# 1.111.3
# Configure and use system log files to meet administrative and security needs
# Weight 3

Linux Professional Institute Certification — 102

Nick Urbanik <nicku@nicku.org>



2005 November

**Outline**

## Contents

**1 Context** **2**

**2 Objectives** **2**

**3 Configuring Syslog** **2**
3.1 syslog facility . . . . . 3
3.2 syslog levels . . . . . 3
3.3 syslog actions . . . . . 4
3.4 `syslog.conf` example . . . . . 4

**4 Rotating Log Files with `logrotate`** **5**
4.1 Configuring `logrotate` . . . . . 5

**5 Examining Log Files** **6**
5.1 Log Messages . . . . . 6
5.2 How to search for particular events . . . . . 7

**6 License Of This Document** **7**



# 1 Context

**Topic 111 Administrative Tasks [21]**

**1.111.1** Manage users and group accounts and related system files [4]

**1.111.2** Tune the user environment and system environment variables [3]

**1.111.3 Configure and use system log files to meet administrative and security needs [3]**

**1.111.4** Automate system administration tasks by scheduling jobs to run in the future [4]

**1.111.5** Maintain an effective data backup strategy [3]

**1.111.6** Maintain system time [4]

# 2 Objectives

**Description of Objective**

Candidate should be able to configure system logs. This objective includes managing the type and level of information logged, manually scanning log files for notable activity, monitoring log files, arranging for automatic rotation and archiving of logs and tracking down problems noted in logs.

**Key files, terms, and utilities include:**

**`/etc/syslog.conf`** — configuration file for `syslogd`

**`/var/log/*`** — where the log files are found

**`logrotate`** — the program that "rotates" log files

**`tail -f`** — the best way to watch log files as things happen

# 3 Configuring Syslog

**`/etc/syslog.conf`**

- Each line in `/etc/syslog.conf` contains comments that start with a '`#`' or rules of the form: ⟨*facility*⟩.⟨*level*⟩⟨*action*⟩

## 3.1 syslog facility

**syslog facility**

**`authpriv`** — security/authorization messages (private)

**`cron`** — clock daemon (cron and at)

**`daemon`** — system daemons without separate facility value

**`ftp`** — ftp daemon

**`kern`** — kernel messages

**`local0...local7`** — reserved for local use

**`lpr`** — line printer subsystem

**`mail`** — mail subsystem

**`news`** — USENET news subsystem

**`syslog`** — messages generated internally by `syslogd`

**`user`** — generic user-level message

**`uucp`** — UUCP subsystem

See `$` **`man 3 syslog`** ↩

## 3.2 syslog levels

**syslog levels**

**security threshhold beyond which messages are logged**
in decreasing importance:

**`emerg`** — system is unusable

**`alert`** — action must be taken immediately

**`crit`** — critical conditions

**`err`** — error conditions

**`warning`** — warning conditions

**`notice`** — normal, but significant, condition

**`info`** — informational message

**`debug`** — debug-level message



## 3.3 syslog actions

**syslog actions**

Can be:

- filename (with full pathname), or
- a hostname preceded with '`@`', or
- a comma-separated list of users, or
- an asterisk '`*`' meaning all logged in users

## 3.4 `syslog.conf` example

**`syslog.conf` example**

```
# 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;cron.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

# Log cron stuff
cron.*                                                  /var/log/cron

# Everybody gets emergency messages
*.emerg                                                 *

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

**`syslog.conf` example — 2**

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

# Note: the rawhide openldap /etc/init.d/ldap script starts slapd with
# the -l daemon option, which was confusing.
# I added the option -l local5 to the (newly created)
# /etc/sysconfig/ldap
local5.*                                                -/var/log/slapd
```

```
# local4.*                                              /var/log/squid

# Now I've set log-facility local1; in dhcpd.conf
local1.*                                              /var/log/dhcp-log

#
# INN
#
news.=crit                                      /var/log/news/news.crit
news.=err                                       /var/log/news/news.err
news.notice                                     /var/log/news/news.notice

daemon,kern.*                                   /var/log/debug
```

# 4 Rotating Log Files with `logrotate`

**Rotating Log Files with `logrotate`**

- Log files grow rapidly
- Can grow to extreme sizes without rotation
- log rotation renames files and redirects logging to the new file: `messages` → `messages.1` → `messages.2` → `messages.3` → `messages.4` → delete
- Run `logrotate` from `cron`

## 4.1 Configuring `logrotate`

**`logrotate` configuration**

- Main configuration file is `/etc/logrotate.conf`
- . . . but most configuration belongs to the software packages, which put a file into directory `/etc/logrotate.d/`

```
$ cat /etc/logrotate.d/ldap ↩
# Nick 17 Aug 2003: copied from my /etc/logrotate.conf on ictl
/var/log/slapd
    weekly
    create 0664 ldap ldap
    rotate 20
    #postrotate
    #        /etc/rc.d/init.d/ldap condrestart
    #endscript
    notifempty
```



```
$ cat /etc/logrotate.d/syslog ↩
/var/log/messages /var/log/secure /var/log/maillog
 /var/log/spooler /var/log/boot.log /var/log/cron
 /var/log/debug
    sharedscripts
    weekly
    rotate 60
    postrotate
        /bin/kill -HUP `cat /var/run/syslogd.pid
          2> /dev/null` 2> /dev/null || true
    endscript
```

# 5 Examining Log Files

**Examining Log Files**

- Many log files are readable by none but root:
- Simplest: `$ sudo tail -f /var/log/messages ↩`
- `$ sudo less /var/log/messages ↩`
  - within `less`, press `F`
- Using either method, new additions to the log file are shown

## 5.1 Log Messages

**Log Messages**

**date and time** — in local time on my machine

**hostname** — of the machine that generated the message

**program or user** — that generated the message, e.g., `kernel`, `named`, `postfix`, `dhcpd`, . . .

**message text**

## 5.2 How to search for particular events

**Searching for particular events**

- Can `grep` for messages relating to a particular program:

```
$ sudo grep dhcpd /var/log/messages ↩
Nov 14 06:30:13 nicku dhcpd: DHCPDISCOVER from 00:04:e2:2e:c3:d6
 via eth0
Nov 14 06:30:13 nicku dhcpd: DHCPOFFER on 192.168.0.8
 to 00:04:e2:2e:c3:d6 via eth0
```

# 6 License Of This Document

**License Of This Document**