Threat Modeling for Cloud Data Center Infrastructures

Nawaf Alhebaishi[1,2], Lingyu Wang[1], Sushil Jajodia[3], and Anoop Singhal[4]

[1] Concordia Institute for Information Systems Engineering, Concordia University
[2] Faculty of Computing and Information Technology, King Abdulaziz University
{n_alheb,wang}@ciise.concordia.ca
[3] Center for Secure Information Systems, George Mason University
jajodia@gmu.edu
[4] Computer Security Division, National Institute of Standards and Technology
anoop.singhal@nist.gov

Abstract. Cloud computing has undergone rapid expansion throughout the last decade. Many companies and organizations have made the transition from traditional data centers to the cloud due to its flexibility and lower cost. However, traditional data centers are still being relied upon by those who are less certain about the security of cloud. This problem is highlighted by the fact that there only exist limited efforts on threat modeling for cloud data centers. In this paper, we conduct comprehensive threat modeling exercises based on two representative cloud infrastructures using several popular threat modeling methods, including attack surface, attack trees, attack graphs, and security metrics based on attack trees and attack graphs, respectively. Those threat modeling efforts provide cloud providers practical lessons and means toward better evaluating, understanding, and improving their cloud infrastructures. Our results may also imbed more confidence in potential cloud tenants by providing them a clearer picture about potential threats in cloud infrastructures and corresponding solutions.

1 Introduction

Cloud computing has emerged as an attractive option for many enterprises, government agencies and organizations due to its flexibility and reduced costs. The shifting to this new paradigm is, however, still impeded by various security concerns, which are exacerbated by the lack of a clear understanding of security threats facing cloud data centers. Unlike traditional computer networks, cloud data centers usually exhibit some unique characteristics, such as the presence of significant redundancy in terms of hardware configurations, and the co-existence of both physical and virtual components. Such unique characteristics imply the need for modeling and measuring security threats specifically for cloud data centers.

On the other hand, modeling and measuring security threats for cloud data centers is a challenging task due to the lack of public accesses to the detailed information regarding hardware and software configurations deployed in real cloud data centers. Existing work mainly focus on high level frameworks for risk and impact assessment [19], guidelines or frameworks for cloud security metrics [2, 14], and specific vulnerabilities or threats in the cloud [6, 21] (a more detailed review of related work will be given in Section 6). However, to the best of our knowledge, there lack a concrete study on threat

modeling and measuring for cloud data centers using realistic cloud infrastructures and well established models. Although there already exist many such threat modeling models, such as attack surface, attack tree, attack graph, and their corresponding security metrics, a systematic application of those models to concrete cloud data center infrastructures is yet to be seen.

In this paper, we present a comprehensive study on modeling and measuring threats in cloud data center infrastructures. We first provide the basis for our study as two representative cloud infrastructures, devised based on established technologies of several major players on the cloud market, e.g., Amazon, Microsoft, Google, Cisco, VMware, and OpenStack. We also provide details on the hardware and software components used in the data center to manage the cloud services. We then apply several popular threat modeling methods on such cloud infrastructures, including attack surface, attack tree, attack graph, and security metrics based on attack trees and attack graphs.

The main contribution of this paper is twofold. First, to the best of our knowledge, this is the first comprehensive study of threat modeling based on well established models and concrete cloud data center designs, which incorporate technologies used by major cloud providers on the market. Second, our study provides answers to many practical questions, such as, *How can cloud providers gather and organize knowledge concerning the security of their cloud data center and services? How can cloud providers examine the security of a cloud data center at different abstraction levels? How can cloud providers measure the security of their cloud data center before and after applying a hardening option?* Those threat modeling efforts can not only provide cloud providers practical lessons and means for understanding and improving their cloud infrastructures, but may also imbed more confidence in cloud tenants by providing them a clearer picture about potential threats in cloud infrastructures.

The remainder of this paper is organized as follows. Section 2 provides the background knowledge on threat modeling and security metrics needed later in our work. In Section 3, the cloud data center architecture is presented. In Section 4, the threat modeling is explained in details. In Section 5, security metrics are applied to measure the level of security. Related work are reviewed in Section 6, and the paper concluded in Section 7.

2 Background

The following briefly reviews the threat models and security metrics that are applied in this paper, including attack surface, attack tree, attack graph, attack tree-based metric (ATM), and Bayesian network (BN)-based metric.

– Attack surface: Originally proposed as a metric for software security, attack surface captures software components that may lead to potential vulnerabilities. These may include entry and exit points (i.e., methods in a software program that either take user inputs or generate outputs), communication channels (e.g., TCP or UDP), and untrusted data items (e.g., configuration files or registry keys read by the software) [15]. Due to the complexity of examining source code, most existing work applies the concept in a less formal manner. For example, between an end user, the cloud provider, and cloud services, six attack surfaces can be composed [11].

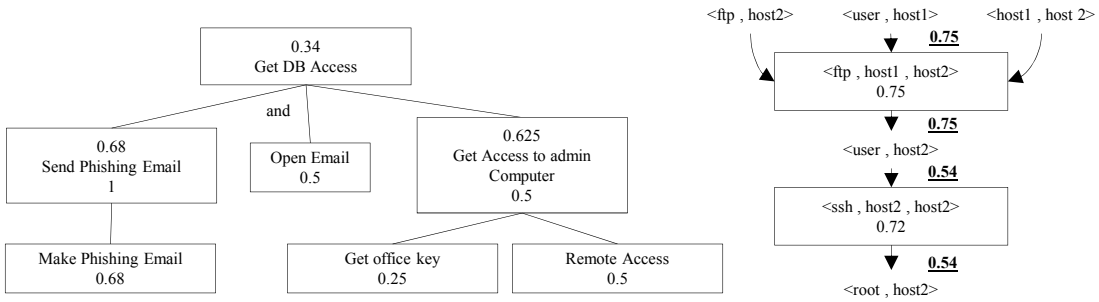

Fig. 1: Attack Tree (Left) and Attack Graph (Right)

– Attack tree: While attack surface focuses on what may provide attackers initial privileges or accesses to a system, attack trees demonstrate the possible attack paths which may be followed by the attacker to further infiltrate the system [20]. The left-hand side of Figure 1 shows an attack tree example in which the attacker's goal is to get accesses to the database. In the example, there are two ways to reach the root node (the goal). First, the attacker can follow the left and middle paths at the same time (due to the *and* label), or the attacker can follow the right path for reaching the root node.

– Attack graph: As a more fine-grained model, an attack graph depicts all possible attack steps and their causal relationships [22]. In the right-hand side of Figure 1, each triplet inside a rectangle indicates an exploit <service vulnerability, source host, destination host>, and each pair in plaintext indicates a pre- or post-condition <condition, host> of the exploits. The logic relationships between the nodes are represented based on the assumption that any exploit can be executed if and only if all of its pre-conditions are already satisfied (e.g., In Figure 1, the first exploit requires all three pre-conditions to be satisfied), whereas any condition may be satisfied by one exploit for which the former is a post-condition.

– The above threat models are all qualitative in nature. The attack tree-based metric (ATM) quantifies the threat in an attack tree using the concept of *probability of success* [8]. The probability of each node in the attack tree is typically determined based on historical data, expert opinions, or both. In Figure 1, a number above the label represents the overall probability of success, and a number below the label represents the probability of each node alone. The probability on the root node indicates the most risky path, which should be prioritized in security hardening. The BN-based metric [24, 9] can be applied to attack graphs to calculate the probability for an average attacker to compromise a critical asset. The conditional probabilities that an exploit can be executed given its pre-conditions are all satisfied can usually be estimated based on standard vulnerability scores (e.g., the CVSS scores [16]). In Figure 1, the probability inside a rectangle is the CVSS score divided by 10, and each underlined number represents the probability for successfully executing that exploit. In this example, the attack goal has a probability of 0.54, and if we change the ftp service on host2 and suppose the new probability becomes 0.4, then the new attack probability for the goal will become 0.228, indicating increased security.

3 Devising Cloud Data Center Infrastructures

In this section, we devise two cloud data center infrastructures that will be used for threat modeling in Section 4 and Section 5. To make our infrastructures more representative, we have base our infrastructures upon concepts and ideas borrowed from major players on the market, including Cisco, VMware, and OpenStack, as follows.

- Cisco presents a cloud data center design for both public and private clouds [4], which is divided into multiple layers with suggested hardware for the physical network and software used to virtualize the resources. We borrow the multi-layer concept and some hardware components, including Carrier Routing System (CRS), Nexus (7000,5000,2000), Catalyst 6500, and MDS 9000.
- VMware vSphere suggests the hardware and software components to run a private cloud data center [12]. They also tag the port numbers used to connect services together. We borrow the concepts of Authentication Server, Domain Name System(DNS), and Storage Area Network (SAN) and synthesize these to represent the main functionality of some hardware components in our cloud infrastructures.
- OpenStack is one of the most popular open source cloud operating systems [17]. We borrow following components of OpenStack: Dashboard, Nova, Neutron, Keystone, Cinder, Swift, Glance, and Ceilometer [17].

Table 1 compares the main concepts used in our infrastructures to the major cloud providers in the market, including Amazon [5], Microsoft [23], and Google [3] (some of those concepts will be discussed later in this section).

	AWS	Microsoft Azure	Google Compute
Multiple Layers	×	×	×
Authentication Sever	×	×	
Domain Name System	×	×	×
One service in each cluster	×	×	×
Multi-tier	×	×	×

Table 1: Concepts Used by Major Cloud Providers

We discuss two different infrastructures since OpenStack components can either run centrally on a single server or be distributed to multiple servers [17].

Infrastructure 1 Figure 2 illustrates our first infrastructure, which is based on concepts and technology presented by Cisco [4], VMware vSphere [12], and OpenStack [17]. The physical network provides accesses to both cloud users and cloud administrators. Cloud administrators connect to the data center through firewalls ($node$ 17) and ($node$ 19), an authentication server ($host$ 18), and Nexus 7000 ($node$ 20), which is connected to the other part of the network. For cloud users, Cisco's multi-layer concept is used [4] as follows.

- In Layer 1, a CRS ($node$ 1) is used to connect the cloud to the internet, which then connects to a firewall ($node$ 2, ASA 5500-X Series) while simultaneously being connected to two different types of servers (authentication servers ($host$ 3) as well as DNS and Neutron Servers ($node$ 4)). Those servers provide services to the cloud tenants and end users. The servers then connect to Cisco Nexus 7000 with Catalyst 6500 ($node$ 5) to route the requests to destination machines.

– In Layer 2, a firewall ($node$ 6, ASA 5500-X Series) connects the first layer to this layer through Nexus 5000 ($node$ 7). The Nexus 5000 is used to connect rack servers through Nexus 2000, which is used to connect servers inside each rack at the computing level ($hosts$ $8, 9, 10, 11,$ and 12). The Nexus 5000 ($node$ 7) then connects to the next layer.
– In Layer 3, another Nexus 7000 ($node$ 13) connects the previous layer to the storage. A firewall ($node$ 14, ASA 5500-X Series) connects the Nexus 7000 ($node$ 13) and MDS 9000 ($node$ 16).

The following outlines how the cloud works. OpenStack components run on the authentication servers among which one ($host$ 3) is designated for cloud tenants, and another ($host$ 18) for cloud administrators. The first runs following components: Dashboard, Nova, Neutron, Keystone, Cinder, Swift, Glance, and MySql. The second runs the same components, but additionally runs Ceilometer for a billing system. The DNS server ($node$ 4) runs a Neutron component that provides the address of the machine running a requested service. At the computing level ($hosts$ $8, 9, 10, 11,$ and 12), all physical servers run four components: Hypervisor, Nova to host and manage VMs, Neutron agent to connect VMs to the network, and Ceilometer agent to calculate the usage. At the computing level, each physical server cluster runs the same VMs service, e.g., all $http$ VMs run on the $http$ server cluster, and the same occurs for application VMs, ftp VMs, smtp VMs, and database VMs. Finally, all physical machines and VMs run ssh for maintenance.

Infrastructure 2 The second infrastructure is illustrated in Figure 3, which is based on concepts and technology presented by Cisco [4], VMware vSphere [12], and OpenStack [17]. This infrastructure has a similar physical network as the previous, with the addition of new machines that separate OpenStack components, which are installed on the authentication servers for cloud tenants in the previous infrastructure, into many different machines. These new machines are Neutron servers ($node$ 25), controller servers ($node$ 36), and network nodes ($node$ 34). In addition, the authentication server ($host$ 23) for cloud tenants will run a Dashboard component to access and manage the VMs related to the tenant user. Moreover, Neutron server ($node$ 25) serves to control the virtual network and connects to the controller node ($node$ 36), which runs Nova API, Neutron API, Keystone, Glance, Swift, Cinder, MySql, and any component needed to manage and control the cloud. The last node is a network node ($node$ 34) which translate between the virtual IPs and the physical IPs to grant accesses to services running on VMs. For example, if a cloud tenant wishes to access their VMs, they will first need to connect to the Dashboard. Next, the Neutron server will send the authentication request to the keystone service on the Controller node. If the user possesses the privilege to access the VM, the controller will send a request to the network node to obtain the address for the VMs, and will then send the address to the Neutron server to connect the user to their VMs.

4 Threat Modeling

This section conducts threat modeling on the two infrastructures that are just introduced.

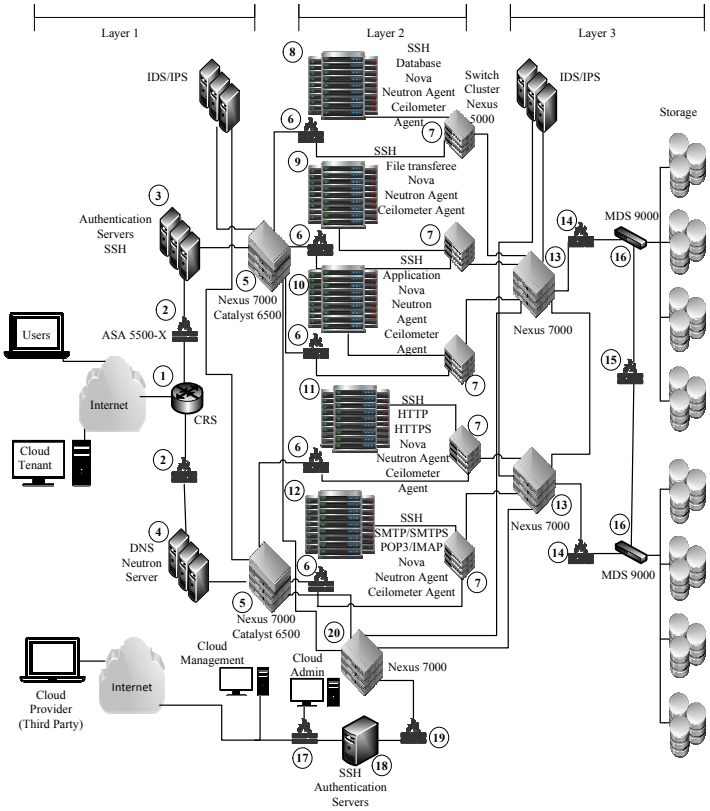

Fig. 2: Cloud Data Center Infrastructure 1

4.1 Attack Surface

In this section, we apply the attack surface concept at the resource level. Gruschka & Jensen categorize attack surfaces into those between user, service, and cloud provider [11]. The same classes are used in our discussions, with the addition of surfaces belonging to the same class. Also, we consider the service class used by Gruschka & Jensen [11] as the intermediate layer between users and the cloud provider in the sense that, if a user wishes to attack a cloud provider, he/she must pass through an attack surface consisting of services. In addition, we focus on entry and exit points [15] which indicate the means through which the attack starts and those through which data is leaked out, respectively.

In Figure 2 and 3 it can be observed that there are three types of attack surfaces in a cloud data center. First, there are attack surfaces related to the physical network, involving hardware and software components, such as switches, routers, servers, applications, and operating systems. Second, there are virtualization-related attack surfaces, such as hypervisors and virtual switches. Last, there are cloud operating systems, such as OpenStack components (Glance, Neutron, Nova, Ceilometer, and Keystone). The first type of attack surface is similar to those in traditional networks, but components related to cloud running on top of the physical network must also be considered. On the other

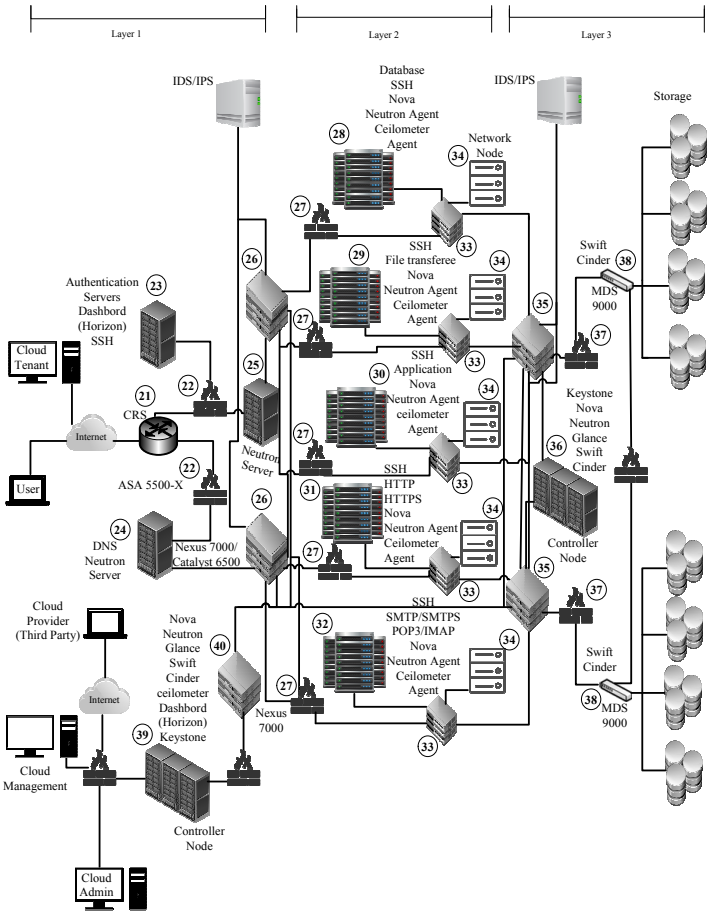

Fig. 3: Cloud Data Center Infrastructure 2

hand, virtualization and cloud operating systems-related attack surfaces are unique to cloud and their analysis will pose new challenges.

Attack Surface w.r.t. Users We consider two types of users. First, the normal user using the cloud service may aim to attack either the cloud tenant who owns that service, another cloud tenant or its users using the same cloud, or the cloud provider. Second, the cloud tenant may aim to attack another cloud tenant and its users, or the cloud provider. Various surfaces can be utilized by users to attack the cloud, including the hypervisor, VMs, APIs and web services, and OpenStack components (e.g., Horizon, Keystone, Neutron, Glance, and Nova).

Example 1. A normal user wants to attack a hypervisor on the database VM server ($host$ 8) to steal information about all VMs running on that machine. First, the entry point to start this attack is the database VM on the hypervisor. After he/she get initial accesses to the database VM, that VM become an exit point to attack the hypervisor. Finally, with accesses to the hypervisor, e.g., through exploiting CVE-2013-4344 [1], the

attacker can get data related to all VMs run on top of this hypervisor and the hypervisor thus becomes an exit point. Next consider a cloud tenant who wants to attack another tenant hosted on the same physical machine. First, the attacker can use his/her VM as entry point to get a privilege to the hypervisor, e.g., by applying CVE-2012-3515 [1], then the attacker will use the hypervisor as an entry point to get accesses to the target VM.

Attack Surface w.r.t. Cloud Providers A cloud provider here refers to an operator who has privileges to access certain components (e.g., switches, firewall, and SAN) for maintenance and management purposes. This type of attackers may use his/her accesses to resources to attack the cloud data center. All three types of attack surfaces explained before can be used by such an attacker.

Example 2. An operator who has accesses to Nexus 7000 ($node$ 13) for management wants to get accesses to sensitive data related to a tenant. First, he/she can use the Nexus 7000 as an entry point to obtain a root privilege on Nexus 7000, and then use this machine as an exit point to start another attack to get data from the storage device ($node$ 16).

4.2 Attack Tree

The previous section shows how attack surface may capture the initial attack attempts. To further study what may happen once an attacker gains initial privileges, we will need attack trees, which represent high level attack paths leading attackers to their goals. Figure 4 shows an attack tree for our cloud data center infrastructures. It is assumed that the root node, or goal node, is a storage device in the cloud that is susceptible to attacks by either a malicious user, a cloud tenant, or a cloud operator. Eight paths in Figure 4 represent the possible ways to reach such a target. Each path represents a capability level of users who can follow the path; not all paths can be used by all users. For example, some paths can be followed by the cloud operator but cannot be accessed by normal users or cloud tenants. In what follows, the paths and corresponding users will be explained in further details.

- **Path 1:** This attack can be executed by a normal user to obtain data from the storage device ($node$ 16). The user must first establish a connection to the $http$ VM server ($host$ 11) and must then acquire the root privilege on this VM. The attacker can then connect to the application VM server ($host$ 10) provided that they have obtained root privilege on that VM. After the user acquires access to the application VM, he/she may create a connection to the database VM server ($host$ 8). From this point, the user can attack the database VM to obtain root privilege on that VM. Finally, the attacker can launch an attack on the hypervisor to gain access to other database VMs ($host$ 8) running on the same physical machine and obtain data related to all database VMs stored on the storage device ($node$ 16).
- **Path 2:** The normal user can use this path to attack the cloud storage device ($node$ 38). The attacker begins the attack by surpassing the firewall ($node$ 22) to obtain privilege on OpenStack ($node$ 36) in order to gain a direct connection to the database VM server ($host$ 28). The remainder of this attack is similar to that of path 1, and serves to gain access to the hypervisor and the storage device.

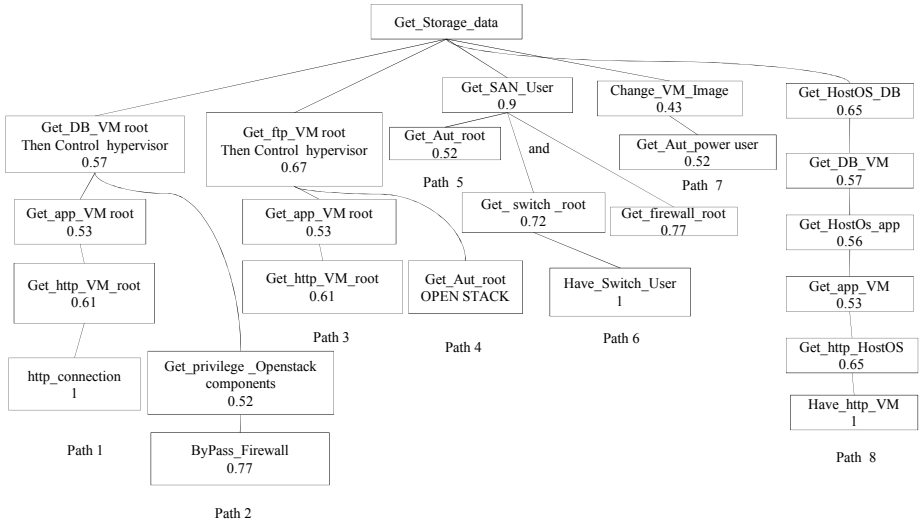

Fig. 4: Attack Tree

– **Path 3:** This path can be used by a cloud tenant user who has user access to the $http$ VM server ($host$ 11) and wishes to access ftp files stored on the storage device ($nose$ 16). First, the cloud tenant user must obtain root privilege on the $http$ VM server ($host$ 11). Then, he/she will need to obtain root privilege on the application VM server ($host$ 10) to start a connection to the ftp VM server ($host$ 9). After this, the user will obtain root privilege on this VM and get the ftp files related to this VM. In addition, the user can attack the hypervisor to obtain the ftp files related to other VMs running on top of this hypervisor.
– **Path 4:** Cloud tenants who do not already possess ftp VM servers running on the cloud can use this path to obtain data from the storage device ($node$ 16) through the ftp VM server ($host$ 9). Cloud tenants on this path will use OpenStack components ($host$ 3) to gain privileges to access the ftp VM ($host$ 9) belonging to another cloud tenant. In this situation, the attacker can obtain all files belonging to this VM. Furthermore, the attacker may attack the hypervisor to gain access to other ftp VMs running on the same physical machine.
– **Path 5:** Cloud operators with access to the admin user authentication server ($host$ 18) can use this path by obtaining root access to the authentication server. They can then use this device to obtain root access on the SAN device ($node$ 16) to control the data stored on the storage device.
– **Path 6:** This path can be used by a cloud operator who has access to a physical machine (e.g., a switch, firewall, or other type of machines) to attack the storage device. Suppose the attacker has user access to a switch device ($node$ 13) for maintaining this device. The attacker can then obtain root access on this device as well as root access to a firewall device ($node$ 14) between the switch device and the SAN ($node$ 16). These two accesses may allow him/her to create a connection to the SAN device and subsequently attack the SAN in order to access the stored data.

– **Path 7:** This path may be used by a third party cloud provider who has access to the authentication server ($host$ 18) of an administrator. The user must obtain root access on the authentication server and must then gain privilege on the VM image storage ($host$ 18) and ($node$ 16). In this case, the user may use this privilege to modify or change the VM images stored on Glance. This new image will have a backdoor that can be used by the attacker to gain access to all VMs with this image.
– **Path 8:** This path can be used by either a cloud tenant or a normal user. The goal for these attackers is to control the data belonging to other cloud tenants on the cloud. The attacker must first have access to the $http$ VM sever ($host$ 31) and must gain access to the Host Operating System (HOS) ($host$ 31). By gaining access to the HOS, the attacker can obtain access to all VMs running on this machine. The attacker may then gain access to all application VMs ($host$ 30) connected to all $http$ VMs to which they have access. Subsequently, the attacker gains access to the application VMs which may run on different physical machines; the attacker may then acquire access over all HOS related to those VMs ($host$ 30). The attacker can then gain root access to the database VM server ($host$ 28) in order to obtain the data stored on the storage device. The attacker may also gain access to all HOS running database VMs ($host$ 28).

4.3 Attack Graph

In the previous section, the attack tree shows how an attacker may follow an attack path to reach the goal. However, this is done at a higher abstraction level without details about specific vulnerabilities. We now construct attack graphs to represent specific exploits of vulnerabilities that can be use to reach the goal. Although we can apply the standard attack graph concept designed for traditional networks, special consideration must be given to the virtualization level, which is unique to cloud, and the fact that machines or VMs may have similar or identical configurations.

We construct our attack scenarios based on real vulnerabilities related to hardware and software components used in our infrastructures as listed in the National Vulnerability Database (NVD) [1]. In our attack graphs, the Common Vulnerability Scoring System (CVSS) [16] scores retrieved from the NVD are depicted inside each node after dividing it by 10 to obtain a probability value between 0 to 1, which is later used in the BN-based metric. An attack graph may be created for different types of users but we will focus on the normal user due to space limitations.

Figure 5 and Figure 6 show two attack graphs for the data center infrastructures depicted in Figure 2 and Figure 3, respectively. It is assumed that the attacker has access to a cloud tenant's services. The main goal for the attacker is to steal data from the storage. The user must have access to the $http$ VM as well as the application VM and database VM before reaching the goal due to the multi-tier infrastructure. The following services are assumed to be used in the data centers.

– Tectia Server version 5.2.3, for ssh running in all VMs.
– Apache $http$ server running on $http$ VM.
– Oracle version 10.1.0.2 installed on the application VM.
– Oracle version 10.2.1 on the database VM.

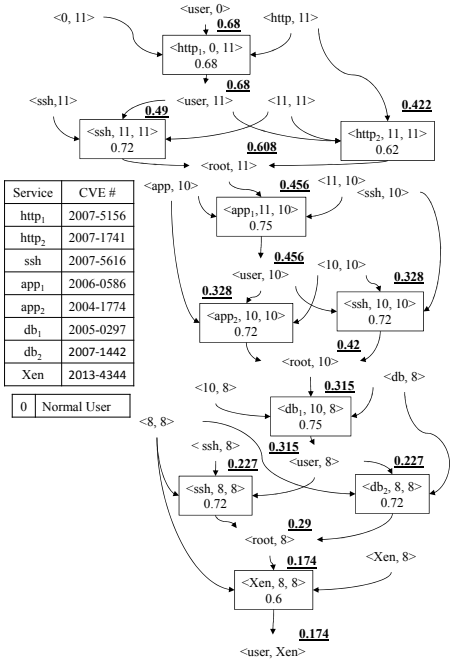

Fig. 5: Attack Graph for Figure 2

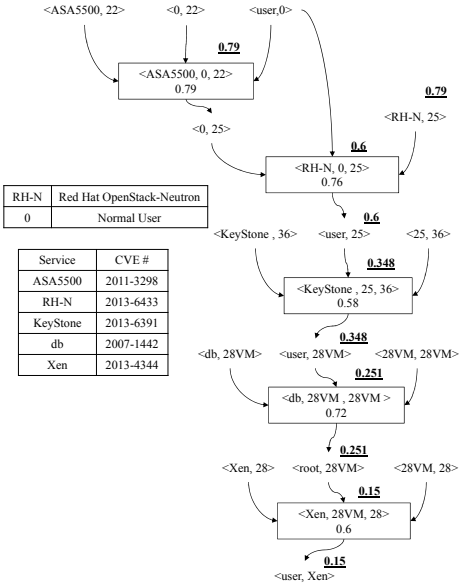

Fig. 6: Attack Graph for Figure 3

– Xen version 4.3.0 is running as a hypervisor to control VMs on physical machines.

Example 3. Figure 5 shows an attack graph corresponding to path 1 in the aforementioned attack tree. Between five to seven vulnerabilities are required to reach the goal. Specifically, five vulnerabilities are required if we assume the ssh vulnerability will be the same in the $http$ server VM, application server VM, and database server VM, whereas seven vulnerabilities are required if the ssh vulnerability is not used to reach the goal. We divide the attack graph to four stages and in each stage the attacker will gain a different level of privileges.

– **Stage 1:** A vulnerability in the $http$ server VM ($host$ 11) (CVE-2007-5156) is employed by the attacker to gain user access by uploading and executing arbitrary code containing .php. in the file extension as well as unknown extensions. Then, another vulnerability on the same VM (CVE-2007-1741) is used to gain root privilege by renaming the directory or performing symlink. A ssh ($host$ 11) vulnerability (CVE-2007-5156) can also be used to gain root privilege on the same VM.
– **Stage 2:** The attacker now can connect to the application server ($host$ 10). Then, by using a vulnerability related to the application sever VM (CVE-2006-0586), the attacker is allowed to gain the user privilege by executing arbitrary sql commands through multiple parameters. To gain root privilege on this VM, the attacker can apply this vulnerability (CVE-2004-1774) or by using an ssh ($host$ 10) vulnerability (CVE-2007-5616), and at this point the attacker can start a connection to the database server VM.

– **Stage 3:** The attacker uses a vulnerability related to the database server ($host$ 8) VM (CVE-2005-0297) to gain user access. Then, on this VM he/she can gain root access by using vulnerability (CVE-2007-1442) or an ssh ($host$ 8) vulnerability (CVE-2007-5616).
– **Stage 4:** The attacker can then obtain data related to this database VM ($host$ 8), and he/she may obtain even more data from another VM running on the same physical machine by gaining access to a hypervisor through exploiting (CVE-2013-4344).

Example 4. The attack graph in Figure 6 is related to the infrastructure shown in Figure 3, where OpenStack components run on more than one physical machine. The goal of this attack is to gain access to date storage in three stages. This attack graph corresponds to path 2 in the attack tree.

– **Stage 1:** A vulnerability in the firewall ($node$ 22) (CVE-2011-3298) is employed by the attacker to bypass the firewall in order to connect to the Neutron server ($node$ 25). The attacker can then use the Neutron vulnerability (CVE-2013-6433) to gain privileges with which he/she can use vulnerability (CVE-2013-6391) to generate EC2 token API in order to gain access to a database VM ($host$ 28).
– **Stage 2:** After the attacker obtains access to the database VM ($host$ 28), he/she used the database vulnerability (CVE-2007-1442) to gain root privilege on the same VM. This allows the attacker to obtain data related to this VM.
– **Stage 3:** To obtain data from another database on the same physical machine, the attacker used the vulnerability (CVE-2013-4344) to gain access to the hypervisor running on this physical machine such that he/she can access all VMs running on this machine and view the data related to these VMs.

By constructing the attack surface, attack tree, and attack graphs for the cloud data center infrastructures, we have demonstrated how each model may capture potential threats at a different abstraction layer. Nonetheless, all those models are qualitative in nature, and we will apply security metrics to measure the threats in the coming section.

5 Cloud Security Metric

In this section, we apply two security metrics based on the attack tree and attack graphs, respectively, to further quantify the threats modeled in the previous section.

5.1 Attack Tree Metric

In this section, an attack tree metric (ATM) will be applied based on the attack tree described in Section 4.2. In Figure 7, all nodes inside the same path are considered as having AND relationships, whereas an OR relationship is assumed between different paths unless if an AND relationship is explicitly stated. Based on such assumptions, the corresponding equations are applied to calculate the probabilities. The highest probability is assigned to the root node after applying the metric. In Figure 7, between the two probabilities in each node, the probability with (+) represents the average CVSS values and the other probability represents the metric result.

In Figure 7, it can be observed that path 5 and 6 are the least secure paths in the attack tree. Those two paths can be followed by a cloud operator to launch an insider

attack to steal data from the storage device. This metric can also be used to verify whether or not adding a new service or disabling existing services can increase security and by how much. As shown in Figure 7, the probability to reach n_8 is 0.45; as such, if the cloud provider wishes to decide whether to increase security levels in that node, he/she can use the metric before and after applying the changes. For example, suppose the cloud provider wishes to add new rules to a firewall to prevent attacks from n_9 and n_{11} to n_8. After re-applying the ATM metric, the probability on n_8 becomes 0.348, showing increased security. Applying the ATM on other potential changes may help the cloud provider to make the right decisions in hardening the cloud.

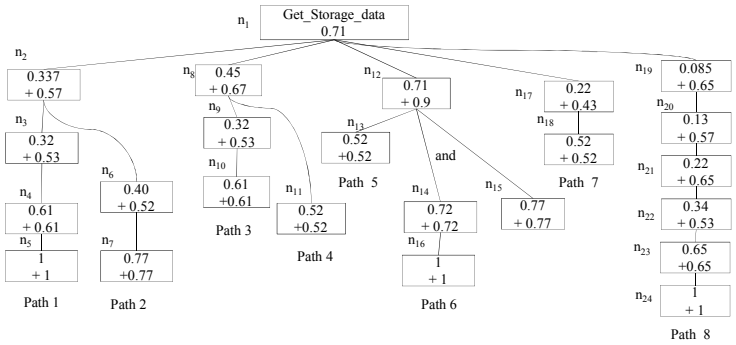

Fig. 7: Attack Tree Metric

5.2 Bayesian Network Metric

In this section, the BN-based security metric [24, 9] will be applied to the attack graph shown in Figure 5 to measure the threat and also the effect of certain changes made to the infrastructure. In particular, we show how the level of redundancy and diversity may affect the security of the cloud infrastructure. For redundancy, the ssh service running on some of the servers will be disabled to see the effect on security. As to diversity, we assume the ssh service may be diversified with other software, e.g., OpenSSH version 4.3, denoted as ssh_2, with a vulnerability CVE-2009-290 and a CVSS score of 6.9 [1].

Table 2 shows how security is affected by reducing redundancy and increasing diversity through disabling or diversifying some of the ssh instances in the infrastructure. In the left-hand side table, the first row shows that the probability for an attacker to reach the goal is 0.174 in the original configuration, and the remaining rows show the same probability after disabling one or more ssh instances on the three servers, e.g., the probability after disabling ssh on the $http$ server is reduced to 0.121, which corresponds to the most secure option by disabling one ssh instance, and the lowest probability after disabling two and three ssh instances is 0.094 and 0.074, respectively.

The middle and right-hand side of Table 2 show the effect of diversifying the ssh instances. In the middle figure, we can observe that, after we replace the ssh service on app and DB servers with ssh_2, the probability for reaching the goal decreases from 0.174 to 0.171, which indicates a slight improvement in security. The next three rows of

the table show that the same effect remains when one of the ssh instances is disabled. The last three rows show the simple fact that, when there is only one ssh instance left, the diversification effort has not effect.

In the right-hand side of Table 2, we change the ssh instance on the $http$ server instead of the app server, as in the above case, in order to see whether different diversification options make any difference to security. We can see the probability decreases in most cases (except the fourth row), which indicates a slightly more effective option than the previous one. Overall, the best option in terms of diversification without disabling any service instance is given in the first row in the right table, with a probability 0.17, and the best option for disabling one service instance is given in the fourth row of the middle table with a probability 0.119 (disabling two instances always yields 0.094). Obviously, more options may be evaluated similarly using the BN-based metric in order to find the best option for making the cloud data center infrastructure more secure.

$\langle user, Xen \rangle$			
$http$	app	DB	T
ssh			T
T	T	T	0.174
T	F	T	0.136
T	T	F	0.136
F	T	T	0.121
T	F	F	0.106
F	F	T	0.094
F	T	F	0.094
F	F	F	0.074

$\langle user, Xen \rangle$			
$http$	app	DB	T
ssh_1	ssh_2	ssh_2	T
T	T	T	0.171
T	F	T	0.135
T	T	F	0.135
F	T	T	0.119
T	F	F	0.106
F	F	T	0.094
F	T	F	0.094
F	F	F	0.074

$\langle user, Xen \rangle$			
$http$	app	DB	T
ssh_2	ssh_1	ssh_2	T
T	T	T	0.17
T	F	T	0.133
T	T	F	0.134
F	T	T	0.12
T	F	F	0.105
F	F	T	0.094
F	T	F	0.094
F	F	F	0.074

Table 2: The BN-Based Metric Results for the Attack Graph Shown in Figure 5

6 Related Work

Cloud environments are usually subject to many security threats some of which exploit existing vulnerabilities related to the cloud [10]. There only exist limited efforts on threat modeling for cloud infrastructures. Ingalsbe et al. present a threat model that cloud tenants can use to evaluate the system [13]. The authors adopt an Enterprise Threat Modeling methodology, which classify all components related to the cloud tenant under three categories (Actor, End Points, and Infrastructure). However, the authors do not provide concrete case studies detailing how such a threat model might be used. Gruschka & Jensen apply the attack surface concept to provide classifications for attacks in a cloud [11]. The authors identify three main entities (User, Cloud provider, and Service) and the attack surfaces between those entities. The authors provide high level examples of attacks but do not mention specific services or vulnerabilities underlying each attack surface. We borrow this classification in devising our threat models. The original attack surface concept [15] is intended to measure the security of a software system focusing on identifying entry/exit points, communication channels, and untrusted data items from the source code. Like most existing work, our work applies those concepts but at a higher abstraction level.

Attack tree is a well known threat model which can be used for many useful analyses, such as analyzing the relative cost of attacks and the impact of one or more attack vectors [20]. Attack trees can also be used in security hardening to determine the best

options to increase security within a budget [7]. Using attack trees can help to understand what kind of attackers may follow an attack tree path [20, 18]. Attack graphs can be automatically generated by modeling the network and vulnerabilities, and many useful analyses may be performed using attack graphs [22]. We borrow the concepts of attack trees and attack graphs but apply them to cloud data center infrastructures that we have devised. There exist many research work on extending attack trees and attack graphs to security metrics. A probabilistic metric is applied to attack graphs to obtain an overall attack likelihood for the network [24]. Edge et al. presented protection trees [8] which are similar to attack trees but contain information on how the system can be secured, and our work borrows part of this work to apply the attack tree-based metric. A BN-based security metric applies attack graphs to measure the security level of a network [9]. The metric converts the CVSS scores of vulnerabilities into attack probabilities and then obtain the overall attack likelihood for reaching critical assets. We apply this metric to our cloud data center infrastructures in this paper. The National Institute of Standards and Technology (NIST) underline the importance of security measuring and metrics for cloud providers by providing high level definitions and requirements but no concrete methodologies [2]. Luna et al. propose a framework with basic building blocks for cloud security metrics [14]. We loosely follow the framework in this paper.

7 Conclusion

In this paper, we have conducted threat modeling and measuring for cloud data center infrastructures. First, we have shown two cloud data center infrastructures which are fictitious but represent many existing technologies adopted at real cloud data centers by major cloud providers. Three threat models were then applied to those infrastructures, namely, the attack surface, attack trees, and attack graphs, which model potential threats from different viewpoints and at different abstraction levels. We have also applied security metrics based on attack trees and attack graphs, respectively, to quantify the threats. This work will benefit cloud providers in demonstrating how threat models and metrics may assist them in evaluating and improving the security of their clouds. Future work will focus on extending the scale and scope of our existing efforts and developing automated hardening algorithms for cloud data centers to generate actionable knowledge from the threat modeling and measuring results.

Disclaimer This paper is not subject to copyright in the United States. Commercial products are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the identified products are necessarily the best available for the purpose.

References

1. National vulnerability database. `http://www.nvd.org`. [Online; accessed 20/02/2015].
2. National Institute of Standards and Technology: Cloud Computing Service Metrics Description. `http://www.nist.gov/itl/cloud/upload/RATAX-CloudServiceMetricsDescription-DRAFT-20141111.pdf`, 2015. [Online; accessed 17/06/2015].

3. B. Adler. Google Compute Engine Performance Test with RightScale and Apica. http://www.rightscale.com/blog/cloud-industry-insights/google-compute-engine-performance-test-rightscale-and-apica, 2013. [Online; accessed 26/03/2016].
4. K. Bakshi. Cisco cloud computing-data center strategy, architecture, and solutions. *DOI= http://www. cisco. com/web/strategy/docs/gov/CiscoCloudComputing_WP. pdf*, 2009.
5. J. Barr. Building three-tier architectures with security groups. https://aws.amazon.com/blogs/aws/building-three-tier-architectures-with-security-groups/, 2010. [Online; accessed 28/03/2016].
6. K. Dahbur, B. Mohammad, and A. B. Tarakji. A survey of risks, threats and vulnerabilities in cloud computing. In *Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications*, ISWSA '11, pages 12:1–12:6, New York, NY, USA, 2011. ACM.
7. R. Dewri, I. Ray, N. Poolsappasit, and D. Whitley. Optimal security hardening on attack tree models of networks: a cost-benefit analysis. *International Journal of Information Security*, 11(3):167–188, 2012.
8. K. S. Edge, G. C. Dalton, R. A. Raines, and R. F. Mills. Using attack and protection trees to analyze threats and defenses to homeland security. In *MILCOM 2006 - 2006 IEEE Military Communications conference*, pages 1–7, Oct 2006.
9. M. Frigault and L. Wang. Measuring network security using bayesian network-based attack graphs. In *Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International*, pages 698–703, July 2008.
10. B. Grobauer, T. Walloschek, and E. Stöcker. Understanding cloud computing vulnerabilities. *Security & privacy, IEEE*, 9(2):50–57, 2011.
11. N. Gruschka and M. Jensen. Attack surfaces: A taxonomy for attacks on cloud services. In *2010 IEEE 3rd International Conference on Cloud Computing*, pages 276–279, July 2010.
12. M. Hany. VMware VSphere In The Enterprise. http://www.hypervizor.com/diags/HyperViZor-Diags-VMW-vS4-Enterprise-v1-0.pdf. [Online; accessed 05/02/2015].
13. J. A. Ingalsbe, D. Shoemaker, and N. R. Mead. Threat modeling the cloud computing, mobile device toting, consumerized enterprise-an overview of considerations. In *AMCIS*, 2011.
14. J. Luna, H. Ghani, D. Germanus, and N. Suri. A security metrics framework for the cloud. In *Security and Cryptography (SECRYPT), 2011 Proceedings of the International Conference on*, pages 245–250, July 2011.
15. P. Manadhata and J. Wing. An attack surface metric. *Software Engineering, IEEE Transactions on*, 37(3):371–386, May 2011.
16. P. Mell, K. Scarfone, and S. Romanosky. Common vulnerability scoring system. *IEEE Security & Privacy*, 4(6):85–89, 2006.
17. Openstack. Openstack Operations Guide. http://docs.openstack.org/openstack-ops/content/openstack-ops_preface.html. [Online; accessed 27/08/2015].
18. I. Ray and N. Poolsapassit. *Computer Security – ESORICS 2005: 10th European Symposium on Research in Computer Security, Milan, Italy, September 12-14, 2005. Proceedings*, chapter Using Attack Trees to Identify Malicious Attacks from Authorized Insiders, pages 231–246. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
19. P. Saripalli and B. Walters. Quirc: A quantitative impact and risk assessment framework for cloud security. In *2010 IEEE 3rd International Conference on Cloud Computing*, pages 280–288, July 2010.
20. B. Schneier. Attack trees. *Dr. Dobb's journal*, 24(12):21–29, 1999.

21. F. B. Shaikh and S. Haider. Security threats in cloud computing. In *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*, pages 214–219, Dec 2011.
22. O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing. Automated generation and analysis of attack graphs. In *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*, pages 273–284, 2002.
23. R. Squillace. Azure infrastructure services implementation guidelines. `https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-linux-infrastructure-service-guidelines/`, 2015. [Online; accessed 28/03/2016].
24. L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia. An attack graph-based probabilistic security metric. In V. Atluri, editor, *Data and Applications Security XXII*, volume 5094 of *Lecture Notes in Computer Science*, pages 283–296. Springer Berlin Heidelberg, 2008.