IJICTR International Journal of Information & Communication Technology Research

Volume 9-Number 3-Summer 2017(25-36)

Multi-Objective Response to Co-Resident Attacks in Cloud Environment

Farzaneh Abazari
School of Computer Engineering,
Iran University of Science and Technology,
Tehran, Iran
F_abazari@iust.ac.ir

Morteza Analoui
(Corresponding Author)
School of Computer Engineering,
Iran University of Science and Technology,
Tehran, Iran
analoui@iust.ac.ir

Hassan Takabi
Department of Computer Science and Engineering,
University of North Texas,
Denton, Texas, US
takabi@unt.edu

Received: March 24, 2017 - Accepted: September 17, 2017

Abstract— **Cloud computing is a dynamic environment that offers variety of on-demand services with low cost. However, customers face new security risks due to shared infrastructure in the cloud. Co-residency of virtual machines on the same physical machine, leads to several threats for cloud tenants. Cloud administrators are often encountered with a more challenging problem since they have to work within a fixed budget for cloud hardening. The problem is how to select a subset of countermeasures to be within the budget and yet minimize the residual damage to the cloud caused by malicious VMs. We address this problem by introducing a novel multi-objective attack response system. We consider response cost, co-residency threat, and virtual machines interactions to select optimal response in face of the attack. Optimal response selection as a multi-objective optimization problem calculates alternative responses, with minimum threat and cost. Our method estimates threat level based on the collaboration graph and suggests proper countermeasures based on threat type with minimum cost. Experimental result shows that our system can suggest optimal responses based on the current state of the cloud.**

Keywords- Cloud Computing; Attack Response; Cloud Security; Co-resident Attack; Graph Theory.

I. INTRODUCTION

Cloud computing is a dynamic environment that provides variety of on-demand services with low cost for users. However, with increased use of cloud services comes security issues that should be addressed carefully.

Takabi et al. [1] described the security challenges associated with outsourcing of data and applications, service level agreement, virtualization, and hypervisors. Since a registration process to have cloud services is very simple, virtual machines (VMs) can easily be misused by malicious users, who can stay totally anonymous in the cloud. Due to the open nature of cloud platform, security of stored data, data utilization management, access management, and trust are the primary security concerns in cloud computing [2]. Moreover, attackers can maliciously use cloud as a source of attacks for various purposes such as, malware distribution, botnet Command and Control (C&C) servers, Distributed Denial of Service (DDoS), spamming, and password cracking. Besides being source of the attack, some attacks occur inside the cloud. Due to the shared infrastructure of the cloud and improper isolation among VMs, a malicious VM is able to access the data from another VM on the same physical host [3]. This issue is much more

prevalent when the tenants of the two VMs are different users. For the rest of the paper, we keep referring to malicious VMs instead of VMs which are managed by a malicious user or containing a malware.

One of the greatest challenges in the cloud security is responding to attacks. However, to date there has been little attention on this area. Recent cyber attacks in cloud environment show the importance of intrusion response system (IRS) in the cloud. *Anwar* et al. [4] divided IRS based on the response type, into three broad categories: notification, manual and automatic response. Notification generates a response in the form of notification. Manual response permits system administrator to generate appropriate responses based on predefined set of response options. On the other hand, automatic IRS provides a real-time response based on the current system threat. However the main drawback of the automatic response is an inappropriate response that can be generated and applied. Therefore, a comprehensive mechanism is needed to suggest optimum response against the current threat. Poorly designed IRS not only threatens more cloud users, but also prevents other users from migrating to the cloud and leads to the reputation loss for the cloud provider. Our goal in this paper is to design a co-resident attack response system to mitigate the threats to the cloud resources.

Leveraging virtualization technology improves resource utilization and enables on-demand resource allocation. Despite its benefits, virtualization brings new security challenges. *Ezhilchelvan* and *Mitrani* outline the security issues in isolation among VMs that cause a malicious VM to get access to a victim VM [5]. Co-residency of VMs in a same host without a proper isolation, leads to series of attack such as side channel, information leakage, malware propagation, and Denial of Service (DoS). If one of the VMs is attacked, the security of the other VMs which are located on the same machine can be compromised by a set of co-resident attacks [6]. We have categorized co-resident attacks and their countermeasures into three groups: side channel, malware propagation, and DoS. Our method suggests proper countermeasure based on the type of the attack and prevents further damage in cloud once the attack to one of the VMs is detected.

Eliminating side channels among VMs is one of the solutions to co-residency attack [7] [8].

However, these methods required cloud platform modification. In this paper, we approach this security problem from a different view and focus on IRS as a mean to respond to cloud attacks. An effective IRS is responsible to:

- choose the right countermeasures,
- provide minimum service interruption to establish security,
- characterize security risks,
- reduce the overall risk, and
- consider response cost.

Although strong responses like stopping the attacker's VM has high ability to mitigate attacks and protect cloud network, it also has very high impact on service availability and increasing provider's costs. In order to propose a comprehensive intrusion response system, we have considered response cost and co-residency threat to select the proper response in the cloud. Intrusion response system must detect the malicious activity and choose the response option according to the nature of the attack [4].

In this paper, we implement a dynamic response solution. To do this, we generate collaboration graph to capture information about recent state of the cloud and VMs communications. Our collaboration graph edges change when a VM migrates. Also, edge's weight changes according to the time. Our solution considers two dimensions of response attributes, countermeasure's effect on the threat and its cost. Our IRS model provides responses after an attack or malicious behavior is detected in the cloud infrastructure. Type of the responses to VMs which are affected by the attacker VM depends on the amount of security threat in the network.

Recently, there have been some efforts to address intrusion response in cloud environments [9], [10], [11]. However, those studies do not investigate attack propagation patterns, and do not consider interactions among compromised VMs. To the best of our knowledge, this is the first work that incorporates the interactions among VMs in responding to co-resident attacks. Our proposed IRS takes the VMs collaboration graph as an input, estimates VM's threat level and employs multi-objective optimization techniques to suggest the best countermeasure for reducing the overall threat in cloud. In other words, this paper aims to minimize the threat levels for all VMs at a reasonable cost.

The rest of the paper is organized as follows. We explore the related work and describe co-resident attacks in Section II. In Section III, the problem formulation and system model are introduced, followed by system response model. In Section IV we conduct set of experiments to evaluate our model and discuss the results, and finally, Section V concludes the paper.

II. Related Work

A considerable amount of literature focuses on designing new methods to detect co-residency in a host. *Ristenpart* et al. introduced this concept for the first time. They exploited VM placement method in the Amazon EC2 cloud infrastructure in order to place VM in a same host with a target VM [12]. *Yu* et al. presented a method to detect co-residency via cache-based side channel attacks [13]. *Alarifi* and *Wolthusen* [14], [15] presented Cloud-Internal DoS (CIDoS) attack, that exploits co-resident VMs to perform DoS on victim host.

The most similar work to this paper was presented by *Altunay* et al. in [16]. They take the collaboration network graph and use optimization techniques to calculate the threat level for each grid participant. They answer to threat by closing sites or monitoring links among sites.

A. Co-resident Attacks

The main protected entities in cloud infrastructure are the VMs running on the physical machines in a

data center, which contain valuable information such as users' data and applications. The threat we focus on this paper is a malicious VM which is located on the physical machines that victim VMs run. In order to provide security for cloud users, as soon as a suspicious activity such as port scanning, resource overconsumption, or malicious traffic generation, is detected in the cloud system, it should be determined whether the detected activity is malicious or not. In the case of malicious activity, type of the attack is important to respond properly.

In this section, attack types among co-resident VMs are introduced. We have divided these attacks into three categories. First category includes information disclosure and other side channel attacks to transfer sensitive information between attacker and victim VMs that may cause irreversible problems for other tenants in the cloud. The next category contains communication threat among VMs that could be leveraged by attackers to propagate worms inside the cloud. In the last type of the attack, attacker's goal is to obtain an unfair share of a resource [17], [18] that leads to DoS in victim VMs on that host.

1) Side Channel

In this type of attacks, attacker collects timing information, shared processor's cache, and power consumption information by gaining access to the physical machine. This collected information can be employed covertly in exploiting the hypervisor and finding sensitive information about the victim VM. Attacker can access this information just by creating a VM in the same physical machine that hosts the victim VM.

Bates et al. [19] presented co-resident watermarking which is a traffic analysis attack to compromise isolation. In this paper a malicious VM injects a watermark signature into the network flow of a victim VM. Recently, a vulnerability in virtual floppy drive code which is used by many virtualized platforms, allows an attacker to escape from his VM and potentially obtain code-execution access to the host. This can lead to access to all other VMs running on that host [20].

2) Malware Propagation

Any malware with a network component such as worms is able to propagate to wherever their addressing allows them to; so, the communication among VMs and their access to the network may lead to malware propagation in the cloud infrastructure [21]. *Mazhar Ali* et al. [3] outlined two types of communications among VMs: external and internal. The former type exists between user and the cloud, while the later occurs inside the virtualized environment. Virtual network that is established among VMs over a physical network, enables internal communication. Virtual network is responsible for managing communications through VMs [22].

The software-based network components such as virtual switches provide networking among VMs over the same physical machine. Since security mechanisms such as IDS and firewalls over the physical machine are blind to traffic over virtualized network, malicious traffic can pass through the network without being detected. This problem is also mentioned in a report released by Symantec [23].

Similar configuration for VMs in the cloud such as virtualization techniques or operating systems leads to the same vulnerability. *Balduzzi* et al. [24] analysis on 5303 Amazon VM images confirmed that 98% of Windows and 58% of Linux images in Amazon EC2 contain software with critical vulnerabilities. Therefore, if a VM gets infected by a malware, multiple VMs could be compromised and as a result, malware propagation is probable [25]. Moreover, installing vulnerable software on VMs by cloud users could increase malware propagation. In another scenario, a malicious user uploads an infected image in the cloud image storage. All VMs that instantiate from the infected image will become a source of introducing malware to the cloud.

3) Denial of Service

The lack of proper isolation in hypervisors leads to new security threats which enable a malicious user to exploit the resource contention between co-located VMs and affect the execution of an application running on a co-resident VM. Since more resources are being shared in cloud environment, particularly in elastic cloud, which could provide users unlimited resource, DoS attacks can be much more influential [26].

Chiang et al. demonstrated a new type of security vulnerability caused by competition between virtual I/O intensive workloads [27]. In this attack, malicious user could slow down the execution of an application in a victim co-resident VM by leveraging the competition for shared I/O resources such as hard-drive and/or network bandwidth.

Varadarajan et al. [17] showed that performance of a cache sensitive workload on a VM can be degraded more than 80% due to interference from another VM. They called this resource-freeing attack. Attacker changes the workload of the victim VM to make victim free up the competitive resource.

There are some attack response methods that are presented for cloud infrastructure. *Vieira* et al. [10] proposed intrusion response autonomic system (IRAS) that uses a utility function to suggest the best response in order to reduce the consequences of the attacks in the cloud. *Raju* and *Geethakumari* [11] proposed a framework for detecting the attacker. They form cluster of VMs that are prioritized based on the interactions with other instances and their resource consumption. As soon as an alert is generated then the most related cluster for further analysis with forensics techniques is selected. *Szefer* et al. [9] proposed a real time cloud intrusion prevention model. Their goal was protecting VMs in the cloud. When an initial sign of potential attack is detected in the network, their system starts responding to attacks. They introduced two kinds of mechanisms: prevention and detection. Since implementing each mechanism has its own cost and time, their method suggested the best response that can be effective in the cloud situation. *Balasubramanian* et al. in [28] introduced an approach for responding to networked computing environment threats such as cloud computing. They classified VM threats into

three risk levels. High risk threats are related to co-resident VMs that are owned by the same customer. Once a threat is detected in a VM instance, a first preventive measure is performed on the high risk VMs. Preventive measures depends on the strength attribute associated to each VM.

III. Problem Formulation and System Model

The majority of security concerns in the virtualized infrastructure relate to the co-resident VMs owned by different users. Cloud providers must maintain a maximum level of isolation between VMs in a same host. The lack of proper isolation leads to new security threats which enable attacker to exploit co-resident VMs.

Our system consists of two modules: 1) threat estimation, and 2) response selection. In order to design an effective response system, we need to evaluate the total threat after an attack occurs in the cloud. To do this, a collaboration graph captures the behavior of VMs [29]. Then, the system estimates the threat level of all of the VMs based on the collaboration graph and finally suggests the optimal response to reduce the overall threat, according to the VM's threat and cloud provider's budget.

In this section, we briefly introduce our new intrusion response system in three phases:

- cloud collaboration graph,
- estimating threat level, and
- modeling optimal response.

Table I summarizes the notation we use throughout this paper.

TABLE I. Parameter Definition

Name	Definition
N	The total number of virtual machines
Q	Number of countermeasures
V	Set of virtual machines
M	Set of malicious VMs
S	Set of stealthy VMs
G	VM Collaboration Graph
C	Matrix q×3 of countermeasures
t_i	Vector 1×3 of VM_i threat level
T	Total threat in Cloud
RC	Matrix q×1 of Response cost vector
x	Optimization problem solution matrix (n×q)

A. Cloud Collaboration Graph

Collaboration graph model captures non-obvious links between resources in a defined network based on security assessment of observable interactions [16]. Based on this graph, we can measure the security threat through VMs and contain the spread of malicious VM across the cloud by responding to attack optimally.

Interactions among VMs inside the cloud can be represented by a collaboration graph where the nodes represent VMs and the edges represent same host for VMs [29]. Nodes in the graph present a VM and a link between nodes shows that both VMs are located in the same physical machine for a period of time.

As it is shown in Fig.1, VMs interaction are modeled as an undirected weighted graph $G := (V, E)$, where V represents the set of VMs and E is the set of edges $\{i, j\}$ for $\{i, j\} \in V$. An edge $\{i, j\} \in E$ exists if and only if VM_i and VM_j are located on the same host and the weight W_{ij} for edge $\{i, j\}$ represents duration of this co-residency. The set V contains two types of VMs: malicious and stealthy VMs that are represented by sets M and S, respectively. Since each VM can belong to either set M or S, the following equation is always true:

$$V = M \cup S \quad (1)$$

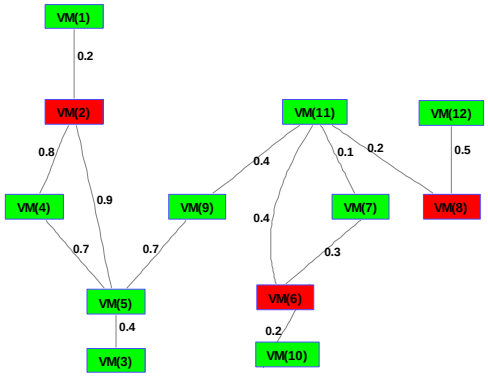

Figure 1. Cloud Collaboration Graph for Toy Network

B. Estimating Threat Levels

In order to estimate threat level in the cloud, first we assume that collaboration graph G and the set of detected malicious VMs M are given. The threat level for stealthy VMs completely depends on the graph G. *Balasubramanian* et al. in [28] classified VM threats into three risk levels. High risk threats are related to co-resident VMs that are owned by the same customer. Co-resident VMs related to the same industry have medium risk. Low risk VMs are related to the same type of service or application.

We assume that when a malicious behavior has been detected in a running VM, the threat level in co-resident VMs increase based on the malicious VM behavior. Note that the main concern of this work is reducing the threat caused by co-resident VMs, rather than cloud attack detection which has been the subject of many studies in cloud computing systems. When a malicious user wants to extract information from a victim by side channel attacks, there are resource usage, system calls and cache miss abnormalities in malicious VM behavior. For example, *Sundareswaran* et al. proposed a method to detect these abnormalities in a physical machine to determine malicious VM [30]. Generally detection can be done through continuous cloud monitoring or receiving user's complaints.

In this paper, the threat level in VM_i can be expressed as a triple $t_i = < t_i^{SC}, t_i^{MP}, t_i^{DoS} >$, where t_i^{SC} is the probability of existing side channel attack, t_i^{MP} expresses the malware propagation possibility and t_i^{DoS} determines the probability of performing denial of service on the VM_i. These values are determined by the cloud administrator based on the VM behavior.

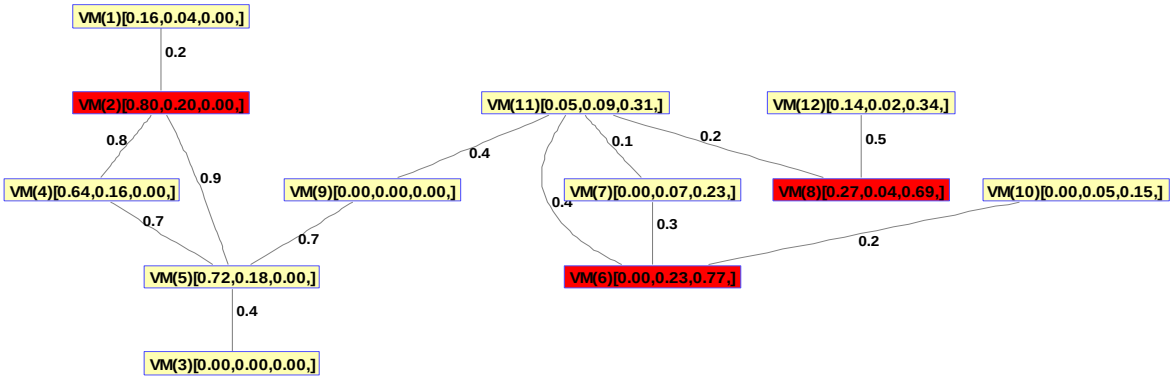

Figure 2. VM Threat Level

In side channel attack, malicious VM collects timing information, shared processor's cache, and power consumption information. In malware propagation, malicious VM scans the network to find vulnerable VM. In addition, DoS attack prevents victim machine to access a specific resource for a time period. The duration of Co-residency is important in all of these attack types. For each attack type we consider the co-resident VM with higher probability of being malicious. If malicious VM stay longer in the victim physical machine, the probability of being attacked is higher [31] [32] [33].Therefore, the threat level at VM_i ($VM_i \in S$) can be obtained by solving Equation 2. As you can see in Equation 2, malware propagation threat is for every VM that had malicious neighbor during its running time. However, other threats propagate in co-resident VMs.

$$
\begin{aligned}
& t_i =< t_i^{SC}, t_i^{MP}, t_i^{DoS} >; \\
& t_i^{SC} = \max_{j|j=1, VM_j \in M, Host(i)=Host(j)} (W_{ij} * t_j^{SC}) \\
& t_i^{MP} = \max_{j|j=1, VM_j \in M} (W_{ij} * t_j^{MP}) \\
& t_i^{DoS} = \max_{j|j=1, VM_j \in M, Host(i)=Host(j)} (W_{ij} * t_j^{DoS})
\end{aligned} \quad (2)
$$

The co-residency time (T_{cr}) is captured for each two co-resident VMs. To normalize the co-resident time we use Equation (4). T_{max} is the maximum time that malicious VM needs to attack its neighbors. This parameter can be estimated by experts based on their analysis on cloud.

$$
W_{i,j} = \begin{cases} 1 & \text{if } T_{cr} \geq T_{\max} \\ \dfrac{T_{cr}}{T_{Max}} & \text{if } T_{cr} < T_{\max} \end{cases} \quad (3)
$$

Total threat in the cloud that is caused by a set of malicious VMs is calculated by Equation (4).

$$
T = \sum_{t=1}^{n} t_i \quad (4)
$$

In Fig.2, threat vector t_i for each VM is illustrated.

C. Modeling Optimal Response

Modeling optimal response to malicious VMs threat is the last step of our system. Once an incident has been detected in the cloud, a quick and effective response must be undertaken to mitigate its harmful impacts [34].

Attack response system determines which VMs should be protected and how. In other words, it determines the optimal response based on the VMs locations and interactions, countermeasures and type of the attack. Response must reduce the threat to uncompromised VMs as much as possible and maintain the response cost on a reasonable value.

Cloud infrastructure consists of several security mechanisms such as network/host based intrusion detection systems, firewalls and resource management systems. In order to provide security for cloud users, as soon as a suspicious activity such as port scanning, resource overconsumption, or malicious traffic generation is detected in the cloud system, it should be determined whether the detected activity is malicious or not. In the case of malicious activity a proper response should be chosen.

In order to respond to co-resident attacks, a vector of countermeasures $C =< c_1, c_2, \ldots, c_q >$ is defined in Equation 5. Each countermeasure has its own restrictions in terms of the time taken to perform the response, feasibility, and computation cost; In our model, response cost vector $RC =< rc_1, rc_2, \ldots, rc_q >$ is determined by the cloud provider based on the power consumption, performance overhead and probability of SLA violation, (see Equation 6). rc_i is the cost associated by countermeasure c_i. In addition to the cost, security protections that countermeasures offer are important. Each countermeasure can protect the VM from some types of co-resident attacks. In this paper, a novel model is introduced that suggests the best countermeasure for each VM based on its threat.

$$(5)\quad C=\begin{bmatrix}c_1\\c_2\\.\\.\\.\\c_q\end{bmatrix}=\begin{bmatrix}c_1^{SC}c_1^{MP}c_1^{D0S}\\c_2^{SC}c_2^{MP}c_2^{D0S}\\.\\.\\.\\c_q^{SC}c_q^{MP}c_q^{D0S}\end{bmatrix}$$

$$(6)\quad RC=\begin{bmatrix}rc_1\\rc_2\\.\\.\\.\\rc_q\end{bmatrix}$$

Each attack can be prevented by specific countermeasures. InTABLE II. , we present a list of countermeasures for each co-resident attack. In this paper responding to co-resident attacks is divided into five scenarios. First scenario incorporates VM migration that is capable of responding to all of the co-resident threats. Second scenario responds to side channel attacks that cause sensitive information from other users covertly accessed. This scenario is based on an approach that adds latency to potentially malicious operation [35]. In the case of side channel attack there are two sub scenarios: response to attacker and response to victim. If a VM is a member of *S*, response can be done by changing the access pattern to the shared resources like memory, CPU and cache to a consistent pattern. The third scenario can mitigate VM communication threat [25]. The forth scenario tries to limit resource access by malicious VM. Final scenario reduces the allowed maximum traffic rate of the associated VM to control malware propagation or DoS attack [28].

D. Problem Formulation

Cloud intrusion response system (CIRS), could be modeled as a multi-objective optimization problem. Multi-objective optimization determines the optimal response to a given problem by optimizing problem objectives and preserving constraints. CIRS goal is to concurrently minimize (1) the overall threat inspired by co-resident attacks, as well as (2) the overall response cost. Since these two objectives are dependent and conflict with each other, minimizing first objective results in compromising the other. For example, achieving lower threat in cloud needs more countermeasures to be applied, whereas having lower cost causes higher threat in the cloud.

To mathematically formulate our problem, assume that set of malicious VMs *M* with their threat vector, set of stealthy VMs *S*, pool of countermeasures *C* and their associate costs *RC* are given. Based on this model, CIRS is defined as finding responses to VM threats to minimize the following two objective functions.

$$(7)\quad \begin{cases} Min\ Threat=\sum_{i=1}^{n}(([111]-\sum_{j=1}^{q}x_{i,j}*C_j)*t_i)\\ Min\ Cost=\sum_{i=1}^{n}x(i)*RC\\ subj.to\\ x_{i,j}=\{0,1\};i=[1,n],j=[1,q]\end{cases}$$

Optimization problem outputs *x* that determines which countermeasures should be used for each VM. Since optimizing each object may worsen the other one, multi-objective optimization offers a set of answers. In case of conflicting objective functions, there exist a number of pareto optimal solutions [36]. Generally, an output *x** is said to be pareto optimal, if all other outputs have a higher value for at least one of the objective functions. A pareto curve or pareto front is a set of pareto optimal solutions. In Fig.3, an example of pareto front is illustrated.

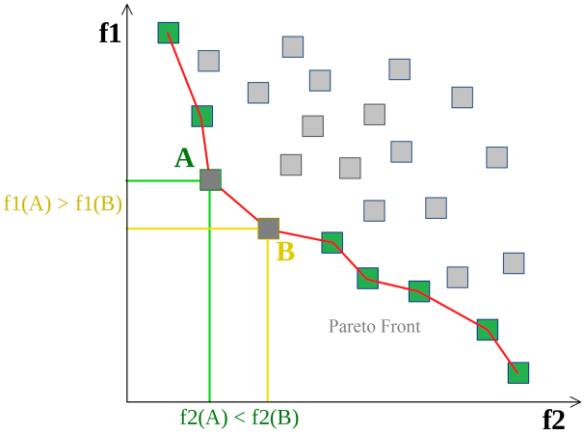

Figure 3. Pareto Front

IV. EXPERIMENTS

In this section, we have conducted a set of experiments using MATLAB 8.3.0 to evaluate our CIRS. The model efficiency is evaluated in terms of the threat (see Equation 3) and cost.

A. Data set

Along with the model discussed in Sections III, we constructed two data sets for our numerical experiments. The first data set is a toy cloud network with 12 VMs. Fig. 1 shows cloud collaboration graph for Toy Network.

The next data set is derived from real life workload traces from CoMon project, a monitoring infrastructure for PlanetLab. These data could be accessed from github repository [37] . In PlanatLab data set we have CPU utilization by more than a thousand virtual machines from servers located in five hundred different places all around the world during ten random days. We randomly chose one day out of those ten days data and schedule workloads on 200 VMs by means of CloudSim [38]. CloudSim is a cloud

simulation toolkit that implements several algorithms for VM allocation and selection. We run 200 PlanetLab workloads on 200 VMs with *Threshold* approach for VM allocation and *Minimum Migration Time* method for VM selection and capture their behavior during 6 hours. Due to the dynamic nature of the cloud, VMs migrate among hosts. In each time stamp VMs position are captured and weights in the graph will be updated based on the Equation (3).

Cloud Collaboration Graph for PlanetLab workload is illustrated in Figure 4.

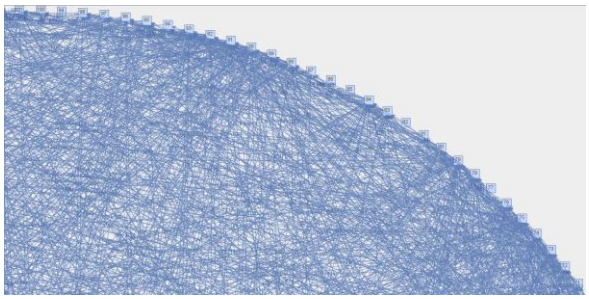

Figure 4. Cloud Collaboration Graph for PlanetLab Network

To implement the problem in MATLAB, we use multi-objective optimization approach called goal attainment by means of *fgoalattain* toolbox [39]. The goal attainment method is represented mathematically in the Equation 8.

$$\begin{aligned} &\min_{\gamma,x} \gamma \\ &subj.to \\ &F_1(x) - w_1\gamma \le F_1^* \\ &F_2(x) - w_2\gamma \le F_2^* \end{aligned} \quad (8)$$

$\{F_1^*, F_2^*\}$ determines goal point of the problem. Since we consider F_1 as total threat and F_2 as total cost, the goal point in our problem is *{0,0}*. The weighting vector $w=<w_1,w_2>$ presents the importance of each object ($w_1+w_2=1$). Actually w enables the problem designer to show a measure of the tradeoffs between the objectives. The term $w_i\gamma$ introduces a parameter of *slackness* into the problem that causes the goals don't rigidly met. x is the solution of optimization problem. Each value for vector w returns a unique solution. To achieve the pareto front, we take weight vector $w =< \beta, 1-\beta >$ for β from 0 through 1, then we solve the goal attainment problem for various values of weights.

B. Parameters

Our experiments are performed on a computer with Intel Core i5 processor (2.50 GHz), 6 GB RAM, and 512 GB disk space. The simulation result shows that our method can generate optimal responses to the co-resident attack.

Our experiment consists of 5 countermeasures that are introduced in TABLE II. The following sets of values in countermeasure matrix *C* and associated cost matrix *RC* are used in our implementation:

$$C = \begin{bmatrix} 1\,1\,1 \\ 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \\ 0\,1\,1 \\ 0\,0\,0 \end{bmatrix} \quad (9)$$

$$RC = \begin{bmatrix} 1 \\ 0.5 \\ 0.2 \\ 0.4 \\ 0.3 \\ 0 \end{bmatrix} \quad (10)$$

Values in countermeasure matrix *C* in Equation 9 can be obtained from TABLE II. If *C(i,j) = 1*, countermeasure *i* can response to threat *j*. For example, *C(3,2)=1* means that traffic isolation can mitigate malware propagation threat or *C(5,3)=1* means that limiting web traffic rate can mitigate DoS threat. Values in response cost matrix *RC* in Equation 10 can be obtained from last column of Table II. For example, *RC(3)=0.2* means that traffic isolation have a low cost to deploy. Last row in both of the matrices refers to no response.

C. Toy Network Results

To achieve pareto optimal solutions, we solve Equation 7 with goal attainment method. Points in Fig.5 present 100 pareto optimal responses to the threat in cloud. The response associated with one of the points that is specified by the arrow, has threat level equal to *1.01* and cost equal to *3.4*. As we can see in Fig.5, there is a trade-off between overall threat and cost.

By analyzing the response *x*, suggested countermeasures for each VM will be determined. Answer *x* that is generated by optimization problem is shown in Equation 11.

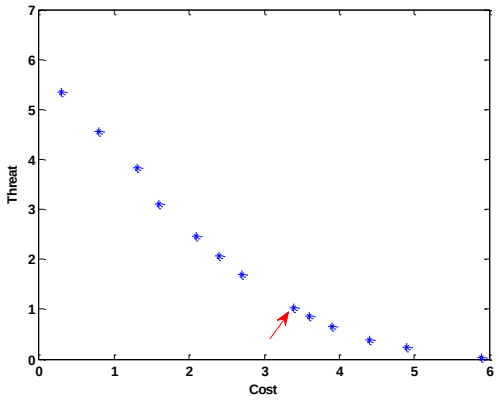

Figure 5. Pareto Optimal Responses

TABLE II. MITIGATING METHODS FOR CO-RESIDENT ATTACK

Num	VM State	Mitigation Methods	Description	SC	MP	DoS	Cost
1	VMєM/S	VM migration	Frequently migrating VMs	yes	yes	yes	(Increase power consumption and probability of SLA violation, performance overhead) => High
2	VMєM	Alter hypervisor [35]	Adding latency to potentially malicious operations	yes	no	no	(Performance overhead) => Medium
2	VMєS	Change application and OS component access patterns	Change access pattern to shared resources like memory, CPU and cache to a consistent pattern. This countermeasure prevent attacker to gain information from behavioral attributes	yes	no	no	(Performance overhead) => Medium
3	VMєM/S	Traffic isolation [25]	Utilizing traffic engineering capabilities of virtual switches to reconfigure the virtual network	no	yes	no	(Network reconfiguration overhead)=> Low
4	VMєM/S	Limit resource allocation [28]	Change hypervisor configurations to lower the allowed maximum computation load or cache capacity	no	no	yes	(Increase power consumption) =>Medium
5	VMєM/S	Limit web traffic rate [28]	Reduce the allowed maximum traffic rate of the associated VM	no	yes	yes	(Network reconfiguration overhead) =>Low

$$(11)\ x = \begin{bmatrix} 0\,0\,0\,0\,0\,0 \\ 0\,1\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \\ 0\,1\,1\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1\,0 \end{bmatrix}$$

Each row in the matrix *x* refers to a specific VM. Columns show countermeasure for VMs. For example *x(6,5)=1* means that *VM(6)* should use countermeasure *5* that can mitigate malware propagation and DoS attacks. Fig.2 confirms that *VM(6)* has higher threat in DoS and MP attack and as a matter of fact, countermeasure *5* mitigates DoS and MP threat by limiting web traffic rate. Rows 1,3,9, and 10 in matrix *x* are zero, so no countermeasure is suggested by system for VMs 1,3,9, and 10. First countermeasure is not suggested to be applied in any VM, because it is not cost effective. Since countermeasure *5* can prevent two types of threat with lower cost than countermeasure *4*, our response system suggests countermeasure *5* whenever a DoS threat is detected. As we can see in Fig.2, these VMs have minimum risk among others. Note that if countermeasure *2* is selected for *VM(k)*, depends on the type of the VM (malicious or stealth) proper countermeasure based on Table II is selected.

D. Results in PlanetLab Network

In this experiment malicious VMs with random threat are generated among 200 VMs with probability 0.1. Points in Fig.6 present 100 pareto optimal responses to the threat in cloud that is executing PlanetLab workload.

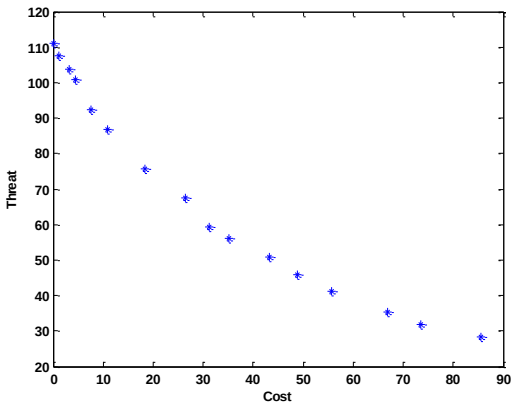

Figure 6. Pareto Optimal Responses

x is a 200×6 matrix that shows suggested countermeasures for each VM. In order to show the efficiency of the model two VMs are selected randomly and the suggested countermeasure is analyzed (see Table III).

TABLE III. CIRS SAMPLE OUTPUT

VM(89)	T_{89} (0.75,0.25,0)	x(89) (0,1,0,0,0,0)
VM(129)	T_{129} (0.11,0.5,0.39	x(129) (0,0,0,0,1,0)

x(129,5)=1 means that *VM(129)* should use countermeasure *5* that can mitigate malware propagation and DoS attacks. *VM(129)* has higher threat in DoS and MP attack. The solution to the threat

reveals that CIRS generate optimal response to threats and suggests proper countermeasures.

TABLE IV. PERFORMANCE COMPARISON AMONG DIFFERENT METHOD

	Our Method	[10]	[11]	[9]	[28]
Optimized	Support	Support	Support	Support	Not Support
Response Cost	Support	Not Support	Not Support	Support	Support
Response Type	Support	Support	Not Support	Support	Support
Automatic	Support	Support	Not Support	Support	Support
VM Communication	Support	Not Support	Support	Not Support	Support

E. Complexity Analysis

In this section, we analyze computation efficiency of our model. The proposed system consists of two modules: 1) threat estimation, and 2) response selection. Threat estimation algorithm estimates threat for each graph node. For each node, threat is calculated based on the neighbors' threat in collaboration graph. So the complexity of threat estimation is $O(|n|*|k|)$, where $|n|$ is the number of VMs and $|k|$ represents average degree in graph. Complexity in response selection algorithm, is $O(|n|*|q|)$, where $|q|$ represents the number of countermeasures. So the total complexity of our approach is $O(|n|*|k|) + O(|n|*|q|)$.

F. Comparison with the Existing Methods

To clearly present the accuracy of our method, the brief comparison of related references is shown in TABLE IV. Various methods are compared based on five different measures. Some methods generate best response by means of optimization theory. Considering response cost and type make the model applicable in real world. In addition, our approach considers VM interaction by means of collaboration graph. As shown in Table IV, our work is able to provide a response that takes into consideration the impacts of the VM interaction, response type, and cost across the Cloud environment. We can get a conclusion that the accuracy in our approach is more comprehensive than other methods. In addition the complexity of our approach is polynomial that makes it applicable in real world.

G. Possible Applications

In this section, we enhance our results to show the empirical aspect of this paper. As we mentioned before, the goal is to response to co-resident threat in the cloud. We have introduced a model to calculate threat and cost based on the VMs interaction. By means of multi-objective optimization, the optimal responses are calculated. If a cloud provider has a limited budget, our approach can suggest the best response with minimum threat and cost less than the budget.

Example Cloud administrator detects malicious behavior in some particular VMs (see Fig.*2*). Cloud provider wants to cost *2* units to mitigate these threats. As it is depicted in Fig.5, the optimal response reduces the threat to *2.7* with cost *1.7*. This response is presented in Equation 11.

$$(12)\quad x = \begin{bmatrix} 0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1\,0 \end{bmatrix}$$

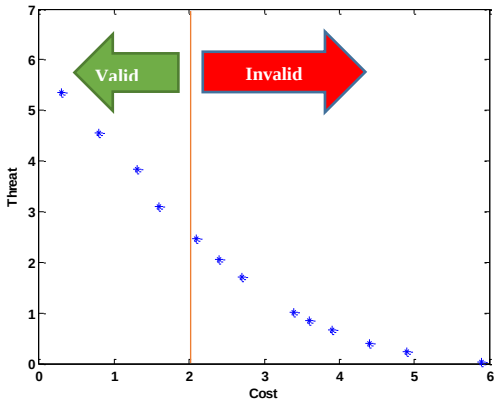

Figure 7. Responses with Defined Cost

V. CONCLUSION

Cloud environment consists of several virtualized data centers. Virtual machines in these data centers, similar to any physical machine, are subject to security threat. In this work, we have proposed a systematic attack response system to co-resident VMs threats in virtualized infrastructure. We incorporated VMs interactions to calculate VM threat. Our system generates optimal response via multi-objective optimization and then use goal attainment algorithm to solve it. For future work, we plan to response optimally to other threats, such as attacks from outside the cloud. In addition, precisely estimating the threat based on the malicious VM behavior could be another future work.

REFERENCES

[1] Hassan Takabi, James BD Joshi, and Gail-Joon Ahn, "Security and privacy challenges in cloud

computing environments," *IEEE Security& Privacy*, pp. 24--31, 2010.

[2] Zahir Tari, Xun Yi, Uthpala S Premarathne, Peter Bertok, and Ibrahim Khalil, "Security and Privacy in Cloud Computing: Vision, Trends, and Challenges," *Cloud Computing, IEEE*, vol. 2, pp. 30-38, 2015.

[3] M. Ali, S. U. Khan, and A. V. Vasilakos, "Security in cloud computing: Opportunities and challenges," *Information Sciences*, vol. 305, pp. 357–383, 2015.

[4] Shahid Anwar et al., "Response option for attacks detected by intrusion detection system," in *Software Engineering and Computer Systems (ICSECS), 2015 4th International Conference on*, 2015, pp. 195-200.

[5] Paul and Mitrani, Isi Ezhilchelvan, "Evaluating the Probability of Malicious Co-residency in Public Clouds," *Cloud Computing, IEEE Transactions on*, 2015.

[6] Y. Han, J. Chan, T. Alpcan, and C. Leckie, "Using Virtual Machine Allocation Policies to Defend against Co-resident Attacks in Cloud Computing," *Dependable and Secure Computing, IEEE Transactions on*, 2015.

[7] J. Szefer, E. Keller, R. B. Lee, and J. Rexford, "Eliminating the hypervisor attack surface for a more secure cloud," , 2011, pp. 401–412.

[8] T. Kim, M. Peinado, and G. Mainar-Ruiz, "Stealthmem: system-level protection against cache-based side channel attacks in the," in *Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)*, 2012, pp. 189–204.

[9] Jakub and Jamkhedkar, Pramod and Perez-Botero, Diego and Lee, Ruby B Szefer, "Cyber defenses for physical attacks and insider threats in cloud computing," in *Proceedings of the 9th ACM symposium on Information, computer and communications security*, 2014, pp. 519--524.

[10] Kleber MM Vieira, Daniel SM Pascal Filho, Carlos B Westphall, Joao Bosco M Sobral, and Jorge Werner, "Providing response to security incidents in the cloud computing with autonomic systems and big data," in *Telecommunications(AICT) , 2015 11th Advanced International Conference on*, 2015.

[11] Bksp Kumar Raju and G Geethakumari, "A novel approach for incident response in cloud using forensics," in *Proceedings of the 7th ACM India Computing Conference*, 2014, p. 20.

[12] Thomas Ristenpart, Eran Tromer, Hovav Shacham, and Stefan Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," in *Proceedings of the 16th ACM conference on Computer and communications security*, 2009, pp. 199--212.

[13] Si Yu, Gui Xiaolin, Lin Jiancai, Zhang Xuejun, and Wang Junfei, "Detecting vms co-residency in cloud: Using cache-based side channel attacks," *Elektronika ir Elektrotechnika*, vol. 19, no. 5, pp. pp. 73–78, 2013.

[14] Suaad Alarifi and Stephen Wolthusen, "Mitigation of cloud-internal denial of service attacks," in *Service Oriented System Engineering (SOSE), 2014 IEEE 8th International Symposium on*, 2014, pp. 478--483.

[15] Suaad Alarifi and Stephen D Wolthusen, "Robust coordination of cloud-internal denial of service attacks," in *Cloud and Green Computing (CGC), 2013 Third International Conference on*, 2013, pp. 135--142.

[16] Mine Altunay, Sven Leyffer, Jeffrey T Linderoth, and Zhen Xie, "Optimal response to attacks on the open science grid," *Computer Networks*, vol. 55, no. 1, pp. 61--73, 2011.

[17] Venkatanathan Varadarajan, Thawan Kooburat, Benjamin Farley, Thomas Ristenpart, and Michael M Swift, "Resource-freeing attacks: improve your cloud performance (at your neighbor's expense," in *Proceedings of the 2012 ACM conference on Computer and communications security*, 2012, pp. 281-292.

[18] Fangfei Zhou, Manish Goel, Peter Desnoyers, and Ravi Sundaram, "Scheduler vulnerabilities and coordinated attacks in cloud computing," *Journal of Computer Security*, vol. 21, pp. 533-559, 2013.

[19] Adam Bates et al., "On detecting co-resident cloud instances using network flow watermarking techniques," *International Journal of Information Security*, vol. 13, pp. 171-189, 2014.

[20] "CVE-2015-3456," Technical Report 2015.

[21] Farzaneh Abazari, Morteza Analoui, and Hassan Takabi, "Effect of anti-malware software on infectious nodes in cloud environment," *Computers & Security*, 2016.

[22] Candid Wueest, "Security for Virtualization: Finding the Right Balance," Kaspersky Lab, 2012.

[23] Candid Wueest, "Threats to virtual environments," Symantec, 2014.

[24] Marco Balduzzi, Jonas Zaddach, Davide Balzarotti, Engin Kirda, and Sergio Loureiro, "A security analysis of amazon's elastic compute cloud service," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, 2012.

[25] Chun-Jen Chung, Pankaj Khatkar, Tiany Xing, Jeongkeun Lee, and Dijiang Huang, "NICE: Network intrusion detection and countermeasure selection in virtual network systems," *Dependable and Secure Computing, IEEE Transactions on*, 2013.

[26] Wanchun Dou, Qi Chen, and Jinjun Chen, "A confidence-based filtering method for DDoS attack defense in cloud environment," *Future*

Generation Computer Systems, vol. 29, pp. 1838-1850, 2013.

[27] Ron C Chiang, Sundaresan Rajasekaran, Nan Zhang, and H Howie Huang, "Swiper: Exploiting virtual machine vulnerability in third-party clouds with competition for I/O resources," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 26, pp. 1732-1742, 2015.

[28] Swaminathan Balasubramanian, Matthew M Lobbes, Brian M O'connell, and Brian J Snitzer, "Automated Response To Detection Of Threat To Cloud Virtual Machine," US Patent 20,160,094,568, March 2016.

[29] Farzaneh Abazari and Morteza Analoui, "Exploring the effects of virtual machine placement on the transmission of infections in cloud," in *Telecommunications (IST), 2014 7th International Symposium on*, 2014, pp. 278--282.

[30] Smitha and Squcciarini, Anna C Sundareswaran, "Detecting malicious co-resident virtual machines indulging in load-based attacks," *Information and Communications Security*, pp. 113--124, 2013.

[31] Ahmed Osama Fathy Atya et al., "Malicious Co-Residency on the Cloud: Attacks and Defense," in *IEEE INFOCOM*, 2017.

[32] Zhang et al., "A comprehensive study of co-residence threat in multi-tenant public PaaS clouds," in *Information and Communications Security*, 2016, pp. 361-375.

[33] Qian Sun, Qingni Shen, Cong Li, and Zhonghai Wu, "SeLance: Secure Load Balancing of Virtual Machines in Cloud," in *Trustcom/BigDataSE/I SPA, 2016 IEEE*, 2016, pp. 662-669.

[34] Richard Baskerville, Paolo Spagnoletti, and Jongwoo Kim, "Incident-centered information security: Managing a strategic balance between prevention and response," *Information \& management*, vol. 51, pp. 138--151, 2014.

[35] Jingzheng Wu, Liping Ding, Yuqi Lin, Nasro Min-Allah, and Yongji Wang, "Xenpump: a new method to mitigate timing channel in cloud computing," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, 2012, pp. 678--685.

[36] Kalyanmoy Deb, "Multi-objective optimization," in *Search methodologies.*: Springer, 2014, pp. 403--449.

[37] Beloglazov. (2017) Github. [Online]. https://github.com/beloglazov/planetlab-workload-traces

[38] Rodrigo N Calheiros, Rajiv Ranjan, Anton Beloglazov, C, and Rajkumar Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," in *Software: Practice and Experience*, 2011, pp. 23--50.

[39] FW Gembicki, "Vector optimization for control with performance and parameter sensitivity indices," Case Western Reserve Univ., Cleveland, Ohio, Ph. D. Thesis 1974.

[40] Mazhar Ali, Samee U Khan, and Athanasios V Vasilakos, "Security in cloud computing: Opportunities and challenges," *Information Sciences*, pp. 357--383, 2015.

[41] Atya et al., "Malicious co-residency on the cloud: Attacks and defense," in *INFOCOM 2017-IEEE Conference on Computer Communications*, 2017, pp. 1-9.

Farzaneh Abazari is a Ph.D. candidate in the Department of Computer Engineering at Iran University of Science and Technology. She was a visiting scholar in Department of Computer Science and Engineering of University of North Texas, Denton, TX, USA. She received her B.Sc. (2008) and M.Sc. (2011) from Amirkabir University of Technology (Tehran Polytechnic) in Software engineering and Information Security. Her research interests include cloud computing security and malware propagation.

Morteza Analoui is an Associate Professor in the Department of Computer Engineering at Iran University of Science and Technology, where he is also director of the Networking Laboratory. He received a B.Sc. degree in electrical engineering from Iran University of Science & Technology and a Ph.D. degree in telecommunication from Okayam University, Japan. Dr. Analoui has been an Assistant Professor at Electrical and Electronic Engineering of Okayama University, Japan, and Electrical & Computer Engineering Department of Tarbiat Modaress University, Tehran, Iran. His research interests include modeling and performance evaluation, network protocols and architecture, network measurement, virtualization and cloud computing.

Hassan Takabi is an Assistant Professor of Computer Science and Engineering at the University of North Texas, Denton, TX, USA. He is director and founder of the Information Security and Privacy: Interdisciplinary Research and Education (INSPIRE) Lab and a member of the Center for Information and Computer Security (CICS). His research is focused on various aspects of cybersecurity and privacy including advanced access control models, insider threats, cloud computing security, mobile privacy, privacy and security of online social networks, and usable security and privacy. He is a member of ACM and IEEE.