Delegating Log management to the Cloud ensuring Secure Logging

[1]Jyothi K T, [2]Lissy Anto P, [3] Nisha Peter
[1]MSc Computer Science, [2]Associate Professor,[3]Adhoc Faculty
[1]Dept of Computer Science,
[1]St.Joseph's College Irinjalakuda, Thrissur, India

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 maintaining 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. Managing the Log Records is highly tedious and confidential in any organization. Even though Log Records contain Log Files, it should be protected from third party hackers. Since, Log Files contains privacy details and sensitive information. Delegating log management is cost saving measure.. In order to overcome from hackers we introduce a secure algorithm known as Advance Encryption Standard (AES). It provide secret key to both Client and Data Owners. It provides high Bandwidth & low level Inactivity. It's the cheapest method where an attacker cannot read or modify/destroy the data's. We can implement AES Algorithm for various Security Issues.**

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, 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. EXISTING SYSTEM

Data handling in the cloud goes through a complex and dynamic hierarchical service chain. This does not exist in conventional environments. Ordinary web framework Uses web services for request and responses. The syslog protocol used UDP to transfer log information to the log server[3]. Thus, there is no reliable delivery of log messages more over syslog doesn't protect Log Records during transit or at the end -points.

Limitations

- No security for user's data. No authentication or security provided.
- Cost of implementation is high.

- Not suitable for small and medium level storage users

III. PROPOSED SYSTEM

We provide a solution for Storing and Maintaining Log Records in a Server Operating In a cloud based environment. We use Cryptographic Protocols for Integrity and Confidentiality Issues[2].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.

Advantages

- ✓ It is suitable for large and limited number of storages.
- ✓ Protects from birthday attacks, JVM attacks and meet-in-the-middle attack.

IV. MODULES

Modules Description

There are four different types of modules in this paper, that are listed in the following,

- Log Generators
- Logging Client or Logging Relay
- Logging Cloud
- Log Monitor

Log Generators

These are the computing devices that generate log data. Each organization hat adopts the cloud-based log management service has a number of log generators. Each of these generators is up to 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.

Logging Client or Logging Relay

The logging client is a collector that receives groups of log records generated by one or more log generators. The log data is transferred from the generators to the client in batches, either on a schedule or amount of log data waiting to be transferred. The logging client or relay can be implemented as a group of collaborating hosts.

Logging Cloud

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**.**

Log Monitor

These are hosts that are used to monitor and review log data. They can also ask the log cloud to delete log data permanently, or rotate logs[1]. 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.

V. AES (Advanced Encryption Standard) ALGORITHM

Advance Encryption Standard is a symmetric 128-bit block data encryption technique. It works at multiple networks layers. It has fixed block size of 128-bits and a key size of 128,192 or 256-bits. The algorithm was required to be royalty-free for use worldwide and offer security of a sufficient level to protect data for the next 20 to 30 years[5].It was to be easy to implement in hardware and software, as well as in restricted environments (for example, in a smart card) and offer good defences against various attack techniques.

ALGORITHM EXPLANATION:

AES is based on a design principle known as a Substitution permutation network. Unlike its predecessor, DES, AES does not use a Feistel network.AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits, whereas Rijndael can be specified with block and key sizes in any multiple of 32 bits, with a minimum of 128 bits. The block size has a maximum of 256 bits, but the key size has no theoretical maximum.AES operates on a 4×4 column-major order matrix of bytes, termed the *state* (versions of Rijndael with a larger block size have additional columns in the state)[5]. Most AES calculations are done in a special finite field.

APPLICATIONS OF AES

The applications of Advance Encryption Standard are:

- Digital Cinema Projection System.
- Low Power Implementation for Bluetooth.
- Digital Cryptography

VI. CONCLUSION

In our paper, we discussed the various security Issues by means of Advance Encryption Standard Algorithm. We choose AES Algorithm since the block size is fixed to 128-bit.It is the cheapest method and the attackers can hack data in three steps-- First Attackers can interrupt messages over network. Second Attackers can replicate and replay messages. Third Attackers can justify the participant of the network. We have also provided solution for storing and maintaining log records in cloud-based environment. We have used cryptographic protocols for confidentiality issues. We provide security all the four stages of present

system modules. 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 travelling content using AES algorithm, also going to add a variety of security policies.

VII. FUTURE WORK

In future work, the present system module Log Client implementation is refined to replace the current log process. We have planned to investigate practical Homomorphism Encryption Schemes that allows encryption of Log Records with privacy and confidentiality.

VIII. 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.