Logging and critical log files: the Decision to Effectively and Proactively Manage System logging and Log Files

Matt Morton
December 9 , 2000

There are many current and not so current documents splattered across the Internet covering pieces and parts of the topics in this paper. This document is an attempt to pull these pieces of information together into one cohesive document which will provide an overall framework to help develop an enterprise wide, automated methodology for effectively and proactively managing logging and log files. For this paper, I will focus on logging of perimeter systems, systems which are perceived to be more vulnerable to attack and less trusted than internal systems. These are normally systems such as company web servers or ftp sites, etc. located in secured and hopefully isolated, perimeters networks commonly found between a firewall and the internal (usually the company network) and external (usually the internet) networks. On these systems it is especially important to know what is happening, what has happened, and to be able to audit these events. Logging and Log files provide one of the most elemental forms of this type of system auditing. The methods I will cover apply to logging of all key systems and can be implemented throughout your organization, if required. Log files are normally generated on most types of devices which you can connect to a network. Almost all Operating systems have at least general logging facilities that are usually enabled during the normal system install. To date, the most universal logging method available is the Syslog logging facility found on most UNIX systems and also on many other operating systems. Microsoft NT and other operating systems or devices can also be configured to log to Unix Syslog servers. While Syslog is not the only logging system available, it is currently the most universally available and works with the most diverse set of clients. Note: Syslog-ng, or next generation is being developed to possibly replace Syslog. For more information please see: **http://www.balabit.hu/products/syslog-ng**. Syslog next generation has some interesting development goals, which make it worth watching, some of these include:

- configuration scheme: syslog-ng gives you a much enhanced configuration scheme, which lets you filter messages based on not only priority/facility pairs, but also on message content. You can use regexps to direct log stream to different destinations.
- forwarding logs on TCP
- hash protected log files (not yet implemented in 1.4.x, planned in 1.5.x)
- Multiplatform support: Linux, BSD, AIX, HP-UX and Solaris

Even if Syslog is not the logging method of choice, this example can be applied to any and all logging tools to help the administrator create an enterprise capable, automated, logging methodology for proactively monitoring and managing critical log data.

Why Worry About Logging and Log Files at All:

First of all, unless an administrator has volumes of free time and no problems with remembering all the minute details of their many hosts and network devices, a system setup to collect and store all relevant information automatically is essential to job survival and career advancement. Also, system logs are very important for three additional reasons. First, they provide you with a clear overview of the activity happening on your systems. Second, they provide the data for, current and future, analysis and detection of system problems and security breaches. Third, they may be your most important (and possibly your only) evidence if you ever need to go to court over a security incident. Make sure you have policies covering the length of time and methods used to maintain log data. Our current industry consensus is that logging is absolutely essential and critical for system security. So why is it that so many people continue to administer the logging of their systems inappropriately, defensively managing logs when necessary with no centralized, well thought out approach? Most likely this is due to lack of time and the understanding of how to do things differently. The first issue is one we all have to deal with. This approach will require some up-front work and planning but in the long run it will save you vast amounts of wasted time and energy. The second, issue is the one I hope to directly address with this paper, by providing an overview of an approach to logging which has worked well at our organization, and pointing you to internet publication where you can find more detail, when needed.

> **NOTE: I am assuming that your perimeter systems have been appropriately hardened and that you have implemented the necessary security layers needed to protect your environment

A Commonly Used, Easy to Implement Methodology for Effectively and Proactively Managing Logging and Log Files:

As I stated above, this approach is documented in pieces and parts all over the Internet and is commonly in use. It consists of completing the following:

Configuring All Your Perimeter Systems for Syslog Logging and Forwarding to a Centralized Log Host:

The important thing here is that you take the time to understand how Syslog logging works and how you configure it to log the things you care about and ignore the things you don't. You'll need to spend the most time on the main configuration file, Syslog.conf, this is where you decide what to log and where to log it to. Of course, if you have unlimited disk space, this is not an issue and you can just enable logging of everything. Either way, you want to understand the Syslog.conf file, usually found in /etc to make sure nothing critical is being left out. This is where you will set your perimeter systems to log to your centralized loghost (or whatever else you decide to call your centralized log repository. NOTE, the name you choose must be resolvable through your DNS or specified in the /etc/hosts file on all logging systems. Also, don't forget that by default in /etc/hosts the localhost is set to loghost, which enables logging locally. To change your perimeter systems to log to the logserver you will need to remove this entry for each of your perimeter localhosts.

Note: the following example references, Redhat Linux 6.x and Sun Solaris 2.x.

Syslog Overview:	
Default startup file:	(rc) script usually "syslog" started at runlevel 2
Default configuration File:	/etc/syslog.conf
Daemon:	/usr/sbin/syslogd /etc/syslogd (Solaris)
Logger:	User command for submitting entries to syslog logs (this is useful for testing centralized logging)
Primary Process ID:	written to /etc/syslog.pid or /var/run/syslog.pid
Restarting Syslog Daemon:	Restarting Syslog Daemon: kill –1 (HUP) syslogd (forces a reread of config.)
Example:	kill –1 `/bin/cat etc/syslog.pid`
Syslog Log file Locations:	defined in the syslog.conf file, logs must exist Before Syslog will write to them- i.e. touch /var/log /newlog then kill –HUP syslogd
Port #	514/UDP

For additional information on Syslog, please see:

- **The Syslog, Syslogd and Syslog.conf man pages for your system**
- **http://docs.sun.com Search for Syslog**
- **http://www.cert.org/security-improvement/implementations/i041.08.html (Configuring and Using Syslogd to Collect Logging Messages on Systems Running Solaris 2.x)**

Configuring an Internal System to Act as a Centralized Log Repository (loghost):

Make sure you select a system with adequate resources and especially plenty of disk space (depending on how long you want to keep logs on-line). Essentially, the Syslog configuration will be similar to the forwarding systems, the only difference is the way you start the Syslogd daemon. Usually, you will need to add a startup switch to tell the daemon to accept forwarded logs from the network. This switch varies by operating system, so be sure and check your man pages carefully. Also, you don't have to, but I would recommend that you dedicate this system to being a loghost. This will maximize performance and minimize the effort needed to troubleshoot any problems you may experience. Finally, please note that Syslogd can be started in debug mode (see man pages for details) and this can help you to troubleshoot connectivity issues and configuration problems.

For more detail, please see:

- **The resources listed for the section above**

Opening a Hole in the Firewall to Allow Logging From the Perimeter Systems to the Loghost:

This is simply a matter of allowing port 514/UDP (Syslog) to flow from your perimeter systems to your internal loghost. This is a security issue, but one that I think is well worth the risk. I have searched the

Internet and found very few Syslog related exploits. Even though this doesn't mean that they don't exist, and I am convinced that they do, with enough layers of security in place and hardened host systems, we feel this is an acceptable risk for our organization. Especially given the enormous benefit derived from using a centralized loghost. You need to think this one through and decide if this is acceptable for your organization. Whatever you decide, make sure you document what you are doing and the benefits and risks of doing it. Also, be aware that if you aren't protected with other perimeter security defenses you will be opening your site to a denial of service attack as it could be possible for someone to forward volumes of log data to your loghost, clogging network bandwidth, needlessly tying up your server and filling your disk.

Configure Swatch or Another Real Time, Alerting, Analysis and Monitoring Package to Watch Your Loghost Logs and Alert you When Something Happens:

Swatch, or the simple WATCHer, watches your log files in real time (in this case on your centralized loghost) and can alert you, in multiple ways, based on information it matches. Swatch is Perl based and matches patterns from your log files that you specify in the configuration file. When it makes a match, it can send you a console message, an e-mail, a page (you must configure a paging system separately, but this is not very difficult to do) or several other options. The current release is a beta version 3.0 although version 2.x is still available. You can obtain Swatch from ftp://ftp.stanford.edu/general/security-tools/swatch/swatch.tar. Since swatch is Perl based, it requires version 5.x of Perl and the following four Perl modules:

Time :: HiRes, Date :: Calc,

Date :: Format, File :: Tail

Perl is free and can be found in a number of locations on the internet. Setting up swatch is fairly easy to do, there is a main configuration file (usually located in the swatch directory called .swatchrc) which you need to setup based on the patterns you want to match and actions you want Swatch to take when a pattern is matched. You can find the patterns you want to look for, by looking through your current logs files and noting important incidents, for example, file system full warnings. Lance Spitzner, who has written many very good security related white papers, wrote one about Swatch called "Watching Your Logs." This paper provides a great start to using Swatch, you can find it at: http://enteract.com/~lspitz/swatch.html.

For more information about Swatch, please see:

- Lisa Conference Paper written by the author in 93, http://www.stanford.edu/~atkins/swatch/lisa93.html
- Swatch Man page http://www.stanford.edu/~atkins/swatch/swatch.html
- Swatch 3.0 README http://www.stanford.edu/~atkins/swatch/README.html

Configure Shell Scripts to Rotate, Compress and Archive your Loghost Logs:

Now that you are logging centrally, you will begin to fill the disk on your loghost much more quickly. You will need to take steps to rotate and archive your log files. The best way to do this is to write shell scripts which will automate these steps and set them up to run automatically through cron. In addition to the script below, which we run thru cron once each month, I have a cron job which deletes my compressed log files after 360 days. I do not consider myself more than a hack with shell scripting, but I have included the rotate / compress script I use for those who don't want to have to start from scratch. This should help you to think through the issues related to your centralized log files. Also, don't forget to be sure your archived logs are covered by your backup procedures so that they will be maintained for as long as is required by your company security policy. Use the scripts below at your own risk.

```
#!/bin/bash
#
# Script Name:  rotatelogs_LHS.ksh
# Description:  A script to rotate the system syslog log files on the
#       LogHost Server in a Centralized Logging Configuration
# Created: 1/16/00
# Created by: Matt Morton
# Running On:   Redhat Linux 6.2 Loghost Server
#
# Set Some Variables That Will Be Needed In The Script:
LOGEXT=`date '+.%m%d%y_%H%M'`
```

```
ALOGDIR=/var/adm
LLOGDIR=/var/log
WHICH=/usr/bin/which
CAT=`$WHICH cat`
CHMOD=`$WHICH chmod`
KILL=`$WHICH kill`
GZIP=`$WHICH gzip`
TEST=`$WHICH test`
MV=`$WHICH mv`
PS=`$WHICH ps`
TOUCH=`$WHICH touch`
# Define the Loghost Log Files
LH1=/var/log/messages
LH2=/var/log/auth.log
LH3=/var/log/daemon.log
LH4=/var/log/lpr.log
LH5=/var/log/mail.log
LH6=/var/log/news.log
LH7=/var/log/uucp.log
LH8=/var/log/user.log
# FOR LINUX OS, use the next variable to find and store the PID of syslogd
LINUXSYSLOGPID=`$PS -ew |grep syslogd |cut -c1-5`
# Rotate The Log Files
cd $LLOGDIR
$TEST -f $LH1 && $MV $LH1 $LH1$LOGEXT
$TEST -f $LH2 && $MV $LH2 $LH2$LOGEXT
$TEST -f $LH3 && $MV $LH3 $LH3$LOGEXT
$TEST -f $LH4 && $MV $LH4 $LH4$LOGEXT
$TEST -f $LH5 && $MV $LH5 $LH5$LOGEXT
$TEST -f $LH6 && $MV $LH6 $LH6$LOGEXT
$TEST -f $LH7 && $MV $LH7 $LH7$LOGEXT
$TEST -f $LH8 && $MV $LH8 $LH8$LOGEXT
# Create New Empty Log Files and Change the Permissions to 644
cd $LLOGDIR
$TOUCH $LH1 $LH2 $LH3 $LH4 $LH5 $LH6 $LH7 $LH8
#
$CHMOD 0644 $LH1 $LH2 $LH3 $LH4 $LH5 $LH6 $LH7 $LH
# Restart Syslog to Start Logging Again
$KILL -HUP $LINUXSYSLOGPID
# Compress the Rotated Logs with GZIP
$GZIP -rqbest $LLOGDIR/*_*
```

A Note About Intrusion Detection Systems (IDSs):

Many people today are implementing Intrusion Detection Systems which is a good thing and a valuable complement to any firewall / perimeter security. The problem is that many people are relying on these intrusion detection systems, instead of, not in addition to, effectively and proactively managing logging and log files. Remember, layers of security are the most effective methods to date to stop intrusions. The methods outlined above should be used to compliment your other security layers, not replace them.

For more information on intrusion detection, please the SANS intrusion detection FAQ at http://www.sans.org/newlook/resources/IDFAQ/ID_FAQ.htm.

Summary

If you complete the above recommendations you will end up with a proactive, automated, easy to manage, enterprise capable logging methodology. Not only should this help you to centrally gather your auditing information, but it will do most of the work for you and alert you when you need to take action. This is not to say that this system is bulletproof, you will still need to watch for anomalies and be prepared to dig deeper when necessary. For more information on inspecting your system and network logs for evidence of intrusion, see: http://www.cert.org/security-improvement/implementations/i003.01.html. Also, all of the above steps are useless if you don't use the centralized logs that are created and audit them consistently. Overall, this approach will dramatically cut down on the amount of administration required to maintain your most basic auditing information, that of your Syslog logs. Also, by opening a hole in your firewall to allow Syslog logging from your perimeter into your centralized loghost, you will have opened a fairly gaping security hole. I have spent some time searching for information regarding Syslog attacks and holes and have found very little information. This doesn't mean it doesn't exist or that there aren't wiley hackers exploiting this type of hole as you are reading. Your organization will need to decide if this risk is acceptable. As you know, security is a balancing act between ease of use and level of security required and about a million other factors. In our organization we have implemented other security layers to try and compensate for our Syslog risk, but even without them the benefits we have gained from this approach have far outweighed the risk.

References

Spitzner, Lance. "Watching your logs" 19 July 2000. URL: http://www.enteract.com/~lspitz/swatch.html (3 Dec. 2000)

Carnegie Mellon University "Manage logging and other data collection mechanisms." CERT Coordination Center. URL: http://www.cert.org/security-improvements/practices/p092.html (3 Dec. 2000)

Carnegie Mellon University. "Configuring and using Syslogd to collect logging messages on systems running Solaris 2.x." CERT Coordination Center. 2 Mar. 2000. URL: http://www.cert.org/security-improvements/implementations/i041.08.html (3 Dec. 2000)

Mohr, James. "Managing Your Log Files" Linux Magazine. Nov. 1999. URL:http://www.linux-mag.com/1999-11/guru_04.html (3 Dec. 2000)

Sun Microsystems. "Sun Solaris Documentation." URL:http://docs.sun.com:80/ab2/coll.40.5/REFMAN3/@Ab2PageView/idmatch(Syslog-3) (3 Dec. 2000)

Frisch, Aeleen. "Keeping Track of What Goes On: Part I." Sep. 2000 URL: http://www.linux-mag.com/cgi-bin/printer.pl?issue=2000-09&article=guru (3 Dec. 2000)

Donkers, Arthur. "Assorted Security Tips for UNIX" Systems Administrator Magazine. Nov. 1996. URL: http://www.samag.com/archive/0511/feature.shtml (3 Dec. 2000)