DCDIDP: A Distributed, Collaborative, and Data-driven Intrusion Detection and Prevention Framework for Cloud Computing Environments

Saman Taghavi Zargar, Hassan Takabi, and James B.D. Joshi
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA, USA
Emails: {stzargar, hatakabi, jjoshi}@sis.pitt.edu

***Abstract*—With the growing popularity of cloud computing, the exploitation of possible vulnerabilities grows at the same pace; the distributed nature of the cloud makes it an attractive target for potential intruders. Despite security issues delaying its adoption, cloud computing has already become an unstoppable force; thus, security mechanisms to ensure its secure adoption are an immediate need. Here, we focus on intrusion detection and prevention systems (IDPSs) to defend against the intruders. In this paper, we propose a Distributed, Collaborative, and Data-driven Intrusion Detection and Prevention system (DCDIDP). Its goal is to make use of the resources in the cloud and provide a holistic IDPS for all cloud service providers which collaborate with other peers in a distributed manner at different architectural levels to respond to attacks. We present the DCDIDP framework, whose infrastructure level is composed of three logical layers: network, host, and global as well as platform and software levels. Then, we review its components and discuss some existing approaches to be used for the modules in our proposed framework. Furthermore, we discuss developing a comprehensive trust management framework to support the establishment and evolution of trust among different cloud service providers.**

***Index Terms*—Cloud computing, intrusion detection, collaborative IDPS, distributed IDPS.**

I. Introduction

Recent advances in distributed computing, grid computing, virtualization mechanisms, and utility computing had led Cloud Computing towards becoming one of the industry buzz words of our decade. Cloud computing has been defined by the U.S. National Institute of Standards and Technology (NIST) as follows:
”A model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three delivery models, and four deployment models” [1].

The cloud computing model as defined by NIST, consists of cloud providers and cloud consumers. A cloud provider is a person, organization or entity responsible for making an infrastructure, platform or software available to cloud consumers as a service (IaaS, PaaS or SaaS). The person or organization that maintains a business relationship with, and uses one or more of these services (i.e. IaaS, PaaS or SaaS) from cloud providers, is a cloud consumer [1].

There are three different layers involved in Cloud Computing: the infrastructure layer, the platform layer, and the software layer. The infrastructure layer is the basis for the cloud computing environment and the user does not have direct access to it. Users have access only to the virtualized infrastructure layer through the virtual machine (VM) abstraction of different hardware components (e.g. VM abstraction of a physical server and its related networks) in the cloud infrastructure. Amazon S3 (http://aws.amazon.com/s3) and FlexiScale (http://www.flexiant.com/products/flexiscale) are examples of IaaS (infrastructure layer) for storage and maintaining virtual servers, respectively. The platform layer includes software. For instance, it includes all of the APIs for a specific programming language or virtualized operating system (OS) of a server [2]. Samples of PaaS (platform layer), including Google App Engine (http://code.google.com/appengine) and LoadStorm (http://loadstorm.com), are used to run web applications and test their performance, respectively. Finally, the software or application layer includes other user-specific software (SaaS) offered by the cloud providers, such as Zoho (http://www.zoho.com), Zuora (http://www.zuora.com), and Salesforce (http://www.salesforce.com).

As the popularity of the services provided in the cloud environment grows rapidly, the exploitation of possible vulnerabilities (e.g. malicious resource consumption, disrupting services, etc.) grows at the same pace. The distributed nature of the cloud model makes it an even more attractive target for intruders. Furthermore, there may be various incentives for the competitors to initiate attacks against each other, thanks to the commercialized nature of the cloud environment.

IDPSs are among the most popular of the front line fundamental tools to defend computation and communication infrastructures from intruders. They are usually deployed either at the destination host (Host-based-IDPS)(HIDPS) or at the edge of the network infrastructure (Network-based-IDPS)(NIDPS),

COLLABORATECOM 2011, October 15-18, Orlando, United States

DOI 10.4108/icst.collaboratecom.2011.247158

in order to protect computation and communication infrastructures from external attacks. While some of the current HIDPSs or NIDPSs have emerged and been commercialized for the cloud computing environment and been used in practice for detecting malicious behaviors against protected hosts or network environments, they do not meet the requirements for an extremely challenging collaborative environment; such an environment requires the support of dynamic, real-time, and high-performance applications. For instance, while reactive IDPSs support real-time reactions to the attacks, they may introduce huge overheads in dynamically changing cloud environments. Passive approaches have poor response time; hence, they lead to serious performance degradation and hamper collaboration among cloud providers. Therefore, proactive approaches to provide real-time, high performance attack prevention and analysis are to be implemented to detect any malicious activity before the start of an attack, and prevent it from accessing important resources.

In this paper, we propose a distributed, collaborative, and data-driven IDP framework for cloud computing environments. This framework integrates IDPS in all three layers of cloud computing (i.e. IaaS, PaaS or SaaS). We present the DCDIDP framework, its components and discuss some of the existing approaches to be used for the modules in our proposed framework.

The rest of this paper is organized as follows: Section 2 discusses the motivation and the potential benefits of our proposed framework. Section 3 presents a brief background on intrusion detection and prevention systems. In Section 4, we describe our proposed DCDIDP framework, explain its components and discuss some of the possible approaches to be employed. In Section 5, we discuss trust management among collaborating cloud service providers. Section 6 discusses the related work and, finally, Section 7 offers our conclusions and discusses potential future work.

II. Motivation

Most of the currently employed/proposed IDPSs for the cloud support detection and prevention at each layer and mostly independent of other layers [3]–[5]. Furthermore, complex IDP management is required for heterogeneous cloud environments. Developing a holistic IDP which significantly simplifies IDP management has not been considered to date. Lack of collaboration among different components within a cloud provider or among different cloud providers for detection and prevention of attacks is another drawback to current proposed approaches. Collaboration decreases the risks and damages of prior attacks due to the shared knowledge from other collaborators. Collaboration also increases attack detection and the speed and strength of prevention efforts. Distributed detection and prevention of the attacks within/among cloud providers can reduce the complexity of redundant monitoring of attack flows [6] at different check points which should be one of the main features in the next generation of IDPSs suitable for cloud environments. Creating comprehensive local/global databases to be used for detection tasks by IDPSs is another major requirement in order for IDPSs to act as a comprehensive defense mechanism. Therefore, proposing a comprehensive, holistic, collaborative, distributed, and data-driven framework that considers each and every layer of the cloud's requirements is our primary motivation in this project.

Both cloud providers and cloud customers will benefit significantly if there is a comprehensive DCDIDP that dynamically evolves and gradually mobilizes the cloud's resources as suspicion about attacks increases. Such a system needs to provide a holistic IDPS for all of the cloud providers who collaborate. The system needs to respond to the attacks, by collaborating with peer systems in a distributed fashion, as near as possible to attack sources and at different levels of operations (e.g. network, host, VM). Furthermore, the DCDIDP system needs to support various architectural levels/components including the virtualized components, hosts, and networks. Some of the potential benefits that such a system can provide are as follows:

- **Distributed IDP**: Attacks may originate within the infrastructure or through one of the virtual machines within a physical host itself. A cloud environment may have several administrative domains and possible attacks may be directed against resources located within the cloud infrastructure itself. Hence, a proper defense strategy needs to be distributed to effectively detect and react to the attacks. Distributed IDP potentially relieves high storage and processing overheads caused by central IDP management as the load will be distributed among different points in the cloud. Furthermore, central IDPS can be easily overwhelmed by large-scale, traffic-intensive attacks, as the ability of the victim to both detect attack and filter traffic effectively diminishes considerably [7].

- **Collaborative IDPS**: DCDIDP combines the advantages of cloud computing with that of efficient collaboration to make the attack detection and prevention more effective. Attack detection and prevention tasks could be less burdensome if it can be distributed within and among the cloud providers. Cloud providers may share their knowledge about detecting malicious activities at all three layers of the cloud environment. However, it is important to incentivize cloud providers to collaborate with their peers and to share their experiences with detection and prevention in order to improve these capabilities.

- **Data-driven IDP and interoperability among the cloud providers**: DCDIDP leads to dynamic evolution of filtering rules and access lists among the cloud providers to deal with ever-evolving characteristics of the attacks in diverse cloud environments. Furthermore, some applications in the cloud environments are very diverse, with possible extreme performance requirements that need to be supported by their underlying IDPSs

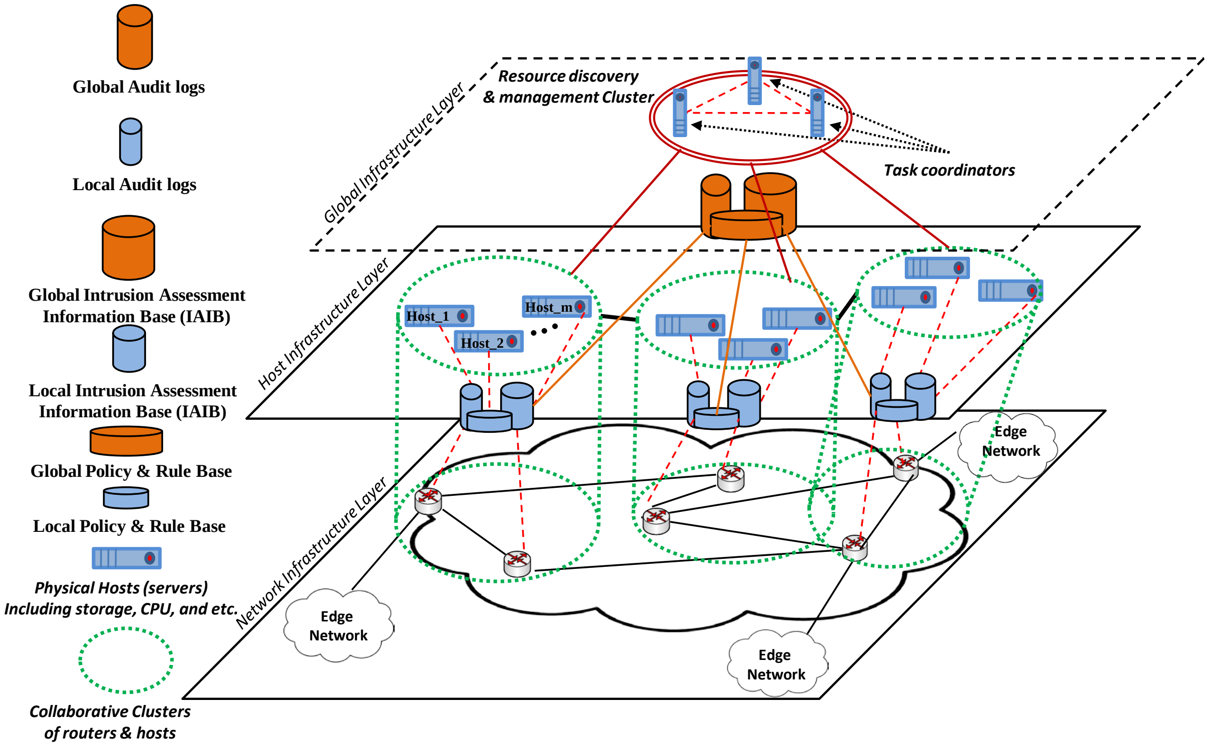
Global Audit logs
Local Audit logs
Global Intrusion Assessment Information Base (IAIB)
Local Intrusion Assessment Information Base (IAIB)
Global Policy & Rule Base
Local Policy & Rule Base
Physical Hosts (servers) Including storage, CPU, and etc.
Collaborative Clusters of routers & hosts
Global Infrastructure Layer
Resource discovery & management Cluster
Task coordinators
Host Infrastructure Layer
Host_1
Host_2
Host_m
Network Infrastructure Layer
Edge Network
Edge Network
Edge Network

Fig. 1. Infrastructure level view of a cloud provider's DCDIDP

to support; Hence, application specific detection and prevention mechanisms should be in place for each type of environments.

- **Integrated IDPS within the cloud**: IDPS could be integrated in different layers of the cloud environment (i.e. application, platform, and infrastructure) considering each layer's specifications and vulnerabilities.
- **Flexibility and elasticity**: Collaborators could be added or removed at any time and at each layer; the framework should be flexible and scale easily without losing any functionality.
- **Security strength levels**: DCDIDP should be able to apply differentiated levels of security strength to cloud customers based on their degree of abnormality.

III. Intrusion Detection and Prevention Systems

Various IDPSs have already been developed as efficient countermeasures against different system/network level attacks. Based on where detection and response occurs in the system, IDPSs can be categorized into three different types: host-based IDPSs (HIDPSs), network-based IDPS (NIDPSs), and Hybrid IDPSs (also known as Distributed IDPSs [2]) which includes both host-based and network-based sensors [6], [8]. HIDPSs are not capable of detecting attacks before the end-system is compromised. NIDPSs aim to detect intrusions inside the network. These systems are able to deal with attacks before they infiltrate the end-systems. Hybrid IDPSs provide comprehensive and complex attack detection and prevention through both network-based and host-based IDPSs. Another way to classify IDPSs is based on their deployment mechanisms [2]. They are classified into three different types: software-based, hardware-based, and VM-based [9]. VM-based IDPSs are more reliable and robust than the two other types of IDPSs because they cannot get subverted by intruders.

Misuse detection (signature based detection) and anomaly detection are two techniques that are typically used to analyze the data collected through HIDPSs, NIDPSs, or Hybrid IDPSs [10]. In misuse detection, collected data are compared to the database of the signatures of known attacks to detect intrusions. Alarms will be raised upon matching an incoming activity with those of known attackers. Misuse detection heavily depends on maintaining an up-to-date database of attack signatures that may require a significant amount of overhead. Anomaly detection is when collected data are compared to previously stored abnormal behaviors to detect suspicious activities. Alarms will be raised upon detecting a suspicious activity. Usually, normal deviations are possible threats to the system. Data mining and machine learning techniques are often used to develop a model of normal behavior. Unlike misuse detection, anomaly detection systems have the ability to deal with previously unseen or modified attacks [7]. PRBs provide each of these detection techniques with the required rules to be launched upon detection and the required policies to be considered for a proper response to the malicious activity.

IV. Distributed, Collaborative, and Data-driven IDP framework

In this section, we provide an overview of the DCDIDP framework for the cloud computing environments[1] and then discuss some of the approaches to address its components.

DCDIDP framework is comprised of three architectural levels:

1) Infrastructure level (As shown in Figure 1)
2) Platform level (As shown in Figure 2)
3) Software level (Virtual machine) (As shown in Figure 2)

The infrastructure level itself is comprised of three logical layers: network, host, and global. Figure 1 shows the infrastructure level with its logically-separated layers for one cloud provider. As shown in Figure 1, several collaborative clusters of routers and hosts can be created within the network and host infrastructure layers for each cloud provider based on metrics such as physical closeness of components, performance, etc. These clusters share and interact with three local databases for collaborative detection and prevention. These three local databases are:

- *Intrusion Assessment Information Base (IAIB)*
- *Policy and Rule Base (PRB)*
- *Audit logs*

We discuss each of these databases and their usage in the following subsections.

The network infrastructure layer, together with the host infrastructure layer, provides a comprehensive hybrid IDPS capability towards global defense. Hybrid IDPSs address the shortcoming of HIDPSs which can be easily overwhelmed in light of large-scale, traffic-intensive attacks. Hybrid IDPSs extend intrusion detection and packet filtering to the network routers by employing NIDPSs. Distributed and collaborative NIDPSs [6], [7] may be employed on the network infrastructure layer to detect and respond to the attacks effectively and in real-time. Each router within a collaborative cluster of routers and hosts interacts with each of the three local databases to detect and respond to the attacks. Furthermore, upon detecting a malicious activity, it updates all three local databases.

All the collaborative clusters on network and host infrastructure layers in Figure 1 cooperate with each other to create comprehensive global versions of the databases. For example, all the PRBs from the collaborative clusters are integrated into one global PRB. All three global databases are stored at the global infrastructure layer where the collaboration among different cloud providers occurs and they are shared and used for detection and prevention by different cloud providers. Global databases could have the same structure as their local peers but their targets are other cloud providers and their data is more complete than each of the local databases. Each cluster updates the associated global databases based on its own local databases. Global databases need to be updated periodically, which may cause overheads, or in an event-driven basis (Push, Pull or hybrid). For instance, in the hybrid updates, some attacks are prioritized and upon detecting those, mandatory update process updates the global databases.

Discovering, assigning, and integrating different services provided through a cloud provider is another responsibility of the global infrastructure layer component known as ***resource discovery & management cluster (RDMC)***. There can be several *task coordinators* within each RDMC for each cloud provider. It is assumed that task coordinators are synchronized regularly. Task coordinators are interacting with all collaborative clusters to receive the updated status on current resource availability. The aforementioned interaction involves notifying task coordinators of possible migration of customers' services based on provider's prevention policies.

There could be several physical hosts (servers) (e.g. Host_1, Host_2,..., Host_m) within each collaborative cluster of routers and hosts in a cloud provider. Each host provides varying physical resources such as CPU, storage, network, etc. to the cloud customers. Furthermore, the hosts also provide virtualized operating systems as well as various APIs to manage those physical resources.

Each host can provide its customers with IDP services in both the platform and software (virtual machine) levels as shown in Figure 2. A cloud customer can be provided with a dedicated virtual machine to run her specific applications through a cloud hypervisor. The hypervisor synthesizes one or more virtual machines by using the system's hardware. A virtual machine is the isolated duplicate of a real system. To provide software level IDP services in our framework, each cloud customer is also provided with an IDPS as an attached service to each virtual machine through the hypervisor. Therefore, each cloud customer is permitted to configure her own IDPS with her application specifics (e.g. thresholds, rules, etc.). Our framework provides DCDIDP service at the platform level (OS) of each host system as a platform level IDP service. A platform level DCDIDP service is for those customers who rent services from cloud providers to create and offer those customized services and applications to *their* own customers. In the cloud environments, the more levels of architecture we provide IDP services at, the stronger and more effective the IDP becomes. At each layer, IDPS services have access to both network-based and host-based sensors deployed at the infrastructure level. Additionally, each of the IDPSs in the VMs should report alerts to a host (platform) layer DCDIDP, which is responsible for gathering and processing the alerts from all sensors. The host DCDIDP has access to all of the local databases within the DCDIDP. The host DCDIDP interacts with all of the local databases in order to update them upon detecting a new attack and to access the updated features of prior attacks in other hosts. This way, each can collaboratively, and in a distributed fashion, detect and prevent attacks.

A. Local/Global IAIB

A key issue involved in building an adaptive, real-time defense system in a cloud environment is that of proper

[1] The poster/extended abstract version of this work has been published in 14th International Symposium on Recent Advances in Intrusion Detection (RAID'11), Menlo Park, California, September 20-21, 2011.

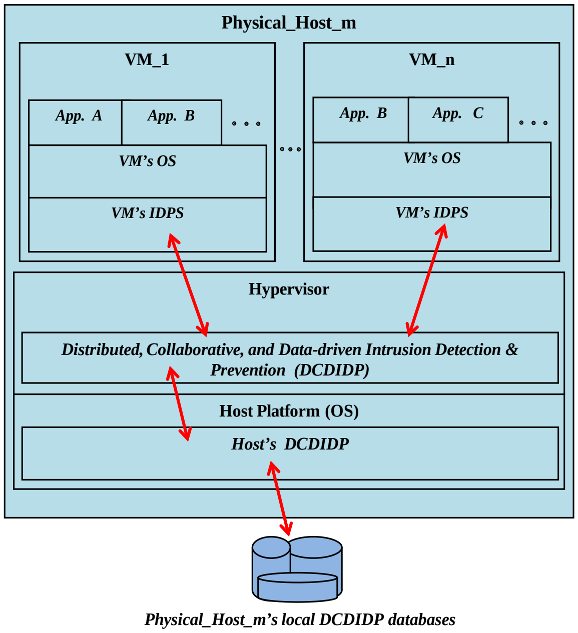
Physical_Host_m
VM_1
App. A
App. B
VM's OS
VM's IDPS
VM_n
App. B
App. C
VM's OS
VM's IDPS
Hypervisor
Distributed, Collaborative, and Data-driven Intrusion Detection & Prevention (DCDIDP)
Host Platform (OS)
Host's DCDIDP
Physical_Host_m's local DCDIDP databases

Fig. 2. Platform and Software (VM) levels' view of host_m's DCDIDP

identification of taxonomic features for general classes of intrusions in different layers (e.g. infrastructure, platform, and software). Public vulnerability databases such as BugTraq, CERT vulnerability databases, etc. contain an abundance of security-related information that can be used to generate an appropriate classification of intrusions in IAIB. Several classifications related to software vulnerability exist in the literature that can be used to derive intrusion classification for use in an adaptive, real-time defense system [11]–[13]. At the network level as well, there are taxonomies (for example [8]) that can be enhanced and used here. In particular, such a classification should emphasize the nature, *cause* (vulnerability/faults) and the *impact* of an intrusion.

Some model checking approaches to correlate intrusion alerts, generate attack trees, and fault trees have been proposed in the literature that can be extended for cloud environments [14]–[16]. However, these approaches rely on exhaustive knowledge of system states and can have serious complexity problems. Hence, the development of a methodology for dynamically generating efficient attack assessment trees (ASTs) for fine-tuning the parameters used in computation by incorporating new information related to intrusions is crucial for building the proposed IAIB for cloud environments.

An AST represents attacks against a system in a tree structure, delineating the goal of an attack as the root node and different ways of achieving that goal as leaf nodes. They can be used to provide a formal, methodical way of describing the security of systems, based on varying attacks [17]. ASTs can be employed to represent different sequence of events or activities leading to an intrusion. Such ASTs are founded on AND-OR trees, in which each node represents an attack goal. Such trees have been used by researchers to model attack strategies or to capture attack scenarios [14]–[16], [18], [19]. The root of an AST represents an ultimate attack goal and its offspring represent different activities or attack sub-goals (which may themselves be intrusions or reconnaissance attempts) that must have been collectively (AND-decomposition) or alternatively (OR-decomposition) achieved by the attacker for the major intrusion to occur. For example, Figure 3(a) depicts a simple AST for attack goal A, which has sub-goals B and C, both of which must be achieved for attack goal A to be successful. Sub-goal B is further broken down into two sub-goals X and Y, either of which needs to be accomplished for attack goal B to be reached. Hence, attack goal A can be achieved in two ways: achieve sub-goals X, B and C or Y, B and C. We can extend this assessment model to include:

1) Predictive parameters that estimate the probability of occurrence of an intrusion and its propagation time, and;
2) Important information related to each node that can assist in dynamically computing, based on real-time data, estimates of these predictive parameters to guide an effective data-driven IDP strategy in different layers of the cloud system.

The following two predictive parameters can be associated with each node in the AST:

1) $P_s(d|D)$: representing the probability of the occurrence of intrusion d given that the set of sub-intrusions D has occurred, and;
2) $T_m(d|D)$: representing the propagation time associated with the occurrence of intrusion d given that the set of sub-intrusions D has occurred.

Typically, a set of $P_s(d|D)$ and $T_m(d|D)$ values may need to be estimated for each AST node and the set of sub-intrusions that had already occurred. We believe that $P_s(d|D)$ and $T_m(d|D)$ will play a key role in guiding the real-time defense strategy against an intrusion at different layers. Computation of $P_s(d|D)$ and $T_m(d|D)$ is a significant challenge and is intensely data-driven, i.e., huge amounts of real-time monitored data and historical profiles need to be utilized to compute these values accurately. A crucial challenge is to develop simulation and estimation techniques for computing $P_s(d|D)$ and $T_m(d|D)$ values, in particular using current knowledge based on historical data and the new data obtained by processing the real-time data.

A closely-related work is *Ning et al.*'s preliminary work on generating attack scenarios through correlation of intrusion alerts [18], which uses pre-conditions and consequences (post-condition + effects) associated with the intrusions. Such an approach can facilitate both identified and anticipated attacks [18]. Our proposed IAIB tries to enrich *Ning*'s approach by incorporating more information related to the intrusion than just pre-conditions and consequences.

The proposed IAIB augments the AST with the following information: (1) system vulnerabilities derived from the current system/network/VM configuration; (2) attacker profile; (3) system state; (4) impact profile; and (5) response strategies at different levels. Identifying system vulnerabilities [20] associ-

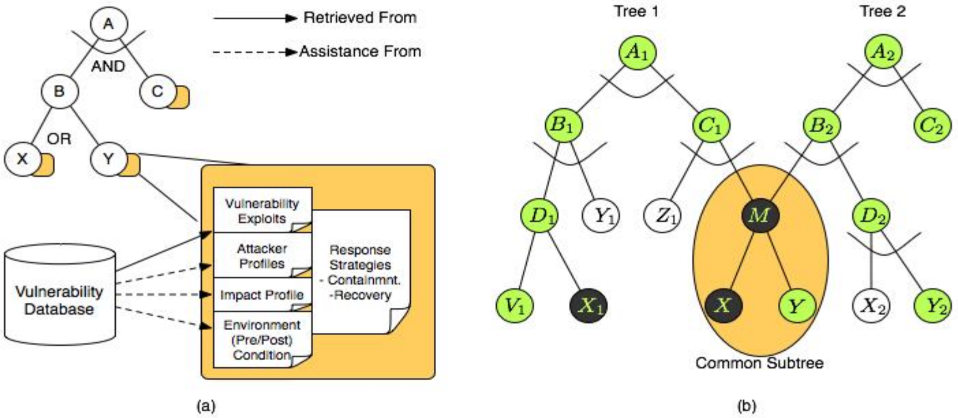

Fig. 3. (a) An AST with associated information (b) ASTs with a common sub-tree

ated with each intrusion helps to estimate the impact it will have on the system. Use of such vulnerability information for each AST node in a systematic way is essential to support real-time defense. Attackers use a variety of tools and techniques to launch an attack [21]. Attacker profiles contain information about any capabilities which we assume that an attacker may need to launch an attack. The values of $P_s(d|D)$ and $T_m(d|D)$ depend on the attacker's capability. An appropriate data-driven attacker model will facilitate recognition and classification of attacker expertise, aided by historical and real-time attack traces in all of the levels. Various factors contributing to the attacking capability [14] include:

1) Resources available (tools, funds, skills, etc.)
2) Amount of time the attacker is willing to spend.
3) Level of risk the attacker is willing to take.
4) Type of access that the attacker has to the system.
5) Motivation behind the attack (financial gain, improving hacking skills, etc.)

A set of pre-conditions captures contextual information about the system state that is required for a particular class of intrusions to occur and can indicate the vulnerability of the system. For example, if a password file is writeable then the system is highly vulnerable to privilege escalation attacks [22]. As another example, if the routers detect some sporadic legitimate packets (like ICMP broadcasts that could potentially be abused) arriving from an edge network, that edge network may be considered as a candidate for exploitation. Post-conditions indicate the system state after an intrusion has occurred. Pre-conditions and post-conditions associated with an intrusion allow us to make an appropriate assessment of the causes and impact of the intrusion, as well as to choose a proper response. Decisions as to which countermeasures are most appropriate and cost-effective against an intrusion will depend on the P_s and T_m values associated with the intrusion node.

In scenarios where two intrusion trees have a common node, such as node M in Figure 3(b), choosing a proper strategy will be more complicated. The IAIB should be designed as a generic information base to support the analysis of attacks that could be related to any architectural levels (Infrastructure, platform, or software). All the DCDIDP components at different architectural levels use and update their local IAIB. Local IAIBs update the global IAIB and the global IAIB checks for any correlations between its received data for possible AST; upon detection, it updates the local IAIBs. Furthermore, global IAIBs will be shared among different cloud providers to collaboratively detect and prevent possible attacks. Hence, local and global IAIBs are used by the DCDIDP framework to increase the accuracy, speed, and effectiveness of detection and prevention.

B. Local/Global PRB

The local PRB has both dynamic rule bases and policy bases to provide customers with a flexible capability to be able to control their systems at different architectural levels.

Within a collaborative cluster, all the components should follow the cloud provider's policies on: how to collect information, how to detect attacks, and how to respond to attacks for different attack scenarios. For instance, upon detecting DDoS attack at the software (VM) architectural level, each provider has its own policies in response to this incident. One may move the client's VM to another physical machine whereas others could easily remove/drop it. All these policies are gathered in a local PRB. Note that policies in the local PRB can be specific to the architectural level; this means that they can be set by the customers at different cloud architectural levels.

Rule bases are used within each collaborative cluster to control information flow among different architectural levels. For instance, rules could define network addresses and virtual port numbers of services that are or to be permitted or denied.

The collaborative cloud providers share their rule bases in the global PRB. Each collaborator also adds associated policies for each of its rules into the global PRB. This way the collaborators could use the policies in the global PRB as recommended policies for their local rules.

Usually, there are multiple administrators at different architectural levels who are adding, removing, or making changes on local/global PRBs. Consequently, local/global PRBs often become large and complicated. Therefore, we need to have a mechanism in place at both local and global levels of PRBs to automatically remove those rules and policies that are either partially or completely unused for a certain period of time, or expired. This mechanism merges same rules and adds all of the different policies (together with their frequency of usage) from all the collaborators as that rule's policies.

C. Local/Global Audit logs

Extensive logging about detected events is usually part of each IDPS. An IDPS's logged data can be used to confirm the validity of alerts, investigate possible incidents, and correlate various events of other IDPSs' logging sources [10]. Some of the common data fields used among various types of IDPSs to log are as follows [10]:

- Event type
- Event date and time
- Event importance rate (e.g., priority, severity, impact, confidence)
- Prevention action performed

There are additional data fields specifically for each type of IDPSs (e.g. NIDPS, HIDPS, Hybrid-IDPS). For instance, NIDPSs perform packet capturing (e.g. selective deep-packet-inspection) and HIDPSs record details related to a specific event such as: IP address and port information, application information, and user IDs. Local audit log databases in our framework receive copies of locally stored logs (e.g. Syslog, etc.) from all of the local hosts and routers within the collaborative clusters. IDPSs in different architectural levels detect different events. Hence the local audit log is comprised of all details of the architectural-level specific events' logs gathered at different layers. For instance, Syslog output gathered at the VM layer provides DCDIDP systems with insight into virtual network activity. Finally, all the local audit logs update the global audit log to provide the required log data for collaborative attack detection and prevention among different cloud providers.

D. DCDIDP's Intrusion Detection Capability & Response Mechanisms

Large scale coordinated attacks targeting different architectural levels of cloud environments can be expected to be of significant complexity and sophistication, particularly in light of the available tools and the damage that can be inflicted at each level. While many attacks may be detected at the infrastructure level (e.g. network or hosts), sophisticated attacks aimed at platform and software levels may be difficult to detect by infrastructure level approaches. Detecting such a sophisticated attack will require correlating events and patterns of activities distributed vertically (across architectural levels of the cloud) and horizontally across the entire cloud environment and among different collaborative cloud providers. A key approach is to correlate events and patterns of activities from all architectural levels, and among the collaborative cloud providers, to evaluate the effects of the intrusion on all architectural levels and providers. For example, simultaneous intrusion alerts may be generated by a router as well as by a host-based IDS system (e.g., a database intrusion detector), indicating a carefully orchestrated attack. Furthermore, coordination and correlation of IDPSs deployed at different architectural levels (i.e., infrastructure, platform, and software) to fine-tune the detection capability has not been a feature of different IDPSs already proposed in the literature. We believe that the effect of an attack can generically be observed at the different architectural levels; hence, correlating the activities of the detectors at these levels can significantly increase the system-wide intrusion detection capability. For instance, a positive indication of an impending/ongoing attack at the software (VM) level may be used to indicate to the NIDPSs (i.e. infrastructure level) to focus on a particular traffic pattern to confirm an impending attack and/or to stop/mitigate the attack.

Upon detecting intrusions, there should be several methods in place to effectively respond to such intrusions in a timely manner and to maintain an acceptable level of system functionality in the presence of a key challenge towards this is to first design techniques to construct intrusion boundaries that form the peripheries of system-wide functionalities that are adversely affected by an intrusion. Subsequently, various methodologies can be used to respond to the intrusions aimed at controlling the spread of the damage and to prevent future intrusions by adapting to new forms of intrusion scenarios. Choosing appropriate response techniques for handling ongoing intrusions and applying recovery techniques are challenging tasks, particularly because of the real-time constraints. High latencies in detection and assessment processes can spread the damage within a system at an unknown rate.

At the time any malicious activity is detected, the system should take measures to ensure that the potential intrusion does not damage the system's functional components and the critical functions remain operational at an acceptable level of trust (LoT). Such a response should effectively react to the problem in a timely manner. Ideally, the impact of an attack should be minimal and after the intrusion period is over, a longer term recovery mechanism can be employed. How to respond to an intrusion will depend on the cloud provider's policies and its service priorities. For example, in some applications, maintaining high LoT in available functionality may be the primary goal. In other cases, availability may be the critical intent although at a reduced LoT. In particular, two intrusion response techniques that may be applied are intruder isolation and/or damage containment, as shown in Figure 4. The module will be referred to as the distributed, collaborative response & recovery module.

One way of achieving intruder isolation is through popular feature of the cloud environment; moving the intruder to a separate virtual machine so she is in a controlled environment. Different versions of objects (e.g., a copy of the database containing a copy of items the user needs) can be created to isolate each suspected malicious user. If the suspected user turns out to be benign, the isolated objects on each VM (possibly modified by the transactional activities that the

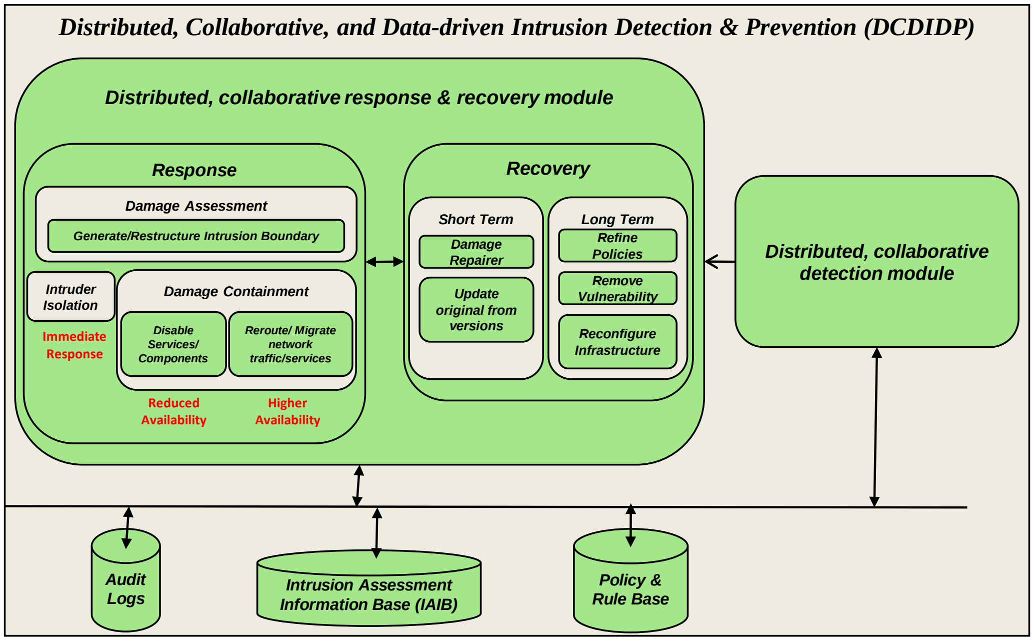
Distributed, Collaborative, and Data-driven Intrusion Detection & Prevention (DCDIDP)
Distributed, collaborative response & recovery module
Response
Damage Assessment
Generate/Restructure Intrusion Boundary
Intruder Isolation
Immediate Response
Damage Containment
Disable Services/ Components
Reroute/ Migrate network traffic/services
Reduced Availability
Higher Availability
Recovery
Short Term
Damage Repairer
Update original from versions
Long Term
Refine Policies
Remove Vulnerability
Reconfigure Infrastructure
Distributed, collaborative detection module
Audit Logs
Intrusion Assessment Information Base (IAIB)
Policy & Rule Base

Fig. 4. Detection, and response & recovery modules of DCDIDP framework

suspected user has carried out) can be merged with the original objects. In other words, the changes in isolated databases can be used to update the original database. At the network level, resource allocation and fairness mechanisms can be exploited to provide some level of isolation. Upon detection of malicious packets, using a traceback method to find the malicious source of attack, effectively isolating attack packets could be utilized.

Damage containment may involve disabling services and access to objects/resources that may be already compromised. Once the affected components are protected from access, the system may need to re-route future transactions or packets in order to increase availability of critical services. For example, if it has been determined that the customer's data has been damaged by an attack, then it is necessary to bar any transaction that uses the customer database or remove the customer's VM from the host system. However, if there is a mirror/backup site containing the customer database, it may be used instead or customer's VM may be migrated to another system. Similarly, zombies in a botnet can be disabled and/or packets from a particular IP address space may be blocked based upon traceback until the attacks are stopped.

For either of the response techniques, we need to designate the boundary indicating the set of objects/resources which either have been used by a malicious user (in case of intruder isolation) or have potentiality been infected (in case of damage containment). We use the term intrusion boundary (IB) to refer to either the isolation boundary or the damage containment boundary. The purpose of IB is to ensure that the effect of the intrusion is isolated and does not propagate to any system outside of the boundary. The intrusion boundary is different at each architectural level. For instance, in the software level, a customer's VM will be considered as their IB; at the platform level, all the resources that are provided through a specific host system will be the IB of that platform's customer.

The IBs in the system are designed based on inputs from the detection module and the control parameters such as the desired LoT and availability. Once the detection module identifies the malicious set of transactions and the current level of activity, we can estimate the propagation rate of the damage as well as the detection and assessment latencies based on the timing information related to the set of suspicious events or transactions. In particular, the data and transaction dependency graphs will be used to identify the possible IB [23]. Typically, in environments requiring a high-level of trust maintained at all times, we may need to take a pessimistic approach in estimating the IB such that the possibility of further propagation is minimized. Thus, trust levels may be maintained at the cost of reducing the degree of availability. Such a pessimistic approach needs to take into account both the detection and assessment latencies.

V. Trust Management Among Collaborative Cloud Providers

As previously discussed, collaboration is at the center of the DCDIDP framework. The collaborative clusters cooperate with each other to create a global version of the databases used for detection and prevention by all collaborating cloud providers. However, to enable cloud service providers to be able to collaborate and share information with ease of mind, trust management should be taken into account and a trust mechanism needs to be developed. The interactions among different cloud service providers (driven by service requirements) are expected to be very dynamic/transient and intensive. There are some critical questions that need to be answered: Do collaborative cloud service providers trust each other? How can they negotiate the trust? Is the trust static/dynamic? How do we manage and maintain dynamically changing trust values and adapt the access requirements as trust evolves?

Existing trust negotiation mechanisms primarily focus on credential exchange and do not address a more challenging need to integrate requirements-driven trust negotiation techniques with fine-grained access control mechanisms [24]. One possible approach is to develop a trust mechanism to support the establishment, negotiation and maintenance of trust based on inter-domain service requirements. It should efficiently capture a generic set of parameters required for establishing trust and to manage evolving trust and interaction/sharing requirements [25].

VI. Related Work

In this section, we provide a brief overview of different IDPSs that are either integrate/apply current IDPSs for cloud environments or use cloud-specific architectures. IDPS for cloud environments is a new field of research which is growing in importance due to the growing popularity of cloud services and the increasing number of attacks targeting both cloud services and cloud infrastructure (i.e. originating from inside a cloud computing infrastructure, exploiting it as an infrastructure for deploying attacks). There are few research papers in the current literature on this topic. There are similarities between Grid and Cloud environments. Some researchers highlight these similarities and use the same IDP solutions already proposed for Grid environments in the cloud infrastructure. These approaches are mainly aimed at defining a new IDPS model that can take advantage of additional information provided by the cloud infrastructure itself. For example, in [26] a distributed IDPS for cloud environments is designed to provide services at the platform level (PaaS), and is structured as an added service of the cloud systems' infrastructure.

In [27], an architecture (CloudSEC) for composing collaborative security-related services in the cloud, such as correlated intrusion analysis, anti-spam, anti-DDOS, automated malware detection and containment is proposed. It uses a peer-to-peer overlay hierarchy to allow services to be integrated into dynamically scheduled tasks with adequate data and computation resource provision. The main goal of CloudSec is to move the analysis and correlation of generalized network alerts from centralized systems into the network cloud and to provide *Security as an In-cloud Service* [27].

Authors of [5] integrate a currently available NIDS into an open source cloud computing environment to address the challenges in detecting Denial of Service (DoS) attacks, performed by means of resources acquired on-demand on a cloud computing platform. They study the consequences of using a distributed strategy to detect and respond to attacks that were initiated by misbehaving customers of a cloud provider. Their preliminary results validate the approach's performance. Their performance evaluations look at employing their approach either close to the cluster controller or next to the physical machines.

Applying mobile agents to provide intrusion detection capability in cloud applications (regardless of location) is proposed in [4]. The authors' goal is to provide scalable, flexible and cost-effective IDS for the cloud environment but they also believed that their approach lack robustness due to insufficient knowledge sharing among mobile agents. They suggest that the research community look into mobile agents intercommunication and negotiation, which can help investigative mobile agents to share their knowledge and therefore build a more robust inter-cloud IDS.

In [28], an autonomic mechanism for anomaly detection in cloud computing environment has been proposed. Authors present a set of techniques to automatically analyze collected data. Their approach transforms data into a uniform format for analysis, extracts features to reduce data size, and learns in an unsupervised fashion to detect the nodes acting differently from others. They evaluate their approach by implementing a prototype on an institute-wide compute cloud environment. Their results showed that their mechanism can effectively detect faulty nodes with high accuracy and low computation overhead.

To the best of our knowledge, our framework is the first comprehensive distributed, collaborative, data-driven IDP framework proposal to integrate IDP at all architectural levels of cloud environments while considering each level's specific requirements.

VII. Conclusions & Future Work

Although security and privacy issues are delaying adoption of cloud computing, it has already become an unstoppable force and we need to provide security mechanisms to ensure its secure adoption. In particular, intrusion detection and prevention systems are the main focus of this paper. We have proposed DCDIDP, a distributed, collaborative, and data-driven intrusion detection and prevention system. All of the cloud service providers that use DCDIDP collaborate in a distributive manner at different levels of operations to respond to attacks and to provide holistic IDPSs. We describe the framework at the infrastructure, platform, and the software levels, and explore how cloud service providers collaborate in order to perform intrusion detection and prevention, and identify various challenges in realizing the framework.

We are currently working on implementing a prototype of the proposed framework in order to show its applicability, using real world cloud service providers.

Acknowledgment

The authors would like to thank anonymous reviewers for their insightful comments and feedback. This work has been supported by NSF award CCF- 0720737.

References

[1] NIST SP 800-145, *The NIST Definition of Cloud Computing*, http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145-cloud-definition.pdf, NIST Special Publication 800-145, January 2011.

[2] S. Roschke, F. Cheng, and C. Meinel, *Intrusion Detection in the Cloud*, Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC'09), pp.729–734, China, 2009.

[3] D. Song, L. Wei-min, Z. Tao, and Y. Yong, *Distributed Proactive Defense based on Cloud Computing*, In International Conference on Intelligent Computing and Integrated Systems (ICISS), pp. 95–98, Guilin, China, October 22-24, 2010.

[4] A. V. Dastjerdi, K. A. Bakar, and S. G. H. Tabatabaei, *Distributed Intrusion Detection in Clouds Using Mobile Agents*, In Third International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP'09), pp. 175–180, Sliema, October 11-16, 2009.

[5] C. Mazzariello, R. Bifulco, and R. Canonico, *Integrating a Network IDS into an Open Source Cloud Computing Environment*, In Sixth International Conference on Information Assurance and Security (IAS'10), pp. 265–270, Atlanta, GA, August 23-25, 2010.

[6] S.T. Zargar, and J. Joshi, *A Collaborative Approach to Facilitate Intrusion Detection and Response against DDoS Attacks*, 6th Intl Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom'10), Chicago, IL, October 9-12, 2010.

[7] T. Znati, J. Amadei, D. R. Pazehoski, and S. Sweeny *On the Design and Performance of an Adaptive, Global Strategy for Detecting and Mitigating Distributed DoS Attacks in GRID and Collaborative Workflow Environments*, Simulation, vol. 83, no. 3, pp. 291–303, 2007.

[8] J. Mirkovic, and P. Reiher, *A taxonomy of DDoS attack and DDoS defense mechanisms*, ACM SIGCOMM Computer Communications Review, vol. 34, no. 2, pp. 39–53, 2004.

[9] M. Laureano, C. Maziero, and E. Jamhour, *Protecting host-based intrusion detectors through virtual machines*, Computer Networks, vol. 51, no. 5, pp. 1275–1283, April 2007.

[10] NIST SP 800-94, *Guide to Intrusion Detection and Prevention Systems (IDPS)*, http://csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf, NIST Special Publication 800-94, February 2007.

[11] S. Jin, Y. Wang, X. Cui, and X. Yun *A review of classification methods for network vulnerability*, in Systems, Man and Cybernetics (SMC'09), San Antonio, TX, October 11-14, 2009.

[12] M. Gegick, and L. Williams *Matching attack patterns to security vulnerabilities in software-intensive system designs*, SIGSOFT Softw. Eng. Notes, Vol. 30, no. 4, pp. 1–7, May 2005.

[13] H. Shahriar, and M. Zulkernine *Taxonomy and classification of automatic monitoring of program security vulnerability exploitations*, Journal of Systems and Software, Vol. 84, no. 2, pp. 250–269, February 2011.

[14] P. Moore, R. J. Ellison, and R. C. Linger *Attack Modeling for Information Security and Survivability*, CMU/SEI-2001-TN-001, Software Engineering Institute, Carnegie Mellon University, March 2001.

[15] B. Schneier *Attack Trees*, Secrets and Lies, John Wiley and Sons, pp. 318–333, New York, 2000.

[16] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing *Automated Generation and Analysis of Attack Graphs*, IEEE Symposium on Security and Privacy, IEEE Computer Society, 2002.

[17] B. Schneier *Modeling security threats*, http://www.schneier.com/paper-attacktrees-ddj-ft.html

[18] P. Ning, Y. Cui, and D. S. Reeves *Constructing Attack Scenarios through Correlation of Intrusion Alerts*, 9th ACM conference on Computer and communications security, November 2002.

[19] W. Yu-Sung, B. Foo, Y. Mei, and S. Bagchi *Collaborative intrusion detection system (CIDS): a framework for accurate and efficient IDS*, 19th Annual Computer Security Applications Conference, pp. 234–244, December 2003.

[20] NIST SP 800-30, *Risk Management Guide for Information Technology Systems*, http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf, NIST Special Publication 800-30, July 2002.

[21] J. B. D. Joshi, A. Ghafoor, W. Aref, and E. H. Spafford *Digital Government Security and Privacy Challenges*, William J. McIver, Jr. and Ahmed K. Elmagarmid (eds) Advances in Digital Government: Technology, Human Factors, and Policy, Boston, Kluwer, Chapter 7, pp. 121-136, 2002.

[22] M. Bishop *A Taxonomy of Unix System and Network Vulnerabilities*, Technical Report CSE-9510, Department of Computer Science, University of California at Davis, May 1995.

[23] Y. Liu, F. L. Presti, V. Misra, D. Towsley, and Y. Gu *Scalable Fluid Models and Simulations for Large Scale IP Networks*, ACM Transactions on Modeling and Computer Simulation, Vol. 14, No. 3, 2004.

[24] H. Takabi, J.B.D. Joshi, and G.-J. Ahn, *SecureCloud: Towards a Comprehensive Security Framework for Cloud Computing Environments*, In Proc. of the 1st IEEE International Workshop Emerging Applications for Cloud Computing (CloudApp 2010), pp. 393-398, Seoul, South Korea, 2010.

[25] H. Takabi, J. B. D. Joshi, and G. J. Ahn, *Security and Privacy Challenges in Cloud Computing Environments*, IEEE Security and Privacy, Vol. 8, No. 6, pp. 25-31, 2010.

[26] K. Vieira, A. Schulter, C. B. Westphall, and C. M. Westphall, *Intrusion Detection for Grid and Cloud Computing*, In IT Professional, Vol. 12, no. 4, pp. 38–43, July-Aug. 2010.

[27] J. Xu, J. Yan, L. He, P. Su, and D. Feng, *CloudSEC: A Cloud Architecture for Composing Collaborative Security Services*, In IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom'10), pp. 703–711, Indianapolis, IN, Nov. 30- Dec. 3, 2010.

[28] D. Smith, Q. Guan, and S. Fu, *An Anomaly Detection Framework for Autonomic Management of Compute Cloud Systems*, In IEEE 34th Annual Computer Software and Applications Conference Workshops (COMPSACW), pp. 376–381, Seoul, Japan, July 19-23, 2010.