Log management – exercises – AfNOGChix 2011, Blantyre, Malawi

Part I

1. Stop the existing syslog daemon:

```
# /etc/rc.d/syslogd stop
```

2. Install syslog-ng:

As root do the following: - This will allow FreeBSD to download a pre-downloaded copy from our FTP local Server:

NB:pay attention to the case.

```
# echo 'MASTER_SITE_OVERIDE=ftp://10.10.10.200/pub/FreeBSD/releases/i386/8.2-RELEASE/distfiles/' >> /etc/make.conf
```

```
then :
      # cd /usr/ports/sysutils/syslog-ng
      # make install clean
```

You should see this when the package finished installing – read the instructions and perform steps 2 & 3 from the list below:

```
syslog-ng is now installed!  To replace FreeBSD's standard syslogd
(/usr/sbin/syslogd), complete these steps:

1. Create a configuration file named /usr/local/etc/syslog-ng/syslog-ng.conf
   (a sample named syslog-ng.conf.sample has been included) :

      # cd /usr/local/etc/syslog-ng
      # cp syslog-ng.conf.sample syslog-ng.conf
      # chmod 644 syslog-ng.conf

2. Configure syslog-ng to start automatically by adding the following
   to /etc/rc.conf:

      syslog_ng_enable="YES"

3. Prevent the standard FreeBSD syslogd from starting automatically by
   adding a line to the end of your /etc/rc.conf file that reads:

      syslogd_enable="NO"

4. Shut down the standard FreeBSD syslogd:

      # kill `cat /var/run/syslog.pid`

5. Start syslog-ng:

    /usr/local/etc/rc.d/syslog-ng start
```

3. Edit the newly created file (syslog-ng.conf) and add this at the **end** of the file:

```
log { source(src); filter(f_local7); destination(local7); };
destination local7 { file("/var/log/local7.log"); };
```

4. Restart syslog-ng:

```
# /usr/local/etc/rc.d/syslog-ng restart
```

... check that the daemon has started:

```
# ps ax | grep syslog-ng
```

... you should see something like:

```
 6054  ??  Is     0:00.00 /usr/local/sbin/syslog-ng -p /var/run/syslog.pid
```

5. Test the syslog service

```
# logger -p local7.info 'this is a test'
```

... confirm that a file /var/log/local7.log now exists:

```
# ls -l  /var/log/
```

... confirm that you see the test message.

```
# tail /var/log/local7.log
```

6. Ask someone else in the room to send a syslog message to your host, using the '-h' option of the logger command. The **other** person should type this on **their** PC – so for example if you are PC1, and you ask PC2 to send you a message, they will type this:

```
# logger -h pc1 -p local7.info 'message from pc2'
```

... check that the message appears in **your** /var/log/local7.log

Part II

1. Edit /usr/local/etc/syslog-ng/syslog-ng.conf, and **change** the line at the bottom:

```
destination local7 { file("/var/log/local7.log"); };
```

to

```
destination local7 {
 file("/var/log/local7/$YEAR/$MONTH/$DAY/$HOST-$YEAR-$MONTH-$DAY-$HOUR.log"
   owner(root) group(root) perm(0644) dir_perm(0755) create_dirs(yes)
   template("$YEAR $DATE $HOST $MSG\n"));
};
```

2. Create the directory /var/log/local7/

```
# mkdir /var/log/local7
```

3. Restart syslog-ng

```
# /usr/local/etc/rc.d/syslog-ng restart
```

4. Repeat steps 5 & 6 from Part I (send a message using logger and ask someone else in the room to send a syslog message to your host, using the '-h' option of the logger command.)

```
# logger -h pc1 -p local7.info 'message from pc2'
```

5. See if messages are starting to appear under

```
/var/log/2011/11/XX/...
```

... what is the advantage of this ?

Part III - Syslog from one host to another

Let's send ALL our messages from our machine over to another machine. Why ? In a complex environment, you might have many servers, and you want to receive syslog messages from all those machines in one place.

1. We split into groups:

The **first PC** on every group will be the server and the rest PCs will use it as log server.

These machines are the "syslog server" for the Group. They will make the following changes to their syslog-ng configuration:

NB: Changes strictly for the group "syslog server"

1. Edit /usr/local/etc/syslog-ng/syslog-ng.conf, and find the line:

```
destination all { file("/var/log/all.log"); };
```

and change it to:

```
destination all {
   file("/var/log/all/$YEAR/$MONTH/$DAY/$HOST-$YEAR-$MONTH-$DAY-$HOUR.log"
   owner(root) group(root) perm(0644) dir_perm(0755) create_dirs(yes)
   template("$YEAR $DATE $HOST $MSG\n"));
};
```

Then find the line

```
#log { source(src); destination(all); };
^
```

... and remove the '#' in the beginning, so that it becomes:

```
log { source(src); destination(all); };
```

To avoid conflicts, we need to comment the previous log source we had added in Part 1:

to do so, find the line:

log { source(src); filter(f_local7); destination(local7); };

and add # in the beginning of the line

Save the file and exit, then create the directory:

```
# mkdir /var/log/all
```

... now restart syslog-ng:

```
# /usr/local/etc/rc.d/syslog-ng restart
```

The instructions for the other machines in the same group are on the next page.

The OTHER machines in the same group will change their syslog configuration to point to the **Syslog server** (1 per group). **DO NOT MAKE THE CHANGES BELOW TO MACHINES WHICH ARE "SERVERS" FOR THE GROUPS – THIS WILL MAKE SYSLOG LOOP MESSAGES!!!**

1. Edit /usr/local/etc/syslog-ng/syslog-ng.conf, and find the line:

```
# destination loghost { udp("loghost" port(514)); };
^
```

... and remove the '#' in the beginning, and replace "loghost"...:

```
destination loghost { udp("10.10.10.X " port(514)); };
```

... where **X** is the IP of the **SYSLOG SERVER** for the group (designated "syslog server" **PC for your group**)

Then, find the line:

```
# log { source(src); destination(loghost); };
^
```

... and remove the '#' in the beginning, changing it to:

```
log { source(src); destination(loghost); };
```

Save the file and exit, then restart syslog-ng:

```
# /usr/local/etc/rc.d/syslog-ng restart
```

Run tcpdump in another window/tab on port 514 (the syslog port):

```
# tcpdump -ni re0 port 514
```

Now, generate some syslog events, for example:

- use "logger" to send messages to syslog manually, any facility:

```
$ logger -p local7.info 'another test message'
$ logger -p mail.warning 'mail warning'
$ logger -p cron.info 'cron info'
```

- log in and out of your server as root (on the console)

... events like these will generate syslog messages.

You should be able to see:

- traffic on the network from your machine to the "syslog server" in your group.

If everything goes well, the syslog **on the syslog server** in your group (see /var/log/all/... files) should receive these messages and store them in the right files.

For fun:

- use "logger" to other machines in the room (**ANY** machine), any facility:

```
$ logger -h pcX -p local7.info 'test message'
$ logger -h pcY -p mail.warning 'mail warning'
$ logger -h pcZ -p cron.info 'cron info'
```

... to generate messages on everyone else!