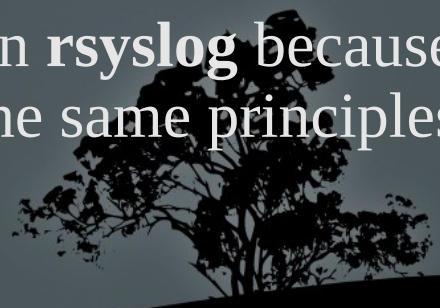
LPIC-1 102-500 – Lesson 10

108.2 System Logging

Logging

- **Logging** is one of the most important services provided by a Linux system because we look at the past and understand how the system behaves.
- The main standard for managing the log files is **Syslog** with **syslogd** as the reference implementation. It uses the Client – Server model where you have a central Syslog server and all other systems are sending their logs to that one.
- Some system services are using the main log files (/var/log/messages, /var/log/syslog) while others have their own log files (/var/log/apache/*)
- Modern systems use more advances Syslog implementations like **rsyslog** and **syslog-ng**. We will be focusing on **rsyslog** because that is the most popular today. They all work on the same principles though.



The */etc/rsyslog.conf* configuration file

- The **/etc/syslog.conf** file contains the configuration for the **rsyslogd**. The file format is as follows:
- **facility.priority action**
- **facility**: sets the message source which can be one of: **auth**, **authpriv**, **cron**, **daemon**, **kern**, **lpr**, **mail**, **ftp**, **mark**, **news**, **syslog**, **user**, **uucp**, and **local0** to **local7**
- **priority**: sets the message severity and can be one of: (sorted from the most severe to the mildest): **emerg**, **alert**, **crit**, **err**, **warning, notice**, **info, debug**.
- **action**: the message destination is defined here. Usually it points to some log file but it may as well be a Terminal, another syslog server or even a user account.

List of facilities

facility	Περιγραφή
authpriv (auth,security)	Authentication, Authorization and Security messages. The use of authpriv is preferred έναντι to auth and security
cron	Cron scheduler messages
daemon	Deamon messages
ftp	FTP messages
kern	Kernel messages
lpr	Printing messages
mail	Email related messages
mark	For syslog internal use
news	nntp (newsgroups) messages
syslog	Messages from syslog itself
user	User messages
uucp	UUCP (Unix-to-Unix Copy) messages
local0,local1,...,local7	These facilities are for custom/local use and they can be set from the admins

List of priorities

priority	Περιγραφή
emerg (panic)	Extremely urgent messages that affect system stability and have the higher priority. emerg is prefered to panic
alert	Messages that need immediate actions. Second top priority
crit	Critical conditions.
err (error)	System or service errord. err is preferred.
warning (warn)	Serious warnings. warning is preffered
notice	Important notices
info	Useful information
debug	Debug messages for troubleshooting. Lowset priority

Configuration examples in rsyslog.conf

- **mail.*** **/var/log/maillog** # send all messages (regardless the priority), comming from the mail system to the **/var/log/maillog** log file.
- ***.emerg** ***** # send all **emerg** messages (regardless of facility) to all user terminals
- ***.*** **@syslog.server.dom** # send (over network) all system messages to the **syslog.server.dom** server.
- **auth,authpriv.*** **/var/log/auth** # send all auth/security messages to the **/var/log/auth** log file.

Configuration examples in rsyslog.conf

- **kern.crit /dev/console** # send the critical and higher kernel messages (crit, alert, emerg) to the console (usually /dev/tty1).
- **kern.=info;kern.=notice /dev/tty8** # send kernel message of information and notice priority only to the **/dev/tty8** (**Ctrl-Alt-F8**).
- **kern.info;daemon.!debug @10.0.0.10** # send kernel messages of severity infoe and above and all daemon messages, excluding debug to the **10.0.0.10** server.
- ***.info;mail.none;cron.none;news.none;authpriv.none \ / var/log/messages** # all system messages of priority info and up, will end up in **/var/log/messages except for the mail**, **cron**, **news**, and **authpriv** facilities.

Creating log entries with `logger`

- The **logger** command can be used to create log entries by a user or a script.
- `$ logger -p user.info "Strange behavior on console"`
  `# send the quoted message to facility user with priority info.`
- `$ logger -t bug -p user.info "Strange behavior on \ console"` `# replace the username at the beginning of the message with ‘bug:’`
- Where the message will be recorded depends on the settings in **/etc/rsyslog.conf**

Archiving of ols logfiles with `logrotate`

- **logrotate** is a utility to prevent logfiles from growing uncontrollably and consuming the system resources.
- It can archive old logs, compress them and delete those that are past their lifecycle. The old logfiles are replaced by new ones with updated information.
- The behavior od lograte is controlled by **/etc/logrotate.conf** configuration file and the individual configuration files under **/etc/logrotate.d/**.
- Old log files are assigned numeric values and even older ones are compressed with gzip, e.g. **logfile**, **logfile.1**, **logfile.2.gz**, **logfile.3.gz**

An example */etc/logrotate.conf* file

```
# Default settings
weekly # archiving on a weekly basis
rotate 4 # Preserve archives for 4 weeks
create # create a new arhive
include /etc/logrotate.d # include configuration files under /etc/logrotate.d

/var/log/wtmp { # customised settings for wtmp
    missingok # no error is the log file is missing
    monthly # archiving on a monthly basis
    create 0664 root utmp # create a new file will 0664 permissions, root ownership
                                    and utmp group
    rotate 1 # preserve archives for a month
}
```

Tools for log-file viewing

- Any text viewer/editor can be used to show log files. For binary log files there are special tools depending on the case e.g. **last** for reading **wtmp**.
- **# less /var/log/messages** # the basic file viewer
- **# view /var/log/syslog** # read-only **vi** flavor
- **# zless /var/log/user.2.gz** # for compressed text files
- **# grep <string> -r /var/log** # recursively search all the logs
- **# zgrep <string> /var/log/auth.log.*.gz** # search in compressed log files
- **# tail -f -n30 /var/log/secure** # show the last lines of log and follow it for new entries.
- **# journalctl** # new tool on systemd systems

The `systemd-journald` daemon

- Syslog is unstructured and finding what you are looking for in massive text file can be a hard task
- The **journald** daemon aims to be a more efficient log facility on **systemd** systems
- It provides an efficient, structured binary file format
- It uses the **journalctl** command to query its database
- It can cooperate with existing syslog systems
- Unlike syslog, it does not work over the network
- Its configuration file is **/etc/systemd/journald.conf**
- There can be **/var/log/journal** log store of persistent storage. If it does not exist, **/run/log/journal** is used instead. If it gets too big you can clean it:

```
# journalctl –vacuum-size=200M # leave only the most recent 200M logs
```

The */etc/systemd/journald.conf* configuration file

- Configuration for **journald** can be set in the **/etc/systemd/journald.conf** file.
- **Storage=persistent** # for persistent storage
- **Compress=yes** # compress log files
- **ForwardToSyslog=yes** # forward logs to syslog
- **SystemMaxUse=10G** # do not let ***var*log/journal** grow more than 10G

Viewing logs with `journalctl`

- **# journalctl** # view all logs from the beginning
- **# journalctl -e** # view all logs from the end
- **# journalctl -ef** # follow logs
- **# journalctl -xe** # -x adds explanation text
- **# journalctl -e -u apache2.server** # show only logs from the apache2 service
- **# journalctl -ef -u apache2.server** # follow logs from the apache2 service
- **$ journalctl -p crit** # show logs with critical priority
- **$ journalctl _PID=7654** # query by process id
- **$ journalctl _UID=999** # query by user id
- **$ journalctl -n 30** # show the 30 most recent entries

The `systemd-cat` command

- The **systemd-cat** command is for **journald** what **logger** is for Syslog. You can use it to send your own entries in journald:

```
# echo ' The end is near! Repent!' | systemd-cat
```

License