Modeling and Mitigating the Insider Threat of Remote Administrators in Clouds

Nawaf Alhebaishi, Lingyu Wang, Sushil Jajodia, Anoop Singhal

Modeling and Mitigating the Insider Threat of Remote Administrators in Clouds

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

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

Abstract. As today's cloud providers strive to attract customers with better services and less downtime in a highly competitive market, they increasingly rely on remote administrators including those from third party providers for fulfilling regular maintenance tasks. In such a scenario, the privileges granted for remote administrators to complete their assigned tasks may allow an attacker with stolen credentials of an administrator, or a dishonest remote administrator, to pose severe insider threats to both the cloud tenants and provider. In this paper, we take the first step towards understanding and mitigating such a threat. Specifically, we model the maintenance task assignments and their corresponding security impact due to privilege escalation. We then mitigate such impact through optimizing the task assignments with respect to given constraints. The simulation results demonstrate the effectiveness of our solution in various situations.

1 Introduction

The widespread adoption of cloud leads to many unique challenges in terms of security and privacy [13]. As the cloud service market becomes more and more competitive, cloud providers are striving to attract customers with better services and less downtime at a lower cost. The search for an advantage in cost and efficiency will inevitably lead cloud providers to follow a similar path as what has been taken by their tenants, i.e., outsourcing cloud maintenance tasks to remote administrators including those from specialized third party maintenance providers [9]. Such an approach may also lead to many benefits due to resource sharing, e.g., the access to specialized and experienced domain experts, the flexibility (e.g., less need for full-time onsite staff), and the lower cost (due to the fact such remote administrators are shared among many clients).

However, such benefits come at an apparent cost in terms of increased security threats. Specifically, the remote administrators must be provided with necessary privileges, which may involve direct accesses to the underlying cloud infrastructure, in order to complete their assigned maintenance tasks. Armed with such privileges, a dishonest remote administrator, or an attacker with the stolen credentials of an administrator, can pose severe insider threats to both the cloud tenants (e.g., causing a large scale leak of

confidential user data) and the provider (e.g., disrupting the cloud services or abusing the cloud infrastructure for illegal activities) [12]. On the other hand, cloud providers are under the obligation to prevent such security or privacy breaches caused by insiders [14], either as part of the service level agreements, or to ensure compliance with security standards (e.g., ISO 27017 [19]). Therefore, there is a pressing need to better understand and mitigate such insider threats.

Dealing with the insider threat of remote administrators in clouds faces unique challenges. First, there is a lack of public access to the detailed information regarding cloud infrastructure configurations and typical maintenance tasks performed in clouds. Evidently, most existing works on insider attacks in clouds either stay at a high level or focus on individual nodes instead of the infrastructure [9, 20, 32] (a more detailed review of related work will be given in Section 6). Second, cloud infrastructures can be quite different from typical enterprise networks in terms of many aspects of security. For instance, multi-tenancy means there may co-exist different types of insiders with different privileges, such as administrators of a cloud tenant, those of the cloud provider, and third party remote administrators. Also, virtualization means a more complex attack surface consisting of not only physical nodes but also virtual or hypervisor layers. To the best of our knowledge, there is a lack of any concrete study in the literature on the insider attack of remote administrators in cloud data centers.

In this paper, we take the first step towards understanding and mitigating such insider threats. Specifically, we first model the maintenance tasks and their corresponding privileges. We then model the insider threats posed by remote administrators assigned to maintenance tasks by applying the existing k-zero day safety metric as follows; remote administrators possess elevated privileges due to the assigned maintenance tasks, and those privileges correspond to initially satisfied security conditions, which are normally only accessible by external attackers after exploiting certain vulnerabilities. Such model allows us to formulate the mitigation of the insider threats of remote administrators as an optimization problem and solve it using standard optimization techniques. We evaluate our approach through simulations and the results demonstrate the effectiveness of our solution under various situations. In summary, the main contribution of this paper is twofold:

- To the best of our knowledge, this is the first study on the insider threat of remote administrators in cloud infrastructures. As cloud providers leverage third parties for better efficiency and cost saving, our study demonstrates the need to also consider the security impact, and our model provides a way for quantitatively reasoning about the tradeoff between such security impact with other related factors.
- By formulating the optimization problem of mitigating the insider threat of remote administrators through optimal task assignments, we provide a relatively effective solution, as evidenced by our simulation results, for achieving the optimal tradeoff between security and other constraints using standard optimization techniques.

The remainder of this paper is organized as follows. Section 2 presents a motivating example and discusses maintenance tasks and privileges. In Section 3, we present our models of task assignment and insider threat. Section 4 formulates the optimization problem and discusses several use cases. Section 5 gives simulation results. Section 6 discusses related work. Section 7 concludes the paper.

2 Preliminaries

This section gives a motivating example and discusses maintenance tasks and privileges.

2.1 Motivating Example

A key challenge to studying security threats in cloud data centers is the lack of public accesses to detailed information regarding hardware and software configurations deployed in real cloud data centers. Existing work mainly focus on either high level frameworks and guidelines for risk and impact assessment [1, 28, 21], or specific vulnerabilities or threats in clouds [15, 30], with a clear gap between the two. To overcome such a limitation, we choose to devise our own fictitious, but realistic cloud data center designs, by piecing together publicly available information gathered from various cloud vendors and providers [5], as shown in Figure 1.

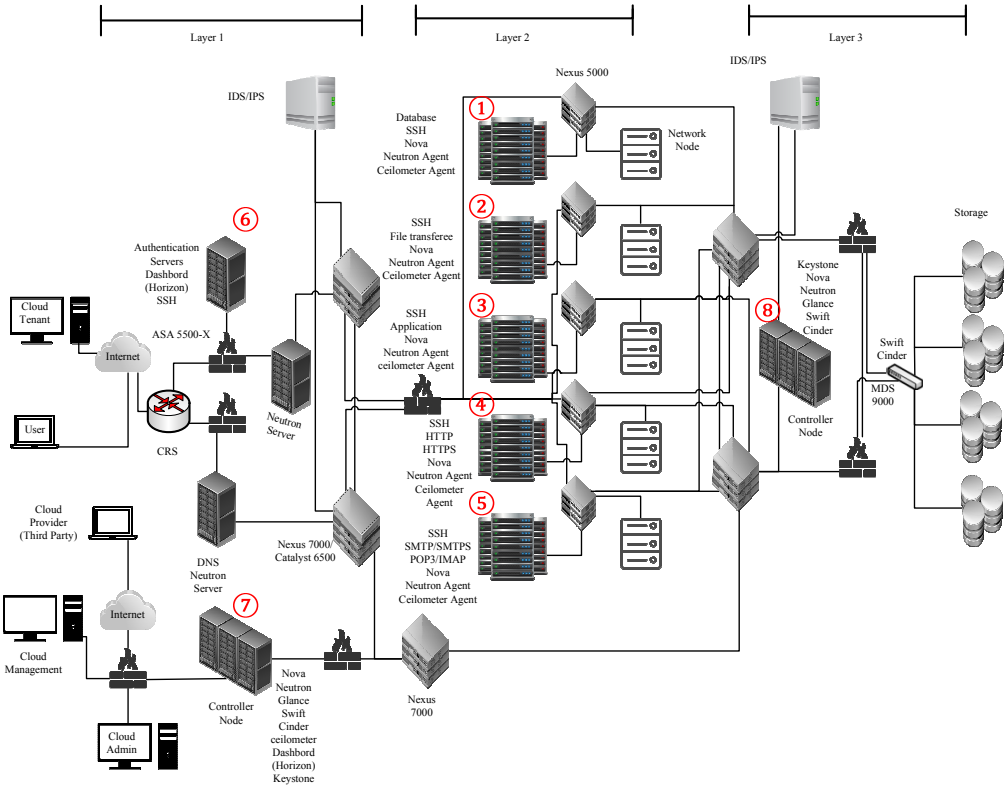

Fig. 1: An example of cloud data center

To make our design more representative, we devise this configuration based on concepts and practices borrowed from major cloud vendors and providers. For example,

we borrow the multi-layer concept and some hardware components, e.g., Carrier Routing System (CRS), Nexus (7000,5000,2000), Catalyst 6500, and MDS 9000, from the cloud data center design of Cisco [7]. We synthesize various concepts of the VMware vSphere [18] for main functionality of hardware components in our cloud infrastructure (e.g., authentication servers, DNS, and SAN). We also assume the cloud employs OpenStack as its operating system [24]. The infrastructure provides accesses to both cloud users and remote administrators through the three layer design. Layer 1 connects the cloud to the internet and includes the authentication servers, DNS, and Neutron Server. Layer 2 includes the rack servers and compute nodes. Layer 3 includes the storage servers. OpenStack components run on the authentication servers, DNS server (a Neutron component provides address translation to machines running the requested services), and compute nodes (Nova to host and manage VMs, Neutron to connect VMs to the network, and Ceilometer to calculate the usage) to provide cloud services.

Such a cloud data center may require many maintenance tasks to be routinely performed to ensure the normal operation of the hardware and software components. Such maintenance tasks may be performed by both internal staff working onsite and remote administrators, including those from specialized third party providers. In our example, assume the cloud provider decides to rely on third party remote administrators for the regular maintenance of the five compute nodes (nodes #1-5 in Figure 1), the authentication servers (node #6), and the two controllers (nodes #7 and 8). Table 1, shows the maintenance tasks need to be performed on those nodes. For simplicity, we only consider three types of tasks here (more discussions about maintenance tasks will be given in next section).

Node Number (in Figure 1)	Maintenance Tasks		
	Read log files	Modify configuration files	Install a new system
1	×	×	
2	×		×
3	×	×	×
4		×	×
5	×		×
6	×	×	
7	×		
8	×		

Table 1: An example of required maintenance tasks

In such a scenario, the cloud provider would naturally raise security concerns due to the fact that necessary privileges must be granted in order to allow the third party remote administrators to perform their assigned maintenance tasks. For instance, the task *read log files* needs certain read privilege to be granted, whereas modifying configuration files and installing a new system would demand much higher levels of privileges. Such privileges may allow a dishonest remote administrator, or attackers with stolen credentials of a remote administrator, to launch an insider attack and cause significant damage to the cloud provider and its tenants. Even though the cloud provider may (to some extent) trust the third party maintenance provider as an organization, it is in its best interest to understand and mitigate such threats from individual administrators.

However, as demonstrated by this example, there are many challenges in modeling and mitigating such insider threats.

- First, as demonstrated in Table 1, there may exist complex relationships between maintenance tasks and corresponding privileges needed to fulfill such tasks, and also relationships between different privileges (e.g., a root privilege implies many other privileges). Those relationships will determine the extent of an insider threat.
- Second, the insider threat will also depend on which nodes in the cloud infrastructure are involved in the assigned tasks, e.g., an insider with privileges on the authentication servers (node #6 in Figure 1) or on the compute nodes (nodes #1-5) may have very different security implications.
- Third, the extent of the threat also depends on the configuration (e.g., the connectivity and firewalls), e.g., an insider having access to the controller node #8 would have a much better chance to compromise the storage servers than one with access to the other controller node #7).
- Finally, while an obvious way to mitigate the insider threat is through assigning less tasks to each remote administrator such as to limit his/her privileges, our study will show that the effectiveness of such an approach depends on other factors and constraints, e.g., the amount of tasks to be assigned, the number of available remote administrators, constraints like each administrator may only be assigned to a limited number of tasks due to availability, or a subset of tasks due to his/her skill set, etc.

Clearly, how to model and mitigate the insider threat may not be straightforward even for such a simplified example (we will give the solution for this example scenario in Section 4.2), and the scenario might become far more complex in practice than the one demonstrated here. The remainder of the paper will tackle those challenges.

2.2 Remote Administrators, Maintenance Tasks, and Privileges

A cloud provider may hire different types of administrators to perform maintenance tasks onsite or through remote accesses [9]. First, *hardware administrators* have physical access to the cloud data center to perform maintenance on the physical components. Second, *security team administrators* are responsible for maintaining the cloud security policies. Third, *remote administrators* (RAs) perform maintenance tasks on certain nodes inside the infrastructure. The first two types can be considered relatively more trustworthy due to their limited quantity and the fact they work onsite, and directly for the cloud provider. The last type is usually considered riskier due to two facts, i.e., they work through remote access which is susceptible to attacks (e.g., via stolen credentials), and they may be subcontracted through third party companies which means less control by the cloud provider. In this paper, we focus on such remote administrators (RAs), even though our models and mitigation solution may equally work for dealing with other types of users if necessary.

There exists only limited public information about the exact maintenance tasks performed at major cloud providers. We have collected such information from various sources, and our findings are summarized on the left-hand side of Table 2, which shows sample maintenance tasks mentioned by Amazon Web Service [2], Google Cloud [3],

and Microsoft Azure [4]. As to privileges required for typical maintenance tasks, Bleikertz et al. provided five sample privileges required for maintaining the compute nodes in clouds [9], which we will borrow for our further discussions, as shown on the right-hand side of Table 2.

Maintenance Task	AWS [2]	GCP [3]	Azure [4]
Review Logs	×	×	×
Hard Disk Scan		×	×
Update Firmware	×	×	×
Patch Operating System	×	×	×
Update Operating System	×	×	×
System Backup	×	×	×
Upgrades System	×	×	×
Maintain Automated Snapshots	×		
Bug Fix	×	×	×
Update Kernel	×	×	

Privilege	Restriction
No privilege	No access
Read	Cannot read VM-related data
Write_L1	The restriction of read privilege applies, software modification restricted to trusted repository
Write_L2	Bootloader, kernel, policy enforcement, maintenance agent, file system snapshots, package manager transaction logs, and certain dangerous system parameters
Write_L3	No restriction

Table 2: Maintenance tasks in popular cloud platforms (left) and the privileges (right)

To simplify our discussions, our running example will be limited to ten maintenance tasks on three compute nodes with corresponding privileges on such nodes, as shown in Table 3. Later in Section 4.2, we will expand the scope to discuss the solution for our motivating example which involves all the eight nodes.

Task Number	Node Number (in Figure 1)	Task Description	Privilege
1	4 ($http$)	Read log files for monitoring	Read
2	4 ($http$)	Modifying configuration files	Write_L1
3	4 ($http$)	Patching system files	Write_L3
4	3 (app)	Read log files for monitoring	Read
5	3 (app)	Modifying configuration files	Write_L1
6	3 (app)	Update kernel	Write_L3
7	1 (DB)	Read log files for monitoring	Read
8	1 (DB)	Modifying configuration files	Write_L1
9	1 (DB)	Update kernel	Write_L3
10	1 (DB)	Install new systems	Write_L2

Table 3: Maintenance tasks and privileges for the running example

3 Models

This section presnts out threat model and models of the maintenance task assignment and insider threat.

3.1 Threat Model and Maintenance Task Assignment Model

Our work is intended to assist the cloud provider in understanding and mitigating the insider threat from dishonest remote administrators or attackers with stolen credentials of a remote administrator. To this end, we assume the majority of remote administrators is trusted, and if there are multiple dishonest administrators (or attackers with their credentials), they do not collude (a straightfoward extension of our models by considering

each possible combination of administrators as one insider can accommodate such colluding administrators, which is considered as future work). We assume the third party provider is trusted as an organization and will collaborate with the cloud provider to implement the intended task assignment. We assume the cloud provider is concerned about certain critical assets inside the cloud, and it is aware of the constraints about task assignments such as the number of remote administrators, their availability and skill set, etc. Finally, as a preventive solution, our mitigation approach is intended as a complementary solution to existing vulnerability scanners, intrusion detection systems, and other solutions for mitigating insider threats.

The cloud provider assigns the maintenance tasks to remote administrators (RAs) based on given constraints (e.g., which tasks may be assigned each RA), and consequently the RA will obtain privileges required by those tasks. This can be modeled as follows (which has a similar syntax as [27]).

Definition 1 (Maintenance Task Assignment Model). *Given*

- *a set of remote administrators RA,*
- *a set of maintenance task T,*
- *a set of privileges P,*
- *the remote administrator task relation $RAT \subseteq RA \times T$ which indicates the maintenance tasks that are allowed to be assigned to each remote administrator, and*
- *the task privilege relation $TP \subseteq T \times P$ which indicates the privileges required for each task,*

a maintenance task assignment is given by function $ta(.) : RA \rightarrow 2^T$ that satisfies $(\forall ra \in RA)(ta(ra) \subseteq \{t \mid (ra, t) \in RAT\}$ (meaning a remote administrator is only assigned with the tasks to which he/she is allowed), and the corresponding set of privileges given to the remote administrator is given by function $pa(ra) = \bigcup_{t \in ta(ra)}\{p \mid (t, p) \in TP\}$.

3.2 Insider Threat Model

We given an overview of our model for the insider threat, which will be demonstrated through an example shown in Figure 2. First, we borrow the resource graph concept [31] to represent the causal relationships between different resources inside the given cloud configuration. Second, we map the privileges given to RAs through maintenance task assignments (Definition 1) to exploits of corresponding resources in the resource graph. Third, we apply the k-zero day safety metric [33] to quantify the insider threat of each RA through his/her k value. Finally, we take the average (and minimum) of all RAs' k values as the average (and worst) case indication of insider threat.

Figure 2 shows an example resource graph for our running example (the dashed lines and shades can be ignored and will be discussed later in Section 4.2; also, only a small portion of the resource graph is shown here due to space limitations). Each triplet inside an oval indicates a potential zero day or known exploit in the format <service or vulnerability, source host, destination host> (e.g. <Xen, RA, 4> indicates an exploit on Xen), and the plaintext pairs indicate the pre- or post-conditions of those exploits in the format <condition, host> where condition can be either a privilege on the host (e.g., <W1,4> means the level 1 write privilege and <R,4> means the read privilege

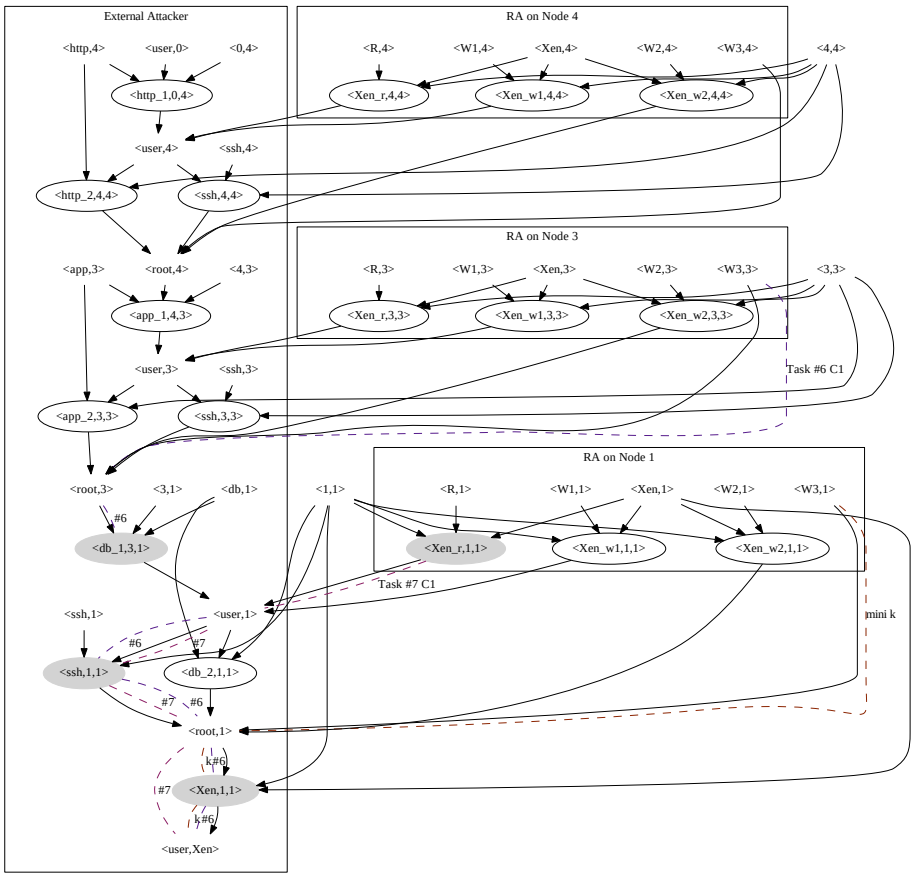

Fig. 2: Modeling insider threat using the resource graph

which are both explained in Section 2.2), the existence of a service on the host (e.g., <Xen,4>), or a connectivity (e.g., <0,4>means attacker can connect to host 4 and <4,4> means a local exploit on host 4). The edges point from pre-conditions to an exploit and then to its post-conditions, which indicate that any exploit can be executed if and only if all of its pre-conditions are satisfied, whereas executing an exploit is enough to satisfy all its post-conditions.

In Figure 2, the left-hand side box indicates the normal resource graph which depicts what an external attacker may do to compromise the critical asset <user, Xen>. The right-hand side boxes depict the insider threats coming from RAs assigned to each of the three compute nodes. The gray color exploits are what captures the consequences of granting privileges to remote administrators. For example, an RA with the level 1 write privilege <W1,4> can potentially exploit Xen (i.e., <Xen_w1,4,4>) to escalate his/her privilege to the user privilege on host 4 (i.e.., <user,4>), whereas a higher level privilege <W2,4> can potentially lead to the root privilege <root,4> through an exploit <Xen_w2,4,4>, and the highest privilege <W3,4> can even directly lead to that priv-

ilege. Those examples show how the model can capture the different levels of insider threats as results of different privileges obtained through maintenance task assignments.

Next, given the maintenance task assignment for each RA, we can obtain all the possible paths he/she may follow in the resource graph, starting from all the initially satisfied conditions (e.g., <Xen,4>) and those implied by the task assignment (e.g., <W1,4>) to the critical asset (i.e., <user,Xen>). To quantify the relative level of such threats, we apply the k-zero day safety metric (k0d) [33] which basically counts the number of zero day exploits (known exploits are not counted, and exploits of the same service are only counted once) along the shortest path. The metric value of each RA provides an estimation for the relative level of threat of each RA, since a larger number of distinct zero day exploits on the shortest path means reaching the critical asset is (exponentially, if those exploits are assumed to be independent) more difficult. For example, an RA with privilege <W3,1> would have a k0d value of 1 since only one zero day exploit <Xen,1,1> is needed to reach the critical asset, whereas an RA with <W2,1> would have a k value of 2 since an additional exploit <Xen_w2,1,1> is needed. Finally, once we have calculated the k values of all RAs based on their given maintenance task assignments, we take the average (and minimum) of those k values as the average (and worst) case indication of the overall insider threat of the given maintenance task assignments. The above discussions are formally defined as follows.

Definition 2 (Insider Threat Model). *Given the maintenance task assignment (i.e., RA, T, P, RAT, TP, ta, and pa, as given in Definition 1) let $C_r = \bigcup_{ra \in RA} pa(ra)$ be the set of privileges implied by the assignment and E_r be the set of new exploits enabled by C_r. Denote by $G(E \cup E_r \cup C \cup C_r, R)$ the resource graph (where E and C denote the original set of exploits and conditions, respectively, and R denote the edges) and let $k0d(.)$ be the k zero day safety metric function. We say $k0d(ra)$, $\frac{\sum_{ra \in RA} k0d(ra)}{|RA|}$, and $min(\{k0d(ra) : ra \in RA\})$ represent the insider threat of ra, the average case insider threat of the maintenance task assignment, and the worst case insider threat of the maintenance task assignment, respectively.*

4 The Mitigation

In this section, we formulate the optimization-based solution for mitigate the insider threat during maintenance task assignment and discuss several use cases.

4.1 Optimization-based Mitigation

Based on our definitions of the maintenance task assignment model and the insider threat model, we can define the problem of optimal task assignment as follows. Note the remote administrator task relation RAT basically gives the constraints for optimization since it states which tasks may be assigned to which RA (in some cases the constraints may also be modeled differently for convenience, e.g., as the maximum number of tasks for each RA).

Definition 3 (The Optimal task assignment problem). *Given a resource graph G, the remote administrators RA, maintenance tasks T, privileges P, the remote administrator task relation RAT, and the task privilege relation TP, find a maintenance task assignment function ta which maximizes the insider threat $\frac{\sum_{ra \in RA} k0d(ra)}{|RA|}$ (or $min(\{k0d(ra) : ra \in RA\})$).*

Theorem 1. *The Optimal task assignment problem (Definition 3) is NP-hard.*

Proof: First, calculating the $k0d$ function is already NP-hard w.r.t. the size of the resource graph [33]. On the other hand, we provide a sketch of a proof to show the problem is also NP-hard from the perspective of the maintenance task assignment. Specifically, given any instance of the well known NP-complete problem, exact cover by 3-sets (i.e., given a finite set X containing exactly $3n$ elements, and a collection C of subsets of X each of which contains exactly 3 elements, determine whether there exists $D \subseteq C$ such that every $x \in X$ occurs in exactly one $d \in D$), we can construct an instance of our problem as follows. We use X for the set of maintenance tasks, and C for the set of RAs, such that the three elements of each $c \in C$ represent three tasks which can be assigned to c. In addition, no RA can be assigned with less than three tasks, and an RA already assigned with three tasks can choose any available task to be assigned in addition. We can then construct a resource graph in which the critical asset can be reached through any combination of four privileges. It then follows that, the insider threat is maximized if and only if there exists an exact cover D due to the following. If the exact cover exists, then every RA $d \in D$ is assigned with exactly three tasks and therefore the k value of every RA, and hence the insider threat, will be equal to infinity since the critical asset cannot be reached with less than four privileges; if the cover does not exist, then to have every task assigned, we will have to assign at least one RA with more than three tasks, and hence the k value will decrease. □

In our study, we use the genetic algorithm to optimize the maintenance task assignments by maximizing k. Specifically, the resource graph is taken as input to the optimization algorithm, with the (either average case or worst case) insider threat value k as the fitness function. We try to find the best task assignment for maximizing the value k within a reasonable number of generations. The constraints can be given either through defining the remote administrator task relation RAT in the case of specific tasks that can be assigned to each RA, or as a fixed number of tasks for each RA. Other constraints can also be easily applied to the optimization algorithm. In our simulations, we choose the probability of 0.8 for crossover and 0.2 for mutation based on our experiences.

4.2 Use Cases

We demonstrate our solution through several use cases with different constraints. The first three use cases are based on the five remote administrators and ten maintenance tasks presented in Table 3 and the last use case is based on the motivating example shown in Section 2.1.

– Use Case A: In this case, each RA should be assigned with two tasks. The three tables shown in Table 4 show three possible assignments and the corresponding k values. Also, Figure 2 shows an example path (dashed lines) for tasks assigned to

User	A_1	B_1	C_1	D_1	E_1
Tasks Number	4 1	5 10	6 7	8 3	9 2
k	3	1	2	2	1
$\bar{k}$			1.8		
Minimum k			1		

User	A_2	B_2	C_2	D_2	E_2
Tasks Number	6 9	4 3	7 10	8 1	5 2
k	1	3	1	2	3
$\bar{k}$			2		
Minimum k			1		

User	A_3	B_3	C_3	D_3	E_3
Tasks Number	4 1	5 2	6 7	8 3	9 10
k	3	3	2	2	1
$\bar{k}$			2.2		
Minimum k			1		

Table 4: Maintenance tasks assignments for use case A

RA C_1 based on the top table, and also the shortest path yielding the minimum k value. We use the GA to find the optimal task assignment that meets the constraint given in this case, as shown in the last table, the maximal average of k values among all RAs is $\bar{k} = 2.2$. It can also be seen that the minimum k value among all RAs is always $k = 1$ in this special case.

- Use Case B: In this case, each RA should be assigned with at least one task. The optimal task assignment under this constraint is (RA1$\{8,9,10\}$, RA2$\{4,5\}$, RA3$\{3\}$, RA4$\{1,2\}$, and RA5 $\{6,7\}$). This relaxed constraint improves the average of k from 2.2 in the previous example to 2.8, which shows relaxing the constraint may increase k (which means less threat).
- Use Case C: In this case, each RA can handle a fixed subset of tasks. In our example, we assume RA1 can be assigned to any task requiring the read privilege, RA2 to tasks requiring write level 1 privilege, RA3 to tasks requiring write level 1 and 2, RA4 to tasks requiring write level 3, and RA5 can be assigned to any task. After applying our solution, the optimal assignment yields the maximal average of k values to be $k = 2.2$.
- Use Case D: This case shows the optimal maintenance task assignment for tasks discussed in our motivating example in Section 2.1. We have eight RAs and each RA can handle maximum two tasks. The upper table in Table 5 shows the 15 maintenance tasks to be assigned. In Table 5, the four tables on the bottom show four different tasks scenarios assigned to RAs and each table shows different average k. The bottom table on the right side shows the optimal task assignment in term of the average $k = 3.125$.

5 Simulations

This section shows simulation results on applying our mitigation solution under various constraints. All simulations are performed using a virtual machine equipped with a 3.4 GHz CPU and 4GB RAM in the Python 2.7.10 environment under Ubuntu 12.04 LTS and the MATLAB R2017bs GA toolbox. To generate a large number of resource graphs for simulations, we start with seed graphs with realistic configurations similar to Figure 1 and then generate random resource graphs by injecting new nodes and edges into those seed graphs. Those resource graphs were used as the input to the optimization toolbox where the fitness function is to maximize the average or worst case insider threat value k (given in Definition 2) with various constraints, e.g., the number of available RAs and maintenance tasks and how many task may be assigned to each RA. We repeat each simulation on 300 different resource graphs to obtain the average result.

Task#	Maintenance task	Task#	Maintenance task
1	Read log files for node 1	2	Modify configuration file for node 1
3	Read log files for node 2	4	Install a new system for node 2
5	Read log files for node 3	6	Modify configuration file for node 3
7	Install a new system for node 3	8	Modify configuration file for node 4
9	Install a new system for node 4	10	Read log files for node 5
11	Install a new system for node 5	12	Read log files for node 6
13	Modify configuration file for node 6	14	Read log files for node 7
15	Read log files for node 8		

User	RA1	RA2	RA3	RA4	RA5	RA6	RA7	RA8
Tasks Number	14 5	1 9	4 15	8 12	2 10	3 11	7 13	6
k	1	3	2	3	2	3	2	3
$\bar{k}$	2.375							
Minimum k	1							

User	RA1	RA2	RA3	RA4	RA5	RA6	RA7	RA8
Tasks Number	1 9	2 10	3 11	4 12	5 13	6 14	7 15	8
k	3	2	3	3	3	1	2	5
$\bar{k}$	2.75							
Minimum k	1							

User	RA1	RA2	RA3	RA4	RA5	RA6	RA7	RA8
Tasks Number	1 4	2 7	3 9	5 10	6 11	15 12	13 14	8
k	3	2	4	4	3	2	1	5
$\bar{k}$	3							
Minimum k	1							

User	RA1	RA2	RA3	RA4	RA5	RA6	RA7	RA8
Tasks Number	1 12	2 7	3 9	5 10	6 11	14 15	4 13	8
k	3	2	4	4	3	1	3	5
$\bar{k}$	3.125							
Minimum k	1							

Table 5: Maintenance task assignments for use case D (the motivating example)

The objective of the first two simulations is to study how the average case insider threat (i.e., the average of k values among all RAs) may be improved through our mitigation solution under constraints on the number of tasks and RAs, respectively. In Figure 3, the number of available RAs is fixed at 500, while the number of maintenance tasks is varied between 500 and 2,000 along the X-axis. The Y-axis shows the average of k among all RAs. The solid lines represent the results after applying our mitigation solution under constraints about the maximum number of tasks assigned to each RA. The dashed lines represent the results before applying the mitigation solution.

Results and Implications: From the result, we can make the following observations. First, the mitigation solution successfully reduces the insider threat (increasing the average of k values) in all cases. Second, the results before and after applying the solution decrease (meaning increased insider threat) following similar linear trends, as the number of maintenance tasks increases until each RA reaches its full capacity. Finally, the result of maximum four tasks per RA after applying the solution is close to the result of maximum ten tasks per RA before applying the solution, which means the mitigation solution may allow more (more than double) tasks to be assigned to the same number of RAs while yielding the same level of insider threat.

In Figure 4, the number of maintenance tasks is fixed at 2,500 while the number of RAs is varied between 400 and 1,000 along the X-axis. The Y-axis shows the average of k among all RAs. The solid lines represent the results after applying the mitigation solution and the dashed lines for the results before applying the solution. All the lines start with sufficient numbers of RAs for handling all the tasks since we only consider one round of assignment. We apply the same constraint as in previous simulation.

Results and Implications: Again we can see the mitigation solution successfully reduces the insider threat (increasing the average of k values) in all cases. More interestingly, we can observe the trend of the lines as follows. The dashed lines all follow a similar near linear trend, which is expected since a larger number of RAs means less insider threat since each RA will be assigned less tasks and hence given less privileges.

On the other hand, most of the solid lines follow a similar trend of starting flat then increasing almost linearly before reaching the plateau. This trend indicates that, the mitigation solution can significantly reduce the insider threat when the number of RAs is within certain ranges past which it becomes less effective (because each RA already receives minimum privileges). The trend of 4 tasks per RA is slightly different mostly due to the limited number of RAs.

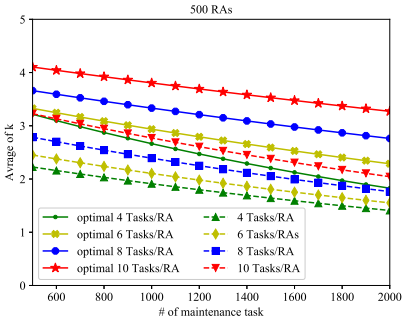

Fig. 3: Average of k among 500 RAs before and after applying the mitigation solution

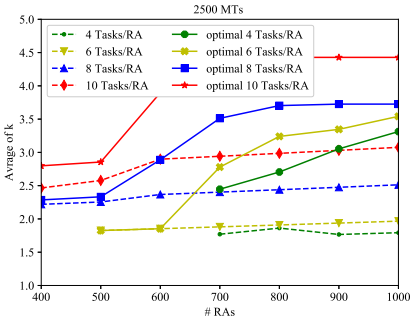

Fig. 4: Average of k among different number of RAs before and after the solution

The objective of the next two simulations is to study how the worst case insider threat (i.e., the minimum k values among all RAs) behaves under the mitigation solution. Figure 5 and Figure 6 are based on similar X-axis and constraints as previous two simulations, whereas the Y-axis shows the minimum k among all RAs (averaged over 300 simulations).

Results and Implications: In Figure 5, we can see that the minimum k values also decrease (meaning more insider threat) almost linearly as the number of tasks increases. In contrast to previous simulation, we can see the minimum k values are always lower than the average k values, which is expected. In Figure 6, we can see the minimum k values also increase almost linearly before reaching the plateau as the number of RAs increases. In contrast to previous simulation, we can see the increase here is slower, which means the worst case results (minimum k values) are more difficult to improve with a increased number of RAs. Also, we can see that the worst case results reach the plateau later (e.g., 900 RAs for 8 tasks per RA) than the average case results (700 RAs).

6 Related Work

The insider threat is a challenging issue for both traditional networks and clouds. Ray and Poolsapassit proposed an alarm system to monitor the behavior of malicious insiders using the attack tree [25]. Mathew et al. used the capability acquisition graphs (CAG) to monitor the abuse of privileges by malicious insiders [23]. Sarkar et al. proposed DASAI to analyze if a process contains a step that meet the insider attack condi-

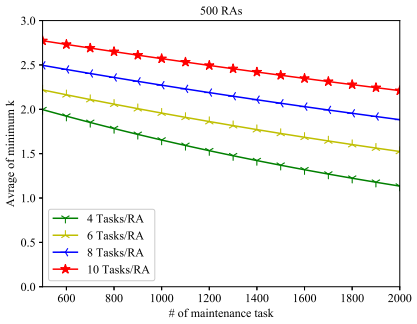

Fig. 5: Minimum k for 500 RAs

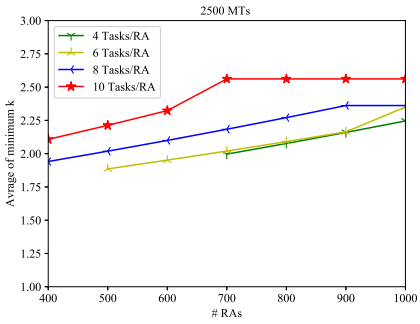

Fig. 6: Minimum k for varying # of RAs

tion [29]. Chinchani et al. proposed a graph-based model for insider attacks and measure the threat [11]. Althebyan and Panda proposed predication and detection model for insider attacks based on knowledge gathered by the internal users during work time in the organization [6]. Bishop et al. presented insider threat definition based on security policies and determine source of risk [8]. Roy et al. studied an employee assignment problem to find an optimal tasks assigned to the employee based on constraints in role-based access control [26].

There is lack of work focusing on the cloud security metrics in general and for insider attacks especially. Our previous work focus on applying threat modeling to cloud data center infrastructures with a focus on external attackers [5]. Gruschka and Jensen devise a high level attack surface framework to show from where the attack can start [16]. The NIST emphasizes the importance of security measuring and metrics for cloud providers in [1]. A framework is propose by Luna et al. for cloud security metrics using basic building blocks [22].

Besides threat modeling, mitigating insider attackers in clouds is also a challenging task. There are many works discuss securing the cloud from insider attack by limiting the trust on the compute node [32]. Li et al. focuses on supporting users to configure privacy protection in compute node [20]. Closest to our work, Bleikertz et al. focus on securing the cloud during maintenance time by limiting the privilege grant to the remote administrator based on the tasks assigned to that administrator [9]. We borrow their categorization of the privileges. Our mitigation approach is also inspired by the network hardening approaches using genetic algorithms [17, 10].

7 Conclusion

In this paper, we have modeled the insider threat during maintenance task assignment for cloud providers to better understand such threat posed by third party remote administrators, and we have formulated the optimal assignment problem as an optimization problem and applied standard optimization algorithm to derive a solution under different constraints. We have also conducted simulations whose results show our solution can significantly reduce the insider threat of remote administrators. Our future work will focus on following directions. First, we will improve our solution to handle more

realistic scenarios, e.g., incremental assignment for streams of new maintenance tasks, and handling dynamics (joining or leaving) of RAs, giving priority or weight to tasks. Second, we will consider explicit cost models for assignments and incorporate the cost into the mitigation solution, e.g., based on the number of RAs, the amount or duration of tasks, and privileges needed.

Acknowledgements. The authors thank the anonymous reviewers for their valuable comments. This work was partially supported by the National Institutes of Standard and Technology under grant number 60NANB16D287, by the National Science Foundation under grant number IIP-1266147, and by Natural Sciences and Engineering Research Council of Canada under Discovery Grant N01035.

References

1. National Institute of Standards and Technology: Cloud Computing Service Metrics Description. `http://www.nist.gov/itl/cloud/upload/RATAX-CloudServiceMetricsDescription-DRAFT-20141111.pdf`, 2015. [Online; accessed 17/06/2015].
2. Amazon Web Services. `https://aws.amazon.com/`, 2018. [Online; accessed 28/02/2018].
3. Google Cloud Platform. `https://cloud.google.com/`, 2018. [Online; accessed 28/02/2018].
4. Microsoft Azure. `https://azure.microsoft.com`, 2018. [Online; accessed 28/02/2018].
5. N. Alhebaishi, L. Wang, S. Jajodia, and A. Singhal. Threat modeling for cloud data center infrastructures. In *Foundations and Practice of Security - 9th International Symposium, FPS 2016, Québec City, QC, Canada, October 24-25, 2016, Revised Selected Papers*, pages 302–319, 2016.
6. Q. Althebyan and B. Panda. A knowledge-base model for insider threat prediction. In *2007 IEEE SMC Information Assurance and Security Workshop*, pages 239–246, June 2007.
7. K. Bakshi. Cisco cloud computing-data center strategy, architecture, and solutions. *DOI= http://www. cisco. com/web/strategy/docs/gov/CiscoCloudComputing_WP. pdf*, 2009.
8. M. Bishop, S. Engle, S. Peisert, S. Whalen, and C. Gates. We have met the enemy and he is us. In *Proceedings of the 2008 New Security Paradigms Workshop*, NSPW '08, pages 1–12, New York, NY, USA, 2008. ACM.
9. S. Bleikertz, A. Kurmus, Z. A. Nagy, and M. Schunter. Secure cloud maintenance: Protecting workloads against insider attacks. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '12, pages 83–84, New York, NY, USA, 2012. ACM.
10. D. Borbor, L. Wang, S. Jajodia, and A. Singhal. Diversifying network services under cost constraints for better resilience against unknown attacks. In *Data and Applications Security and Privacy XXX - 30th Annual IFIP WG 11.3 Conference, DBSec 2016, Trento, Italy, July 18-20, 2016. Proceedings*, pages 295–312, 2016.
11. R. Chinchani, A. Iyer, H. Q. Ngo, and S. Upadhyaya. Towards a theory of insider threat assessment. In *2005 International Conference on Dependable Systems and Networks (DSN'05)*, pages 108–117, June 2005.
12. W. R. Claycomb and A. Nicoll. Insider threats to cloud computing: Directions for new research challenges. In *2012 IEEE 36th Annual Computer Software and Applications Conference*, pages 387–394, July 2012.

13. Cloud Security Alliance. Security guidance for critical areas of focus in cloud computing v 3.0, 2011.
14. Cloud Security Alliance. Top threats to cloud computing, 2018. Available at: https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf.
15. K. Dahbur, B. Mohammad, and A. B. Tarakji. A survey of risks, threats and vulnerabilities in cloud computing. In *Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications*, ISWSA '11, pages 12:1–12:6, New York, NY, USA, 2011. ACM.
16. N. Gruschka and M. Jensen. Attack surfaces: A taxonomy for attacks on cloud services. In *2010 IEEE 3rd international conference on cloud computing*, pages 276–279. IEEE, 2010.
17. M. Gupta, J. Rees, A. Chaturvedi, and J. Chi. Matching information security vulnerabilities to organizational security profiles: a genetic algorithm approach. *Decision Support Systems*, 41(3):592 – 603, 2006. Intelligence and security informatics.
18. M. Hany. VMware VSphere In The Enterprise. http://www.hypervizor.com/diags/HyperViZor-Diags-VMW-vS4-Enterprise-v1-0.pdf. [Online; accessed 05/02/2015].
19. ISO Std IEC. ISO 27017. *Information technology- Security techniques- Code of practice for information security controls based on ISO/IEC 27002 for cloud services (DRAFT),* http://www.iso27001security.com/html/27017.html, 2012.
20. M. Li, W. Zang, K. Bai, M. Yu, and P. Liu. Mycloud: Supporting user-configured privacy protection in cloud computing. In *Proceedings of the 29th Annual Computer Security Applications Conference*, ACSAC '13, pages 59–68, New York, NY, USA, 2013. ACM.
21. J. Luna, H. Ghani, D. Germanus, and N. Suri. A security metrics framework for the cloud. In *Security and Cryptography (SECRYPT), 2011 Proceedings of the International Conference on*, pages 245–250, July 2011.
22. J. Luna, H. Ghani, D. Germanus, and N. Suri. A security metrics framework for the cloud. In *Security and Cryptography (SECRYPT), 2011 Proceedings of the International Conference on*, pages 245–250. IEEE, 2011.
23. S. Mathew, S. Upadhyaya, D. Ha, and H. Q. Ngo. Insider abuse comprehension through capability acquisition graphs. In *2008 11th International Conference on Information Fusion*, pages 1–8, June 2008.
24. Openstack. Openstack Operations Guide. http://docs.openstack.org/openstack-ops/content/openstack-ops_preface.html. [Online; accessed 27/08/2015].
25. I. Ray and N. Poolsapassit. *Computer Security – ESORICS 2005: 10th European Symposium on Research in Computer Security, Milan, Italy, September 12-14, 2005. Proceedings*, chapter Using Attack Trees to Identify Malicious Attacks from Authorized Insiders, pages 231–246. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
26. A. Roy, S. Sural, A. K. Majumdar, J. Vaidya, and V. Atluri. On optimal employee assignment in constrained role-based access control systems. *ACM Trans. Manage. Inf. Syst.*, 7(4):10:1–10:24, Dec. 2016.
27. R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *Computer*, 29(2):38–47, Feb. 1996.
28. P. Saripalli and B. Walters. Quirc: A quantitative impact and risk assessment framework for cloud security. In *2010 IEEE 3rd International Conference on Cloud Computing*, pages 280–288, July 2010.
29. A. Sarkar, S. Khler, S. Riddle, B. Ludaescher, and M. Bishop. Insider attack identification and prevention using a declarative approach. In *2014 IEEE Security and Privacy Workshops*, pages 265–276, May 2014.

30. F. B. Shaikh and S. Haider. Security threats in cloud computing. In *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*, pages 214–219, Dec 2011.
31. O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing. Automated generation and analysis of attack graphs. In *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*, pages 273–284, 2002.
32. W. K. Sze, A. Srivastava, and R. Sekar. Hardening openstack cloud platforms against compute node compromises. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '16, pages 341–352, New York, NY, USA, 2016. ACM.
33. L. Wang, S. Jajodia, A. Singhal, P. Cheng, and S. Noel. k-zero day safety: A network security metric for measuring the risk of unknown vulnerabilities. *IEEE Transactions on Dependable and Secure Computing*, 11(1):30–44, Jan 2014.