CST8177 – Linux II

logging, accounting

Todd Kelley
kelleyt@algonquincollege.com

Today's Topics

- syslog
- logger command line utility
- logwatch
- logrotate
- psacct
- lastcomm
- ac, last, lastlog
- xargs by special request

Syslog Facilities

- auth : authorization
- authpriv : private authorization
- cron : crond
- daemon : other misc daemons
- kern : the kernel
- lpr : printer
- mail : email
- news : usenet news
- syslog : syslog itself
- user : user messages
- uucp : unix to unix copy
- local0 through local7 : local use
- * : all of the above

Syslog Priorities

- In order of increasing priority:
  - debug : debug level messages
  - info : normal information
  - notice : normal but significant, unusual
  - warning : not an error, but action should be taken
  - err : error condition, non-urgent failure
  - crit : critical condition, failure in primary system
  - alert : action needed immediately
  - emerg : panic, system unusable, notify all
  - * : all of the above
  - none: none of the above for the given facility

Syslog actions

- Absolute pathname of a file
  - put the message (log entry) in that file
  - dash in front means omit syncing on every entry
- Terminal or Console : ex /dev/console
  - write the message on that screen
- @hostname : remote machine
  - send the message to syslog on the remote machine
- username
  - write the message on that user's terminal
- * : everyone logged in
  - write the message on everyone's screen
- named pipe (fifo) : useful for debugging

logger command

- logger: the command line utility for putting an entry in the logs

logger [-is] [-f file] [-p pri] [-t tag] [message ...]

Examples:

logger -p user.info -t logger "this is a test"

logger -p authpriv.info -t kelleyt "another test"

logwatch

- With all this logging information being recorded, a sysadmin should be monitoring it
- You would think someone would have written a script to "grep" through logs, or summarize them
- Yes, they have!
- logwatch
- /etc/cron.daily/0logwatch

Daily sysadmin tasks (cron revisited)

- /etc/crontab is the main system crontab
- On CentOS 5.8, we can see it's configured to run hourly, daily, weekly, and monthly jobs
- To do a system admin task daily, for example, put a script that does it into the /etc/cron.daily directory

- Similarly for hourly, weekly, monthly jobs
- /etc/crontab can be considered "root's" crontab file, but notice the "userid" field – the job will run as "userid"

logwatch

- On our CentOS 5.8 systems, logwatch is enabled by default
- There is a link to a perl script in /etc/cron.daily
- By default, it emails a summary of the logs to root, with "low" detail
- We put custom configuration in

/etc/logwatch/conf/logwatch.conf

logwatch.conf

- Examples:

MailTo = root # email sent to root
Detail = Low # low detail in the summary

- Detail can be specified as a number 0 to 10
- Another example

MailTo = tgk00001 # email my sysadmin user
Detail = High # lots of detail

logrotate

- log files grow under normal use of the system
- eventually they would fill the disks
- the logrotate facility manages the log files
- it will save a log file as a "backlog" file, and start a new empty version of that log file
- old backlogs can be deleted, or emailed
- logrotate is another process run daily through a shell script in /etc/cron.daily
- the logrotate process is configured by /etc/logrotate.conf

logrotate.conf

- how often should log files be rotated
- how big should log files get before they're rotated
- how many old backlog files should be kept
- what permissions should new empty log files get, and who should own those files

logrotate.conf

- daily, weekly, or monthly rotations
- rotate 5 : keep 5 backlog files
- specific log files can have specific config
- /etc/logrotate.conf is configured to include individual package configurations from the directory /etc/logrotate.d/
- Example:/etc/logrotate.d/yum

```
/var/log/yum.log {
    yearly    # this applies to yum only
}             # over-riding the setting in /etc/logrotate.conf
```

ac, last, lastlog, faillog

- ac: print statistics about users' connect time

man ac

$ ac –p –d

p: individual totals

d: daily totals

last

- last: listing of last logged in users

- last –t YYYYMMDDHHMMSS

who was logged in at that time

(grep for "still logged in")

lastlog

- lastlog : reports the most recent login of all users or of a given user

- -u LOGIN

for username LOGIN

- -b DAYS

before DAYS ago

- -t DAYS

since DAYS ago

psacct

- psacct is the process accounting service
- disabled by default on CentOS 5.8
- enable it with
  - chkconfig psacct on # default runlevels 2,3,4,5
  - service psacct start

lastcomm

- with psacct enabled, we can view info on previously executed commands
- --user USERNAME
- --command COMMAND

sa

- sa : summarize accounting information
- Fields:
- cpu sum of system and user time in cpu seconds
- re "real time" in cpu seconds
- k cpu-time averaged core usage, in 1k units
- k*sec cpu storage integral (kilo-core seconds)
- u user cpu time in cpu seconds
- s system time in cpu seconds

xargs

- xargs was originally intended to help run commands with "too many arguments"
- roughly, xargs helps run a command with many arguments
- by default, it breaks the arguments into large batchs
- can control how many arguments at a time

xargs

- find /home -name "*.c" -exec mv '{}' /usr/src \;
  - this command runs mv once for each C file
- find /home -name "*.c" | xargs -I '{}' mv '{}' /usr/src
  - this command runs mv once (or a few times if there are many c files)
- The "-I {}" option says "put the arguments here"

xargs

- if there are troublesome filenames (special characters, spaces, etc) then can use
  - -print0 option for find
  - -O option for xargs
- This will cause find to print NULL-delimited file names, and xargs will treat the file names as NULL-delimited (all special characters are part of the filename
- find /home -name "*.c" -print0 | xargs -0 -I{} mv {} /usr/src