Tuesday, March 2. 2010

Less known Solaris features - logadm

The mighty root couldn't sleep at night, so root walked around the castle. Deep down in the foundation of super users castle there was a strange room. It was filled with scrolls and some of the serving daemons of root filled this room with even more scrolls while root was standing there. So root looked in some of them: There were new ones, interesting ones. Then root took some old ones, blew away the dust and after a short look root thought "Damned ... those scrolls are so old, there aren't true anymore. Someone has to clean up this place".

So root spoke a mighty spell of infinite power and another daemon spawned from the ether: "You are the keeper of the scrolls. But don't keep everything. Just the last ones." And so it was done since the day of this sleepless night.

Housekeeping in your logfile directoriesOne of the regular task of any decent admin should be the housekeeping in the logfile directory. Logfiles are really useful, when something went wrong, but often they just filling the directories with data. Albeit sometimes it's useful to have even very old logfiles, most of the times you just need the recent ones in direct access.

logadmOne of the big conundrums with solaris is the point that few people know the logadm tool. It's available with Solaris since the release of Solaris 9. However it's still one of the well-kept secrets of Solaris despite the fact, that the tool is well documented and already in use Solaris. I'm often wondering what users of Solaris thing, where this .0-files were created

CapabilitiesFor all the usual task surrounding logfile handling you could use the command logadm . It's a really capable tool:
rotating logs (by copying/truncating or moving)
configuring rules, when a log rotation should take place. This rules can be based on ...

the size of the log file
the time since last log rotation

executing command before and after a logfile rotation
compressing rotated log files based on rules
specifying your own commands to copy/move the files
specifying commands that should be used instead of a simple deletion for expiration of files

How it workslogadm is the tool to configure it as well it's the tool that do the log rotation. To automatically rotate the logs, logadm is executed by cron once a day. Let's look into the crontab of the root
user.jmoekamp@hivemind:/var/squid/logs# crontab -l
[... CDDL Header omitted ...]
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 [-x /usr/lib/fs/nfs/nfsfind] && /usr/lib/fs/nfs/nfsfind
30 3 * * * [-x /usr/lib/gss/gsscred_clean] && /usr/lib/gss/gsscred_clean
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.shSo as you may have already recognized, the logadm script is executed daily at 3:10 am in a Solaris default configuration.

But how is the rotation itself done? Well, exactly as you would do it by typing in commands on the shell. logadm does the same job - just automatically. It generates a sequence of commands to rotate the logs and send them to a c shell for execution.

Practical side of logadmConfiguring itAlbeit it's possible to invoke the rotation totally from command line each time, this isn't a comfortable method to rotate logfiles. In almost every case you would configure the logadm facility in order to do the rotation task automatically. So how is this done?

logadm.confThe daily run of logadm depends on a config file. When you don't specify a config file, it's /etc/logadm.conf per default. The configuration is rather short and just rotates the usual suspects. The following file is a little bit extended, as the default configuration file doesn't use all important options:

jmoekamp@hivemind:~$ cat /etc/logadm.conf
[... CDDL Header omitted ...]
/var/log/syslog -C 8 -P 'Sat Feb 20 02:10:00 2010' -a 'kill -HUP `cat /var/run/syslog.pid`'
/var/adm/messages -C 4 -P 'Sat Feb 20 02:10:00 2010' -a 'kill -HUP `cat /var/run/syslog.pid`'
/var/cron/log -P 'Thu Dec 17 02:10:00 2009' -c -s 512k -t /var/cron/olog
/var/lp/logs/lpsched -C 2 -N -t '$file.$N'
/var/fm/fmd/errlog -M '/usr/sbin/fmadm -q rotate errlog && mv /var/fm/fmd/errlog.0- $nfile' -N -s 2m
/var/fm/fmd/fltlog -A 6m -M '/usr/sbin/fmadm -q rotate fltlog && mv /var/fm/fmd/fltlog.0- $nfile' -N -s 10m
smf_logs -C 8 -s 1m /var/svc/log/*.log
/var/adm/pacct -C 0 -N -a '/usr/lib/acct/accton pacct' -g adm -m 664 -o adm -p never
/var/log/pool/poold -N -a 'pkill -HUP poold; true' -s 512k
/var/squid/logs/access.log -P 'Tue Feb 23 06:26:23 2010' -C 8 -c -p 1d -t '/var/squid/logs/access.log.$n' -z 1 You can edit this file directly, but it's preferred to change it with the logadm command itself. Let's dissect some lines of this configuration.

Standard log rotation - introducing -C,-P and -a [...]
/var/log/syslog -C 8 -P 'Sat Feb 20 02:10:00 2010' -a 'kill -HUP `cat /var/run/syslog.pid`'
[...]This line is responsible for rotating /var/log/syslog. The -C 8 specifies, that the logadm should hold 8 old versions before it expires (read: deletes) old logfiles. With the -a 'kill -HUP `cat /var/run/syslog.pid` option the syslog gets an HUP signal after the log rotation. The syslogd needs this to recreate the logfile and to restart logging.

The -P isn't there in the pristine version of the file. Those options will are inserted when you run the logadm command. With this option, you tell logadm when when the last rotation was done. logadm inserts it each time it rotates a logfile. It's important to know, that this time is GMT time, so don't wonder about the offset to the local time shown with your logfiles.

In this statement you don't find a configuration of the time or size that leads to a log rotation. In this case the default values "1 byte and 1 week" are kicking in. So /var/log/syslog is rotated when the last rotation was at least one week in the past {\emph and} the file is at least one Byte in size.

Explicit definition of a rotation time span - introducing -pWith -p you can control the period between log rotations. For example 1d specifies that you rotate the log files on a daily schedule. [...]
/var/squid/logs/access.log -C 8 -c -p 1d -t '/var/squid/logs/access.log.$n'
[...]So this logfiles is rotated every day by the logadm execution initiated by the entryin the crontab.

A template for the name of the rotated log - introducing -t-t specifies the way, how the names for logfiles are created by logadm. The template for the new name of the logfile isn't just a fixed string, you can use several variables to control the name of the file.[...]
/var/squid/logs/access.log -C 8 -c -p 1d -t '/var/squid/logs/access.log.$n'
[...]This is a relatively simple example. $n is just the version number of the file, starting with 0. Thus a configuration would lead to a filename like this one:
-rw-r----- 1 webservd webservd 652486 2010-02-22 16:36 /var/squid/logs/access.log.0
$n is just one possible variable available for use in the templates. The logadm man page specifies further possible variables.

Explicit definition of maximum logfile size - introducing -sSometimes you want to set your current logfile a limit based on file size and not based on a time interval. The -s option allows you to do so:[...]
/var/cron/log -P 'Thu Dec 17 02:10:00 2009' -c -s 512k -t /var/cron/olog
[...]With this option logadm will rotate the logfile as soon it's 512k or larger at the time logadm runs.

Specifying both: Space and timeWhen you specify a maximum logfile size (-s) as well as a maximum logfile period (-p), both conditions are connected with an AND. So the default "1 byte and 1 week" can be translated: Rotate when the logfile has a size of at least 1 byte AND it's one week old, thus a week old zero-length logfile is not rotated by the default configuration.

Copy and truncate instead of moving - introducing -cRotating logfiles isn't unproblematic. Some application don't like it if you simply move the file away. They may use the rotated log instead of a new one, or they simply don't create a new logfile. Thus you have to restart the service or send a signal. There is an alternative. It's called truncating./var/cron/log -P 'Thu Dec 17 02:10:00 2009' -c -s 512k -t /var/cron/olog
[...]The -c option forces logadm to use cp instead of mv to rotate the log. To get a fresh start in the new log /dev/null is copied to the current logfile effectively make a 0 byte file out out it. This circumvents the need for a restart of the

application to restart logging.

Compress after rotation - introducing -zDue to their structure, logfiles can yield an excellent compression ratio. So it's sensible to compress them after rotation. You can configure this with the -z session. However often it's somewhat unpractical to work with compressed files, thus the parameter after -z forces logadm not to compress the stated number of the most recent log files. By doing so you have the recent logfiles available for processing without decompressing, without sacrifying space by leaving all logfiles uncompressed.[...]
/var/squid/logs/cache.log -C 8 -c -p 1d -t '/var/squid/logs/access.log.$n' -z 1
[...]-z 1 configures logadm to leave the most recent rotated logfile untouched and compresses the next one. This results to the following files:jmoekamp@hivemind:/var/squid/logs$ ls -l /var/squid/logs/access*
-rw-r----- 1 webservd webservd   0 2010-02-23 07:26 /var/squid/logs/access.log
-rw-r----- 1 webservd webservd 652486 2010-02-22 16:36 /var/squid/logs/access.log.0
-rw-r----- 1 webservd webservd 39411 2010-02-22 09:22 /var/squid/logs/access.log.1.gz

```
 Posted by Joerg Moellenkamp in English, Oracle, Solaris at 21:19
```

Cool! very useful! Didn't know about existence of this feature!
Anonymous on Mar 3 2010, 00:32

Hm... isn't there a race between copying the logfile and truncating it? I mean that in the window between copy and truncate my daemon could have actually logged something which would then disappear, right?
Anonymous on Mar 3 2010, 11:29

There is such a window, but that's the reason for this change in Solaris:
http://bugs.opensolaris.org/bugdatabase/view_bug.do?bug_id=6913894
Anonymous on Mar 3 2010, 11:54

Hi,

Is there a way where the log rotation happens on 1st of every month only and the log file should be moved to abc.log.%Y-%m, where m is the last month and not the current month.

Regards,
Gaurav
Anonymous on Apr 12 2011, 07:54