Survey paper on different Cloud Auditing Systems

Poonam Malik[a],Dr. Varsha Bodade[b]
[a]Terna Engineering College, Navi Mumbai, 400706,India
[b]Associate Professor, Terna Engineering College, Navi Mumbai, 400706,India
[a]mpoonam.malik@gmail.com

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: Let us start by considering that there is a public pool of computer resources, these resources are made available as and when required i.e., are offered on-demand to the users. This is Cloud Computing in its simplest and most basic form. The different cloud services being offered can be categorized as application as a service (PaaS), infrastructure as a service (IaaS), and software as a service (SaaS). The requirements of a cloud user fall under any of these services and accordingly can be offered to the cloud user.

In current times, there is lot of interest in cloud computing as well as in its adoption. But the cloud users are fearful of losing the power and governance due to lack of transparency, accountability and confidence in the cloud. To improve the trust of cloud users, the cloud can be audited and verified against cloud user's security properties. This helps in instilling a sense of faith in cloud users that their security properties are respected in the cloud. The cloud presents several problems in collection of data and processing due to the irregularity of information architecture and the lack of correlation. Furthermore, on one hand the size of cloud is humongous and on the other hand there is continuous or runtime need of validation, hence the verification of security properties becomes a difficult task.

Still, lot of work is happening in cloud security auditing. In this paper, we will try to review and summarize some of the recent work done in this area.

Keywords: Cloud, security auditing, retroactive cloud auditing, intercept and check cloud auditing, proactive cloud auditing

1. Introduction

Cloud service providers usually use a multi-tenancy model, in which multiple tenants can access cloud services at the same time, to reduce costs and optimise resources. Although multi-tenancy allows for resource sharing at a low cost, it also increases the security risk associated with hosted applications. Multi-tenancy arrangements may result in denial of service and data leakage between tenants in the cloud. Security auditing can be an effective solution for reducing these concerns.

However, there are several obstacles to cloud auditing. First one being the vast gap between high-level recommendations provided in cloud-specific standards (e.g., Cloud Control Matrix (CCM) [1] and ISO 27017 [2]) and the low-level logging information in cloud infrastructures (e.g., OpenStack [3]), and cloud scale (e.g., a large-size cloud has about 10,000 tenants and considering that each tenant will have 10 users in average, a total of 100,000 users [4]). Along with this, the use of heterogeneous cloud solutions complicates data collection and analysis in auditing due to its self-provisioning existence, organisational difficulty due to multi-tenancy, and self-provisioning nature.

Before moving ahead, let us see existing cloud auditing techniques which can be divided into three types:

1. Retroactive cloud auditing techniques – These approaches detect security violations after they have occurred (e.g., [5], [6]) and therefore are unable to avoid security breaches until they occur. As a consequence, tenant data is released to the public domain or service is interrupted.

It can also be said that in cloud, this approach is a traditional way to check the compliance of various cloud properties

2. Intercept and check cloud auditing techniques – These approaches grant or reject incoming user requests after checking compliance (e.g., [7], [8]). While the compliance is being verified, the corresponding event instances remain blocked. Since each user request is checked first, responding to each user request takes time.

3. Proactive cloud auditing techniques – The idea of proactive security auditing for clouds differs from the conventional concept of security auditing. These systems (e.g. [7], [8]) learn from the intercepted events and keep proactively analysing the changes in cloud. The changes can be done by management or admin operations which needs to be audited with respect to security policies. Hence these approaches try to verify the user request in advance i.e., even before the requests are intercepted by the cloud system.

Different auditing works have been done on different cloud layers e.g., data, user, virtual network and SDN. Below is the discussion on works done in different cloud layers (user and virtual network) which fall into different cloud auditing techniques.

We will discuss work done in Retroactive approach, Intercept and check approach and Proactive cloud auditing approaches below.

2. Literature Review

The work proposed by S. Majumdar, T. Madi, Y. Wang, Y. Jarraya, M. Pourzandi, L. Wang, and M. Debbabi in [5] follows a retroactive approach. As discussed above, this approach can only capture a violation of security compliance after they have occurred. The coverage area of this security auditing framework is limited to user cloud layer and it supports OpenStack cloud platform. It utilises first order logic to verify the security properties and supports multi-domain RBAC (Role Based Access Control) authorization and authentication model.

OpenStack [3] is now one of the most widely used cloud management systems. The proposed cloud protection compliance auditing system is applied and evaluated on OpenStack. In a multi-domain cloud setting, the architecture proposed in [5] places a special emphasis on identity and access management. The experimental results also indicate that auditing large clouds using structured methods is feasible. According to the results of the proposed auditing solution, 60 thousand users can be handled in less than one minute.

The work suggested by T. Madi, S. Majumdar, Y. Wang, Y. Jarraya, M. Pourzandi, and L. Wang in [6] follow a retroactive approach. Hence it can only capture a security compliance violation after the fact. The coverage area of this security auditing framework is spread across network level and Virtual Infrastructure cloud layer. It supports OpenStack cloud platform and utilises first order logic to verify the security properties.

The research presented in [6] suggests an automated method for auditing cloud infrastructure from a structural standpoint. The focus of the audit is on virtualization-related security properties and ensuring continuity across the various control layers. The proposed auditing framework is built on top of OpenStack, the most widely used cloud infrastructure management platform. Numerous experimental findings are presented on assessing or evaluating properties relevant to: a) auditing inter-layer continuity, b) virtual system co-residence, and c) virtual property isolation. These findings support the proposed framework's scalability and validity.

S. Majumdar, T. Madi, Y. Wang, Y. Jarraya, M. Pourzandi, L. Wang, and M. Debbabi suggested a system in [7] that uses an intercept and check approach. This approach verifies each user request for security invariants before granting or refusing it. This security auditing framework's coverage area is restricted to the user cloud layer, and it supports the OpenStack cloud platform. It supports multi-domain RBAC (Role Based Access Control), Attribute Based Access Control (ABAC), and Single Sign-on (SSO) authorization and authentication models and uses first-order logic to validate security properties.

The work in [7] proposes that by conducting expensive operations only once, the response time of the auditing framework can be reduced to a realistic level. After the time-consuming operations are finished, incremental runtime verification is performed as and when a request from the cloud management system is issued. The results of the experiments show that runtime security auditing in large cloud environments is feasible using this method (e.g., solution in [7] takes 500 milliseconds or less to perform runtime auditing of 100,000 users).

Y. Luo, W. Luo, T. Puyang, Q. Shen, A. Ruan, and Z. Wu introduced an intercept and review mechanism in [8] that verifies protection invariants for each user request before approving or refusing it. The OpenStack Security Modules (OSM) project [8] has built a less restrictive access management system for use with OpenStack. The various access control models can be implemented as loadable modules in OpenStack using this least-invasive access control system. A framework for integrating several policies into a single decision is also suggested.

The work in [8] includes OSM design and implementation, as well as the development of a new service called patron. This paper also contains an attachment module called access endpoint middleware (AEM) in addition to OSM and patron. The access control rules specified by cloud tenants are audited at the user level during runtime auditing. Patron also enforces these laws in the cloud by using the OpenStack-supported middleware.

The experimental results show that using OSM improves the stability and security of policy management without impacting other services. In addition, the average performance overhead is 7.3 percent, which is deemed sufficient for practical use.

S. Bleikertz, C. Vogel, T. Groß, and S. Modersheim's work [9] takes a proactive approach, trying to check user requests in advance, i.e., before they are intercepted by the cloud system. This security auditing framework's coverage area is restricted to the virtual infrastructure layer, and it supports the VMware cloud platform.

Weathermen [9] is a security system that investigates changes triggered by management operations in compliance with security policies in a proactive manner. This is achieved by contributing the first structured model of cloud management operations that uses graph transformations to capture their effect on infrastructure. The used method joins a model of service with information flow analysis together with a policy verifier for list of

security and organisational policies. This proactive framework does not only include runtime implementation for infrastructure security policies but also a what-if study for change planning.

S. Majumdar, Y. Jarraya, T. Madi, A. Alimohammadifar, M. Pourzandi, L. Wang, and M. Debbabi also suggest a proactive approach in [10] that tries to check user requests in advance, i.e., before they are intercepted by the cloud system. This security auditing framework's coverage area includes the user, network, and virtual infrastructure layers, and it supports the OpenStack cloud platform.

This work [10] uses dependency models to pre-compute security compliances and proactively check them. The proposed system's key concept is to start planning for verification ahead of time, when the system is a few steps ahead of possible violation-causing operations. While this work will verify a broader range of security properties, properties such as proper constraint checking, minimum exposure and session timeout are not captured by the dependency models.

A cloud with 100,000 virtual ports is considered to be large, and response time for same is measured at about 8.5ms. This can be considered practical for a large cloud.

The work we discussed above is summarised in the table below. The current proposals are classified according to their methods (e.g., Retroactive, Intercept and Check, Proactive), coverage (e.g., User level, Network level, Virtual infrastructure), features (e.g., First order logic, RBAC, ABAC, and SSO verification), and supporting platforms (e.g., OpenStack, Azure, VMware).

Table – 1: Comparison of discussed different existing solutions

	Approaches			Features				Coverage			Platforms Supported				Experimental Results
Proposals	Retroactive	Intercept-and-check	Proactive	First-order-logic	Verifying RBAC	Verifying ABAC	Verifying SSO	User-level	Network-level	Virtual Inf.	OpenStack	Azure	Vmware	Adaptable to others	
Majumdar et al. [5]															Handles auditing of 60000 users in less than 1 minute.
Madi et al. [6]															Handles auditing of 10000 users in less than 8 seconds.
Majumdar et al. [7]															Handles runtime auditing of 100,000 users within 500 ms.
Patron [8]															-
Weatherman [9]															Handles runtime auditing of 100 VMs within 500ms.
Majumdar et al. [10]															Handles 100000 ports with maximum response time of 8.5ms.

The symbol (•) signifies that corresponding feature is offered by the proposal.

3. Conclusion

First, retroactive methods (e.g., [5,6]) identify enforcement breaches after they have occurred by reviewing various cloud settings and logs. They can't, however, stop security breaches from spreading or causing permanent harm (e.g., leak of data or denial of service). Second, intercept-and-check methods (e.g., [7, 8]) verify the compliance of a user request before approving or rejecting it, resulting in a substantial delay in responding to each

user request. Third, the proactive approaches in [9, 10] double-check the future modification plan to spot any possible deviations from the intended structure. Due to the complex and unpredictable existence of clouds, having a potential adjustment plan in advance isn't always possible, and therefore this approach isn't appropriate for clouds. In conclusion, current works suffer from a minimum of one limitation of the following:

(1) security properties of limited set are supported,

(2) since incoming requests are checked after getting intercepted, hence framework responds with delay, and

(3) lots of manual efforts are required or involved.

4. Future Scope

We propose enhancing the security auditing framework [7] with a signature verification system as a potential project. The proposal in [7] was chosen because it supports the following distinct characteristics:

1. This study has the potential to avoid the pitfalls of retroactive approaches while also obviating the need for a future reform plan, as proactive approaches do.

2. This paper suggests using an intercept and check method to inspect users at runtime.

3. The security properties of all three authentication and authorization mechanisms are supported by this work (RBAC, ABAC and SSO).

4. This work is adaptable to other cloud platforms as well.

To improve the security and the response time further, it is proposed to intercept queries between the database server and code server using a signature verification system. If the intercepted query signature (at runtime) does not match with any of the signatures from the signature store, the further execution of the intercepted query can be prevented to avoid the security threat. The prevention of the execution of spurious queries by signature verification method will not only improve the security but will also improve the response time of runtime auditing in clouds.

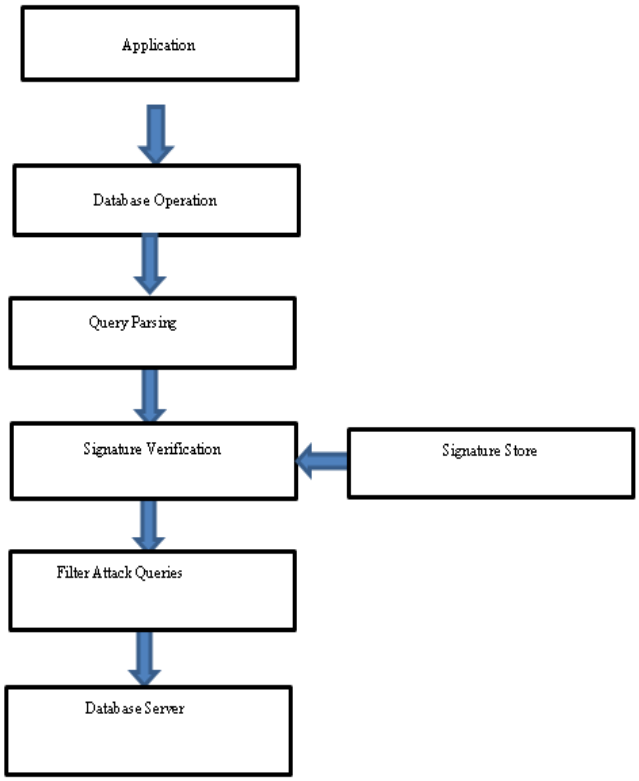

Figure – 1: Standalone Signature Verification Block Diagram

The main idea towards use of signature verification system is: to find/locate the source of spurious query intercepted by interceptor module, to prevent the execution of spurious queries thereby providing an additional security feature and to improve the CPU & memory usage, time & memory efficiency of the system.

This will help to alleviate cloud tenants' fear of losing power and governance, which continues due to a lack of accountability and confidence.

Acknowledgements

I would like to thank my guide Dr. Varsha Bodade for her expert advice and encouragement in completing this paper

References

1. Cloud Security Alliance, “Cloud control matrix CCM v3.0.1,” 2014, available at: https://cloudsecurityalliance.org/research/ccm/.
2. ISO Std IEC, “ISO 27017,” Information technology- Security techniques (DRAFT), 2012.
3. OpenStack, “OpenStack opensource cloud computing software,” 2015, available at: http://www.openstack.org.
4. OpenStack, “OpenStack user survey,” 2016, available at:https://www.openstack.org.
5. S. Majumdar, T. Madi, Y. Wang, Y. Jarraya, M. Pourzandi, L. Wang, and M. Debbabi, “Security compliance auditing of identity and access management in the cloud: Application to OpenStack,” in CloudCom, 2015.
6. T. Madi, S. Majumdar, Y. Wang, Y. Jarraya, M. Pourzandi, and L. Wang, “Auditing security compliance of the virtualized infrastructure in the cloud: Application to OpenStack,” in CODASPY, 2016.
7. S. Majumdar, T. Madi, Y. Wang, Y. Jarraya, M. Pourzandi, L. Wang, and M. Debbabi, "User-Level Runtime Security Auditing for the Cloud," Springer Advances in Information Security, 2019.
8. Y. Luo, W. Luo, T. Puyang, Q. Shen, A. Ruan, and Z. Wu, OpenStack security modules: a least invasive access control framework for the cloud, in IEEE 9th International Conference on Cloud Computing (CLOUD) (2016).
9. S. Bleikertz, C. Vogel, T. Groß, and S. M¨odersheim, “Proactive security analysis of changes in virtualized infrastructure,” in ACSAC, 2015.
10. S. Majumdar, Y. Jarraya, T. Madi, A. Alimohammadifar, M. Pourzandi, L. Wang, and M. Debbabi, “Proactive verification of security compliance for clouds through pre-computation: Application to OpenStack,” in ESORICS, 2016