Global Information Assurance Certification Paper

Importance of Event Logging

October 28, 2003

GSEC Practical Assignment
Version 1.4b
Option 1

Linda V. Ornelas

Abstract

Diagnosis and recovery of any system problem begins with a review of the system's log files. These files record activity occurring on the system and indicate sources of problems. It is essential to maintain the integrity and accuracy of the log files. It is recommended that a system have a dedicated log server with regularly scheduled backups. A system should also have a dedicated log printer to record a hard copy of system events as well.

Maintaining the accuracy and integrity of the system logs ensures essential security of the log files. Additionally, it facilitates troubleshooting of problems and tracks utilization of system resources. Regular backups ensure that no logs are lost. Regular reviews of the logs allow the administrator to immediately identify any irregularities that are not typical under normal operations. A dedicated log server allows the system administrator to save time in reviewing the log files of servers on the network. The administrator eliminates the need to remotely login to every server by having the log files on a centralized host.

Table of Contents

Introduction

In any private or corporate network, there is plenty of network activity occurring at any given time. All systems should have a logging mechanism and a policy developed that can be utilized by the system administrator to monitor the activity of the system. Logging is essential to a network because it gives the ability to troubleshoot, secure, investigate or debug problems that arise in the system. The first step in troubleshooting problems begins by viewing the event logs. The logs record messages and times of events occurring on the system. It can also identify system problems that can result in server down time.

The Internet world is expanding every day, every hour, every second. There are many malicious intruders who seek opportunities to target vulnerable systems connected to the Internet. A system that is not secure and that lacks proper logging procedures can be compromised in a matter of seconds. Logging is a way of tracking down historical events that can help administrators investigate any intrusion in the system. Logging also helps to identify any failure, warning, error, success, or any type of problem occurring in our systems.

As system administrators, we should consider logging systems that have remote hosts as the log servers. We should try to convince our companies to have remote log hosts specified in the company's policies. Having this in place will help dramatically in having less to no downtime in operations. Having a dedicated person to monitor the log servers can be boring and time consuming, but there are different ways to automate the system logs to send warning messages to the system administrator either as a page or e-mails.

In the UNIX systems, there is a very important utility that handles the reporting of all the event messages generated by the system, applications, programs, utilities, hardware and many other processes running on the system. This utility is called Syslog.

Syslog

Every UNIX system has a utility called syslog. This utility was developed at the University of California Berkeley. The Syslog utility is used to log any user activity as well as any events coming from application from the system. Syslog starts when the server starts-up and reads the syslog.conf configuration file that is located in the /etc directory. This utility allows a server to send event notification messages across the network to log files that have been configured to received these events http://rfc.sunsite.dk/rfc/rfc3164.html. The syslog.conf file holds configuration parameters that will inform the applications where to send the event messages generated by the system. It also holds the messages' severity as well as their type.

The following is a sample of the syslog.conf configuration file.

```
#
# syslog configuration file.
#
# logging events in the remote loghost
auth.info                                                    @loghost.domain.com
*.err;kern.debug;auth.notice;user.none                       /dev/console
*.err;kern.debug;daemon,auth.notice;mail.crit;user.none      /var/adm/messages
lpr.debug                                                    /var/adm/lpd-errs

*.alert;kern.err;daemon.err;user.none          operator
*.alert;user.none                              root

*.emerg;user.none                              *

user.debug                                                   /dev/console
user.debug                                                   /var/adm/messages
user.alert                                     root, operator
user.emerg                                      *
```

In this example of the /etc/syslog.conf configuration file, we show various types of event messages that are sent to the /var/adm/messages file. In this file the word *loghost* should be replaced with the hostname of the remote log server in order for all the authentication system messages to be sent to the remote host as well as stored locally. Empty lines and lines started with "#" signs will be ignored. The white spaces in this file are not single-spaces. Using single-spaces in this file will produce unpredictable results or that particular line will be ignored. For this reason the white-spaces should be tabs in order for the file to be read correctly http://rfc.sunsite.dk/rfc/rfc3164.html.

The utility syslog can be configured to log all events that occur on the TCP/IP network. This utility should be running in all servers in order to log messages remotely. Syslog can be easily configured for this purpose. The only file that the system administrator has to modify is the syslog.conf configuration file, which is located in the /etc directory in most UNIX systems http://rfc.sunsite.dk/rfc/rfc3164.html.

As mentioned previously, in order to send the messages remotely, the only file that can be modified is the syslog.conf file. As in the example above, "@loghost" must be added to the auth.info line. It is a good idea to restrict user access to the log server for security reasons. Any program is able to generate syslog messages, and each message consists of the following:

- Program Name
- Facility
- Priority
- Log messages

A log file is very easy to read since it is a text file. It can be viewed with any text editor. The user may be able to pinpoint what is occurring in the system by

viewing the file and understanding all the messages that are logged in the file. The following is a sample of a log file taken from a local server.

```
Aug 5 16:03:51 hostname appl1: 1399 REFEINX 33371 BlocHead  03:51.38
Aug 5 16:03:51 hostname appl1: 1399 REFEINX 33372 BlocHead  03:51.38
Aug 5 16:03:51 hostname appl1: 1399 REFEINX 33373 BlocHead  03:51.38
Aug 5 16:03:51 hostname appl1: 1399 REFEINX 33374 BlocHead  03:51.38
Aug 5 16:03:56 hostname appl2: 2591 .tepnim Rmic  11 82 MICRO  03:56.11
Aug 5 16:03:56 hostname appl2: 2010 .tepnim NimSet =0   MICRO  03:56.11
Aug 5 16:06:02 hostname appl3: 3107 ASTNSTAT  0     2   4      06:02.92
Aug 5 16:06:02 hostname appl3: 2365 doAcknSB  NoSt  1   2      06:02.92
Aug 5 16:06:30 hostname appl3: 3107 ASTNSTAT  27    2   4      06:30.05
Aug 5 16:06:30 hostname appl3: 2365 doAcknSB  NoSt  36  2      06:30.05
Aug 5 16:35:23 hostname appl4:(1001)heavy load (1851):lifecheck inactive
35:24.00
Aug 5 16:35:44 hostname appl4:(1001)heavy load canceled:lifecheck active
35:44.00
```

In this example, hostname is the name of the server that generated these messages. Appl1, appl2, appl3 and appl4 are the name of the applications that are running in that server and generated those events. The rest of the message is a description of the event generated. The system administrator should be familiar with these messages in order to determine if the system is running in a proper manner or not.

According to the Practical UNIX & Internet Security Ch. 10 section 5, the syslog facilities and syslog priorities are summarized as follows.

Name	**Facility**
kern	Kernel
user	Regular user processes
mail	Mail system
lpr	Line printer system
auth	Authorization system, or programs that ask for user names and passwords(login, su, getty, ftpd, etc.)
daemon	Other systems daemons
news	News subsystems
uucp	UUCP subsystems
local10… local7	Reversed for site-specific use
mark	A timestamp facility that sends out a message every 20 minutes

Priority	**Meaning**
emerg	Emergency condition, such as an imminent system crash, usually broadcast to all users
alert	Condition that should be corrected immediately, such as a corrupted system database
crit	Critical condition, such as a hardware error

err	Ordinary error
warning	Warning
notice	Condition that is not an error, but possibly should be handled in a special way
info	Informational message
debug	Messages that are used when debugging problems
none	Do not send messages from the indicated facility to the selected file. For example, specifying *.debug;mail.none sends all messages except mail messages to the selected file

Source: http://www.tunzi.ch/unixadm/puis/ch10_05.htm

The name and the priority columns identify the parameters that can be used to configure the syslog.conf configuration file. These parameters are used to identify which log files to send the log messages to.

Security

As a security measure, it is very important to have a dedicated server for event logging. For example, a network intruder can erase all traces of their presence in the local host logs of the server that they hacked into. However, having a remote log server provides an additional copy of any potentially lost events at a remote host that resides across the network. We call this remote logging.

By default, syslog logs events locally in the systems. If an intruder edits the entire local log files where there is a trace of the attack, it will be very difficult for the intruder to edit the logs that were recorded in the remote loghost http://www.isg.rhbnc.ac.uk/msc/teaching/ic4/2002/groups/Group10.doc.

For redundant purposes, the system administrator should plan on having a redundant logging system. For example, if there is only one log server and this server fails, the event messages that are generated in the system will not be logged. For this reason, having only one log server is considered as a single point of failure. For instance, if an intruder is able to gain access to the system when this single log server is down and the intruder erases all their trails, there will be no trace of such an intrusion. This is why it is very important to have a redundant log server system.

In addition, the system administrator should send all event messages to a printer to have a hard copy on hand. If an intruder accesses the system and erases all the events messages from the log files that show the system was compromised, there will be no way to trace the attack through the log files. There should be a log printer connected to the system which prints every single event occurring in the system. The intruder cannot erase these events unless he/she gains

physical access to these logs. It is important that the log printer not be used as the only logging mechanism in place because the printer may jam, get powered off by mistake or for any reason may go down; therefore there will be no logs http://www.tunzi.ch/unixadm/puis/ch10_05.htm.

The log printer should be used as a backup only. This printer should be dedicated for strictly logging purposes. It is suggested to use line printers because these printers print one line at a time. It also enables the user to see every event as soon as it occurs. It is not recommended to use laser printers because they do not print the page until the entire page is completed http://www.tunzi.ch/unixadm/puis/ch10_05.htm.

Why should we log every event in the system? The system administrator can view several things such as:

- time stamps of when users login or logout of the systems
- status and debug messages of particular program or application
- system reboots
- system is shutdowns intentionally either by a user or by a program
- if network devices are having problems

Logging is very important because if a system has more than one user that logs into the system, one must have the ability to view who logged in, when they logged in, what they did, and what was run on the system. Also, logs are good to view if users abused their privileges or the security of the system has been compromised http://www.tunzi.ch/unixadm/puis/ch10_07.htm.

The log files contain timestamps, hostnames, application names, error messages and other important information that is useful for the system administrator to use if there is a need to investigate any event message or error generated by a server. There are many messages that occur on the system every second, so the user has to be able to determine which messages are critical and which are not.

With this in mind, the system administrator can trace out intruder trails in the event logs that were recorded in the log server. An investigation can be easily approached. The administrator does not have to go from server to server to find out what happened. The administrator can review all logs from that log server without having several connections to different servers open http://csrc.nist.gov/nissc/1998/proceedings/paperD1.pdf.

A problem that the syslog utility has is that it will not identify who generates the event messages. Syslog will process information from any source as long as the data coming in is in the correct format. Attackers can easily generate false event messages and log them in the system. For instance, an intruder can repeatedly send false messages indicating that a certain user or users are trying to gain root access, or messages indicating that the home partition is full. These messages

can be generated every second for several hours causing the log server to be useless since the log partitions can get full from the overflow http://www.cymru.com/Documents/syslog-bug.html.

Messages can be sent by mistake, by accident or by intruders. Syslog will not differentiate where the messages are coming from. "The receiver of the packet will not be able to ascertain that the message was indeed sent from the reported sender, or if the packet was sent from another device" http://rfc.sunsite.dk/rfc/rfc3164.html. For this reason, log servers should be protected in a secure environment, to assure that all the events generated and logged are legitimate. Firewalls and IDS systems configured properly will give a system more security.

Syslog communicates and listens for logging information through the UDP port 514 across the network. This port should be closed at the firewall to prevent intruders from sending malicious or bogus events. Syslog is not able to validate the originator of a log message. Therefore, it would not be able to differentiate valid log messages from malicious ones. Blocking this port at the firewall would add an extra layer of security for the critical vulnerability of syslog http://www.cymru.com/Documents/syslog-bug.html.

As a security measure, the users or system administrator should review the log files on a regular basis. Storing the logs in a remote host and locally in the server helps the system administrator compare both log files from both places. This allows the administrator to view if there is a discrepancy between the two or if they are identical. If there is a difference found on these files it would be obvious that there is a problem or that an intrusion has occurred. At this point the system administrator can take the appropriate actions.

The files do not give information out. These logs should be reviewed regularly, so that the administrators become familiar with what is normal and what is not normal in the system. "A system administrator who regularly checks logs will learn a lot about how the system functions, can guarantee less downtime and at the same time should notice when security breaches occur, specially if alerts are used" http://www.boran.com/security/unix1.html. The system administrator should differentiate valid log messages from false or invalid messages. This further emphasizes the need for regular reviews of the log files.

It is possible to view signs of attacks, or signs that report problems in the system or network. Also, it is important that the log server is located in a very secure network. As a good practice, when an administrator configures a UNIX system, the administrator should consider keeping the log data on different disks rather than on the same disk of the operating system and its applications. By doing this, it will not make the operating system of that server fail if the disk containing the log data gets full. Furthermore, the applications and programs can continue running with normal operations http://www.boran.com/security/unix1.html.

According to the IT Security Cookbook – Security UNIX #1, in the UNIX systems there are several types of logging files. Most of them are saved to the /var partition, and for security reasons these logs should be monitored regularly. The following list shows the various types of logging.

- *Syslog*: is a centralized logging service used extensively on most UNIX systems. Syslog stores log data in /var/adm/messages or /var/adm/syslog or /var/log depending on the configuration.
- Process accounting *(/var/adm/acct, pacct**): most UNIX systems provide accounting, but are normally switched off by default.
- Audit trail (*utmp* and *utmpx*): A log of who is logged in is kept in utmp, normally to be found in /var/adm, /etc or /usr/adm.
- *wtmp* and *wtmpx*: A log of logins, logouts, reboots are kept in these files. These are automatically trimmed if accounting is switched on, otherwise a cron entry should be created to trim then just before the year end (if there's enough disk space).
- The binary file */var/adm/lastlog* contains a log when a user last logged in and is displayed to the user on login.
- The public domain utility wu-ftp logs to /var/adm/xferlog by default.

Source: http://www.boran.com/security/unix1.html.

Troubleshooting

Furthermore, troubleshooting is an important reason why a system should have a logging system. When an application is running and it has problems, or it is not working as the programmer desires, it is easy for programmers to send debug messages to a log file to view and troubleshoot the problem. Additionally, it is good to have a logging mechanism to record events when network devices have performance problems or failures, hardware problems, or any other issues.

A remote log server helps the administrator to search or trace problems from the remote host. This eliminates the need to login to every single device and view multiple servers' log messages. This can potentially decrease the amount of time a system administrator spends on troubleshooting any particular event. As a result, if there is a production server down, centralized logging will decrease the amount of down time a production system can have since the log history will be available on the dedicated log server. When a server goes down, the administrator would want to check the reasons why the server failed before the server is brought back online http://ebuzzsaw.com/whitepapers/Case_for_Centralize_Logging.htm.

As previously mentioned, if there is a remote log server present in the system, the administrator can view and investigate the reasons for the servers' failure. It can save the administrator time in making a decision on what action to take. For

instance, should a server experience any critical problem with a disk, memory or any other hardware problem, the administrator would not have access to the logs since the server is down.

However, if a remote log server existed, the logs of the failed server would be available to the administrator to diagnose and troubleshoot the problem. If the data of that server gets lost or corrupted, there is a remote location that will contain all the information that is needed http://ebuzzsaw.com/whitepapers/Case_for_Centralize_Logging.htm.

Resource Tracking

Also, logging is important in tracking system resources. The log files will indicate whether a resource is not functioning optimally. When reviewing the log files, the system administrator can identify the need for upgrades in the system that can improve performance and utilization of the system resources. Log files can also identify any indication of potential problems with hardware or applications running on a server before it results in server down time. The system administrator can easily track and manage resources easier by utilizing the log files http://ebuzzsaw.com/whitepapers/Case_for_Centralize_Logging.htm.

Backups

System log files should be rotated and backed up for future reference and investigations. In order to save disk space in the system, the log files should be rotated and compressed. Rotating log files gives the system administrator the ability to view a log file with no problems. For instance, if there is no rotation in place and a log file grows very large, the system administrator will not be able to open or view this log file at a certain point.

Rotating log files means a script is run at the end of the day or at a fixed file size. This script would move the log file to a file with the current time and date as the name for easy identification. The file should be compressed after the renaming occurs, this will save disk space in the servers http://www.qb0x.net/projects/PassiveIDS/pids/pids.pdf. Rotating the log files helps the system administrator retrieve log files from different servers for specific date and time. This allows the investigation of a given situation or problem to be more efficient and faster, since all the logs to be reviewed have matching times.

Also, after rotating the log files the system administrator should back-up these files. A strategy should be developed as to when these log files should be backed-up. For example, keep seven days worth of log files in the system. At the end of the week these log files should be backed up to the media. Incremental backups every day will work too.

This action can be defined as a security measure. If the log server of a system is compromised and all of the data (log files) is lost or corrupted, the system administrator would be able to recuperate lost files that have been backed-up with media. The system administrator can then investigate the problem or intrusion after restoring the logs from the external media.

Conclusion

This document has indicated many reasons why it is important to log event messages in our systems. Having log files is a way to have history of everything occurring in our environment. This will help track the utilization of the system resources to anticipate and correct any potential failures before these actually occur.

To conclude, I would suggest that any given system should have various types of logging. First of all, there should be a dedicated server for logging purposes only. With this, every event generated by the system can be logged both locally and remotely.

In addition to this, there should be a redundant logging system in place. There is one main reason why we would want to have a redundant logging system. If one of the log servers fails at any given time, there would still be a second log server capturing every event in the system, even though one of the log servers failed. As mentioned before, there should also be a log printer dedicated for logging event messages only as a backup. The log printer should be considered as a backup only, not as the main logging system because if for any reason the printer fails we will not have a system logging the event messages.

If we have a log printer and both of our log servers get compromised in the event of someone hacking into the network system, we can still see and evaluate the system activity through the hard copy.

Therefore, having a log printer in addition to a dedicated log server provides sufficient redundancy to ensure that no messages are lost. To avoid any of the previously identified problems, it is essential to incorporate these guidelines to ensure security and integrity of a critical production system.

Finally, the log servers should be backed-up on a regular basis. This procedure will help system administrators retrieve event messages from prior weeks, months, etc. for investigation of an intrusion, or simply to debug application messages that can help solve application problems. Also, it is a good way to recover log files after a permanent hardware or disk failure.

References

Axelson, Stefan, et al. "An Approach to UNIX Security Logging". Chalmers University of Technology. 22 Jul. 2003 <http://csrc.nist.gov/nissc/1998/proceedings/paperD1.pdf>.

Boran, Sean. IT Security Cookbook. 2 June 2003. Boran Consulting, Inc. 23 July 2003 <http://www.boran.com/security/unix1.html>

DeFrance, Fred. "A Case for Centralized Logging". 7 December 2001. 23 Jul. 2003 <http://ebuzzsaw.com/whitePapers/Case_for_Centralize_Logging.htm>

Garfinkel, Simson and Gene Spafford. Oreilly Practical UNIX & Internet Security. Chapter 10, Sections 5 and 7. Second Edition, April 1996. 22 Jul. 2003 <http://www.tunzi.ch/unixadm/puis/index-4.htm>.

Lonvick, C. "RFC 3164 – The BSD Syslog Protocol". The Internet Society, August 2001. 22 Jul. 2003 <http://rfc.sunsite.dk/rfc/rfc3164.html>

Oubiña, F. M. and H. F. Stamati. "Passive IDS A Centralized Logging System". 12 August 2001. 23 Jul. 2003 <http://www.qb0x.net/projects/PassiveIDS/pids/pids.pdf>

"Security Weaknesses in Unix based systems". Royal Holloway University of London. 23 Jul. 2003 <http://www.isg.rhbnc.ac.uk/msc/teaching/ic4/2002/groups/Group10.doc>

Thomas, Rob. "A weakness in the Syslog Service". 2 Jul. 1997. 23 Jul. 2003 <http://www.cymru.com/Documents/syslog-bug.hmtl>