Mitigating the Insider Threat of Remote Administrators in Clouds through Maintenance Task Assignments

Nawaf Alhebaishi [a,b,*], Lingyu Wang [a], Sushil Jajodia [c] and Anoop Singhal [d]
[a] *Concordia Institute for Information Systems Engineering, Concordia University, Quebec, Canada E-mails:{n_alheb,wang}@ciise.concordia.ca*
[b] *Faculty of Computing and Information Technology, King Abdulaziz University, Jeddag, KSA*
[c] *Center for Secure Information Systems, George Mason University, VA, USA E-mail: jajodia@gmu.edu*
[d] *Computer Security Division, National Institute of Standards and Technology, MD, USA Email: anoop.singhal@nist.gov*

Abstract. Today's cloud providers strive to attract customers with better services and less downtime in a highly competitive market. The need for minimizing the operational cost unavoidably leads cloud providers to rely on third party remote administrators for fulfilling regular maintenance tasks. In such a scenario, the lack of trust in those third party remote administrators paired with the extra privileges granted to them to complete the maintenance tasks usually implies undesirable security threats. A dishonest remote administrator, or an attacker armed with the stolen credential of a remote administrator, can pose severe insider threats to both the cloud provider and its tenants. In this paper, we take the first step towards understanding and mitigating such insider threats of remote administrators in clouds. Specifically, we first 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. Finally, the simulation results demonstrate the effectiveness of our solution in various scenarios.

Keywords: Security Metric, Cloud Security, Insider Attack, Attack Graph, Service Dependency Graph

1. Introduction

Cloud computing has become the cost-saving IT solution for 73% of organizations worldwide [1] and is predicted to grow to a $300 billion business by 2021 [2]. Cloud computing is also affecting our daily lives through its impact on politics (e.g., politicians are increasingly turning to social networks, which are mostly cloud-based), education (e.g., Massive Open Online Course (MOOC) is mostly delivered via cloud), healthcare, entertainment, etc. The success of cloud computing comes from the many benefits it brings to IT management, e.g., the pervasive access from anywhere with an Internet connection, the flexibility of scaling services up or down to fit changing needs, and the efficiency to deploy applications quickly without worrying about underlying costs or maintenance of the infrastructure.

On the other hand, the widespread adoption of cloud computing also attracts more attention to its unique security and privacy challenges [3, 4]. In particular, as the cloud service market becomes more

[*] Corresponding Author: E-mail: n_alheb@ciise.concordia.ca

and more competitive, cloud providers are striving to attract customers with better services and less downtime at a lower cost. Consequently, 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 [5]. 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 that such remote administrators are usually shared among many clients).

However, the benefits of outsourcing cloud maintenance tasks come at an apparent cost, i.e., the increased insider threats from remote administrators. Specifically, in order to complete their assigned maintenance tasks, the remote administrators must be provided with necessary privileges, which may involve accesses to physical and/or virtual resources of the underlying cloud infrastructure. Armed with such privileges, a dishonest remote administrator, or an attacker with the stolen credentials of such 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) [6]. On the other hand, cloud providers are under the obligation to prevent such security or privacy breaches caused by insiders [7], either as part of the service level agreements, or to ensure compliance with security standards (e.g., ISO 27017 [8]). Therefore, there is a pressing need to better understand and mitigate the insider threats of remote administrators in clouds.

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 [5, 9, 10] (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 the insider threat of remote administrators in clouds. Specifically, we first model the maintenance tasks and their corresponding privileges based on industrial practices from major cloud vendors and providers. We then model the insider threats posed by remote administrators assigned to maintenance tasks by applying existing security metrics; 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. 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 preliminary version of this paper has previously appeared in [11]. In this paper, we have substantially improved and extended the previous version, with the following most significant extensions. First, while the previous version focuses on physical and virtual resources (e.g., physical hosts and virtual machines), we have added Section 3.4 to additionally consider a higher level of abstraction, i.e., services or business functions which may involve multiple physical and virtual resources; we do so by integrating the service dependency graph concept [12, 13] with the existing resource graph in order to model the impact of service dependencies on cloud security during maintenance time. Second, while the previous version only relies on the *k*-zero day safety metric [14, 15], which only considers the shortest attack path and zero day exploits, we have added Section 3.3 to additionally model the impact of all possible attack paths and exploits of known vulnerabilities using the Bayesian network-based security metric. Those extensions in our models correspondingly lead to additional use cases (Section 4.2) and a series of new simulations (Section 5).

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

2. Preliminaries

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

2.1. Motivating Example

The insider threat of remote administrators depends on the underlying cloud infrastructures. Therefore, we will need the detailed configuration of cloud data centers in order to construct a concrete example of such insider threats. A key challenge here is the lack of public accesses to detailed information regarding hardware and software configurations deployed in real cloud data centers. Consequently, most existing works focus on either high level frameworks and guidelines for risk and impact assessment [16–18], or specific vulnerabilities or threats in clouds [19, 20], 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 [21], as shown in Figure 1.

To above configuration is based on existing concepts and common practices borrowed from major cloud vendors and providers to make our design more representative. 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 [22]. We synthesize various concepts of the VMware vSphere [23] 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

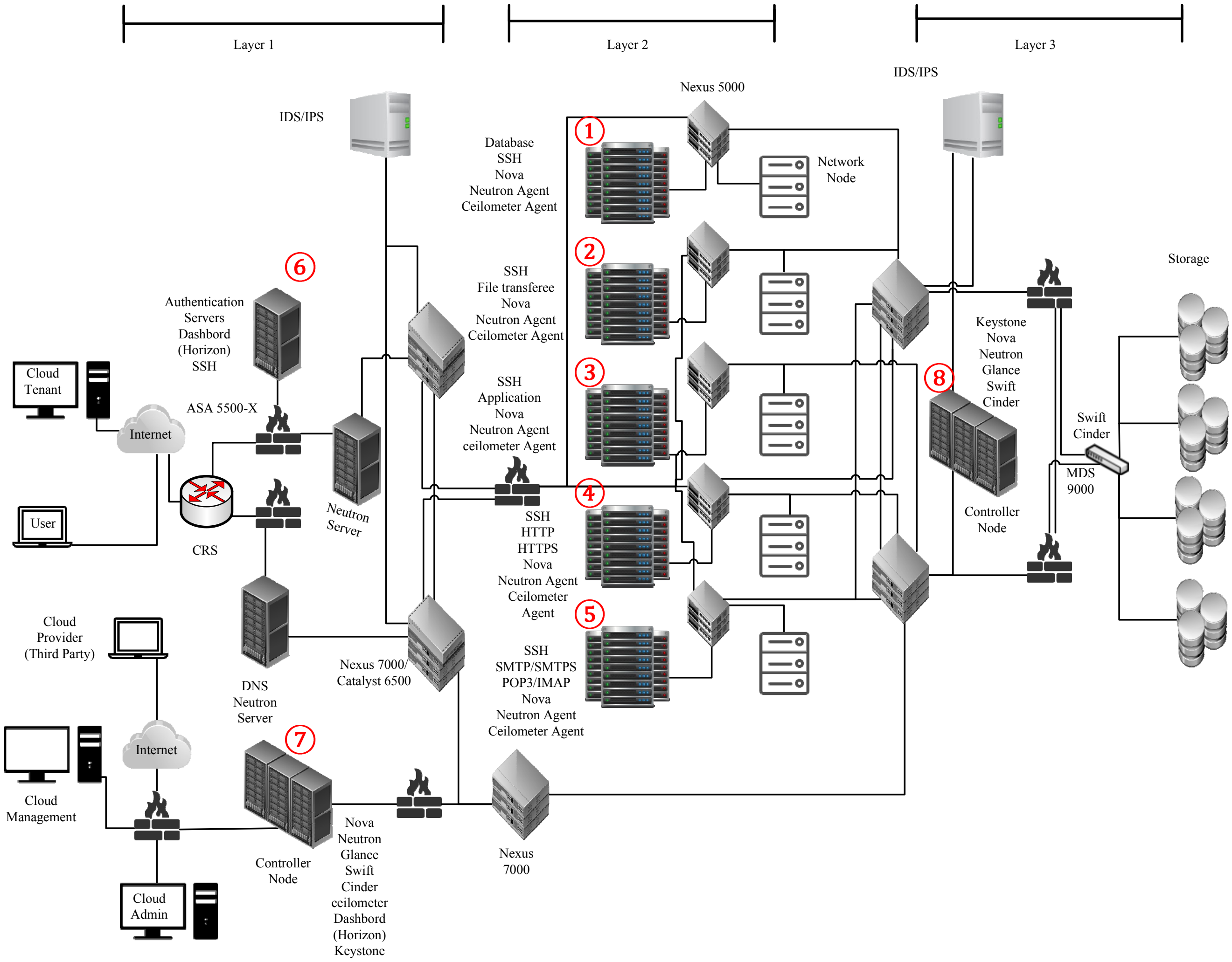

Fig. 1. An example of cloud data center

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 server (node #6), and the two controllers (nodes #7 and #8). As an example, Table 1 shows the maintenance tasks that need to be performed on those nodes. For simplicity, we only consider three

Table 1
An example of required maintenance tasks

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	×		

types of tasks here (more discussions about maintenance tasks will be given in next section).

In such a scenario, the cloud provider naturally faces security challenges due to the fact that necessary privileges must be granted to allow the third party remote administrators to perform their assigned maintenance tasks. For instance, the task of reading 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. Even though the cloud provider may (to some extent) trust the third party maintenance provider as an organization, the granted 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. It is in the cloud provider's best interest to better understand and proactively mitigate such potential threats. However, this toy example is enough to demonstrate that there exist many challenges in modeling and mitigating such threats.

- First, as demonstrated in Table 1, there may exist complex relationships between maintenance tasks and corresponding privileges needed to fulfill such tasks, relationships between different privileges (e.g., a root privilege implies many other privileges), and dependency relationships between services or business functions and the underlying physical and virtual resources used to host such services or functions. 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, as our study will show, the effectiveness of such an approach depends on many 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, modeling and mitigating the insider threat of remote administrators may not be straightforward even for such a simplified example (the solution for this example scenario is given in Section 4.2),

Table 2
Maintenance tasks in popular cloud platforms

Maintenance Task	AWS [25]	GCP [26]	Azure [27]
Review Logs	×	×	×
Hard Disk Scan		×	×
Update Firmware	×	×	×
Patch Operating System	×	×	×
Update Operating System	×	×	×
System Backup	×	×	×
Upgrades System	×	×	×
Maintain Automated Snapshots	×		
Bug Fix	×	×	×
Update Kernel	×	×	

and the scenario is likely far more complex for real clouds than the one demonstrated here. The remainder of the paper will present a systematic approach to tackle those challenges.

2.2. Remote Administrators, Maintenance Tasks, and Privileges

There exist different types of administrators in cloud data centers who perform maintenance tasks either onsite or through remote accesses [5]. For example, *hardware administrators* have physical access to the cloud data center to perform maintenance on the physical components. *Security team administrators* are responsible for maintaining the cloud security policies. *Remote administrators* (RAs) perform maintenance tasks on certain nodes of the infrastructure through network connections from remote sites. The first two types can be considered relatively more trustworthy due to their usually 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 accesses which are 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 the security risk of such remote administrators (RAs), even though our models and mitigation solution may be adapted to deal with other types of administrators and users if necessary.

There exists only limited public information about the exact maintenance tasks performed by major cloud providers. We have collected such information from various sources, and our findings are summarized in Table 2, which shows sample maintenance tasks mentioned by Amazon Web Service [25], Google Cloud [26], and Microsoft Azure [27]. As to privileges required for typical maintenance tasks, Bleikertz et al. provided five sample privileges required for maintaining the compute nodes in clouds [5], which we will borrow for our further discussions, as shown in Table 3.

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 4. Later in Section 4.2, we will expand the scope to discuss the solution for our motivating example which involves all the eight nodes.

3. Models

This section presents out threat model and the proposed models of the maintenance task assignment and insider threats.

Table 3

Privileges used in this work

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 4

Maintenance tasks and privileges for the running example

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

3.1. *The Threat Model and Maintenance Task Assignment Model*

The in-scope threats we consider include insider attacks from dishonest remote administrators or attackers with stolen credentials of such administrators. Consequently, 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 straightforward extension of our models by considering each possible combination of administrators as one insider can accommodate such colluding administrators). The third party provider is considered trusted as an organization and it will collaborate with the cloud provider to implement the intended task assignment. The cloud provider is concerned about certain critical assets, such as physical or virtual resources and services or business functions, 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 prevention or mitigation solutions.

The cloud provider assigns the maintenance tasks to remote administrators (RAs) based on given constraints (e.g., which tasks may be assigned to 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 [28]).

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. The Insider Threat Model

To model various resources and their relationships in cloud infrastructures, we borrow the resource graph concept [29, 30] to represent hardware hosts (e.g., servers and networking devices), software resources (e.g., network services and applications) running on such hosts (only remotely accessible resources are considered), and the causal relationships between different resources (e.g., a zero day exploit on the Web server may lead to user privilege on that server which subsequently causes the application server to be accessible). This concept is more formally stated in Definition 2 and will be illustrated through an example.

Definition 2 (Resource Graph [29, 30])**.** *Given a network with the set of hosts H, the set of resources R, with the resource mapping $res(.) : H \rightarrow 2^R$, the set of zero day exploits $E = \{\langle r, h_s, h_d \rangle \mid h_s \in H, h_d \in H, r \in res(h_d)\}$ and their pre- and post-conditions C, a resource graph is a directed graph $G(E \cup C, R_r \cup R_i)$ where $R_r \subseteq C \times E$ and $R_i \subseteq E \times C$ are the pre- and post-condition relations, respectively.*

To quantify the insider threat of remote administrators based on resource graphs, we extend the *k*-zero day safety security metric [14, 15]. Roughly speaking, the metric starts with the worst case assumption that the relative severity of unknown (zero day) vulnerabilities are not measurable; it then simply counts how many different resources must be compromised through such unknown vulnerabilities in order to compromise a given critical asset; a larger count will indicate a relatively more secure network, since the likelihood of having more unknown vulnerabilities all available at the same time, inside the same network, and exploitable by the same attacker, would be significantly lower. The following provides a simplified version of this concept, which will be illustrated through an example.

Definition 3 (Attack Path and *k*-Zero Day Safety [14, 15])**.** *Given a resource graph $G(E \cup C, R_r \cup R_i)$, we call $C_I = \{c : c \in C, (\nexists e \in E)(\langle e, c \rangle \in R_i)\}$ the set of initial conditions; we call any sequence of zero day exploits $e_1, e_2, \ldots, e_n$ an attack path if all the pre-conditions of each e_i are either initial conditions, or post-conditions of some $e_j (j < i)$. For any given critical asset $c \in C$, we say the network is k-zero day safe if there does not exist any attack path which involves k or less distinct resources, and includes at least one exploit having c as its post-condition.*

Figure 2 shows an example resource graph for our running example (the dashed lines and shades represent our extension to the model, which can be ignored for now 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 triple inside an oval indicates a potential zero day or known exploit in the format <service or

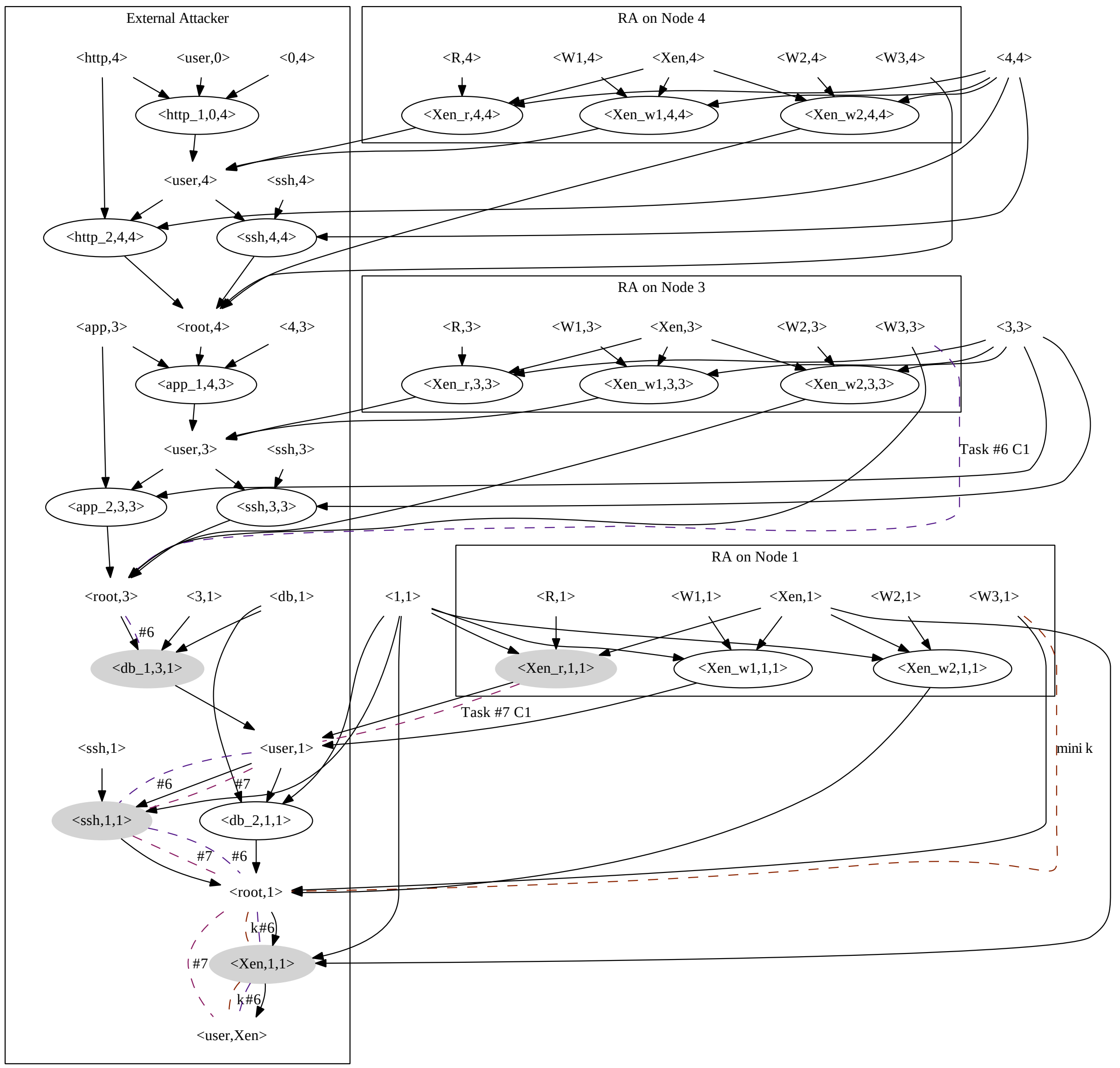

Fig. 2. Modeling insider threat using the resource graph

vulnerability, source host, destination host> (e.g. <Xen, RA, 4> indicates an exploit of Xen on host 4 from host RA), and the plaintext pairs indicate the pre- or post-conditions of those exploits in the format <condition, host> where a 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 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 privilege. 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 (*k*0d) mentioned above. 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 *k*0d 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 4 (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 $k0d(ra)$, $\frac{\Sigma_{ra \in RA} k0d(ra)}{|RA|}$, 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.*

3.3. The Bayesian Network Model

The previous section has applied the *k*-zero day safety metric to model the insider threat of remote administrators. This is a conservative model since the *k* value is defined based on the shortest attack paths, which attacker may or may not be able to follow in practice. Moreover, the model only considers zero day exploits and known vulnerabilities do not contribute to the *k* value. In this section, we extend this model by applying the Bayesian network (BN)-based metric [31] instead of the *k*-zero day safety.

The BN-based metric is based on the conditional probability of reaching the given critical assets given that all initial conditions are satisfied. We first construct a Bayesian network based on the resource graph and the conditional probability that each exploit can be executed given its pre-conditions are all satisfied. Such conditional probabilities can be assigned to both known vulnerabilities based on standard vulnerability scores (e.g., the CVSS scores [32]), and zero day exploits based on a nominal value (e.g., 0.08 [33]). Therefore, the model captures both zero day and known vulnerabilities, and it also takes all attack paths into consideration. Finally, the model can also capture additional casual dependencies, e.g., the same vulnerability appearing on multiple hosts may yield a higher probability (e.g., 0.9 in our examples).

By applying the BN-based metric to the resource graph given in Definition 4, we can obtain the probability for each RA to compromise the given critical assets given all the privileges implied by a maintenance task that is assigned to the RA. Since an RA may be assigned to multiple maintenance tasks, the RA can compromise the critical assets as long as at least one of the assigned tasks enables him/her to do so whose probability can be computed as in Equation 1. We redefine the insider threat model based on those discussions in Definition 5. The model also allows assigning the relative likelihood of each RA to be misbehaving, which can be estimated either based on the background (e.g., third party RAs should be assigned a higher probability than RAs of the cloud provider) or behavior-based detection results if available.

Definition 5 (The BN-based Insider Threat Model)**.** *Given the maintenance task assignment (i.e., RA, T, P, RAT, TP, ta, and pa), 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 $BN = (G, \theta)$ be a Bayesian network where θ denotes the BN parameters. Let $P_{BN}(t)$ be the conditional probability that an RA assigned with task t can compromise the given critical assets, and $P_M(ra)$ the given probability that ra will misbehave. We say $P(ra)$, $\frac{\sum_{ra \in RA} P(ra)}{|RA|}$, and $min(\{P(ra) : ra \in RA\})$ 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, where*

$$P(ra) = 1 - [\prod_{t \in T, (ra,t) \in RAT} (1 - P_{BN}(t) \cdot P_M(ra))] \tag{1}$$

3.4. The Service Dependency Model

The insider threat models introduced in previous sections are based on resource graphs, which are mainly designed to model hardware and software resources. The resource graphs, however, do not directly indicate any higher level services or business functions, the relationships between such services or functions, or their dependencies on the underlying hardware and software resources. For this purpose, the concept of service dependency graph [12, 13] has been proposed to model security impact on services [34]. For example, Figure 3 demonstrates an example in which the lower figure shows an attack graph (which is syntactically equivalent to a resource graph but designed for exploits of known vulnerabilities) depicting various exploits and their relationships, and the upper figure shows the service dependency graph depicting various services; the dashed line edges show the dependencies between the services and corresponding resources involved in the vulnerabilities. The service dependency graph and the attack graph can be integrated and flattened as an extended model. This model can be used to identify attack paths exploiting services or leading to critical assets given as services.

We apply the service dependency model [12] to extend our insider threat models introduced in previous sections. For example, in Figure 4, the left side of Figure 4 shows examples of service dependency-related exploits which are integrated into the previous resource graph. Each triple inside a shadowed oval indicates a service dependency exploit, and the plaintext nodes in between shadowed ovals indicate the

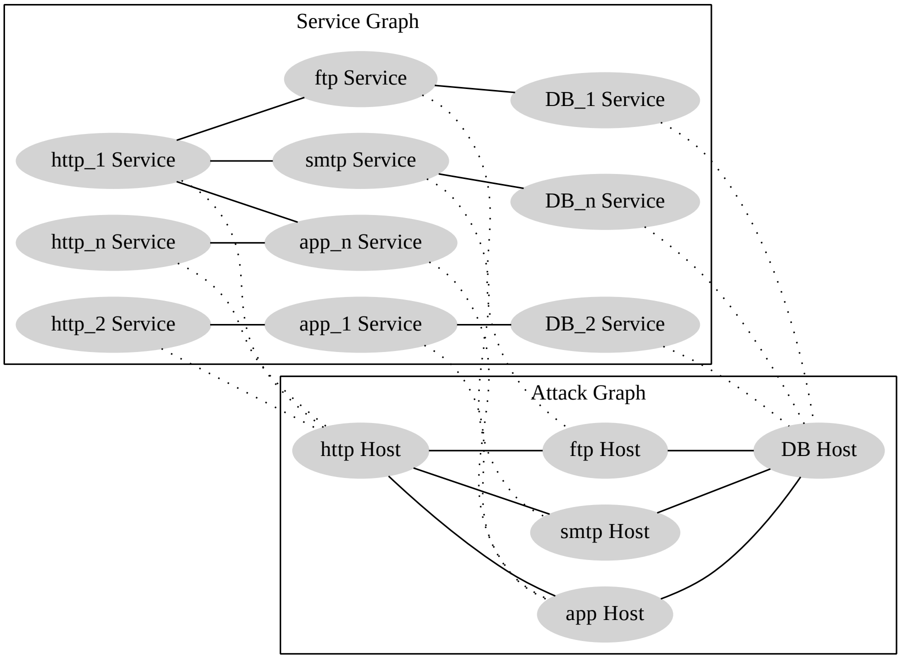

Fig. 3. An example of service dependency graph

type of impacted service, which can run on either virtual or physical resources; the dash line shows the dependency between services. This model can be used to represent various causal relationships between services and resources, e.g., all services running on top of a server may be compromised if attackers gain full control over the server, a server (and other services) may not be affected when one of the services running on the server is compromised, and a service involving multiple resources may be compromised either when one of the resources is compromised (e.g., a Web service may become unavailable if either the Web, application, or database server is down) or multiple resources are compromised at the same time (e.g., a Web service might be supported by multiple redundant Web servers).

We formalize the service dependency resource graph concept in Definition 6. The model extends the resource graph by adding nodes for services and their pre- and post-conditions, edges connecting services to those conditions, and edges inter-connecting the services (or their pre- and post-conditions) and the pre- and post-conditions of exploits (or the exploits). Definition 7 then extends the previous insider threat models based on the service dependency resource graph. We will apply this model in the upcoming sections to study the solution for mitigating the insider threats of remote administrators, and to conduct simulations to evaluate the effectiveness of the solution. In practice, the choice between those different models (e.g., *k*-zero day safety versus BN, or whether to consider service dependencies) will depend on the needs of specific applications and the available information or assumptions.

Definition 6 (Service Dependency Resource Graph)**.** *Given a network with the set of hosts H, the set of resources R, with the resource mapping $res(.) : H \to 2^R$, the set of zero day exploits $E = \{\langle r, h_s, h_d \rangle \mid h_s \in H, h_d \in H, r \in res(h_d)\}$ and their pre- and post-conditions C, the set of services S, their pre- and post-conditions C_s, a service dependency resource graph is a directed graph $G(E \cup C \cup S \cup C_s, R_r \cup R_i$ where $R_r \subseteq (C \cup C_s) \times (E \cup S)$, $R_i \subseteq (E \cup S) \times (C \cup C_s)$ are the dependency relations.*

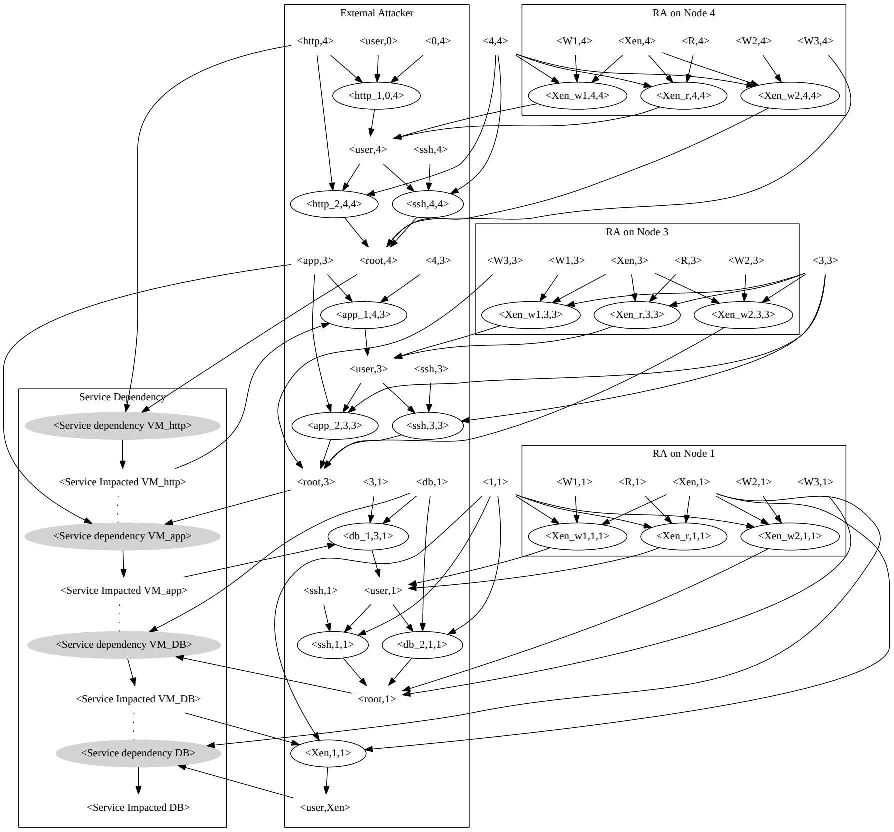

Fig. 4. Modeling insider threats using service dependency graph

Definition 7 (The Service Dependency-based Insider Threat Model)**.** *Given the maintenance task assignment (i.e., RA, T, P, RAT, TP, ta, and pa), 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 $G(E \cup S \cup C \cup C_s \cup E_r \cup C_r, R)$ the service dependency resource graph (where E, S, C, and C_s denote the original set of exploits, services, and conditions, respectively, and R denote the edges). We say $k0d(ra)$, $\frac{\sum_{ra \in RA} k0d(ra)}{|RA|}$, and $min(\{k0d(ra) : ra \in RA\})$ (or $P(ra)$, $\frac{\sum_{ra \in RA} P(ra)}{|RA|}$, and $min(\{P(ra) : ra \in RA\})$ in the case of BN-based metrics) 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 and Use Cases

In this section, we formulate the optimization-based solution for mitigating the insider threat of remote administrators during maintenance task assignment.

4.1. The Optimization-based Mitigation

Based on our definitions of the maintenance task assignment model and the insider threat model, we formulate the problem of optimal task assignment in Definition 8. The remote administrator task relation *RAT* basically defines the optimization constraints since it states which tasks may be assigned to which RA. Additional constraints in other forms may also be introduced, e.g., the maximum number of tasks that can be assigned to an RA.

Definition 8 (The Optimal Task Assignment Problem)**.** *Given a resource graph G, the set of 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* $\frac{\sum_{ra\in RA} k0d(ra)}{|RA|}$ *(or* $min(\{k0d(ra) : ra \in RA\})$*).*

Theorem 1. *The Optimal Task Assignment Problem (Definition 8) is NP-hard.*

Proof: First, evaluating the $k0d$ function is already NP-hard w.r.t. the size of the resource graph [14]. On the other hand, we provide a sketch of 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* [35] (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 k value for 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 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 (GA) [36] 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 added to the optimization problem. 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 4 and the fourth use case is based on the motivating example shown in Section 2.1. The last use case is based on the service dependency resource graph with three remote administrator and five maintenance tasks.

Table 5

Maintenance tasks assignments for use case A

User	A_1	B_1	C_1	D_1	E_1	A_2	B_2	C_2	D_2	E_2	A_3	B_3	C_3	D_3	E_3
Tasks Number	4	5	6	8	9	6	4	7	8	5	4	5	6	8	9
	1	10	7	3	2	9	3	10	1	2	1	2	7	3	10
k	3	1	2	2	1	1	3	1	2	3	3	3	2	2	1
$\bar{k}$	1.8					2					2.2				
Minimum k	1					1					1				

Use Case A:. In this case, each RA should be assigned with exactly two tasks (e.g., to evently distribute the tasks among all the RAs). The three tables shown in Table 5 show three possible assignments and the corresponding k values. Also, Figure 2 shows an example path (dashed lines) for tasks assigned to RA C_1 based on the left 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 right 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 (e.g., to ensure all RAs are employed while there is no consideration for their workload). 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 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 (e.g., due to the level of training or skill). 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 6 shows the 15 maintenance tasks to be assigned. In Table 6, the four tables on the bottom show four different scenarios of tasks assigned to RAs and each table shows different average k. The bottom table on the right shows the optimal task assignment in term of the average $k = 3.125$. In Figure 5, the red dashed line represents the path used by RA1 to reach the critical asset when task number 1 is assigned to RA1. Also, the solid red line shows the path when task number 12 is assigned to RA1.

Use Case E:. In this use case, we demonstrate how service dependencies may affect the task assignment. We have five maintenance tasks as presented in Table 7. Assume all VMs running on the http compute node have backups but some VMs running on the app compute node do not have a backup. The critical asset is given as the DB service. We have three RAs each of which can be assigned with a maximum of two maintenance tasks. Table 8 shows two possible ways to assign the maintenance tasks to RAs and the corresponding k values. We use the GA to find the optimal task assignment that satisfies the constraints given in this case. As shown in the right table, the maximal average of k values among all RAs is $\bar{k} = 2.3$. It can also be seen that the minimum k value among all RAs is always $k = 2$ in this special case. Figure 6, shows the service dependency graph. The shadowed oval represents the path followed by the attacker to compromise the service when the app VM (VMb) does not have a backup.

Table 6

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

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	1	4	8	2	3	7	6
	5	9	15	12	10	11	13	
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	2	3	4	5	6	7	8
	9	10	11	12	13	14	15	
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	2	3	5	6	15	13	8
	4	7	9	10	11	12	14	
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	2	3	5	6	14	4	8
	12	7	9	10	11	15	13	
k	3	2	4	4	3	1	3	5
$\bar{k}$	3.125							
Minimum k	1							

5. Simulations

This section shows simulation results on applying our mitigation solution under various constraints.

Experimental Settings. 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 R2017b's GA toolbox. To generate a large number of resource graphs and service dependency graphs for simulations, we start with seed graphs with realistic configurations similar to Figure 1 and then generate random resource graphs and service dependency graphs by injecting new nodes and edges into those seed graphs. Those resource graphs and service dependency graphs were used as the input to the optimization toolbox where the fitness function is to maximize the average or worst case insider threat values (given in Definition 4 and Definition 7); also, we used the optimization toolbox where the fitness

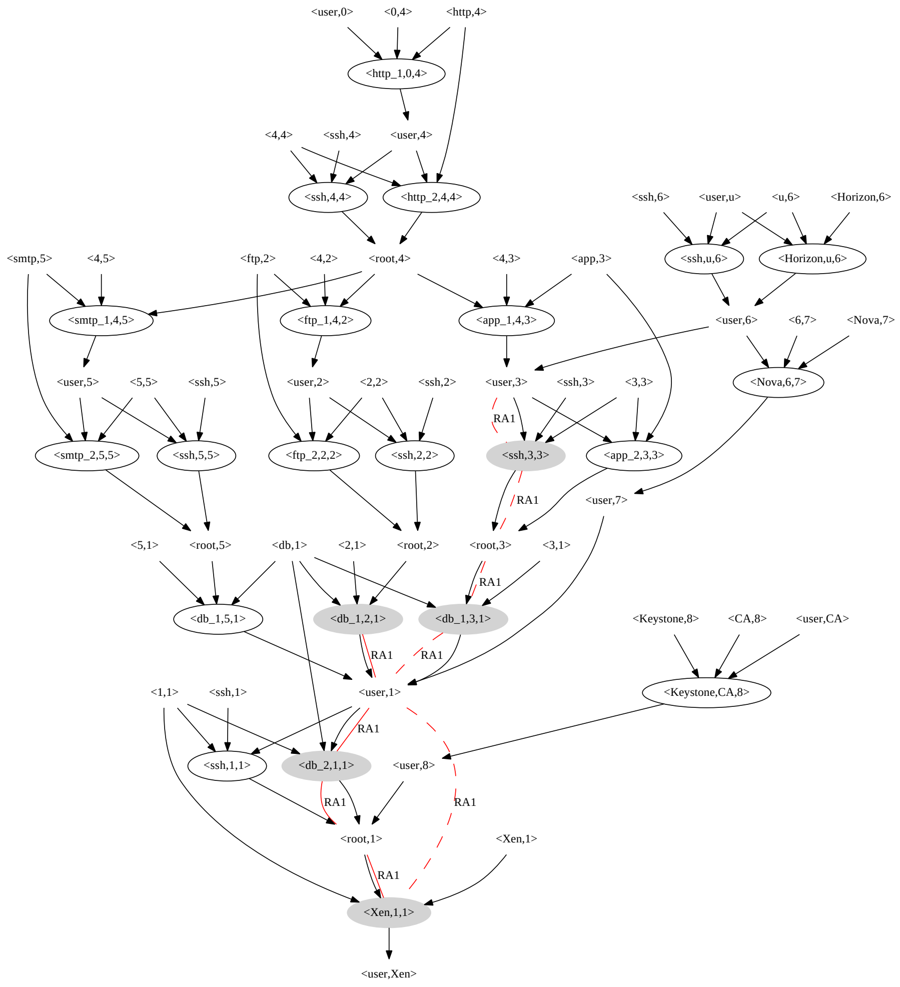

Fig. 5. Resource graph for the motivating example

function is to minimize the probability of reaching the critical asset by using the BN-based metric (given in Eq. 1) with various constraints, e.g., the number of available RAs and maintenance tasks, how many task may be assigned to each RA, assigning a fixed number of RAs with specific privilege, and assigning some of the maintenance tasks to the local administrators. We repeat each simulation on 300 different resource graphs to obtain the average result.

The Average Case Insider Threats. 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 7, 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 values 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.

Table 7

Maintenance tasks and privileges for the service dependency

Task Number	Node Number (in Figure 1)	Task Description	Privilege
1	4 ($http$)	Read log files for monitoring	Read
2	4 ($http$)	Install new systems	Write_L2
3	3 (app)	Read log files for monitoring	Read
4	3 (app)	Install new systems	Write_L2
5	1 (DB)	Read log files for monitoring	Read

Table 8

Maintenance tasks assignments for use case E

User	A_1	B_1	C_1	A_2	B_2	C_2
Tasks Number	1 4	2 5	3	1 2	3 4	5
k	2	2	2	3	2	2
$\bar{k}$		2			2.33	
Minimum k		2			2	

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 8, 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 values 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 four tasks per RA is slightly different mostly due to the limited number of RAs.

The Worst Case Insider Threats. 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 9 and Figure 10 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).

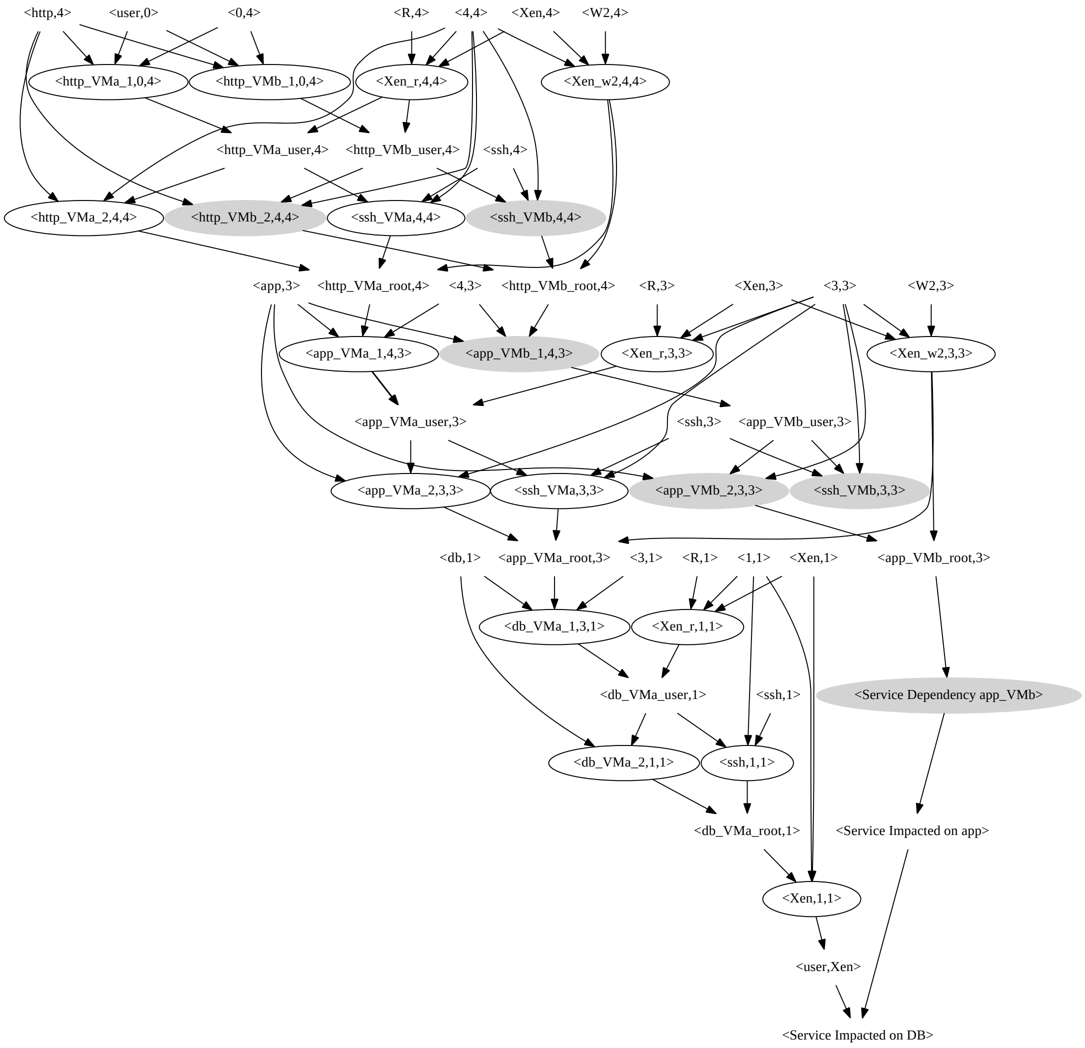

Fig. 6. The service dependency resource graph for use case E

Results and Implications: In Figure 9, 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 10, 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).

The Impact of the Highest Privileges. The objective of the next two simulations is to study how the average case insider threat (i.e., the average of *k* values among all RAs) can be when we assign some RAs with the highest privilege (W3) under our mitigation solution. In Figure 11, the number of available

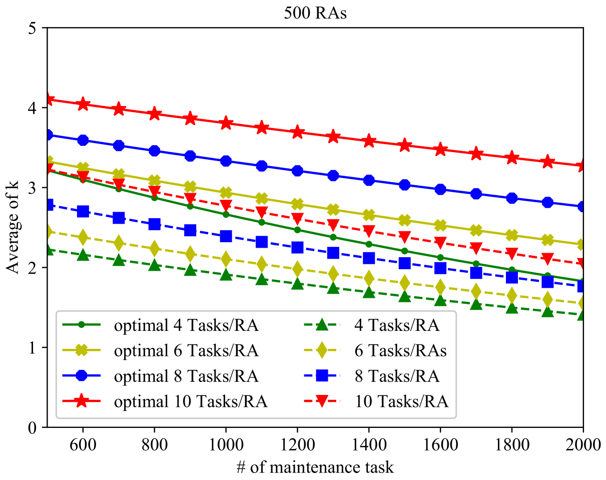

Fig. 7. The average k among 500 RAs before and after applying the mitigation solution

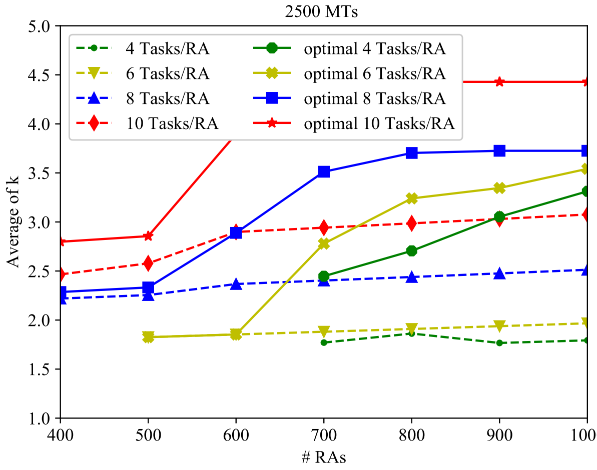

Fig. 8. The average k among different number of RAs before and after the solution

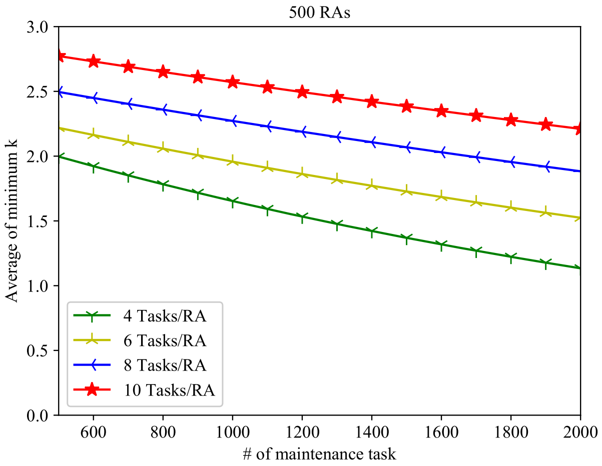

Fig. 9. The minimum k for 500 RAs

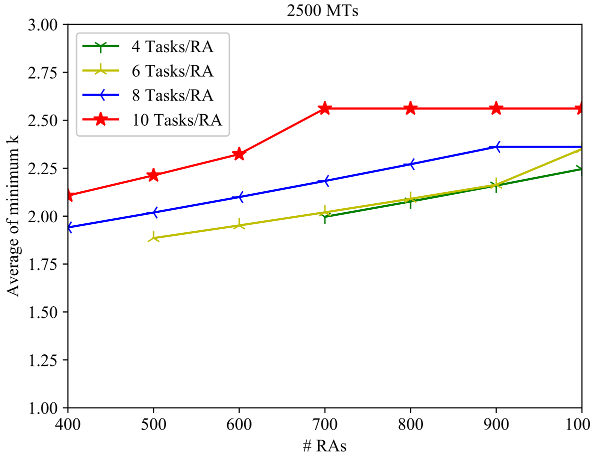

Fig. 10. The minimum k for varying # of RAs

RAs is fixed at 500 and each RA can handle 4 tasks as maximum, while the number of maintenance tasks is varied between 500 and 2,000 along the X-axis. The Y-axis shows the average k among all RAs. The solid lines show the results of average k after applying our mitigation solution under constraints about the number of RAs grant the W3 privilege before assigning tasks which are 20 RAs, 10 RAs, and no RA are granted the W3 privilege before task assignment, respectively.

Results and Implications: From the results, we can make the following observations. Grant the highest privilege to some of the RAs before assigning maintenance tasks can increase the average k to some degree when compared to the case when RAs are only granted privilege based on tasks needed to be performed. However, this decreases slower than others for the RAs who are granted privileges based on the maintenance tasks. As we can see in the figure, the trend (average k) of 20 RAs granted the highest privilege decreases faster (the insider threat increases) than others, which is expected because the highest privilege does not always correspond to the shortest path (e.g. W3 on the http node corresponds to a longer path than W2 on the app node).

In Figure 12, 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 k among all RAs. Each RA can perform 10 maintenance tasks at most. The solid lines show the results of average k after applying our mitigation solution under constraints about the number of RAs grant the W3 privilege before assigning tasks which are 40 RAs, 20 RAs and no RA are granted the W3 privilege before task assignment, respectively.

Results and Implications: From the results, we can make the following observations. Granting the highest privilege to some of the RAs before assigning maintenance tasks can increase the average k in all cases, and all cases follow the similar trend of starting flat then increasing almost linearly before reaching the plateau. This trend shows granting the highest privilege to some RAs will increase the average k since the number of RAs are increased and the number of tasks are fixed.

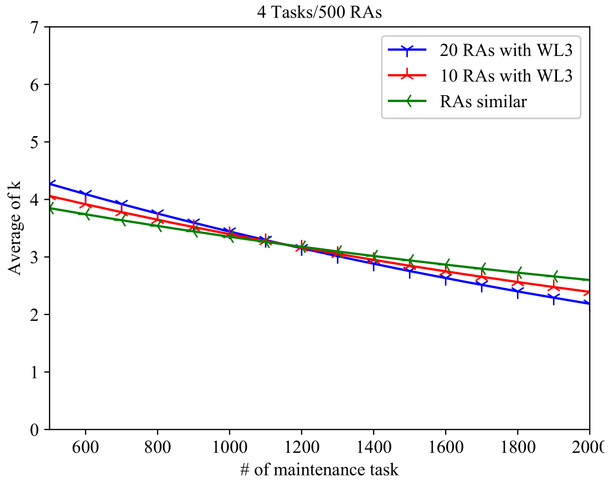

Fig. 11. The average k among 500 RAs with some of them granted the highest privilege

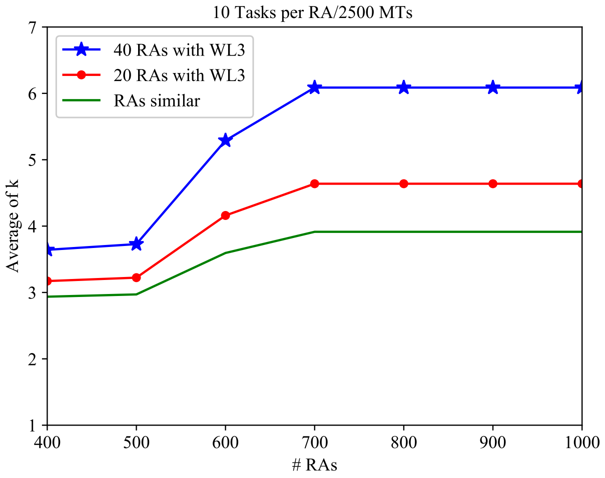

Fig. 12. The average k among different numbers of RAs with some of them granted the highest privilege

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 when we assign some RAs with the highest privilege (W3). In Figure 13, the number of available RAs is fixed at 500 and each RA can handle four tasks at most, while the number of maintenance tasks is varied between 500 and 2,000 along the X-axis. The Y-axis shows the minimum k among all RAs. In Figure 13, 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 k among all RAs. Each RA can perform 10 maintenance tasks at most. The solid lines show the results of minimum k after applying our mitigation solution under constraints about the number of RAs granted the W3 privilege before assigning tasks, which are 10 RAs, 20 RAs, and no RA granted W3 privilege in Figure 13, respectively, and 20 RAs, 40 RAs, and no RA granted W3 privilege in Figure 14, respectively.

Results and Implications: From the result, we can make the following observations. The minimum k in Figure 13 follows the similar trend as in Figure 11 which decreases almost linearly as the number of tasks increases and decrease faster as the number of RAs granted with the highest privilege increases. In Figure 14, the minimum *k* increases almost linearly before reaching the plateau.

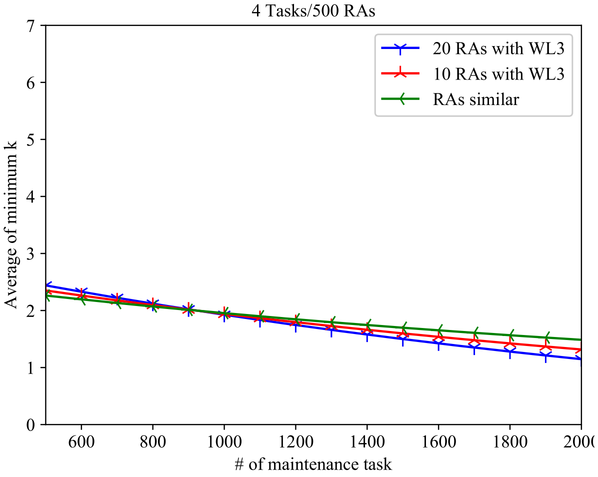

Fig. 13. The minimum *k* among 500 RAs with some of them granted the highest privilege

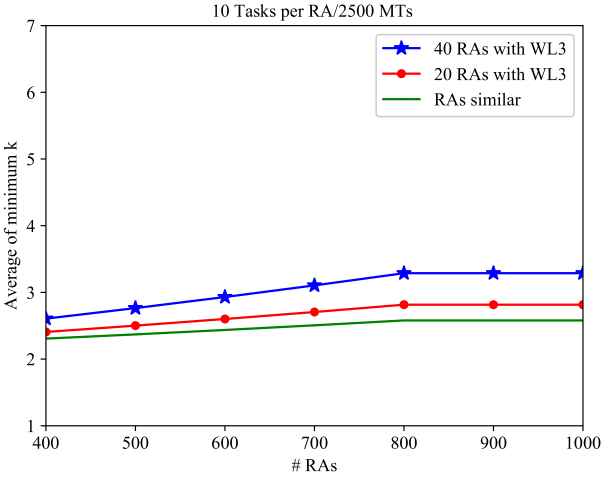

Fig. 14. The minimum *k* among different numbers of RAs with some of them granted the highest privilege

The Impact of Local Administrators. The objective of the next two simulations is to study how the average case insider threat (i.e., the average of *k* values among all RAs) behaves when we add a local administrator (LA) to perform MTs with their *k* value equal to the minimum *k*. In Figure 15, 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. In Figure 16, 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 *k* among all RAs in both figures. The solid lines show the results of average *k* after applying our mitigation solution under the constraint that an LA can perform MT with its *k* value equal to the minimum *k*.

Results and Implications: In Figure 15, we can see that the the average *k* mostly decreases slowly. The local administrator corresponds to the shortest path (minimum *k*) MTs needed to be performed. Increasing the number of tasks that can be assigned to each RA can increase the average *k* and the value of the average *k* decreases more slowly. In Figure 16, we can see that increasing the number of RAs and eliminating the highest risk tasks (minimum *k*) by assigning those tasks to the LAs will increase the average *k* linearly before reaching the plateau.

The Impact of Service Dependencies. The objective of the next simulations is to study how the service dependency can affect the average *k*. In Figure 17, 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 *k* among all RAs. The solid lines show the results of average *k* after considering the service dependency in our mitigation solution under constraints about the maximum number of tasks assigned to each RA. The dashed lines represent the results without considering the service dependency in our mitigation solution.

Results and Implications: In Figure 17, we can see that considering the service dependencies will decrease the average *k* because we would have to consider more critical assets at the same time (e.g. if we want to secure the database VM service from being compromised, we will need to consider any

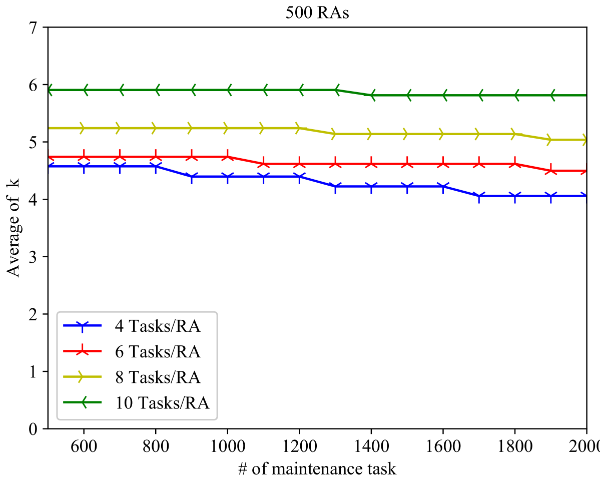

Fig. 15. The average k among 500 RAs while assigning some of the tasks to a local administrator

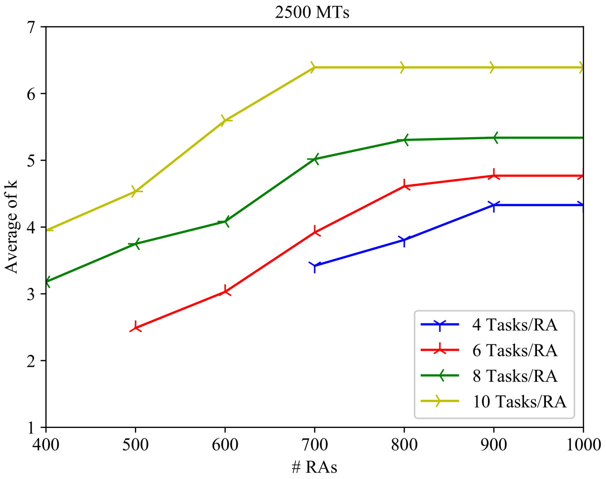

Fig. 16. The average k among different numbers of RAs while assigning some of the tasks to a local administrator

service connected to the database VM as a critical asset). Also, we can see that, the result of k when considering the service dependency is relatively close to the result of k without considering it, which means the mitigation solution is relatively effective for protecting services as well as resources.

In Figure 18, 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 k among all RAs. The solid lines show the results of average k after considering the service dependency in our mitigation solution under constraints about the maximum number of tasks assigned to each RA. The dashed lines represent the results without considering the service dependency in our mitigation solution.

Results and Implications: From the result, we can make the following observations. Considering the service dependency will increase the average k almost linearly before reaching the plateau. However, when we compare the average k with the result of the resource graph, we find that the average k under service dependency is lower which is expected because we are essentially considering more critical assets under the service dependency.

The objective of this simulation is to show how the minimum k behaves when considering the service dependency. In Figure 19, 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 k among all RAs. In Figure 20, 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 k among all RAs.

Results and Implications: In Figure 19, we can see that the minimum k values also decrease almost linearly as the number of tasks increases, which means the insider threat increases. From the results in Figure 20, we can see that the minimum k values also increase almost linearly before reaching the plateau as the number of RAs increases. The increase here is slower than that in Figure 18, which means the worst case (minimum k values) are more difficult to improve.

The BN-based Metric. The objective of this simulation is to apply the BN-based metric instead of the k-zero day safety metric on both the resource graph and the service dependency graph. In Figures 21 and 22, 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 probability to compromise the critical asset among all RAs.

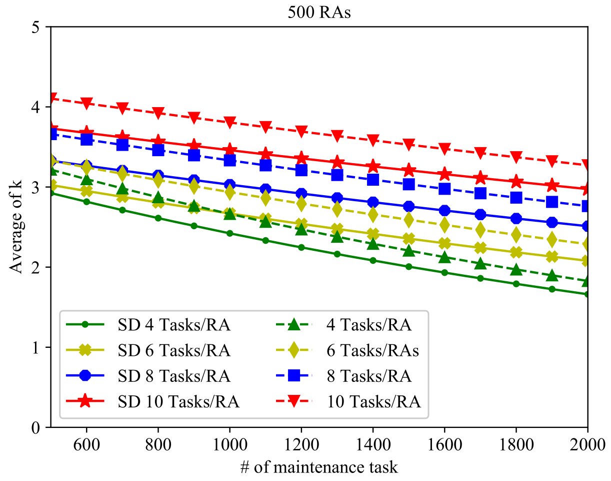

Fig. 17. The average *k* among 500 RAs with and without considering service dependency

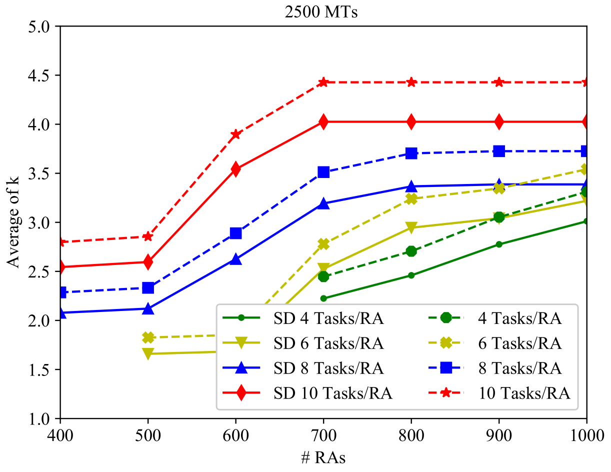

Fig. 18. The average *k* among different numbers of RAs with and withoutconsidering service dependency

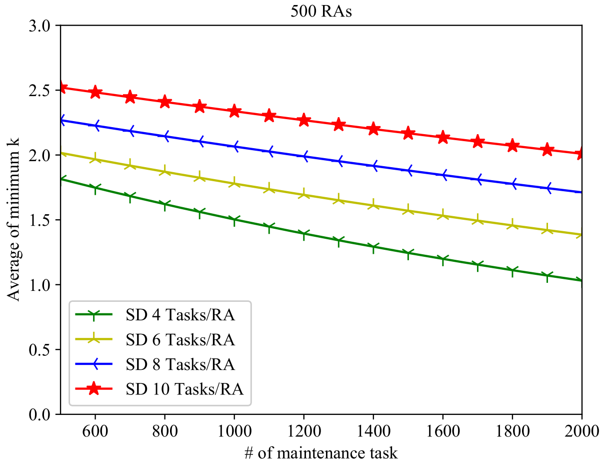

Fig. 19. The minimum *k* among 500 RAs under service dependency

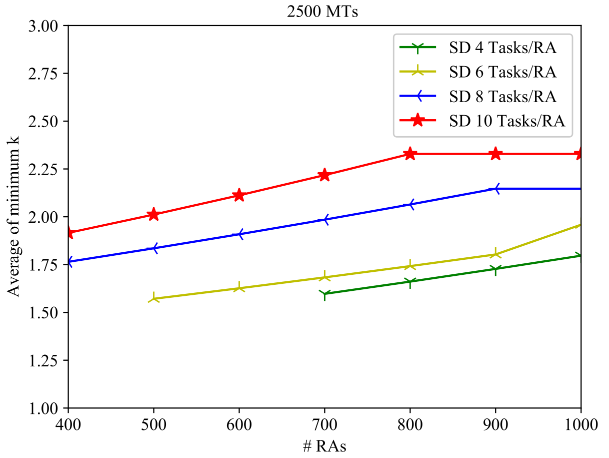

Fig. 20. The minimum *k* among different numbers of RAs under service dependency

Results and Implications: From the results, we can make the following observations. In Figure 21, The probability to reach a critical asset almost increases linearly when the number of maintenance tasks increases while the number of RAs are fixed. Also, we find increasing the maximum number of tasks that can be assigned to each RA slows the increasing rate of the probability which means slower increase in the insider threat. In Figure 22, we find the result follows the similar trend as the previous figure. However, the probability to reach the critical assets in the service dependency resource graph is much higher than that in the resource graph (e.g. for a maximum four tasks for each RA, the probability is almost 25% higher in the service dependency resource graph), which is expected because the service dependency means attackers would have more options to compromise the critical assets.

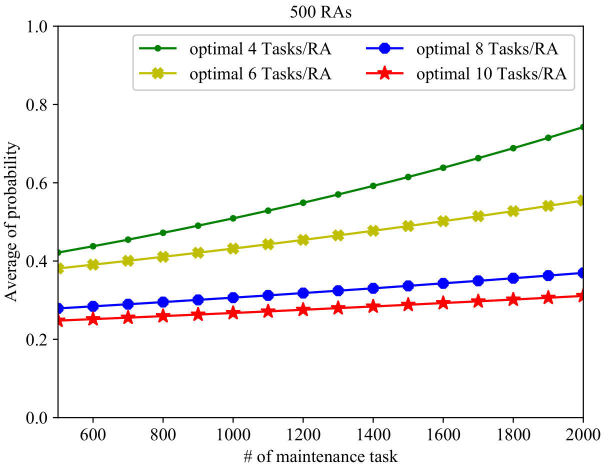

Fig. 21. The probability to compromise critical assets based on the resource graph

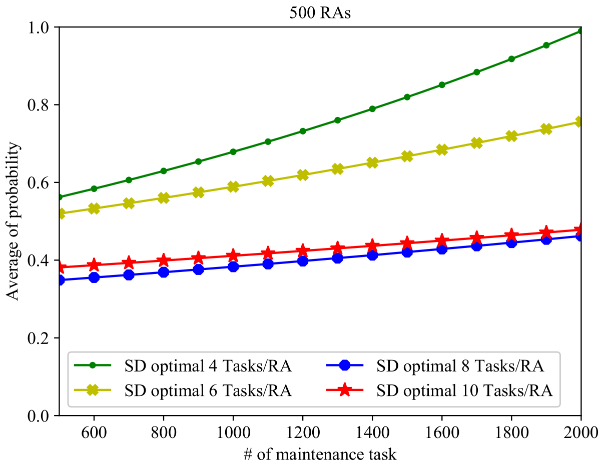

Fig. 22. The probability to compromise critical assets based on the service dependency resource graph

6. Related Work

The insider threat is a challenging issue for both traditional networks and clouds. Ray and Poolsapassit propose an alarm system to monitor the behavior of malicious insiders using the attack tree [37]. Mathew et al. use the capability acquisition graphs (CAG) to monitor the abuse of privileges by malicious insiders [38]. Sarkar et al. propose DASAI to analyze if a process contains a step that meet the insider attack condition [39]. Chinchani et al. propose a graph-based model for insider attacks and measure the threat [40]. Althebyan and Panda propose predication and detection model for insider attacks based on knowledge gathered by the internal users during work time in the organization [41]. Bishop et al. present insider threat definition based on security policies and determine the source of risk [42]. Roy et al. study an employee assignment problem to find an optimal tasks assigned to the employee based on constraints in role-based access control [43].

There exists only limited effort on insider attacks in the context of clouds. Our previous work focuses on applying different threat modeling techniques to cloud data center infrastructures where the focus is on external attackers [21]. Gruschka and Jensen devise a high level attack surface framework to show from where the attack can start [44]. The NIST emphasizes the importance of security measuring and metrics for cloud providers in [16]. A framework is propose by Luna et al. for cloud security metrics using basic building blocks [45]. Resource graphs can be automatically generated by modeling the network and vulnerabilities and many useful analyses may be performed using resource graphs [46–49]; however, our work is the first to use resource graphs for modeling insider attacks.

There exist many works on network security metrics in general [50, 51]. Some of those works focus on modeling known vulnerabilities for network security [48, 52] while other works focus on modeling unknown vulnerabilities (zero day attacks) [14, 53–55], which are usually considered unmeasurable due to the uncertainties involved [56]. A BN-based security metric applies resource graphs to measure the security level of a network [31]; the metric converts the CVSS scores of vulnerabilities into attack probabilities and then obtain the overall attack likelihood for reaching critical assets. We apply this metric to measure insider threats in this paper. Security metrics and measurements in clouds still face many challenges as shown in [57]. Following security standards is shown to be not enough to ensure the security of cloud infrastructures and security metrics may help to evaluate the security level [58]. Halabi and Bellqich use the Goal-Question-Metric to develop quantitative evaluation metric to help the cloud provider to evaluate its cloud security service and to know the level of security [59]. Finally, there exist some works focusing on high level risk assessment for clouds, such as the framework to evaluate the security of clouds based on the security impact in six categories and abstract levels of security impact [17].

The proactive mitigation of security threats can be performed through network hardening. Early works on network hardening focus on breaking all the attack paths that an attacker can follow to compromise a critical asset, either in the middle of the paths or at the beginning (disabling initial conditions) [60–62]. Network hardening using optimization is proposed by Gupta et al. in [63], refined with multiple objective optimization by Dewri et al. in [64] and with dynamic conditions by Poolsappasit et al. in [65], and extended as vulnerability analysis with cost/benefit assessment [66] and risk assessment [67]. More recent works [68, 69] focus on combining multiple hardening options through optimization, and improving the diversity of networks, respectively. We borrow the optimization-based hardening techniques [63, 69] to mitigate the insider threats in this paper. In the context of clouds, there are works on securing the cloud from insider attacks by limiting the trust on the compute node [10]. Li et al. focuses on supporting users to configure privacy protection in compute nodes [9]. Closest to our work, Bleikertz et al. focus on securing the cloud during maintenance time by limiting the privileges granted to the remote administrators based on the tasks assigned to that administrator [5].

There are many works that focus on the service dependency. Some of those existing works focus on the mission impact which relies on the service dependency model to capture the threats that can impact a mission [70, 71]. Chen et al. [34] show how the service dependency can affect the system since compromising a service can affect other services hosted in same network. Natarajan et al. [72] develop a tool for automaticaly finding dependency between services in large networks. Another work develops different techniques to monitor service dependencies in distributed systems [73]. Closest to our work, Sun et al. [12] embeds mission impact and service dependency in an attack graph to to find how service dependency can impact different missions. We borrow their model of service dependency in evaluating the insider threats of remote administrators.

7. Conclusion

In this paper, we have modeled the insider threat during maintenance task assignment for cloud providers to better understand the threats posed by third party remote administrators. We have formulated the optimal assignment as an optimization problem and applied a standard optimization technique to derive solutions under different constraints. We have extended our insider threat models to consider service dependencies. Also, we applied the BN-base metric to take all attack paths and known vulnerabilities into consideration. Based on such models, we have conducted simulations for different use cases whose results show our solution can significantly reduce the insider threat of remote administrators. The limitations and corresponding future directions are as follows. First, the current mitigation solution is static in nature, and we will devise incremental solutions to handle streams of new maintenance tasks and dynamics (joining or leaving) of RAs, changing priority or weight of tasks, etc.). Second, our model has kept the cost implicit, and we will consider explicit cost models (e.g., based on the nature of the tasks, the amount or duration of tasks, and privileges needed) and incorporate such cost models into the mitigation solution.

Acknowledgements

Authors with Concordia University are partially supported by the Natural Sciences and Engineering Research Council of Canada under Discovery Grant N01035. Sushil Jajodia was supported in part by the National Institute of Standards and Technology grants 60NANB16D287 and 60NANB18D168, National Science Foundation under grant IIP-1266147, Army Research Office under grant W911NF-13-1-0421, and Office of Naval Research under grant N00014-15-1-2007.

Disclaimer

Commercial products are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the identified products are necessarily the best available for the purpose.

References

[1] M. Doucet and M. Lari, "Cyber security and cybercrime in canada, 2017," 2018. Available at: https://www150.statcan.gc.ca/n1/en/catalogue/71-607-X2018007.

[2] Gartner, "Gartner forecasts worldwide public cloud revenue to grow 21.4 percent in 2018," 2018. Available at: https://www.gartner.com/newsroom/id/3871416.

[3] Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing v 3.0," 2011.

[4] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of network and computer applications*, vol. 34, no. 1, pp. 1–11, 2011.

[5] 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, (New York, NY, USA), pp. 83–84, ACM, 2012.

[6] 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*, pp. 387–394, July 2012.

[7] Cloud Security Alliance, "Top threats to cloud computing," 2018. Available at: https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf.

[8] 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.

[9] 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, (New York, NY, USA), pp. 59–68, ACM, 2013.

[10] 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, (New York, NY, USA), pp. 341–352, ACM, 2016.

[11] N. Alhebaishi, L. Wang, S. Jajodia, and A. Singhal, "Modeling and mitigating the insider threat of remote administrators in clouds," in *Data and Applications Security and Privacy XXXII - 32nd Annual IFIP WG 11.3 Conference, DBSec 2018, Bergamo, Italy, July 16-18, 2018, Proceedings*, pp. 3–20, 2018.

[12] X. Sun, A. Singhal, and P. Liu, "Towards actionable mission impact assessment in the context of cloud computing," in *Data and Applications Security and Privacy XXXI - 31st Annual IFIP WG 11.3 Conference, DBSec 2017, Philadelphia, PA, USA, July 19-21, 2017, Proceedings*, pp. 259–274, 2017.

[13] M. Albanese and S. Jajodia, "A graphical model to assess the impact of multi-step attacks," *The Journal of Defense Modeling & Simulation*, vol. 15, pp. 79–93, 01 2018.

[14] 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*, vol. 11, pp. 30–44, Jan 2014.

[15] L. Wang, S. Jajodia, A. Singhal, and S. Noel, "k-zero day safety: Measuring the security risk of networks against unknown attacks.," in *ESORICS*, pp. 573–587, Springer, 2010.

[16] "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].

[17] 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*, pp. 280–288, July 2010.

[18] 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*, pp. 245–250, July 2011.

[19] 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, (New York, NY, USA), pp. 12:1–12:6, ACM, 2011.

[20] F. B. Shaikh and S. Haider, "Security threats in cloud computing," in *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*, pp. 214–219, Dec 2011.

[21] 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*, pp. 302–319, 2016.

[22] K. Bakshi, "Cisco cloud computing-data center strategy, architecture, and solutions," *DOI= http://www. cisco. com/web/strategy/docs/gov/CiscoCloudComputing_WP. pdf*, 2009.

[23] 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].

[24] Openstack, "Openstack Operations Guide." http://docs.openstack.org/openstack-ops/content/openstack-ops_preface.html. [Online; accessed 27/08/2015].

[25] "Amazon Web Services." https://aws.amazon.com/, 2018. [Online; accessed 28/02/2018].

[26] "Google Cloud Platform." https://cloud.google.com/, 2018. [Online; accessed 28/02/2018].

[27] "Microsoft Azure." https://azure.microsoft.com, 2018. [Online; accessed 28/02/2018].

[28] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, pp. 38–47, Feb. 1996.

[29] M. Zhang, L. Wang, S. Jajodia, A. Singhal, and M. Albanese, "Network diversity: A security metric for evaluating the resilience of networks against zero-day attacks," *IEEE Transactions on Information Forensics and Security*, vol. 11, pp. 1071–1086, 2016.

[30] L. Wang, M. Zhang, S. Jajodia, A. Singhal, and M. Albanese, "Modeling network diversity for evaluating the robustness of networks against zero-day attacks," in *Proceedings of ESORICS'14*, pp. 494–511, 2014.

[31] M. Frigault and L. Wang, "Measuring network security using bayesian network-based attack graphs," in *Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International*, pp. 698–703, July 2008.

[32] P. Mell, K. Scarfone, and S. Romanosky, "Common vulnerability scoring system," *IEEE Security & Privacy*, vol. 4, no. 6, pp. 85–89, 2006.

[33] W. Nzoukou, L. Wang, S. Jajodia, and A. Singhal, "A unified framework for measuring a network's mean time-to-compromise," in *2013 IEEE 32nd International Symposium on Reliable Distributed Systems*, pp. 215–224, Sep. 2013.

[34] X. Chen, M. Zhang, Z. M. Mao, and P. Bahl, "Automating network application dependency discovery: Experiences, limitations, and new solutions," in *8th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2008, December 8-10, 2008, San Diego, California, USA, Proceedings*, pp. 117–130, 2008.

[35] V. Kann, "A compendium of np optimization problems," in *WWW Spring 1994*, 1994.

[36] D. E. Golberg, "Genetic algorithms in search, optimization, and machine learning," *Addion wesley*, vol. 1989, 1989.

[37] I. Ray and N. Poolsapassit, *Computer Security – ESORICS 2005: 10th European Symposium on Research in Computer Security, Milan, Italy, September 12-14, 2005. Proceedings*, ch. Using Attack Trees to Identify Malicious Attacks from Authorized Insiders, pp. 231–246. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[38] 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*, pp. 1–8, June 2008.

[39] A. Sarkar, S. KÃűhler, S. Riddle, B. Ludaescher, and M. Bishop, "Insider attack identification and prevention using a declarative approach," in *2014 IEEE Security and Privacy Workshops*, pp. 265–276, May 2014.

[40] 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)*, pp. 108–117, June 2005.

[41] Q. Althebyan and B. Panda, "A knowledge-base model for insider threat prediction," in *2007 IEEE SMC Information Assurance and Security Workshop*, pp. 239–246, June 2007.

[42] 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, (New York, NY, USA), pp. 1–12, ACM, 2008.

[43] 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.*, vol. 7, pp. 10:1–10:24, Dec. 2016.

[44] N. Gruschka and M. Jensen, "Attack surfaces: A taxonomy for attacks on cloud services," in *2010 IEEE 3rd international conference on cloud computing*, pp. 276–279, IEEE, 2010.

[45] 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*, pp. 245–250, IEEE, 2011.

[46] 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*, pp. 273–284, 2002.

[47] L. Wang, A. Singhal, and S. Jajodia, "Toward measuring network security using attack graphs," in *Proceedings of the 2007 ACM Workshop on Quality of Protection*, QoP '07, (New York, NY, USA), pp. 49–54, ACM, 2007.

[48] L. Wang, A. Singhal, and S. Jajodia, *Measuring the Overall Security of Network Configurations Using Attack Graphs*, pp. 98–112. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[49] P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, graph-based network vulnerability analysis," in *Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002*, pp. 217–224, 2002.

[50] M. Pendleton, R. Garcia-Lebron, J.-H. Cho, and S. Xu, "A survey on systems security metrics," *ACM Comput. Surv.*, vol. 49, pp. 62:1–62:35, Dec. 2016.

[51] L. Wang, S. Jajodia, and A. E. Singhal, *Network Security Metrics*. Springer, 2017.

[52] M. Albanese, S. Jajodia, and S. Noel, "Time-efficient and cost-effective network hardening using attack graphs," in *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012)*, pp. 1–12, June 2012.

[53] L. Wang, M. Zhang, S. Jajodia, A. Singhal, and M. Albanese, "Modeling network diversity for evaluating the robustness of networks against zero-day attacks," in *Computer Security - ESORICS 2014 - 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part II*, pp. 494–511, 2014.

[54] M. Zhang, L. Wang, S. Jajodia, A. Singhal, and M. Albanese, "Network diversity: A security metric for evaluating the resilience of networks against zero-day attacks," *IEEE Trans. Information Forensics and Security*, vol. 11, no. 5, pp. 1071–1086, 2016.

[55] L. Wang, S. Jajodia, A. Singhal, and S. Noel, "*k*-zero day safety: Measuring the security risk of networks against unknown attacks," in *Computer Security - ESORICS 2010, 15th European Symposium on Research in Computer Security, Athens, Greece, September 20-22, 2010. Proceedings*, pp. 573–587, 2010.

[56] J. McHugh, "Quality of protection: measuring the unmeasurable?," in *Proceedings of the 2nd ACM Workshop on Quality of Protection, QoP 2006, Alexandria, VA, USA, October 30, 2006*, pp. 1–2, 2006.

[57] T. Branco, Jr. and H. Santos, "What is missing for trust in the cloud computing?," in *Proceedings of the 2016 ACM SIGMIS Conference on Computers and People Research*, SIGMIS-CPR '16, (New York, NY, USA), pp. 27–28, ACM, 2016.

[58] J. Bayuk and A. Mostashari, "Measuring systems security," *Systems Engineering*, vol. 16, no. 1, pp. 1–14, 2013.

[59] T. Halabi and M. Bellaiche, "Towards quantification and evaluation of security of cloud service providers," *Journal of Information Security and Applications*, vol. 33, no. Supplement C, pp. 55 – 65, 2017.

[60] L. Wang, S. Noel, and S. Jajodia, "Minimum-cost network hardening using attack graphs," *Computer Communications*, vol. 29, no. 18, pp. 3812 – 3824, 2006.

[61] 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*, pp. 273–284, IEEE, 2002.

[62] L. Wang, M. Albanese, and S. Jajodia, *Network Hardening: An Automated Approach to Improving Network Security*. Springer Publishing Company, Incorporated, 2014.

[63] M. Gupta, J. Rees, A. Chaturvedi, and J. Chi, "Matching information security vulnerabilities to organizational security profiles: a genetic algorithm approach," *Decision Support Systems*, vol. 41, no. 3, pp. 592 – 603, 2006. Intelligence and security informatics.

[64] R. Dewri, N. Poolsappasit, I. Ray, and D. Whitley, "Optimal security hardening using multi-objective optimization on attack tree models of networks," in *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 204–213, ACM, 2007.

[65] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using bayesian attack graphs," *Dependable and Secure Computing, IEEE Transactions on*, vol. 9, no. 1, pp. 61–74, 2012.

[66] R. Dewri, I. Ray, N. Poolsappasit, and D. Whitley, "Optimal security hardening on attack tree models of networks: a cost-benefit analysis," *International Journal of Information Security*, vol. 11, no. 3, pp. 167–188, 2012.

[67] S. Wang, Z. Zhang, and Y. Kadobayashi, "Exploring attack graph for cost-benefit security hardening: A probabilistic approach," *Computers & security*, vol. 32, pp. 158–169, 2013.

[68] D. Borbor, L. Wang, S. Jajodia, and A. Singhal, "Securing networks against unpatchable and unknown vulnerabilities using heterogeneous hardening options," in *Data and Applications Security and Privacy XXXI* (G. Livraga and S. Zhu, eds.), (Cham), pp. 509–528, Springer International Publishing, 2017.

[69] 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*, pp. 295–312, 2016.

[70] G. Jakobson, "Mission cyber security situation assessment using impact dependency graphs," in *14th International Conference on Information Fusion*, pp. 1–8, July 2011.

[71] R. E. Sawilla and X. Ou, “Identifying critical attack assets in dependency attack graphs,” in *Computer Security - ESORICS 2008, 13th European Symposium on Research in Computer Security, Málaga, Spain, October 6-8, 2008. Proceedings*, pp. 18–34, 2008.

[72] A. Natarajan, P. Ning, Y. Liu, S. Jajodia, and S. E. Hutchinson, “Nsdminer: Automated discovery of network service dependencies,” in *Proceedings of the IEEE INFOCOM 2012, Orlando, FL, USA, March 25-30, 2012*, pp. 2507–2515, 2012.

[73] B. W. P. III, P. Ning, and S. Jajodia, “On the accurate identification of network service dependencies in distributed systems,” in *Strategies, Tools , and Techniques: Proceedings of the 26th Large Installation System Administration Conference, LISA 2012, San Diego, CA, USA, December 9-14, 2012*, pp. 181–194, 2012.