself (or if syslog is not installed). 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. LetÃ‚Â’s look into the script that was installed in /etc/cron.daily for this:

```
less /etc/cron.daily/logrotate
#!/bin/sh

test -x /usr/sbin/logrotate || exit 0
/usr/sbin/logrotate /etc/logrotate.conf
```

Logrotate will look into /etc/logrotate.conf for its configuration directives.

```
less /etc/logrotate.conf
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

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

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

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

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

```
}

/var/log/btmp {
missingok
monthly
create 0664 root utmp
rotate 1
}

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

So we can see it defines some default parameters (weekly, rotate 4, create, compress) and includes all the files from /etc/logrotate.d/. Also it defines the rotation for some files that are not handled by syslog itself, like wtmp. For example, I would want to keep more than one month of old wtmp logs, then I would have to change the parameter rotate 1.

Inside the /etc/logrotate.d/ various packages will install their own configuration file that will ensure their logs are properly rotate (on my fresh Debian install I have the following files: acpid apache2 aptitude base-config dpkg exim4-base). As long as you donÃ‚Â’t change the paths to those logs the rotation will work out of the box. But in case you change them you might want to look inside this folder and make the proper adjustments to the log file definitions, to assure they will be rotated. For example, letÃ‚Â’s look at the apache rotation file created here by the apache2 package:

```
less /etc/logrotate.d/apache2
/var/log/apache2/*.log {
weekly
missingok
rotate 52
compress
delaycompress
notifempty
create 640 root adm
sharedscripts
postrotate
if [ -f /var/run/apache2.pid ]; then
/etc/init.d/apache2 restart > /dev/null
fi
endscript
}
```

We can see that by default it will rotate apache logs found in /var/log/apache2/ that have the extension *.log, on a weekly basis and keep 52 archives (about 1 year) of the old data. Once the rotation is completed it will restart the apache daemon. You can check logrotate manual page for all the available parameters, as they are self-explanatory.
Now, if I would like to keep my own apache log files in a different location (/var/weblogs for example) and rotate them monthly then I will need to make the following changes:

```
/var/weblogs/*.log {
monthly
...
```

Probably, I will also want to change the default hour when the daily cron is running to have it on midnight. Anyway this is just an example and you will most certainly configure this based on your needs.

Even though I didnÃ‚Â’t intended with this article to describe what each configuration parameter of logrotate means (as you can easily find out yourself), but to show what is the logic and its functionality, I would like to add that while configuring and testing this you might find very useful the debug option:

```
logrotate -d file
```

This will show you what it will do, without actually rotating anything, and this is most valuable while testing complex setups that you donÃ‚Â’t want to Ã‚Â‘playÃ‚Â’ with the logs to see if your configuration will work as you want it.

Also logrotate -f file will force the rotation even if that would have normally not occurred (logrotate will only assume it need to run and rotate logs once per day).

Note: RedHat based systems will also rotate by default the Ã‚Â’system logsÃ‚Â’ using logrotate and not syslogÃ‚Â’s internal method as Debian systems. This is handled by default with the logrotate configuration file:

```
less /etc/logrotate.d/syslog
/var/log/messages /var/log/secure /var/log/maillog /var/log/spooler /var/log/boot.log /var/log/cron {
sharedscripts
postrotate
/bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true
endscript
}
```

The sharedscripts parameter means that the postrotate script will only be run once (after the old logs have been compressed), not once for each log which is rotated.
So nothing special defined here, besides the log files that will be rotated, and it will use the defaults from /etc/logrotate.conf.