НАСТРОЙКА RSYSLOG В LINUX

Сервер Декабрь 15, 2016 6 admin

Когда в системе происходит та или иная ошибка, нужно выяснить почему она произошла, исправить ее и попытаться сделать так, чтобы такой ошибки больше не было. В этом системным администраторам очень сильно помогает логирование всех ошибок. Например, общие сообщения ядра и программ сохраняются в /var/log/messages.

Но рано или поздно файлы логов становятся слишком большими, они занимают все место на диске и это приводит к новым ошибкам. Поэтому важно контролировать, как и куда сохраняются файлы журналов. На протяжении многих лет в операционной системе Linux используется сервис Syslog для управления логами. В современных версиях применяется его модификация — rsyslog.

В этой статье мы рассмотрим как выполняется установка и настройка rsyslog, рассмотрим основы настройки локального логирования в Linux, а также пойдем дальше и настроем удаленный сбор логов. Эта информация также поможет вам улучшить свои навыки поиска ошибок и неисправностей.

Содержание статьи:

ЧТО ТАКОЕ RSYSLOG?

Развитие rsyslog началось в 2004 году, в качестве форка используемого тогда сервиса Syslog. Программа очень быстро набрала популярность среди пользователей и сейчас она поставляется по умолчанию во многих дистрибутивах Linux.

Rsyslog — это очень быстрый, расширяемый сервис для управления логами с огромным количеством возможностей. Среди его возможностей можно отметить поддержку фильтрации контента, а также передачу логов по сетям. Разработчики утверждают, что система очень быстрая, программа может обрабатывать до миллиона сообщений в секунду.

Вот основные возможности:

- Многопоточность;
- TCP, SSL, TLS, RELP;
- Поддержка MySQL, PostgreSQL, Oracle;
- Фильтрация журналов;
- Полностью настраиваемый формат вывода.

КАК ПРОИСХОДИТ ЛОГИРОВАНИЕ?

Все программы Linux ведут лог путем отправки сообщений об ошибках или своем состоянии с помощью сокета syslog или просто записывая все сообщения в файл, который будет находиться в каталоге /var/log/.

Но важное значение имеет уровень подробности логирования. Вы можете настраивать подробность в каждой отдельной программе, или же с помощью syslog. Это поможет уменьшить использование дискового пространства, на хранение логов. Но тут нужно найти компромисс между количеством информации и использованием диска.

Например, ядро Linux определяет такие уровни логов, или как мы будем называть их ниже — приоритеты:

- **KERN_EMERG** — система неработоспособна;
- **KERN_ALERT** — нужно немедленно принять меры;
- **KERN_CRIT** — критическая ошибка;
- **KERN_ERR** — обычная ошибка;
- **KERN_WARNING** — предупреждение;
- **KERN_NOTICE** — замечание;
- **KERN_INFO** — информационное сообщение;
- **KERN_DEBUG** — сообщения отладки.

Подобные уровни лога поддерживаются в большинстве программ, которые ведут логи.

НАСТРОЙКА RSYSLOG В LINUX

Все настройки Rsyslog находятся в файле /etc/rsyslog.conf и других конфигурационных файлах из /etc/rsyslog.d. Вы можете посмотреть существуют ли у вас эти файлы выполнив:

```
ls /etc/rsys*
```

```
rsyslog.conf rsyslog.d/
```

Основной конфигурационный файл — /etc/rsyslog.conf, в нем подключены все файлы из папки rsyslog.d с помощью директивы IncludeConfig в самом начале файла:

```
IncludeConfig /etc/rsyslog.d/*.conf
```

В этих файлах могут содержаться дополнительные настройки, например, аутентификация на Rsyslog сервере. В главном конфигурационном файле содержится очень много полезных настроек.

Обычно он обеспечивает управление локальными логами по умолчанию, но для работы через сеть нужно добавить настройки. Сначала давайте рассмотрим что представляет из себя этот файл.

Синтаксис конфигурационного файла очень прост:

$переменная значение

Все директивы начинаются со знака доллара, содержат имя переменной, а дальше связанное с ней значение. Так выглядит каждая строка конфигурационного файла. В его первой части размещены общие настройки программы и загрузка модулей. Во второй — ваши правила сортировки и фильтрации лог файлов.

```
ModLoad imuxsock # provides support for local system logging
$ModLoad imklog # provides kernel logging support
#$ModLoad immark # provides --MARK-- message capability



# provides UDP syslog reception

#$ModLoad imudp
#$UDPServerRun 514

# provides TCP syslog reception



#$ModLoad imtcp
#$InputTCPServerRun 514
```

В этом участке загружаются все необходимые модули программы. Существуют четыре типа модулей

- **Модули ввода** — можно рассматривать, как способ сбора информации из различных источников, начинаются с im.
- **Модули вывода** — позволяют отправлять сообщения в файлы или по сети, или в базу данных, имя начинается на om;
- **Модули фильтрации** — позволяют фильтровать сообщения по разным параметрам, начинаются с fm;
- **Модули парсинга** — предоставляют расширенные возможности для синтаксического анализа сообщения, начинаются с pm.

Сначала загружается модуль imuxsock, который позволяет сервису получать сообщения из сокета, а второй imklog получает сообщения ядра. Следующим загружается модуль mark, который позволяет маркировать соединения, или же выводить сообщения о том, что syslog

все еще работает. Например, вы можете попросить rsyslog выводить сообщения каждые 20 минут:

```
MarkMessagePeriod 1200
```

Дальше идут глобальные директивы:

```
ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat
```

В этой строке мы указываем, что нужно использовать стандартный формат хранения времени, в секундах с 1970 года.

Дальше следует набор прав разрешений для файлов журналов, которые будут созданы в вашей системе:

```
FileOwner root
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022
```

После создания этих файлов их права можно менять, на те, которые вам нужны.

Выше была более общая настройка syslog, ну а теперь самое интересное — правила сортировки логов. Вот набор правил по умолчанию:

```
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
```

Каждое правило имеет свой синтаксис, сначала идет источник и приоритет, затем действие. Если источник и приоритет совпадают, сообщение отправляется в указанный файл. Например, мы можем отправить больше сообщений в лог системы linux /var/messages:

*.info;mail.none;authpriv.none;cron.none /var/log/messages

В этой строке мы выбираем все сообщения уровня info, кроме mail,authpriv и cron. Шаблон mail.none будет означать, что не нужно логировать ни один уровень сообщений. Соответственно конструкция *.info — значит логировать сообщения от всех источников, но только с уровнем info, *.* — это вообще все сообщения, со всеми уровнями.

Источник и приоритет нечувствительны к регистру. Также заметьте, что приоритеты уровней error, warn и panic больше не используются, так как считаются устаревшими. В целом, в качестве источников вы можете использовать:

- auth;
- authpriv;
- cron;
- daemon;
- kern;
- lpr;
- mail;
- mark;
- news;
- security (эквивалентно auth);
- syslog;
- user;
- uucp;
- local0 … local7;

А в качестве приоритетов вы можете применить:

- emerg (раньше panic);
- alert;
- crit;
- error (раньше err);
- warn (раньше warning);
- notice;
- info;
- debug;

Как я уже говорил, звездочки на любом месте означают все варианты. Для фильтрации логов могут использоваться не только источник и приоритет, но и более сложные выражения на основе условий и сравнений.

Вы можете выполнять фильтрацию сообщений с помощью такого синтаксиса:

:поле, сравнение, «значение» путь_к_файлу

В качестве операции сравнения вы можете использовать такие варианты:

- **contains** — поле содержит указанное значение;
- **isequal** — поле должно быть идентичным значению;
- **startswith** — поле должно начинаться со значения;
- **regex** — сравнивает поле с регулярным выражением.

Например, отфильтруем сообщения только от определенной программы:

```
:syslogtag, isequal, "giomanager:" /var/log/giomanager.log
& stop
```

Кроме того, можно использовать более простой синтаксис, в виде выражения if. Вот основной синтаксис:

if $поле сравнение ‘значение’ then файл

Здесь используются те же самые компоненты, только выглядит немного проще. Например:

```
if $syslogtag == 'giomanager' then /var/log/giomanager.log
```

НАСТРОЙКА SYSLOG ДЛЯ УДАЛЕННОГО ЛОГИРОВАНИЯ

Было бы очень удобно, если бы логи со всех серверов сети собирались на одной машине. Здесь бы были все важные сообщения об ошибках и неполадках. Вы могли бы все это очень быстро проанализировать. Отправить лог на удаленный сервер достаточно просто, для этого достаточно указать @ и ip адрес удаленной машины, на которой запущен rsyslog:

```
*.info;mail.none;authpriv.none;cron.none @xx.xx.xx.xx:514
```

Здесь 514 — это порт, на котором слушает rsyslog. Настройка rsyslog на прием логов заключается в запуске сервиса с модулями imtcp и imudp. Далее, все что нужно для того чтобы получить логи с определенной машины, отфильтровать их из общего потока с помощью фильтров:

```
if $fromhost-ip contains '192.168.1.10' then /var/log/proxyserver.log
```

Без фильтров, сообщения с разных машин будут писаться в один общий лог системы linux в зависимости от того как вы их распределите.

ВЫВОДЫ

В этой статье мы рассмотрели как выполняется настройка Rsyslog в Linux для сбора логов на локальном компьютере, а также передачи их на удаленный сервер. Изначально все кажется очень сложным, но если разобраться, то и это можно настроить.

Обратите внимание, что при использовании rsyslog в качестве сетевого сервера для хранения логов место на диске будет очень быстро уменьшаться. Поэтому лучше применять в дополнение к программе такую утилиту, как Logrotate.