Global Information Assurance Certification Paper

Log Consolidation with syslog

Donald Pitts
December 23, 2000

There may be more elaborate third party proprietary solutions for log consolidation, but the syslog capability within UNIX is simple and ubiquitous on UNIX platforms. This paper will specifically deal with the Sun Solaris environment when not noted otherwise.

Introduction to syslog

The syslog program provides a standardized framework under which programs (both operating system and applications) can issue messages to be handled by none, any, or all of the following actions based upon the configuration of syslog:

- recorded to a file (i.e. */var/adm/messages*) or device (i.e. */dev/console*)
- sent directly to a user or users if currently logged in (i.e. *root*)
- forwarded to another machine (i.e. *@loghost*)

Each message is a single line of text with an associated facility and severity. The facility can be thought of as a category that depends upon the program from which the message originates as shown in Table 1. The developer of a program decides which facility a program will utilize. In some cases, it may be configurable by the end user. Severities are hierarchical and range from *emerg* being the most important down to *debug* as the least significant as shown in Table 2. Refer to the man pages for syslog.conf and the include file /usr/include/sys/syslog.h for more details on the valid facility and severity keywords and values. [1] Syslog permits you to decide how particular facilities and severities of messages are logged, if at all. The behavior of syslog is in the process of being formally defined by the IETF Syslog Working Group. [2]

Table 1: Facility Definitions

Keyword	Description	Symbol in sys/syslog.h	syslog number
kern	Kernel	LOG_KERN	0
user	User Processes	LOG_USER	1
mail	Electronic Mail	LOG_MAIL	2
daemon	Background System Processes	LOG_DAEMON	3
auth	Authorization	LOG_AUTH	4
syslog	System Logging	LOG_SYSLOG	5
lpr	Printing	LOG_LPR	6
news	Usenet News	LOG_NEWS	7
uucp	Unix-to-Unix Copy Program (uucp)	LOG_UUCP	8
sys9 – sys14	Reserved for System (defined only on Cisco router)	Not Defined	9 - 14
cron	Daemon to Execute Scheduled Commands	LOG_CRON	15
local0 – local7	For Local Use	LOG_LOCAL*n*	16 - 23

Table 2: Severity Definitions

Keyword for syslog.conf	Keyword for Cisco Router	Symbol in syslog.h	syslog number
emerg	emergencies	LOG_EMERG	0
alert	alerts	LOG_ALERT	1
crit	critical	LOG_CRIT	2
err	errors	LOG_ERR	3
warning	warnings	LOG_WARNING	4
notice	notifications	LOG_NOTICE	5
info	informational	LOG_INFO	6
debug	debugging	LOG_DEBUG	7

Centralized Logging Host

Upon initial setup of a new machine, syslog messages are stored locally in most cases. Automatically routing your log files to a centralized location as your enterprise grows can provide the following benefits:

- Easier to analyze what may have happened (normal behavior versus curious event)
- Less likely that a successful infiltrator could corrupt or alter relocated logs
- Simplifies the archival of collected logs off-line to removable media, or even a line printer

A logging host, also known as a loghost, is a machine with significant disk storage dedicated to the sole purpose of receiving these log messages. A logging host should ideally be hardened with all external services disabled except syslog and be accessed only directly from the console for administration. [3] Normal users should not have accounts on this machine. Procedures and scripts are available detailing how to harden a Solaris machine. [4-5]

Syslog Configuration File

The configuration file for syslog (/etc/syslog.conf) is where the administrator can tailor syslog behavior based upon the facility and severity of each arriving message. Each rule consists of a set of facility/severity pairings and an associated action separated by one or more tab characters (not space characters!). Your editor may have the capability to accentuate the difference between tabs and spaces by controlling how much whitespace to use when representing tabs or you can use a search/replace function to change all spaces into tabs.

The components of a pairing are delimited by a period with the facility being followed by the severity (i.e. *kernel.info*). The facility of a message must match exactly and the severity of the message must be an equal or more important value before the corresponding action is taken. Severities in the file are essentially thresholds. So, *kernel.info* in the configuration file would match all kernel messages that might be generated except for those at the *debug* severity level, which is the only level less important than *info*. If you do not want to differentiate a particular facility from another for a particular action, then you may use an asterisk instead (i.e. *.*info*). Refer to the man pages for syslog.conf for more details. [1]

To redirect syslog messages to another machine, you must at a minimum modify the configuration of syslog on the originating system and probably the recipient system as well. On the originating system, you add a new line with an action consisting of an "at" sign immediately followed by the name of the recipient system to receive the message (in this case "another").

```
local7.debug   @another
```

The hostname used above must be resolvable and accessible by the originating system. Messages can be redirected to more than one remote loghost by using multiple lines with the same pairing, but a different action.

Notifying syslogd of configuration changes

You must manually trigger the syslogd to reload the configuration file by sending it a hangup **signal (kill -HUP *pid***). The process ID (PID) of the current invocation is kept in /etc/syslog.pid. [6] To simplify this effort, you can add the following lines to /etc/init.d/syslog before the line with "*)":

```
'reload')
      if [ -f /etc/syslog.pid ]; then
            syspid=`/usr/bin/cat /etc/syslog.pid`
            [ "$syspid" -gt 0 ] && kill -HUP $syspid
      fi
      ;;
```

Subsequently you only need issue the command **/etc/init.d/syslog reload** to get syslogd to reload its configuration file.

Syslogd Configuration

Each syslogd daemon accepts remote messages by default under Solaris. With Solaris 8, a new *-t* option is introduced whose presence disables this. Conversely, Linux requires a *-r* option before accepting remote messages. [7] Any changes in options can be configured to occur automatically on boot up by modifying the syslog startup/shutdown script (/etc/init.d/syslog). The syslog configuration file, /etc/syslog.conf, may need to be modified to specify rules to deal appropriately with any messages of interest. The syslogd daemon listens on User Datagram Protocol (UDP) port 514 by default for external messages.

Syslog Clients

Each of your client syslog processes must be configured to forward any messages of interest to your new logging host. The specifics may vary based upon each client such as a UNIX machine, Windows NT machine, router, or firewall. Some limited local logging should probably be retained on each client for use in simple problem solving as well as redundancy.

Client Configuration for UNIX

A UNIX machine can be configured to redirect its messages to a logging host by modifying the syslog configuration file to indicate the remote logging host for each of the appropriate groups of messages to be forwarded as shown earlier. The syslogd daemon must be told to reload its configuration file.

Client Configuration for Gauntlet UNIX Firewall

The Gauntlet firewall from Network Associates, Inc. (NAI) running on Solaris uses syslog for handling its messages so it works essentially the same as a generic UNIX client. [8] Testing with version 5.5 has determined that security alerts are issued with a facility of *kern* and a severity of *info* and proxies issue their messages with a facility of *daemon* and a severity of *notice*.

```
Dec  6 10:14:29 myfw unix: securityalert: tcp if=hme0 from 192.168.0.5:26004 to
                                    192.168.0.6 on unserved port 3823
Dec  6 10:15:32 myfw http-gw[28546]: connecting to host 192.168.0.6 port 80
```

Client Configuration for Windows NT

Windows NT does not natively utilize or interface with syslog. With third party software, Windows NT can act as a syslog client and redirect messages from its event log to a centralized syslog-based logging host or even run its own syslog daemon and serve as logging host. [9-10]

Client Configuration for Cisco Router

A router can be configured to redirect messages of interest to a common logging host with the command **logging *loghost_ip_address***. The command **show logging** will display the current configuration of trap logging in the router. By default the router will mark all of its messages with the facility *local7* and this can be altered to a different facility with the command **logging facility *facility_name***. Messages generated by the router are filtered by default to only send those with a severity threshold of *informational* and more severe. This threshold can be changed with the command **logging trap *severity_name*** [11-12]. Notice that the severity keywords used by the router differ slightly from syslog.conf (see Table 2). The **service timestamps log datetime** command will inject a router timestamp within each message [13].

Log messages concerning changes to the router configuration occur with a severity of *notifications* as well as interfaces going up or down and rebooting of the router. Note that the "-5-" within the Cisco message tag corresponds to the severity level the message is reported under.

```
Dec  6 09:56:52 [192.168.0.7.34.15] 336: 000335:
        .Dec  6 09:54:43 CST: %SYS-5-CONFIG_I: Configured from console by tty65
```

Adding *log-input* or *log* to the end of **access-list** commands enables logging for corresponding events. It is important to realize that this type of logging does not generate a message for each packet, but is a "representative sample" [14]. Access list violations (%SEC-6-IPACCESSLOGP) are reported with a severity of *informational*.

```
Dec  6 09:57:14 [192.168.0.7.34.15] 337: 000336:
      .Dec  6 09:55:05 CST: %SEC-6-IPACCESSLOGP:
      list 102 denied tcp 192.168.0.8 (33876) -> 0.0.0.0(23), 1 packet
```

Logging Host (Server) Configuration

The syslog configuration must be altered to contain rules dictating the proper disposition of any anticipated messages of interest that might arrive from the clients. Messages can be stored together in a single log file or multiple log files based upon the facility according to taste. Log files should be rotated periodically to avoid a file being too large to manage or archive easily. [15-16]

Syslog Relays

Syslogd can be configured as both a server and a client. Therefore it could accept messages from an originator and also forward them on behalf of the original client to yet another server on its behalf. This could be useful when sending messages through a firewall with a need to strictly limit the hosts directly involved. One negative is that any messages ultimately logged by the final recipient will only show the name of the last relay instead of the actual originating host.

Conditional configuration

A single syslog configuration file can be constructed and shared amongst normal machines as well as logging hosts and the behavior can be different. By default, Solaris is setup so each machine is its own loghost (i.e. syslog processes everything locally). Note the alias *loghost* in an example /etc/hosts on a host named *mine*:

```
127.0.0.1       localhost
192.168.0.1     mine loghost
192.168.0.2     another
```

You can reassign the host alias name *loghost* to an alternate machine named *another* and messages configured to be sent to *loghost* will now be sent to *another*. This is shown below in an updated /etc/hosts for the host named *mine*:

```
127.0.0.1       localhost
192.168.0.1     mine
192.168.0.2     another loghost
```

There are entries within the standard syslog.conf file provided with Solaris that work differently depending upon whether this machine has the "loghost" alias defined for itself or not.

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

Messages of mail.debug are stored locally within /var/log/syslog if this machine is a logging host otherwise such messages are redirected to the machine corresponding to the *loghost* alias. The syslogd daemon decides which way to handle this based upon the *loghost* alias. You may customize with your own ifdef() statements to utilize the embedded m4 preprocessor capability.

Troubleshooting configuration changes

It is useful to run the syslogd daemon in debug mode with the *-d* option when making changes to the configuration. You can do this by stopping the current syslog daemon (**/etc/init.d/syslog stop**) and starting it by hand in the foreground (**syslogd -d**). You will then see the parsing of the configuration file with any errors and then the arrival of any external messages and their disposition displayed in this terminal window. When done debugging, you can use Control-C to terminate it and then restart it normally (**/etc/init.d/syslog start**). Messages not matching any of the rules in the configuration file will be discarded, so this is something to keep in mind if things don't appear to be working as expected. It can also be useful to run a packet sniffer such as snoop to monitor any remote syslog traffic. The message may be preceded by a priority code (i.e. <190>) indicating both the facility and severity of the message. The number is determined by multiplying the facility number by 8 and adding in the severity number (refer to Tables 1 and 2 for the numbers used as input to this process). In the example below, the facility is *local7* (23) and the severity is *info* (6), which is determined by 190 = 23 * 8 + 6.

```
SYSLOG: ----- SYSLOG:  -----
SYSLOG:
SYSLOG: "<190>338: 000337: .Dec  6 09:56:43 CST: %SEC-6-IPACCESSLOGP:"
SYSLOG:
```

Sending your own messages to syslog

The **logger** UNIX command makes it very simple to send a message to syslog with control over the message text, facility, severity, and program tag. This ease of injecting messages of your own construction can be useful in testing and can also be used in conjunction with **tail –f** or your own filter to inject selected contents of plain text files into syslog.

tail -f *filename* | logger -i -p local3.err -t *program_tag*

Syslog security issues

There are known security issues with the design of the standard syslog service:

- Syslog uses the UDP protocol, which provides no inherent assurance or feedback to the sender whether the message arrived safely at its destination.
- Messages are sent in unencrypted plain text over the network, so it would not be particularly difficult for someone to intercept logging traffic and have insight into the installation.
- Anyone can direct messages of a misleading nature or significant quantity to the syslog daemon with no authentication. This might obscure the tracks of an intruder, overwhelm the network, or fill any available disk space allocated for logs.

Syslog Replacements

The IETF Syslog Working Group is endeavoring to develop a standard that outlines enhancements to secure the syslog protocol. [2] In the meantime, a number of alternative versions of syslog running on UNIX are available which purport to provide enhanced security and functionality versus the standard service. Most of them support the original syslog's capabilities to some extent for backward compatibility. Many are being continually improved.

Table 3: Syslog Replacements

Name	Author(s)	Platforms	Comments
Nsyslogd	Darren Reed	Solaris, IRIX, NetBSD, FreeBSD, BSD/OS, OpenBSD	TCP connections, SSL encrypted message delivery, queuing for deferred message delivery, supports TCP Wrapper [17]
Syslog-ng	Balázs Scheidler	Solaris, Linux, BSDi	TCP connections, filtering based on message contents, logging of complete chain of forwarding loghosts [18]
Modular Syslog	Core SDI S.A.	Linux, BSD	Successor of Secure Syslog, hash protection of logs, filtering based on message contents, logs into some SQL databases, modular construction [19]
Secure Remote Streaming (SRS)	Matt Conover, Mark Zielinski, et. al.	Solaris, Linux, BSDi	Originally written as a product for RepSec Inc. and never commercially released, no enhancement work scheduled, SSL encrypted message delivery [20]

Considerations

One or more of these actions that may help mitigate some of the shortfalls of the original syslog are:

- Configure the syslogd daemon on clients to ignore remotely arriving messages
- Log redundantly to two logging hosts and some locally as well
- Dedicate disk partitions or slices for the storage of logs to reduce the potential impact to the well being of the system itself should they be filled completely
- Utilize routers, firewalls, and/or dedicated physical networks to restrict unnecessary access to logging network traffic.
- Dedicate a logging host for each distinct segregated security zone (i.e. exterior, DMZ, and interior) to avoid weakening firewalls
- Redirect log messages over a serial line (i.e. /dev/term/b) connected to a non-port monitored serial port on the logging host and capture them into syslog locally to avoid using the syslog protocol over the general network [21]
- Examine the available syslog replacements in more detail to see if they may better meet your requirements

What's Next

Once the logs have been consolidated for the enterprise, there are many packages which can be used to examine the logs for interesting or unexpected events and potentially react in different ways. [22-23]

References

[1] "File Formats - syslog.conf(4)." 22 January 1997. URL: http://www.bama.ua.edu/cgi-bin/man-cgi?syslog.conf+4 (23 December 2000).

[2] Internet Engineering Task Force. "Security Issues in Network Event Logging (syslog)." 23 October 2000. URL: http://www.ietf.org/html.charters/syslog-charter.html (23 December 2000).

[3] Carnegie Mellon University. "Manage logging and other data collection mechanisms." 18 October 2000. URL: http://www.cert.org/security-improvement/practices/p092.html (23 December 2000).

[4] Orebaugh, Angela. "Securing Solaris." October 2, 2000. URL: http://www.sans.org/infosecFAQ/sec_solaris.htm (23 December 2000).

[5] Bezroukov, Nikolai. "Softpanorama University Pages: Solaris Hardening and Security." URL: http://www.softpanorama.org/Security/sos.shtml (23 December 2000).

[6] "Maintenance Commands - syslogd(1M)." 27 February 1997. URL: http://www.bama.ua.edu/cgi-bin/man-cgi?syslogd+1M (23 December 2000).

[7] "Tucows Linux Man Pages syslogd.8." 12 October 1998. URL: http://howto.tucows.com/man/man8/syslogd.8.html (23 December 2000).

[8] Carnegie Mellon University. "Configure firewall logging and alert mechanisms." CERT Security Improvement Modules. 2 August 1999. URL: http://www.cert.org/security-improvement/practices/p059.html (23 December 2000).

[9] Adiscon. "EventReporter." URL:

http://www.eventreporter.com/en/Product/Integrate-NT-Event-Log-into-Unix-Syslogd.asp (23 December 2000).

[10] Kiwi Enterprises. "Kiwi's Software." 23 December 2000. URL: http://www.kiwi-enterprises.com/products.htm (23 December 2000).

[11] Cisco Systems. "Logging." Improving Security on Cisco Routers. URL: http://www.cisco.com/warp/public/707/21.html#logging (23 December 2000).

[12] Cisco Systems. "Troubleshooting Commands." 19 December 2000. URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/fun_r/frprt3/frtroubl.htm (23 December 2000).

[13] Cisco Systems. "Managing the System." URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios111/mods/1mod/1cbook/1csysmgt.htm (23 December 2000).

[14] Cisco Systems. "Logging and Counter Caveats." Characterizing and Tracing Packet Floods Using Cisco Routers. URL: http://www.cisco.com/warp/public/707/22.html#3f (23 December 2000).

[15] Carnegie Mellon University. "Using newsyslog to rotate files containing logging messages on systems running Solaris 2.x." 2 March 2000. URL: http://www.cert.org/security-improvement/implementations/i041.09.html (23 December 2000).

[16] Carnegie Mellon University. "Understanding system log files on a Solaris 2.x operating system." 2 March 2000. URL: http://www.cert.org/security-improvement/implementations/i041.12.html (23 December 2000).

[17] Reed, Darren. "Nsyslogd." URL: http://coombs.anu.edu.au/~avalon/nsyslog.html (23 December 2000).

[18] Scheidler, Balázs. "syslog-ng." 6 November 2000. URL: http://www.balabit.hu/products/syslog-ng (23 December 2000).

[19] "Core FreeSoft." URL: http://www.core-sdi.com/english/freesoft.html (23 December 2000).

[20] Conover, Matt. "Index of /files/SRS." URL: http://www.w00w00.org/files/SRS/ (23 December 2000).

[21] Stokely, Celeste. "Celeste's Tutorial On Solaris 2.x Modems & Terminals." 20 May 2000. URL: http://www.stokely.com/unix.serial.port.resources/modem.html (23 December 2000).

[22] Spitzner, Lance. "Watching Your Logs." 19 July 2000. URL: http://www.enteract.com/~lspitz/swatch.html (23 December 2000).

[23] Rowland, Craig. "Psionic Logcheck Version 1.1.1." 10 May 2000. URL: http://www.psionic.com/abacus/logcheck/ (23 December 2000).