The sys admin's daily grind: RSyslog

WHERE TO NEXT?

Well-used services write reams of log information to disk, which is not only bothersome from a storage perspective but also pushes grep and the usual group of statistics tools to their limits. Will hitching the syslog daemon up to a database help? **BY CHARLY KÜHNAST**

Who said length doesn't matter? My spam filters alone give me a 3GB logfile daily, which would be fine if I just needed the beast to check up on the occasional error. Because I need to extract a whole bunch of statistics about spam and virus threats from the file, grepping such enormous files takes ages, besides creating far too much I/O overhead.

RSyslog [1] took me a giant leap closer to finding a solution – it logs directly to a MySQL or PostgreSQL database, meaning that I can replace my grep commands with fast SQL statements. RSyslog is included with many distributions and is the default application in Fedora 8, for example. By default, my Ubuntu lab environment runs sysklogd, making it easy for me to switch to RSyslog. For the time being, I can even keep my old *syslog.conf*. For the most part, RSyslog's configuration file is the same as the legacy format, although it does support a couple of additional options for linking up with the database.

First I need to set up a database. To do so, I run the *createDB.sql* script, which is provided with the RSyslog package:

```
mysql -u root ➥
-pPassword ➥
< ./createDB.sql
```

This command line automatically creates a database called *Syslog* along with the required tables.

At the MySQL prompt, I then create a user and assign privileges:

```
> grant ALL ON ➥
Syslog.* to ➥
rsyslog@localhost
 identified by 'secret';
> flush privileges;
```

Next, telling RSyslog to use the database is refreshingly simple – just two lines in */etc/rsyslog.d/mysql.conf*:

```
$ModLoad MySQL
mail.* >localhost,Syslog,➥
rsyslog,secret
```

The first line loads the module that RSyslog needs to access the database. The second line defines the log facility containing the entries RSyslog will push to the database. The mail facility data are all I need to create spam filter statistics, followed by the access parameters for the MySQL database: hostname, database name, MySQL user name, and password.

After rebooting, I was pleased to see the RSyslog daemons filling the database (Figure 1). All done! This won't magically improve your spam statistics, but at least it will get the job done faster. Incidentally, the example here works with the current 2.0.2. stable release. RSyslog's author, Rainer Gerhards, is working hard on a 3x version, which he promises will add even more neat features. ■

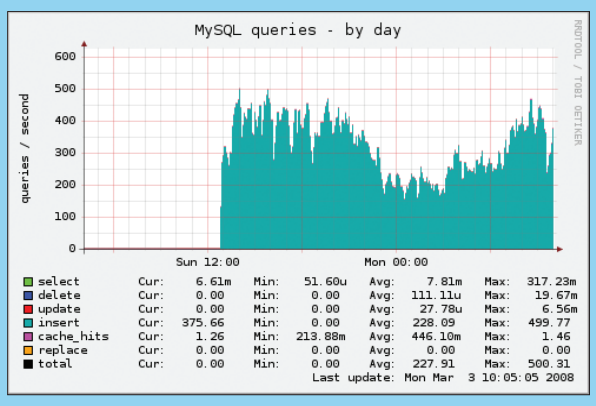

Figure 1: RSyslog feeds log information to a database instead of a file. The database provides the back end for statistics, such as the spam information shown here.

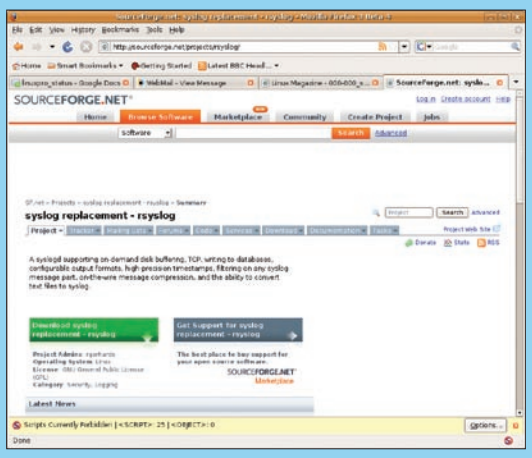

Figure 2: If your system doesn't come with rsyslog, you can download the latest version [1].

THE AUTHOR

Charly Kühnast is a Unix System Manager at the data center in Moers, near Germany's famous River Rhine. His tasks include ensuring firewall security and availability and taking care of the DMZ (demilitarized zone).

INFO

[1] RSyslog: *http://www.rsyslog.com*

SYSADMIN