ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 226 – 230 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

Maintaining Log Management to the Cloud And Providing confidentiality

M Hanmanthu Naik[1] M.Udaya Prakash Reddy[2]
[1]Student, Department Of Computer Science and Engineering,
Malla Reddy Institute Of Engineering and Technology.
email-hmtnaik@gmail.com

[2]Assistant professor, Department Of Computer Science and Engineering
Malla Reddy Institute Of Engineering and Technology.
email- udhayreddi@gmail.com

Abstract:-
Securely maintaining log records over extended periods of time is very important to the proper functioning of any organization. Integrity of the log files and that of the logging process need to be ensured at all times. In addition, as log files often contain sensitive information, confidentiality and privacy of log records are equally important. However, deploying a secure logging infrastructure involves substantial capital expenses that many organizations may find overwhelming. Delegating log management to the cloud appears to be a viable cost saving measure. In this paper, we identify the challenges for a secure cloud-based log management service and propose a framework for doing the same.

Keywords -Cloud computing, logging, privacy, security.

I INTRODUCTION

ALOG IS a record of events occurring within an organization's system or network [1]. Logging is important because log data can be used to troubleshoot problems, fine tune system performance, identify policy violations, investigate malicious activities, and even record user activities. Log records play a significant role in digital forensic analysis of systems. Regulations such as HIPAA [2], Payment Card Industry Data Security Standard [3], or Sarbanes-Oxley [4]often require forensically sound preservation of information.
To comply with these regulations, evidence produced in a court of law, including log records, must be unbiased, non-tampered with, and complete before they can be used. Since log files contain record of most system events including user activities, they become an important target for malicious attackers. An attacker, breaking into a system, typically would try not to leave traces of his or her activities behind. Consequently, the first thing an attacker often does is to damage log files or interrupt the logging services. Furthermore, the sensitive information contained in log files often directly contributes to confidentiality breaches. An example of this is when logs contain database transaction data. Frequently, log information can be helpful to an attacker in gaining unauthorized access to system. One example of this is the case when a user mistakenly enters her password in the username field while logging into a system. Logging programs will store the password as the user-id to record the information that a user has failed to log in. Last, but not least, information in log file can also be used to cause privacy breaches for users in the system since the log file contains record of all events in the system. In light of the above observations, it is very important that logging be provided in a secure manner and that the log records are adequately protected for a predetermined amount of time (maybe even indefinitely). Traditional logging protocols that are based on syslog [5] have not been designed with such security features in mind. Security extensions that have been proposed, such as reliable delivery of syslog [6], forward integrity for audit logs [7], syslog-ng [8], and syslog-sign [9], often provide either partial protection, or do not protect the log records from end point attacks. In addition, log management requires substantial storage and processing capabilities.

The log service must be able to store data in an organized manner and provide a fast and useful retrieval facility. Last, but not least, log records may often need to be made available to outside auditors who are not related to the organization.

Deploying a secure logging infrastructure to meet all these challenges entails significant infrastructural support and capital expenses that many organizations may find overwhelming. The emerging paradigm of cloud computing promises a low cost opportunity for organizations to store and manage log records in a proper manner.

ISSN 2278-3091
International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 226 – 230 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

Organizations can outsource the long-term storage requirements of log files to the cloud. The challenges of storing and maintaining the log records become a concern of the cloud provider. Since the cloud provider is providing a single service to many organizations that it will benefit from economies of scale. Pushing log records to the cloud, however, introduces a new challenge in storing and maintaining log records. The cloud provider can be honest but curious. This means that it can try not only to get confidential information directly from log records, but also link log record related activities to their sources. No existing protocol addresses all the challenges that arise when log storage and maintenance is pushed to the cloud.

In this paper, we propose a comprehensive solution for storing and maintaining log records in a server operating in a cloud-based environment. We address security and integrity issues not only just during the log generation phase, but also during other stages in the log management process, including log collection, transmission, storage, and retrieval. The major contributions of this paper are as follows. We propose an architecture for the various components of the system and develop cryptographic protocols to address integrity and confidentiality issues with storing, maintaining, and querying log records at the honest but curious cloud provider and in transit. We also develop protocols so that log records can be transmitted and retrieved in an anonymous manner over an existing anonymizing infrastructure such as Tor [16]. This successfully prevents the cloud provider or any other observer from correlating requests for log data with the requester or generator. Finally, we develop a proof-of-concept prototype to demonstrate the feasibility of our approach and discuss some early experiences with it. To the best of our knowledge, ours is the first work to provide a complete solution to the cloud based secure log management problem.

The remainder of this paper is organized as follows. Section II identifies the security properties that must be ensured in a cloud-based log management system. Section III discusses some of the existing secure logging protocols in light of these properties. In Section IV, we present the architecture for our cloud-based secure logging facility and describe the services provided by each component. The threat model is discussed in V. Section VI presents the main cryptographic techniques used to protect the log records in transit and at the cloud provider. We reason why our protocol guarantees the desirable security properties. Section VII describes the protocols used to anonymously store and retrieve the log files from the cloud. We present a brief discussion of the implementation in Section VIII and some performance study that we have undertaken. We conclude this paper with a discussion of our future work in Section IX.

II RELATED WORK

A number of approaches have been proposed for logging information in computing systems. Most of these approaches are based on *syslog* which is the de facto standard for network wide logging protocol. The syslog protocoluses UDP to transfer log information to the log server. Thus, there is no reliable delivery of log messages. Moreover, syslog does not protect log records during transit or at the end-points. *Syslog-ng*is a replacement that is backward compatible with syslog. Some of its features include support for IPv6, capability to transfer log messages reliably using TCP, and filtering the content of logs using regular expressions. Syslogng prescribes log record encryption using SSL during transmission so as to protect the data from confidentiality and integrity breaches while in transit. However, syslog-ng does not protect against log data modifications when it resides at an end-point. Syslog-sign is an another enhancement to syslog that adds origin authentication, message integrity, replay resistance, message sequencing, and detection of missing messages by using two additional messages—"signature blocks" and "certificate blocks." Unfortunately, if signature blocks associated with log records get deleted after authentication, tamper evidence and forward integrity is only partially fulfilled. Syslog-sign also does not provide confidentiality or privacy during the transmission of data or at the end points.

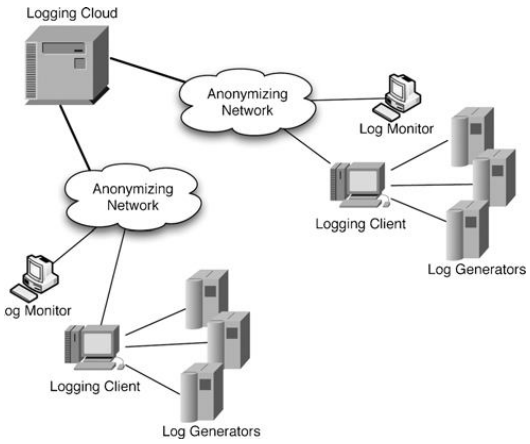
Syslog-pseudo proposes a logging architecture to pseudonymize log files. The main idea is that log records are first processed by a pseudonymizer before being archived. The pseudonymizer filters out identifying features from specific fields in the log record and substitutes them with carefully crafted pseudonyms. Thus, strictly speaking, this protocol does not ensure correctness of logs. That is, the log records that are stored are not the same as the ones that are generated.
The other problem with this paper is that while the protocol anonymizes each log record individually it does not protect log records from attacks that try to correlate a number ofanonymized records. Our objective, on the other hand, is precisely this. Moreover, privacy breaches that can occur from scenarios such as the user erroneously typing the user id in a password field (as discussed earlier) or identifying information available in fields that are not anonymized, are also not addressed in this

ISSN 2278-3091
International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 226 – 230 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

paper. TheAnonymouse log file anonymizer[12] performs a similar anonymization of identifying information by substitution with default values or more coarse values. However, the problem with this paper is that if the original values are needed in investigation, they cannot be restored. Neither syslog-pseudo nor anonymouse log file anonymizer protects log records from confidentiality and integrity violations and other end-point attacks.

Reliable-syslog [6] aims to implement reliable delivery of syslog messages. It is built on top of the blocks extensible exchange protocol (BEEP [13]) which runs over TCP to provide the required reliable delivery service. The Reliablesyslog Protocol allows device authentication and incorporatesmechanisms to protect the integrity of log messages and protect against replay attacks of log data; however it does not prevent against confidentiality or privacy breaches at the end-points or during transit.

The notion of *forward-integrity* of log records was proposed by Bellare and Yee [7] to protect precompromise log data from postcompromise insertion, deletion, modification, and reordering. Forward integrity is established by a secret key that becomes the starting point of a hash-chain. The hash-chain is generated by a cryptographically strong one-way function in which the key is changed for every log record. Schneierand Kelsey [14] also proposed a logging scheme that ensures forward integrity. This scheme is based on forward-secure message authentication codes and one-way hash chains similar to that suggested by the Bellare-Yee protocol. However, the major problem of both schemes is that both require online trusted servers to maintain the secret key and verification of log records. If the trusted server is attacked or compromised, it breaks the security of the logging scheme. Holt [15] improves upon the Schneier-Kelsey protocol by incorporating public verifiability of log records. However, this scheme being a public-key based scheme, the overhead is significantly more.None of these three schemes consider the privacy concernsof storing and retrieving log records. In addition, all thesethree scheme suffer from truncation attacks where an attackerdeletes a contiguous subset of log records from the very end.This attack can be launched on log records that have yet to bepushed to the trusted server where they are finally stored. Maand Tsudik address this problem in [10]. They use the notion of forward-secure sequential aggregate authentication in which individual signatures are folded into one single aggregated signature and all other signatures are deleted. An attacker cannot recreate this signature without knowing all previous signatures. The concern with Ma and Tsudik's scheme is that it is very expensive to verify only a single log record (or block, depending on what is considered a unit of log entries over which the aggregated message authentication code is generated). The whole chain of log records over which the accumulated signature has been generated, needs to be verified



to achieve this. Further, Ma and Tsudik's scheme does not address the confidentiality and privacy problems with log file storage and retrieval.

III SYSTEM PROCESS

The overall architecture of the cloud based secure log management system is shown in Fig. 1. There are four major functional components in this system.

1) *Log Generators*: These are the computing devices that generate log data. Each organization that adopts the cloud-based log management service has a number of log generators. Each of these generators is equipped with logging capability. The log files generated by these hosts are not stored locally except temporarily till such time as they are pushed to the logging client.

2) *Logging Client or Logging Relay*: The logging client is a collector that receives groups of log records generated by one or more log generators, and prepares the log data so that it can be pushed to the cloud for long term storage. The log data is transferred from the generators to the client in batches, either on a schedule, or as and when needed depending on the amount of log data waiting to be transferred. The logging client incorporates security protection on batches of accumulated log data and pushes each batch to the logging cloud. When the logging client pushes log data to the cloud it acts as a logging relay. We use the terms logging client and logging relay

ISSN 2278-3091
International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 226 – 230 (2014)
Special Issue of ICETETS 2014 - Held on 24-25 February, 2014 in Malla Reddy Institute of Engineering and Technology, Secunderabad– 14, AP, India

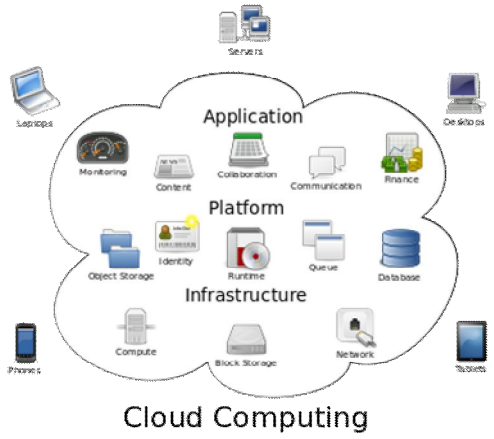
interchangeably. The logging client or relay can be implemented as a group of collaborating hosts. For simplicity however, we assume that there is a single logging client.

3) *Logging Cloud*: The logging cloud provides long term storage and maintenance service to log data received from different logging clients belonging to different organizations. The logging cloud is maintained by a cloud service provider. Only those organizations that have subscribed to the logging cloud's services can upload data to the cloud.

4) *Log Monitor*: These are hosts that are used to monitor and review log data. They can generate queries to retrieve log data from the cloud. Based on the log data retrieved, these monitors will perform further analysis as needed. They can also ask the log cloud to delete log data permanently, or rotate logs.

IV CLOUD COMPUTING

The issue of cloud computing has been obtaining immense coverage in recent years for variety of reasons – just like the new cookie rules, the word 'cloud' offers journalists the chance to return up with simple wordplay headings regarding "storm clouds" or "cloudy outlook". Moreover, with a myriad of various firms massive (Apple, Microsoft, Google) and tiny giving a range of cloud product to each organisation and customers, the horizon is clouded (see what I did there?) with press releases, interviews and advertorials, all designed to



persuade folks to dispense with their information. rather than maintaining a technical infrastructure so as to control, calculate or no matter else you are doing along with your info, you log in through the net and know on the cloud provider's systems instead. there's no single cloud model, then beneath the umbrella (I'm at it again), you would possibly opt for a wholesale transfer of services, place one a part of your organization onto the cloud, select one cloud service (i.e. email, just like the pilot being administrated by Warwickshire Council), or perhaps simply however a cloud back-up.

Applications:

The original motivation for identity-based secret writing is to assist the preparation of a public key infrastructure. during this section, we have a tendency to show many alternative unrelated applications.

V CONCLUSION

Logging plays a very important role in the proper operation of an organization's information processing system. However, maintaining logs securely over long periods of time is difficult and expensive in terms of the resources needed. The emerging paradigm of cloud computing promises a more economical alternative. In this paper, we proposed a complete system to securely outsource log records to a cloud provider. We reviewed existing solutions and identified problems in the current operating system based logging services such as syslog and practical difficulties in some of the existing secure logging techniques. We then proposed a comprehensive scheme that addresses security and integrity issues not just during the log generation phase, but also during other stages in the log management process, including log collection, transmission, storage and retrieval.

One of the unique challenges is the problem of log privacy that arises when we outsourced log management to the cloud. Log information in this case should not be casually linkable or traceable to their sources during storage, retrieval and deletion. We provided anonymous upload, retrieve and delete protocols on log records in the cloud using the Tor network. The protocols that we developed for this purpose have potential for usage in many different areas including anonymous publish-subscribe.

VI REFERENCES

[1] K. Kent and M. Souppaya. (1992). *Guide to Computer Security Log Management, NIST Special Publication 800-92* [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-92/SP800-92.pdf

[2] U.S. Department of Health and Human Services.(2011,Sep.).*HIPAA—General Information*

[Online]. Available: https://www.cms.gov/ hipaageninfo

[3] PCI Security Standards Council. (2006, Sep.) *Payment Card Industry(PCI) Data Security Standard—Security Audit Procedures Version 1.1*[Online].Available: https://www.pcisecuritystandards.org/pdfs/pci−audit−procedures−v1-1.pdf

[4] Sarbanes-Oxley Act 2002. (2002, Sep.). *A Guide to the Sarbanes-Oxley Act* [Online]. Available: http://www.soxlaw.com/

[5] C. Lonvick, *The BSD Syslog Protocol*, Request for Comment RFC 3164, Internet Engineering Task Force, Network Working Group, Aug. 2001.
[6] D. New and M. Rose, *Reliable Delivery for Syslog*, Request for Comment RFC 3195, Internet Engineering Task Force, Network Working Group, Nov. 2001.

[7] M. Bellare and B. S. Yee, "Forward integrity for secure audit logs," Dept. Comput. Sci., Univ. California, San Diego, Tech. Rep., Nov. 1997.

[8] BalaBit IT Security (2011, Sep.). *Syslog-ng—Multiplatform Syslog Server and Logging Daemon* [Online].Available:http://www.balabit.com/network-security/syslog-ng

[9] J. Kelsey, J. Callas, and A. Clemm, *Signed Syslog Messages*, Request for Comment RFC 5848, Internet Engineering Task Force, Network Working Group, May 2010.

[10] D. Ma and G. Tsudik, "A new approach to secure logging," *ACM Trans. Storage*, vol. 5, no. 1, pp. 2:1–2:21, Mar. 2009.

[11] U. Flegel, "Pseudonymizing unix log file," in *Proc. Int. Conf. Infrastruture Security*, LNCS 2437. Oct. 2002, pp. 162–179.

[12] C. Eckert and A. Pircher, "Internet anonymity: Problems and solutions," in *Proc. 16th IFIP TC-11 Int. Conf. Inform. Security*, 2001, pp. 35–50 .

[13] M. Rose, *The Blocks Extensible Exchange Protocol Core*, Request for Comment RFC 3080, Internet Engineering Task Force, Network Working Group, Mar. 2001.

[14] B. Schneier and J. Kelsey, "Security audit logs to support computer forensics," *ACM Trans. Inform. Syst. Security*, vol. 2, no. 2, pp. 159– 176, May 1999.

[15] J. E. Holt, "Logcrypt: Forward security and public verification for secure audit logs," in *Proc. 4th Australasian Inform. Security Workshop*, 2006, pp. 203–211.

[16] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The secondgeneration onion router," in *Proc. 12th Ann. USENIX Security Symp.*, Aug. 2004, pp. 21–21.