Secure Logging and Automatic Crash Notification System in Cloud Environment

Bommu Naga Naveen Reddy [#1], R Senthil Kumar [#2],Venu Kumar B [#3]

#1 School of Computing Sc. and Engg.VIT University, Vellore .
#2 School of Computing Sc. and Engg.VIT University, Vellore.
#3 School of Computing Sc. and Engg.VIT University,Vellore.

ABSTRACT

Maintaining the **log records** in a secured manner over a period 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 the secure logging infrastructure involves substantial capital expenses that many organizations may find overwhelming. Using the log management to the cloud appears to be a cost saving measure. We identified the challenges for a secure Cloud-based log management service and propose a system to securely transfer the log file to the cloud and we use the Log collector to collect the all the system log files and use the keys generator to add the unique signature to each log to ensure the security and finally we use the log analyzer send the error report and the cause of the error or warning to the administrator of the server .In the Present System to show how its works we implemented using the banking System application to generate the logs and using these log files it Would be easy to the Administrator to know which caused the error and the resolving the error can be easy to the administrators instead of searching for the possible cause of the Error or Warning .Using this Alert Notifications it would be time saving and Efficient.

Key words: Log files, keys generator, privacy, cloud-based log management, warning, error

Corresponding Author: Bommu Naga Naveen Reddy

INTRODUCTION

Log File is a file that records either the events which happen while an operating system or other software that system runs in the System .The **Log** is a record of events occurring within an organization's system or network. They record noteworthy Events such as user Activity, Program execution Status, System resource usage and data changes. Logs provide a valuable view of past and current states of almost any type of a complex system Log data can be used to troubleshoot the system and tune the systems performance and to identify policy violations and also useful to check for malicious activities and even record user activities. System and application logs are of nice value for administrators such as for observance, **fault** management, and forensics.

Log records which plays an important role in digital forensic analysis of the Systems. Because the log files which contain record of most system events which includes User activities and also the for the aplications running status in the system become an important target for malicious Attackers. An attacker, who breaks into a system, typically would try not to leave traces of his activities behind. So the Log file is the key for tracing the intruders or attackers and also useful to alert the server administrator about the possible error causing issue by using the log file data. In this System we use the

Limited functioning banking application which will handle the account creation of the user with the password and the withdraw amount and the deposit amount and the user specific operation which will generate the log files in the real time environment. Which will create the Log files when the User uses the application so that the log files are gathered by the log collector and the system use the Keyed-Hash Message Authentication Code (HMAC) algorithm which is more secure with the encryption and the decryption ,which will encrypt the files in batches and convert them to the and then they will be send to the server side and then the log analyser will decrypted at the server side and it will be used for the notification sender to send the notifications if any errors or warnings are generated in the log file.

EXISTING SYSTEM

1. The existing system the uses UDP to transfer log information to the log server. Thus, there is no reliable delivery of log messages.
2. Moreover, syslog does not protect log records during transit or at the end-points and traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted due to the users loss control of data under Cloud Computing.

PROPOSED SYSTEM:

1. In this system, we propose a System that will Securely Transfer the Log Files by using the HMAC algorithm. In this System we used an banking application which will generate the log files
2. By using these Log files, we use a Keywords in that Log Files and with that information we Provide the Alerts to the Administrator if any warning or errors are found.
3. In this System we also provide the possible Cause of the Error by using the Log files and Notify to Administrator there by the Administrator work can be easy without searching the reason for Cause and can work easily for fixing the system.

Example for the Log Files in windows:

Example 1:
Warning 22-Jan-14 10:38:30 AM Microsoft-Windows-CAPI2 102 None "Reached crypt32 threshold of 50 events and will suspend logging for 60 minutes."

Example 2:
Error 22-Jan-14 10:38:30 AM Microsoft-Windows-CAPI2 4107 None" Failed extract of third-party root list from auto update cab at: <http://www.download.windowsupdate.comauthrootstl.cab> with error: The data is invalid.
Example of Log Files created in banking application used in our system

Example 3:
Feb 25, 2014 11:15:33 AM GuiAccTest actionPerformed
SEVERE: Error message.Invalid Account Number
Feb 25, 2014 11:15:46 AM GuiAccTest actionPerformed
WARNING: Warning message. not a valid size
Feb 25, 2014 11:16:00 AM GuiAccTest actionPerformed
INFO: account created :Accno1111111

we will notify to the **ADMIN** of the system that error is being created with the reason like in this example it was Memory overload .If the server itself will crash we will send the Crash report and the last log file received .By using this system we can achieve the security and we can notify the admin for the system error and resolved with that error.

SYSTEM ARCHITECTURE:

SYSTEM MODULES DESCRIPTION

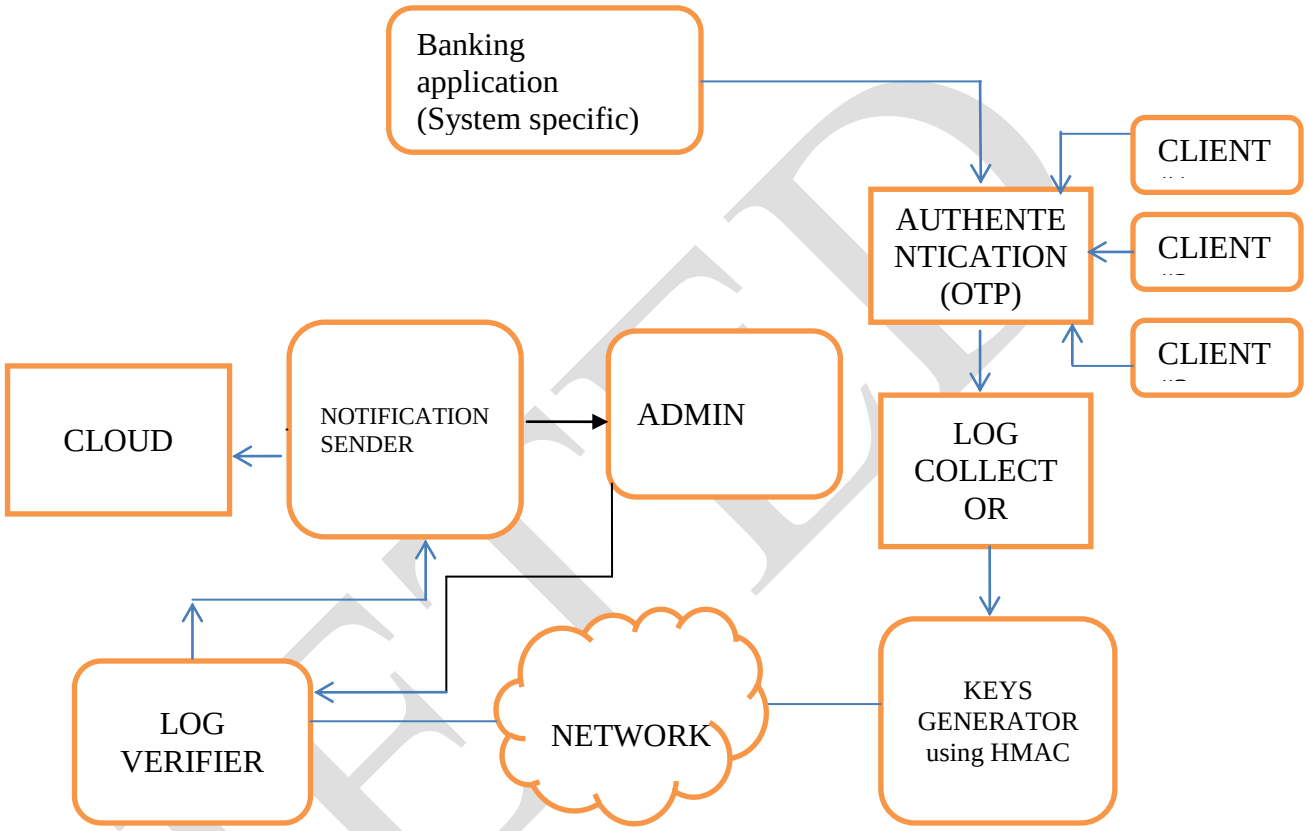
BANKING APPLICATION
The main source of the log files is the banking application which is capable of generating the log files for different errors and warnings and the application is for the limited functionality and here we are showing that the system is capable to handle the log files efficiently.

AUTHENTENTICATION (OTP-one time password)
This component is used to generate the one time password for the clients logged in our system and we send the OTP to the users mail for the secure login of the system.

LOG COLLECTOR
The **Log Collector** is the client side component and the log collector duty is collect all the log files from different clients systems and push these log files to the Keys generator.

KEYS GENERATOR
The **Keys Generator** is another component in our System that takes the Log Files from the Log Collector and then add the unique keys generated by using the **HMAC algorithm** and then send to the Log analyzer Which is in the server side.

Issue 4, Vol.3 (May 2014)
ISSN 2249-6149



HMAC ALGORITHM

This algorithm which is used for the applications requiring message authentication. Message authentication which is achieved via the construction of a message authentication code (MAC). These MACs based on cryptographic hash functions are known as HMACs. The main purpose of these a MAC is to authenticate both the source of a message and its integrity without the use of any additional mechanisms. HMACs does have two functionally distinct parameters, a message input and a secret key known only to the message originator and intended receiver(s).Additional applications of keyed hash functions include their use in challenge-response identification protocols for computing responses, which are a function of both a secret key and a challenge message.

HMAC ALGORITHM :

```
function hmac (key, message)
   if (length(key) > blocksize) then
      key = hash(key) // keys longer than blocksize are shortened
   end if
   if (length(key) < blocksize) then
      key = key ‖ [0x00 * (blocksize - length(key))] // keys shorter than blocksize are zero-padded (where ‖ is concatenation)
   end if

   o_key_pad = [0x5c * blocksize] ⊕ key // Where blocksize is that of the underlying hash function
   i_key_pad = [0x36 * blocksize] ⊕ key // Where ⊕ is exclusive or (XOR)

   return hash(o_key_pad ‖ hash(i_key_pad ‖ message)) // Where ‖ is concatenation
end function
```

An **HMAC function** which is used by the message sender to produce a value (the MAC) and that is formed by condensing the secret key and the message input. The system that we are using the MD5 to operate on 512-bit blocks. The MAC is typically sent to the message receiver along with the message. The receiver computes the MAC on the received message using the same key and HMAC function as was used by the sender, and compares the result computed with the received MAC. If the two values match, the message has been correctly received, and the receiver is assured that the sender is a member of the community of users that share the key.

Message authentication code (mac) :The cryptographic checksum which results from passing data through a message authentication algorithm and its is a standard, the message authentication algorithm is called HMAC.

Cryptographic key (key): The parameter used in conjunction with a cryptographic algorithm which determines the specific operation of that algorithm and it is standard, the cryptographic key is used by the HMAC algorithm to produce a MAC on the data.

Keyed hash-based message authentication code (hmac) : A message authentication code uses a cryptographic key in conjunction with a hash function.

Secret key: A cryptographic key which is uniquely associated with one or more entities. The secret which is context does not imply a classification level; rather the term which implies the need to protect the key from disclosure or substitution.

LOG VERIFIER The main purpose of Verifier is to receive the encrypted Log files from Keys Generator decrypted using MD-5 256 bit cryptographic hash function which will used to simultaneously verify both the data integrity and the authentication of a log files.

NOTIFICATION SENDER The purpose of Notification sender is to detect the Warning messages and the Error messages by using Predefined Key Words in Log Files by String manipulation using the function "string.StartsWith()" in java and notify the admin of the server and if the server itself will crash it will send the last received log files to the admin.

ADMIN Admin is the central part in which he can use these notifications and send to the concerned technical employee of the companies who can take required steps to solve if any error happens knowing the reason for the system is easy to solve the error in faster way instead of searching for the error causing reason this gives the faster and effective solution for the system crash .

CONCLUSION

In our system we tried to implement the log file notification system using the banking application and this application which is having the limited functionality and produce the log files and we tried to show that how these log files are handled securely by using the HMAC algorithm and later send to the notification senders which will detect the predefined words in by using the string search and send these notifications to the Clients and then send to the Admin and will admin will in turn send to the Cloud for Permanent Storage. we are using the log files in between the transmission to the cloud and using for the detecting the error/warnings and sending the user what made the cause of the error and such that the user can detect the cause of the error instead of rechecking the complete system in case of the system crash .but here we are implementing with the smaller application with limited functionality and for higher log files such as in Giga Bytes we need to have high end systems that can process the log files .

REFERENCE

[1]. J. Kelsey, J. Callas, and A. Clemm, Signed Syslog Messages, Request for Comment RFC 5848, Internet Engineering Task Force, Network Working Group, May 2010.
[2]. R. C.Merkle, "Protocols for public key cryptosystems," in Proc. of IEEE Symposium on Security and Privacy, Los Alamitos, CA, USA, 1980
[3]. Ma, D.: Practical forward secure sequential aggregate signatures. In: ACM Symposium on Information,Computer and Communications Security (ASIACCS'08).(March 2008).
[4]. Rafael Accorsi "A Secure Log Architecture to Support to Remote Auditing "; 57(2013): pp 1578-1591
[5]. A New Approach to Secure Logging by DI MA and GENE TSUDIK, University of California, Irvine.
[6]. M. Bellare and B. S. Yee, "Forward integrity for secure audit logs," Dept. Comput. Sci., Univ. California, San Diego, Tech. Rep., Nov. 1997.
[7]. BalaBit IT Security (2011, Sep.). Syslog-ng—Multiplatform Syslog Server and Logging Daemon [Online]. Available: http://www.balabit. com/network-security/syslog-ng
[8]. S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained access control in cloud computing," in Proc. of IEEE INFOCOM'10, San Diego, CA, USA, March 2010.

[9].U.S. Department of Health and Human Services. (2011, Sep.). HIPAA General Information [Online].
[10].PCI Security Standards Council. (2006, Sep.) Payment Card Industry (PCI) Data Security Standard—Security
[11].Sarbanes-Oxley Act 2002. (2002, Sep.). A Guide to the Sarbanes-Oxley Act [Online].
[12] L.H.Seawright et al. VM/370 A Study of Multiplicity and Usefulness,
IBM System Journal, 1979
[13] R.J. Creasy. The Origin of the VM/370 Time-sharing System. IBM
Journal of Research and Development, 1981
[14] P. H. Gum. System/370 Extended Architecture:Facilities for Virtual Machines, IBM Journal of Research and Development, 1983
[15] VMware , http://www.vmware.com
[16] Microsoft Hyper-V,http://www.microsoft.com/windowsserver2008/en/us/hyperv.aspx
[17] Xen hypervisor, http://xen.org/
[18] MENDEL ROSENBLUM and JOHN K. OUSTERHOUT, The Design and Implementation of a Log-Structured File System, ACM Transactions on Computer Systems, Vol 10, No. 1,February 1992,Pages 26-52.
[19] Robert Hagmann,Reimplementing the cedar file system using logging and group commit. In Proceedings of the 11th Symposium on Operating Systems Principles (Austin, Tex,, Nov. 1987), pp. 155-162.
[20] Douglas S. Santry1, Michael J. Feeley, Norman C. Hutchinson, Alistair C. Veitchy, Ross W,17th ACM Symposium on Operating Systems Principles (SOSP '99), Published as Operating Systems Review, 34(5):110–123,Dec. 1999
[21] Rootkit, part 1 of 3 the growing threat, White Paper, April 2006, www.McAfee.com .
[22] Zachary N.J. Peterson, Randal Burns, Giuseppe Ateniese, and Stephen Bono. Design and implementation of verifiable audit trails for a versioning file system. In Proceedings of the 5th conference on USENIX Conference on File and Storage Technologies (FAST'07), Feb 2007.
[23] Bruce Schneier and John Kelsey. Secure audit logs to support computer forensics. ACM Transactions on Information and Systems Security, 1999, 2(2): 159~176
[24] Tal Garfinkel, Ben Pfaff, Jim Chow, Mendel Rosenblum and Dan Boneh, Terra: A Virtual Machine-Based Platform for Trusted Computing, In the 19th Symposium on Operating System Principles (SOSP 2003).