SECURE CLOUD-BASED LOG MANAGEMENT SERVICE: A LITERATURE SURVEY

Asha Vijayan T[1], Divya K V[2]

[1]*M.Tech Student, Vidya Academy of Science & Technology, University of Calicut, Kerala, (India)*

[2]*Assistant Professor, Vidya Academy of Science & Technology, University of Calicut, Kerala, (India)*

ABSTRACT

Since the beginning, life has relied upon the transmission of messages. As the operating systems, processes and applications grew ever more complex, systems were devised to categorize and log these diverse messages and allow the operations staff to more quickly differentiates the notifications of problems from simple status messages. A log is a record of performance, events, or day-to-day activities taking place in an organization's systems and networks. 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. In this paper we discuss some logging protocols and we will subsequently analyze each method against the five desirable properties such as correctness, tamper resistance, verifiability, confidentiality, privacy, that a secure cloud based log management service should possess.

Keywords: Log Management, Reliable Syslog, Syslog, Syslog Pseudo, Syslog Sign

I INTRODUCTION

Since the beginning, life has relied upon the transmission of messages. For the self-aware organic unit, these messages can relay many different things. The messages may signal danger, the presence of food or the other necessities of life, and many other things. In many cases, these messages are informative to other units and require no acknowledgement. As people interacted and created processes, this same principle was applied to societal communications. As an example, severe weather warnings may be delivered through any number of channels - a siren blowing, warnings delivered over television and radio stations, and even through the use of flags on ships. The expectation is that people hearing or seeing these warnings would realize their significance and take appropriate action. In most cases, no responding acknowledgement of receipt of the warning is required or even desired. Along these same lines, operating systems, processes and applications were written to send messages of their own status, or messages to indicate that certain events had occurred. These event messages generally had local significance to the machine operators. As the operating systems, processes and applications grew ever more complex, systems were devised to categorize and log these diverse messages and allow the operations staff to more quickly differentiates the notifications of problems from simple status messages.

A log 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. 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. Frequently, log information can be helpful to an attacker in gaining unauthorized access to system. 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.

Based on the above observations, it is important that logging be provided in a secure manner and that the log records are adequately protected for a long time. 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. The emerging paradigm of cloud computing promises a low cost opportunity for organizations to store and manage log records in a proper manner. 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.

According to these requirements, the desirable properties that we seek from a secure log management service based on the cloud computing paradigm [2] are

1) Correctness
2) Tamper Resistance
3) Verifiability
4) Confidentiality
5) Privacy

In this paper we discuss some logging protocols and we will subsequently analyze each method against these properties.

II METHODS USED FOR LOGGING

2.1. The Syslog Protocol

In many cases, a message is informative to other units and requires no acknowledgement. The syslog [3] process is a system that has been widely accepted in many operating systems for message logging. Flexibility was designed into this process so the operations staffs have the ability to configure the destination of messages sent from the processes running on the device. In one dimension, the events that were received by the syslog process could be logged to different files and also displayed on the console of the device. In another dimension, the syslog process could be configured to forward the messages across a network to the syslog process on another machine. The syslog process had to be built network-aware for some modicum of scalability since it was known that the operators of multiple systems would not have the time to access each system to review the messages

logged there. The syslog process running on the remote devices could therefore be configured to either add the message to a file, or to subsequently forward it to another machine.

In its most simplistic terms, the syslog protocol provides a transport to allow a machine to send event notification messages across IP networks to event message collectors - also known as syslog servers. Since each process, application and operating system was written somewhat independently, there is little uniformity to the content of syslog messages. For this reason, no assumption is made upon the formatting or contents of the messages. The protocol is simply designed to transport these event messages. In all cases, there is one device that originates the message. The syslog process on that machine may send the message to a collector. No acknowledgement of the receipt is made.

One of the fundamental tenets of the syslog protocol and process is its simplicity. No stringent coordination is required between the transmitters and the receivers. Indeed, the transmission of syslog messages may be started on a device without a receiver being configured, or even actually physically present. Conversely, many devices will most likely be able to receive messages without explicit configuration or definitions.

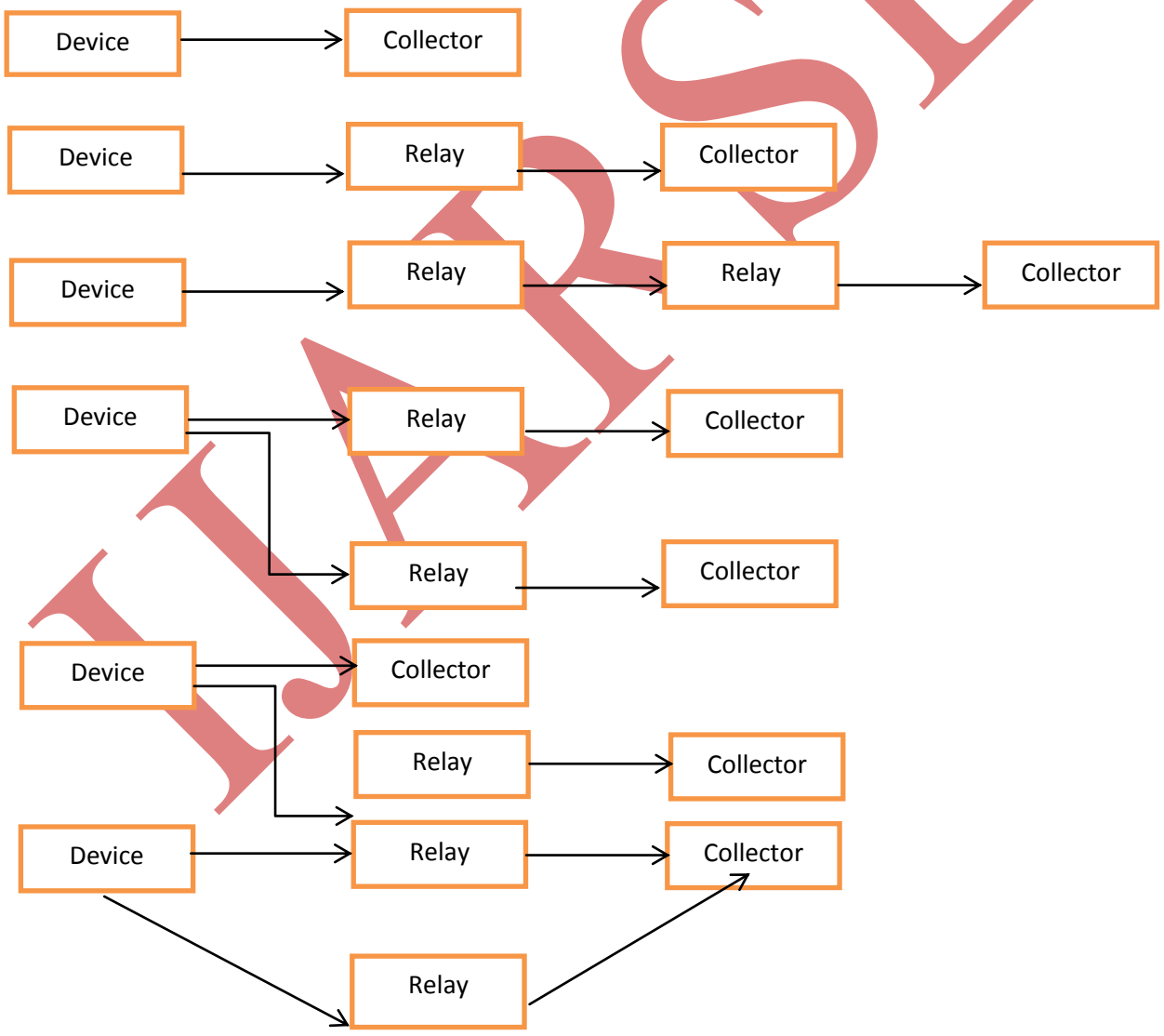

Figure1. Some Possible Syslog Architectures

This simplicity has greatly aided the acceptance and deployment of syslog. Syslog uses the user datagram protocol (UDP) as its underlying transport layer mechanism.

The architecture of the devices may be summarized as follows: Senders send messages to relays or collectors with no knowledge of whether it is a collector or relay. Senders may be configured to send the same message to multiple receivers. Relays may send all or some of the messages that they receive to a subsequent relay or collector. In the case where they do not forward all of their messages, they are acting as both a collector and a relay. In the following diagram, these devices will be designated as relays. Relays may also generate their own messages and send them on to subsequent relays or collectors. In that case it is acting as a device. These devices will also be designated as a relay in the Fig 1.

2.2. Reliable Delivery for syslog

How to realize the syslog protocol when reliable delivery is selected as a required service is the question here. To provide reliable delivery when realizing the syslog protocol, reliable syslog [4] mechanism defines two BEEP profiles. BEEP [5] is a generic application protocol framework for connection-oriented, asynchronous interactions. Within BEEP, features such as authentication, privacy, and reliability through retransmission are provided. There are two profiles defined:

1. The RAW profile is designed to provide a high-performance, low impact footprint, using essentially the same format as the existing UDP-based syslog service.
2. The COOKED profile is designed to provide a structured entry format, in which individual entries are acknowledged (either positively or negatively).

Both profiles run over BEEP. BEEP defines "transport mappings," specifying how BEEP messages are carried over the underlying transport technologies. The transport mechanism used is TCP. All transport mappings are required to support enough reliability and sequencing to allow all BEEP messages on a given channel to be delivered reliably and in order. Hence, both the RAW and COOKED profile provide reliable delivery of their messages.

BEEP is used to provide communication security but not object integrity. In other words, the messages "on the wire" can be protected, but a compromised device may undetectably generate incorrect messages, and relays and collectors can modify, insert, or delete messages undetectably. Other techniques must be used to assure that such compromises are detectable.

2.3. Signed Syslog Messages

Syslog-sign [6] is a mechanism that adds origin authentication, message integrity, replay resistance, message sequencing, and detection of missing messages to syslog. Essentially, this is accomplished by sending a special syslog message. The content of this syslog message is called a Signature Block. Each Signature Block contains, in effect, a detached signature on some number of previously sent messages. It is cryptographically signed and contains the hashes of previously sent syslog messages.

Additionally, a signer sends Certificate Blocks to provide key management information between the signer and the collector. The collector may verify that the hash of each received message matches the signed hash contained in the corresponding Signature Block. A collector may process these Signature Blocks as they arrive,

building an authenticated log file. Alternatively, it may store all the log messages in the order they were received. This allows a network operator to authenticate the log file at the time the logs are reviewed.

The process of signing works as long as the collector accepts the syslog messages, the Certificate Blocks and the Signature Blocks. Once that is done, the process is complete. After that, anyone can go back, find the key material, and validate the received messages using the information in the Signature Blocks.

2.4. Syslog-pseudo

For diverse reasons most internet services store information about their customers' requests on a long term basis. One of the reasons to do this is a service provider's need to be able to establish accountability for certain requests. Regardless of the required effort, IP addresses can often be linked to an actual person accessing a service. Since the stored request information contains personal data, we need to consider customer demands regarding privacy. In times when companies appreciate the value of personal information, customers will increasingly appreciate the protection of their valuable personal data. The vast majority of users strongly values being able to anonymously use internet services. This situation calls for privacy enhancing technologies that can be used today and that help service providers to comply with user expectancies as well as with legal regulations concerning personal data. Service providers that can credibly assure their customers the protection of their personal data may gain a competitive advantage over those that can't.

Syslog-pseudo [7] addresses the privacy problems. It 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 is that while the protocol anonymizes each log record individually it does not protect log records from attacks that try to correlate a number of anonymized records.

III DISCUSSIONS

Above we have discussed four different approaches for logging information in computer systems. The syslog protocol and process is good because of its simplicity. But this delivery mechanism does not strongly associate the message with the message sender. The receiver of that packet will not be able to ascertain that the message was indeed sent from the reported sender, or if the packet was sent from another device. One possible consequence of this behavior is that a misconfigured machine may send syslog messages to a collector representing itself as another machine. Also message forgery is possible. An attacker may transmit syslog messages to a collector. The syslog process and protocol do not ensure ordered delivery. Without any sequence indication or timestamp, messages may be recorded and replayed at a later time. As there is no mechanism within either the syslog process or the protocol to ensure delivery, and since the underlying transport is UDP, some messages may be lost. They may either be dropped through network congestion, or they may be maliciously intercepted and discarded. Besides being discarded, syslog messages may be damaged in transit, or an attacker may maliciously modify them. Neither the syslog protocol nor the syslog applications have mechanisms to provide confidentiality of the messages in transit or at the end points.

Reliable-syslog 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 Reliable syslog protocol allows device authentication and incorporates mechanisms 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.

Syslog-sign is an another enhancement to syslog that adds origin authentication, message integrity, replay resistance, efficient verification of logs, 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.

The main idea of Syslog-pseudo is that log records are first processed by a pseudonymizer before being archived. 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 of anonymized records. Moreover, privacy breaches that can occur from scenarios such as the user erroneously typing the use ride in a password field or identifying information available in fields that are not anonymized, are also not addressed with this approach Syslog-pseudo does not protect log records from confidentiality and integrity violations and other end-point attacks.

IV 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. As we have already seen, syslog protocol does not provide reliable delivery of messages. Reliable syslog and signed syslog does not prevent against confidentiality and privacy breaches during transit or at the end points. And syslog pseudo does not ensure correctness. Each of the above protocols is useful in different situations. But based on the above discussions we conclude that it is better to use a protocol that 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. Also the protocol should address integrity and confidentiality issues with storing, maintaining, and querying log records and privacy issues.

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] Indrajit Ray, Kirill Belyaev, Mikhail Strizhov, Dieudonne Mulamba, and Mariappan Rajaram, "*Secure Logging As a Service-Delegating Log Management to the Cloud*", IEEE SYSTEMS JOURNAL, VOL. 7, NO. 2, JUNE 2013

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

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

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

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

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