A Novel Cloud Log Assuring Soundness and Secrecy Scheme for Cloud Forensics

[1]*Silla Sai Tejaswini,* [2] *Mr. M. Dharani Kumar*
[1]PG Scholar, Dept. of CSE, P.V.K.K Institute of Technology, Anantapuramu-AP
[2]Assistant Professor, Dept. of CSE, P.V.K.K Institute of Technology, Anantapuramu-AP

ABSTRACT: *Cloud storage, security and privacy are fairly established research areas, which is not surprising considering the widespread adoption of cloud services and the potential for criminal exploitation (e.g. compromising cloud accounts and servers for the stealing of sensitive data). Interestingly though, cloud forensics is a relatively less understood topic. In the event that a cloud service, cloud server, or client device has been compromised or involved in malicious cyber activity (e.g. used to host illegal contents such as radicalization materials, or conduct Distributed Denial of Service (DDoS) attacks, investigators need to be able to conduct forensic analysis in order to "answer the six key questions of an incident – what, why, how, who, when, and where". The problem with such data is that we must trust the cloud service provider to give us the right information. They might give us false information or hold back some very important information.*

INTRODUCTION

Cloud storage, security and privacy are fairly established research areas, which is not surprising considering the widespread adoption of cloud services and the potential for criminal exploitation (e.g. compromising cloud accounts and servers for the stealing of sensitive data).

Interestingly though, cloud forensics is a relatively less understood topic. In the event that a cloud service, cloud server, or client device has been compromised or involved in malicious cyber activity (e.g. used to host illegal contents such as radicalization materials, or conduct Distributed Denial of Service (DDoS) attacks, investigators need to be able to conduct forensic analysis in order to "answer the six key questions of an incident – what, why, how, who, when, and where". The problem with such data is that we must trust the cloud service provider to give us the right information. They might give us false information or hold back some very important information.

For Securing Cloud and making a Investigation. A Log Management is a record of all the activities and events occurring and processed within an industry, organizations, application, network. Logging of activities is important aspect because log data can be used for decision making and troubleshooting of problems occurred at the time of working and also helps in fine tuning of system performance and identify various users who are violating policies. Project proposes an enhanced secure log management for securing user activities from malicious attacks. Log records play a significant role in digital forensic analysis of systems. To maintain log security and provide protecting from attackers we design a integrated novel log security algorithm which provides security to log files at all times. As the log files contain sensitive information we require confidentiality and privacy of log records is an important. Designing and deploying a secure logging software involves significant capital expenses that many organizations may find irresistible overwhelming. Our proposed delegating log management provides viable cost saving measure. The project identifies a novel frame work for a challenging secure cloud based log management service shown in Figure 1.1.

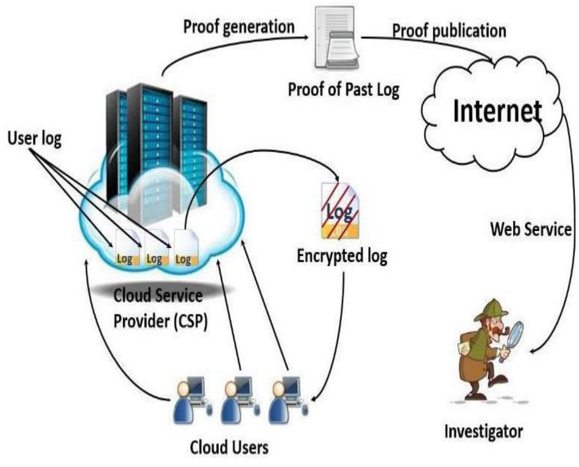

Fig-1: System Architecture

LITERATURE SURVEY

[1] P. McDaniel, “Data provenance and security,” IEEE Security and Privacy, vol. 9, no. 2, pp. 83–85, 2011

McDaniel addressed that accurate, timely, and detailed provenance information leads to good security decisions. One of the unanticipated consequences of the Internet age is a pervasive loss of context. Information is often filtered, sampled, repackaged, condensed, or altered to suit any number of purposes. Over time, the entropy of these processes causes information to lose its essential validity. This column argues the needs, applications, and challenges of providing greater access to data provenance in information systems.

[2] S. Rajbhandari, I. Wootten, A. Ali, and O. Rana, “Evaluating provenance-based trust for scientific workflows,” in 6th IEEE International Symposium on Cluster Computing and the Grid, vol. 1, Singapore, 16-19 May 2006, pp. 365–372

Provenance has been used to verify trust, trustworthiness, or correctness of information in many research areas. Rajbhandari et al. examined how provenance information is associated with a workflow in a Bio-Diversity application. Provenance is the documentation concerning the origin of a result generated by a process, and provides explanations about who, how, what resources were used in a process, and the processing steps that occurred to produce the result. Such provenance information is important to improve a scientist's ability to judge and place certain amount of trust on the generated data. We illustrate how provenance information associated with a workflow can be used to evaluate trust. This work is based on several use cases from a Bio-Diversity application. We also propose a simple architecture to illustrate our trust framework.

The next generation computing is Cloud computing, where we have centralized computing resources (both hardware and software) and the centralized resources are delivered as service over a network i.e. Internet, Intranet or Extranet. Cloud Computing provides huge storage, processing, applications, Operating systems, Network and various other infrastructures, all the specified features are centralized in big server called cloud server. These features can be accessed in various shapes required by the surfer, they can access in Systems, Mobiles, Tabs and other media required. Briefly discussing the common use of cloud is a symbol of abstraction in a complex infrastructure in centralized location. Cloud computing provides trust to remote services with a user's data, software, applications, security and computations accessed in any media. Central Cloud computing consists of hardware, Software and Application resources made available on the Internet and Mobile wireless technology as managed by third party services, all the cloud servers are accessed to third party and from third party users or surfers take access to use the resources in their required form. These services typically provide access to advanced software applications and high-end networks of server computers.

The next generation of computing in Internet will be cloud computing, through cloud computing we can reduce the infrastructure, maintenance of huge systems and provide green computing with one centralized system providing resources services to a wide range of users. To overcome the drawbacks of investment, maintenance and over rid of attackers the proposed architecture is cloud architecture. The following figure shows the structure of cloud computing.

SYSTEM ANALYSIS

Analysis is a logical process. The objective of this phase is to determine exactly what must be done to solve the problem. Tools such as Class Diagrams, Sequence Diagrams, Data Flow Diagrams and data dictionary are used in developing a logical model of system.

EXISTING SYSTEM

Existing system not guarantees the security, there are lot of disadvantages are there. Here we are going to overcome that.

Concern for security issues when users are surfing the Internet has increased recently. Now a days, many users are unaware that when they are browsing websites, these websites can track them and create profiles on the elements they access, the advertisements they see, the different links they visit, from which websites they come from and to which sites they exit, and so on.

In order to maintain user privacy, several techniques, methods and solutions have appeared. SecLaaS encrypts the log(s) using the investigating agency's public key and stores the encrypted log(s) in a cloud server. This ensures privacy and confidentiality of the cloud user, unless the particular user is subject to an investigation (e.g. via a court order).

To facilitate log integrity, SecLaaS generates proof of past log (PPL) with the log chain and publishes it publicly after each predefined epoch. A trust model was also suggested that stores the PPL in other clouds to minimize the risk of a malicious cloud entity altering the log. However, in SecLaaS, it is difficult to ensure or verify that the CSP is writing the correct information to the log, or that any information pertinent to the investigation is not omitted or modified. Specifically, SecLaaS does not provide the user the ability to verify the accuracy of the log (since the log is encrypted with the agency's public key).

Disadvantages

- No Security, attempt to block the account, hacking password etc.
- Log information can be easily attacked.
- The attacker can be a legitimate member of the network or can try to impersonate legitimate hosts.
- The attacker records a set of log messages "M" sent by a logging client to the logging cloud. Later, the attacker attacks the logging client and, in order to hide evidence about the attack sends "M" to the logging cloud.
- The attacker has access to the communication medium and can modify data during the transmission.
- The attacker impersonates as the logging client and begins sending log messages to the logging cloud.
- During transmission the attacker intercepts and reads log messages. In addition, the attacker reads log data stored at the logging cloud.
- The attacker can actively try to correlate log messages or network traffic to associate these messages with specific logging client, log monitor or log generators, causing privacy breaches.

PROPOSED SYSTEM

Extending SecLaaS, we propose a secure cloud logging scheme, Cloud Log Assuring Soundness and Secrecy (CLASS), designed to ensure CSP accountability (i.e. writing the correct information to the log) and preserve the user's privacy. Specifically, we include the capability for the user to verify the accuracy of their log. To do this, the log will be encrypted using the user's public key (rather than the agency's public key).

To avoid introducing unnecessary delays to the forensic investigation, during user registration with the cloud service, both the CSP and the user will collectively choose a public/private key pair referred to as content concealing key (CC-key) for the user. The corresponding (content concealing) private key will be shared with other CSPs secret sharing schemes. This would allow the private key to be regenerated whenever necessary. We also demonstrate how we can leverage Rabin's fingerprint and bloom filter in PPL generation to establish log veracity. We then implement CLASS in Open Stack and evaluate its performance.

Advantages

- Credentials, the account will be blocked. Only account holder can renew it. Log data is available to correct user and provides high security.
- Prevent Attackers in attacking the log through security.
- A protected log is tamper resistant in such a way that no one other than the creator of the log can shows valid entries no tampering is possible. The entries cannot be changed without detection.
- Provides to check that all entries in the log are present and haven't been altered. Each entry must contain enough information to check 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

IMPLEMENTATION

In the implementation phase software development is concerned with translating design specifications into source code. The primary goal of implementation is to write the source code internal documentation so that conformance of the code to its specification can be easily verified, and so that debugging, testing and modifications are erased. This goal is achieved by making the source code as clear and straightforward as possible. Simplicity, clarity and elegance are the hallmarks of good programs. Obscurity, cleverness and complexity are indications of inadequate design and misdirected thinking.

Source code clarity is enhanced by strutted techniques, by good coding style, by appropriate documents, by go internal comments, and by the features provided in the modern programming languages.

The main aim of structured coding is to adhere to single entry, single exit constructs in the majority of situations since it allows one to understand program behavior by reading the code from beginning to end. Bust strict adherence to this construct may cause problems it raises concerns for the time and space efficiency of the code. In some cases, single entry and single exit programs will require repeated code segments or repeated subroutines calls. In such cases, the usage of this construct would prevent premature loop exits and branching to exception handling code. So, in certain situations we violate this construct to acknowledge the realities of implementation although our intent is not encouraging poor coding style.

In computer programming, coding style is manifest in the patterns used by programmers to express a desired action or outcome good coding style can overcome the deficiencies of primitive programming languages, while poor style can defeat the intent of an excellent language. The goal of good coding style is to provide easily understood straightforward, elegant code.

Every good coding style performs the following Do's

- Introduce user-defined data types to model entities in the problem domain.
- Use a few standard, agreed-upon control statements.
- Hide data structures behind access functions.
- Use goto's in a disciplined way.
- Isolate machine dependencies in a few routines.
- Use indentation, parenthesis, blank lines and borders around comment blocks to enhance readability.
- Carefully examine the routines having fewer than 5 or more than 25 executable statements.

The following are the Don'ts of good coding style

- Avoid null then statements.
- Don't put nested loops very deeply.
- Carefully examine routines having more than five parameters.
- Don't use an identifier for multiple purposes.

Adherence implementation standards and guidelines by all programmers on a project results in a product of uniform quality. Standards were defined as those that can be checked by an automated tool. While determining adherence to a guideline requires human interpretation. A programming standard might specify items such as:

- The nested depth of the program constructs will not executed five levels.
- The go to statements will not be used.

- Subroutines lengths will not exceed 30 Lines.

Implementation was performed with the following objectives.

- Minimize the memory required.
- Maximize output readability or clarity.
- Maximize source text readability.
- Minimize the number of source statements.
- To ease modification of the program.
- To facilitate formal verification of the program.
- To put the tested system into operation while holding costs, risks and user irritation to minimum.

Supporting documents for the implementation phase include all base-lined work products of the analysis and design phase.

MODULES

User Module:

In this module if a user wants to login into the database ,he/she should register their details first.These details are maintained in a Database.

Admin Module:

In this module admin has to enter username and password,if its matches then admin can the view details of the registered users.Also admin can see the details that are stored in the cloud database.

Account Blocking Module:

If an user have entered by giving fake username/password more than three times then his/her account will be blocked.

Account Renewal Module:

If someone's account was blocked Then he/she has to answer some security questions for their renewal, then only their account is in active state. This process could be done by admin.

TTP (Trusted Third Party) Login Module:

In this module TTP has monitors the users data by verifying it and stored the data in a database. Also ttp checks the CSP(CLOUD SERVICE PROVIDER),and find out whether the csp is authorized one or not.

CSP (Cloud Service Provider) LOGIN:

In this module CSP has to login first. Then only he can store the users information in his cloud server. Ttp can only check the csp whether the csp is authorized csp or not.If its fake,ttp wont allow the file to store in cloud server.

CONCLUSION

In this paper, we proposed a secure logging scheme for cloud computing with features that facilitate the preservation of user privacy and that mitigate the damaging effects of collusion among other parties. CLASS preserves the privacy of cloud users by encrypting cloud logs with a public key of the respective user while also facilitating log retrieval in the event of an investigation. Moreover, it ensures accountability of the cloud server by allowing the user to identify any log modification. This has the additional effect of preventing a user from repudiating entries in his own log once the log has had its PPL established. Our implementation on OpenStack demonstrates the feasibility and practicality of the proposed scheme. The experimental results show an improvement in efficiency thanks to the features of the CLASS scheme, particularly in verification phase.

REFERENCES:

[1] C. Wilson, H. Ballani, T. Karagiannis, and A. Rowtron, “Better never than late: Meeting deadlines in datacenter networks,” SIGCOMM Comput. Commun. Rev., vol. 41, no. 4, pp. 50–

61, 2011.

[2] A. D. Papaioannou, R. Nejabati, and D. Simeonidou, “The benefits of a disaggregated data centre: A resource allocation approach,” in Proc. IEEE GLOBECOM, pp. 1–7, Dec 2016.

[3] A. Tchernykh, U. Schwiegelsohn, V. Alexandrov, and E. ghazaliTalbi, “Towards understanding

uncertainty in cloud computing resource provisioning,” in Proc. ICCS, pp. 1772–1781, 2015.

[4] J. Hu, J. Gu, G. Sun, and T. Zhao, “A scheduling strategy on load balancing of virtual machine resources in cloud computing environment,” in Proc. PAAP, pp. 89–96, 2010.

[5] K.-M. Cho, P.-W. Tsai, C.-W. Tsai, and C.-S. Yang, “A hybrid metaheuristic algorithm for vm

scheduling with load balancing in cloud computing,” Neural Comput. Appl., vol. 26, no. 6, pp.

1297– 1309, 2015.

[6] S. Rampersaud and D. Grosu, “Sharing-aware online virtual machine packing in

heterogeneous resource clouds,” IEEE Transactions on Parallel and Distributed Systems, vol.

28, pp. 2046–2059, July 2017.

[7] S. S. Rajput and V. S. Kushwah, “A genetic based improved load balanced min-min task scheduling algorithm for load balancing in cloud computing,” in 2016 8th International

Conference on Computational Intelligence and Communication Networks (CICN), pp. 677–681, 2016.

[8] S. T. Maguluri, R. Srikant, and L. Ying, “Stochastic models of load balancing and scheduling

in cloud computing clusters,” in Proc. IEEE INFOCOM, pp. 702–710, 2012.

[9] S. H. H. Madni, M. S. A. Latiff, Y. Coulibaly, and S. M. Abdulhamid, “Resource scheduling for

infrastructure as a service (iaas) in cloud computing: Challenges and opportunities,” Journal of Network and Computer Applications, vol. 68, no. Supplement C, pp. 173–200, 2016.