QuantiC: Distance Metrics for Evaluating Multi-Tenancy Threats in Public Cloud

Taous Madi
CIISE
Concordia University
Montreal, QC, Canada
t_madi@encs.concordia.ca

Mengyuan Zhang
Ericsson Security Research
Ericsson Canada
Montreal, QC, Canada
mengyuan.zhang@ericsson.com

Yosr Jarraya
Ericsson Security Research
Ericsson Canada
Montreal, QC, Canada
yosr.jarraya@ericsson.com

Amir Alimohammadifar
CIISE
Concordia University
Montreal, QC, Canada
ami_alim@encs.concordia.ca

Makan Pourzandi
Ericsson Security Research
Ericsson Canada
Montreal, QC, Canada
makan.pourzandi@ericsson.com

Lingyu Wang
CIISE
Concordia University
Montreal, QC, Canada
wang@encs.concordia.ca

Mourad Debbabi
CIISE
Concordia University
Montreal, QC, Canada
debbabi@encs.concordia.ca

***Abstract*—As a cornerstone of cloud computing, multi-tenancy brings not only the benefit of resource sharing but also additional security implications. To achieve an optimal trade-off between security and resource sharing, cloud providers are obliged to evaluate the potential threats related to multi-tenancy. However, quantitative approaches for evaluating those threats are largely missing in existing works. In this paper, we propose a set of multi-level distance metrics that quantify the proximity of tenants' virtual resources inside a cloud. Those metrics are defined based on the configuration and deployment in a cloud, such that a cloud provider may apply them to evaluate the risk related to potential multi-tenancy attacks. We conduct case studies and experiments on both real and fictitious clouds. The obtained results show the effectiveness and applicability of our metrics. We further implement our metrics in OpenStack and show how they can be applied for distance auditing.**

***Index Terms*—Multi-Tenancy, Security Metrics, SDN-Based Cloud, OpenFlow**

I. Introduction

Multi-tenancy of the cloud is a double edged sword. On one side, the economic gain fulfilled through resource sharing constitutes one of the most appealing cloud advantages that attract prospective customers. On the other side, the security challenges driven by multi-tenancy and the associated risks [1] constitute some of the main concerns that are holding back the migration of critical applications to cloud.

In fact, the proximity with the victim can be exploited by malicious cloud users to mount several attacks. In Table I, we roughly classify those attacks into two categories according to the required proximity (the list of attacks is not meant to be exhaustive; other, including future or unknown, attacks may also fit into those categories). When an attacker shares the same host with the targeted victim, (s)he can launch type I attacks (e.g., side channel attacks [2]), whereas type II attacks (e.g., power attack [3]) can be mounted when resources are shared with the victim at higher levels of the cloud infrastructure, (e.g., rack-level). Successful attacks may affect security properties of both victim's virtual machines (VMs) and their generated network flows at various levels of the hierarchy. As an example, recent works have demonstrated the feasibility of real-life attacks conducted in commercial clouds including Amazon EC2, aiming at forcing malicious VMs to be placed within a specific zone, which could be a host, a rack or a larger scale area inside the cloud data center [4], [5].

Today's cloud service providers (CSPs) are well aware of such multi-tenancy-related threats, and they are often under obligation to protect their tenants against such threats, either as part of the service level agreements or to demonstrate compliance with security standards (e.g., CCM 3.0.1 [10]). Nonetheless, addressing multi-tenancy threats remains a challenging issue. First of all, completely avoiding multi-tenancy is certainly impractical since it reduces the financial benefit, which is an important factor to cloud adoption. Alternatively, enabling resource sharing naturally implies a degree of exposure to multi-tenancy threats. A mid-way solution for the CSP would be to balance between the security implications and the economic benefits of resource sharing. In this respect, evaluating multi-tenancy threats based on the proximity between tenants sharing the same cloud constitutes a valuable means towards reaching an optimum trade-off between tolerated risks and costs according to negotiated contracts.

Particularly, existing approaches (e.g., [11], [12]) propose metrics to evaluate the overall cloud security risk based on vulnerabilities in cloud deployments (a detailed review of the related work will be given in Section V). Nonetheless, none of them provides the potential impact at tenant-level according to the degree of resource sharing. Furthermore, those works focus only on the multi-tenancy threat related to type I attacks, while evaluating the threat of type II multi-tenancy attacks has not been tackled yet.

To the best of our knowledge, this is the first work that

Multi-Tenancy Attacks		Cloud Inf. Levels		Targeted Resources		Targeted Sec. Prop.		
		Host only	Different Levels	Compute	Network	C	I	A
Type I	Side channel attacks [2]	•		•		•		
	Host-based DoS attack [6]	•		•	•			•
	SDN-based freeloading attack [7]	•			•	•	•	
Type II	Power attacks [3]		•	•	•			•
	Bandwidth attack [8]		•		•			•
	Resource abuse [9]		•		•			•

TABLE I: Multi-tenancy attacks, their scopes, targeted resources and the affected security properties, namely, confidentiality (C), integrity (I) and availability (A)

proposes multi-level metrics to quantify the distance between tenants' resources in an SDN-based cloud, as a means to evaluate the multi-tenancy threats related to both type I and type II attacks and assess the corresponding risk per tenant. Specifically, the main contributions of this work are as follows.

- We devise a multi-level model capturing tenants' virtual infrastructures deployment inside SDN-based cloud.
- We propose novel metrics, namely, physical, compute and network distances, to quantify the multi-tenancy threat in an SDN-based cloud.
- We present three case studies based on both a real cloud and fictitious clouds. The first and second case studies show how our metrics correlate with the two types of multi-tenancy attacks. In the third case study, we implement our metrics in OpenStack and show how they can be used to define the CSP's compliance with tenants' distance requirements.

The remainder of this paper is organized as follows. Section II discusses the threat model, provides a running example and presents our multi-level cloud infrastructure model. Section III provides the formal definitions of our distance metrics. Section IV presents the case studies and discusses how our metrics can be used for per tenant risk assessment. Section V summarizes the related work. Section VI concludes the paper.

II. Models

In the following, we discuss our threat model, and present the running example and the cloud infrastructure model.

A. Threat Model

In this study, we assume that tenants do not have any prior knowledge on the identities of other tenants hosted inside the same cloud. Our in-scope attacks include any multi-tenancy attacks that require an adversary to share resources with the victim tenant at multiple levels of the cloud data center. Any attacks involving administrator privileges are out of scope. Consequently, we assume the information collected from the cloud infrastructure management system to calculate our metrics are trusted.

Our metrics are meant for evaluating the multi-tenancy threats against the in-scope attacks, and they are not designed to detect such attacks, identify the malicious tenant, or pinpoint the vulnerabilities. In fact, our metrics are to be applied before the attacks actually happen (unlike [13]), and without any prior knowledge of the attacker's identity (unlike [14]). Thus, our metrics are complementary to other attack-specific security solutions, e.g., attack detection and vulnerability analysis.

B. Running Example

In Figure 1, tenant t_A shares the same data center with many other tenants (to better illustrate the case, we consider an exemplary tenant t_B). Assume the CSP wants to evaluate the impact of potential type I and type II multi-tenancy attacks depicted in Table I against t_A. Based on the deployment in Figure 1, the CSP can make the following observations:

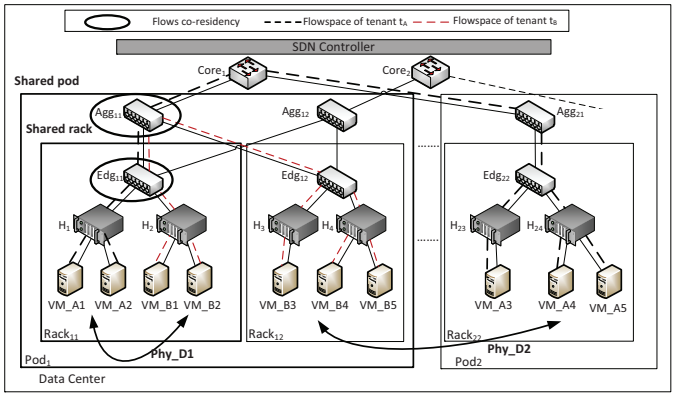


Fig. 1: An example demonstrating the physical distance between tenants' virtual infrastructures, where *VM_A1*,.., *VM_A5* belong to tenant t_A and *VM_B1*,.., *VM_B5* belong to tenant t_B

- None of t_A's VMs are co-located with t_B at the host-level, therefore, it is unlikely for t_B to perform type I attacks (e.g., side channel attacks [2]) against t_A's VMs, or to abuse their network flows (e.g., freeloading attack [7]).
- Although launching type I attacks is out of t_B's reach, a closer look reveals that t_A is still under the risk of type II attacks that take advantage of the shared infrastructure at higher levels without requiring host-level co-residency. For example, t_B can perform power attack [3] at $Rack_{11}$ using *VM_B1* and *VM_B2* to disturb services running at *VM_A1* and *VM_A2* located at the same rack. This attack also disturbs the communication of *VM_A1* and *VM_A2* with *VM_A3*, *VM_A4* and *VM_A5* located at $Rack_{22}$.
- Furthermore, *VM_A3*, *VM_A4* and *VM_A5*, that are located in a different rack and pod than t_B, are less exposed to type II attacks since their physical distance with respect to t_B is larger than the physical distance of *VM_A1* and *VM_A2* with respect to the same tenant ($Phy\_D2 > Phy\_D1$).

The above observations intuitively show the correlation between measuring distances between tenants' virtual infrastructures and evaluating the degree of exposure to multi-tenancy threats at different levels of the shared cloud infrastructure.

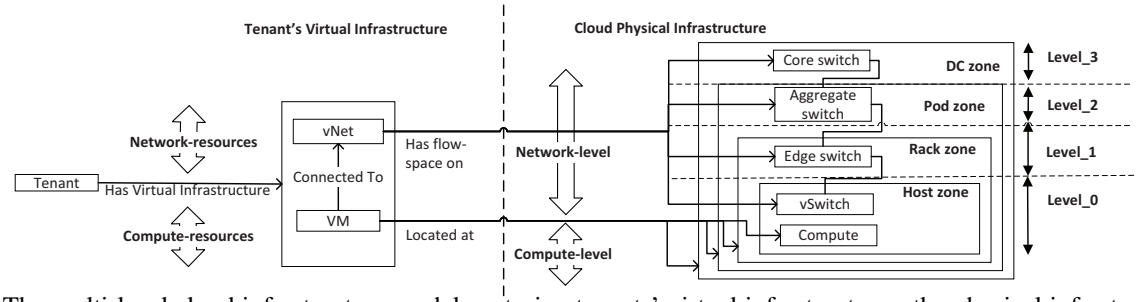


Fig. 2: The multi-level cloud infrastructure model capturing tenants' virtual infrastructures, the physical infrastructure and their mapping. Note that the presented three-tiered network hierarchy is shared by most cloud data center topologies [15]

C. Multi-Level Cloud Infrastructure Model

To measure the distance between tenants, we derive an entity-relationship model that captures tenants' virtual infrastructure elements, the cloud infrastructure elements and their relationships. Figure 2 illustrates such a model. The cloud physical infrastructure includes servers and switches that are hierarchically structured in different management zones shown as aggregated nodes (e.g., several hosts can be aggregated into a rack zone). A *Tenant*'s virtual infrastructure consists of a set of *VMs* and their connecting virtual networks (*vNet*). Tenants' VMs are located at *compute* services running inside hosts. VMs are connected to vNets that are typically implemented using *flowspaces* constituted of a set of OpenFlow rules [16] segregated with flow tags[1]. These rules are configured in some physical and virtual switches in different levels of the hierarchy to enable the communication between VMs. We use FS_{vNet} to denote the cloud-wide flowspace of *vNet*, FS^{i}_{vNet} to denote the flowspace of *vNet* at *Level_i*, and $FS^{sw_{ij}}_{vNet}$ to denote a flowspace in a given switch sw_{ij} at *Level_i*.

On the right side of Figure 2, we define four physical *levels* (*Level_*0 to *Level_*3) where tenants' virtual infrastructures (depicted on the left side of Figure 2) might be located. As detailed later in Section III, we use those levels to define our distance metrics. In the following, we provide the formal definition for the multi-level cloud infrastructure model.

Definition 1 (Multi-Level Cloud Infrastructure Model): We define the cloud infrastructure model as an array $\overrightarrow{CInf}$ of dimension four, where $CInf[i]$.zone and $CInf[i]$.switch are respectively the sets of zones and switches at Level_i ($0 \leq i \leq 3$).

Example 1: Figure 3 illustrates an instance of the aforementioned multi-level cloud infrastructure model (Figure 2) capturing the example of Figure 1. In Figure 3, an excerpt of the OpenFlow table in Edg_{11} shows the co-residency of the flow rules belonging to *vNet_A* (i.e., *r1* and *r2*) and *vNet_B* (i.e., *r3*). Specifically, *VM_A1* and *VM_A2* of t_A located at $Rack_{11}$ communicate with *VM_A3*, *VM_A4* and *VM_A5* (not shown for space limitation) located at $Rack_{22}$ through *vNet_A*. Similarly, *VM_B1* located at $Rack_{11}$ communicates with *VM_B5* located at $Rack_{12}$ through *vNet_B*. Those communications are made possible through flowspaces installed inside Edg_{11}, Agg_{11} and other switches in the topology depending on the location of the communicating VMs. Since *VM_A1* and *VM_A2* of t_A co-reside with *VM_B1* at $Rack_{11}$, the flowspaces governing their flows will inevitably share Edg_{11} at the rack-level and possibly Agg_{11} at the pod-level. □

[1]A flow tag is a special match field in OpenFlow rules that enables to segregate flow rules belonging to different virtual networks

III. Multi-Tenancy Distance Metrics

We first define the multi-level physical distance between a pair of tenants to capture their symmetric distance based on the level of physical resource sharing, then we refine this distance along the compute and network dimensions to quantify their asymmetric distances based on their resources' deployment.

A. Physical Distance

The physical distance captures the symmetric relationship between a pair of tenants in terms of the levels of shared resources. We define this distance between two tenants' virtual infrastructures (VMs and their flowspaces) as a four-dimensional vector D_ϕ, where $D^i_\phi = 0$ (resp. $D^i_\phi = 1$) means *Level_i* is (not) shared. We provide an illustrative example followed by the formal definition.

Example 2: In Figure 3, VMs of tenant t_A do not co-locate in the same hosts at *Level_0* with the VMs of tenant t_B, their physical distance at *Level_0* is therefore $D^0_\phi = 1$. However, *VM_A1* and *VM_A2* share $Rack_{11}$ at *Level_1* with *VM_B1* and *VM_B2*, and since management zones are nested, it follows that all the upper levels of the cloud infrastructure are also shared. Additionally, the flowspaces associated with *vNet_A* and *vNet_B* share Edg_{11} at *Level_1* and Agg_{11} at *Level_2*. Thus, the physical distance between the two tenants can be quantified using the vector $(1,0,0,0)$. □

Let t and t' be two tenants hosted at the cloud data center. The virtual infrastructure belonging to tenant t (resp. tenant t') is composed of a set of VMs, VM_s (resp. VM'_s) connected to *vNet* (resp. *vNet'*), where FS^i_{vNet} (resp. $FS^i_{vNet'}$) is the associated flowspace at a given *Level_i* ($0 \leq i \leq 3$). We define the set of shared zones between t and t' at *Level_i* to be the set of zones that are simultaneously accommodating at least one VM belonging to tenant t and one VM belonging to tenant t'. We denote it $sz_i\{VM_s, VM'_s\}$. We similarly define the set of shared switches between t and t' at $Level_i$ to be the set of switches on which is installed at least one OpenFlow rule r from each of t and t' flowspsaces. We denote it $ss_i\left\{FS^i_{vNet}, FS^i_{vNet'}\right\}$. We define the symmetric physical distance between the pair of tenants $\{t,t'\}$ as follows:

Definition 2 (Physical Distance):

Let $sz_i\{VM_s, VM'_s\}$ and $ss_i\left\{FS^i_{vNet}, FS^i_{vNet'}\right\}$ respectively the sets of shared zones and switches between t and t' at

A subset of OpenFlow Rules in the edge switch Edg11

Rule	Flow_tag	Src_MAC	Dst_MAC	Src_IP	Dst_IP	Src_Port	Dst_Port	Action
r1	**A**	**MAC(VM_A1)**	**MAC(VM_A3)**	**IP(VM_A1)**	**IP(VM_A3)**	**20**	**20**	**OutPort:10**
r2	**A**	**MAC(VM_A1)**	**MAC(VM_A2)**	**IP(VM_A1)**	**IP(VM_A2)**	**22**	**22**	**OutPort:8**
r3	B	MAC(VM_B5)	MAC(VM_B1)	IP(VM_B5)	IP(VM_B1)	80	80	OutPort:6

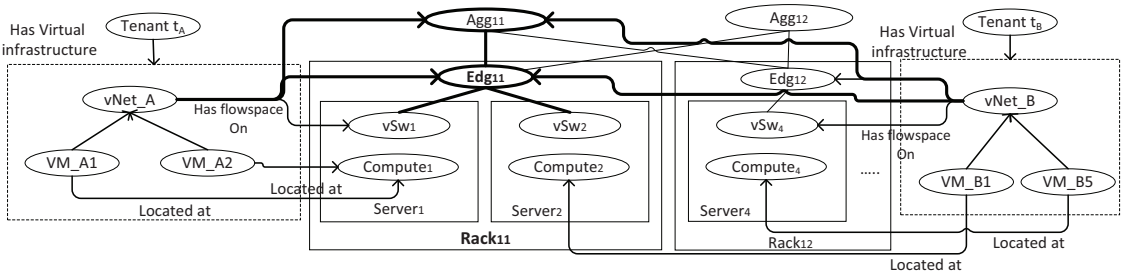


Fig. 3: An instance of the multi-level cloud infrastructure model capturing a subset of the deployment of Figure 1

Level_i. Then, their physical distance is given by the four dimensional vector $D_\phi\{t,t'\}$, where the values of its elements D^i_ϕ are computed as follows:

$$D^i_\phi\{t,t'\} = \begin{cases} 1 & \text{if } sz_i\{VM_s, VM'_s\} = \emptyset \text{ and } ss_i\{FS^i_{vNet}, FS^i_{vNet'}\} = \emptyset \\ 0 & \textit{Otherwise} \end{cases}$$

B. Compute Distance

The compute distance is an asymmetric distance that captures the degree of exposure of a tenant t's VMs to another tenant t'.

Example 3: From Example 2 we have $D_\phi\{t_A,t_B\} = (1,0,0,0)$. *VM_A1* and *VM_A2* of t_A share $Rack_{11}$ with t_B's VMs, while *VM_A3*, *VM_A4* and *VM_A5* share the cloud infrastructure with t_B at *Level_3* only, which corresponds to the data center. Consequently, the compute distance for t_A with respect to t_B at *Level_1* and *Level_2* is the fraction of VMs that *do not* share the same racks and pods, which is $3/5$. Hence, the multi-level compute distance for tenant t_A with respect to t_B is $(1,3/5,3/5,0)$. □

More formally, we define the average compute distance of tenant t with respect to tenant t' according to the number of shared zones as follows (VM^z_s is the set of VMs located at zone z):

$$D^i_\varsigma(VM_s, VM'_s) = \begin{cases} D^i_\phi\{t,t'\} & \text{if } sz\{VM_s, VM'_s\} = \emptyset \\ \frac{\sum_{z \in CInf[i].zone \setminus sz\{VM_s, VM'_s\}} |VM^z_s \sqcap VM_s|}{|VM_s| \times |sz\{VM_s, VM'_s\}|} & \textit{Otherwise} \end{cases}$$

We consider the average distance because the more resources share the same zone the higher the risk related to multi-tenancy attacks would be, as will be discussed in Section IV-A. Note that D^3_ς is different than zero when tenants' VMs are deployed over multiple data centers.

C. Network Distance

By analogy to the compute distance, the network distance is also an asymmetric distance that captures the degree of exposure of a specific tenant's network resources with respect to another tenant.

Example 4: The OpenFlow rules depicted in Figure 3 have six match fields, source/destination MAC, source/destination IP and source/destination port, in addition to the flow-tag. The bit sequence composing those match fields can be either a wildcard or an exact-match, i.e., fixed to zero or to one, where rules with more wildcarded bits define larger flows. Since sharing more flows with other tenants increases the risk of network isolation breaches (e.g., freeloading attacks [7]) and unavailability (e.g., bandwidth attack [8]), we quantify the network distance of *vNet_A* with respect to *vNet_B* based on the size of flowspaces that are not sharing the same switches. As illustrated in Figure 3, a case of co-residency for the flowspaces of *vNet_A* and *vNet_B* is reported at *Level_1* in Edg_{11}. In the latter switch, both flow rules *r1* and *r2* have all the match fields as exact match meaning that each rule handles a flow composed of *one packet* only. Since not all flowspaces can be shown for space limitation, we assume that the flow size of *vNet_A* at Edg_{11} is equal to 10, and that its total flow size at *Level_1* is 16. Then, the network distance at this level is $D^1_\eta = (16-10)/16$. Additionally, if we assume that all *vNet_A* flowspaces are shared with *vNet_B* at both *Level_2* and *Level_3*, then the network distance vector for *vNet_A* with respect to *vNet_B* would be equal to $(1,6/16,0,0)$. □

Let ω be the length in terms of bits of an OpenFlow rule match sequence. Similarly to [17], we abstract away from the meaning associated with each OpenFlow rule's header match field, and consider a match sequence to be a sequence of bits defined over $\{0,1,*\}^\omega$, where * is the wildcard symbol. Let ψ be the number of exact match bits of an OpenFlow rule r, where $\psi \leq \omega$, and let *sizeof(_)* be a function that measures the flow size of the OpenFlow rules. The flow size of r is equal to $sizeof(r) = 2^{\omega-\psi}$. Particularly, the flow size defined by a rule where all bits in the match sequence are exact match, is equal to $sizeof(r) = 2^0 = 1$ (as $\psi = \omega$). The size of all flowspaces for a given virtual network at a specific level can be computed by aggregating the size of all OpenFlow rules associated with it (for simplicity, we assume that OpenFlow rules do not overlap). This is given by $size(FS^i_{vNet}) = \sum_{r \in FS^i_{vNet}} sizeof(r)$. We define the average network distance between the flowspaces of *vNet* and *vNet'* at a given *Level_i* as:

$$D^i_\eta(FS^i_{vNet}, FS^i_{vNet'}) = \begin{cases} 1 & \text{if } ss_i\{FS^i_{vNet}, FS^i_{vNet'}\} = \emptyset \\ \frac{size(\bigcup_{s \in CInf[i].switch \setminus ss_i\{FS^i_{vNet}, FS^i_{vNet'}\}} FS^s_{vNet})}{size(FS^i_{vNet}(t)) \times |ss_i\{FS^i_{vNet}, FS^i_{vNet'}\}|} & \textit{otherwise} \end{cases}$$

IV. Case Studies

In this section, we illustrate through case studies the applicability of our distances with both fictitious and real clouds. We also present a quantitative auditing approach.

A. Case Study 1 (Correlation with Multi-Tenancy Attacks)

We consider the fictitious cloud data center illustrated in Figure 4, which is constituted of four pods, eight racks (two racks per pod) and 96 physical servers (12 servers per rack). This data center is shared by several tenants. For illustrative purposes, we consider four tenants, namely, t_A, t_B, t_C and t_D.

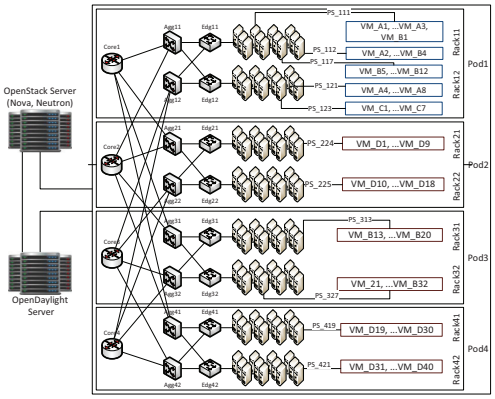


Fig. 4: An illustrative case study of a cloud data center topology. Physical servers are named *PS_xyz*, where x is the index of the pod, y is the index of the rack, and z is the index of the physical server

In the following, we show how our physical distance correlates with the two types of multi-tenancy attacks (see Table I). The rows of matrix $D_\phi(t_A)$ report the physical distance of t_A with respect to tenants, t_B (first row), t_C (second row) and t_D (third row) based on the deployment of Figure 4, where each column represents a physical level of the cloud infrastructure.

$$D_\phi(t_A) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

The following shows how larger physical distances reduce the multi-tenancy threats. Assume t_B, t_C and t_D are malicious and want to take advantage of the multi-tenancy situation to launch type I or type II attacks (see Table I) against t_A. Based on Table I, we can discuss the required distance and potential impact for each category of attacks as follows.

- Type I attacks require co-residency with the targeted victim at the *same host* (e.g., side channel attacks [2]). As $D^0_\phi\{t_A, t_B\} = 0$, the only potential risk of this type of attacks is limited to tenant t_B.
- Type II attacks do not necessarily require co-residency at the host-level to succeed. However, the following reasoning shows that the larger the physical distance is, the less the risk related to those attacks would be. We consider power attack [3] as an example and similar reasoning can be applied to other type II attacks (e.g., bandwidth attacks [8]).

The power attack exploits the power over-subscription vulnerability, which consists of overloading a power supply with more workloads than it supports with the assumption that workloads will never reach their peak simultaneously. If the attacker succeeds to place many VMs inside a zone (server, rack or a larger zone) alimented with the same power facility, then he can generate simultaneous power spikes, which would lead to power outage when the power consumption exceeds the power capacity for that specific zone. However, the larger the zone attacker is targeting, the more controlled VMs need to be deployed to increase the power consumption, since smaller zones converge faster to their peak power[1]. Based on that and considering $D_\phi(t_A)$, we can infer the following:

- If t_B or t_C launch their attack against $Rack_{11}$, this would be enough for them to cause damage to all the resources of t_A (VMs and their flows) that are located at this rack zone, since both tenants share the same rack as the victim.
- However, it is more difficult for t_D to affect t_A resources since this would require him to launch this attack at the data center scale (as no racks or pods are shared), which would require much more effort than for t_B or t_C.

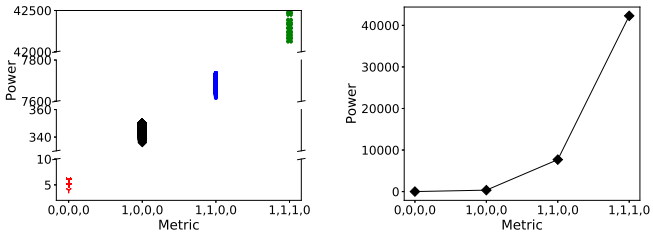
To show the correlation between the physical distance and the effort required to launch power attack, we simulated the cloud architecture described in [19], with a number of tenants' workloads following an exponential distribution [9]. Power is defined per units, where each unit power supports one VM. We assume each host has the capability to accommodate eight VMs, and the power consumption at higher levels is obtained by summing up the power consumption of aggregated lower levels. Figure 5 reports the effort required by an attacker at each level of the cloud infrastructure in terms of the number of deployed VMs and their consumed power.

We observe that launching power attack at *Level_0* requires the lowest effort, while launching the attack at the data center scale requires consuming four orders of magnitude more energy by deploying more VMs. From this analysis, we can conclude that larger physical distances reduce the multi-tenancy risk for power attack. In the next case study, we show with real cloud data, the need for refined distance metrics to capture the impact of potential multi-tenancy attacks.

B. Case Study 2 (Real Cloud Data Center)

This case study is based on a real community cloud hosted at a major telecommunication company. We collect data from part of this cloud composed of 22 hosts organized into two racks as depicted in Figure 6. We perform our study on a dataset composed of 372 VMs belonging to 37 tenants. The focus of this case study is to show the complex co-residency relationships between tenants in real world cloud, and therefore, the need for metrics to measure distances between tenants' resources. For illustration, we randomly choose

[1]It has been reported in [18] that racks reach 96% of their peak power, while pods and data centers do not exceed respectively 86% and 72% of their peak power



(a) (b)

Fig. 5: (a) Attacker's requirements, and (b) average attacker's requirement in terms of power consumption to disrupt services of a victim at different levels of the cloud infrastructure. The X axis corresponds to possible physical distance metric values

three tenants, t_1, t_2 and t_3. Note that the dimension of our distances is equal to three for this hierarchy, since the latter is only composed of hosts, access and aggregate layers.

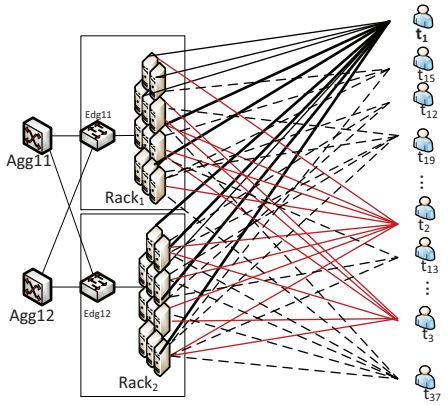


Fig. 6: Part of a real cloud data center topology constituted of 22 physical servers organized into two racks hosting 372 VMs belonging to 37 tenants

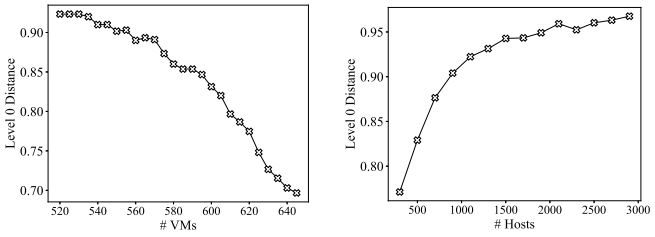
Table II reports the number of VMs of tenants t_1, t_2 and t_3 inside each physical host of the considered part of the cloud data center. One can notice that tenants' VMs are scattered over multiple physical nodes in both racks. Specifically, t_1 has VMs co-residing with both t_2 and t_3's VMs in many different locations. Consequently, the flowspace of t_1's virtual network co-resides with the flowspaces of t_2 and t_3 virtual networks at Edg_1, Edg_2 and Agg, in addition to the virtual switches running at the physical servers. Due to lack of space, we only discuss the compute distance. The matrix $D_\varsigma(t_1)$ reports the compute distance of t_1 with respect to t_2 (first row) and t_3 (second row) based on the deployment in Table II.

$$D_\varsigma(t_1) = \begin{pmatrix} 0.005 & 0.5 & 0 \\ 0.049 & 0.5 & 0 \end{pmatrix}$$

We can infer the following from the compute distances:

- Both t_2 and t_3 can perform type I attacks against t_1 since both are co-residing with the victim at some physical hosts (*Level_0*). However, t_1 has more VMs sharing the same hosts as t_2, and hence has smaller distance with respect to t_2 than t_3 ($0.005 < 0.049$). Therefore, the impact of t_2 attack on t_1 VMs will be higher than the impact of t_3 attack. Note that similar reasoning can be applied on the network distances.
- Both t_2 and t_3 can perform type II attacks either at the rack-level or at the pod-level as they have many VMs deployed over $Rack_1$ and $Rack_2$. Since the distance of t_1 with respect to t_2 is equal to his distance with respect to t_3 both at the rack-level ($D^1_\varsigma = 0.5$) and at the pod-level ($D^2_\varsigma = 0$), attacks from the two tenants will have similar impact on t_1.

We further evaluate through simulations how the compute distance changes while increasing the cloud data center's workload and size. As illustrated in Figure 7, our compute distance at *Level_0* captures the expected increase in the degree of resource sharing while increasing the total number of data center's VMs (see Figure 7(a)), and the decrease in resource sharing while increasing the data center's size (see Figure 7(b)), which shows the effectiveness of our metric.



(a) (b)

Fig. 7: The compute distance at *Level_0* (a) while increasing the number of data center's VMs, and (b) while increasing the number of data center's hosts

C. Case Study 3 (Quantitative Auditing)

In this case study, we show how our metrics can be used to quantitatively audit the compliance of deployed virtual infrastructures against tenants' requirements in terms of the distance. As a continuity of the case study in Section IV-A, we assume that tenant t_A's security team is aware of the multi-tenancy attacks and specifies accordingly a compute distance requirement for his own VMs against other tenants as $D\varsigma(t_A) = (1,1,0.5,0)$.

To evaluate the compliance deviation, the CSP first measures the distances for the current cloud deployment, then he checks the measured distances against the required one to evaluate the deviations. In the following, matrices $M_\varsigma(t_A)$ and $\triangle D_\varsigma(t_A)$ respectively report measured distances and deviations for t_A with respect to tenants t_B, t_C and t_D (represented respectively by the first, second and third row in matrices) based on the cloud configuration in Figure 4 and the required compute distance $D\varsigma(t_A)$. The obtained deviation matrix reports how much the current cloud implementation has deviated from the required specification, where higher values correspond to more deviations and consequently reduced distances.

$$M_\varsigma(t_A) = \begin{pmatrix} 0.625 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \triangle D_\varsigma(t_A) = \begin{pmatrix} 0.375 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

We integrated the described auditing approach into OpenStack [20], one of the most commonly used infrastructure management platforms. Algorithm 1 describes the compliance deviation evaluation procedures based on the required distances. First, the procedure

Racks	$Rack_1$											$Rack_2$										
Hosts	*S1*	*S2*	*S3*	*S4*	*S5*	*S6*	*S7*	*S8*	*S9*	*S10*	*S11*	*S12*	*S13*	*S14*	*S15*	*S16*	*S17*	*S18*	*S19*	*S20*	*S21*	*S22*
t_1	**0**	**1**	**0**	**0**	**0**	**4**	**6**	**4**	**4**	**8**	**9**	**4**	**10**	**4**	**6**	**16**	**1**	**4**	**8**	**7**	**2**	**0**
t_2	4	0	0	2	0	4	6	12	8	5	4	2	3	6	6	6	1	10	2	0	0	1
t_3	0	0	0	0	0	0	2	1	1	1	1	0	5	0	1	0	0	0	0	2	0	0

TABLE II: Number of VMs of tenants t_1, t_2 and t_3 insider each physical host in the considered part of the cloud data center

Per_Tenant_Implemented_Distance measures the implemented distances based on data collected mainly from Nova[1] database for the compute distances, and on the OpenDaylight[2] [21] database for the network distances. Then, the procedure *Per_Tenant_Deviation* evaluates the deviation with respect to different tenants accommodated by the same data center. Finally, a matrix is generated to report deviations at different cloud levels. Note that if tenant t_A has multiple outsourced virtual infrastructures, he can specify distance-based policies with multiple rules according to the sensitivity-level of different workloads.

Algorithm 1 Compliance Deviation Evaluation

```
procedure GLOBAL_DEVIATION( D(t))
    for each tenant t' belonging to the data center do
        M(t,t')=Per_Tenant_Implemented_Distance(t, t')
        △D(t,t')=Per_Tenant_Deviation(D(t), M(t,t'))
    Return(△D(t))
procedure CALCULATE_LOCAL_DEVIATION(D(t), M(t,t'))
    for i = 0 to 3 do
        △D[i] = 0
    if M[0] < D[0] then
        △D[0] = D[0] − M[0]
    Return (△D(t,t'))
```

To evaluate our quantified auditing approach, we simulate the K-ary tree data center topology [22] with 40 core switches, and deploy the virtual infrastructures of 20 tenants. We assign tenants' VMs to servers in a round robin fashion and build their connections in switches at different levels.

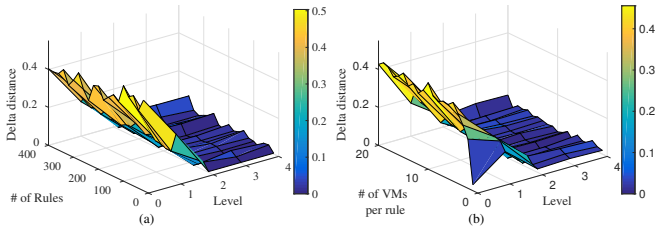


Fig. 8: Changes in the deviation vectors (a) while varying the number of rules, and (b) while varying the number of VMs per rule

In Figure 8(a), we fix the number of VMs per rule to 20 and vary the number of rules, whereas in Figure 8(b), we fix the number of rules to eight and vary the number of VMs per rule. In both figures, we can notice that the most significant deviations (delta distances) are recorded for *Level_0* (up to 0.45), which correspond to the host-level. This is due to the higher security threats related to host-level co-residency (type I attacks), leading t_A to set higher distances at *Level_0* compared to other levels. Therefore, deviations from those distance requirements drastically decimate the overall security with respect to the distance. As for *Level_1* and beyond, the deviation average does not exceed 0.1. This stems from the less significant security threats at higher levels leading t_A to relax the requested distances to reduce costs. Note that our approach is flexible to accommodate different tenants' security needs as they could specify their distances at deployment time.

[1]OpenStack Nova [20] is a project designed to provide on-demand access to compute resources

[2]OpenDayLight is an open source SDN controller

D. Discussions

Based on the presented case studies, we can conclude that the physical distance correlates with the degree of difficulty for multi-tenancy attacks, while the compute and network distances provide the potential impact of those attacks according to the degree of resource sharing at each level. Therefore, our distance metrics can be applied for evaluating the preliminary tenant pair-wise multi-tenancy risk incurred by a given cloud deployment. To this end, the CSP first defines a diagonal probability matrix P, where each element p_{ii} corresponds to the likelihood of different types of multi-tenancy attacks at *Level_i*. Those probabilities can be defined using existing approaches as presented in [11]. Then, the multi-tenancy risk for a given tenant t with respect to another tenant t' will be given by the weighted norm of tenant t's distance with respect to tenant t'. This can be expressed as $Risk(t,t') = ||D(t,t')||_P = \sqrt{D(t,t')^\top \times P \times D(t,t')}$. Since potential attackers' identity is not known a priory, the overall multi-tenancy risk for a tenant t can be defined as the average of tenant pair-wise risks given by $Risk(t) = \frac{\sum_{t' \in T \setminus \{t\}} Risk(t,t')}{|T|-1}$, where T is the set of all tenants.

Due to the dynamic nature of the cloud, calculated metric values can be quickly invalidated by various management operations such as VM migration events. By integrating our metrics into the cloud infrastructure management platform (e.g., OpenStack [20]), the CSP can monitor those operations and evaluate the metrics at runtime to continuously control the co-residency threats. Additionally, in this work, we assume that all VMs are equally sensitive, which might not be the case for some applications (e.g., three-tier applications). We leave the study of systematic approaches for runtime metrics evaluation and considering resources with different levels of sensitivity as part of future work.

V. Related Work

To the best of our knowledge, this is the first work proposing metrics for quantifying the distance between tenants' virtual infrastructures in cloud deployments.

Few works provide quantitative assessment frameworks to evaluate Security SLAs (SecLAs) [23], [24]. For instance, Luna et al. [23] developed a set of metrics to quantitatively compare, benchmark and evaluate the compliance of CSPs' reference SecSLAs. Authors in [24] propose a framework enabling cloud customers to choose the appropriate CSP according to their security requirements. While those approaches

provide valuable frameworks for prospective customers to choose the right CSP based on the advocated SecLAs, our approach provides CSPs with a tool to evaluate proximity between tenants' resources inside cloud deployments, which enables to evaluate the multi-tenancy risk.

In [11] and [12], authors propose metrics to evaluate the cloud-level risk from multiple perspectives (VMs, hosts and network connections). Those metrics enable to assess the cloud-level risks, while our metrics enable to evaluate the multi-tenancy rick at tenant-level, which makes them interesting for the auditing use case as discussed in SectionIV-C.

Most existing works on cloud auditing propose a binary audit answer [25]–[27]. Bleikertz et al. [25] propose a graph-based static information flow analysis for virtual infrastructures towards verifying information flow isolation. Majumdar et al. [26] propose a user-level multi-domain cloud auditing. Madi et al. [27] propose auditing cloud virtual infrastructures using a constraint satisfaction problem solver for checking security properties. While those approaches aim at detecting isolation breaches, they do not provide tenants with quantitative results reflecting the security-level of their resources.

In [28], authors propose a VM migration service that aims at limiting the information leakage due to side channel attacks by applying the moving target defense technique. Our metrics can be used to evaluate the effectiveness of this approach in reducing the multi-tenancy threats inside cloud deployments.

VI. Conclusion

In this paper, we proposed three metrics to quantify proximity between tenants inside cloud deployments. We showed through case studies the effectiveness and applicability of those metrics to evaluate multi-tenancy threats. We believe our metrics can be extended to evaluate other threats in cloud. Therefore, they should be considered as a first step toward a more general tool-set for threat evaluation in the cloud.

As future work, we intend to study multi-tenancy attacks taking advantage from shared storage and propose a storage distance accordingly. We also plan to propose cloud management strategies to enforce distances as a means to control the multi-tenancy risk.

Acknowledgment

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

References

[1] CSA, "The notorious nine cloud computing top threats in 2013," 2013. [Online]. Available: https://downloads.cloudsecurityalliance.org/initiatives/top_threats/The_Notorious_Nine_Cloud_Computing_Top_Threats_in_2013.pdf

[2] L. Fangfei, Y. Yuval, G. Qian, H. Gernot, and L. Ruby B., "Last-level cache side-channel attacks are practical," in *S&P'15*, May 2015, pp. 605–622.

[3] Z. Xu, H. Wang, Z. Xu, and X. Wang, "Power attack: An increasing threat to data centers," in *NDSS'14*, 2014.

[4] Z. Xu, H. Wang, and Z. Wu, "A measurement study on co-residence threat inside the cloud," in *USENIX Security 15*. Washington, D.C.: USENIX Association, 2015, pp. 929–944.

[5] V. Varadarajan, Y. Zhang, T. Ristenpart, and M. Swift, "A placement vulnerability study in multi-tenant public clouds," in *USENIX Security 15*. Washington, D.C.: USENIX Association, 2015, pp. 913–928.

[6] T. Zhang and R. B. Lee, "Host-based dos attacks and defense in the cloud," in *HASP '17*. New York, NY, USA: ACM, 2017, pp. 3:1–3:8.

[7] Y. Park, S. Y. Chang, and L. M. Krishnamurthy, "Watermarking for detecting freeloader misbehavior in software-defined networks," in *ICNC'16*, Feb 2016, pp. 1–6.

[8] H. Liu, "A new form of dos attack in a cloud and its avoidance mechanism," in *CCSW '10*. New York, NY, USA: ACM, 2010, pp. 65–76.

[9] A. Shieh, S. Kandula, A. Greenberg, C. Kim, and B. Saha, "Sharing the data center network," in *NSDI'11*. Berkeley, CA, USA: USENIX Association, 2011, pp. 309–322.

[10] Cloud Security Alliance, "Cloud control matrix CCM v3.0.1," 2014. [Online]. Available: https://cloudsecurityalliance.org/research/ccm/

[11] J. Han, W. Zang, S. Chen, and M. Yu, "Reducing security risks of clouds through virtual machine placement," in *Data and Applications Security and Privacy XXXI*, G. Livraga and S. Zhu, Eds. Cham: Springer International Publishing, 2017, pp. 275–292.

[12] S. Al-Haj, E. Al-Shaer, and H. V. Ramasamy, "Security-aware resource allocation in clouds," in *2013 IEEE International Conference on Services Computing*, June 2013, pp. 400–407.

[13] Z. Yinqian, J. Ari, O. Alina, and R. Michael K., "Homealone: Co-residency detection in the cloud via side-channel analysis," in *S&P'11*, May 2011, pp. 313–328.

[14] Y. Han, J. Chan, T. Alpcan, and C. Leckie, "Virtual machine allocation policies against co-resident attacks in cloud computing," in *ICC'14*, 2014, pp. 786–792.

[15] M. H. Ferdaus, M. Murshed, R. N. Calheiros, and R. Buyya, *Network-Aware Virtual Machine Placement and Migration in Cloud Data Centers*. IGI Global, Hershey, USA, pp. 31–42.

[16] ONF, "Openflow switch specification," April 2013, available at:http://www.gesetze-im-internet.de/englisch_bdsg.

[17] P. Kazemian, G. Varghese, and N. McKeown, "Header space analysis: Static checking for networks," in *NSDI 12*. San Jose, CA: USENIX, 2012, pp. 113–126.

[18] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *ISCA '07*. New York, NY, USA: ACM, 2007, pp. 13–23.

[19] D. S. Marcon, R. R. Oliveira, M. C. Neves, L. S. Buriol, L. P. Gaspary, and M. P. Barcellos, "Trust-based grouping for cloud datacenters: Improving security in shared infrastructures," in *2013 IFIP Networking Conference*, May 2013, pp. 1–9.

[20] Openstack, "Openstack," https://www.openstack.org/.

[21] Opendaylight, "The OpenDaylight platform," 2015, available at: https://www.opendaylight.org/.

[22] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, Aug. 2008.

[23] J. Luna, H. Ghani, T. Vateva, and N. Suri, "Quantitative assessment of cloud security level agreements: A case study," *Security and Cryptography*, pp. 64–73, 2012.

[24] J. Luna Garcia, R. Langenberg, and N. Suri, "Benchmarking cloud security level agreements using quantitative policy trees," in *CCSW'12*. ACM, 2012, pp. 103–112.

[25] S. Bleikertz, T. Groß, M. Schunter, and K. Eriksson, "Automated information flow analysis of virtualized infrastructures." in *ESORICS*, ser. Lecture Notes in Computer Science, V. Atluri and C. Diaz, Eds., vol. 6879. Springer, 2011, pp. 392–415.

[26] S. Majumdar, T. Madi, Y. Wang, Y. Jarraya, M. Pourzandi, L. Wang, and M. Debbabi, "Security compliance auditing of identity and access management in the cloud: Application to openstack," in *IEEE CloudCom*, 2015.

[27] T. Madi, S. Majumdar, Y. Wang, Y. Jarraya, M. Pourzandi, and L. Wang, "Auditing security compliance of the virtualized infrastructure in the cloud: Application to openstack," in *CODASPY '16*. New York, NY, USA: ACM, 2016, pp. 195–206.

[28] S.-J. Moon, V. Sekar, and M. K. Reiter, "Nomad: Mitigating arbitrary cloud side channels via provider-assisted migration," in *CCS '15*. New York, NY, USA: ACM, 2015, pp. 1595–1606.