https://doi.org/10.48009/2_iis_2006_43-47

LOCKING DOWN LOG FILES: ENHANCING NETWORK SECURITY BY PROTECTING LOG FILES

Bernie Lantz, Utah State University, bernie.lantz@usu.edu
Rob Hall, Utah State University, rob.hall@usu.edu
Jason Couraud, Utah State University, jasoncouraud@cc.usu.edu

ABSTRACT

Network security is one of the major issues facing IS today. Securing a network from unauthorized intrusion and detecting unauthorized intrusion are vitally important to ensure that information on the network is free from malicious corruption. Preventing unauthorized access to a network is not always possible. Consequently, the ability to detect a network intrusion is of paramount importance. Log files are one of the means used by system administrators to detect intrusions. However, experienced hackers will often erase the log files. While this proves that network security has been penetrated, it leaves no details on how the hacker entered the system or what he accomplished. Even worse, the experienced hacker can replace the log files with a file showing normal network traffic flows. In this case, it is impossible to detect network intrusion using the log files. Therefore, the log files must be protected to be of any use in network security. This paper will discuss the best practices to use to secure the network system logs. The different methods will range from the least secure to the most secure, often depending on the number of safeguards being employed. These safeguards have been taken from academic research, current business implementations, and our own research.

Keywords: Securing Log Files, Log Files, Information Security

INTRODUCTION

Computer forensics is one of the growing concerns in the IT field [8]. Computer forensics is similar to the field of forensics. Police use the science of forensics to scour a crime scene for evidence of what happened, to whom it happened, and who did what to whom. In the case of computer forensics, the crime scene is the machine that was hacked, the victim is the entity to which the computer belongs, and the hacker is the criminal. The evidence in the case of computer forensics is the trail left by the hacker, which is recorded in the log files. In order for computer forensics to be effective, one must have accurate and trustworthy log files.

Log files are one of the means available for securing a network against intrusion. Log files record network traffic, taking note of the IP addresses trying to access your network, on which ports access was attempted, time and date of the attempt, whether or not the attempt was successful, etc. If used correctly log files can be very helpful in maintaining network security and integrity. However, for log files to provide any measure of security, logging must be activated and the log file must be checked periodically. Using log files in this manner will provide protection against novice hackers.

More experienced hackers are aware of log files and will take steps to hide their activities from the network administrator by either deleting the log file altogether or replacing the log file with a copy showing normal network activity. In the first instance, the network administrator will know that an intrusion was detected, but will have no clues as to the identity of the intruder or how the intruder entered the system. In the second instance, the network administrator will have no clues whatsoever, and will have to rely on other methods for detecting intrusion.

Assuring that log files are accurate and trustworthy is fast becoming a difficult proposition [1]. From the authors' observations, great pains are taken to exclude unauthorized access to any one system. However, little attention is given to securing log files against deletion or modification.

In Section 1, we will introduce the focus of our research. Section 2 will outline the protection procedures for securing log files. Section 3 will discuss the tangible and intangible costs and benefits of protecting log files. Section 4 will conclude the paper and provide ideas for future research.

PROTECTION PROCEDURES

A discussion on the means to implement each of the following security solutions is beyond the scope of this paper. The many different versions of UNIX and Linux make it impractical to list all the means to activate these security solutions as each version of

UNIX or Linux will perform these tasks in a different manner.

Make Log Files Append Only

Set up log files to write any new information to the end of the file only. Do not allow log files to overwrite or delete information already contained in the log files: log files could be the only evidence of the hacker's actions. In case a hacker does break into the system, this preventive measure will not allow the hacker to change the log files. A sophisticated hacker with access to root will be able to undo this change, but it will protect the system from log cleaning scripts run by the non-elite hackers, or script-kiddies [12]. Making log files append only is also an important step in setting up a separate log server, which we will discuss later.

Set Permissions

Permissions should be configured to deny access to any user who is not the system administrator [12]. Allowing read access to log files will show a hacker typical traffic patterns on the network and which ports are open, exposing potential vulnerabilities in the network. By setting the permissions too low, a hacker will be able to more easily spoof network traffic, thus hiding all trace of his presence. Additionally, the hacker will now be able to see which ports are open, potentially giving him another route of access to the system.

Password Protect Log Files

Put a password on the log files or the directory that contains the log files. If a hacker gets into your system, he will then have to break into the log files. This technique needs to be accompanied by other techniques to be effective. If a hacker breaks into a system as the root user, password protection will be of little use.

Create Duplicate Log Files

Have the log files written in more than one location. Even if one set of log files is compromised, the other set will allow you to track the intrusions. This method for protecting log files should be used in conjunction with using a separate logging server, which will be discussed later. The hacker will find one set of the log files and make changes or delete the log file and may think that he is in the clear as he has nullified one copy of the log files. By nullifying the log files, the hacker may get complacent, thinking that he has hidden all trace of his passing.

Hide Log Files

Don't put your log files in an obvious spot. Hackers know the default location for the log files. To keep hackers away from your log files, put them in an unexpected spot. This is another technique that can be used in conjunction with multiple techniques. Hiding the log files can be done in conjunction with creating duplicate log files and using a logging server. In addition, the log files can be written to the default location and also to a hidden location to keep hackers from compromising the integrity of the log files. In addition, do not give your log files obvious names such as .log or xxxlog.txt, if possible.

Use Separate Logging Server

Using a separate logging server can be a very effective way of securing log files. The logs stored on different servers such as mail, DNS, web, or file servers will be copied to this central log server. This technique adds a substantial layer of security to your log files. An attacker would have to penetrate your server and then penetrate the security on the log server. The log server could then employ other security implements that are discussed in this paper.

Use a separate server to store your log files [12]. By storing your log files on a separate server, the hackers are one more step away from your log files. This is one of the best ways to secure log files. Even if a hacker gains access to your system, he must then gain access to the log server before he is able to change the log files. If a separate logging server is used, it must be properly patched and maintained and all unused ports must be closed. A hacker could gain access to an entire network if all of the system logs are kept in an unprotected location.

In addition, steps must be taken to make sure that the hacker cannot disable the remote logging by attacking DNS. The hostname of the log server must be included in the /etc/hosts. The central log server should not be changing IP addresses too frequently, so this should not be a major problem.

In SecSyslog: an Approach to Secure Logging Based on Covert Channels, Forte et al [5] suggest using covert channels to transmit log information to a logging server. Using covert channels means transmitting your logging information using TCP/IP, ICMP tunnels, HTTP/S tunnels, or DNS. The covert channel refers to the use of field other than the data field to transmit your information. For instance, once the logging information is properly encrypted, the identification header of the IP packet can be used to

transmit the logging information. The receiving computer will know that the logging information is not in the data field and decrypt the identification header properly. The hacker that is trying to sniff the network traffic will see the packet, but will not get the logging information as it is hidden in the identification field. The capabilities and limitations of covert channels are discussed further by Forte et al [5] in the above mentioned paper.

Meng et al [9] proposed a more cost-friendly solution to storing log files on a separate server by using virtual machines. Xen, a virtual machine monitor, was installed first. Then the desired OS can be loaded on top of Xen. Xen then handles logging system activities. The nature of Xen makes it very difficult for hackers to detect, keeping the log files safe. A full treatment of Xen and system logging can be found in A Novel Method for Secure Logging System Call. More information on Xen can be found at http://www.cl.cam.ac.uk/Research/SRG/netos/xen/.

Save Log Files to Write-Once Media

Saving log files to write-once media, such as CDs, protects log files from deletion or modification. This physical security protects the log files from tampering because an attacker would have to physically impair the CD for logs to be lost. Although specific techniques for implementing this technique are not discussed in this report, software and techniques for all operating systems are widely available.

The problem with saving the log file in this manner is the noticeable difference in performance between a hard drive and a CD-Write drive. A better method would be to flush the logs to the CD-ROM periodically. Flushing the log files once a day or when the files reach a predetermined size are two common methods [12].

Encrypt Log Files

Encrypting your log files will make them difficult for the hacker to read or spoof. A hacker will not be able to make meaningful changes to the log files without the encryption key. It will be easier to delete the log files than to change them, a very noticeable action. Encrypting the log files can be combined with using a separate log server to add an extra measure of security [12].

Encrypting the log files could occur in different ways. Static log files could be archived and hashed using MD5 or sha1 secure hash standard. This does not encrypt the log files itself, but it does give the administrator an opportunity to check if someone has tampered with a log file. An administrator could checksum a log file and record this hash, if needed he could return to the file to see if it has the same hash. If even one character in the log has changed, the hash will be completely different.

The problem with encryption is that if the encryption is done on the computer violated by the hacker, it is possible that the hacker could decrypt the file. Schneier and Kelsey [11] proposed using a trusted machine, a machine relatively safe from intrusion, to develop the encryption keys for an untrusted machine or to use a network of untrusted machines to cooperatively build the encryption keys. By employing this method, the untrusted machine can encrypt the log files while at the same time being unable to decrypt them. In this manner the hacker's choices for dealing with log files are limited to deletion, an obvious indication of intrusion.

In addition to encrypting log files Jiang et al [6] suggest adding a message authentication code (MAC) to each of the records in the log file. The MACs are time sensitive, so it is very difficult to spoof log records for a prior time period. The MAC is calculated for the first time period and is then used to calculate the MAC for the next time period. The first MAC is then irretrievably deleted to protect the previous log files from being spoofed.

COSTS AND BENEFITS

Tangible

Tangible Costs

Keeping log files secure is not going to be free. To secure log files requires expertise at the very least. Without buying new equipment, changing the permission levels, making log files append only, etc. all require a certain amount of expertise with UNIX and Linux systems. To implement solutions using write-once media or a separate logging server requires an outlay of capital before these solutions can be implemented, in addition to the cost of expertise required. Storage for write-once media is another cost that must be accounted for when evaluating secure logging solutions.

Tangible Benefits

Network outages caused by hackers can be extremely expensive, costing companies millions of dollars each hour that the network is out of service[10]. The

ability to detect and resolve a network intrusion can save a company millions of dollars in lost revenue. Log files are one of the most important tools for after the fact intrusion detection. Therefore, uncompromised log files are of paramount importance.

For example, using the Network Downtime Cost Calculator from Sciodata, found at http://www.sciodata.com/CostofDowntimeCalc.asp, the cost of one hour of network downtime can be calculated. The following table was generated using the calculator found at Sciodata.com.

Table 1. Network Outage Costs

Employee Productivity	
Annual Revenue of Company	$20,000,000
Number of Staff Affected	100 employees
Average Revenue per Employee / Hour	$100
Lost Data and Rework	
Time required to replace lost data and re-run processes (enter total man hours)	40 hours
Lost Data and Rework Cost	$4,000
Employee Productivity Loss	
Hours of System Outage	1 hour
Lost Employee Productivity	$10,000
Sales Opportunity Cost	
Number Sales per Year	10,000 sales
Number of Customers	2,000 customers
Estimate Sales Lost Due to Outage	4 sales
Sales Opportunity Loss	$8,000
Customer and Reputation Cost	
Estimate Lost Customers	2 customers
Customer and Reputation Cost	$20,000
Cost of Network Downtime	$42,000

Intangible

Intangible Costs

Intangible costs are hard to quantify, but can still have dramatic impact on a company's bottom line. Loss of consumer confidence is one such intangible cost. If a company's network is seen as insecure or prone to failure, consumers will take their business to a company with a reputation for network security. In addition, employee satisfaction may suffer. Dissatisfied employees do not work as hard as they could and some may start looking for alternative employment.

Intangible Benefits

At the opposite end, a company with a reputation for security will enjoy the benefits of greater consumer confidence and greater employee satisfaction. Consumers who have a good opinion of a company will recommend that company to their friends and family. While satisfied employees will work harder, show greater loyalty, and will try to promote their employer. The effect of these benefits is hard to measure, as with the intangible costs, but will surely be felt.

CONCLUSION

For a network to enjoy the greatest benefit from these methods of log file protection more than one method should be employed in a layered manner. Kawaguchi et al [7] proposed a solution that combines elements of storing logs on a remote server, authentication by a trusted sources, and addition of MACs to the log entries.

The decision to which methods are the correct methods for any one business will have to be determined on a case by case basis. A one-size-fits-all method for logging security does not exist. What is right for one company will not necessarily be right for another, especially if the companies are in different industries. For any one method to be successful, an evaluation of current security needs must be performed. In addition to the evaluation of current security needs, a feasibility study should be conducted before implementation to avoid failure before successful implementation.

Future Research

Future research suggested by this topic is to evaluate the various combinations of the suggested security methods. The purpose of the evaluation will be to determine which methods work well together, which combinations of methods are the most secure, which methods are technically and economically feasible, and how best to layer the suggested methods for best effect.

REFERENCES

1. Adelstein, F. (2006). Live forensics: Diagnosing your system without killing it first. *Communications of the ACM, 49(2)*, 63-66.
2. Bauer, K. (2000). Who Goes There?. *Online, 24*(1), 25.
3. Fansler, K. W. & Riegle, R. (2004). Using log file analysis as an instructional design tool. *Online Classroom, Jan2004,* 1-4.
4. Fichter, D. (2003). Server Logs: Making Sense of Cyber Tracks. *Online, 27*(5), 47-50.
5. Forte, D, Maruti, C., Vetturi, M. & Zambelli, M. (2005). SecSysLog: An approach to secure logging based on covert channels. *Proceedings on the First International Workshop on Systematic Approaches to Digital Forensic Engineering,* 248-263.
6. Jiang, T., Liu, J. & Han, Z. (2004). Secure audit logs with forward integrity message authentication codes. *ICSP'04 Proceedings*, 2655-2658
7. Kawaguchi, N., Ueda, S., Obata, N., Miyaji, R., Kaneko, S., Shigeno, H., et al. (2004). A secure logging scheme for forensic computing. *Proceedings of the 2004 IEEE,* 386-393.
8. Kessler, M. G. (2006). Kessler's Corner: The growing field of computer forensics. *The Kessler Report, 9*(1), 7.
9. Meng, J., Lu, X., & Dong, G. (2005). A novel method for secure logging system call. *Proceedings of ISCIT2005,* 924-927.
10. Niccolai, J. (2000). *Analyst Puts Hacker Damage at $1.2 Billion and Rising.* Retrieved July 13, 2006 from http://www.infoworld.com/articles/ic/xml/00/02/10/000210icyankees.html
11. Schneier, B. & Kelsey, J. (1999). Secure audit logs to support computer forensics. *ACM Transactions on Information and System Security, 2*(2), 159-176.
12. Skoudis, E. (2001). *Defending Your Log Files* Retrieved May 2, 2006 from http://www.phptr.com/articles/article.asp?p=23464&seqNum=1.