

# Introduction

John Coleman  
Yale Library Systems  
john.coleman@yale.edu

- ◆ What is syslog?
- ◆ Why care about log files?

# Syslog Basics

- ◆ Configuration file is `/etc/syslog.conf`

- ◆ Starts up on boot

Example: Solaris is `/etc/rc2.d/s74syslog` (link to `/etc/init.d/syslog`)

- ◆ Restarting after configuration changes

```
# ps -ef | grep syslogd
```

```
root 6921  1 1 06:30:13 ?      0:00 /usr/sbin/syslogd
```

```
root 6923 6912 0 06:30:19 pts/0  0:00 grep syslogd
```

```
# kill -HUP 6921
```

/etc/syslog.conf format

facility.level

action

auth.notice

/var/log/authlog

All messages from the auth facility at notice or higher will go to /var/log/authlog

# Solaris facilities and levels

## ◆ Facilities

- ◆ user
- ◆ kern
- ◆ mail
- ◆ daemon
- ◆ auth
- ◆ lpr
- ◆ news
- ◆ uucp
- ◆ cron
- ◆ local0-7
- ◆ mark
- ◆ \*

## ◆ Levels

- ◆ emerg
- ◆ alert
- ◆ crit
- ◆ err
- ◆ warning
- ◆ notice
- ◆ info
- ◆ debug
- ◆ none

## /etc/syslog.conf

### Commonly available actions

#log to file

\*.info,mail.none            /var/adm/messages

#log to specific logged in user

\*.err                        dog, cat

#log to all logged in users

\*.crit                        \*

# log to the console (and/or tty on some systems)

\*.emerg                     /dev/console

\*.crit                        /dev/tty7

# log to a remote loghost

mail.info                    @loghost

## syslog.conf tips

◆ Use tabs only, no spaces

◆ Use **:set list** in **vi** to display control characters

```
*.notice^I^I^I/var/adm/messages$
```

◆ \*.level works in all syslogs

```
*.debug /var/adm/debug
```

◆ facility.\* and other more elaborate syntax is available in FreeBSD and Linux syslogs ([man syslog.conf](#))

```
 #(FreeBSD Example)
```

```
 # Log daemon messages at debug only
```

```
 daemon.=debug /var/log/daemon.debug
```

```
 # Log kernel firewall reports to a separate file
```

```
 !ipfw
```

```
 *. * /var/log/ipfw
```

◆ File needs to exist before logging will occur

```

#ident "@(#)syslog.conf    1.4   96/10/11 SMI" /* SunOS 5.0 */
#
# Copyright (c) 1991-1993, by Sun Microsystems, Inc.
#
# syslog configuration file.
#
# This file is processed by m4 so be careful to quote (`) names
# that match m4 reserved words.  Also, within ifdef's, arguments
# containing commas must be quoted.
#
*.err;kern.notice;auth.notice          /dev/console
*.err;kern.debug;daemon.notice;mail.crit  /var/adm/messages

*.alert;kern.err;daemon.err            operator
*.alert                                  root

*.emerg                                  *

# if a non-loghost machine chooses to have authentication messages
# sent to the loghost machine, un-comment out the following line:
auth.notice          ifdef(`LOGHOST', /var/log/authlog, @loghost)

mail.debug           ifdef(`LOGHOST', /var/log/syslog, @loghost)

#
# non-loghost machines will use the following lines to cause "user"
# log messages to be logged locally.
#
ifdef(`LOGHOST', ,
user.err          /dev/console
user.err          /var/adm/messages
user.alert        `root, operator'
user.emerg        *
)

```



Extract from [/usr/include/sys/syslog.h](#)

```
/*
 * Facility codes
 */
#define LOG_KERN      (0<<3) /* kernel messages */
#define LOG_USER      (1<<3) /* random user-level messages */
#define LOG_MAIL      (2<<3) /* mail system */
#define LOG_DAEMON    (3<<3) /* system daemons */
#define LOG_AUTH      (4<<3) /* security/authorization messages */
#define LOG_SYSLOG    (5<<3) /* messages generated internally by syslogd */
#define LOG_LPR       (6<<3) /* line printer subsystem */
#define LOG_NEWS      (7<<3) /* netnews subsystem */
#define LOG_UUCP      (8<<3) /* uucp subsystem */
#define LOG_CRON      (15<<3) /* cron/at subsystem */
/* other codes through 15 reserved for system use */
#define LOG_LOCAL0    (16<<3) /* reserved for local use */
#define LOG_LOCAL1    (17<<3) /* reserved for local use */
#define LOG_LOCAL2    (18<<3) /* reserved for local use */
#define LOG_LOCAL3    (19<<3) /* reserved for local use */
#define LOG_LOCAL4    (20<<3) /* reserved for local use */
#define LOG_LOCAL5    (21<<3) /* reserved for local use */
#define LOG_LOCAL6    (22<<3) /* reserved for local use */
#define LOG_LOCAL7    (23<<3) /* reserved for local use */
....
/*
 * Priorities (these are ordered)
 */
#define LOG_EMERG     0 /* system is unusable */
#define LOG_ALERT     1 /* action must be taken immediately */
#define LOG_CRIT      2 /* critical conditions */
#define LOG_ERR        3 /* error conditions */
#define LOG_WARNING   4 /* warning conditions */
#define LOG_NOTICE    5 /* normal but signification condition */
#define LOG_INFO       6 /* informational */
#define LOG_DEBUG     7 /* debug-level messages */
```

```
# syslogd -d
getnets() found 1 addresses, they are: 0.0.0.0.2.2
amiloghost() testing 172.30.1.2.2.2
I am loghost
cfln(*.err;kern.notice;auth.notice /dev/console)
syslogd: line 11: unknown priority name "notice "
cfln(*.err;kern.debug;daemon.notice;mail.crit /var/adm/messages)
```

## Logger shell interface to syslog

- ◆ Use for debugging syslog problems

`logger -p kern.emerg 'test kern.emerg'`

- ◆ Use in shell scripts for local error recording

- ◆ Perl interface to syslog

see: `man Sys::Syslog`

# Time Synchronization

- Critical if logs are to have any audit value
- Easiest way is to sync to clock.yale.edu from cron
  - Solaris `/usr/bin/rdate clock.yale.edu`
  - Linux `/usr/bin/rdate -s clock.yale.edu`
  - AIX `/usr/bin/setclock clock.yale.edu`

## Checking Logs

- ◆ Automate it or you won't do it
- ◆ Filter out noise or you won't read them
- ◆ You can't know every message from every current and future program that may be logging on you computers
- ◆ Strategy
  - ◆ Some things are "always interesting"
  - ◆ Some things are "never interesting"
  - ◆ Everything else is "interesting"

## Abacus Logcheck

- ♦ <http://www.psionic.com/abacus>
- ♦ Consists of 3 parts:
  - ♦ logcheck.sh shell script run from cron that mails the output of simple grep commands to the administrator
  - ♦ logtail c program that remembers the last position in a file so you only see what has come in since the last run
  - ♦ configuration files
    - ♦ logcheck.hacking
    - ♦ logcheck.violations
    - ♦ logcheck.violations.ignore
    - ♦ logcheck.ignore

# Abacus Logcheck Configuration Files

## logcheck.hacking

Labelled "ACTIVE SYSTEM ATTACK"  
(Excerpt)

"wiz"  
"WIZ"  
"debug"  
"DEBUG"  
ATTACK  
nested  
VRFY bbs  
VRFY decode  
VRFY uudecode  
VRFY lp  
VRFY demo  
VRFY guest

# Abacus Logcheck Configuration Files

## logcheck.violations

Labelled "Security Violations"

Keywords that are usually seen as negative, like 'denied' or 'refused' or 'su'

(Excerpt)

alias database  
debug  
denied  
deny  
deny host  
expn  
failed  
illegal  
nested  
permitted

# Abacus Logcheck Configuration Files logcheck.violations.ignore

Keywords that are caught by loghceck.violations but are not really violations. A reverse grep is done on these to exclude them from the report  
(Excerpt)

```
stat=refused  
stat=Deferred
```

# Abacus Logcheck Configuration Files

## logcheck.ignore

- ◆ This is a file of patterns to look for in the logs and NOT REPORT.
- ◆ Use specific strings to match
- ◆ Be careful with wildcards so you don't miss something important.
- ◆ This file is checked last and anything that makes it past is reported as "Unusual System Activity"

(Excerpt)

```
sendmail.*putoutmsg  
sendmail.*return to sender  
sendmail.*stat=  
sendmail.*timeout waiting  
smap.*host=  
smapd.*daemon running  
smapd.*delivered  
telnetd.*connect from
```

# Abacus logcheck message

Subject: dec 11/26/99:13.24 ACTIVE SYSTEM ATTACK!

## Active System Attack Alerts

=====

Nov 26 13:23:00 dec sendmail[6380]: NOQUEUE: "wiz" command from dell [172.30.1.1 ] (172.30.1.1)

## Security Violations

=====

Nov 26 13:23:00 dec sendmail[6380]: NOQUEUE: "wiz" command from dell [172.30.1.1 ] (172.30.1.1)

Nov 26 13:23:24 dec in.ftpd[6381]: refused connect from dell

Nov 26 13:24:11 dec su: 'su root' failed for coleman on /dev/pts/2

Nov 26 13:24:18 dec su: 'su root' succeeded for coleman on /dev/pts/2

## Unusual System Events

=====

Nov 26 13:23:00 dec sendmail[6380]: NOQUEUE: "wiz" command from dell [172.30.1.1 ] (172.30.1.1)

Nov 26 13:23:24 dec in.ftpd[6381]: refused connect from dell

Nov 26 13:23:47 dec ipmon[4510]: 13:23:46.727483 elxl0 @0:1 b 172.30.1.1,1192 -> 172.30.1.2,111 PR tcp len 20 60 -S IN

Nov 26 13:24:11 dec su: 'su root' failed for coleman on /dev/pts/2

Nov 26 13:24:18 dec su: 'su root' succeeded for coleman on /dev/pts/2





# Rotating/Archiving

- ◆ Most versions of Unix come with a logrotate program
  - ◆ *logrotate* on Linux
  - ◆ *newsyslog* on Solaris
- ◆ Examples for those that don't
  - ◆ <http://www.redbooks.ibm.com/abstracts/sg244564.html>
  - ◆ <http://www.performancecomputing.com/unixreview/backissu/9711/9711dae.htm>
- ◆ Set permissions strictly (only root needs to see logs)
- ◆ Timestamp rotated logs messages.112099 instead of messages1, messages2
- ◆ Compress or move to a compressed file system instead of deleting rotated logs
- ◆ scp logs to a remote host
- ◆ Run logchecker before rotating

# Remote Logging Advantages

- ◆ Audit trail if machine is compromised
- ◆ Central place to run logcheck scripts
- ◆ Dedicated machine could be tuned for logchecking
  - ◆ Big fast disk dedicated to just logs
  - ◆ Highly secure with no other services
  - ◆ Packet filtered/firewalled to allow only approved machines
- ◆ Combine local logging with critical messages also sent to a loghost

## Turn Off Incoming Remote Logging if You Don't Need It

- From the FreeBSD man page for syslogd:

...

The ability to log messages received in UDP packets is equivalent to an unauthenticated remote disk-filling service, and should probably be disabled by default.

...

- It's OFF by default on most Linux
- It's ON by default on most commercial Unix
- see [man syslogd](#) for the proper syntax
- Use a port scanner or [lsof](#) to check to see if it is listening on the network

# Remote Logging Disadvantages

- ◆UDP is a "best effort" protocol

Messages could be lost if the network or loghost daemon is flooded

- ◆Cross-platform compatibility issues

Solaris doesn't have an authpriv facility (tcp\_wrappers on Redhat example).

- ◆Be careful what you expose to the network

Nov 25 21:24:24 localhost login[1156]: FAILED LOGIN 1 FROM localhost FOR r3dXxpZ99, User not known to the underlying authentication module

- ◆DON'T do this if you ARE the 'loghost'

\*.info /var/adm/messages

\*.info @loghost

Most current syslogs seem to prevent a recursive bomb but you still get duplicate messages

# Miscellaneous

- Sending AIX error\_log messages to syslog

[http://www.austin.ibm.com/doc\\_link/en\\_US/a\\_doc\\_lib/aixprob/prbslvgd/manerrlog.htm#D298FE3647aic0](http://www.austin.ibm.com/doc_link/en_US/a_doc_lib/aixprob/prbslvgd/manerrlog.htm#D298FE3647aic0)

- HP Jet Direct Card syslog

<http://www.hp.com/cgi-bin/cposupport/cspt/ljxxx/dyndocwrap.pl?lid=general&fid=bpj05217&pid=>

- Alternate implementations

- Nsyslogd

<http://coombs.anu.edu.au/~avalon/nsyslog.html>

- Secure Syslog

<http://www.core-sdi.com/english/freesoft.html>

# Snoop of syslog

```
# snoop -x 42 port 514
```

```
Using device /dev/elxl (promiscuous mode)
```

```
dec -> dell      SYSLOG C port=33004 <34>Nov 28 21:19:45
```

```
0: 3c33 343e 4e6f 7620 3238 2032 313a 3139 <34>Nov 28 21:19
16: 3a34 3520 7375 3a20 2773 7520 726f 6f74 :45 su: 'su root
32: 2720 6661 696c 6564 2066 6f72 2063 6f6c ' failed for col
48: 656d 616e 206f 6e20 2f64 6576 2f70 7473 eman on /dev/pts
64: 2f32 /2
```

```
dec -> dell      SYSLOG C port=33004 <22>Nov 28 21:19:54
```

```
0: 3c32 323e 4e6f 7620 3238 2032 313a 3139 <22>Nov 28 21:19
16: 3a35 3420 696e 2e74 656c 6e65 7464 5b37 :54 in.telnetd[7
32: 3239 345d 3a20 636f 6e6e 6563 7420 6672 294]: connect fr
48: 6f6d 206c 6f63 616c 686f 7374 om localhost
```