

# Analyzing Cluster Log Files using Logsurfer

James E. Prewett

download@hpc.unm.edu

June 11, 2003



<http://www.hpc.unm.edu/>

# Introduction to Logsurfer

- Developed by: Wolfgang Ley and Uwe Ellerman
- homepage: `http://www.cert.dfn.de/eng/logsurf/`



# What are we interested in?

- Symptoms of a Hardware Failure
- Symptoms of a Software Failure



# What can our log analysis program do for us?

- identify problems with our cluster
- report them to the staff
- repair them without staff intervention



# What is the problem?

```
13:25:17 aloe xntpd[567]: kernel pll status change 89
13:27:09 aloe sshd[14719]: Connection closed by 129.24.241.11
13:28:05 1246 CROND[28891]: (root) CMD (run-parts /etc/cron.hourly)
13:28:10 1104 xntpd[515]: time reset (step) 0.870206 s
13:28:10 1104 xntpd[515]: synchronisation lost
13:28:34 1232 CROND[6421]: (root) CMD (run-parts /etc/cron.hourly)
13:29:15 1229 CROND[32300]: (root) CMD (run-parts /etc/cron.hourly)
13:29:19 1104 PAM_pwdb[18970]: (su) session closed for user download
13:29:42 1252 CROND[798]: (root) CMD (run-parts /etc/cron.hourly)
13:29:48 1242 CROND[13833]: (root) CMD (run-parts /etc/cron.hourly)
13:29:55 1006 CROND[7252]: (root) CMD (run-parts /etc/cron.hourly)
13:30:04 1235 CROND[31602]: (root) CMD (run-parts /etc/cron.hourly)
13:31:30 1104 xntpd[515]: synchronized to 129.24.240.21, stratum=3
```



# Finding the important stuff!

```
13:25:17 aloe xntpd[567]: kernel pll status change 89
13:27:09 aloe sshd[14719]: Connection closed by 129.24.241.11
13:28:05 l246 CROND[28891]: (root) CMD (run-parts /etc/cron.hourly)
13:28:10 ll04 xntpd[515]: time reset (step) 0.870206 s
```

## 13:28:10 ll04 xntpd[515]: synchronisation lost

```
13:28:34 l232 CROND[6421]: (root) CMD (run-parts /etc/cron.hourly)
13:29:15 l229 CROND[32300]: (root) CMD (run-parts /etc/cron.hourly)
13:29:19 ll04 PAM_pwd[18970]: (su) session closed for user download
13:29:42 l252 CROND[798]: (root) CMD (run-parts /etc/cron.hourly)
13:29:48 l242 CROND[13833]: (root) CMD (run-parts /etc/cron.hourly)
13:29:55 l006 CROND[7252]: (root) CMD (run-parts /etc/cron.hourly)
13:30:04 l235 CROND[31602]: (root) CMD (run-parts /etc/cron.hourly)
```

## 13:31:30 ll04 xntpd[515]: synchronized to 129.24.240.21, stratum=3

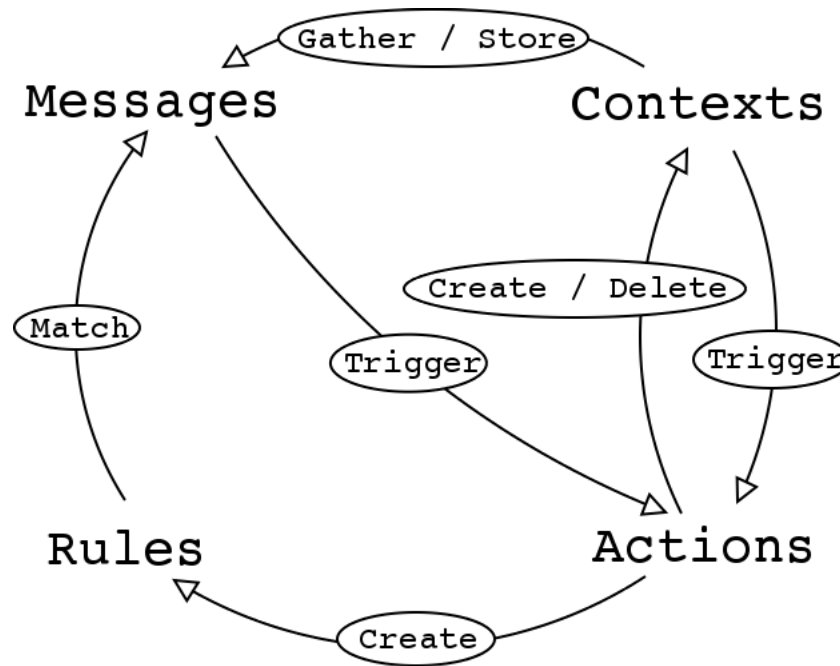


# We need contextual information!

- A single line can only show us one symptom



# Logsurfer components



# Logsurfer rules

Logsurfer Rules consist of:

- a description of matching messages
- a description of when to remove this rule
- the maximum number of seconds the rule can be active
- an action to execute when the rule is triggered



# A simple rule

```
# match every line  
'.*' —  
# never delete this rule  
— —  
# never time out  
0  
# execute /bin/cat with the matching line as its input  
pipe '/bin/cat'
```



# Logsurfer as a filter

```
# our cluster nodes each have two processors.  
# ignore messages that confirm this  
'1.* kernel: Processors: 2' - - - 0  
ignore
```



# Alert me of a hardware failure!

```
# report the problem when a machine doesn't report the  
# right number of processors on boot.  
'(1.*) kernel: Processors: ([0-9]+)' - - - 0  
    exec '/usr/adm/bin/alert \"Wrong number of \  
processors: $3 detected on node: $2 \"'
```



# Correct a problem without staff intervention

```
'(l.*) lastlog_get_entry: Error reading \  
  from /var/log/lastlog: No such file or directory'  
-- -- 0  
exec '/usr/adm/bin/fix_lastlog $2''
```



# Logsurfer dynamic rules

Logsurfer rules are dynamic in that they can:

- be created or superseded at run-time
- or
- delete themselves at run-time



# Why dynamic rules?

They are used when our response to a given message needs to change.

- remove a rule when a given message has been seen
- limit the rate at which reports are generated



# Logsurfer actions

Actions define our response to a message.

- store the message
- run an external program
- delete stored data
- create a new rule
- ignore the message



# Logsurfer contexts

Logsurfer contexts collect and store related messages.

messages may be related because they:

- come from the same process on the same host
- come from one of the members of a compute job
- have a more complex relationship



# Using contexts to monitor compute jobs

- Notice a job start
- gather log messages from all nodes involved and alert on failures from a node involved
- Notice a job finish and summarize



# Notice a job start

```
# notice when a job starts up
';S;([0-9]+).ll02.alliance.unm.edu;user=[a-z]+ .* \
exec_host=(.*) Resource_List.lcpus'
  - - - 0
      exec '/usr/logtools/bin/job_start.sh $2 $3'
```



# Monitor a newly started job

```
'Job Start jobid: ([0-9]+) nodes: ([10-9|]+)'  
  - - - 0 continue  
    rule before  
      "($3) kernel: Out of Memory: Killed process"  
    - - - 0  
      report "/usr/logtools/bin/job_failure_report.sh" $2  
  
'Job Start jobid: ([0-9]+) nodes: ([10-9|]+)'  
  - - - 0  
    open "$3"  
  - - - -  
    report "/usr/local/logtools/bin/summarize_job.pl $2" $3
```



# Notice a job finish and summarize the messages

```
'Job End jobid: ([0-9]+) nodes: ([10-9|]+)'  
-- -- 0  
report "/usr/local/logtools/bin/summarize_job.pl $2" $3
```



# Further examples



# Detecting the CRC32 Compensator Exploit

- The CRC32 attack compensator being used
- Corrupted check bytes on input
- A timeout before authentication



# Using Logsurfer to detect this Exploit

```
# look for the compensation attack message
# or the corrupted check bytes message
# do:
# start looking for a timeout before authentication
'sshd.*: fatal: Local: crc32 compensation attack: network
attack|fatal: Local: Corrupted check bytes on input.'
  - - - 0 continue
rule before
  'ssh.*: fatal: Timeout before authentication.'
  - - - 0
  report '/bin/ssh_attack_report'
  'ssh.*: fatal'
```



# Using Logsurfer to detect this exploit (continued)

```
# look for the compensation attack message
# or the corrupted check bytes message
# do:
# save these to a context
'sshd.*: fatal: Local: crc32 compensation attack: network
attack|fatal: Local: Corrupted check bytes on input.'
  --- 0
  open 'ssh.*: fatal'
  ----
  ignore
```



# Reporting a service that is down

We often want to know when a service has been down for a certain amount of time.

```
'xntpd\[.*\]: synchronization lost'  
- - - 0  
open  
''xntpd\[.*\]: synchronization lost''  
- 3600 - '/usr/adm/bin/ntp_problem.sh''
```

```
'xntpd\[.*\]: synchronized to'  
- - - 0  
delete ''xntpd\[.*\]: synchronization lost''
```

