Secure Logging as a Service In Cloud

T. Sakthisree, Kumaresan S, Manisha D, Prathapkannan M

Computer Science and Engineering, Kathir College of Engineering, Coimbatore, Tamil Nadu, India

ABSTRACT

Data and authorization security are needed everywhere. In case dealing with huge number of data in a cloud server, secured logging is must. This is because cloud servers are easily accessible and any one can access anywhere at any time. This is because intruders may have knowledge about the network where they are going to intrude. So data should be preserved well for intruders, hackers and unauthorised user. The main objective of this paper is to develop a secured logging as a service in cloud architecture. So in the proposed method, privacy and preservation methods are implemented. The secured logging contains six major functionalities to ensure more securities: Correctness, Confidentiality, data logs, Privacy, Preservation and VPS (Virtual proxy server). Confidentiality deals with sensitive information not displaying during search. Data logs deals with the data history for identifying appropriate users. Privacy scheme deals with file linking and data access history. So that secured logging as a service is much important for all kind of cloud server environment in order to provide proper login for authorized user and triggers out the unauthorized users.Preservation deals with enhanced colour code. Hackers can be avoided and intruders are can't be avoided.

Keywords : Secure Logging, Cloud Server, Virtual Proxy Serve, Network Security, Cloud Computing, SLAS, IaaS, PaaS, NAT

I. INTRODUCTION

Cloud Computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., computer networks, servers, storage, applications and services),which can be rapidly provisioned and released with minimal management effort.

Network Security consists of the policies and practices adopted to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Network security involves the authorization of access to data in a network, which is controlled by the network administrator. Users choose or are assigned an ID and password or other authenticating information that allows them access to information and programs within their authority.

II. METHODS AND MATERIAL

OBJECTIVE

Primary Objective

- The primary objective of this project is creating a secured data access with Secured Logging as a Service.
- The Secured Logging as a service will be enabled for both admin and user.
- The motivation is to find out the difference between User, Intruder, and Hackers. From the way of logging in to their networks

Secondary Objective

- Even thou admin login is more secured user should mention the security level of the uploading file.
- Improved Gaussian Mixture and Keystroke password enabled.
- Even keystroke value got leaked, the another security level of colour code was enabled.
- Log will records each activities of the user logging in
  - Last Accessed date
  - Last Accessed file

- Last Accessed IP address.

- Loggings in records are much important to finds out the type of user.

GOAL

The Goal of this project is to create a secured logging contains six major functionalities to ensure more securities: Correctness, Confidentiality, data logs, Privacy, Preservation and VPS (Virtual proxy server).

Advantages of the Proposed System

- Data logs
- Enhanced privacy
- Preservation
- VPS (Virtual Proxy Server)

Modules:

- Cloud Formation
- Data Access Creation
- Secured logging
- Virtual Proxy Server
- Logging report data

Modules Description:

Cloud Formation

The public cloud environment is the IaaS/PaaS Infrastructure or Platform as a Service that we rent from Linux (IaaS) or Microsoft (PaaS). Both are enabled for web hosting. Then, your SaaS stack will run under your Internet environment most likely in a virtualized one on your own equipment which would make it private. In this project we specialize in private cloud technology. Here we execute in a cloud environment. If strict security requirements go public or hybrid and if not, try the public or community cloud environment. So that here we are implementing a web services for the output purpose as well as the environment will be shown in actual while hosting the application. So finally SaaS can be fully utilized in cloud environment as IaaS/PaaS. Thus we formed cloud environment

Data Access Creation

According to this module, some data or records will be created by the admin or user. The creating data will be the data set to be accessed by any one. But while uploading the records into the server, the file's security levels will be mentioned. The security level will be spited into four major categories, namely Sensitive data records, Confidential records, Private records and public records. From the category itself the data security levels can be identified. This security level will be categorized according to the data access strength.

Secured Logging

This module provides a secured logging for user. It contains a colour code mechanism, Gaussian mixture and key stroke procedure, last file accessed date, last file accessed and last accessed IP address. From the input the SLAS procedure will finds out the user type. Here the user types are categorized into three variations namely, User, Intruder, Hacker. Exact user knows everything about the logging details. Intruder knows something about the network or access details and Hacker may have less knowledge about the network. A consolidated report will be updated in the user login zone for viewing the type of user, along with that a chart is shown for the percentage for attacks.

Methodologies

Properties of Secure Logging as A Service

Secure log management service based on the cloud computing paradigm. We will subsequently analyze our framework against these properties.

Correctness:

Log data is useful only if it reflects true history of the system at the time of log generation. The stored log data should be correct, that is, it should be exactly the same as the one that was generated.

Verifiability:

It must be possible to check that all entries in the log are present and have not been altered. Each entry must contain enough information to verify its authenticity independent of others. If some entries are altered or deleted, the ability to individually verify the remaining entries (or blocks of entries) makes it possible to recover some useful information from the damaged log. Moreover, the individual entries must be linked together in a way that makes it possible to determine whether any entries are missing.

Confidentiality:

Log records should not be casually browsed able or searchable to gather sensitive information. Legitimate search access to users such as auditors or system administrators should be allowed. In addition, since no one can prevent an attacker who has com-promised the logging system from accessing sensitive information that the system will put in future log entries, the goal is to protect the pre compromised log records from confidentiality breaches.

Privacy:

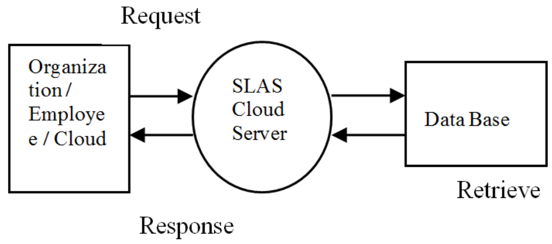
Log records should not be casually traceable or linkable to their sources during transit and in storage.

Secure Logging-as-a-Service

(SaaS) is an outsourcing model for security management. Typically, Security as a Service involves applications such as higher end data security services like Government Information, Military information, Banking Information and etc. These kinds of software delivered over the Internet but the term can also refer to security management provided in-house by an external organization. Storing important data with cloud storage providers comes with serious security risks. The cloud can leak confidential data, modify the data, or return inconsistent data to different users. This may happen due to bugs, crashes, operator errors, or misconfigurations. Further-more, malicious security breaches can be much harder to detect or more damaging than accidental ones: external adversaries may penetrate the cloud storage provider, or employees of the service provider may commit an insider attack. These concerns have prevented security-conscious enterprises and consumers from using the cloud despite its benefits. These concerns are not merely academic. Amazon started receiving public reports that data on its popular Simple Storage Service (S3) had been corrupted due to an internal failure; files no longer matched customers' hashes. One day later, Amazon confirmed the failure, and cited a faulty load balancer that had corrupted single bytes in S3 responses intermittently, under load. Another example of data security violation in the cloud occurred when Google Docs had an access-control bug that allowed inadvertent sharing of documents with unauthorized readers. Even worse a cloud storage provider went out of business after losing 45% of client data because of administrator error. Amazon's S3, Google's Big Table, HP, Microsoft's Azure, Nirvanix Cloud NAS, or others provide security guarantees in their Service Level Agreements (SLAs). For example, S3's SLA and Azure's SLA only guarantee availability: if availability falls below 99:9%, clients are reimbursed a contractual sum of money. As cloud storage moves towards a commodity business, security will be a key way for providers to differentiate themselves. In this paper, we tackle the problem of designing a cloud storage system that makes it possible to detect violations of security properties, which in turn enables meaningful security SLAs.

The cloud security setting is different from the set-ting of previous secure storage or file systems research. The first difference is that there is a financial contract between clients and the cloud provider: clients pay for service in exchange for certain guarantees and the cloud is a liable entity. In most previous work, the server was some group of untrusted remote machines that could not guarantee any service. The second difference is that scalability is more important, as it is one of the primary promises of the cloud. Enterprises are important customers for the cloud; they have many employees requiring highly scalable access control and have large amounts of data.

We design, build, implement, and evaluate Cloud Proof, a secure and practical storage system specifically designed for the cloud setting. Our first novelty is the idea and the mechanism of enabling customers to prove to third parties when the cloud violates the IWF properties. This enabling of proofs is in addition to detecting the violations and is not present in previous work. It includes the fact that the cloud can disprove false accusations made by clients; that is, in Cloud-Proof, clients cannot frame the cloud. We believe that such proofs are key to enabling security in SLAs with respect to these three properties. Customers and cloud can now establish a financial contract by which clients pay a certain sum of money for the level of security desired; customers have assurance that the cloud will pay back an agreed upon compensation in case their data security is forfeited because they can prove this violation. Without such proofs, the cloud can claim a smaller amount of damage to protect itself against significant financial loss and clients can falsely accuse the cloud. These proofs are based on attestations, which are signed messages that bind the clients to the requests they make and the cloud to a certain state of the data. For every request, clients and cloud exchange attestations. These attestations will be used in a lightweight auditing protocol to verify the cloud's behaviour.



Gaussian Mixture and Keystroke

Cloud computing security is an evolving sub-domain of computer security, network security, and, more broadly, information security. It refers to a broad set of policies, technologies, and controls deployed to protect data, applications, and the associated infrastructure of cloud computing.The method used here for security is keystroke logging. This allows only the right user to login at the right time. It is the action of tracking the keys struck on a keyboard, so that the person using the keyboard is unaware that their actions are being monitored. Whenever a user is created, the keystroke time of typing his/her password should be noted. When a user logins to sends details, the keystroke time for typing his/her password should matches with the time that is generated in the user creation. So this will provides a well security for the user's id and password from hackers. Working with Gaussian Mixture and Keystroke: The working of Gaussian Mixture and Keystroke based on the keyboard input given by the user.

These values are calculated into 3 values, namely

- Mean Value
- Actual Value
- Median Value

In case of the key stroke value said to be Mean value will be lesser than the actual value, the value will be 3.75 or 3.76 The actual values will the same value The median will be increased from the actual value; the value will be 3.76 or 3.77.

Enabling high security can be done by the actual value only.

III. CONCLUSION

Thus we are concluding that all the result obtained according to the committed abstract. In this paper, we consider a cloud storage system consists of storage servers and key servers. We integrate a newly proposed threshold encryption scheme and codes over exponents. The encryption scheme supports encoding, forwarding, and partial decryption operations in a distributed way. To decrypt a message of k blocks that are encrypted and encoded ton code word symbols, each key server only has to partially decrypt two codeword symbols in our system. By using the threshold proxy re-encryption scheme, we present a secure cloud storage system that provides secure data storage and secure data forwarding functionality in a decentralized structure. Moreover, each storage server independently performs encoding and re-encryption and each key server independently perform partial decryption. Our storage system and some newly proposed content addressable file systems and storage systems are highly compatible. Our storage servers act as storage nodes in a content addressable storage system for storing content addressable blocks. Our key servers act as access nodes for providing a front-end layer such as a traditional file system interface. Further study on detailed cooperation is required.

Future Enhancement

In future enhancement, the proxy server can be enhanced for more secured data transfer. Most of the time proxy refers to a layer-7 application on the OSI reference model. However, another way of proxying is through layer-3 and is known as Network Address Translation (NAT). The difference between these two technologies is the tier in which they operate, and the way of configuring the clients to use them as a proxy. In client configuration of NAT, configuring the gateway is sufficient. However, for client configuration of a layer-7 proxy, the destination of the packets that the client generates must always be the proxy server (layer-7), and then the proxy server reads each packet and finds out the true destination. Because NAT operates at layer-3, it is less resource-intensive than the layer-7 proxy, but also less flexible. As we compare these two technologies, we might encounter a terminology known as 'transparent firewall'. Transparent firewall means that the layer-3 proxy uses the layer-7 proxy advantages without the knowledge of the client. The client presumes that the gateway is a NAT in layer-3, and it does not have any idea about the inside of the packet, but through this method the layer-3 packets are sent to the layer-7 proxy for investigation.

- Layer 7 virtual proxy servers can be used.

- Output can be shown using some medical domain for real time implementation.
- EC2 Cloud server can be implemented.
- Key character length can be increased for more security
- HTTPS can be implemented.
- Network Address Translation (NAT) can be implemented

Big data concepts can be included in case of huge number of data transaction.

IV.REFERENCES

[1] U. Flegel, "Pseudonymizing unix log file," inProc. Int. Conf. Infrastru-ture Security, LNCS 2437. Oct. 2002, pp. 162–179.

[2] C. Eckert and A. Pircher, "Internet anonymity: Problems and solutions," in Proc. 16th IFIP TC-11 Int. Conf. Inform. Security, 2001, pp. 35–50 .

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

[4] 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.

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

[6] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," inProc. 12th Ann. USENIX Security Symp., Aug. 2004, pp. 21–21.

[7] The Tor Project, Inc. (2011, Sep.) Tor: Anonymity Online Online]. Available: http://www.torproject.org

[8] 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.

[9] A. Shamir, "How to share a secret," Commun. ACM, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[10] G. R. Blakley, "Safeguarding cryptographic keys," inProc. Nat. Comput. Conf., Jun. 1979, p. 313.

[11] R. Ostrovsky and M. Yung, "How to withstand mobile virus attack," in Proc. 10th Ann. ACM Symp. Principles Distributed Comput., Aug. 1991, pp. 51–59.

[12] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive secret sharing or: How to cope with perpetual leakage," inProc. 15th Ann. Int. Cryptology Conf., Aug. 1995, pp. 339–352.

[13] I. Teranishi, J. Furukawa, and K. Sako, "k-times anonymous authen-tication (extended abstract)," in Proc. 10th Int. Conf. Theor. Appl. Cryptology Inform. Security, LNCS 3329. 2004, pp. 308–322.

[14] D. L. Wells, J. A. Blakeley, and C. W. Thompson, "Architecture of an open object-oriented database management system,"IEEE Comput., vol. 25, no. 10, pp. 74–82, Oct. 1992.

[15] K. Nørv˚ ag, 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.