International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 3, Issue 3, September 2014)

Secured Authentication Service Using Cloud Architecture

Anagha Dinkar Dhongade[1], Prof. Niranjan L. Bhale[2]

[1]*Department of Computer Engineering,* [2]*Department of Information Technology, Matoshri College of Engineering and Research Center, Eklahare, Nashik, University of Pune., India*

Abstract **- Now a days, maintaining the security of log records is one of the most important task of any organization. Log records often contain sensitive information which should be protected for proper functioning of an organization. But development of such system for protecting log records is an overhead for an organization as well as it requires additional cost. Cloud computing can be used for maintaining security of log records with very less expenses. In this paper, we elaborate the techniques for securely creating, storing and retrieving the log files using cloud-based framework.**

Keywords- **Cloud computing, Logging, Privacy, Security, Confidentiality.**

I. INTRODUCTION

Log files contain records of all the activities performed within an organization. The log files are used to troubleshoot the problems, to identify users violating the policies or performing malicious activities and to store each and every activity of user. As log file contains every single activity of the users on the network , it is the most important target for malicious attacks. The reason behind this is that attacker wants to leave no traces of the activities performed by him after the attack on the system. So the first target of attack is normally log files of the organization.

Not only this, but log files also contain information about confidential transactions over network such data should be protected. Also log file information can be used by the attacker for getting unauthorized access to the system.

From this scenario it is clear that security of log files is the most important task of an organization.

But development of such a framework is an expensive job and also it is time consuming. Delegating log management to cloud computing reduces the overhead of an organization by saving time of log management as well as cost of developing the framework.

A. Desirable properties

The desirable properties of secure logging as a service using cloud are as follows:

1) *Correctness:* The data in the stored log files should be exactly same as the one that was generated by the creator.
2) *Tamper Resistance:* No one other than creator of the log can introduce valid log entries. Also any manipulations should be detected immediately.
3) *Verifiability:* There should be a technique present to check that all the entries in the log are present and have not been altered.
4) *Confidentiality:* Log files should not be easily available to anyone for gathering sensitive data.
5) *Privacy:* Log file contents should not traceable during the transit or the storage.

B. Major Functional Components

The major components required to develop the cloud-based framework for log security they are as follows:

1) *Log Generator:* These are the computing devices in the organization generating log data by performing various activities.
2) *Logging Client:* The logging client acts as a collector which collects log data of several log generators together.
3) *Logging Cloud:* The long term storage and maintenance of log data are the services provided by logging cloud to several organizations.
4) *Log Monitor:* Log monitor can generate multiple queries to retrieve data from the cloud.

C. Threat Model

This subsection elaborates the different types of attacks scenarios that should be protected against are as follows:

1) Integrity of Log Data during Transit:

The communication medium can be accessed by the attacker and the attacker can modify the data during the transmission.

2) Authenticity of Logging Client:

The attacker gets unauthorized access to the logging client and begins sending log data to logging cloud.

3) Privacy of Logging Cloud:

The attacker can continuously trying to link or trace the data send and receive by logging cloud so that he can have access to the data stored in logging cloud.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 3, Issue 3, September 2014)

II. Literature Survey

Various approaches have been developed to secure the log data of the organization. In this section we have elaborated existing secure logging protocols.

In Aug 2001,C.Lonvick proposed a technique syslog[2].This technique is used as de facto standard for network logging protocol.

Pros:

1. The protocol uses UDP for transmission of log files to log server.

Cons:

1. Due to use of UDP there is no reliable delivery of messages.

D. New & M. Rose, Nov 2001derived a technique Relaible-syslog[3].

Pros:

1. This technique uses TCP protocol for reliable delivery of messages.

Cons:

1. This technique cannot prevent against confidentiality.
2. Privacy breaches may occur during transmission or at end-points.

Syslog-pseudo[8] is the technique which provides pseudonymize log data proposed by U. Flegel.

Pros:

1. Before the transmission of the log file it is processed by a pseudonymizer.
2. Specific fileds are substituted with some values carefully so that log data cannot be understood by the attacker.

Cons:

1. The protocol provides no guarantee of correctness of log record contents.

In 2010, J. Kelsey & J. Calls proposed a technique Syslog-sign[6].It is enhancement of syslog.

Pros:

1. This technique adds signature block and certificate block with each sequence of log data.

Cons:

1. No privacy is provided during transmission of data.

Syslog-ng [5] is a technique proposed by Balabit IT Security in 2010.

Pros:

1. It provides backward compatibility with syslog.

Cons:

1. There is no protection against data modification when data is at the end points. In 2013, Indrajiit. Ray proposed a technique “Secure Logging as Service”[1]. The advantage provided this method is that log management is delegated by cloud server.

III. Implementation Details

This section elaborates implementation details of cloud-based architecture for securing log data. This first subsection presents a software and hardware requirements. The second subsection elaborates Algorithms required by our technique. And the final subsection describes the mathematical modeling of our architecture.

A. Platform

This section elaborates the hardware and software requirements of all the components like Logging Cloud, Log Generator, Logging Client, Log monitor.

The Hardware Requirements of our system are as follows:

1) At least 4GB RAM for all the components
2) At least 160 GB of Hard Disk for logging cloud and 80 GB of Hard Disk for remaining components.
3) Intel Core i3 processor or above for all the components.

The Software Requirements are as follows:

1) Windows XP or above for all the components
2) Programming Languages like Java, J2EE, JavaScript, Tomcat
3) AWS (Amazon Web Services)Cloud
4) AWS SDK

B. Algorithms

The basic aim to design cloud based log security architecture is to provide a low cost and efficient solution for securing log data. Here the task of log security is done by cloud service provider of an organization by reducing the overhead of an organization. The secure logging architecture under four different stages using which the complete cloud security is achieved.

Stage 1 Log Generation

The log generators are the computing devices that are used by users of organization.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 3, Issue 3, September 2014)

Log generators are responsible for log generation and storing these log entries in the log file. Every network activity of the user is recorded in log file. In this stage, along with log file generation the log file is monitored and notifications are sent when new lines are added to log file. Along with this task, one more important task performed is to establish the HTTP connection between log generator and logging client. Using this connection log data sent collectively to logging client after fixed time interval (for example after every 5 seconds).

Stage 2 Log Data Collection

The logging client is responsible for log data collection. In this phase, the log client receives the log data from one or more log generators. This collectively received log data is encrypted using triple DES. Also, tag is generated for each line which will be helpful while searching or validating log data. Further using HTTP connection this encrypted log data and the generated tags are sent to logging client after fixed time interval (for example after every 30 minutes)

Stage 3 Log Data Storage

The logging cloud is performs the task of log data storage. There logging cloud stores the encrypted log data received from various logging clients of various organizations. Along with secure storage facility, the logging cloud is also responsible for execution of queries sent by log monitor. The tags generated and appended with each line are used for execution of these queries.

Stage 4 Query Processing

The task of query processing is done by log monitor. Whenever the administrator of an organization wants to access log data stored at logging cloud, he gets access to it through log monitor. The log monitor sends the queries to logging cloud such as search operation, validation of a log data chunk. The encryption scheme is designed in such a way that the logging cloud should execute all these queries on encrypted data only and sents encrypted response to log monitor.

Following Fig.1 gives framework of proposed system.

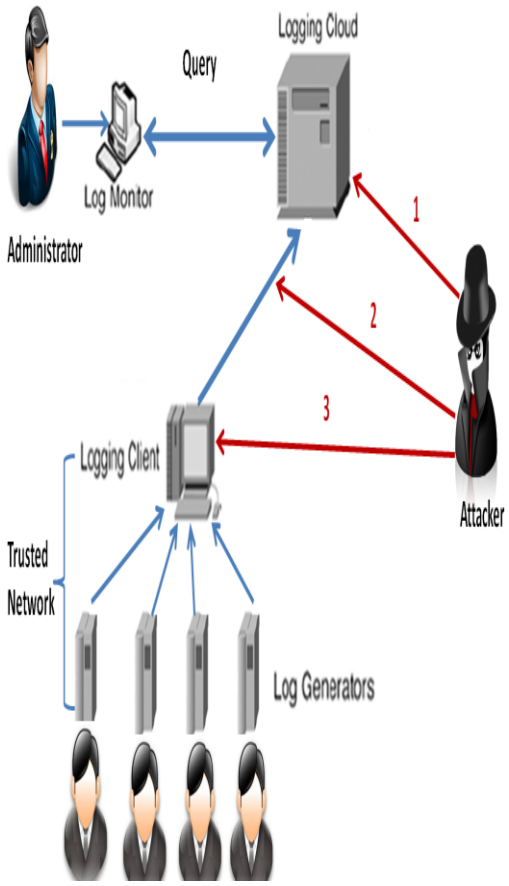

Fig. 1 Proposed System

C. Mathematical Model

This subsection elaborates the mathematical modelling required for implementation of cloud based architecture for log security. The encryption scheme designed is based on triple DES algorithm.

1. Logging client generates three independent secret keys K1, K2, K3 each of 56 bits using SecretKeyFactory class of java.

2. String rotateBits(String value)

Shift the characters of the string to the left one and moved the first position to the last position. Returns the rotated bits.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 3, Issue 3, September 2014)

3. char XOR(char, char)

Return the exclusive or operation on these two characters.

4. String DES(String msg, String k, boolean encrypt)

DES takes in the message and the original key as well as a boolean to decrypt or encrypt. Encryption and decryption is the same function but applying determines the order of how to apply the keys. createSubKeys(k) is called to generate the 16 subkeys. The msg is goes through the IP permutation. The msg is then split into right and left halves. The right halve goes through the feistel function, which uses the SBOX( ) in 6 bit chunks with the proper key of that iteration and is then XOR'd with the XOR() with the left half, this becomes the new right half and and the previous right half becomes the left and it goes through the feistel function again, with the next key. This is done for all 16 keys. At the end the left and right halves are switched then concatenated then goes through the IIP permutation. Returns the cipher text.

5. String SBOX(int, String)

Returns the SBOX value which takes the first and last 2 bits of a chunk to create the row of the SBOX matrix and the middle 4 bits to create the column. Then at the point the value is retrieved and returned to be used s a 4bit chunk.

IV. Results And Analysis

This section elaborates the results generated by the cloud based log security architecture. The evaluation of the architecture is done under two criteria, first is effect of encrypted log records over performance of entire architecture. Second, a criterion is effect of anonymous communication over performance of the architecture.

The result for effect of encrypted log records over performance of entire architecture is analyzed by testing effectiveness of different encryption settings in proposed architecture. We created test scenarios to understand the run time that is introduced by the number of messages. In our experiments, we used batch sizes of 10, 20, 30, 40, and 50 encrypted log records. For each log batch, we measured three different log preparation settings. First, we measured the time that it takes to fill up the log batch with log records without any security related preparation i.e we do not apply either tag generation process and encryption scheme. Thus, the first experiment gives us the best achievable performance of our proposed log security architecture.

In second case we measure the run time for filling up log batch with partial security. In this case we do not apply encryption scheme but only tag generation process. This includes unencrypted log records (no confidentiality) with calculated MD5 Hash value. In third case, we calculated the run time for security preparation that includes encrypted log records with corresponding MD5 hash. The results are shown in Fig. 2. Bottom, middle and top lines correspond to the runtime of no security, partial and full security settings. From Fig. 2 the run time in adopting our complete security preparation is about twice as much as no security.

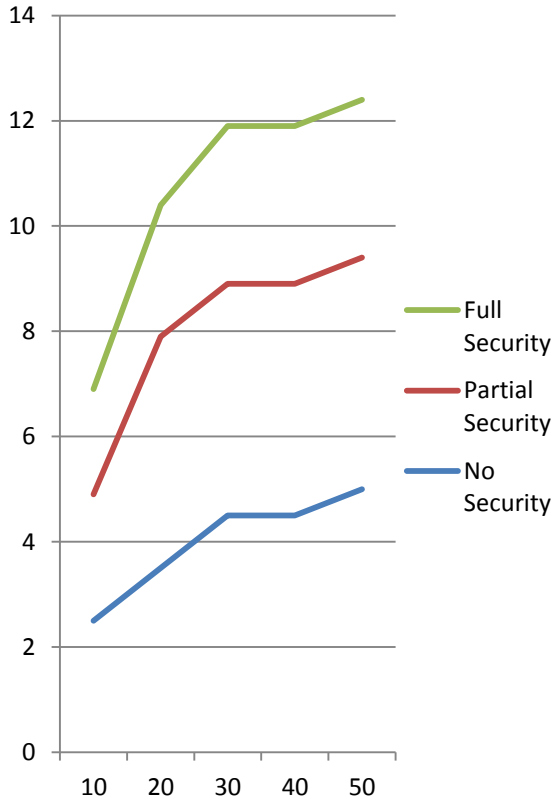

Fig 2 Run time of no security, partial and full security settings

We designed a set of experiments that measured the performance of anonymous communication. We kept the log batch size fixed to 100 encrypted log records. In our experiments, we compared the time it takes a log batch to traverse between the logging client and logging cloud. Fig.3 shows results from our experiments. The top line corresponds to the run time form proposed architecture of log security if executed on Amazon cloud using Amazon Web Services. From Fig. 3 we find that Amazon Web Services has a fairly constant run time. Also, we found interesting the fact that it takes about 10 s to send the first log batch to the cloud via Amazon web services.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 3, Issue 3, September 2014)

From our experience with Amazon Cloud, we found that it provides reliable data transmission. The bottom line shows the performance our architecture if we execute it using local network like LAN instead of Amazon cloud. Fig 3 shows the results of execution of our proposed architecture on Amazon web services and local network.

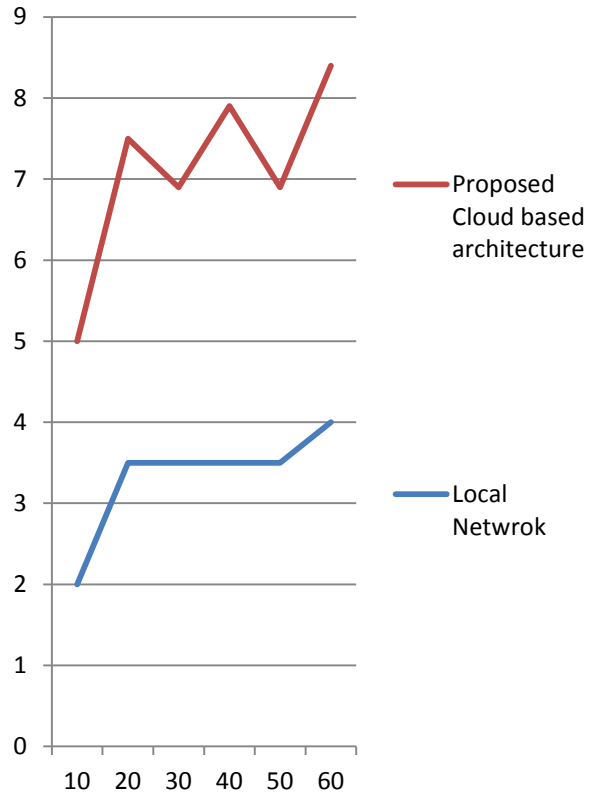

Fig 3 Results of execution of Amazon web services and local network

- *Analysis of Major Components of the Architecture*

We have developed and implemented architecture of cloud based log security. This architecture is based on four major components, log generator, logging client, logging cloud and log monitor. This section analyzes the working of major components of the proposed architecture.

Each machine generating log records is configured with a syslog-ng server to forward the log records to the log client residing on a different machine. All the components of the network topology are located on separate machines within the same network.

The log client is implemented as a multithreaded server application that receives log messages from syslog-ng log generators and puts them in a log batch.

It creates a Java object for a log batch and uploads the batch in a special packet as soon as the batch is filled. All the inter-node communications are carried out in the form of packet of different types depending on the operation. The protocol at the log client side starts by generating an initial set of keys. The batch is uploaded to the cloud with appropriate upload and deletes tags included in the packet object.

The implementation of the log monitor at present is somewhat fundamental. It is able to perform the batch retrieval and validate operations since they are supported by the cloud. The actual cloud server is implemented as a multithreaded server that accepts simultaneous connecting log clients and log monitors. Our implementation is such that it can be easily deployed on any existing cloud system. We have used Amazon Cloud provided by Amazon Web Services to store log data. We store all the log data in this cloud, along with each log entry we append a tag value generated by calculating MD5 hash value. The encryption scheme proposed is based on Triple DES algorithm. The encryption scheme allows encryption of log records in such a way that logging cloud can execute some queries on the encrypted logs. The only way for log monitor to retrieve the log data is to execute the queries and get response from cloud database.

- *Analysis of Triple DES Algorithm*

Triple Data Encryption algorithm is an encryption algorithm that is based off of the Data Encryption Standard encryption algorithm. The block size is 64 bits and the key sizes are 168, 112, or 56 bits with respect to keying option 1, 2, or 3. The input key sizes are 3 64 bit keys, which are shortened to 56 bits because of the internal key scheduler. The block of data is encrypted 3 times with each of the keys according to the keying options:

Keying Option 1: All of the keys are independent
Keying Option 2: K1 and K2 are independent and K3 = K1
Keying Option 3: All keys are identical K1=K2=K3

It reinforces DES by expanding the key space which is it's main weakness and protects itself from brute force attacks. The earliest standard that defines TDEA algorithm is in ANS X9.52. Today TDEA is widely used.

Example of Triple DES algorithm

Plain Text: 0123456789ABCDEF

Key Option: 1
Cipher Text: 4a9e56452906d49b
Key 1: 133457799BBCDFF1
Key 2: EFDCBA9876543210
Key 3: ABCDEF1234567890

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 3, Issue 3, September 2014)

Plain Text: 0123456789ABCDEF

Key Option: 2
Cipher Text: 56353b965baca3ca
Key 1: 133457799BBCDFF1
Key 2: EFDCBA9876543210
Key 3: 133457799BBCDFF1
Plain Text: 0123456789ABCDEF

Key Option: 3
Cipher Text: 85e813540f0ab405
Key 1: 133457799BBCDFF1
Key 2: 133457799BBCDFF1
Key 3: 133457799BBCDFF1

Variables Used:
MSG = 0123456789ABCDEF
K1=133457799BBCDFF1 K2= 133457799BBCDFF1
K3=133457799BBCDFF1
Key Option = 3
Mode = Encrypt

For 10,000 encryptions.
Profile for first 10 ranks
rank self accum count trace method
1 4.07% 4.07% 583 300034 java.util.Arrays.copyOfRange
2 3.56% 7.63% 510 300082 java.math.BigInteger.<init>
3 3.48% 11.11% 498 300043 java.util.Arrays.copyOfRange
4 3.36% 14.47% 481 300030 java.util.Arrays.copyOfRange
5 3.18% 17.64% 455 300072 java.util.Arrays.copyOfRange
6 2.44% 20.09% 350 300060 java.util.Arrays.copyOfRange
7 2.02% 22.10% 289 300044 java.math.BigInteger.toString
8 1.78% 23.88% 255 300075 java.math.BigInteger.<init>
9 1.76% 25.64% 252 300069
java.lang.AbstractStringBuilder.<init>
10 1.75% 27.40% 251 300061
java.lang.AbstractStringBuilder.<init>
Time
real 2m28.747s
user 2m28.241s
sys 0m0.232s

V. Conclusion And Future Scope

In this paper we elaborated the algorithms for securely creating, storing and retrieving log data stored in cloud-based framework. With all these algorithm our framework achieves all the desirable properties like correctness, tamper resistance, verifiability, confidentiality and privacy , required for secure logging using cloud-based architecture.

Our major objective is to provide a framework for an organization to store log data for long period of time and to reduce the additional cost of resources required for that. The designed framework mainly deals with all possible security attacks which can be done by attacker on the log files. In addition to the base architecture we have introduced an encryption schemes with which logging cloud can execute queries on log data stored without violating the confidentiality and privacy. The designed architecture deals with security and privacy issues not only during slog generation but also at log transmission, storage at cloud and retrieval.

As future work, some alternative approaches can be developed for providing non traceability and non sslinkability in network data transmission or communication. The alternatives like JAP,Ultrasurf and Freegate should be implemented to check their performance over our architecture. This is the future scope of our paper.

REFERENCES

[1] Indrajit Ray,K.Belyaev,"Secure Logging As A Service-Delegating log management to the cloud ", IEEE Systems Journal,June 2013.

[2] C.Lonvick,The BSD Syslog Protocol,RFC 3164,Internet Engineering Task Force, Network Working Group, 2001.

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

[4] M. Bellare and B. S. Yee, "Forward integrity for secure audit logs," Dept. Computer Science, University of California, San Diego, Tech. Rep., Nov. 1997.

[5] BalaBit IT Security , "Syslog-ng—Multiplatform Syslog Server and Logging Daemon" Available: http://www.balabit. com/network-security/syslog-ng

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

[7] D. Ma and G. Tsudik, "A new approach to secure logging," ACM Transaction Storage, VOL 5, No1, Mar. 2009.

[8] U. Flegel, "Pseudonymizing unix log file," Infrastruture Security, LNCS 2437. Oct. 2002.

[9] C. Eckert and A. Pircher, "Internet anonymity: Problems and solutions," 16th IFIP TC-11 Int. Conference. Inform. Security, 2001.

[10] B. Schneier and J. Kelsey, "Security audit logs to support computer forensics," ACM Trans. Inform. Syst. Security, VOL 2, NO2, May 1999.

[11] J. E. Holt, "Logcrypt: Forward security and public verification for secure audit logs," 4th Australasian Inform. Security Workshop, 2006.