Cloud Security Auditing: Major Approaches and Existing Challenges

Suryadipta Majumdar[1], Taous Madi[2], Yosr Jarraya[3],
Makan Pourzandi[3], Lingyu Wang[2], and Mourad Debbabi[2]

[1] Information Security and Digital Forensics, University at Albany, Albany, NY, USA
smajumdar@albany.edu
[2] CIISE, Concordia University, Montreal, QC, Canada
{t_madi,wang,debbabi}@encs.concordia.ca
[3] Ericsson Security Research, Ericsson Canada, Montreal, QC, Canada
{yosr.jarraya,makan.pourzandi}@ericsson.com

Abstract. Cloud computing is emerging as a promising IT solution for enabling ubiquitous, convenient, and on-demand accesses to a shared pool of configurable computing resources. However, the widespread adoption of cloud is still being hindered by security and privacy concerns. Various cloud security and privacy issues have been addressed in the literature. However, the mere existence of such security mechanisms is usually insufficient to fully relieve cloud tenants from their security and privacy concerns. To increase tenants' trust in cloud, it is of paramount importance to provide adequate auditing mechanisms and tools to verify the security postures of their applications. However, there are currently many challenges in the area of cloud auditing and compliance validation. There exists a significant gap between the high-level recommendations provided in most cloud-specific standards and the low-level logging information currently available in existing cloud infrastructures. Furthermore, the unique characteristics of cloud computing may introduce additional complexity to the task, e.g., the use of heterogeneous solutions for deploying cloud systems may complicate data collection and processing and the sheer scale of cloud, together with its self-provisioning, elastic, and dynamic nature. In this paper, we conduct a survey on the existing cloud security auditing approaches. Additionally, we propose a taxonomy identifying the classifications based on auditing objectives and auditing techniques. We further devise a systematic process flow for cloud security auditing. Also, we conduct a comparative study on existing works to identify their strengths and weaknesses. Finally, we report existing challenges in cloud security auditing.

Keywords: Security auditing, cloud security, auditing challenges, survey.

1 Introduction

Cloud computing has been gaining momentum as a promising IT solution for enabling cost-effective, ubiquitous, and on-demand access to a shared pool of configurable computing resources. Based on the provided services, cloud computing has been divided into three main models, namely, infrastructure as a service (IaaS), platform as a service

(PaaS), and software as a service (SaaS). In those models, there exist at least three main stakeholders: cloud service providers, tenants and their users.

A cloud service provider owns a significant amount of computational, storage and networking resources, and offers different paid services (e.g., IaaS, PaaS, etc.) to its customers by utilizing this pool of resources. A cloud tenant, the direct customer of cloud providers, enjoys the ad-hoc and elastic (i.e., on demand provisioning and deprovisioning) nature of the cloud to use the shared pool of resources for conducting his operations. Usually, tenants are different companies or departments within a company, while users are customers availing services offered by cloud tenants.

While cloud computing has seen such increasing interests and adoption, the fear of loosing control and governance still persists due to the lack of transparency and trust [41]. Security auditing and compliance validation may increase cloud tenants' trust in the service providers by providing assurance on the compliance with the applicable laws, regulations, policies, and standards. However, there are currently many challenges in the area of cloud auditing and compliance verification. For instance, there exists a significant gap between the high-level recommendations provided in most cloud-specific standards (e.g., Cloud Control Matrix (CCM) [7] and ISO 27017 [22]) and the low-level logging information currently available in existing cloud infrastructures (e.g., OpenStack [38]). In practice, limited forms of auditing may be performed by cloud subscriber administrators [36], and there exist a few automated compliance tools (e.g., [12, 47]) with several major limitations, which are discussed later in this section. Furthermore, the unique characteristics of cloud computing may introduce additional complexity to the task, e.g., the use of heterogeneous solutions for deploying cloud systems may complicate data collection and processing, and the sheer scale of the cloud together with its self-provisioning, elastic, and dynamic nature, may render the overhead of many verification techniques prohibitive. In particular, the multi-tenancy model and self-service nature of clouds usually imply significant operational complexity, which may prepare the floor for misconfigurations and vulnerabilities leading to violations of security compliance. Therefore, the security compliance verification with respect to security standards and policies is desirable to boost the trust relationship between the cloud stakeholders. Evidently, the Cloud Security Alliance (CSA) has recently introduced the Security, Trust & Assurance Registry (STAR) for security assurance in clouds, which defines three levels of certification, namely, self-auditing, third-party auditing, and continuous, near real-time verification of security compliance [8]. However, above-mentioned complexities coupled with the sheer size of clouds (e.g., a decent-size cloud is said to have around 1,000 tenants and 100,000 users [39]) implies one of the main challenges in cloud security auditing. In summary, the major challenges are to handle the unique nature of cloud and to deal with the sheer size of cloud in providing a scalable and efficient security auditing solution for clouds.

To this end, existing approaches can be roughly divided into three categories. First, the retroactive approaches (e.g., [29, 32, 48, 50, 12, 47, 23, 11, 3]) catch compliance violations after the fact by verifying different configurations and logs of the cloud. However, they cannot prevent security breaches from propagating or causing potentially irreversible damages (e.g., leaks of confidential information or denial of service). Second, the intercept-and-check approaches (e.g., [27, 5, 37, 19, 43]) verify the compliance of each user request before either granting or denying it, which can solve the limitation of the former approach. However, existing intercept-and-check methods cause a substantial delay in responding to each user request. Third, the proactive approaches, as in [31,30,51,5,37], address the limitations of previous approaches by starting the auditing process in advance and responding in a practical time at runtime. However, this approach is still suffering from certain practicality issues, such as how to decide about triggering the proactive step and how to reduce the manual process involved in the auditing process (a detailed discussion is provider in Section 4).

Contributions. The main contributions of our paper are as follows:

- As per our knowledge, this is the first effort to study the existing work on cloud security auditing and categorize the current techniques based on their adopted techniques and auditing objectives. To this end, we first study the landscape of cloud security auditing, then identify the existing categories and finally propose a taxonomy to present the whole landscape.
- In addition, we are the first to identify the structure of the automated security auditing process. For this purpose, we utilize our above-mentioned study to identify the mandatory steps of an automated security auditing system, and present the process flow of such auditing process.
- Furthermore, we are the first to conduct a qualitative comparison between existing works to highlight their coverage, strengths and weaknesses.
- Finally, we report the unaddressed challenges in cloud security auditing as the key observations of this survey. Our hope is that those challenges will draw the attention among security researchers to further improve the field of cloud security auditing.

The remainder of the paper is organized as follows. Section 2 discusses the structure of the automated security auditing process. Then, Section 3 describes the existing works, presents our proposed taxonomy and summarizes the findings of our comparative study. Afterwards, in Section 4, we report the existing challenges in cloud security auditing. Section 5 discussed different aspects of cloud security auditing. Finally, Section 6 concludes the paper discussing potential future work.

2 Structure of the Automated Security Auditing Process

Though security auditing is not a new process, automation of this process and complexity of targeted infrastructures introduce non-trivial challenges. Manual auditing is still in practice, where internal or third party auditors conduct the auditing process based on the collected data/evidence. Initial approaches of automating the auditing process are mostly to detect network intrusions. Later it has been adapted in other domains, such as data systems, access control and distributed systems. One of the most recent additions in the list is the cloud infrastructure. Based on the proposed solutions and best practices, we identify different phases (as in Figure 1) of an automated security auditing process.

Defining the Scope and Threat Model. As a very first step, an organization should define the scope of its auditing. Part of it is to identify the critical and sensitive assets, operations and the modules in the system that deal with those assets and operations.

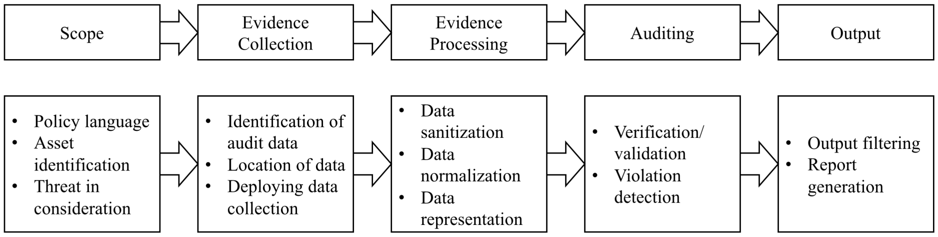

Fig. 1: Different steps of the cloud security auditing process

The following step is to identify threats or nature of threats to be considered for the auditing process. Most of the time, threat model depends on the nature of the business and demand of customers. Part of this step is to describe security assumptions while considering each threat. To this end, last few years different studies have been conducted to identify risks and threats in the cloud computing ecosystem. Based on those threats, several security properties are proposed by CSA [6], ENISA [14], ISO [22], NIST [35], CUMULUS [9], etc.

Data/Evidence Collection. The next phase is to gather evidences/data to conduct the audit process. Based on the target system and threat model, audit data is enlisted. In some cases (e.g., cloud and distributed systems), locating those audit data is non-trivial.

The data collection phase has become more dynamic with the virtualization and multi-tenancy; which results in an increase in the amount of data to be collected. We also consider security aspects of data collection in addition to the different runtime and continuous data collection techniques of different data types. The trust model ensures that the audit data provided by a tenant is real and fresh. At the same time, there might exist the privacy concerns in a central auditing system, such as any tenant must not leak any sensitive information to the auditor, which can benefit any other tenants in case of colluding with the auditor.

In the cloud, most of the audit data is any events, logs and system configurations. Data collection techniques vary in terms of targeted environments and data, e.g., what data to collect based on the scope, threat model and objectives, and how to collect data (more challenging in a cloud-based system).

Data/Evidence Processing. The previous step collects raw data from the system. It requires further processing to be able to conduct auditing. In case of verifying compliance with a policy language, it depends on the language. Collected data needs to be sanitized, as data is collected from different sources. For better understanding and interpretation, different correlation methods are applied on sanitized data to categorize them. There are different techniques (e.g., call graph, information flow graph, reachability graph) to represent the audit data. Heterogeneous data is normalized by different methods, e.g., [10]. Storing this processed audit data is also an important phase specially when dynamic cloud auditing generates enormous amount of data over time.

Auditing. In the auditing phase, processed data is verified against the policies for any violation. The process either validates the system or detects if any anomaly exists.

There are different auditing techniques proposed over time, though comparatively less automated techniques exist for the cloud. To understand better and to adapt other approaches, automated auditing methods in other analogous environments, such as intrusion detection systems and event correlation in multi-domain network/infrastructure, might be interesting. We consider different techniques of verifying policy compliance or detection of any policy violation including formal verification and validation (V&V) methods.

Audit Output. The proper representation of auditing output is the last and one of the important phases of security auditing. The audit report varies depending on the different demands and requirements of the customers (e.g., tenants). Hierarchy-based reporting helps to fulfill different levels of expectation. A major concern in outputting the result is not to leak any sensitive and unnecessary information to any tenant. Proper information isolation must be ensured.

3 Survey on Cloud Security Auditing

This section first categorizes the existing cloud security auditing, then elaborate each category mainly based on their coverage and adopted verification techniques and finally present a taxonomy based on these works.

There exist mainly three categories of cloud security auditing approaches. In the following, we discuss each of the approach with corresponding example works.

3.1 Retroactive Auditing

Retroactive auditing approach (e.g., [29, 32, 48, 50, 12, 47, 23, 11, 3]) in the cloud is a traditional way to verify the compliance of different components of a cloud. Works under this approach in the cloud targets a wide range of security properties that cover various cloud layers, such as data, user and virtual infrastructure.

There are several works that target auditing data location and storage in the cloud (e.g., [48, 23, 21, 50]). Wang et al. [48] propose a cloud storage system which enables privacy-friendly public auditing to ensure data security in the proposed system. The work leverages public key based homomorphic linear authenticator (HLA) to significantly reduce the communication and computation overhead at the auditor side. Kai et al. [23] can handle multiple auditing requests to verify the data integrity in the multi-cloud environment. In addition, similar as the former this work preserves the privacy of the audit data. On the other hand, Ismail et al. [21] propose a game theory based auditing approach to verify the compliance of data backup requirements of users. Unlike previous ones, Wang et al. [50] offer auditing of data origin and consistence in addition to data integrity.

There exist other works, which target virtual infrastructure change auditing (e.g., [12, 47, 11, 29, 32, 28]). These works cover different layers (e.g., user, virtual network, etc.) in the virtual infrastructure. Particularly, Ullah et al. [47] propose an architecture to build an automated security compliance tool for cloud computing platforms focusing on auditing clock synchronization and remote administrative & diagnostic port protection. Doelitzscher [11] proposes on-demand audit architecture for IaaS clouds and an implementation based on software agents to enable anomaly detection systems to identify anomalies in IaaS clouds for the purpose of auditing. The works in [47, 11] have the same general objective, which is cloud auditing, but they use empirical techniques to perform auditing whereas we use formal techniques to model and solve the auditing problem. Madi et al. [28, 29] verify a list of security properties to audit the cross-layer consistencies in the cloud.

In addition, several industrial efforts include solutions to support cloud auditing in specific cloud environments. For instance, Microsoft proposes SecGuru [3] to audit Azure datacenter network policy using the SMT solver Z3. IBM also provides a set of monitoring tool integrated with QRadar [20], which is their security information and event management system, to collect and analyze events in the cloud. Amazon is offering web API logs and metric data to their AWS clients by AWS CloudWatch & CloudTrail [2] that could be used for the auditing purpose. Although those efforts may potentially assist auditing tasks, none of them directly supports auditing a wide range of security properties covering authentication, authorization and virtual infrastructure on cloud standards.

Furthermore, there are several auditing solutions (e.g., [32, 16, 46, 17, 18]) targeting the user-level (e.g., authentication and authorization) of the cloud. Majumdar et al. [32] verify the role-based access control implementation in OpenStack, a popular cloud platform. This work also verifies a list of security properties to ensure proper implementation of authentication steps in the cloud. To accommodate the need of secure collaborative environments such as cloud computing, there have been some efforts towards proposing multi-domain/multi-tenant access control models (e.g., [16, 46, 17]). Gouglidis and Mavridis [17] leverage graph theory algorithms to verify a subset of the access control security properties. Gouglidis et al. [18] utilize model-checking to verify custom extensions of RBAC with multi-domains [17] against security properties. Lu et al. [26] use set theory to formalize policy conflicts in the context of inter-operation in the multi-domain environment.

3.2 Intercept-and-Check Auditing

Existing intercept-and-check approaches (e.g., [27, 5, 37, 19, 43, 45, 33]) perform major verification tasks while holding the event instances blocked. Works under this category cover the virtual network, user-level and software defined network (SDN) layers of a cloud environment as discussed in the following.

The works (e.g., [5, 37]) at the virtual network level are mainly verifying the security properties to safeguard multiple layers in a virtual network through an intercept-and-check approach. These works focus on operational network properties (e.g., black holes and forwarding loops) in virtual networks, whereas our effort is oriented toward preserving compliance with structural security properties that impact isolation in cloud virtualized infrastructures. Designing cloud monitoring services based on security service-level agreements have been discussed in [40].

The user-level runtime auditing is proposed in Patron [27] and Majumdar et al. [33]. more specifically, Patron [27] audits the access control rules defined by the cloud tenants. In addition, Patron enforces these rules on the cloud by leveraging the middleware supported in OpenStack, one of the major cloud platforms. Majumdar et al. [33] utilize similar interception approach in OpenStack and audit the proper deployment of various authentication and authorization plugins, such as single sign-on (SSO), role-based access control (RBAC) and attribute-based access control (ABAC) in the cloud.

There are also few works (e.g., TopoGuard [19] and TopoGuard+ [43]) which adopt the intercept-and-check approach in the software defined network (SDN) environment. TopoGuard [19] and TopoGuard+ [43] perform the interception and enforcement to prevent topology tempering attacks in SDN. Those works in SDN can be complements to the above-mentioned solutions for other layers in the cloud.

3.3 Proactive Auditing

The concept of proactive security auditing for clouds is different than the traditional security auditing concept. The first proactive auditing approach for clouds is proposed in [5]. Additionally, the Cloud Security Alliance (CSA) recommends continuous auditing as the highest level of auditing [8], from which latter works (e.g., [30, 31]) are inspired. The current proactive and runtime auditing mechanisms are more of a combination of traditional auditing and incident management. For example, LeaPS [31] learns from incidents and intercepted events to process or detect in a similar manner as a traditional incident management system. At the same time, LeaPS verifies and enforces compliance against different security properties, which are mostly taken from different security standards, and provide detailed evidence for any violation through LeaPS dashboard. Therefore, the concept of proactive security auditing is a combination of incident management and security auditing.

Proactive security analysis has also been explored for software security enforcement through monitoring programs' behaviors and taking specific actions (e.g., warning) in case security policies are violated. Many state-based formal models are proposed for those program monitors over the last two decades. First, Schneider [42] modelled program monitors using an infinite-state-automata model to enforce safety properties. Those automata recognize invalid behaviors and halt the target application before the violation occurs. Ligatti [24] builds on Schneider's model and defines a more general program monitors model based on the so called edit/security automata. Rather than just recognizing executions, edit automata-based monitors are able to suppress bad and/or insert new actions, transforming hence invalid executions into valid ones. Mandatory Result Automata (MRA) is another model proposed by Ligatti et al. [25, 13] that can transform both actions and results to valid ones. Narain [34] proactively generates correct network configurations using the model finder Alloy, which leverages a state of the art SAT solver. To this end, they specify a set of end-to-end requirements in First Order Logic and determine the set of existing network components. Alloy uses a state of the art SAT solver to provide the configurations that satisfy the input requirements for each network component. Considering the huge size of cloud environments and the tremendous space of possible events, adapting those solutions in the cloud is possibly very challenging.

Weatherman [5] is aiming at mitigating misconfigurations and enforcing security policies in a virtualized infrastructure. Weatherman has both online and offline approaches. Their online approach intercepts management operations for analysis, and relays them to the management hosts only if Weatherman confirms no security violation caused by those operations. Otherwise, they are rejected with an error signal to the requester. The work defines a realization model, that captures the virtualized infrastructure configuration and topology in a graph-based model. The latter is synchronized with the actual infrastructure using the approach in [4]. Two major limitations of this proposition are: i) the model capturing the whole infrastructure causes a scalability issue for the solution, and ii) the time consuming operation-checking that should be performed on the emergence of each event, makes security enforcement not feasible for large size data centers. Congress [37] is an OpenStack project offering both online and offline policy enforcement approaches. The offline approach requires submitting a future change plan to Congress, so that the changes can be simulated and the impacts of those changes can be verified against specific properties. In the online approach, Congress first applies the operation to the cloud, then checks its impacts. In case of a violation, the operation is reverted. However, the time elapsed before reverting the operation can be critical to perform some illicit actions, for instance, transferring sensitive files before loosing the assigned role. Foley et al. [15] provide an algebra to assess the effect of security policies replacement and composition in OpenStack. Their solution can be considered as a proactive approach for checking operational property violations.

3.4 Taxonomy of Cloud Security Auditing

Based on the above-mentioned study on cloud security auditing, we devise a primary taxonomy for these works (as in Figure 2). We consider the whole landscape from the perspective of their coverage and applied techniques. Therefore, we first categorize them based on their targeted cloud layers (e.g., data, user, virtual network and SDN), then further identify various high-level security properties that these works support, and finally show their adopted approaches. Thus, it is trivial to understand which approaches are already explored for certain security problems under a particular cloud layer. Furthermore, our taxonomy can be useful towards building a fine-grained classification of cloud security auditing approaches.

3.5 Comparative Study

We conduct a comparative study based on the taxonomy presented in the previous section. Table 1 summarizes the findings of this study. The first and second columns of the table enlist existing works and their verification methods. The next four columns present their covered layers in the cloud. We mainly include works on four cloud layers: data, user, virtual network and software defined network (SDN). In next three columns, we show the approaches (retroactive, intercept-and-check and proactive) that a work adopts. Afterwards, there are five features enlisted to demonstrate the special skills of these works. The caching feature is marked when a work enables caching of verification results to enhance the efficiency of the auditing process. We mark the dependency model when a work utilizes the dependency relationship in the cloud to improve the efficiency and accuracy of the auditing process. The pre-computation step is to identify the works which performs a significant part of the verification step in advance to reduce the response time of the runtime (usually in intercept-and-check and proactive) solutions.

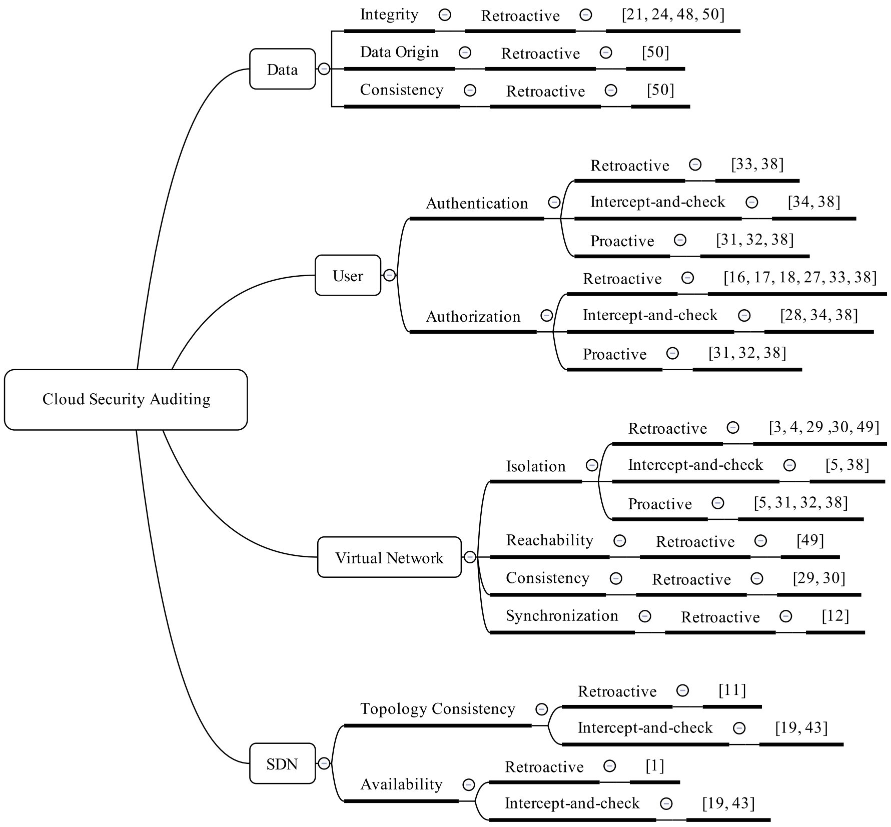

Fig. 2: A taxonomy of cloud security auditing

There exist few works which support auditing of multiple requests together. For them, we mark the batch auditing feature. The active auditing feature is an active-probing-based auditing solution which does not fully rely on the cloud provider for the audit data and instead actively participate in the targeted protocol to verify certain properties. The next four columns indicate the supporting cloud platforms for these auditing solutions. We mark the adaptable to others column when a work provide detailed discussion on the process of porting the solution to different platforms. In the last four columns, we evaluate existing works based on the commonly observed constraints in the field of cloud security auditing. The after-the-fact constraint is marked if a work cannot prevent a security violation. The prohibitive delay is checked when a runtime work (i.e., intercept-and-check and proactive approaches) causes significant delay in responding to a user request. For retroactive solutions, we mark this column as not applicable (N/A). If a work involves significant manual effort (apart from the inputs from the users) in the auditing process, then we check the manual effort constraint. The limited coverage constraint is defined based on the expressiveness of a auditing solution. For instance, a work supporting first order logic to define security properties does not suffer from this constraint.

The key observations of this comparative study are as follows. First, there is no single auditing solution to verify multiple layers of the cloud. Therefore, today's cloud tenants require at least three different solutions to fulfill their auditing need; which might not be very usable for the tenants. Second, even though intercept-and-check approach is designed to prevent security violations, existing works under this category are not practical due to their prohibitive delay. Third, the proactive auditing approach is a promising solution to overcome the limitations of both retroactive and intercept-and-check approach. However, this approach still suffers from several practical issues, such as relying on manual efforts and limiting the expressiveness of security properties. Finally, there exist several features in the wild which significantly can improve the efficiency and accuracy of the auditing solution. However, there is a need of a unified solution with all these features at least to overcome major constraints.

4 Challenges in Cloud Security Auditing

In the following, we discuss the key challenges in cloud security auditing that we identified during our survey.

High-level Security Properties. There is a significant gap between the high-level standards (defining security requirements) and the low-level cloud configurations (providing the audit data). Even though several works (e.g., [32, 29, 28] highlight this challenge and partially address the concern, the issue still persists in interpreting security guidelines and defining security properties ready to be used in auditing solutions. Current solutions rely on manual identification of security properties, which is infeasible and error-prone especially when we consider the multiple layers of cloud and intend to provide a unified security solution for the whole cloud.

Non-Trivial Log Processing. One mandatory and non-trivial step of cloud security auditing is log processing. This step involves several challenging tasks. First, identifying the heterogeneous sources of audit data requires well realization of the deployed cloud system, which usually consists of several complex components, e.g., management platform and layer-2 plugins. Second, due to the different nature (e.g., database and text files) of storing the configurations and logs, the collection of audit data has to be performed by adopting multiple methods. Finally, the diverse format of the logs require extensive processing efforts to uniform the format before using them in auditing.

Reducing Manual Involvement. Automating the auditing process is a must in a dynamic environment like cloud to ensure the accuracy and efficiency. However, the current solutions still rely on manual efforts in several critical steps. Fully eliminating or at least reducing manual effort is not trivial mainly for the following two reasons. First, defining the security properties is a mandatory step for any auditing process and we fully rely on human inputs for this step. Existing rule mining techniques in access control might be useful in automating this step. Second, all intercept-and-check and proactive approaches (as reported in Section 4) rely on manual identification of critical operations (which potentially can violate a property). Applying machine learning

Proposals	Methods	Layers				Approaches			Features					Platforms				Constraints			
		Data	User	Virtual Net.	SDN	Retroactive	Intercept-and-Check	Proactive	Caching	Dependency Model	Pre-Computation	Batch Auditing	Active Auditing	Supporting OpenStack	Supporting Azure	Supporting VMware	Adaptable to others	After-the-fact	Prohibitive Delay	Manual Effort	Limited Coverage
Wang et al. [48]	Cryptographic	●	-	-	-	●	-	-	-	-	-	●	-	-	-	-	●	●	N/A	-	-
Kai et al. [23]	Cryptographic	●	-	-	-	●	-	-	-	-	-	●	-	-	-	-	●	●	N/A	-	-
Doelitzscher et al. [12]	Custom Algorithm	-	-	●	-	●	-	-	-	-	-	-	-	●	-	-	●	●	N/A	-	-
Ullah et al. [47]	Custom Algorithm	-	-	●	-	●	-	-	-	-	-	-	-	●	-	-	-	●	N/A	-	-
Solanas et al. [44]	Classifiers	-	●	-	-	●	-	-	-	-	-	-	-	●	-	-	-	●	N/A	-	-
Majumdar et al. [32]	CSP	-	●	-	-	●	-	-	-	-	-	-	-	●	-	-	-	●	N/A	-	-
Madi et al. [29, 28]	CSP	-	-	●	-	●	-	-	-	-	-	-	-	●	-	-	-	●	N/A	-	-
Cloud Radar [4]	Graph Theory	-	-	●	-	●	-	-	-	-	-	-	-	-	-	●	-	●	N/A	-	-
TenantGuard [49]	Graph Theory	-	-	●	-	●	-	-	-	-	-	-	-	●	-	-	-	●	N/A	-	-
SecGuru [3]	SMT	-	-	●	-	●	-	-	-	-	-	-	-	-	●	-	-	●	N/A	-	-
QRadar [20]	Custom	-	-	●	-	●	-	-	-	-	-	-	-	-	-	●	-	●	N/A	-	-
SPV [1]	Custom	-	-	-	●	●	-	-	-	-	-	-	●	●	-	-	●	●	N/A	-	-
Patron [27]	Custom Algorithm	-	●	-	-	-	●	-	-	●	-	-	-	●	-	-	-	-	●	●	-
Weatherman (V1) [5]	Graph Theory	-	-	●	-	-	●	-	-	-	-	-	-	-	-	●	-	-	●	●	●
Congress (V1) [37]	Datalog	-	●	●	-	-	●	-	-	-	-	-	-	●	-	-	-	-	●	●	-
TopoGuard [19, 43]	Custom	-	-	-	●	-	●	-	-	-	-	-	-	-	-	-	●	-	●	-	●
Majumdar et al. [33]	CSP + Custom	-	●	-	-	-	●	-	-	-	-	-	-	●	-	-	●	-	●	●	-
Weatherman (V2) [5]	Graph Theory	-	-	●	-	-	-	●	-	-	●	-	-	-	-	●	-	-	-	●	●
Congress (V2) [37]	Datalog	-	●	●	-	-	-	●	-	-	●	-	-	●	-	-	-	-	-	●	●
PVSC [30]	Custom Algorithm	-	●	●	-	-	-	●	●	●	●	-	-	●	-	-	●	-	-	●	●
LeaPS [31]	Custom + Bayesian	-	●	●	-	-	-	●	-	●	●	-	-	●	-	-	●	-	-	●	●

Table 1: Summary of existing cloud security auditing solutions highlighting their adopted methods, covered cloud layers, applied approaches, offered features, supported platforms and constraints. The symbols (●), (-) and N/A mean supported/required, not supported/required, and not applicable, respectively. Note that, for both Weatherman and Congress, V1 and V2 refer to their proactive and intercept-and-check variants, respectively.

or more specifically interactive machine learning techniques may reduce the manual efforts involved with this step.

Unified Auditing Solution for Multi-Layer Clouds. Table 1 pinpoints that a tenant requires at least three auditing solutions if s/he wants to verify all four layers of her/his cloud, and there is a need of unified auditing solution supporting multi-layer of a cloud. However, to propose a unified solution is non-trivial for the following fact. First, each layer of the cloud contains unique auditing requirements (e.g., audit data type and security properties). Second, there exist security threats involving multi-layer (as reported in [28]); which currently being ignored in the solutions dedicated for a single layer. Finally, it is very difficult to be comprehensive in covering security properties from various layers. However, we believe that this is a more generic problem in the field of auditing and require more attention from the researchers to overcome the concern.

Privacy Concerns in Audit Inputs and Outputs. Both third party and cross-tenant auditing raise privacy concerns resulting from both audit inputs and outputs. In addressing these privacy concerns, there exist at least two major challenges. First, how we can preserve tenants' privacy in the input data so that the utility (i.e., auditing capability) is not much affected. Second, how to hide cross-tenant sensitive information so that the usefulness of auditing output remains unchanged.

5 Discussion

In the following, we discuss several important aspects of cloud security auditing.

Why Traditional Auditing is not Enough for the Cloud. Based on the previously discussed cloud security issues, it is obvious that traditional security auditing techniques are not enough to be directly applied to the cloud. The two most existing on-premise IT models are IT housing and IT outsourcing [11]. In IT housing, it belongs to the customer to provide and manage his own hardware. The datacenter provider just provides the remaining facilities such as network components, cooling and power. Traditional IT outsourcing is generally a medium to long term contract. In the latter, the customer rents all the infrastructure components from the service provider. The rented infrastructure is exclusively used by one customer which is called the single-tenant model. A prior communication with the provider is required whenever any modification is to be applied to the rented infrastructures. In these two models, the IT organization has full governance over the different IT technology layers. In the cloud, however, as we move from IaaS to PaaS to SaaS, the level of control of cloud providers increases and the burden of access, control, management and the infrastructure's trust boundaries is considerably shifted to the cloud provider and responsibilities become more or less shared between the latter and its customers, which raises trust issues between the two parties.

How Cloud Auditing Helps Mitigating Security Issues. In a cloud environment, though asymmetric, trust needs to be boosted in both directions. Most importantly, the potential customer needs to trust the cloud provider in order to feel comfortable when outsourcing his assets. The other way around, the cloud provider needs as well to gain some assurance that the customer will benefit from the offered services in a honest way and does not use it for cybercrime, but at the same time, the provider is supposed to immune his services against malicious insiders. Although trust plays a vital role in the cloud ecosystems, it should be further boosted with other tools. With this regard auditing is a good fit to increase the confidence of different stakeholders. Continuous auditing allows to analyze the service conditions and the infrastructure health through detailed log records to access conformity between security measures and policies. Although it seems that cloud providers might not be willing to allow for auditing tasks, they actually should have their own incentives. In effect, auditing helps reducing the scope of search and identifying responsible parties in case of incidents or legal actions which, in some cases, can exonerate the provider and prevent him considerable money loss. For instance, the cloud security alliance recommends auditing different critical components of a cloud including privileged user access, regulatory compliance, isolation, tenant segregation, monitoring, and data storage and processing

6 Conclusion

Cloud computing has seen a lot of interests and adoption lately. Nonetheless, the widespread adoption of cloud is still being hindered by the lack of transparency and accountability, which has traditionally been ensured through security compliance auditing techniques. In this paper, we conducted a survey on the existing cloud security auditing approaches. To this end, we first categorized the existing solutions and elaborate each category with example works. Second, we proposed a taxonomy identifying the classifications mainly based on auditing objectives and auditing techniques. Third, we conducted a comparative study on these works to identify the strengths and weaknesses of these works. Finally, we identified current challenges in cloud security auditing; which potentially may draw the attention of security researchers. However, there are few limitations of this work which we intend to overcome in our future work. For instance, we plan to increase the granularity of the proposed taxonomy to pinpoint more precisely the gaps in cloud security auditing. In addition to qualitative comparison presented in this paper, we intend to compare existing works quantitatively to understand better how to improve the efficiency and accuracy of these approaches.

Acknowledgement. The authors thank the anonymous reviewers for their valuable comments. This work is partially supported by the Natural Sciences and Engineering Research Council of Canada and Ericsson Canada under CRD Grant N01823 and by PROMPT Quebec.

References

1. A. Alimohammadifar, S. Majumdar, T. Madi, Y. Jarraya, M. Pourzandi, L. Wang, and M. Debbabi. Stealthy probing-based verification (spv): An active approach to defending software defined networks against topology poisoning attacks. In *European Symposium on Research in Computer Security*, pages 463–484. Springer, 2018.
2. Amazon Web Services. Security at scale: Logging in AWS. Technical report, Amazon, 2013.
3. N. Bjørner and K. Jayaraman. Checking cloud contracts in Microsoft Azure. In *Distributed Computing and Internet Technology*. Springer, 2015.

4. S. Bleikertz, C. Vogel, and T. Groß. Cloud Radar: near real-time detection of security failures in dynamic virtualized infrastructures. In *Proceedings of the 30th annual computer security applications conference (ACSAC)*, pages 26–35. ACM, 2014.
5. S. Bleikertz, C. Vogel, T. Groß, and S. Mödersheim. Proactive security analysis of changes in virtualized infrastructures. In *Proceedings of the 31st annual computer security applications conference (ACSAC)*, pages 51–60. ACM, 2015.
6. Cloud Security Alliance. Security guidance for critical areas of focus in cloud computing v 3.0, 2011.
7. Cloud Security Alliance. Cloud control matrix CCM v3.0.1, 2014. Available at: `https://cloudsecurityalliance.org/research/ccm/`, last accessed on: February 14, 2018.
8. Cloud Security Alliance. CSA STAR program and open certification framework in 2016 and beyond, 2016. `https://downloads.cloudsecurityalliance.org/star/csa-star-program-cert-prep.pdf`, last accessed on: February 14, 2018.
9. CUMULUS. Certification infrastructure for multi-layer cloud services project (cumulus). *EU project*, 2012.
10. Distributed Management Task Force, INC. Cloud auditing data federation, 2016. `https://www.dmtf.org/standards/cadf`.
11. F. Doelitzscher. *Security Audit Compliance for Cloud Computing*. PhD thesis, Plymouth University, 2014.
12. F. Doelitzscher, C. Fischer, D. Moskal, C. Reich, M. Knahl, and N. Clarke. Validating cloud infrastructure changes by cloud audits. In *Eighth World Congress on Services (SERVICES)*, pages 377–384. IEEE, 2012.
13. E. Dolzhenko, J. Ligatti, and S. Reddy. Modeling runtime enforcement with mandatory results automata. *International Journal of Information Security*, 14(1):47–60, 2015.
14. ENISA. European union agency for network and information security, 2016. `https://www.enisa.europa.eu`.
15. S. N. Foley and U. Neville. A firewall algebra for OpenStack. In *Conference on Communications and Network Security (CNS)*, pages 541–549. IEEE, 2015.
16. N. Ghosh, D. Chatterjee, S. K. Ghosh, and S. K. Das. Securing loosely-coupled collaboration in cloud environment through dynamic detection and removal of access conflicts. *IEEE Trans. on Cloud Comp.*, 2014.
17. A. Gouglidis and I. Mavridis. domRBAC: An access control model for modern collaborative systems. *Computers & Security*, 2012.
18. A. Gouglidis, I. Mavridis, and V. C. Hu. Security policy verification for multi-domains in cloud systems. *Int. Jour. of Info. Sec.*, 2014. 13(2).
19. S. Hong, L. Xu, H. Wang, and G. Gu. Poisoning network visibility in software-defined networks: New attacks and countermeasures. In *Proceedings of 2015 Annual Network and Distributed System Security Symposium (NDSS'15)*, February 2015.
20. IBM. Safeguarding the cloud with IBM security solutions. Technical report, IBM Corporation, 2013.
21. Z. Ismail, C. Kiennert, J. Leneutre, and L. Chen. Auditing a cloud provider's compliance with data backup requirements: A game theoretical analysis. *IEEE Transactions on Information Forensics and Security*, 11(8):1685–1699, 2016.
22. ISO Std IEC. ISO 27017. *Information technology- Security techniques- Code of practice for information security controls based on ISO/IEC 27002 for cloud services (DRAFT)*, 2012. Available at: `http://www.iso27001security.com/html/27017.html`, last accessed on: February 14, 2018.
23. H. Kai, H. Chuanhe, W. Jinhai, Z. Hao, C. Xi, L. Yilong, Z. Lianzhen, and W. Bin. An efficient public batch auditing protocol for data security in multi-cloud storage. In *8th ChinaGrid Annual Conference (ChinaGrid)*, pages 51–56. IEEE, 2013.

24. J. Ligatti, L. Bauer, and D. Walker. Run-time enforcement of nonsafety policies. *ACM Transactions on Information and System Security (TISSEC)*, 12(3):19, 2009.
25. J. Ligatti and S. Reddy. A theory of runtime enforcement, with results. In *European Symposium on Research in Computer Security (ESORICS)*, pages 87–100. Springer, 2010.
26. Z. Lu, Z. Wen, Z. Tang, and R. Li. Resolution for conflicts of inter-operation in multi-domain environment. *Wuhan University Journal of Natural Sciences*, 12(5), 2007.
27. 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.
28. T. Madi, Y. Jarraya, A. Alimohammadifar, S. Majumdar, Y. Wang, M. Pourzandi, L. Wang, and M. Debbabi. ISOTOP: Auditing virtual networks isolation across cloud layers in OpenStack. *ACM Transactions on Privacy and Security (TOPS)*, pages xx–xx, 2018.
29. 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 *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy (CODASPY)*, pages 195–206. ACM, 2016.
30. S. Majumdar, Y. Jarraya, T. Madi, A. Alimohammadifar, M. Pourzandi, L. Wang, and M. Debbabi. Proactive verification of security compliance for clouds through precomputation: Application to OpenStack. In *European Symposium on Research in Computer Security (ESORICS)*, pages 47–66. Springer, 2016.
31. S. Majumdar, Y. Jarraya, M. Oqaily, A. Alimohammadifar, M. Pourzandi, L. Wang, and M. Debbabi. Leaps: Learning-based proactive security auditing for clouds. In *European Symposium on Research in Computer Security (ESORICS)*, pages 265–285. Springer, 2017.
32. 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 *7th International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 58–65. IEEE, 2015.
33. S. Majumdar, T. Madi, Y. Wang, Y. Jarraya, M. Pourzandi, L. Wang, and M. Debbabi. User-level runtime security auditing for the cloud. *IEEE Transactions on Information Forensics and Security*, 13(5):1185–1199, 2018.
34. S. Narain. Network configuration management via model finding. In *Proceedings of the 19th Conference on Large Installation System Administration Conference (LISA)*, pages 15–15, 2005.
35. NIST. SP 800-53. *Recommended Security Controls for Federal Information Systems*, 2003.
36. Open Data Center Alliance. Open data center alliance usage: Cloud based identity governance and auditing rev. 1.0. Technical report, Open Data Center Alliance, 2012.
37. OpenStack. OpenStack Congress, 2015. Available at: `https://wiki.openstack.org/wiki/Congress`, last accessed on: February 14, 2018.
38. OpenStack. OpenStack open source cloud computing software, 2015. Available at: `http://www.openstack.org`, last accessed on: February 14, 2018.
39. OpenStack. OpenStack user survey, 2016. Available at: `https://www.openstack.org/assets/survey/October2016SurveyReport.pdf`, last accessed on: Feb 14, 2018.
40. D. Petcu and C. Craciun. Towards a security SLA-based cloud monitoring service. In *Proceedings of the 4th International Conference on Cloud Computing and Services Science (CLOSER)*, pages 598–603, 2014.
41. K. Ren, C. Wang, and Q. Wang. Security challenges for the public cloud. *IEEE Internet Computing*, 16(1):69–73, 2012.
42. F. B. Schneider. Enforceable security policies. *Transactions on Information and System Security (TISSEC)*, 3(1):30–50, 2000.

43. R. Skowyra, L. Xu, G. Gu, T. Hobson, V. Dedhia, J. Landry, and H. Okhravi. Effective topology tampering attacks and defenses in software-defined networks. In *Proceedings of the 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'18)*, June 2018.
44. M. Solanas, J. Hernandez-Castro, and D. Dutta. Detecting fraudulent activity in a cloud using privacy-friendly data aggregates. Technical report, arXiv preprint, 2014.
45. A. Tabiban, S. Majumdar, L. Wang, and M. Debbabi. Permon: An openstack middleware for runtime security policy enforcement in clouds. In *Proceedings of the 4th IEEE Workshop on Security and Privacy in the Cloud (SPC 2018)*, June 2018.
46. B. Tang and R. Sandhu. Extending openstack access control with domain trust. In *Network and System Security*, pages 54–69. Springer, 2014.
47. K. W. Ullah, A. S. Ahmed, and J. Ylitalo. Towards building an automated security compliance tool for the cloud. In *12th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1587–1593. IEEE, 2013.
48. C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou. Privacy-preserving public auditing for secure cloud storage. *IEEE transactions on computers*, 62(2):362–375, 2013.
49. Y. Wang, T. Madi, S. Majumdar, , Y. Jarraya, M. Pourzandi, L. Wang, and M. Debbabi. Tenantguard: Scalable runtime verification of cloud-wide vm-level network isolation. In *Proceedings of 2017 Annual Network and Distributed System Security Symposium (NDSS'17)*, February 2017.
50. Y. Wang, Q. Wu, B. Qin, W. Shi, R. H. Deng, and J. Hu. Identity-based data outsourcing with comprehensive auditing in clouds. *IEEE Transactions on Information Forensics and Security*, 12(4):940–952, 2017.
51. S. S. Yau, A. B. Buduru, and V. Nagaraja. Protecting critical cloud infrastructures with predictive capability. In *8th International Conference on Cloud Computing (CLOUD)*, pages 1119–1124. IEEE, 2015.