|  |  |
| --- | --- |
|  | **[Jaar]** |
|  | The Company  Martien Huijsmans |

|  |
| --- |
| **[Logging on Linux]** |
| Logging On Linux |

Inhoud

[Linux logging services 1](#_Toc402044905)

[Posix Syslog 2](#_Toc402044906)

[Kernel logging 3](#_Toc402044907)

[RSyslog Architecture 3](#_Toc402044908)

[Rsyslog queueing 3](#_Toc402044909)

[RSyslog.config 4](#_Toc402044910)

[Logformat 4](#_Toc402044911)

[Template description: 4](#_Toc402044912)

[Log Levels 4](#_Toc402044913)

[Kernel Ring buffer 5](#_Toc402044914)

[Logg, issues to address 5](#_Toc402044915)

[Centos & rsyslog 6](#_Toc402044916)

[Managing logfiles on Linux: Logrotate 6](#_Toc402044917)

[Moving Log data 6](#_Toc402044918)

[Log analysis 6](#_Toc402044919)

[Security 6](#_Toc402044920)

[Reliability 7](#_Toc402044921)

[Logging Libraries 7](#_Toc402044922)

[C++ 7](#_Toc402044923)

[Java 7](#_Toc402044924)

[Misc 7](#_Toc402044925)

# Linux logging services

Syslog / Rsyslog / syslog-ng

Rsyslog consists of different modules. Through configuration, modules are selected  
<http://www.rsyslog.com/doc/rsyslog_conf_modules.html>

RF5424 describes the rsyslog message format. Other RFC’s describe tansport, such as UDP.  
<http://tools.ietf.org/html/rfc5424>

On Centos 6.4, rsyslogd is running, but the udp/tcp port are not open.  
ref: <http://techtots.blogspot.nl/2011/12/rsyslog-enabling-remote-logging-service.html>  
$ sudo vi /etc/rsyslogd.conf  
Remove comments for  
# provides UDP syslog reception  
$ModLoad imudp  
$UDPServerRun 514  
For TCP syslog service:  
# provides TCP syslog reception  
$ModLoad imtcp  
$InputTCPServerRun 514  
Next starts rsyslogd service  
$ sudo service rsyslogd restart  
Next check that ports are open:  
$ sudo netstat -tlnup | grep 514  
or  
$ sudo netstat -tlnup | grep rsyslogd

If you running a VM, check that the VM is reachable (e.g. network adapter = bridged adapter).

Next create  
/etc/rsyslog.d/prog1  
with context  
if $programname == 'prog1' then {  
 action(type="omfile" file="/var/log/prog1.log")  
}

If this do not work as expected, check  
/var/log/messages

Ref: <http://wiki.gentoo.org/wiki/Rsyslog>

## Posix Syslog

<http://www.unix.com/man-page/POSIX/3posix/syslog/>

Use case  
(ref: <http://unix.stackexchange.com/questions/64158/how-does-rsyslog-receive-log-messages-automatically>)

* the Application uses the Posix API to write log message.
* The kernel stores the log message in a log ring buffer
* Rsyslog read the buffer via /dev/log  
  This ref contains a imuxsock configuration for reading from /dev/log  
  <http://www.rsyslog.com/doc/master/configuration/modules/imuxsock.html>

Note: rsyslopg has at least the following input handlers: imfile, imklog (kernel log messages) , imuxsock (for reading unix domain sockets, such as /dev/log)

Rsyslog manpage (<http://man7.org/linux/man-pages/man8/rsyslogd.8.html>) mentions /dev/log as input. /dev/log is a unix domain socket from rsyslogd perspective.

All syslogd implementation read /dev/log

## Kernel logging

Description + architecture diagram.  
<http://www.ibm.com/developerworks/linux/library/l-kernel-logging-apis/index.html>

## RSyslog Architecture

### Rsyslog queueing

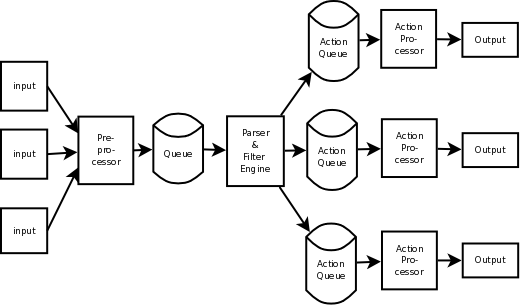


Figure 1 Turning Lanes and Rsyslog Queues

Ref: <http://www.rsyslog.com/doc/queues_analogy.html>

Starting with rsyslog 5.3.4, there can be multiple main-queue, providing a higher level of concurrency. See: <http://www.rsyslog.com/doc/master/concepts/multi_ruleset.html>   
For the to work the input module must support binding to non-standard rulesets. Next link suggest that for imuxsock, it is not possible: <http://unix.stackexchange.com/questions/45226/rsyslog-how-to-bind-rulesets-to-unix-sockets-imuxsock?rq=1>

## RSyslog.config

### Logformat

Default in rsyslog.conf: $ActionFileDefaultTemplate RSYSLOG\_TraditionalFileFormat  
ref: <http://www.rsyslog.com/using-a-different-log-format-for-all-files/>

Own log format can be specified e.g. identical to input  
<http://www.rsyslog.com/using-a-different-log-format-for-all-files/>

### Template description:

<http://www.rsyslog.com/doc/rsyslog_conf_templates.html>

At this site  
I found the following (for OLD syslog)  
<http://ftp.ics.uci.edu/pub/centos0/ics-custom-build/BUILD/rsyslog-3.19.7/doc/rsyslog-example.conf>   
following:   
$template TraditionalFormat,"%timegenerated% %HOSTNAME% %syslogtag%%msg:::drop-last-lf%\n"

## Log Levels

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Appl Log level (log4j) | Appl Log level (java util logging) | Boost::logging log level | rsyslog levels | kernel log levels (ref-1) |
|  |  |  | emergency | emergency |
| fatal |  | fatal | alert | alert |
|  | severe |  | critical | critical |
| error |  | error | error | error |
| warn | warning | warning | warning | warning |
|  |  |  | notice | notice |
| info | info | info | info | info |
|  | config |  |  |  |
| debug | fine | debug | debug | debug |
| trace | finer | trace |  |  |
|  | finest |  |  |  |
|  |  |  |  | default |
|  |  |  |  | continued |

Ref-1: <http://elinux.org/Debugging_by_printing#Log_Levels>

**RsysLog message info:**

* At open syslog
  + Identity
  + Log options
  + facility
* in log message
  + Level/ priority
  + Message

Rsyslog can filter on above information.

## Kernel Ring buffer

<http://elinux.org/Debugging_by_printing#Internals_.2F_Changing_the_size_of_the_printk_buffer>

# Logg, issues to address

Ref: <http://stackoverflow.com/questions/158457/logging-in-linux>

* API
  + Condense log code
  + Near-zero overhead if logging level is not active / a specific log statement is not active.
  + Ability to compile out for release built.
* Filtering
  + On level of log messages as close to source as possible
  + Selectively turn on/off logging, e.g. for a subsystem. File, etc.
  + Dynamic adjustment of log level
    - Configuration Files
    - Global
    - Module / class based loggers
* Formatting
  + Separating log message from formatting;  
    allows for XML formatting overall message
  + Formatting of messages
    - after filtering
    - async from logging
  + Support message items
    - Timestamp (ms)
    - Process
    - Thread
    - Function
    - LineNo
* Sinkes
  + Support for several sinks, with formatting per sink
    - Console
    - File
    - Syslog
* Performance
  + Throughput: Buffering & auto-flush-after-every-record, user flushing of log data
  + Delay: minimal impact on main application
* Threading
  + Single threaded (efficiency)
  + MT model
    - MT: logger / thread
    - MT: logger usage from multiple threads
  + Synchronization of log-records: in front-end or back-end.
* Syslog sink
  + Rsyslog posix
  + Udp
  + Tcp
    - Flow Control
* File Sink
  + Aging of files
  + Log rotation
  + Compression of files
  + Disk full prevention
* Send to remote system
* Robustness:
  + what if an application goes crazy  
    linux uses ringbuffer (source??) to ensure that it cannot eat up the disk space,
  + test: generates 1000 messages of 130 bytes in loop. All messages ended up in /var/log/messages. Total size = 130.000 = 127k  
    $ sudo cat /var/log/messages | grep "this is a very" | wc –l  
    printed 1000.
* Documentation
* library selection criteria
  + licensing model
  + community

# Centos & rsyslog

Centos 6.5 has rsyslog 5.8.1  
Latest version of rsyslog is : 8.2.1

# Managing logfiles on Linux: Logrotate

10 commands:  
<http://www.thegeekstuff.com/2010/07/logrotate-examples/>

# Moving Log data

<http://flume.apache.org/>

# Log analysis

Gist that discusses use of HIVE, HBase for log analysis  
<https://gist.github.com/emk/1556097#file-apache-logs-hive-sql>

Setting up Hadoop with RPM.  
<http://hortonworks.com/blog/set-up-apache-hadoop-in-minutes-with-rpms/>

Data management (for Hadoop, uses Ozie)  
<http://falcon.incubator.apache.org/>   
  
MongoDB

Logstash / Elasticsearch

# Security

See RFC5242

# Reliability

RFC5424 describes this.  
Both TCP and UDP are supported. But also with TCP, buffered data can be lost.  
RELP is an attempt to have reliable logging. But if that uses async

# Logging Libraries

## C++

2 articles that provide and describe an implementation. Very nice.  
<http://www.drdobbs.com/cpp/logging-in-c/201804215?pgno=1>   
<http://www.drdobbs.com/cpp/logging-in-c-part-2/221900468>   
Includes ref to atomic memory update operation / Boehm's atomic\_ops library : <http://www.hpl.hp.com/research/linux/atomic_ops/>

Wrapper over posix syslog.  
<http://www.liblogging.org/p/blog-page_24.html>

Boost Logging  
Supports syslog API and UDP-514 logging.  
Note that UDP is also robustness. I app goes wild, packets are dropped.

<http://logging.apache.org/log4cxx/>   
Supports Syslog UDP  
<https://android.googlesource.com/platform/external/apache-log4cxx/+/bf7b5ebcf7ce58764c86ab8c6f00ea1791d64026/src/main/cpp/syslogwriter.cpp>

<https://code.google.com/p/google-glog/>

## Java

Syslog Handler for Java util logging.  
<https://code.google.com/p/agafua-syslog/>

# Misc

Interesting example, unclear how to use it  
<http://stackoverflow.com/questions/17947611/log4j2-syslog-appender-and-patternlayout>

Example log message  
<135>1 2014-05-26T23:01:03.219+02:00 LPT008 MyApplLog4j2 – Audit – <here is the actual log message>

Mapping this to RFC5424 gives the following data

* Pri: <135>
* Version: 2014-05-26T23:01:03.219+02:00
* Hostname: LPT0088
* App-name: MyApplLog4j2
* ProcId: -
* MsgId: Audit
* Structured-data: -
* Msg = <here is …>

Interesting trouble shoot analysis:  
<http://serverfault.com/questions/518975/rsyslog-is-not-working-properly-it-does-not-log-anything>   
root cause: permissions