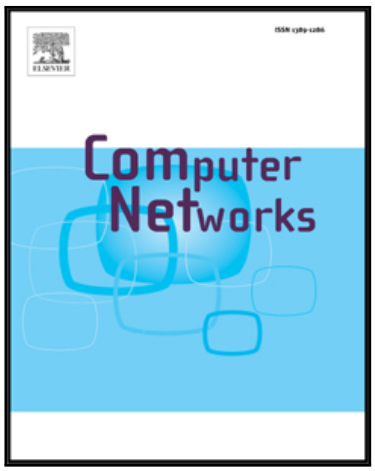

Systematic Identification of Threats in the Cloud: A Survey

Jin B. Hong, Armstrong Nhlabatsi, Dong Seong Kim, Alaa Hussein, Noora Fetais, Khaled M. Khan

PII: S1389-1286(18)30825-9
DOI: https://doi.org/10.1016/j.comnet.2018.12.009
Reference: COMPNW 6674

To appear in: *Computer Networks*

Received date: 29 August 2018
Revised date: 16 November 2018
Accepted date: 10 December 2018

Please cite this article as: Jin B. Hong, Armstrong Nhlabatsi, Dong Seong Kim, Alaa Hussein, Noora Fetais, Khaled M. Khan, Systematic Identification of Threats in the Cloud: A Survey, *Computer Networks* (2018), doi: https://doi.org/10.1016/j.comnet.2018.12.009

Systematic Identification of Threats in the Cloud: A Survey

Jin B. Hong[a], Armstrong Nhlabatsi[c], Dong Seong Kim[b], Alaa Hussein[c], Noora Fetais[c], Khaled M. Khan[c]

[a]*Department of Computer Science and Software Engineering, University of Western Australia, Australia*
[b]*School of Information Technology and Electrical Engineering, University of Queensland, Australia*
[c]*Department of Computer Science and Engineering, KINDI Computing Lab, Qatar University, Qatar*

Abstract

When a vulnerability is discovered in a system, some key questions often asked by the security analyst are what threat(s) does it pose, what attacks may exploit it, and which parts of the system it affects. Answers to those questions provide the necessary information for the security assessment and to implement effective countermeasures. In the cloud, this problem is more challenging due to the dynamic characteristics, such as elasticity, virtualization, and migration - changing the attack surface over time. This survey explores threats to the cloud by investigating the linkages between threats, attacks and vulnerabilities, and propose a method to identify threats systematically in the cloud using the threat classifications. First, we trace vulnerabilities to threats by relating vulnerabilities-to-attacks, and then relating attacks-to-threats. We have established the traceability through an extensive literature review and synthesis that resulted in a classification of attacks in the cloud, where we use the Microsoft STRIDE threat modeling approach as a guide for relating attacks to threats. Our approach is the genesis towards a concrete method for systematically identifying potential threats to assets provisioned and managed through the cloud. We demonstrate the approach through its application using a cloud deployment case study scenario.

Keywords: Attack Classification; Cloud Computing; Threats Classification; Threat

Email addresses: jin.hong@uwa.edu.au (Jin B. Hong), armstrong.nhlabatsi@qu.edu.qa (Armstrong Nhlabatsi), dongseong.kim@canterbury.ac.nz (Dong Seong Kim), a.hussein@qu.edu.qa (Alaa Hussein), n.almarri@qu.edu.qa (Noora Fetais), k.khan@qu.edu.qa (Khaled M. Khan)

Identification; Vulnerabilities.

1. Introduction

Cloud computing, an advanced networking technology, enables a better use of services and resource utilization, at a reduced operational cost (Mell and Grance, 2011). As a result, many stakeholders, such as enterprises, governments, and individuals, are shifting their current networking platform to the cloud. The cloud is delivered through three basic service models: Software-as-a-Service (SaaS); Platform-as-a-Service (PaaS); and Infrastructure-as-a-Service (IaaS) (Liu et al., 2011b). The service models determine the level of customization available for the cloud users. For example, virtual machines (VMs) can be configured by users (referred to as *tenants*) with the IaaS model, whereas the network administrator will configure the VMs in the SaaS or PaaS models. Many business models implementing the cloud technology concept are available (Alcatel-Lucent, 2018; Amazon, 2018), providing the users with the capabilities to control the network more efficiently in order to aid the business operations to its fullest while minimizing the cost.

1.1. Research Problem

Although the cloud computing technology promises more robust and dynamic networking infrastructure, it also suffers from various security issues arising from the existing and new threats (Chen et al., 2010; Vaquero et al., 2011; Coppolino et al., 2017; Pham et al., 2014). Here, we refer the term *threat* to any events that could potentially result in the unintended behavior of the cloud that could ultimately lead to the damage of assets. The term *attack* refers to the act of performing (or carrying-out) a threat event (or a series of threat events). In particular, we focus on intentional events that pose the threats (e.g., Denial of Service (Yan et al., 2016) attacks on a credit card payment service hosted on the cloud which can render it unavailable for processing legitimate transactions). The term *vulnerability* refers to security holes or weaknesses in a cloud deployment that can be leveraged to launch attack events.

Attacks thrive on the existence of *vulnerabilities*. As such, discovering vulnerabilities is a critical and fundamental step in the security analysis (Varadharajan and Tupakula, 2017). Once a vulnerability is discovered, some key challenges which security analysts often face are finding out what threat(s) does it pose, what attacks may exploit it, and which parts of the system are affected by it. There are different solutions proposed to address various security issues in the cloud (Vieira et al., 2010; Houmansadr et al., 2011; Hong et al., 2014; Chung et al., 2013), but there are no

systematic means for security administrators to identify and address possible threats in the cloud. Therefore, it is of paramount importance to provide systematic means to classify and identify the threats in the cloud. In order to do this, one must be able to (1) identify threats, (2) identify attacks that materialize threats, (3) identify vulnerabilities in the cloud computing resources and components, and (4) link vulnerabilities that are exploited by attacks and the posed threats. The key research problem is to combine these techniques and methods for the threat classification and identification process.

1.2. Importance of the Problem

The classification and identification of the threats in the cloud is a necessary step in the process of assessing the security risk to assets managed and provisioned through the cloud. A common method used for a risk assessment involves knowing the *likelihood* and the *impact* of an attack (Naskos et al., 2016; Rezvani et al., 2015). The likelihood, which can also be perceived as a probability, measures how many attacks are likely to succeed. On the other hand, the impact is a measure of the amount of damage to an asset that would result from a successful attack. While the probability of attack success can be mainly based on a statistical analysis of previous attacks (Dudorov et al., 2013), the impact is linked to the threats on the asset, which varies depending on the threat that is the focus of a damage assessment. As the cloud also yields certain threats, the key security questions that many cloud users/clients/providers *etc.* would ask; (1) what are the threats, (2) how can we assess those threats (severity), and (3) how can we mitigate them. As outlined in the research problem above, there are various solutions to address each step towards answering those key questions. However, such information must be consolidated in order to fully answer them.

For example, a credit card company may care more about the threat of *tampering* with transactions, as they are transmitted from point-of-sale terminals to payment authorization servers, in comparison to the threat of a *information disclosure* via a man-in-middle attack. As such, a security administrator in the credit card company may give the information disclosure threat a lower impact rating compared to the tampering threat. Classification provides a basis for rating the impact of threats comparatively. As a result, different categories of the threats, with different impact ratings, may result in different risk values. The impact of an attack resulting from any of the new or previously unknown threats can be determined by allocating it into one of the known threats categories defined through the threat classification. A byproduct of the classification is that it makes it possible to establish the concrete relationship between vulnerabilities, attacks, and threats - thus enabling identifica-

tion of threats by tracing them back to attacks and vulnerabilities. Therefore, the classification and identification of the threats in the cloud is an important step in the evaluation of security risk to assets.

1.3. Limitations of Existing Work

Many studies have been conducted to address various threats to the cloud (Varadharajan and Tupakula, 2012; Somorovsky et al., 2011; Chapade et al., 2013b; Zhang and Reiter, 2013), but they do not provide a detailed understanding of how those threats are posed (i.e., what classes of threats affect which components in the cloud, and which vulnerabilities lead to those threats). Different studies have addressed specific threats. For example, Varadharajan and Tupakula (Varadharajan and Tupakula, 2012) proposed an approach for certification of virtual machines (VMs) belonging to different tenants by the cloud provider. The VMs that exhibit behaviors that deviate from certified properties are terminated or isolated by the cloud provider. The work does not offer any insight into what vulnerabilities in the VMs could be exploited by what kind of attacks that would lead to the threat being addressed. Chapade *et al.* (Chapade et al., 2013b) proposed an approach towards protection against the flooding-based DDoS attacks. Again, they did not address the issue of how the vulnerabilities can be systematically tracked to attacks, and how the attacks could be linked to threats.

Such an understanding is important for providing effective security solutions (i.e., where do those threats arise from and what is their impact on the cloud?). Hence, there is a need for the classification of threats in relation to their possible attack vectors to understand how each threat can be imposed and its associated impact on the cloud. There are various modeling methods to classify threats (Saini et al., 2008; Vidalis and Jones, 2003; Singh et al., 2014; Cheng et al., 2012), but they only provide an abstract view without considering the concrete, low-level details at the cloud component level. As a result, these models cannot capture the impact of different attacks at the cloud component level to associate with their corresponding threats. It is important to know what components of the cloud are affected by what attacks (and eventually threats) as this can inform a better design and selection of their countermeasures and mitigation strategies.

1.4. Proposed Approach

This survey paper aims to fill the existing gaps in knowledge about the classification and systematic identification of the threats in the cloud. Our approach is to use the threat categorization model to classify threats with respect to various attack categories and the cloud components. We use the *STRIDE* threat model (LeBlanc

and Howard, 2002) for categorizing the threats as an illustration of our approach. We chose the STRIDE threat model because it categorizes the threats according to the consequences of their realization such as "corruption of information, the disclosure of information, denial of service, and elevation of privileges" (Xu et al., 2012). Consequences of threats can be directly linked to the impact of their realization as such consequences are important in the assessment of the security risk in the cloud (Rezvani et al., 2015; Naskos et al., 2016; Almutairi and Ghafoor, 2014; Almutairi et al., 2018). However, other threat categorization models can be used in our approach. The STRIDE threat model has been used previously for the cloud (Saripalli and Walters, 2010; Deng et al., 2011), but they only considered the high-level threat evaluation. We go beyond this by identifying the threats at the low-level of the cloud in order to trace threats to various cloud components. Furthermore, we propose to identify the threats in the cloud by (i) specifying the cloud components and their associated vulnerabilities, (ii) identifying the attacks that could exploit the discovered vulnerabilities, and (iii) identifying the threats posed by each attack applicable to the cloud. Our approach allows a high-level overview of the cloud threats, as well as capturing the details of threat realization at the cloud component level.

We first present related work on threat identifications and classifications in the cloud, and identify the limitations of the existing work in this area. To understand threats at the lower levels of the cloud, we present a brief overview of the cloud computing architecture, components and its deployment models. Then, we specify the research methodology for the threat classification and identification in the cloud, an overview of our approach to classify and identify the threats in the cloud. The details of threat classification using the STRIDE threat model and mapping to the attack categories based on OWASP model is described, followed by the threat identification method via tracing in the cloud using the threat, attack and vulnerability mapping. Finally, we demonstrate the threat identification method using an example cloud computing scenario.

1.5. Contributions

To the best of our knowledge, this is the first work to propose a mechanism for systematically identifying the threats in the cloud by providing a two-step solution for tracing vulnerabilities to threats. First, we conduct a comprehensive literature survey to classify the threats in the cloud using the STRIDE threat model. This classification enables us to relate attacks and threats as well as cloud components that are susceptible to those attacks. Second, we propose a method for identifying the threats in the cloud. The novel and key aspect of our threat identification method is a way of associating vulnerabilities with the attacks that can exploit them. Once this

association is established we then use the relationship between attacks and threats to relate a vulnerability to a threat and hence identify the threat resulting from a vulnerability. The contributions of this paper are summarized as follows.

- Providing a comprehensive survey on attacks targeting the cloud and their mitigation;
- Developing a three-way relationship between threats, attacks and vulnerabilities to trace threats to the cloud via mapping;
- Providing a systematic threat classification and identification in the cloud via tracing threats;
- Demonstration of the threat identification using the example cloud.

The rest of the paper is structured as follows. Section 2 presents the related work and Section 3 describes the cloud computing architecture and its functionalities and section 4 outlines the methodology we used in conducting our survey. Section 5 presents the cloud threats, attacks, and their classifications. Section 6 presents our proposed approach to the identification of the threats in the cloud, and Section 7 discusses the current and future directions of threat identification and classification of the cloud. Finally, Section 8 concludes the paper.

2. Related Work

The problem of identifying the threats in the cloud is not new. It has been investigated in several studies(Dahbur et al., 2011; Shaikh and Haider, 2011; Modi et al., 2013a; Khorshed et al., 2012; d. Silva et al., 2013; Xiao and Xiao, 2013; Irfan et al., 2015; Bindra et al., 2012). In this section, we review some of the key surveys on the threats in the cloud and compare them to the approach we have taken in our study.

2.1. Threat Classification

Similar to our approach, Silva *et al.* (d. Silva et al., 2013) surveyed research efforts on publications available in the literature which address seven security threats. While we focused on the distribution of publications by cloud component, STRIDE, and attack type, they studied the distribution of the literature with respect to security domains. In addition to research efforts, we also collected and analyzed data on actual attack incidents reported. We envisage that having both the research and attack incidents perspective would allow us to draw better conclusions on the correlation

between the efforts in cloud security research and practice, and whether the research efforts are resulting in practical approaches that are mitigating security threats in an effective way.

Xiao and Xiao (Xiao and Xiao, 2013) argued that "confidentiality, integrity, availability, accountability, privacy-preservability" (Fernandes et al., 2014) are the key attributes for security and privacy. They presented vulnerabilities associated with these attributes and attack models of how the vulnerabilities can be exploited in a cloud computing scenario. The cloud attacks they identified are similar to the ones we identified in our study. However, they classified the attacks with respect to security and privacy objectives/goals. While focusing on security objectives is useful in security analysis, such classification does not give an insight on the categories of threats that may exploit those vulnerabilities to realize attacks. Moreover, by associating attacks with the cloud components where the attacks may be successfully executed, our survey has proposed a method for systematically identifying the threats in the cloud. Existing surveys (Dahbur et al., 2011; Shaikh and Haider, 2011; Modi et al., 2013a) mention already known threats but they do not provide a mechanism for identifying gaps in how unknown potential threats can be identified and mitigated against.

Coppolino *et al.* (Coppolino et al., 2017) extensively surveyed threats and attacks in the cloud. Their survey categorizes attacks according to the components in the cloud in which they can be applied. They classified attacks into three classes: (1) Network, (2) Hardware, and (3) Hypervisor. Examples of Network-based attacks are DDoS, Spoofing, Code Injection, and Malware Injection. Hardware-based attacks include theft of encryption keys, cross VM and a cache-based side channel, prime+trigger+probe, and boot integrity. They cited guest VM escape, code injection on Dom0, direct kernel structure manipulation(DKSM), and blue pill as instances of hypervisor attacks.

We took a similar approach to attack categorization in our study. However, we first enumerated attacks in the cloud, categorized them using the OWASP attack categorization. We then mapped the attack categories into cloud components. We considered a more fine-grained set of six cloud components (Application, Web Server, Operating System, Virtual Machine, Hypervisor (Virtual Machine Manager), and Host). Mapping the attacks to these six cloud components gave us a more refined view of the relationship between the attacks and components. Such a detailed view is important for a thorough understanding of the attacks. Copolino *et al.* also identified attack vectors (external users, internal users, and cloud provider). In our study, we did not consider attack vectors but focused on the threat and attack categorization, and how these relate to components.

2.2. Attack Categorization

Khorshed *et al.* (Khorshed et al., 2012) presented an extensive review of gaps and security concerns in the cloud, identified top security threats and their mitigation strategies. They also proposed a proactive threat detection model with features for attack detection, alert, and classification. They concluded that the support vector machine pattern detection technique is highly accurate in identifying attacks. Although our survey does not propose any attack detection techniques, by using the STRIDE as a guide for systematically analyzing and classifying the threats in the cloud, our approach has the potential for advancing towards a solid framework on which approaches to generic attack detection can be built on. Furthermore, we have identified a wider set of threats and attacks which we have linked to the cloud components where they can be realized. The sets of threats and attacks identified by Khorshed *et al.* (Khorshed et al., 2012) and others (Jouini and Rabai, 2014) are limited.

Resiliency in the cloud is one of the topics that received more attention in recent research (Azab et al., 2016; Colman-Meixner et al., 2016; Tunc et al., 2014; Benameur et al., 2013). According to Colman *et al.* (Colman-Meixner et al., 2016) any of the main cloud components can be a source of failures. Their work discussed classes of failures and their consequences that may occur based on a layered cloud architecture and their consequences. Their study surveyed approaches to incorporating resiliency in each of the main cloud computing layers (Physical, Virtualization, and Application) components. Although resiliency is commonly related to failures in general, when viewed from the perspective of failures that can result from attacks, approaches to resiliency are attack mitigation strategies. However, the Colman *et al.* (Colman-Meixner et al., 2016) study reviewed resiliency in general with no particular focus on secuirty attacks as a cause of failure.

Gupta *et al.* (Gupta and Badve, 2017; Stergiou et al., 2018; Gupta and Badve, 2013) focused on denial of service attacks. They provided the taxonomy of the denial of service attack types and their applicability in the cloud computing environment. However, our work is concerned with all attack types applicable in the cloud.

2.3. Surveys on Threat Identification

Table 1 presents a summary of the contributions of some of the most recent surveys reviewing attacks, threats, and countermeasures in the cloud. Roy *et al.* (Roy et al., 2015) discusses attack scenarios that leverage virtualization often experienced from the perspective of organizations providing their services through the cloud. The attack scenarios discussed include those that require securing against hypervisor compromise, multi-tenancy of VMs, VM image management, cloud storage,

insider threats, and data propagation. They also review security solutions designed to mitigate against these attack scenarios. Gupta *et al.* (Gupta et al., 2016; Bhushan and Gupta, 2017) also provided an overview of the current cloud security challenges, and presented the applicability of the existing cybersecurity solutions in the cloud. This survey overlaps with our work in some few research issues. We also study cloud attacks and the components of the cloud where those attacks are most often materialized and also map the attacks on their mitigation mechanisms. However, their work is narrowed to only attacks that leverage virtualization. Our study also covered attacks arising from other aspects of the cloud. Their study does not discuss the threats from which the attacks scenarios arise. This is an issue that our work has studied in-depth.

Sookhak *et al.* (Sookhak et al., 2015) reviewed remote data auditing and recovery techniques for distributed clouds. They proposed a taxonomy for classifying such techniques and identified possible research areas on remote auditing of data stored in the cloud. Their survey is focused on classifying countermeasures related to the specific area of remote data auditing in distributed clouds. On the other hand, our study has focused on classifying a wide range of attacks and threats. The taxonomy they created on countermeasures for data tampering attacks can complement our work by providing a comprehensive classification of countermeasures for this specific type of threat.

Shan *et al.* (Shan et al., 2018) surveyed technologies for secure outsourcing of computation in the cloud and identified security threats and requirements for the design of schemes for outsourcing computation. The outsourcing solutions are systematically mapped to the specific outsourcing problems they address such as matrix computation, data confidentiality, and computation integrity. The survey identified a limited set of threat categories - they considered only threats related to *data confidentiality* and *computation integrity*. By using the STRIDE we considered a wider range of threats in the cloud. Similar to Shan *et al.*'s work (Shan et al., 2018), Tang *et al.* (Tang et al., 2016) discussed the state-of-the-art security solutions towards protection of privacy in untrusted clouds. They identified threats and security requirements in outsourcing data services to the cloud. They also identified open research challenges in key areas of data outsourcing such as "data search, data computation, data sharing, data storage, and data access" (Shan et al., 2018). Although they listed the known security threats in the cloud, they did not provide the means for identifying unknown threats.

Pearce *et al.*(Pearce et al., 2013) identified the different scenarios by which security breaches can occur in virtualized environments and the related countermeasures that can be taken to protect the cloud against such breaches. They also identified

Table 1: Summary of existing surveys, their contributions and limitations

Survey	Contributions	Limitations	Our Contribution
Roy *et al.* (Roy et al., 2015)	• Identify attack scenarios in the cloud that lead to VM crashes from the perspective of an organization providing their services through the cloud. • Identify mitigation mechanisms often used to thwart the attacks.	• Do not discuss the threats from which the attacks scenarios arise. • They are focused on those attacks that are related to virtualization (Similar to the work of Pearce *et al.*(Pearce et al., 2013)).	• We systematically identify threats from attacks. • We cover a wider variety of attacks such as those arising from VM migration.
Sookhak *et al.* (Sookhak et al., 2015)	• Reviews remote data auditing and recovery techniques for distributed clouds. • Propose a taxonomy for classifying distributed data auditing techniques and identify possible research areas.	• Limited to classification of countermeasures related to the specific area of remote data auditing in distributed clouds.	• Our study focused on classifying a wide range of attacks and threats.
Shan *et al.* (Shan et al., 2018)	• Identified threats and requirements that should be considered when designing outsourcing schemes. • Reviews solutions for secure outsourcing of computations to the cloud.	• They identified a limited set of threat categories - they considered only threats related to *data confidentiality* and *computation integrity*.	• By using the STRIDE we considered a wider range of the threats in the cloud.
Tang *et al.* (Tang et al., 2016)	• Discussed cloud threats and security requirements in outsourcing data services to the cloud.	• Although their survey of threats related to outsourcing is comprehensive they do not propose any approach for eliciting unknown threats.	• We developed a method for systematically identifying the threats in the cloud.
Pearce *et al.* (Pearce et al., 2013)	• Identified security threats resulting from virtualization. • Identified security breaches that may occur as a result of the threats.	• The threats identified were not categorized using a known threat categorization model. • Only a standard way to compare virtualization threats with threats in other components of the cloud provided.	• We used STRIDE as a standard model for categorizing the threats in different components in the cloud.
Sgandurra *et al.* (Sgandurra and Lupu, 2016)	• Provides insight on the evolution of attacks and threat models in virtualization. • Shows how security solutions have evolved to thwart new attacks.	• They do not provide any method for identifying threats.	• We propose a method for threat identification.
Gupta *et al.* (Bhushan and Gupta, 2017)	• Provides the state of the art security challenges for the cloud computing. • Identifies various security issues of the cloud environment.	• They do not provide any method for identifying threats and evaluation approaches of them.	• We propose a method for threat evaluation and identification.

security threats resulting from virtualization such as untrusted components, VMM insertion and hijacking, introspection and intervention by VMM, and VM cloning. We also identify attacks and their known countermeasures. However, our work is not restricted only in virtualization but also in other technologies that realize the cloud. Although limited in scope, the insight provided by their work on virtualization attacks and their countermeasures complements our study, and it provides a comprehensive understanding of the attacks related to virtualization.

Sgandurra *et al.* (Sgandurra and Lupu, 2016) provides insight into the evolution of attacks and threat models in virtualization due to changing trust assumptions and how security solutions have evolved to thwart new attacks. Their work has similarities to ours as they discussed the classification of threats. However, they studied threats at different layers of a virtualized system. Although we also studied the threats and attacks at different layers of the cloud, we did not restrict ourselves to virtualization. By studying the evolution of threats, attacks, and their countermeasures their work has potential application to our method of identifying the threats in the cloud. The evolution trends may help in the correlation of threats, attacks, and countermeasures - thus (possibly) enabling prediction of future attacks.

3. Cloud Computing

The cloud provides an on-demand and pay-per-use model for its customers. It provides large computational capabilities, resources and memory space, which also minimizes the cost of setup and deployment (Mell and Grance, 2011). This enables the cloud to be accessible not only for large enterprises but also for small and medium businesses (SMB) to outsource IT setup and management (Gupta et al., 2013). The cloud has three service models, namely: "Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (Iaas)"(Spring, 2011). These service models provide various flexibility in the ability to control the network and its applications via the cloud. The cloud provider and developer manage the cloud provisioning. One or more users then can connect to the cloud and subscribe to the services provided. We look at the architecture of the cloud in Section 3.1, and Section 3.2 describes the cloud components.

3.1. Cloud Architecture

The cloud reference architecture proposed by NIST (Liu et al., 2011b) summarizes various entities and their associated services. The cloud reference architecture provides important information regarding which entity is responsible for which service they provide. There are five entities that form the ecosystem of the cloud; (1)

Cloud Consumer (also referred to as users or tenants), (2) Cloud Auditor, (3) Cloud Broker, (4) Cloud Provider, and (5) Cloud Carrier. Cloud consumers can utilize the functionalities of the cloud computing resources (e.g., SMB outsourcing IT infrastructure), and usually are customers who purchase the cloud computing resources. Cloud Auditors ensure that the cloud provider renders security, privacy, and performance to the cloud consumer in accordance with relevant service level agreements, regulations, and policies. Cloud providers manage cloud components (e.g., management, service, resources). Cloud brokers are typically third parties who handle service intermediation, aggregation, and arbitrage between the cloud provider and the consumer. Cloud carriers are responsible for providing the "connectivity and transport of cloud services between" (Lenkala et al., 2013) the cloud provider and the cloud consumers.

The cloud architecture shows that malicious activities can be conducted by those five different entities. The cloud consumers typically have no trust on the cloud, but the other entities have some form of trust relationship with the cloud. Hence, the cloud auditor, provider, broker, and carrier would all be referred to as *inside attacker* given they have a trusted access to one or more component(s) of the cloud. However, the severity of an attack would differ due to the different trust level that each entity has. For example, the cloud provider can offer malicious software to the client in a SaaS model, or carry out a VM co-location attack (Zhang et al., 2012b). As a result, it could affect all security objectives of the client. On the other hand, the cloud carrier has a limited control over the cloud configurations. However, the cloud carrier can still conduct a denial of service attack. For a deeper understanding of various attacks to the cloud, we investigate various cloud components, and how they can be exploited in the next Section.

3.2. Cloud Components

From the cloud service model point of view, we take into account five main cloud components (Coppolino et al., 2017); Applications & Servers, Operating System, Virtual Machine, Virtual Machine Monitor (VMM), and Host. As we only consider the attacks in the cyberspace, we omit the hardware and its related components in this survey. Based on the cloud reference architecture 3.1, those cloud components can be managed by "either the cloud consumer (e.g., tenants) or the cloud provider"(Mei et al., 2013). The cloud consumers can customize certain cloud components, as the provider may not be able to supply services and modules the consumers need. This limits the visibility of different components for the cloud provider. A host in the cloud forms the underlying structure of the cloud, where it manages physical storage servers and network, VMM, and VMs. VMMs are divided into two types: (i) type

I VMMs can "run directly on the hardware without the" (Qi et al., 2017) need of hosting OS (e.g., ESXi (VMWare, 2009)), and (ii) type II VMMs run on the hosting OS (e.g., KVM (Kivity et al., 2009) or Xen (Barham et al., 2003)). VMs are located within hosts, and the number of available VMs depends on the resource constraints of the host.

This survey not only classifies and identifies threats to the cloud, but also look at in detail what threats and attack vectors each of the cloud component have. By looking at the component level, system and security administrators can better understand what threats are imposed, and can provide more concise mitigation strategies.

3.3. Cloud Deployment Models

Control and visibility of cloud components depend on the deployment model (Subashini and Kavitha, 2011). The cloud deployment model dictates how different cloud components are visible to external entities. By understanding the operation of the cloud with the deployment model, we can examine potential flaw in the design and adopt the best security solutions. There are mainly four cloud deployment models; (1) Public, (2) Private, (3) Hybrid, and (4) Community (Na et al., 2010). Choosing the right deployment and service models depends on the needs of the consumers.

Public clouds provide services over the network that are shared with the public. Typically, the cloud provider renders service and infrastructure to its consumers, and consumers do not have control on where the infrastructure is located. Because this deployment model serves general users and is open to the public, the security requirements are to satisfy various types of users. Private clouds have similar settings as the public clouds, but the deployment environment is private where it is protected by a firewall governed by a particular corporate (i.e., the communication medium is private, allowing only the authenticated users to utilize the cloud computing resources). The cloud providers, as well as consumers, can manage more precisely the level of security requirements as needed in comparison to the public clouds.

Hybrid clouds integrate more than one cloud deployment models (Subashini and Kavitha, 2011) as individual entities but have been set up to be bound together. This model allows users to manage different workloads based on its sensitivity (e.g., front-end servers hosted in the public cloud while database servers are stored in the private cloud). From a security perspective, hybrid deployment models inherit all vulnerabilities associated with all integrated models. Community clouds are established by many organizations in a particular community, sharing the responsibility of setup. Due to its nature, the community deployment model can suffer from lacking adequate security protocols, management and mitigations. Because there are two or more stakeholders that contributed to establishing the cloud, any malicious

stakeholder can misuse the cloud and disturb genuine users.

4. Research Methodology

4.1. Research Questions

The key objective of our survey is to systematically classify and identify the threats in the cloud. In order to achieve this, we consulted the literature on information security for references addressing the sub-questions below:

- *Q1*: What threats are posed in the cloud?
- *Q2*: How can we identify the threats in the cloud?
- *Q3*: Which cyberattacks target the cloud?
- *Q4*: What are the posed threats associated with cyberattacks in the cloud?

We addressed these key questions in order to compile an approach to systematically classify and identify the threats in the cloud. The following subsections describe the methods used to gather and analyze data, and we present our survey on threats and attacks in Section 5.

4.2. Data Gathering

We gathered data by collecting all research articles related to the security of the cloud based on intentional events (i.e., cyberattacks), and we categorized them according to attack types provided by the OWASP (OWASP Foundation, 2018). We utilized various online resources and digital libraries. For description and categorization of an attack, we only considered a representative article for our survey. There is a significant overhead compiling such data as well as for processing them, as there are no automated means to carry out such as task. To make the process more efficient, an automated method is needed. However, this is out of scope in this paper. As far as we know, no research articles or technical papers exist that enlist all possible attack types applicable in the cloud. Hence, this survey provides a first step toward systematically identifying the threats in the cloud as well as providing a comprehensive collection of a set of attacks related to the cloud.

Figure 1 shows the categories of threats, attacks and cloud components, where the collected data are used to map between those categories. The mappings between those categories are then used to trace the vulnerabilities of the cloud components to attacks, and attacks to threats. The STRIDE threat model is used to categorize threats, and the OWASP attack classification is used to categorize attacks. The cloud

components are based on the description given in Section 3.2. Note, the STRIDE threat model and the OWASP attack classification are used for classifying them, and our approach is not limited to using those models only (i.e., alternative threat and attack classification models can be used instead). Those mapping between categories enable the traceability from vulnerabilities to the threats posed to the cloud. Further details of populating the data for each classification are provided in Section 5.

4.3. Data Analysis and Synthesis

While gathering the data, we also specified the following characteristics of attacks: (i) threats posed, (ii) targeted cloud components, (iii) attack category. To evaluate the threat posed by an attack, we used a STRIDE threat model to map the implication of the attack on the threat category. The details of the STRIDE threat modeling are presented in Section 5.2. For example, an SQL injection attack (Gruschka and Jensen, 2010; Dessiatnikoff et al., 2011) is applicable at the Server component of the cloud, which may involve tampering with the server (Tampering), disclosing the sensitive information (Information disclosure), exhausting the connection bandwidth (Denial of service), and obtaining illegally the user credentials (Elevation of privilege). On the other hand, an IP spoofing attack (Subashini and Kavitha, 2011) could affect the IP address of VMs and their hosts, which involves spoofing (Spoofing) and may result in a non-repudiation scenario (Repudiation).

5. Threats and Attacks

There are many ways to look at how threats are posed on the cloud. We particularly focus on threats with malicious intentions (i.e., cyber attacks). To provide the traceability of the threats to the cloud, we define the threat and attack classification models and methods. Using these two, we map attacks to its posed threat(s). We can also identify vulnerabilities in the cloud with respect to each cloud component. Then, the cloud component is mapped to attacks that can exploit them, hence, providing the traceability from vulnerabilities in the cloud components to the threats. The use of the traceability for threat identification is presented later in Section 6.

In this section, first we investigate previous attack patterns targeting the cloud in the past decade as presented in Section 5.1. This highlights the critical attack patterns. Second, we specify a threat model, namely the STRIDE (LeBlanc and Howard, 2002), for classifying the threats in the cloud in Section 5.2. Then, we specify attack categories in Section 5.3. Finally, we classify the threats in the cloud by relating the STRIDE threat model with respect to the attack categories and cloud components in Section 5.4.

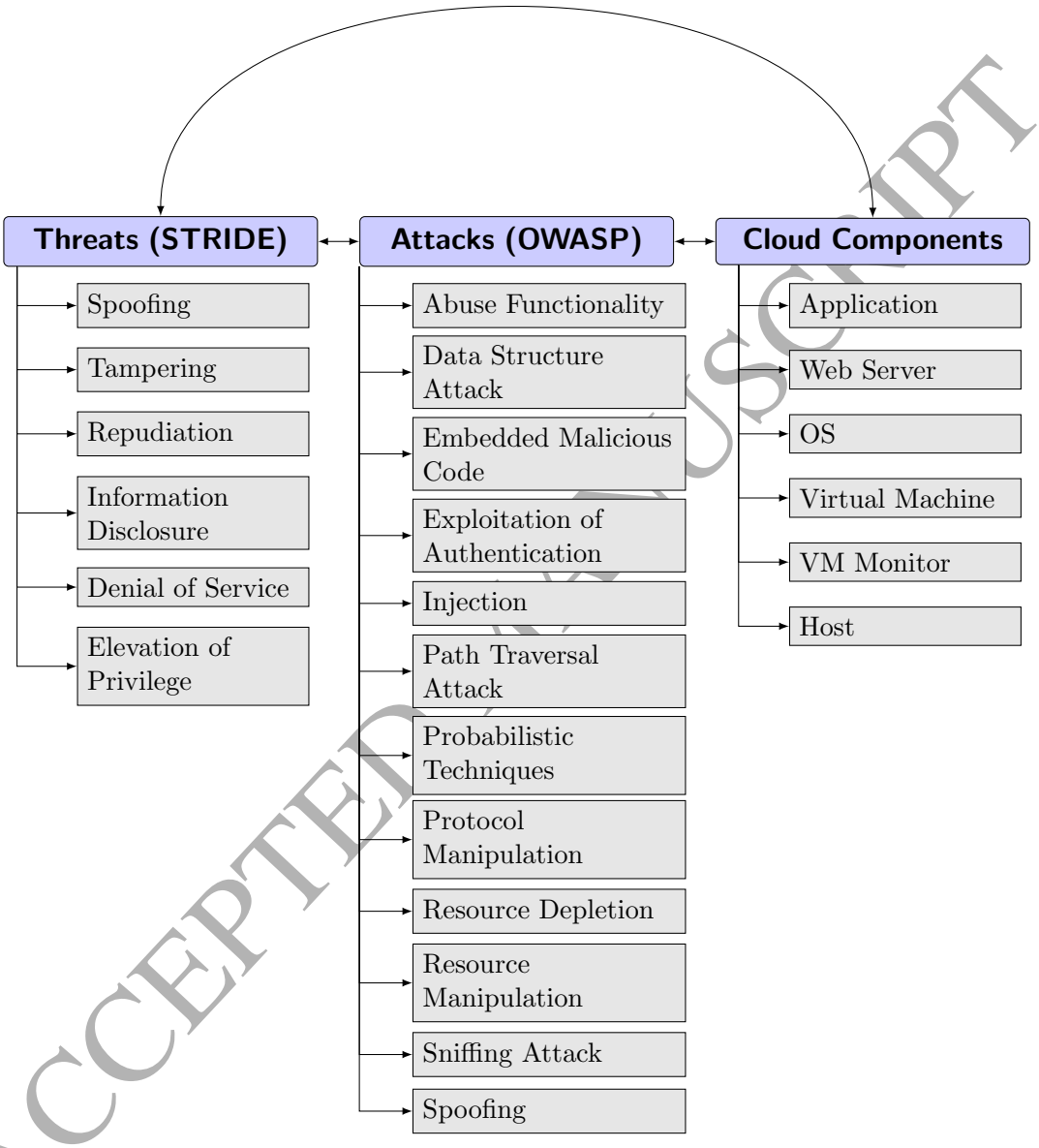

Figure 1: Mapping the threat, attack and cloud component classifications

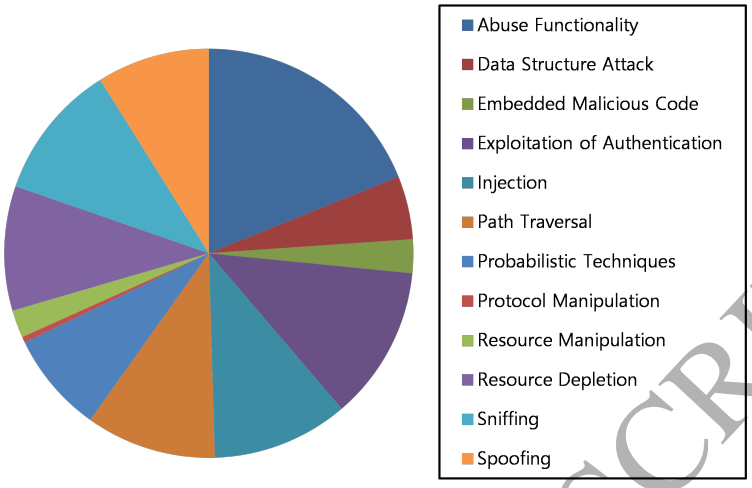

Figure 2: Attacks Patterns in the Cloud Between 2008 and 2012

5.1. Attack Patterns Exploiting the Cloud

We explore the importance of cloud security by taking into account the attack patterns in cybersecurity incidents that occurred in the past decade or so. The Cloud Security Alliance has published an article that reports the number of security incidents that happened in the cloud during the period between 2008 and 2012 (Ko et al., 2013). First, Figure 2 shows the distribution of different attacks reported against the cloud platform between 2008 and 2012. In the original paper in (Ko et al., 2013), they categorized the incidents into their defined threat classifications. We have re-categorized the incidents into attack categories defined by Open Web Application Security Project (OWASP) (OWASP Foundation, 2018), which groups different attacks based on the attack characteristics (further descriptions are given in Section 5.3). We will use this attack categorization to investigate what threats are posed in the cloud, as well as identifying attack vectors that affect different cloud components. Hence, a representation in terms of attack categories gives us a better insight in relation to the threat classification we present later.

Figure 2 shows that the largest amount of attack patterns falls into the *Abuse Functionality* attack category, while the *Protocol Manipulation* attack category has the least amount. Other popular attack categories include *Exploitation of Authentication*, *Injection*, *Path Traversal*, *Resource Depletion* and *Sniffing*. We further specify examples of attacks for each of those attack categories in Section 5.3. The attack patterns in the past decade show that a wide range of attack patterns tar-

geted the cloud without any particular attack categories are more significant than others. Hence, all aspects of attack categories and their posed threats need to be investigated closely.

5.2. The STRIDE Threat Model

The STRIDE threat model (STRIDE) (LeBlanc and Howard, 2002) provides an overview of threats in a given system by categorizing them into six categories. The STRIDE threat model has been used for the cloud (Saripalli and Walters, 2010; Deng et al., 2011), but they only capture the high-level view of the threats posed. To address this, we extend the threat classification to the cloud component level in this survey. The threat categories used by the STRIDE are "spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege" (LeBlanc and Howard, 2002). Each category of the STRIDE captures unique characteristics of the attacks that pose a particular type of threat. It is possible that an attack can pose multiple threats in the STRIDE (e.g., an SQL injection attack (Gruschka and Jensen, 2010; Dessiatnikoff et al., 2011) can pose threats such as tampering with data, disclosing information and/or grant attackers with privileges). Figure 3 shows the mapping of each threat category from the STRIDE to the individual cloud components, where the faded letter represents a particular threat category not applicable for that cloud component based on our survey. For example, at the host level, there exist threats that pose "spoofing, tampering, repudiation, information disclosure, and denial of service" (LeBlanc and Howard, 2002), but not an elevation of privilege. The details of each STRIDE threat model category are as follow.

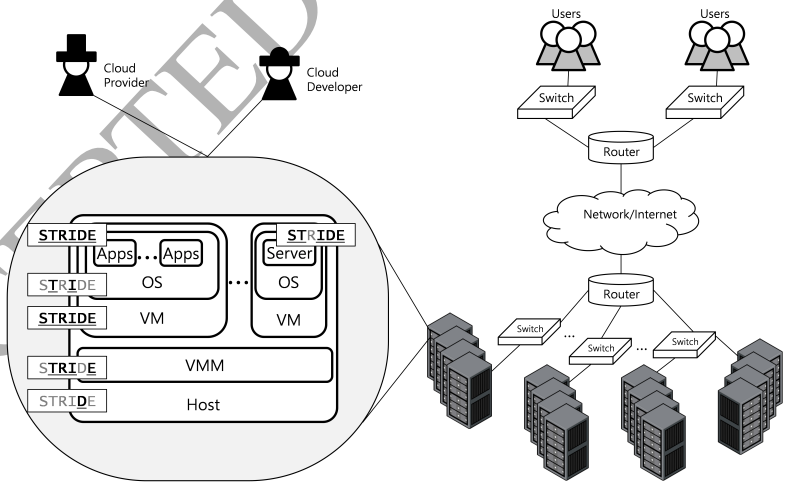

Figure 3: Cloud Components and STRIDE Mapping

Spoofing: This threat specifies when there is illegitimate information or data altered by the attacker with illegitimate information. The targeted users or programs themselves are not modified, but they may misbehave due to the illegitimate information given. For example, the Domain Name System (DNS) can be spoofed causing incorrect IP addresses for requested sites (Xu et al., 2013).

Tampering: Attackers may maliciously modify the system components (such as data, functionalities, communications) to interfere with operations. This threat directly alters the system in comparison to spoofing where illegitimate information is provided to alter the behavior, such as XML poisoning can change commands and codes to make the system malfunction (Saripalli and Walters, 2010).

Repudiation: Attackers can deny actions which cannot be proven due to the lack of ability to provide evidence. Attacks leaving no trace behind can fall into this threat category. For example, DoS attacks may spoof the source IP address to avoid network traces (Chapade et al., 2013a; Osanaiye et al., 2016).

Information Disclosure: Information can leak to unintended individuals due to malicious activities. There are various ways information can leak from the system, such as VM configuration stealing (Cheng et al., 2012), exposing authentication data by URL forgery (Yu et al., 2011), and scanning for open ports to discover services and their associated vulnerabilities (Modi et al., 2013b).

Denial of Service: Valid users are denied from the service due to malicious activities caused by cyber attacks. This threat violates the availability of the system by means of exhausting resources and/or exploiting the system's communication flaws. Such examples include exhausting network bandwidth, memory, and computing capabilities (Chapade et al., 2013b).

Elevation of Privilege: Cyber attacks may allow unauthorized individuals to gain privileged access to the system. Vulnerabilities in the system can be exploited by attackers to bypass the system authentication, and various attack types can be used to exploit those vulnerabilities. For example, unprotected authentication data can be stolen (Saripalli and Walters, 2010), or the authentication data can be exposed due to forgery (Yu et al., 2011).

We classify attacks in the next section to investigate how those threats are posed at the cloud component level, as well as to enlist attacks that are applicable in the cloud. By doing so, we can identify the posed threats and also categorize cloud components affected by the attack. Further, we correlate those attacks to their posed threats. This correlation is used to identify different threats for each cloud component.

5.3. Attack categorization in the cloud

To grasp the full picture of the potential threats in the cloud, we first need to categorize threats and be able to classify the threats in the cloud. This survey focuses on threats posed by attacks, that is, intentional events that cause the cloud to misbehave. Therefore, we investigate the relationship between threats and attacks, and specify what threats are posed by each attack. Due to a large variety of different concrete attacks, we employ a top-down approach to first look at what attack categories they are. For the attack categorization, we use the OWASP attack categorization (OWASP Foundation, 2018) that groups attacks by their characteristics. There are other means of categorization methods such as vulnerability type-based, consequence-based and target-based, but they are not considered in this paper as they only provide a different view on how attacks can be characterized. OWASP lists 12 attack categories: "Abuse Functionality, Data Structure Attack, Embedded Malicious Code, Exploitation of Authentication, Injection, Path Traversal Attack, Probabilistic Techniques, Protocol Manipulation, Resource Depletion, Resource Manipulation, Sniffing Attacks, and Spoofing" (OWASP Foundation, 2018). Some characteristics overlap into multiple categories, and for simplicity, such low-level attacks are put into all categories that they match the description. Each attack category is described in detail with example low-level attacks in the following. Here, we listed representative low-level attacks for each category, as it is practically infeasible to list all variants of low-level attacks. We describe the OWASP attack categories and their details as follows.

Abuse Functionality: Various aspects of computing can have their functionalities abused. For example, IaaS in the cloud without authentication can allow any users (including the attacker) to utilize the cloud computing resources to launch malicious attacks. Hence, we categorize attacks in this category that are utilizing functionalities of the cloud, but in an unintended way. Denial of service attacks fall into this category as they utilize legitimate procedures to make connections, but they exhaust resources for other genuine connections. Such attacks include volume-based (e.g., UDP, ICMP, SYN, Ping-of-Death (Chapade et al., 2013a; Osanaiye et al., 2016)), protocol exploitation (Slowloris, NTP amplification, Smurf attack (Chapade et al., 2013a; Alani, 2016)), application flaws (HTTP flood (Osanaiye et al., 2016)), forcing errors to collecting log data (Saripalli and Walters, 2010), stealing or modifying VM configuration (Cheng et al., 2012; Brohi et al., 2012), launching a malicious VM (Szefer et al., 2011), and redundant interfaces/functions/features, hidden parameters and exposed configuration data can lead to backdoor vulnerabilities (Jansen, 2011; Modi et al., 2013b).

Data Structure Attack: Data structure attacks exploit characteristics of

system data structures, which may result in a violation of its normal usage and protections. By exploiting the vulnerabilities of system process and management for data structures, attackers can access the system data or violate security properties directly. Such attacks include object reference manipulation (Yu et al., 2011), attacking shared memory (Zhang and Reiter, 2013), exploiting API vulnerabilities (Gracia-Tinedo et al., 2013), manipulating decompiled code of RIA components (e.g., Flash, Active X Controls) to bypass security (Saripalli and Walters, 2010), and buffer overflow attacks (Boyd et al., 2010).

Embedded Malicious Code: Applications may contain code that is malicious, which may subvert the security of the application or its host system (e.g., Trojan horse, trapdoor, timebomb) (Subashini and Kavitha, 2011; Yu et al., 2011). Here, we differentiate attacks directly manipulating the system with malicious code such as injection-types of attacks. Typically, an embedded malicious code will not be executed until the user executes the application with the malicious code. Such attacks include malicious script execution by interpreter frameworks (Saripalli and Walters, 2010), poisoning XML with malicious commands/code (Saripalli and Walters, 2010), backdoor installation (Jansen, 2011; Modi et al., 2013b), audio steganography attack (Liu et al., 2011a), and planting malicious files (e.g., malware) that contain codes/instructions to harm user's system (Oberheide et al., 2008; Liu et al., 2011a). Attackers can also embed and execute malicious code that causes VMs to escape (Subashini and Kavitha, 2011), where the attacker can access and manipulate the VMM.

Exploitation of Authentication: System identification and authentication mechanisms with vulnerabilities can be targeted and exploited to expose sensitive data. As a result, attackers may obtain any trust of the targeted systems. For this category, we concern only the attacks manipulating the operations specific to authentications. Such attacks include service engine exploitation (Saripalli and Walters, 2010), stealing unprotected authentication data (e.g., credentials or session tokens) (Saripalli and Walters, 2010), forgery of URL to expose authentication data (Yu et al., 2011), and exploitation and bypass credential validations (Skrupsky et al., 2012). Also, exposed administration and management interfaces, redundant user profiles, and improper authentication and authorization can allow attackers to exploit backdoor vulnerabilities (Jansen, 2011; Modi et al., 2013b).

Injection: Attackers can inject code into a program or query, or execute remote commands by injecting malware onto a computer in order to read and/or modify a database or a website (e.g., "SQL injection" (Shar and Tan, 2013), "code injection" (Riley et al., 2010), "cross-site scripting" (Yusof and Pathan, 2016)). Injection attacks differs to embedded malicious code as malicious code is executed when in-

jected, without having the user to execute applications to trigger the malicious code execution. Such attacks include "cross-site scripting" (XSS for short) (Yusof and Pathan, 2016) (Yu et al., 2011), malware injection (Khalil et al., 2014), SQL injection (Gruschka and Jensen, 2010; Dessiatnikoff et al., 2011), Javascript injection (Provos et al., 2009), OS commanding (Dessiatnikoff et al., 2011), XPATH injection (Saripalli and Walters, 2010; Dessiatnikoff et al., 2011), LDAP injection (Modi et al., 2013a; Skrupsky et al., 2012).

Path Traversal Attack: Various path vulnerabilities can be exploited in order for the attacker to access files or directories not supposed to be visible. For example, a website may have some pages not visible to the public. However, the attacker may discover the URL of the hidden page to access its data. This type of attacks targets any networking systems. More specifically, the attacker can access unauthorized system files through shared folders to manipulate cloud settings (e.g., allowing VM escape) (Ganesan et al., 2012).

Probabilistic Techniques: Probabilistic Techniques refer to successful attacks based on probability. The nature of these attacks is such that there is a probability that the attack would be successful, where other attack categories would either succeed or fail due to a different reason. The attacker can exploit weak cryptographic systems using various statistical and analytical approaches (Yu et al., 2011). Such attacks include Bruteforce attack (Ristenpart et al., 2009), "man in the middle attack" (Subashini and Kavitha, 2011), Side-channel attack (Zhang et al., 2012a), and misconfiguration of the client side validation to bypass authentication (Saripalli and Walters, 2010; Skrupsky et al., 2012).

Protocol Manipulation: Attackers can subvert legitimate communications, allowing them to gather/find information, control the outcome of a session, impersonate others or other attacks. Incorrect implementations or vulnerabilities directly relating to the protocols can lead to this type of attacks. Such attacks include denial of service (e.g., volume-based flooding (Chapade et al., 2013a; Osanaiye et al., 2016), protocol exploitations (Chapade et al., 2013a; Alani, 2016), or exploiting application communication flaws (Osanaiye et al., 2016), manipulating variables in SOAP messages to cause the receiving server to malfunction (Gruschka and Iacono, 2009; Nasridinov et al., 2012), and modifying the contents of the XML information passed "between the user and the server" (Saripalli and Walters, 2010) to discover the security of the target (Saripalli and Walters, 2010).

Resource Depletion: Attackers can exhaust cloud resources, such as network bandwidth, memory, and computing capabilities (Chapade et al., 2013b). Although the cloud provides scalability to deal with the workload size, it is still prone to resource depletion type of attacks such as volume-based flooding (Chapade et al.,

2013a; Osanaiye et al., 2016), protocol exploitations (Chapade et al., 2013a; Alani, 2016), or exploiting application communication flaws (Osanaiye et al., 2016).

Resource Manipulation: Resource Manipulation violates the integrity with intended changes in the cloud. Attackers can cause malfunction of the cloud by tampering with data and resources. Such examples include malfunction of the receiving server by manipulating variables in SOAP messages (Gruschka and Iacono, 2009; Nasridinov et al., 2012), bypassing security from manipulating decompiled code of RIA components (e.g., Flash, Active X Controls) (Saripalli and Walters, 2010), manipulate a direct object reference to access unauthorized data (Bleikertz et al., 2014; Yu et al., 2011), and modifying the XML content information between the user to server communication (Saripalli and Walters, 2010).

Sniffing Attacks: Attackers can gather sensitive data by sniffing network traffic (Ristenpart et al., 2009), allowing remotely stored user data in the cloud to leak if security mechanisms are misconfigured (Squicciarini et al., 2010; Wang et al., 2010). Sniffing attacks are largely divided into passive and active sniffing, where passive sniffing only gathers data communicating between the two parties (Duncan et al., 2013), and active sniffing would use tools and techniques to discover information about the system. Sniffing attacks can often reveal other vulnerabilities and misconfiguration in the system, which are used as stepping stones to launch other types of attacks. For example, attackers can scan for open ports to discover services and its associated vulnerabilities to exploit (Modi et al., 2013b), sniff visible WSDL interface to obtain sensitive information of the network and security services (Masood, 2013; Ibrahim and Hassan, 2015), and conduct side-channel attacks (Zhang et al., 2012a). Attackers can also sniff exposed interfaces, functions, features, parameters, and profiles that would allow backdoor installation (Jansen, 2011; Modi et al., 2013b).

Spoofing: Attackers can impersonate a legitimate trusted entity in the cloud (usually a device or a user) to conduct malicious activities. This allows them to access sensitive data, bypass access control and spread malware. Such attacks include spoofing metadata by impersonating a trusted email sender (Duman et al., 2016), ARP spoofing (Wu et al., 2010), DNS spoofing (Xu et al., 2013), IP spoofing (Subashini and Kavitha, 2011), phishing (Jensen et al., 2009). The attacker can also conduct "cross-site request forgery" (Ron et al., 2016) by forcing "the user's browser" (Somorovsky et al., 2011) to transmit an unauthorized command, such as forged HTTP request, forcing the user to execute malicious actions on a web application (Somorovsky et al., 2011).

5.4. Threat Classification

We specified attack categories and their details in Section 5.3. Using the compiled attack information, we perform three tasks in order to specify the threats in the cloud: (i) identify the threats posed by each attack category using the STRIDE threat model (Section 5.4.1), (ii) identify the cloud components violated by each attack category (Section 5.4.2), and (iii) identify the threats posed to the different cloud components (Section 5.4.3). This enables us to relate all attacks and their posed threats in the cloud at the component level, and also realize how the different cloud components face which threats. Furthermore, we use the threat classification proposed in this section to identify the threats in the cloud in Section 5.

5.4.1. Attack categories vs STRIDE

We summarize the relationship between the attack categories and threats (in terms of STRIDE) in Table 2 using various attacks we gathered in Section 5.3. This table represents the posed threats by different categories of attacks. We observe that certain types of attacks would potentially pose all threat types (e.g., Abuse functionality, Embedded malicious code), whereas some attacks have specific threat types (e.g., Sniffing for information disclosure). The number of references in each slot represents the number of unique references that map the attack category to the threat. Hence, more references in the slot imply the attention and significance gathered by the research community of that specific attack and threat.

Attack categories that do not focus specific intention of the attacker tend to violate all threat categories. The outcome of abuse functionality, which implies the misuse of provided functionalities that causes the malicious or unintended behavior of the cloud, depends on what the attacker intends to do. Malicious code is crafted to do specific tasks depending on the attacker's intention. Resource manipulation could similarly be used as the embedded code to cause the specific outcome of the attacker's intention.

5.4.2. Attack categories vs cloud components

Next, we look at how different categories of attacks affect the cloud components. Table 3 summarizes which attacks are realized at the cloud component level. A large proportion of attack categories is realized at the application, web server and the OS levels, some at the virtual machine and host levels, and only a few at the virtual machine monitor level.

In the cloud service model in Section 3, applications and web servers are the entry points to the cloud system for outsiders. As a result, they are the most focused point of an attack. The table shows the number of unique references that are applicable to that attack category to violate the cloud component.

As shown in Figure 3, VMs and VMMs are two separate modules that sit inside the Host. Given the VMM is not the primary target for most attacks, if the attacker can exploit the host directly, then VMM is not necessarily exploited. That is, taking control of the host can also manipulate the VMM, allowing access to other hosted VMs.

5.4.3. STRIDE vs cloud components

Finally, our gathered data allow us to understand the relationship of which threats are posed in the cloud at the component level. Table 4 shows that there exists a threat to almost all cloud components. However, there are more efforts put into the applications and web servers of the cloud components. On the other hand, the VMM does not have many threats posed. The difficulty to access the VMM makes it a less favorable target, as well as compromising the host when escaping the VM grants the attacker more control than exploiting the VMM.

5.5. Mitigation Techniques

There are various attack types to violate the security of the cloud. This section discusses some of the mitigation techniques to countermeasure attacks on the cloud enlisted above. There may be multiple mitigation techniques available, but we only list some of the latest mitigation techniques highlighted in the references. This is summarized in Table 5, showing the attack and countermeasure pairs. To evaluate the effectiveness of those countermeasures, users should utilize various techniques such as using formal security models with various metrics (Kordy et al., 2013). However, this is out of scope in this paper.

6. Threat Identification in the Cloud

6.1. Threat Identification Method

We use our classification of threats from the tables presented in Section 5.4.2 to identify what attacks can exploit the discovered vulnerabilities of the cloud components, and what threats they pose. The proposed threat identification approach has three main steps, illustrated in figure 4: *(i) Cloud components identification; (ii) Vulnerability identification;* and *(ii) Threat identification.*

6.1.1. Components identification

Components identification entails identifying the cloud components that should be considered in the analysis and their associated information. This includes their configuration and settings, which will be assessed in the next step to identify vulnerabilities. There are challenges in doing this step due to different visibility of cloud

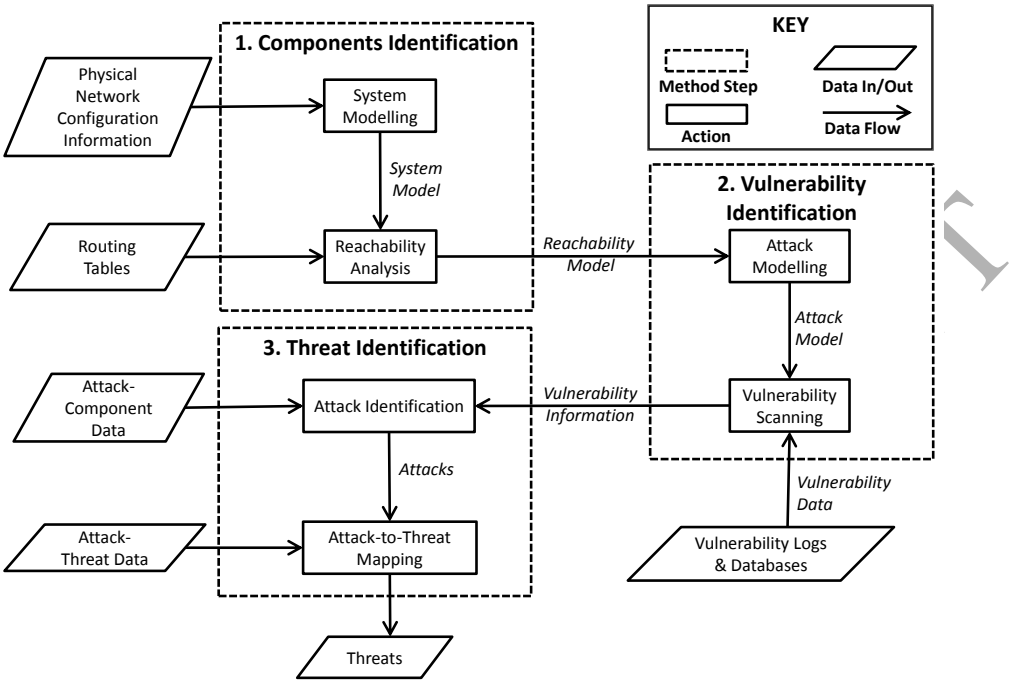

Figure 4: Threat Identification Method Steps and Actions

components depending on the service and deployment models (e.g., with the IaaS, the cloud administrator may not be able to access VM configuration). However, in this work, we assume that the administrator has access to all cloud components and can gather their associated information regardless of the service and deployment models used.

This components identification step consists of two actions: *system modeling* and *reachability analysis*. System modeling creates a model of the network showing the cloud components related to a particular function and how they are physically connected. A system model can be obtained from documentation of an organization's cloud deployment such as network configuration information. Reachability analysis creates a logical model showing which components are able to communicate which ones. This may involve actively analyzing packets as they are being transmitted and received or consulting network routing tables in the hypervisor documenting information on reachability between components. For example, the security groups in the Amazon cloud, EC2 (Jin et al., 2016) contain such routing information.

6.1.2. Vulnerability identification

In the vulnerability identification phase, weaknesses in the network components based on the reachability model that an attacker may exploit to launch a successful attack are identified. This step involves *attack modeling* and *vulnerability scanning*. Attack modeling enriches the reachability modeling with attack entry points and targets. Attack entry points are those components through which an attacker from outside connects to reach the network such as a public-facing Web Server. An attacker that intends to damage one or more target asset(s), such as a database of medical records in a hospital management system that he may want to steal, connects through the Web Server.

Vulnerability scanning checks for known vulnerabilities in the cloud components. The process of finding vulnerabilities uses data from public vulnerability databases and vulnerability logs that document network specific weaknesses known by the security administrator from his experience with the cloud deployment. There are vulnerability scanners that may be used in the cloud platform, such as NESSUS (Beale et al., 2002). Then, this information can be used to match the attack patterns to the corresponding vulnerabilities (Gegick and Williams, 2005; Chen et al., 2017).

6.1.3. Threat Identification

Threat identification is the final step and one of the key contributions of our work. This step uses the information on vulnerabilities in components to identify the posed threats to the cloud using identified attack patterns. It involves two key actions: *attack identification* and *attack-to-threat mapping*. Attack identification seeks to elicit attack scenarios in which the vulnerabilities identified in the cloud components may be exploited to cause damage to an asset.

As vulnerabilities are associated with specific instances of cloud components, we use them to identify the components and the known attacks on those components and finally the attack category. This is achieved through table 3. Using the specific instance of the attack, its category, and the attack-to-threat category in table 2 we identify the threats posed in the *Attack-to-Threat Mapping* action. In order to better understand the steps for identifying the threats in the cloud, we describe a use case study scenario using an example cloud setup in Section 6.2.

6.2. Threat Identification in the Cloud: an Example

Using an example of a cloud deployment, This section illustrates how our approach can be used to identify threats.

Table 2: Attack Category vs STRIDE

ATTACK CATEGORY	STRIDE					
	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privileges
Abuse Functionality	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), malicious VM (Szefer et al., 2011)	steal/modify VM configuration (Cheng et al., 2012; Brohi et al., 2012)	alter hidden parameter (Jansen, 2011; Modi et al., 2013b), DoS/flooding exploiting application/protocol (Chapade et al., 2013a; Alani, 2016; Osanaiye et al., 2016)	exposed configuration, steal/modify VM configuration (Jansen, 2011; Modi et al., 2013b), error log collection (Saripalli and Walters, 2010) steal/modify VM configuration (Cheng et al., 2012; Brohi et al., 2012), malicious VM (Szefer et al., 2011)	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), DoS/flooding exploiting application/protocol (Chapade et al., 2013a; Osanaiye et al., 2016; Alani, 2016)	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), malicious VM (Szefer et al., 2011)
Data Structure Attack		alter object reference (Yu et al., 2011), exploit shared memory (Zhang and Reiter, 2013), exploit API vulnerabilities (Gracia-Tinedo et al., 2013), exploit RIA component code (Saripalli and Walters, 2010), buffer overflow (Boyd et al., 2010)		exploiting shared memory (Zhang and Reiter, 2013), exploit API vulnerabilities (Gracia-Tinedo et al., 2013), exploit RIA component code (Saripalli and Walters, 2010)	exploit shared memory (Zhang and Reiter, 2013), exploit API vulnerabilities (Gracia-Tinedo et al., 2013), buffer overflow (Boyd et al., 2010)	exploit RIA component code (Saripalli and Walters, 2010), buffer overflow (Boyd et al., 2010)
Embedded Malicious Code	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), malicious file (Oberheide et al., 2008), audio steganography (Liu et al., 2011a)	malicious script execution, XML poisoning (Saripalli and Walters, 2010), malicious file (Oberheide et al., 2008), audio steganography (Liu et al., 2011a), VM excape (Subashini and Kavitha, 2011)	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), malicious file (Oberheide et al., 2008), audio steganography (Liu et al., 2011a), VM excape (Subashini and Kavitha, 2011)	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), malicious script execution, XML poisoning (Saripalli and Walters, 2010)	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), malicious file (Oberheide et al., 2008; Liu et al., 2011a)	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), VM excape (Subashini and Kavitha, 2011)

Table 2: Attack Category vs STRIDE (cont...)

	Spoofing	*Tampering*	*Repudiation*	*Information Disclosure*	*Denial of Service*	*Elevation of Privileges*
Exploitation of Authentication	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), steal authentication data (Saripalli and Walters, 2010), URL forgery (Yu et al., 2011), bypass credential validation (Skrupsky et al., 2012)		alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), steal authentication data (Saripalli and Walters, 2010), URL forgery (Yu et al., 2011), bypass credential validation (Skrupsky et al., 2012)	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), steal authentication data (Saripalli and Walters, 2010), URL forgery (Yu et al., 2011), bypass credential validation (Skrupsky et al., 2012)	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b)	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), steal authentication data (Saripalli and Walters, 2010), URL forgery (Yu et al., 2011), bypass credential validation (Skrupsky et al., 2012)
Injection		XSS (Yu et al., 2011), malware injection (Khalil et al., 2014), SQL injection (Gruschka and Jensen, 2010), XPATH injection (Dessiatnikoff et al., 2011), Javascript injection (Provos et al., 2009), OS commanding (Saripalli and Walters, 2010), LDAP injection (Modi et al., 2013a; Skrupsky et al., 2012)		XSS (Yu et al., 2011), malware injection (Khalil et al., 2014), SQL injection (Gruschka and Jensen, 2010), XPATH injection (Dessiatnikoff et al., 2011), Javascript injection (Provos et al., 2009), OS commanding (Saripalli and Walters, 2010), LDAP injection (Modi et al., 2013a; Skrupsky et al., 2012)	XSS (Yu et al., 2011), malware injection (Khalil et al., 2014), SQL injection (Gruschka and Jensen, 2010), XPATH injection (Dessiatnikoff et al., 2011), Javascript injection (Provos et al., 2009), OS commanding (Saripalli and Walters, 2010), LDAP injection (Modi et al., 2013a; Skrupsky et al., 2012)	XSS (Yu et al., 2011), malware injection (Khalil et al., 2014), SQL injection (Gruschka and Jensen, 2010), XPATH injection (Dessiatnikoff et al., 2011), Javascript injection (Provos et al., 2009), OS commanding (Saripalli and Walters, 2010), LDAP injection (Modi et al., 2013a; Skrupsky et al., 2012)
Path Traversal Attack		exploit shared folder, VM escape (Ganesan et al., 2012)	exploit shared folder, VM escape (Ganesan et al., 2012)			exploit shared folder, VM escape (Ganesan et al., 2012)

Table 2: Attack Category vs STRIDE (cont...)

	Spoofing	*Tampering*	*Repudiation*	*Information Disclosure*	*Denial of Service*	*Elevation of Privileges*
Probabilistic Techniques	"man-in-the-middle attack" (Subashini and Kavitha, 2011)		"man-in-the-middle attack" (Subashini and Kavitha, 2011), client side validation misconfiguration (Saripalli and Walters, 2010; Skrupsky et al., 2012)	bruteforce attack (Ristenpart et al., 2009), side-channel attack (Zhang et al., 2012a), client side validation misconfiguration (Saripalli and Walters, 2010; Skrupsky et al., 2012)		bruteforce attack (Ristenpart et al., 2009), client side validation misconfiguration (Saripalli and Walters, 2010; Skrupsky et al., 2012)
Protocol Manipulation	manipulate SOAP message variable (Gruschka and Iacono, 2009; Nasridinov et al., 2012), exploit application communication flaw (Osanaiye et al., 2016), manipulate XML content (Saripalli and Walters, 2010)	manipulate SOAP message variable (Gruschka and Iacono, 2009; Nasridinov et al., 2012), modify/manipulate XML content (Saripalli and Walters, 2010)	volume-based flooding (Chapade et al., 2013a; Osanaiye et al., 2016), exploit protocol (Alani, 2016), manipulate SOAP message variable (Gruschka and Iacono, 2009; Nasridinov et al., 2012), manipulate XML content (Saripalli and Walters, 2010)		volume-based flooding (Chapade et al., 2013a; Osanaiye et al., 2016), protocol exploitation (Alani, 2016), XML content manipulation (Saripalli and Walters, 2010)	
Resource Manipulation	manipulate SOAP message variable (Gruschka and Iacono, 2009; Nasridinov et al., 2012), exploit application communication flaw (Osanaiye et al., 2016)	manipulate SOAP message variable (Gruschka and Iacono, 2009; Nasridinov et al., 2012), modify XML content, manipulate RIA components (Saripalli and Walters, 2010), manipulate direct object references (Bleikertz et al., 2014)	manipulate SOAP message variable (Gruschka and Iacono, 2009; Nasridinov et al., 2012)	modify XML content, manipulate RIA components (Saripalli and Walters, 2010), manipulate direct object references (Bleikertz et al., 2014)	manipulate direct object references (Bleikertz et al., 2014)	modify XML content, manipulate RIA components (Saripalli and Walters, 2010)

Table 2: Attack Category vs STRIDE (cont...)

	Spoofing	*Tampering*	*Repudiation*	*Information Disclosure*	*Denial of Service*	*Elevation of Privileges*
Resource Depletion			volume-based flooding (Chapade et al., 2013a; Osanaiye et al., 2016), protocol exploitation (Alani, 2016)		volume-based flooding (Chapade et al., 2013a; Osanaiye et al., 2016), protocol exploitation (Alani, 2016)	
Sniffing				exposed configuration (Jansen, 2011; Modi et al., 2013b), passive sniffing (Duncan et al., 2013), visible WSDL interface (Masood, 2013; Ibrahim and Hassan, 2015), side-channel attack (Zhang et al., 2012a), scan open ports (Modi et al., 2013b)		
Spoofing	spoof metadata (Duman et al., 2016), ARP spoofing (Wu et al., 2010), IP spoofing (Subashini and Kavitha, 2011), DNS spoofing (Xu et al., 2013), phishing (Jensen et al., 2009), CSRF (Somorovsky et al., 2011)		spoof metadata (Duman et al., 2016), ARP spoofing (Wu et al., 2010), IP spoofing (Subashini and Kavitha, 2011), DNS spoofing (Xu et al., 2013), phishing (Jensen et al., 2009), CSRF (Somorovsky et al., 2011)			

Table 3: Attack Category vs cloud Components

ATTACK CATEGORY	**CLOUD COMPONENTS**					
	Application	*Web Server*	*OS*	*Virtual Machine*	*VM Monitor*	*Host*
Abuse Functionality	hidden parameter manipulation, exposed configuration, install backdoor (Jansen, 2011; Modi et al., 2013b), application flaw-based flooding (Osanaiye et al., 2016), error log collection (Saripalli and Walters, 2010)	hidden parameter manipulation, exposed configuration, install backdoor (Jansen, 2011; Modi et al., 2013b) , application flaw-based flooding (Osanaiye et al., 2016), error log collection (Saripalli and Walters, 2010)	hidden parameter manipulation, exposed configuration, install backdoor (Jansen, 2011; Modi et al., 2013b)	steal/modify VM configuration (Cheng et al., 2012; Brohi et al., 2012) protocol exploitation-based denial of service (Chapade et al., 2013a; Alani, 2016; Osanaiye et al., 2016), malicious VM (Szefer et al., 2011)	malicious VM (Szefer et al., 2011)	volume-based flooding, protocol exploitation-based denial of service (Chapade et al., 2013a; Osanaiye et al., 2016; Alani, 2016)
Data Structure Attack	object reference manipulation (Yu et al., 2011), exploiting API vulnerabilities (Gracia-Tinedo et al., 2013), manipulate RIA component (Saripalli and Walters, 2010), buffer overflow (Boyd et al., 2010)	object reference manipulation (Yu et al., 2011), manipulate RIA component (Saripalli and Walters, 2010), buffer overflow (Boyd et al., 2010)		exploiting shared memory, VM escape (Zhang and Reiter, 2013)		exploiting shared memory, VM escape (Zhang and Reiter, 2013)
Embedded Malicious Code	install backdoor (Jansen, 2011; Modi et al., 2013b), malicious script execution, XML poisoning (Saripalli and Walters, 2010), malicious file (Oberheide et al., 2008; Liu et al., 2011a)	install backdoor (Jansen, 2011; Modi et al., 2013b), malicious script execution, XML poisoning (Saripalli and Walters, 2010), malicious file (Oberheide et al., 2008; Liu et al., 2011a)	install backdoor (Jansen, 2011; Modi et al., 2013b)		embedded code execution, VM excape (Subashini and Kavitha, 2011)	

Table 3: Attack Category vs cloud Components (cont...)

	Application	*Web Server*	*OS*	*Virtual Machine*	*VM Monitor*	*Host*
Exploitation of Authentication	expose interfaces, alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), steal authentication data, exploit service engine (Saripalli and Walters, 2010), URL forgery (Yu et al., 2011), bypass credential validation (Skrupsky et al., 2012)	expose interfaces, alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), steal authentication data, exploit service engine (Saripalli and Walters, 2010), URL forgery (Yu et al., 2011), bypass credential validation (Skrupsky et al., 2012)	expose interfaces, alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b)			
Injection	XSS (Yu et al., 2011), malware injection (Khalil et al., 2014), SQL injection (Gruschka and Jensen, 2010), XPATH injection (Dessiatnikoff et al., 2011), Javascript injection (Provos et al., 2009), OS commanding (Saripalli and Walters, 2010), LDAP injection (Modi et al., 2013a; Skrupsky et al., 2012)	XSS (Yu et al., 2011), malware injection (Khalil et al., 2014), SQL injection (Gruschka and Jensen, 2010), XPATH injection (Dessiatnikoff et al., 2011), Javascript injection (Provos et al., 2009), OS commanding (Saripalli and Walters, 2010), LDAP injection (Modi et al., 2013a; Skrupsky et al., 2012)	XPATH injection (Dessiatnikoff et al., 2011)			
Path Traversal Attack					manipulate shared folder, VM escape (Ganesan et al., 2012)	
Probabilistic Techniques	bruteforce attack (Ristenpart et al., 2009), "man-in-the-middle attack" (Subashini and Kavitha, 2011), client side validation misconfiguration (Saripalli and Walters, 2010; Skrupsky et al., 2012)	bruteforce attack (Ristenpart et al., 2009), "man-in-the-middle attack" (Subashini and Kavitha, 2011), client side validation misconfiguration (Saripalli and Walters, 2010; Skrupsky et al., 2012)	bruteforce attack (Ristenpart et al., 2009), "man-in-the-middle attack" (Subashini and Kavitha, 2011)	"man-in-the-middle attack" (Subashini and Kavitha, 2011)		"man-in-the-middle attack" (Subashini and Kavitha, 2011), side-channel attack (Zhang et al., 2012a)

Table 3: Attack Category vs cloud Components (cont...)

	Application	*Web Server*	*OS*	*Virtual Machine*	*VM Monitor*	*Host*
Protocol Manipulation	application communication flaw exploitation (Osanaiye et al., 2016), XML content manipulation (Saripalli and Walters, 2010)	SOAP message variable manipulation (Gruschka and Iacono, 2009; Nasridinov et al., 2012), application communication flaw exploitation (Osanaiye et al., 2016), XML content manipulation (Saripalli and Walters, 2010)	SOAP message variable manipulation (Gruschka and Iacono, 2009; Nasridinov et al., 2012), XML content manipulation (Saripalli and Walters, 2010)	volume-based flooding (Chapade et al., 2013a; Osanaiye et al., 2016), protocol exploitation (Alani, 2016)		volume-based flooding (Chapade et al., 2013a; Osanaiye et al., 2016), protocol exploitation (Alani, 2016)
Resource Depletion	application communication flaw exploitation (Osanaiye et al., 2016)	application communication flaw exploitation (Osanaiye et al., 2016)		volume-based flooding (Chapade et al., 2013a; Osanaiye et al., 2016), protocol exploitation (Alani, 2016)		volume-based flooding (Chapade et al., 2013a; Osanaiye et al., 2016), protocol exploitation (Alani, 2016)
Resource Manipulation	direct object reference manipulation (Bleikertz et al., 2014; Yu et al., 2011), XML content manipulation (Saripalli and Walters, 2010)	SOAP message variable manipulation (Gruschka and Iacono, 2009; Nasridinov et al., 2012), XML content manipulation (Saripalli and Walters, 2010)	SOAP message variable manipulation (Gruschka and Iacono, 2009; Nasridinov et al., 2012), XML content manipulation (Saripalli and Walters, 2010)			
Sniffing	exposed configuration, install backdoor (Jansen, 2011; Modi et al., 2013b), passive sniffing (Duncan et al., 2013), sniff visible WSDL interface (Masood, 2013; Ibrahim and Hassan, 2015)	exposed configuration, install backdoor (Jansen, 2011; Modi et al., 2013b), passive sniffing (Duncan et al., 2013), sniff visible WSDL interface (Masood, 2013; Ibrahim and Hassan, 2015)	exposed configuration, install backdoor (Jansen, 2011; Modi et al., 2013b), passive sniffing (Duncan et al., 2013)	passive sniffing (Duncan et al., 2013), scan open ports (Modi et al., 2013b)		passive sniffing (Duncan et al., 2013), scan open ports (Modi et al., 2013b), side-channel attack (Zhang et al., 2012a)
Spoofing	spoof metadata (Duman et al., 2016), phishing (Jensen et al., 2009), "cross-site request forgery" (Somorovsky et al., 2011)	"cross-site request forgery" (Somorovsky et al., 2011)		ARP spoofing (Wu et al., 2010), DNS spoofing (Xu et al., 2013), IP spoofing (Subashini and Kavitha, 2011)		ARP spoofing (Wu et al., 2010), DNS spoofing (Xu et al., 2013), IP spoofing (Subashini and Kavitha, 2011)

Table 4: STRIDE vs cloud Components

STRIDE	**CLOUD COMPONENTS**					
	Application	*Web Server*	*OS*	*Virtual Machine*	*VM Monitor*	*Host*
Spoofing	hidden parameter manipulation, install backdoor (Jansen, 2011; Modi et al., 2013b), spoof metadata (Duman et al., 2016), phishing (Jensen et al., 2009), cross-site request forgery (Somorovsky et al., 2011), IP spoofing (Subashini and Kavitha, 2011), malicious file (Oberheide et al., 2008; Liu et al., 2011a), buffer overflow (Boyd et al., 2010)	hidden parameter manipulation, install backdoor (Jansen, 2011; Modi et al., 2013b), cross-site request forgery (Somorovsky et al., 2011), SOAP message variable manipulation (Gruschka and Iacono, 2009; Nasridinov et al., 2012), IP spoofing (Subashini and Kavitha, 2011), malicious file (Oberheide et al., 2008; Liu et al., 2011a), buffer overflow (Boyd et al., 2010)	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), manipulate SOAP message variable (Gruschka and Iacono, 2009; Nasridinov et al., 2012), IP spoofing (Subashini and Kavitha, 2011)	ARP spoofing (Wu et al., 2010), DNS spoofing (Xu et al., 2013), IP spoofing (Subashini and Kavitha, 2011), malicious VM (Szefer et al., 2011)	malicious VM (Szefer et al., 2011)	ARP spoofing (Wu et al., 2010), DNS spoofing (Xu et al., 2013), IP spoofing (Subashini and Kavitha, 2011)
Tampering	manipulate RIA components (Saripalli and Walters, 2010), direct object reference manipulation, XSS (Bleikertz et al., 2014; Yu et al., 2011), LDAP (Modi et al., 2013a; Skrupsky et al., 2012), SQL, Javascript (Provos et al., 2009), malware (Khalil et al., 2014) injections, OS commanding (Dessiatnikoff et al., 2011), malicious file (Oberheide et al., 2008; Liu et al., 2011a), buffer overflow (Boyd et al., 2010), API vulnerability exploitation (Gracia-Tinedo et al., 2013)	manipulate RIA components (Saripalli and Walters, 2010), manipulate SOAP message variable (Nasridinov et al., 2012), SQL (Gruschka and Iacono, 2009), LDAP (Skrupsky et al., 2012), Javascript (Provos et al., 2009), malware (Khalil et al., 2014) injections, OS commanding (Dessiatnikoff et al., 2011), direct object reference manipulation (Yu et al., 2011), malicious file (Oberheide et al., 2008), buffer overflow (Boyd et al., 2010)	SQL injection (Gruschka and Iacono, 2009), SOAP message variable manipulation (Nasridinov et al., 2012), OS commanding (Dessiatnikoff et al., 2011)	attack shared memory (Zhang and Reiter, 2013), modify VM configuration (Cheng et al., 2012; Brohi et al., 2012)	manipulate shared folder (Ganesan et al., 2012), malicious code execution, VM escape (Subashini and Kavitha, 2011)	attack shared memory (Zhang and Reiter, 2013)

Table 4: STRIDE vs cloud Components (cont...)

STRIDE	**CLOUD COMPONENTS**					
	Application	*Web Server*	*OS*	*Virtual Machine*	*VM Monitor*	*Host*
Repudiation	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), CSRF (Somorovsky et al., 2011), phishing (Jensen et al., 2009), spoof metadata (Duman et al., 2016), application flaw exploitation (Osanaiye et al., 2016), stealing unprotected authentication data, client side validation misconfiguration (Saripalli and Walters, 2010; Skrupsky et al., 2012), embedded malicious code (Subashini and Kavitha, 2011), malicious files (Oberheide et al., 2008; Liu et al., 2011a)	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), CSRF (Somorovsky et al., 2011), manipulate SOAP message variable (Gruschka and Iacono, 2009; Nasridinov et al., 2012), exploit application flaw (Osanaiye et al., 2016), steal authentication data, misconfigure validation (Saripalli and Walters, 2010), embedded malicious code (Subashini and Kavitha, 2011), malicious files (Oberheide et al., 2008)	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), SOAP message variable manipulation (Gruschka and Iacono, 2009; Nasridinov et al., 2012), embedded malicious code (Subashini and Kavitha, 2011)	IP (Subashini and Kavitha, 2011), DNS (Xu et al., 2013), ARP (Wu et al., 2010) spoofing, volume-based flooding, protocol manipulation (Chapade et al., 2013a; Alani, 2016; Osanaiye et al., 2016), malicious VM (Szefer et al., 2011)	exploit shared folder, VM escape (Ganesan et al., 2012), man-in-the-middle attack (Subashini and Kavitha, 2011), malicious VM (Szefer et al., 2011), exploit service engine (Saripalli and Walters, 2010)	man-in-the-middle attack, IP (Subashini and Kavitha, 2011), DNS (Xu et al., 2013), ARP (Wu et al., 2010) spoofing, flooding/protocol manipulation (Chapade et al., 2013a; Alani, 2016; Osanaiye et al., 2016)

Table 4: STRIDE vs cloud Components (cont...)

STRIDE	**CLOUD COMPONENTS**					
	Application	*Web Server*	*OS*	*Virtual Machine*	*VM Monitor*	*Host*
Information Disclosure	exposed configuration (Jansen, 2011; Modi et al., 2013b), visible WSDL interface (Masood, 2013; Ibrahim and Hassan, 2015), direct object reference manipulation (Bleikertz et al., 2014; Yu et al., 2011), error log collection (Saripalli and Walters, 2010), bypass credential validation (Skrupsky et al., 2012), brute-force attack (Ristenpart et al., 2009), scan open ports (Modi et al., 2013a), OS commanding (Dessiatnikoff et al., 2011), SQL (Gruschka and Iacono, 2009), Javascript (Provos et al., 2009), malware (Khalil et al., 2014) injections, API exploitation (Gracia-Tinedo et al., 2013)	exposed configuration (Jansen, 2011; Modi et al., 2013b), visible WSDL interface (Masood, 2013; Ibrahim and Hassan, 2015), error log collection (Saripalli and Walters, 2010), bypass credential validation (Skrupsky et al., 2012), brute-force attack (Ristenpart et al., 2009), scan open ports (Modi et al., 2013a), OS commanding (Dessiatnikoff et al., 2011; Gruschka and Iacono, 2009), SQL, Javascript (Provos et al., 2009), malware (Khalil et al., 2014) injections, API exploitation (Gracia-Tinedo et al., 2013)	exposed configuration (Jansen, 2011; Modi et al., 2013b), side-channel attack (Zhang et al., 2012a), brute-force attack (Ristenpart et al., 2009), OS commanding (Dessiatnikoff et al., 2011)	steal/modify VM configuration (Cheng et al., 2012; Brohi et al., 2012), scan open port (Modi et al., 2013b), passive sniffing (Duncan et al., 2013), shared memory exploitation (Zhang and Reiter, 2013)		scan open port (Modi et al., 2013b), passive sniffing (Duncan et al., 2013), shared memory exploitation (Zhang and Reiter, 2013), side-channel attack (Zhang et al., 2012a)
Denial of Service	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), manipulate direct object reference (Bleikertz et al., 2014; Yu et al., 2011), exploit application flaws (Osanaiye et al., 2016), LDAP (Modi et al., 2013a), Javascript (Provos et al., 2009), XPATH (Dessiatnikoff et al., 2011), malware (Khalil et al., 2014) injections, malicious file (Oberheide et al., 2008), buffer overflow (Boyd et al., 2010), exploit API (Gracia-Tinedo et al., 2013)	alter hidden parameter, install backdoor (Jansen, 2011; Modi et al., 2013b), exploit application flaws (Osanaiye et al., 2016), LDAP (Modi et al., 2013a; Skrupsky et al., 2012), SQL (Gruschka and Iacono, 2009), Javascript (Provos et al., 2009), XPATH (Dessiatnikoff et al., 2011), malware (Khalil et al., 2014) injections, XSS (Yu et al., 2011), malicious file (Oberheide et al., 2008), buffer overflow (Boyd et al., 2010)	hidden parameter manipulation, install backdoor (Jansen, 2011; Modi et al., 2013b), XPATH (Dessiatnikoff et al., 2011), OS commanding (Dessiatnikoff et al., 2011)	volume-based flooding, protocol exploitation (Chapade et al., 2013a; Alani, 2016; Osanaiye et al., 2016), shared memory exploitation (Zhang and Reiter, 2013)		volume-based flooding, protocol exploitation (Chapade et al., 2013a; Alani, 2016; Osanaiye et al., 2016), shared memory exploitation (Zhang and Reiter, 2013)

Table 4: STRIDE vs cloud Components (cont...)

STRIDE	**CLOUD COMPONENTS**					
	Application	*Web Server*	*OS*	*Virtual Machine*	*VM Monitor*	*Host*
Elevation of Privileges	improper authentication/ authorization, install backdoor (Jansen, 2011; Modi et al., 2013b), bypass credential validation (Skrupsky et al., 2012), brute-force (Ristenpart et al., 2009), LDAP (Modi et al., 2013a), SQL (Dessiatnikoff et al., 2011), Javascript (Provos et al., 2009), XPATH (Dessiatnikoff et al., 2011), malware (Khalil et al., 2014) injections, XSS (Yu et al., 2011), buffer overflow (Boyd et al., 2010)	bypass credential validation (Skrupsky et al., 2012), brute-force attack (Ristenpart et al., 2009), LDAP (Modi et al., 2013a), SQL (Dessiatnikoff et al., 2011), Javascript (Provos et al., 2009), XPATH (Dessiatnikoff et al., 2011), malware (Khalil et al., 2014) injections, XSS (Yu et al., 2011), buffer overflow (Boyd et al., 2010)	improper authentication/ authorization, install backdoor (Jansen, 2011; Modi et al., 2013b), brute-force attack (Ristenpart et al., 2009), XPATH injection, OS commanding (Dessiatnikoff et al., 2011)	malicious VM (Szefer et al., 2011)	manipulate shared folder, VM escape (Ganesan et al., 2012), malicious code (Subashini and Kavitha, 2011), malicious VM (Szefer et al., 2011), exploit service engine (Saripalli and Walters, 2010)	

Table 5: Attack and Mitigations

Attack	Mitigation
volume-based flooding (Chapade et al., 2013a; Osanaiye et al., 2016)	detection, filtering (Chapade et al., 2013a)
protocol manipulation (Chapade et al., 2013a; Alani, 2016)	detection, encryption, filtering (Alani, 2016)
application flaw DoS (Osanaiye et al., 2016)	detection, filtering (Osanaiye et al., 2016)
error log collection (Saripalli and Walters, 2010)	error handling (Arroyo et al., 2016)
steal/modify VM configuration (Cheng et al., 2012)	modeling and analysis (Cheng et al., 2012)
hidden parameters (Jansen, 2011)	data isolation / sanitization (Jansen, 2011)
exposed configuration data (Jansen, 2011)	access control, encryption (Jansen, 2011)
backdoor installation (Jansen, 2011)	cloud data encryption (Amoroso, 2014)
object reference manipulation (Yu et al., 2011)	detection and firewall (Yu et al., 2011)
exploit shared memory (Zhang and Reiter, 2013)	cache cleansing (Zhang and Reiter, 2013)
exploit API (Gracia-Tinedo et al., 2013)	access control, session time limits, anomaly detection (Gracia-Tinedo et al., 2013)
manipulate RIA components (Saripalli and Walters, 2010)	security policy enforcement, script execution monitoring (Erlingsson et al., 2014)
buffer overflow (Boyd et al., 2010)	instruction set randomization (Boyd et al., 2010)
malicious script execution (Saripalli and Walters, 2010)	security policy enforcement, script execution monitoring (Erlingsson et al., 2014)
XML poisoning (Saripalli and Walters, 2010)	use a local copy or a known good repository (Arnaboldi, 2016), security slicing (Thome et al., 2015)
audio steganography (Liu et al., 2011a)	stegAD (Liu et al., 2011a)

Table 5: Attack and Mitigations (cont...)

Attack	Mitigation
malicious file (Oberheide et al., 2008)	N-version protection (Oberheide et al., 2008)
VM escape (Subashini and Kavitha, 2011)	security framework and architecture (Subashini and Kavitha, 2011)
malicious VM (Szefer et al., 2011)	NoHype (Szefer et al., 2011)
service engine exploitation (Saripalli and Walters, 2010)	NoHype (Szefer et al., 2011)
steal authentication data (Saripalli and Walters, 2010)	access control, encryption (Jansen, 2011)
URL forgery (Yu et al., 2011)	secret token, referrer header, origin header (Barth et al., 2008)
bypass credential validation (Skrupsky et al., 2012)	web application validation extraction and synthesis (Skrupsky et al., 2012)
improper authentication (Jansen, 2011)	access control, encryption (Jansen, 2011)
XSS (Yu et al., 2011)	taint-tracking and taint-aware parsers (Stock et al., 2014)
CSRF (Somorovsky et al., 2011)	secret token, referrer header, origin header (Barth et al., 2008)
malware injection (Khalil et al., 2014)	N-version protection (Oberheide et al., 2008)
SQL injection (Gruschka and Jensen, 2010)	defensive coding, detection, runtime prevention (Shar and Tan, 2013; Dessiatnikoff et al., 2011)
Javascript injection (Provos et al., 2009)	JS Guard (Kishore et al., 2014)
OS commanding (Dessiatnikoff et al., 2011)	vulnerability detection (Dessiatnikoff et al., 2011)
LDAP injection (Modi et al., 2013a)	vulnerability detection (Shahriar et al., 2016)
XPATH injection (Saripalli and Walters, 2010)	security slicing, vulnerability detection (Thome et al., 2015)
unauthorized system file access (Ganesan et al., 2012)	log analysis (Ganesan et al., 2012)
brute-force (Ristenpart et al., 2009)	obfuscate internal structure and placement policy (Ristenpart et al., 2009)

Table 5: Attack and Mitigations (cont...)

Attack	Mitigation
man-in-the-middle (Subashini and Kavitha, 2011)	encryption, policy (Stojmenovic and Wen, 2014)
side-channel (Zhang et al., 2012a)	avoid co-residency, core scheduling (Zhang et al., 2012a)
invalid client side validation (Skrupsky et al., 2012)	web application validation extraction and synthesis (Skrupsky et al., 2012)
manipulate SOAP message (Gruschka and Iacono, 2009)	defensive coding, detection, runtime prevention (Shar and Tan, 2013; Dessiatnikoff et al., 2011)
modify XML (Saripalli and Walters, 2010)	use a local copy or a known good repository (Arnaboldi, 2016), security slicing (Thome et al., 2015)
alter direct object reference (Bleikertz et al., 2014)	Cloud Radar (Bleikertz et al., 2014)
passive sniffing (Duncan et al., 2013)	detection, encryption (Duncan et al., 2013)
scan open port (Modi et al., 2013b)	detection (Sengaphay et al., 2016)
visible WSDL interface (Masood, 2013)	security framework, security extension (Shahgholi et al., 2011)
impersonating (Duman et al., 2016)	EmailProfiler (Duman et al., 2016)
ARP spoofing (Wu et al., 2010)	reliable ARP table (Kang et al., 2015), detection, encryption, filtering (Alani, 2016)
DNS spoofing (Xu et al., 2013)	detection, encryption, filtering (Alani, 2016; He et al., 2017)
IP spoofing (Subashini and Kavitha, 2011)	detection, encryption, filtering (Alani, 2016)
phishing (Jensen et al., 2009)	access control, encryption, data sanitization (Jansen, 2011)

6.2.1. Cloud Components

Figure 5 shows a public IaaS setup for a hospital medical record management application. The cloud service provider (CSP) has two hosts, Host1 and Host2, where the record management application is deployed. The hosts are connected through a physical switch and a load balancer that distributes the incoming traffic between them. Each host has a physical storage device, hypervisor, and two virtual machines. Host2 act as a backup for Host1 as such the virtual machines in Host2 run the same applications as Host1. The two virtual machines in each host are connected through a virtual switch.

The medical record application consists of two main components deployed in two separate virtual machines. The components are MedScan and a MySQL database server. The virtual machine instance that hosts the MedScan application in Host1 is running windows 7 while MySQL Server is hosted on an Ubuntu VM. In Host2 both the MedScan application and MySQL Server run on an RHEL virtual machines.

MedScan is the front end of the application which acts as a web server. It receives two types of access requests through the internet: database administration and user requests. The hospital IT administration staff issues database administration requests related to managing user accounts for accessing medical records including assigning and revoking privileges for accessing medical records.

The main users of the medical record system are patients, nurses, and doctors. Patients access the system to book appointments with their doctors. Nurses access medical records to record basic vital signs parameters (such as temperature, blood pressure, etc) when a patient visits the hospital either on emergency or on a scheduled appointment. Through the medical records management system, a doctor checks a patient's medical history (previous ailments and prescriptions) during consultations. The cloud service provider (CSP) admin staff also access the hosts through a *hypervisor management console*, which is connected through the network established by the physical switch. Through this console, CSP staff are able to create, delete, migrate, and modify virtual machines in the hosts.

Figure 5 also shows the security control boundaries and component visibility in the deployment of the public IaaS cloud. Virtual machines and the applications installed in them are visible to the hospital IT administration staff. The CSP staff has control and visibility of the virtual machines (external), hypervisor, virtual switch, physical disk, bare metal, physical switch, and load balancer. Note that the visibility of the virtual machine for CSP staff is limited - they are not able to see the internal configurations of the VMs.

6.2.2. Vulnerabilities and Attack Scenarios

In this example, we focus on vulnerabilities related to the operating systems in the virtual machines to illustrate the identification of threats. We use the CVE Details (https://www.cvedetails.com/) as a data source for the latest vulnerabilities. As an illustration, we examine one of the vulnerabilities in the VM running Windows 7 virtual machine that hosts the MedScan application.

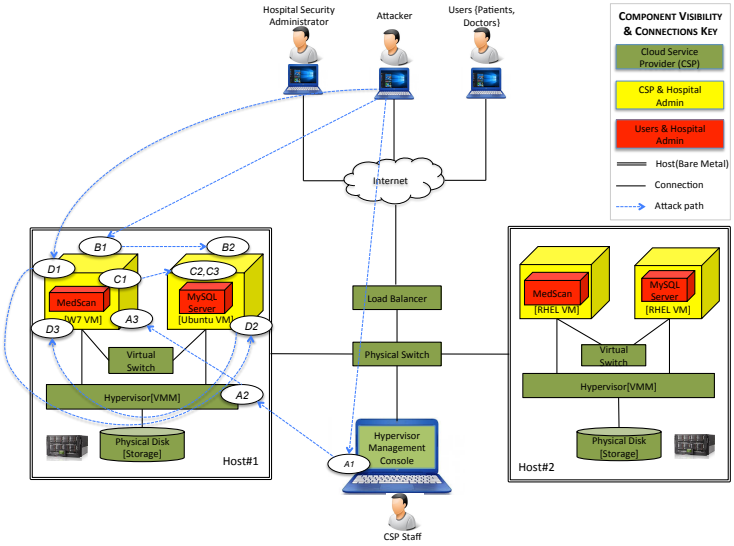

Figure 5: Public IaaS Cloud Setup System Model Example showing security boundaries between cloud provider and customer

Windows 7 SP1 Remote Code Execution Vulnerability(CVE-2017-8589) allows a "remote code execution" (Shar et al., 2015) due to the way that Windows search handles objects in memory. The vulnerability affects confidentiality, integrity, and availability. If this vulnerability is exploited successfully, the attacker is able to execute code in the target. Unsuccessful exploitation results into the target being unreachable due to the resulting denial of service.

With a remote code execution vulnerability, an attacker is able to execute commands and/or instructions on the MedScan virtual machine. This may give him administrative privileges. Some possible attack scenarios and the threats they pose are described below. We use Table 2 to map the attack scenarios to attack categories and then to the STRIDE model which helps us identify the threats. Table 6 described the steps in the attack scenarios corresponding to the labels in Figure 5 that indicate the components in which the attacks are applicable.

(A) *Manipulating VM Configuration(Cheng et al., 2012; Brohi et al., 2012)*: With admin privileges, the attacker can manipulate MedScan VM configuration such as disabling ports for incoming connections or changing firewall rules. This would render the MedScan VM not available to the users (doctors and patients) requesting access to medical records - resulting in a denial of service. The admin privileges can also be used to maliciously suspend legitimate user accounts

Table 6: Attack Steps

Attack Name	Attack Steps
(A) Manipulating VM Configuration	**A1**: Compromises console to executes hypervisor management software; **A2**: Changes firewall rules in hypervisor such that all network traffic destined for MedScan VM is rejected; **A3**: Suspends user's accounts on MedScan VM thus denying doctors and patients access to medical records.
(B) Creating Illegitimate Credentials	**B1**: Exploits trust relationship MedScan and MySQL VMs to steal MySQL Administration application credentials; **B2**: Launches MySQL administration program and create a fake user account.
(C) Unauthorized Altering of Medical Records	**C1**: Logins to the MySQL data with illegitimate user account credential created in B2; **C2**: Selects a victim medical record through a query of the medical records database; **C3**: Alters the medical record by making unauthorized entries.
(D) Using Stolen Credentials	**D1**: Using admin privileges, logins into the MYSQL user accounts management console; **D2**: Selects an existing user account as a victim and changes its password; **D3**: Logins into MedScan using the altered credentials performing actions that impersonate the legitimate account owner.

leading to another form of denial of service. This scenario maps to *Abuse Functionality*, and the most significant threats posed by attacks in this category are *Information Disclosure(Saripalli and Walters, 2010; Cheng et al., 2012; Brohi et al., 2012; Jansen, 2011; Modi et al., 2013b), and Denial of Service (Chapade et al., 2013a; Osanaiye et al., 2016; Alani, 2016; Jansen, 2011; Modi et al., 2013b).*

(B) *Creating Illegitimate Credentials*: Administrator credentials set in the Windows 7 OS can be used to the connect to the MySQL database to create an illegitimate user account. The account may later be used for unauthorized access to medical records thus breaching confidentiality. The attacker leverages on the trust relationship that could exist between the MedScan VM and the MySQL VM, and thus this attack scenario falls under *Exploitation of Authentication* category. Attacks in this category mainly pose *Spoofing(Saripalli and Walters, 2010; Yu et al., 2011; Skrupsky et al., 2012; Jansen, 2011; Modi et al., 2013b), Repudiation(Saripalli and Walters, 2010; Yu et al., 2011; Skrupsky et al., 2012; Jansen, 2011; Modi et al., 2013b), Information Disclosure(Saripalli and Walters, 2010; Yu et al., 2011; Skrupsky et al., 2012; Jansen, 2011; Modi et al., 2013b) , and Elevation of Privileges(Saripalli and Walters, 2010; Yu et al.,*

2011; Skrupsky et al., 2012; Jansen, 2011; Modi et al., 2013b) threats.

(C) *Unauthorized Altering of Medical Records*: Once an adversary has control of the MedScan virtual machine he may alter(Saripalli and Walters, 2010) the medical records stored using his illegitimate user account created in the previous attack. For example, a patient with a medical condition that would disqualify him from driving (such as epilepsy) could have the condition removed by the attacker from his medical record to allow the patient to get medical clearance for a driving job. This would compromise the integrity of the medical record database. This attack scenario is in the *Resource Manipulation* category. This attack poses a *Tampering(Gruschka and Iacono, 2009; Nasridinov et al., 2012; Saripalli and Walters, 2010; Bleikertz et al., 2014; Yu et al., 2011)* threat.

(D) *Using Stolen Credentials*: With the ability to execute commands for performing operations on the medical record database remotely, an attacker can log in into the system using stolen credentials(Saripalli and Walters, 2010) of a legitimate user and perform actions as if they are the actual user, thus impersonating the victim. For example, a fake prescription can be issued as if done by a legitimate doctor for a given patient. This is an *Exploitation of Authentication* attack and it poses *Spoofing, Repudiation(Saripalli and Walters, 2010; Yu et al., 2011; Skrupsky et al., 2012; Jansen, 2011; Modi et al., 2013b), Repudiation, Information Disclosure, and Elevation of Privileges(Saripalli and Walters, 2010; Yu et al., 2011; Skrupsky et al., 2012; Jansen, 2011; Modi et al., 2013b)* threats.

From the above analysis, we observe that the single Windows 7 vulnerability poses several threats. It shows that our proposed approach can be used to identify the threats using the mapping between attack and threat categories. An analysis of the threats posed by vulnerabilities in the rest of the components in the public IaaS deployment can be done in a similar way as described above.

7. Discussion and Future Work

This section presents findings in our work, the limitations of the study we conducted, and the potential pointers for further work to advance the ideas we propose in the cloud security.

7.1. Coverage of attacks

We have enumerated the different types of attacks that pose threats to the cloud. However, it is impractical to report all possible types of attacks due to (1) limited

resources, and (2) an ever-growing number of new attacks in the cloud environment. We believe that we have enumerated all the critical attacks to the cloud and categorized them in this paper, but readers should be aware that the list provided is not a complete set due to the above two reasons.

Subsequently, new and emerging or unknown attacks and their threats cannot be included in this survey due to the lack of means to profile their properties and consequences. However, the systematic analysis of the attacks we have adopted in this paper is a blueprint that can be applied for eliciting and categorizing the unknown attacks and threats. Hence emerging and unknown attacks can be categorized using the methods and techniques for the classification that we have proposed. Using the literature on the threats in the cloud, we have developed a method for identifying the threats by mapping the vulnerabilities to attacks and the attacks to threats. Updating the knowledge base of the mappings between vulnerabilities, attacks, and threats should enable our techniques to capture the new and emerging attacks into our threat classification and identification approaches.

An important aspect of handling the attacks is to identify how to detect and mitigate against them. The dynamic nature of the cloud implies a potential for the emergence of the unknown threats. In order to deal with such new and emerging threats, the detection and countermeasure techniques for the cloud need to fulfill a critical security requirement - they need to be generic enough to cover a wider range of attacks of a particular category, and yet specific enough to be effective in detecting and mitigating against the individual attacks.

The categorization of attacks and threats that we have proposed is the genesis for the methods of characterizing the attack detection and countermeasure techniques. We envision that through such characterization, it would be possible to cover more attacks including those that were unknown at the time of compiling our survey results - unless the attack is fundamentally different and exploit attack surfaces that we did not consider in our attack categories. To investigate, the proposed approach can be used in various practical scenarios of the cloud computing. This is one of the future directions we are exploring as an extension of this work.

7.2. Attacks against other aspects of the Cloud

We've focused mainly on how the attacks can impact the cloud in terms of the cloud components, which belongs to the cloud provider entity as described in Section 3.1. However, we must also consider the security concerns with other entities (e.g., the cloud broker with insufficient service arbitrage, invalid or malicious cloud auditors, etc).

The consideration of the security concerns related to other entities (other than the

cloud provider) that could interact with the cloud is important, as it can result in the consideration of a larger attack surface and threats. A security assessment without the consideration of these other entities is incomplete. Since the roles and functions of the service provider and other entities are different, the security assessment from the perspective of the other entities may be different.

For example, a cloud provider for a storage service, such as Google Docs, has the key security requirement of ensuring that the confidentiality of the data stored by its clients is enforced (i.e., sensitive documents stored in the service are only to be accessible by the owners of such documents). On the other hand, a cloud auditor assessing the extent to which the cloud provider complies with this security requirement may need to have access to the client's sensitive data. Such access by a third party violates the confidentiality agreement that the cloud provider has with its clients. There is no guarantee that the cloud auditor will not violate the service agreement.

Given the above scenario, the question is whether the attacks and threats related to other entities and their components can be analyzed in the same way as we have done from the provider's perspective. In particular, we need to investigate the differences in analyzing the threats from the different perspectives (e.g., the cloud provider, user, auditor etc). Moreover, the applicability of the attack and threat categories with respect to different perspectives of the security risk analysis of the cloud.

7.3. Threat classification and attack categorization

We have utilized Microsoft's STRIDE threat model to classify threats and OWASP attack categorization. However, there are other means to categorize those attacks and threats. The origin of the STRIDE threat model is based on traditional IT systems. A question that could be asked is whether this threat classification approach is sufficient for the cloud threats (i.e., can all the cloud threats be captured using the STRIDE threat model?). In this study, we have used the STRIDE threat model without answering this question. Are there other threat categorization methods we could have used, and how are they different compared to the STRIDE threat model? If the STRIDE threat model is not suitable, what are the characteristics of a threat classification method that would be appropriate for the cloud? Similar questions are relevant to the OWASP attack categorization. However, such questions will be answered in our future work.

Another key aspect of the threat classification and attack categorization is that there is no automated process of collecting and sorting raw data. In this paper, the process has been completed manually, which is time consuming to ensure all the data

sources are checked and they are correctly sorted. To improve this, an automated process is needed. However, this will be explored in our future work.

8. Conclusion

The cloud overcomes many limitations of the traditional network, such as scalability and adaptability, by simplifying the resource management and control, as well as reducing the cost of implementations. However, the new infrastructure brings various threats, both existing and new, ultimately increasing the complexity of security management. In this paper, we have systematically categorized the attacks in the cloud using the OWASP attack categories, mapped these attacks to the cloud threats through the STRIDE threat model, and also mapped the attacks to the cloud components and their associated vulnerabilities. Using this mapping approach, we have proposed a tracing method for identifying the threats in the cloud. We have reviewed statistics on the research efforts into the different attacks and compared the research efforts to security incidents on these attacks reported in practice. The results show that although there is a extensive research that has been conducted in the field of cloud security, there are still new types of attack incidents that resulted from the exploitation of unknown vulnerabilities. Hence, further research is needed to consider the new types of attack incidents to capture their posed threats in the cloud.

Acknowledgment

This paper was made possible by Grant NPRP8-531-1-111 from Qatar National Research Fund (QNRF). The statements made herein are solely the responsibility of the authors.

References

Alani, M., 2016. About Cloud Security. Springer International Publishing, Cham, pp. 15–23.
URL http://dx.doi.org/10.1007/978-3-319-41411-9_2

Alcatel-Lucent, 2018. Alcatel-Lucent CloudBand.
URL http://www.alcatel-lucent.com/solutions/cloudband

Almutairi, A., Sarfraz, M. I., Ghafoor, A., Jan 2018. Risk-aware management of virtual resources in access controlled service-oriented cloud datacenters. IEEE Transactions on Cloud Computing 6 (1), 168–181.

Almutairi, A. A., Ghafoor, A., Sept 2014. Risk-Aware Virtual Resource Management for Multitenant Cloud Datacenters. IEEE Cloud Computing 1 (3), 34–44.

Amazon, 2018. Amazon EC2.
URL https://aws.amazon.com/ec2/

Amoroso, E., May 2014. Practical Methods for Securing the Cloud. IEEE Cloud Computing 1 (1), 28–38.

Arnaboldi, F., 2016. Assessing and Exploiting XML Schema's Vulnerabilities. Tech. rep., IOActive, Seattle, Washington, United States.

Arroyo, J., Graham, C., Oberly, J., Schimke, T., apr 2016. Implementing Enhanced Error Handling of a Shared Adapter in a Virtualized System. US Patent 9,304,849.
URL https://www.google.com/patents/US9304849

Azab, M., Mokhtar, B. M., Abed, A. S., Eltoweissy, M., Nov 2016. Smart Moving Target Defense for Linux Container Resiliency. In: Proc. of the 2nd IEEE International Conference on Collaboration and Internet Computing (CIC 2016). pp. 122–130.

Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A., 2003. Xen and the Art of Virtualization. In: Proc. of the 9th ACM Symposium on Operating Systems Principles (SOSP 2003). ACM, New York, NY, USA, pp. 164–177.
URL http://doi.acm.org/10.1145/945445.945462

Barth, A., Jackson, C., Mitchell, J., 2008. Robust Defenses for Cross-site Request Forgery. In: Proc. of the 15th ACM Conference on Computer and Communications Security (CCS 2008). CCS '08. ACM, New York, NY, USA, pp. 75–88.
URL http://doi.acm.org/10.1145/1455770.1455782

Beale, J., Deraison, R., Meer, H., Temmingh, R., Walt, C., 2002. The NESSUS project. Syngress Publishing.
URL http://www.nessus.org

Benameur, A., Evans, N. S., Elder, M. C., Aug 2013. Cloud Resiliency and Security via Diversified Replica Execution and Monitoring. In: Proc. of the 6th International Symposium on Resilient Control Systems (ISRCS 2013). pp. 150–155.

Bhushan, K., Gupta, B., 2017. Security Challenges in Cloud Computing: State-of-art. International Journal of Big Data Intelligence 4 (2).

Bindra, G. S., Singh, P. K., Kandwal, K. K., Khanna, S., June 2012. Cloud Security: Analysis and Risk Management of VM Images. In: 2012 IEEE International Conference on Information and Automation. pp. 646–651.

Bleikertz, S., Vogel, C., Groß, T., 2014. Cloud Radar: Near Real-time Detection of Security Failures in Dynamic Virtualized Infrastructures. In: Proc. of the 30th Annual Computer Security Applications Conference (ACSAC 2014). ACSAC '14.

ACM, New York, NY, USA, pp. 26–35.
URL http://doi.acm.org/10.1145/2664243.2664274

Boyd, S., Kc, G., Locasto, M., Keromytis, A., Prevelakis, V., July 2010. On the General Applicability of Instruction-Set Randomization. IEEE Transactions on Dependable and Secure Computing 7 (3), 255–270.

Brohi, S. N., Bamiah, M. A., Brohi, M. N., Kamran, R., Dec. 2012. Identifying and Analyzing Security Threats to Virtualized Cloud Computing Infrastructures. In: 2012 International Conference on Cloud Computing Technologies, Applications and Management (ICCCTAM). pp. 151–155.

Chapade, S., Pandey, K., Bhade, D., April 2013a. Securing Cloud Servers Against Flooding Based DDOS Attacks. In: Proc. of the International Conference on Communication Systems and Network Technologies (CSNT 2013). pp. 524–528.

Chapade, S. S., Pandey, K. U., Bhade, D. S., Apr. 2013b. Securing Cloud Servers Against Flooding Based DDOS Attacks. In: 2013 International Conference on Communication Systems and Network Technologies (CSNT). pp. 524–528.

Chen, Y., Khandaker, M., Wang, Z., 2017. Pinpointing Vulnerabilities. In: Proc. of the ACM on Asia Conference on Computer and Communications Security (ASIA CCS 2017). ASIA CCS '17. ACM, New York, NY, USA, pp. 334–345.
URL http://doi.acm.org/10.1145/3052973.3053033

Chen, Y., Paxson, V., Katz, R., Jan 2010. What's New About Cloud Computing Security? Tech. Rep. UCB/EECS-2010-5, EECS Department, University of California, Berkeley.
URL http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-5.html

Cheng, Y., Du, Y., Xu, J., Yuan, C., Xue, Z., Oct. 2012. Research on Security Evaluation of Cloud Computing Based on Attack Graph. In: 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems. Vol. 01. pp. 459–465.

Chung, C., Khatkar, P., Xing, T., Lee, J., Huang, D., Jul. 2013. NICE: Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems. IEEE Transactions on Dependable and Secure Computing 10 (4), 198–211.
URL http://dx.doi.org/10.1109/TDSC.2013.8

Colman-Meixner, C., Develder, C., Tornatore, M., Mukherjee, B., thirdquarter 2016. A survey on resiliency techniques in cloud computing infrastructures and applications. IEEE Communications Surveys Tutorials 18 (3), 2244–2281.

Coppolino, L., D'Antonio, S., Mazzeo, G., Romano, L., 2017. Cloud Security: Emerging Threats and Current Solutions. Computers & Electrical Engineering 59, 126 – 140.

URL http://www.sciencedirect.com/science/article/pii/S0045790616300544

d. Silva, C. M. R., d. Silva, J. L. C., Rodrigues, R. B., Campos, G. M. M., d. Nascimento, L. M., Garcia, V. C., June 2013. Security Threats in Cloud Computing Models: Domains and Proposals. In: 2013 IEEE Sixth International Conference on Cloud Computing. pp. 383–389.

Dahbur, K., Mohammad, B., Tarakji, A. B., 2011. A Survey of Risks, Threats and Vulnerabilities in Cloud Computing. In: Proc. of the International Conference on Intelligent Semantic Web-Services and Applications (ISWSA 2011). ISWSA '11. ACM, New York, NY, USA, pp. 12:1–12:6.
URL http://doi.acm.org/10.1145/1980822.1980834

Deng, M., Petkovic, M., Nalin, M., Baroni, I., July 2011. A Home Healthcare System in the Cloud–Addressing Security and Privacy Challenges. In: Proc. of the 4th IEEE International Conference on Cloud Computing (CLOUD 2011). pp. 549–556.

Dessiatnikoff, A., Akrout, R., Alata, E., Kaaniche, M., Nicomette, V., Dec 2011. A Clustering Approach for Web Vulnerabilities Detection. In: Proc. of the 17th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2011). pp. 194–203.

Dudorov, D., Stupples, D., Newby, M., Aug 2013. Probability Analysis of Cyber Attack Paths against Business and Commercial Enterprise Systems. In: 2013 European Intelligence and Security Informatics Conference. pp. 38–44.

Duman, S., Kalkan-Cakmakci, K., Egele, M., Robertson, W., Kirda, E., Jun. 2016. EmailProfiler: Spearphishing Filtering with Header and Stylometric Features of Emails. In: Proc. of the 40th IEEE Annual Computer Software and Applications Conference (COMPSAC 2016). Vol. 1. pp. 408–416.

Duncan, A., Creese, S., Goldsmith, M., Quinton, J., July 2013. Cloud Computing: Insider Attacks on Virtual Machines during Migration. In: Proc. of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2013). pp. 493–500.

Erlingsson, U., Xie, Y., Livshits, B., Fournet, C., mar 2014. Enhanced Security and Performance of Web Applications. US Patent 8,677,141.
URL https://www.google.com/patents/US8677141

Fernandes, D. A. B., Soares, L. F. B., Gomes, J. V., Freire, M. M., Inácio, P. R. M., Apr 2014. Security issues in cloud environments: a survey. International Journal of Information Security 13 (2), 113–170.
URL https://doi.org/10.1007/s10207-013-0208-7

Ganesan, R., Sarkar, S., Tewari, N., June 2012. An Independent Verification of

Errors and Vulnerabilities in SaaS Cloud. In: Proc. of the IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN 2012). pp. 1–6.

Gegick, M., Williams, L., May 2005. Matching Attack Patterns to Security Vulnerabilities in Software-intensive System Designs. SIGSOFT Software Engineering Notes 30 (4), 1–7.
URL http://doi.acm.org/10.1145/1082983.1083211

Gracia-Tinedo, R., Artigas, M., Lopez, P., June 2013. Cloud-as-a-Gift: Effectively Exploiting Personal Cloud Free Accounts via REST APIs. In: Proc. of the 6th IEEE International Conference on Cloud Computing (CLOUD 2013). pp. 621–628.

Gruschka, N., Iacono, L., July 2009. Vulnerable Cloud: SOAP Message Security Validation Revisited. In: Proc. of IEEE International Conference on Web Services (ICWS 2009). pp. 625–631.

Gruschka, N., Jensen, M., July 2010. Attack Surfaces: A Taxonomy for Attacks on Cloud Services. In: Proc. of the IEEE 3rd International Conference on Cloud Computing (CLOUD 2010). pp. 276–279.

Gupta, B., Agrawal, D., Yamaguchi, S., 2016. Handbook of Research on Modern Cryptographic Solutions for Computer and Cyber Security. Advances in Information Security, Privacy, and Ethics. IGI Global.

Gupta, B., Badve, O., Mar 2013. Enhanced CBF Packet Filtering Method to Detect DDoS Attack in Cloud Computing Environment. International Journal of Computer Science Issues 10 (2), 142–146.

Gupta, B., Badve, O., Dec 2017. Taxonomy of DoS and DDoS Attacks and Desirable Defense Mechanism in a Cloud Computing Environment. Neural Computing and Applications 28 (12), 3655–3682.

Gupta, P., Seetharaman, A., Raj, J., 2013. The Ssage and Adoption of Cloud Computing by Small and Medium Businesses. International Journal of Information Management 33 (5), 861 – 874.
URL http://www.sciencedirect.com/science/article/pii/S026840121300087X

He, Z., Zhang, T., Lee, R., June 2017. Machine Learning Based DDoS Attack Detection from Source Side in Cloud. In: Proc. of the 4th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud 2017). pp. 114–120.

Hong, J., Eom, T., Park, J., Kim, D., Dec 2014. Scalable Security Analysis using a Partition and Merge Approach in an Infrastructure as a Service Cloud. In: Proc. of the the 11th International Conference on Ubiquitous Intelligence & Computing (UIC 2014). pp. 50–57.

Houmansadr, A., Zonouz, S., Berthier, R., June 2011. A Cloud-based Intrusion De-

tection and Response System for Mobile Phones. In: Proc. of the 41st IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W 2011). pp. 31–32.

Ibrahim, B. M., Hassan, M. F., May 2015. A new customizable security framework for preventing wsdl attacks. In: 2015 International Symposium on Mathematical Sciences and Computing Research (iSMSC). pp. 24–29.

Irfan, M., Usman, M., Zhuang, Y., Fong, S., Dec 2015. A Critical Review of Security Threats in Cloud Computing. In: Proc. of the 3rd International Symposium on Computational and Business Intelligence (ISCBI 2015). pp. 105–111.

Jansen, W., Jan 2011. Cloud Hooks: Security and Privacy Issues in Cloud Computing. In: Proc. of the 44th Hawaii International Conference on System Sciences (HICSS 2011). pp. 1–10.

Jensen, M., Schwenk, J., Gruschka, N., Iacono, L. L., Sep. 2009. On Technical Security Issues in Cloud Computing. In: 2009 IEEE International Conference on Cloud Computing. pp. 109–116.

Jin, C., Srivastava, A., Zhang, Z. L., April 2016. Understanding Security Group Usage in a Public IaaS Cloud. In: Proc. of the 35th Annual IEEE International Conference on Computer Communications (INFOCOM 2016). pp. 1–9.

Jouini, M., Rabai, L. B. A., Nov 2014. Surveying and Analyzing Security Problems in Cloud Computing Environments. In: 2014 Tenth International Conference on Computational Intelligence and Security. pp. 689–693.

Kang, H., Son, J., Hong, C., Aug 2015. Defense Technique Against Spoofing Attacks Using Reliable ARP Table in Cloud Computing Environment. In: Proc. of the 17th Asia-Pacific Network Operations and Management Symposium (APNOMS 2015). pp. 592–595.

Khalil, I., Khreishah, A., Azeem, M., Feb. 2014. Cloud Computing Security: A Survey. Computers 3 (1), 1–35.
URL http://www.mdpi.com/2073-431X/3/1/1

Khorshed, M. T., Ali, A. S., Wasimi, S. A., 2012. A Survey on Gaps, Threat Remediation Challenges and Some Thoughts for Proactive Attack Detection in Cloud Computing. Future Generation Computer Systems 28 (6), 833 – 851, including Special sections SS: Volunteer Computing and Desktop Grids and SS: Mobile Ubiquitous Computing.
URL http://www.sciencedirect.com/science/article/pii/S0167739X12000180

Kishore, K., Mallesh, M., Jyostna, G., Eswari, P., Sarma, S., Feb 2014. Browser JS Guard: Detects and Defends Against Malicious JavaScript Injection Based Drive by Download Attacks. In: Proc. of the 5th International Conference on the

Applications of Digital Information and Web Technologies (ICADIWT 2014). pp. 92–100.

Kivity, A., Kamay, Y., Laor, D., Lublin, U., Liguori, A., 2009. KVM: The Linux Virtual Machine Monitor. In: Proc. of the Linux Symposium. pp. 225–230.

Ko, R., Lee, S., Rajan, V., 2013. Cloud Computing Vulnerability Incidents: A Statistical Overview. Tech. rep., Cloud Security Alliance Cloud Vulnerabilities Working Group.
URL https://cloudsecurityalliance.org/download/cloud-computing-vulnerability-incidents-a-statistical-overview/

Kordy, B., Pietre-Cambacedes, L., Schweitzer, P., 2013. DAG-Based Attack and Defense Modeling: Don't Miss the Forest for the Attack Trees. CoRR abs/1303.7397.

LeBlanc, D., Howard, M., 2002. Writing Secure Code, 2nd Edition. Pearson Education.

Lenkala, S. R., Shetty, S., Xiong, K., May 2013. Security Risk Assessment of Cloud Carrier. In: 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing. pp. 442–449.

Liu, B., Xu, E., Wang, J., Wei, Z., Xu, L., Zhao, B., Su, J., Dec. 2011a. Thwarting Audio Steganography Attacks in Cloud Storage Systems. In: Proc. of the International Conference on Cloud and Service Computing (CSC 2011). pp. 259–265.

Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, M., Leaf, D., 2011b. NIST Cloud Computing Reference Architecture. Tech. rep., NIST Special Publication (NIST SP) - 500-292.
URL https://www.nist.gov/publications/nist-cloud-computing-reference-architecture

Masood, A., Nov 2013. Cyber Security for Service Oriented Architectures in a Web 2.0 World: An Overview of SOA Vulnerabilities in Financial Services. In: Proc. of the IEEE International Conference on Technologies for Homeland Security (HST 2013). pp. 1–6.

Mei, Y., Liu, L., Pu, X., Sivathanu, S., Dong, X., First 2013. Performance Analysis of Network I/O Workloads in Virtualized Data Centers. IEEE Transactions on Services Computing 6 (1), 48–63.

Mell, P., Grance, T., 2011. SP 800-145. The NIST Definition of Cloud Computing. Tech. rep., NIST, Gaithersburg, MD, United States.

Modi, C., Patel, D., Borisaniya, B., Patel, A., Rajarajan, M., 2013a. A Survey on Security Issues and Solutions at Different Layers of Cloud Computing. The Journal of Supercomputing 63 (2), 561–592.
URL http://dx.doi.org/10.1007/s11227-012-0831-5

Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., Rajarajan, M., 2013b. A Survey of Intrusion Detection Techniques in Cloud. Journal of Network and

Computer Applications 36 (1), 42 – 57.
URL http://www.sciencedirect.com/science/article/pii/S1084804512001178

Na, S. H., Park, J. Y., Huh, E. N., Dec 2010. Personal cloud computing security framework. In: 2010 IEEE Asia-Pacific Services Computing Conference. pp. 671–675.

Naskos, A., Gounaris, A., Mouratidis, H., Katsaros, P., Sept 2016. Online analysis of security risks in elastic cloud applications. IEEE Cloud Computing 3 (5), 26–33.

Nasridinov, A., Byun, J. Y., Park, Y. H., Nov 2012. UNWRAP: An Approach on Wrapping-Attack Tolerant SOAP Messages. In: 2012 2nd International Conference on Cloud and Green Computing. pp. 794–798.

Oberheide, J., Cooke, E., Jahanian, F., 2008. CloudAV: N-version Antivirus in the Network Cloud. In: Proc. of the 17th USENIX Security Symposium (USENIX Security 2008). Berkeley, CA, USA, pp. 91–106.
URL http://dl.acm.org/citation.cfm?id=1496711.1496718

Osanaiye, O., Choo, K., Dlodlo, N., 2016. Distributed Denial of Service (DDoS) Resilience in Cloud: Review and Conceptual Cloud DDoS Mitigation Framework. Journal of Network and Computer Applications 67, 147 – 165.
URL http://www.sciencedirect.com/science/article/pii/S1084804516000023

OWASP Foundation, 2018. The Open Web Application Security Project.
URL https://www.owasp.org

Pearce, M., Zeadally, S., Hunt, R., Mar. 2013. Virtualization: Issues, Security Threats, and Solutions. ACM Computing Surveys 45 (2), 17:1–17:39.
URL http://0-doi.acm.org.mylibrary.qu.edu.qa/10.1145/2431211.2431216

Pham, C., Estrada, Z., Cao, P., Kalbarczyk, Z., Iyer, R., June 2014. Reliability and Security Monitoring of Virtual Machines Using Hardware Architectural Invariants. In: Proc. of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2014). pp. 13–24.

Provos, N., Rajab, M., Mavrommatis, P., Apr. 2009. Cybercrime 2.0: When the Cloud Turns Dark. Communications of the ACM 52 (4), 42–47.
URL http://doi.acm.org/10.1145/1498765.1498782

Qi, Z., Xiang, C., Ma, R., Li, J., Guan, H., Wei, D. S. L., July 2017. ForenVisor: A Tool for Acquiring and Preserving Reliable Data in Cloud Live Forensics. IEEE Transactions on Cloud Computing 5 (3), 443–456.

Rezvani, M., Sekulic, V., Ignjatovic, A., Bertino, E., Jha, S., Nov 2015. Interdependent Security Risk Analysis of Hosts and Flows. IEEE Transactions on Information

Forensics and Security 10 (11), 2325–2339.

Riley, R., Jiang, X., Xu, D., Oct 2010. An Architectural Approach to Preventing Code Injection Attacks. IEEE Transactions on Dependable and Secure Computing 7 (4), 351–365.

Ristenpart, T., Tromer, E., Shacham, H., Savage, S., 2009. Hey, You, Get off of My Cloud: Exploring Information Leakage in Third-party Compute Clouds. In: Proc. of the 16th ACM Conference on Computer and Communications Security (CCS 2009). CCS '09. ACM, New York, NY, USA, pp. 199–212.
URL http://doi.acm.org/10.1145/1653662.1653687

Ron, A., Shulman-Peleg, A., Puzanov, A., Mar 2016. Analysis and Mitigation of NoSQL Injections. IEEE Security Privacy 14 (2), 30–39.

Roy, A., Sarkar, S., Ganesan, R., Goel, G., Feb. 2015. Secure the Cloud: From the Perspective of a Service-Oriented Organization. ACM Comput. Surv. 47 (3), 41:1–41:30.
URL http://0-doi.acm.org.mylibrary.qu.edu.qa/10.1145/2693841

Saini, V., Duan, Q., Paruchuri, V., 2008. Threat Modeling using Attack Trees. Journal of Computer Science in Colleges 23 (4), 124–131.
URL http://dl.acm.org/citation.cfm?id=1352079.1352100

Saripalli, P., Walters, B., July 2010. QUIRC: A Quantitative Impact and Risk Assessment Framework for Cloud Security. In: Proc. of the 3rd IEEE International Conference on Cloud Computing (CLOUD 2010). pp. 280–288.

Sengaphay, K., Saiyod, S., Benjamas, N., 2016. Creating Snort-IDS Rules for Detection Behavior Using Multi-sensors in Private Cloud. Springer Singapore, Singapore, pp. 589–601.
URL https://doi.org/10.1007/978-981-10-0557-2_58

Sgandurra, D., Lupu, E., Feb. 2016. Evolution of attacks, threat models, and solutions for virtualized systems. ACM Comput. Surv. 48 (3), 46:1–46:38.
URL http://doi.acm.org/10.1145/2856126

Shahgholi, N., Mohsenzadeh, M., Seyyedi, M., Qorani, S., Oct 2011. A New SOA Security Framework Defending Web Services against WSDL Attacks. In: Proc. of the 3rd IEEE International Conference on Privacy, Security, Risk and Trust (PASSAT 2011) and 3rd IEEE International Conference on Social Computing (SocialCom 2011). pp. 1259–1262.

Shahriar, H., Haddad, H., Bulusu, P., June 2016. OCL Fault Injection-Based Detection of LDAP Query Injection Vulnerabilities. In: Proc. of the 40th IEEE Annual Computer Software and Applications Conference (COMPSAC 2016). Vol. 2. pp. 455–460.

Shaikh, F. B., Haider, S., Dec 2011. Security Shreats in Cloud Computing. In: 2011

International Conference for Internet Technology and Secured Transactions. pp. 214–219.

Shan, Z., Ren, K., Blanton, M., Wang, C., Feb. 2018. Practical secure computation outsourcing: A survey. ACM Comput. Surv. 51 (2), 31:1–31:40.
URL http://0-doi.acm.org.mylibrary.qu.edu.qa/10.1145/3158363

Shar, L., Tan, H., March 2013. Defeating SQL Injection. IEEE Computer 46 (3), 69–77.

Shar, L. K., Briand, L. C., Tan, H. B. K., Nov 2015. Web application vulnerability prediction using hybrid program analysis and machine learning. IEEE Transactions on Dependable and Secure Computing 12 (6), 688–707.

Singh, P., Manickam, S., Rehman, S. U., Oct. 2014. A survey of mitigation techniques against Economic Denial of Sustainability (EDoS) attack on cloud computing architecture. In: 2014 3rd International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions). pp. 1–4.

Skrupsky, N., Monshizadeh, M., Bisht, P., Hinrichs, T., Venkatakrishnan, V., Zuck, L., Dec 2012. WAVES: Automatic Synthesis of Client-Side Validation Code for Web Applications. In: Proc. of the International Conference on Cyber Security (CyberSecurity 2012). pp. 46–53.

Somorovsky, J., Heiderich, M., Jensen, M., Schwenk, J., Gruschka, N., Lo Iacono, L., 2011. All Your Clouds Are Belong to Us: Security Analysis of Cloud Management Interfaces. In: Proc. of the 3rd ACM Workshop on Cloud Computing Security Workshop (CCSW 2011). CCSW '11. ACM, New York, NY, USA, pp. 3–14.
URL http://doi.acm.org/10.1145/2046660.2046664

Sookhak, M., Gani, A., Talebian, H., Akhunzada, A., Khan, S. U., Buyya, R., Zomaya, A. Y., May 2015. Remote Data Auditing in Cloud Computing Environments: A Survey, Taxonomy, and Open Issues. ACM Comput. Surv. 47 (4), 65:1–65:34.
URL http://0-doi.acm.org.mylibrary.qu.edu.qa/10.1145/2764465

Spring, J., March 2011. Monitoring cloud computing by layer, part 1. IEEE Security Privacy 9 (2), 66–68.

Squicciarini, A., Sundareswaran, S., Lin, D., July 2010. Preventing Information Leakage from Indexing in the Cloud. In: Proc. of the 3rd IEEE International Conference on Cloud Computing (CLOUD 2010). pp. 188–195.

Stergiou, C., Psannis, K., Kim, B., Gupta, B., 2018. Secure integration of IoT and Cloud Computing. Future Generation Computer Systems 78, 964 – 975.

Stock, B., Lekies, S., Mueller, T., Spiegel, P., Johns, M., 2014. Precise Client-side Protection against DOM-based Cross-Site Scripting. In: Proc. of the 23rd USENIX Security Symposium (USENIX Security 2014). USENIX Association,

San Diego, CA, pp. 655–670.
URL https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/stock

Stojmenovic, I., Wen, S., Sept 2014. The Fog Computing Paradigm: Scenarios and Security Issues. In: Proc. of the Federated Conference on Computer Science and Information Systems (FedCSIS 2014). pp. 1–8.

Subashini, S., Kavitha, V., 2011. A Survey on Security Issues in Service Delivery Models of Cloud Computing. Journal of Network and Computer Applications 34 (1), 1 – 11.
URL http://www.sciencedirect.com/science/article/pii/S1084804510001281

Szefer, J., Keller, E., Lee, R., Rexford, J., 2011. Eliminating the Hypervisor Attack Surface for a More Secure Cloud. In: Proc. of the 18th ACM Conference on Computer and Communications Security (CCS 2011). ACM, New York, NY, USA, pp. 401–412.
URL http://doi.acm.org/10.1145/2046707.2046754

Tang, J., Cui, Y., Li, Q., Ren, K., Liu, J., Buyya, R., Jun. 2016. Ensuring security and privacy preservation for cloud data services. ACM Comput. Surv. 49 (1), 13:1–13:39.
URL http://0-doi.acm.org.mylibrary.qu.edu.qa/10.1145/2906153

Thome, J., Shar, L., Briand, L., Nov 2015. Security Slicing for Auditing XML, XPath, and SQL Injection Vulnerabilities. In: Proc. of the 26th IEEE International Symposium on Software Reliability Engineering (ISSRE 2015). pp. 553–564.

Tunc, C., Fargo, F., Al-Nashif, Y., Hariri, S., Hughes, J., Sept 2014. Autonomic resilient cloud management (arcm) design and evaluation. In: 2014 International Conference on Cloud and Autonomic Computing. pp. 44–49.

Vaquero, L., Rodero-Merino, L., Moran, D., 2011. Locking the Sky: A Survey on IaaS Cloud Security. Journal of Computing 91 (1), 93–118.
URL http://dx.doi.org/10.1007/s00607-010-0140-x

Varadharajan, V., Tupakula, U., Jun. 2012. TREASURE: Trust Enhanced Security for Cloud Environments. In: 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications. pp. 145–152.

Varadharajan, V., Tupakula, U., July 2017. On the design and implementation of an integrated security architecture for cloud with improved resilience. IEEE Transactions on Cloud Computing 5 (3), 375–389.

Vidalis, S., Jones, A., 2003. Using Vulnerability Trees for Decision Making in Threat Assessment. In: Proc. of the 2nd European Conference on Information Warfare and Security (ECIW 2003). Academic Conferences Limited, pp. 329–342.

Vieira, K., Schulter, A., Westphall, C., Westphall, C., 2010. Intrusion Detection for Grid and Cloud Computing. IT Professional 12 (4), 38–43.

VMWare, 2009. VMWARE ESX Server.

Wang, C., Wang, Q., Ren, K., Lou, W., March 2010. Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing. In: Proc. of the IEEE International Conference on Computer Communications (INFOCOM 2010). pp. 1–9.

Wu, H., Ding, Y., Winer, C., Yao, L., Nov 2010. Network Security for Virtual Machine in Cloud Computing. In: Proc. of the 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT 2010). pp. 18–21.

Xiao, Z., Xiao, Y., Second 2013. Security and privacy in cloud computing. IEEE Communications Surveys Tutorials 15 (2), 843–859.

Xu, D., Tu, M., Sanford, M., Thomas, L., Woodraska, D., Xu, W., July 2012. Automated Security Test Generation with Formal Threat Models. IEEE Transactions on Dependable and Secure Computing 9 (4), 526–540.

Xu, L., Li, L., Nagarajan, V., Huang, D., Tsai, W., March 2013. Secure Web Referral Services for Mobile Cloud Computing. In: Proc. of the 7th IEEE International Symposium on Service-Oriented System Engineering (SOSE 2013). pp. 584–593.

Yan, Q., Yu, F. R., Gong, Q., Li, J., 2016. Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges. IEEE Communications Surveys Tutorials 18 (1), 602–622.

Yu, Y., Yang, Y., Gu, J., Shen, L., Dec. 2011. Analysis and Suggestions for the Security of Web Applications. In: Proc. of the International Conference on Computer Science and Network Technology (ICCSNT 2011). Vol. 1. pp. 236–240.

Yusof, I., Pathan, A. S. K., Mar 2016. Mitigating cross-site scripting attacks with a content security policy. Computer 49 (3), 56–63.

Zhang, Y., Juels, A., Reiter, M., Ristenpart, T., 2012a. Cross-VM Side Channels and Their Use to Extract Private Keys. In: Proc. of the ACM Conference on Computer and Communications Security (CCS 2012). CCS '12. ACM, New York, NY, USA, pp. 305–316.
URL http://doi.acm.org/10.1145/2382196.2382230

Zhang, Y., Li, M., Bai, K., Yu, M., Zang, W., 2012b. Incentive Compatible Moving Target Defense against VM-Colocation Attacks in Clouds. In: Gritzalis, D., Furnell, S., Theoharidou, M. (Eds.), Information Security and Privacy Research. Vol. 376 of IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg, pp. 388–399.
URL http://dx.doi.org/10.1007/978-3-642-30436-1_32

Zhang, Y., Reiter, M. K., 2013. DuPpel: Retrofitting Commodity Operating Systems to Mitigate Cache Side Channels in the Cloud. In: Proc. of the 2013 ACM SIGSAC Conference on Computer & Communications Security. CCS '13. ACM, New York, NY, USA, pp. 827–838.
URL http://doi.acm.org/10.1145/2508859.2516741

Jin B. Hong is a lecturer in the Department of Computer Science and Software Engineering at the University of Western Australia, Australia. He received his PhD degree in Computer Science from the University of Canterbury, New Zealand. His research interests are security modeling and analysis of computer and networks including cloud computing, SDN and IoT, and Moving Target Defense.

Armstrong Nhlabatsi is a post doctoral researcher at Qatar University, Qatar. He received his PhD degree in Computer Science from the Open University, United Kingdom. His research interests include security requirements engineering, security risk evaluation, requirements traceability, and the feature interaction problem for information security. He previously worked at the University of Swaziland as a lecturer in programming techniques and digital electronics.

Dong Seong Kim is an Associate Professor in Cybersecurity in the School of Information Technology and Electrical Engineering at the University of Queensland, Australia. His research interests include dependability, cybersecurity and survivability modeling and analysis.

Alaa Hussein is a graduate student in the Department of Computer Science and Engineering, KINDI Computing Research Centre at Qatar University. She has completed her undergraduate degree in computer science at Qatar University.

Noora Fetais is the director of KINDI Computing Research Centre at Qatar University. She received her PhD in computer engineering from the University of Sussex. She is currently holding various positions such as Vice Chair of IEEE-Qatar Section, Qatar Ambassador of Women in Data Science (WiDS) at Stanford University, among others. She was the first Women to chair the Faculty Senate of Qatar University.

Khaled MD Khan is an associate professor in the department of Computer Science and Engineering and the Manager of KINDI Computing Research Lab at Qatar University. Prior to these, he served Western Sydney University (Australia) as a senior lecturer, and was the Head of postgraduate programs. He received his PhD in computing from Monash University, and BS and MS degrees both in computer science from the Norwegian University of Science and Technology.