Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 2, February 2014, pg.788 – 793

RESEARCH ARTICLE

DELEGATING LOG MANAGEMENT TO THE CLOUD USING SECURE LOGGING

[1]Mary Michael, [2]J.Mary Metilda

[1]PG scholar, Department of Computer Science and Engineering, Anna University Chennai, India
[2]Assistant Professor, Department of Computer Science and Engineering, Anna University Chennai, India
[1],[2]Roever Engineering College, Perambalur
[1] marymichael990@yahoo.in, [2] cse.metilda@gmail.com

Abstract-Many security issues are involved in log management. Integrity of the log file and that of the logging process need to be ensured at all time. Main goal of a log manager is to provide high bandwidth and low level inactivity. In many real world applications and sensitive information must be kept in log files on an untreated machine. The event that an attacker captures this machine and would like to guarantee that he will gain little or no information from the log files and to limit his ability to corrupt the log file. It describes a computationally cheap method for making all log entries generated prior to the logging machine's compromise impossible for the attacker to read and also impossible to undetectably modify or destroy. In this work, find out the challenges for a secure cloud based log management service. It Provide a comprehensive solution for storing and maintaing log records in a server operating in cloud-based environment. Also address security and integrity issues not only just during the log generation phase but also during other stage in the log management. It implement how to store secure log file in cloud and that file we can change read, write, delete, upload and download.

*Index Terms—****Cloud computing, logging, privacy, Integrity, security.***

I. INTRODUCTION

A log is a record of the events occurring within an organization's systems and networks. Logs are composed of log entries; each entry contains information related to a specific event that has occurred within a system or network. Many logs within an organization contain records related to the security.

These securities of logs are generated by many sources. The security log management is the process for generating, transmitting, storing, analyzing and disposing of security log data. Log management is essential to ensuring that security of log records is stored in sufficient detail for an appropriate period of time. Routine log analysis is beneficial for identifying security incidents, policy violations, fraudulent activity, and operational problems. Logs are also useful when performing auditing and forensic analysis, supporting internal investigations, establishing baselines, and identifying operational trends and long-term problems.

There are many approaches developed for the log security. But traditional logging protocols that are based on syslog have not been designed with security features in mind. Security extensions that have been proposed, such as reliable delivery of syslog, forward integrity for audit logs, syslog-ng, and syslog-sign, often provide either partial protection, or do not protect the log records from end point attacks .Main disadvantage of existing system is provide the security and integrity during the log generation phase. Here 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 are 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. Log records can be transmitted and retrieved in an anonymous manner. This successfully prevents the cloud provider or any other observer from correlating requests for log data with the requester or generator. Also develop a proof-of-concept prototype to demonstrate the feasibility. Implement AES algorithm that for log monitor and log generator.

II. PROPOSED WORK

The major contributions are the 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. One of the main disadvantages of existing system is that it cannot point out the confidentiality and privacy with log file storage and retrieval.

The logging client uploads data in batches where each batch is delimited by a start-of-log record and an end-of log record. The cloud provider will accept log records only from its authorized clients. Thus, during upload a logging client has to authenticate to the logging cloud to prove that the client had obtained prior authorization from the logging cloud to use the latter’s services. However, it can't want the identity of the logging client to be linked to any of its transactions including the authentication process. For this purpose develop four different protocols for anonymous upload, retrieval and deletion of log data.

2.1 Anonymous Upload-Tag Generation

An uploaded log batch of log records needs to be indexed by a unique key value. However, it needs to ensure that this key value cannot be traced back to the logging client that uploaded the data nor the log monitor that seeks the data. For this purpose, the log data is stored at the cloud indexed by an anonymously generated upload-tag. This upload-tag is created by the logging client in cooperation with the log monitor. It has the property that it is created by publicly available information.

To retrieve log data from the cloud, the log monitor sends a retrieve request to the logging cloud using an upload tag. The upload-tag is not sent in an encrypted manner. Thus, any adversary can use the upload-tag to retrieve the corresponding log data. However, the log data can be deciphered if and only if the corresponding decryption key is available.

2.2 Anonymous Upload

The entity that needs to upload the log data sends a request message. In the request message contain with the upload-tag of corresponding the desired log data. Using the anonymous communication channel, the logging monitor send log data to the logging cloud .None of the values in the upload message individually or in a group can be tied to the logging client. The logging cloud and logging monitor send formatted message to the logging cloud in order to the upload or retrieve any piece of information

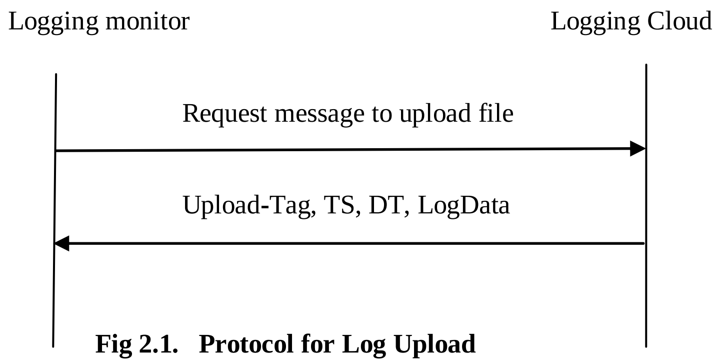

Fig 2.1. Protocol for Log Upload

2.3 Anonymous Retrieve

This protocol is straight forward. The entity that needs to download log data (most of the time the log monitor), sends a retrieve request (anonymously)together with the upload-tag corresponding to the desired log data. The logging cloud gets the data from its storage and sends it over the anonymous channel to the requester. The cloud provider does not have to authenticate the requester. This is because, by virtue of the log batches being encrypted, the retrieved data is useful only to those who have the valid decryption keys.

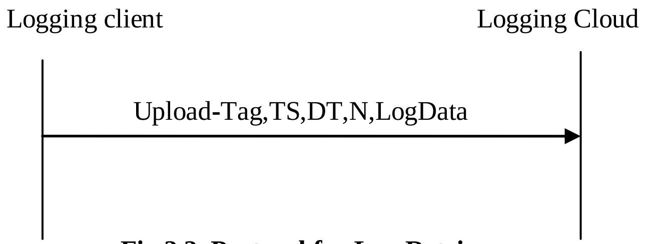

Fig 2.2. Protocol for Log Retrive

2.4 Anonymous Delete

To delete log data, the delete requester sends an appropriate delete message to the logging cloud. In response, the logging cloud throws a challenge to the requester. The requester proves authorization to delete by presenting a correct delete tag.

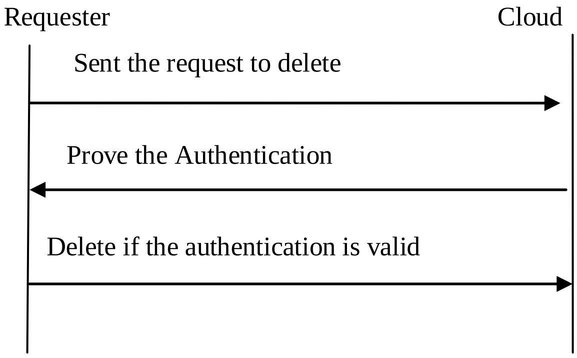
Requester
Cloud
Sent the request to delete
Prove the Authentication
Delete if the authentication is valid

Fig 2.3. Protocol for Log Delete

III. SYSTEM DESIGN

The architecture of the proposed system is shown in fig.3.1, which is explained below.

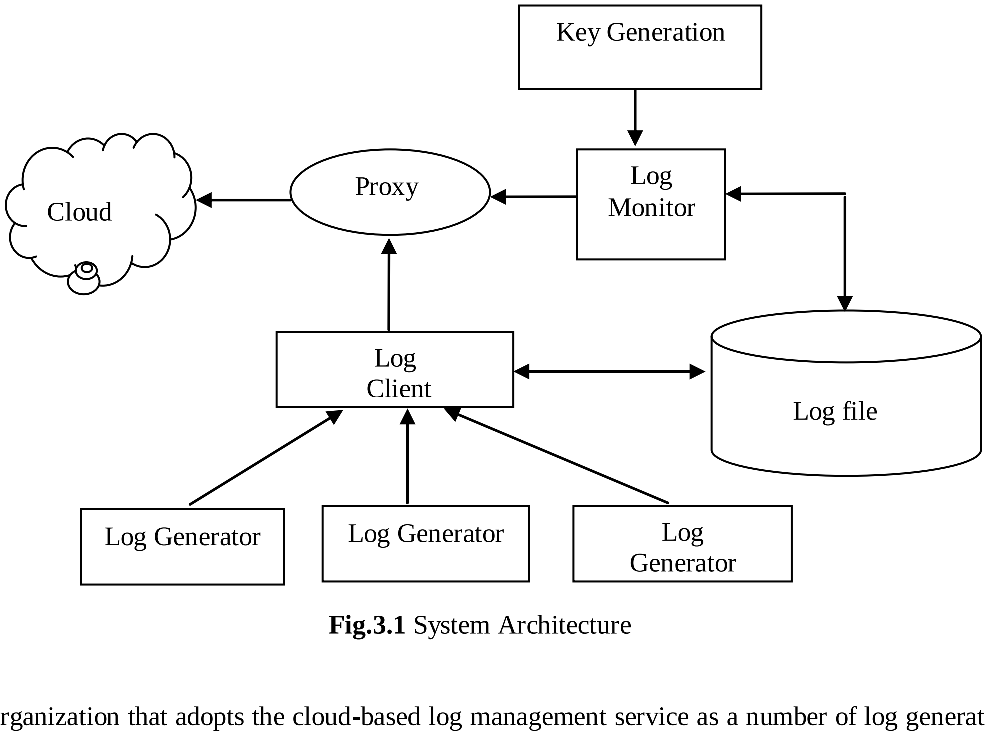

Fig.3.1 System Architecture

Each organization that adopts the cloud-based log management service as a number of log generators. These generators generate log data. Log data stored as a log file. The log files generated by these hosts are not stored except temporarily till such time as they are pushed to the logging client. The log file 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 file waiting to be transferred. The logging client is receives groups of log records generated by one or more log generators, and prepares the log file for pushed to the cloud for long term storage. The logging cloud provides long term storage and maintenance service to log data received from different logging clients belonging to different organizations. Only those organizations that have subscribed to the logging cloud's services can upload data to the cloud.

The cloud, on request from an organization can also delete log data .Before the logging cloud will delete log data it needs a proof from the requester. The logging client generates such a proof. Log Monitor 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. If a logging client wants to send some data to the log monitor (or vice versa), the sender cannot expect the receiver to be online to receive the data. As a result the sender has to publish the data in some location and the receiver needs to retrieve the data from there when needed. The logging cloud facilitates this communication by receiving and servicing appropriate requests. The data upload and retrieved in encrypted and decrypted manner. Here use AES algorithm for the key generation to the encryption and decryption.

IV. CONCLUSION

In this project, a complete system to securely outsource log records to a cloud provider. It 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 technique and find out the challenges for a secure cloud based log management service. Also implement how to store secure log file in cloud and that file we can change read, write, delete, upload and download. AES algorithm that uses for log monitors and log generator. 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 log management to the cloud. In the future, there is a plan to design a comprehensive and more generic object-oriented approach to facilitate autonomous protection of traveling content using AES algorithm. Also going to add a variety of security policies.

REFERENCES

[1] K. Kent and M. Souppaya. (1992). "*Guide to Computer Security Log Management"*, NIS Special Publication 800-92 [Online].

[2] D. New and M. Rose, *"Reliable Delivery for Syslog", Request for Comment RFC 3195*, Internet Engineering Task Force, Network Working Group, Nov. 2001.

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

[4] 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.

[5] 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.

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

[7] D. Dolev and A. Yao*, "On the security of public key protocols,"* IEEE Trans. Inform. Theory, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[8] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "*Proactive secret sharing or: How to cope with perpetual leakage,"* in Proc. 15th Ann. Int. Cryptology Conf., Aug. 1995, pp. 339–352.

[9] D. L. Wells, J. A. Blakeley, and C. W. Thompson*, "Architecture of an open object-oriented database management system,"* IEEE Computer, vol. 25, no. 10, pp. 74–82, Oct. 1992.

[10] K. Nørvag, O. Sandst a, and K. Bratbergsengen*, "Concurrency control in distributed object oriented database systems,"* in Proc. 1st East-Eur. Symp. Adv. Databases Inform. Syst., Sep. 1997, pp. 32–32.

AUTHORS BIBLIOGRAPHY

Ms.Mary Michael – She is Currently pursuing her M.E in Computer Science and Engineering at Roever Engineering College, Perambalur. She received the B. Tech. Degree in Information Technology from Karapaga Vinayaga College of Engineering and Technology, Madhuranthagam, chennai, in 2011.Her areas of interest are cloud computing and network security.

Mrs.J.Mary Metilda – She completed her M.E (CSE) from Coimbatore Institute of Engg & Info.Tech(CIET) College, Coimbatore .She is presently working as Assistant Professor in Dept.of Computer Science and Engineering in Roever Engineering College, Perambalur, India. His areas of interest are networking, network Security and cloud computing.