Vulnerable Taxonomy like a Tune-up Delegate Account Board to the Cloud

Abirami.V.R

Department of Computer Science and Engineering,
Annai Mathammal Sheela Engineering College, Namakkal,
(Anna University, Chennai.)
Tamilnadu, India.
abiramicse02@gmail.com

Abstract—**Securely maintaining log records are over extended periods of time is very important to the proper functioning of any organization. Integrity of the log files during the logging process need to be ensured all times. In count, as log files often contain sensitive information, confidentiality and privacy of log records are equally important. Deploying a secure logging infrastructure involves substantial capital expenses that many organizations may find over-come completely. Delegating log management to the cloud appears to be a feasible cost saving measure. Here we identify the challenges for a secure cloud-based log management service and propose a frame work for doing the same.**

Keywords—**Cloud computing, logging, privacy, security.**

I. Introduction

A log is a record of events occurring within an organisation’s system or network. Logging is important because log data can be used troubleshoot problems fine tune system performance, identify policy violations, investigate malicious activity even log record user activities. Log records can play a significant role in digital forensic analysis of systems. Regulations such as HIPAA, Payment Card Industry Data Security Standard, or Sarbanes-Oxley often require forensically sound preservation of information. To fulfill with these regulations, evidence produced in a court of law, including log records, must be complete before they can be used.Since log files contain record of most important target for malicious attackers. An attacker can breaking into the system, typically would try not to leave traces of his or her activities behind. Here we develop a Proof-of-concept prototype to demonstrate the feasibility of our approach and discuss some early experiences with it.

II. Properties of Secure Logging as a Service

1) *Correctness:* Log data is useful only if it reflects true history of the system at the time of log generation.
2) *Tamper Resistance:* A secure log must be tamper resistant in such a way that no one other than the creator of the log can introduce valid entries. No one can prevent an attacker who has compromised the logging system from altering what that system will put in future log entries
3) *Verifiability:* It must be possible to check that all entries in the log represent and have not been altered. If some of the entries are deleted, the ability to individually verify their main in gentries makes it possible to recover some useful information from the damaged log.
4) *Confidentiality:* Log records should not be casually brows able or searchable to gather complex information.
5) *Privacy:* Log records should not be usually traceable or linkable to their sources during transit and in storage

III. System Architecture

The overall architecture of the cloud based secure log management system is shown in Fig.3.1

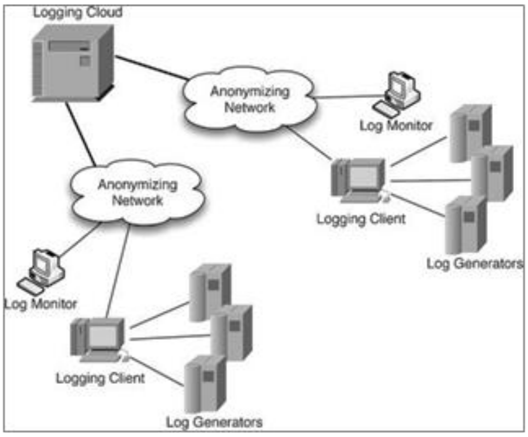


Fig.3.1 System Architecture

There are four functional components in this architecture system

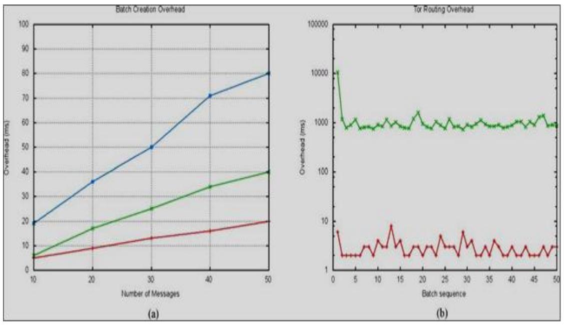
1) *Log Generators:* Log Generators can generate the log data. The log files generated from the hosts are not stored locally except temporarily till such time as they are pushed to logging client
2) *Logging Client:* Logging Client can collect the log records generated by log generators and prepare a log data push to a cloud for long term storage
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.
4) *Log Monitor:* Log Monitors are hosts that are used to monitor and review log data.

IV. Existing System

A number of approaches can be used in the logging information in computing systems. Most of them can be based on the Sys-log which is the de facto standard of network wide logging protocol.

1) *Syslog-ng* can replace by support for IPv6, TCP.Syslog protocol uses UDP to transfer log information to log server.
2) *Syslog-sign* can originate by using two additional messages —“signature blocks” and “certificate blocks.”
3) *Syslog-pseudo* cans logging the architecture to pseudonymize log files. While the protocol analyse the log record each log record does not protect log records from attacks that try to correlate a number of anonym zed records.
4) *Forward-integrity* is a secret key that becomes the starting point of hash-chain. This key can generate by a cryptographically strong one-way function in which the key is changed for every log record.

The record can be explained in the Graph 4.1 as result of integrity



Graph 4.1 (a) Batch creation overhead (b) Tor routing overhead

Disadvantages

1) It does not ensure correctness of logs.
2) High Cost required to maintain the logs

V. Proposed System

This system proposes a comprehensive solution for storing and maintaining log records in a server operating in a cloud-based environment. This system has security and integrity issues not only just during the log generation phase. This successfully prevents the cloud provider or any other observer from correlating requests for log data with the log generator. Finally, we develop a proof-of-concept prototype to demonstrate the feasibility of our approach.

Symbols Used in Log Preparation Protocol during the log data preparation using log generator can be in Table 5.1

Table 5.1

Symbol Used	**Interpretation**
$E_p[M]$	Encryption of message M with some secret key p
$M_1 \| M_2$	Concatenation of messages M_1 and M_2
$H^i[M]$	Crypto graphical of message M with key k and
K	Hard to invert hash function H with the hash performed I number of times
$H^n[M]$	Message M hashed n number of times
TS	A global time stamp

This can be used for protocol upload and log retrieval Fig 5.2

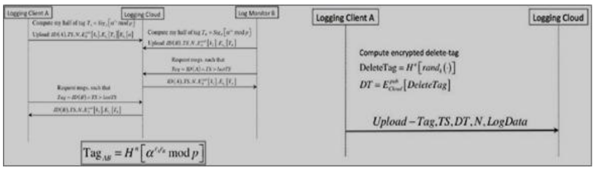

Logging Client A
Logging Cloud
Log Monitor B
Logging Client A
Logging Cloud
Compute encrypted delete-tag
DeleteTag = $H^{n}[rand_{k}(\cdot)]$
$DT = E^{pub}_{Cloud}[DeleteTag]$
$Upload-Tag, TS, DT, N, LogData$
$Tag_{AB} = H^{n}[\alpha^{r_A r_B} \bmod p]$

Fig 5.2 upload-tag generation and protocol log retrieval

The adversaries in our protocol are then the honest but curious cloud provider and any attacker that can attack the network and the cloud provider. This assumes that all encryption is resistant to cryptanalysis and that they do not leak information.

Fig 5.3 protocol for log upload and log delete can be used in proposed system to avoid attacker in the system

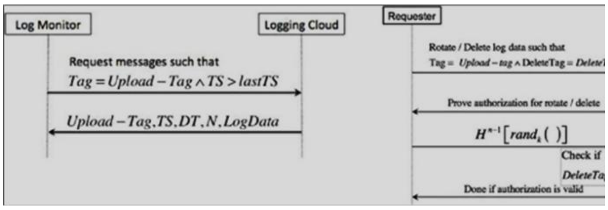

Log Monitor
Logging Cloud
Requester
Request messages such that
$Tag = Upload-Tag \wedge TS > lastTS$
$Upload-Tag, TS, DT, N, LogData$
Rotate / Delete log data such that
Tag = $Upload-tag \wedge$ DeleteTag = Delete
Prove authorization for rotate / delete
$H^{n-1}[rand_{k}(\ )]$
Check if
DeleteTa
Done if authorization is valid

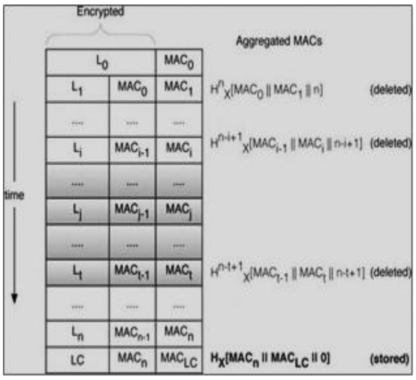
Fig 5.3 protocol for log upload and log delete

Advantages

1) More secure and confidential
2) Low cost for store and manage log records.
3) Very fast to transmit the log file.

VI. Conclusion and Future Work

In the future, System plan to refine the log client implementation so that it is tightly integrated with the OS to replace current log process. This system plan to investigate practical homomorphism encryption schemes that will allow encryption of log records in such a way that the logging cloud can execute some queries on the encrypted logs without breaking confidentiality or privacy. This will greatly reduce the communication overhead between a log monitor and the logging cloud needed to answer queries on logs. The proposed system can aggregate message authentication in cloud in Fig 6.1



L_0		MAC_0		
L_1	MAC_0	MAC_1	$H^{n}_{X}[MAC_0 \parallel MAC_1 \parallel n]$	(deleted)
.....	.....	.....		
L_i	MAC_{i-1}	MAC_i	$H^{n-i+1}_{X}[MAC_{i-1} \parallel MAC_i \parallel n-i+1]$	(deleted)
.....	.....	.....		
L_j	MAC_{j-1}	MAC_j		
.....	.....	.....		
L_t	MAC_{t-1}	MAC_t	$H^{n-t+1}_{X}[MAC_{t-1} \parallel MAC_t \parallel n-t+1]$	(deleted)
.....	.....	.....		
L_n	MAC_{n-1}	MAC_n		
LC	MAC_n	MAC_{LC}	$H_X[MAC_n \parallel MAC_{LC} \parallel 0]$	(stored)

Fig 6.1 Log batch uploaded on cloud with aggregated message authentication code.

References

[1] Arellano.A (2004) Vulnerability& Taxonomy reference of the book in online.

[2] Blakley.G.R, "Safeguarding cryptographic keys," in Proc. Nat. Comput.Conf., Jun. 1979, p. 313.

[3] Department U.S of Health and Human Services. (2011, Sep).HIPAA—General Information [Online].Available: https://www.cms.gov/Hipaageninfo

[4] Flegel.U, "Pseudonym zing Unix log file," in Proc. Int. Conf. Infrastruture*Security*, LNCS 2437. Oct. 2002, pp. 162–179.

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

[6] John Wiley & Sons, Hoboken, NJ "Auditing Cloud Computing: A Security and Privacy Guide"Sep (2009)

[7] Kent and Souppaya.M (1992). Guide to Computer Security Log Management, NIST Special Publication800-92.

[8] Lonvick.C., *The BSD Syslog Protocol*, Request for Comment RFC 3164, Internet Engineering Task Force, Network Working Group, Aug. 2001.

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

[10] Dr.Mark Watts Bristow's (2009),"Data Protection to ensure compliance "of publications online.

[11] Mariappan Raja ram received the B.E. degree in computer science from Bharathiar University, Coimbatore, India, and the M.S. degree from Colorado State University, Fort Collins, in 2009, majoring in computer security.

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

[13] Ranguelov.B, Nordvik.J."Taxonomy And "Vulnerability": Of the GEO Environmental Risk Process" (2004)

[14] Schneider's, BandKelsey.J, "Security audit logs to support computer forensics," ACM Trans. Inform. Syst. Security, vol. 2, no. 2, pp. 159–176, May 1999.