rsyslog

rsyslog (http://www.rsyslog.com/) is an alternate logger to **syslog-ng** and offers many benefits over **syslog-ng**.

Related articles

syslog-ng

Contents

Installation

Note: It is recommended to disable and uninstall the `syslog-ng (https://www.archlinux.org/packages/?name=syslog-ng)` package to prevent possible conflicts.

Install the `rsyslog (https://aur.archlinux.org/packages/rsyslog/)`[AUR] package.

Starting service

You can **start/enable** the `rsyslog (https://aur.archlinux.org/packages/rsyslog/)`[AUR] service after installation.

Configure Hostname

Rsyslog uses the `glibc (https://www.archlinux.org/packages/?name=glibc)` routine `gethostname()` or `gethostbyname()` to determine the hostname of the local machine. The `gethostname()` or `gethostbyname()` routine check the contents of `/etc/hosts` for the fully qualified domain name (FQDN) if you are not using **BIND** or **NIS**.

You can check what the local machine's currently configured FQDN is by running `hostname --fqdn`. The output of `hostname --short` will be used by rsyslog when writing log messages. If you want to have full hostnames in logs, you need to add

`$PreserveFQDN on` to the beginning of the file (before using any directive that write to files). This is because, rsyslog reads config file and applies it on-the-go and then reads the later lines.

The `/etc/hosts` file contains a number of lines that map FQDNs to IP addresses and that map aliases to FQDNs. See the example `/etc/hosts` file below:

```
/etc/hosts
```

```
#<ip-address>   <hostname.domain.org>   <hostname>
#<ip-address>      <actual FQDN>                      <aliases>
127.0.0.1       localhost.localdomain somehost.localdomain      localhost somehost
::1                     localhost.localdomain somehost.localdomain      localhost somehost
```

`localhost.localdomain` is the first item following the IP address, so `gethostbyname()` function will return **localhost.localdomain** as the local machine's FQDN. Then `/var/log/messages` file will use **localhost** as hostname.

To use **somehost** as the hostname. Move **somehost.localdomain** to the first item:

```
/etc/hosts
```

```
#<ip-address>   <hostname.domain.org>                           <hostname>
#<ip-address>      <actual FQDN>                                        <aliases>
127.0.0.1       somehost.localdomain localhost.localdomain      localhost somehost
::1                     somehost.localdomain localhost.localdomain      localhost somehost
```

Configuration

Since systemd 216 (August 2014) there is no longer a default forward from systemd-journal to a running syslog daemon - so in order to gather system logs you either have to **turn journald Forward Feature on** or **use the imjournal** module of rsyslog go gather the logs by importing it from the systemd journald.

imjournal

By default, all syslog messages are handled by **systemd**. If you want rsyslog to pull messages from systemd, load the *imjournal* module:

```
/etc/rsyslog.conf
$ModLoad imjournal
```

journald's syslog-forward feature

```
/etc/systemd/journald.conf
ForwardToSyslog=yes
```

The **rsyslog (https://aur.archlinux.org/packages/rsyslog/)**[AUR] doesn't create its working directory `/var/spool/rsyslog` defined by the `$WorkDirectory` variable in the configuration file. You might need to create it manually or change its destination.

Log output can be fine tuned in `/etc/rsyslog.conf`. The daemon uses Facility levels (see below) to determine what gets put where. For example:

```
/etc/rsyslog.conf

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

States that all messages falling under the **authpriv** facility are logged to `/var/log/secure`.

Another example, which would be similar to the behaviour of *syslog-ng* for the old `auth.log`:

```
/etc/rsyslog.conf

auth.*                                                  -/var/log/auth
```

References

- **Archwiki reference for systemd-syslog integration**
- **Structure of the rsyslog.conf file (http://www.rsyslog.com/doc/rsyslog_conf.html).**
- **Reference documentation on imjournal input module (http://www.rsyslog.com/doc/v8-stable/configuration/modules/imjournal.html?highlight=imjournal)**

Facility Levels

Note: The mapping between Facility Number and Keyword is not uniform over different operating systems and different syslog implementations. Use the keyword where possible, until it is determined which numbers are used by Arch.

Facility Number	Keyword	Facility Description
0	kern	kernel messages
1	user	user-level messages
2	mail	mail system
3	daemon	system daemons
4	auth	security/authorization messages
5	syslog	messages generated internally by syslogd
6	lpr	line printer subsystem
7	news	network news subsystem
8	uucp	UUCP subsystem
9		clock daemon
10	authpriv	security/authorization messages
11	ftp	FTP daemon
12	-	NTP subsystem
13	-	log audit
14	-	log alert
15	cron	clock daemon
16	local0	local use 0 (local0)
17	local1	local use 1 (local1)
18	local2	local use 2 (local2)
19	local3	local use 3 (local3)
20	local4	local use 4 (local4)
21	local5	local use 5 (local5)
22	local6	local use 6 (local6)
23	local7	local use 7 (local7)

Security Levels

As defined in **RFC 5424 (http://tools.ietf.org/html/rfc5424)**, there are eight security levels:

Code	Severity	Keyword	Description	General Description
0	Emergency	emerg (panic)	System is unusable.	A "panic" condition usually affecting multiple apps/servers/sites. At this level it would usually notify all tech staff on call.
1	Alert	alert	Action must be taken immediately.	Should be corrected immediately, therefore notify staff who can fix the problem. An example would be the loss of a primary ISP connection.
2	Critical	crit	Critical conditions.	Should be corrected immediately, but indicates failure in a primary system, an example is a loss of a backup ISP connection.
3	Error	err (error)	Error conditions.	Non-urgent failures, these should be relayed to developers or admins; each item must be resolved within a given time.
4	Warning	warning (warn)	Warning conditions.	Warning messages, not an error, but indication that an error will occur if action is not taken, e.g. file system 85% full - each item must be resolved within a given time.
5	Notice	notice	Normal but significant condition.	Events that are unusual but not error conditions - might be summarized in an email to developers or admins to spot potential problems - no immediate action required.
6	Informational	info	Informational messages.	Normal operational messages - may be harvested for reporting, measuring throughput, etc. - no action required.
7	Debug	debug	Debug-level messages.	Info useful to developers for debugging the application, not useful during operations.

Tip: A common mnemonic used to remember the syslog levels in reverse order: "Do I Notice When Evenings Come Around Early".

Examples

journald with rsyslog for kernel messages

Since the syslog component of systemd, journald, does not flush its logs to disk during normal operation, these logs will be gone when the machine is shut down abnormally (power loss, kernel lock-ups, ...). In the case of kernel lock-ups, it is pretty important to have some

kernel logs for debugging. Until journald gains a configuration option for flushing kernel logs, rsyslog can be used in conjunction with journald.

Summary of requirements:

- journald must still get all log messages.
- rsyslog must only log kernel messages, all other logs are handled by journald.
- Kernel logs must be logged separatedly to `/var/log/kernel.log`.
- Use systemd to start the service.

Installation and configuration steps:

1. Install **rsyslog (https://aur.archlinux.org/packages/rsyslog/)**[AUR].
2. Edit `/etc/logrotate.d/rsyslog` and add `/var/log/kernel.log` to the list of logs. Without this modification, the kernel log would grow indefinitely.
3. Edit `/etc/rsyslog.conf` and comment everything except for `$ModLoad imklog`. I also kept `$ModLoad immark` to have a heart-beat logged.
4. Add the next line to the same configuration file:

```
kern.*    /var/log/kernel.log;RSYSLOG_TraditionalFileFormat
```

The `kern.*` part catches all messages originating from the kernel. `;RSYSLOG_TraditionalFileFormat` is used here to use a less verbose date format. By default, a date format like `2013-03-09T19:29:33.103897+01:00` is used. Since

the kernel log contains a precision already (printk time) and the actual log time is irrelevant, I prefer something like `Mar 9 19:29:13`.

5. Since rsyslog should operate completely separated from systemd, remove the option that shares a socket with systemd:

```
sed 's/^Sockets=/#&/' /usr/lib/systemd/system/rsyslog.service | sudo tee /etc/systemd/system/rsyslog.service
```

6. Next, make rsyslog start on boot and start it for this session by **starting** and enabling `rsyslog.service`.

Note: rsyslog reads from `/proc/kmsg`. This means that subsequent reads from that file (either the user or a syslog daemon) will not read "old" logs from that file anymore. journald is not affected as it reads from `/dev/kmsg` which allows multiple readers.

See also

- **Rsyslog manual (http://www.rsyslog.com/doc/manual.html)**
- **rsyslog's versus syslog-ng (http://www.rsyslog.com/doc-rsyslog_ng_comparison.html)**.

Retrieved from "https://wiki.archlinux.org/index.php?title=Rsyslog&oldid=508677"