Infrastructure as a Service Security: Challenges and Solutions

DATASET · JANUARY 2013

CITATIONS
7

READS
710

3 AUTHORS, INCLUDING:

Wesam Dawoud
Hasso Plattner Institute
16 PUBLICATIONS **91** CITATIONS
SEE PROFILE

Ibrahim Takouna
Hasso Plattner Institute
19 PUBLICATIONS **81** CITATIONS
SEE PROFILE

Infrastructure as a Service Security: Challenges and Solutions

Wesam Dawoud [#1], Ibrahim Takouna [*2], Christoph Meinel [#3]

[#] *Hasso Plattner Institute*
Potsdam, Germany
[1] wesam.dawoud@hpi.uni-potsdam.de
[3] meinel@hpi.uni-potsdam.de

[*] *Ministry of Education & Higher Education*
Palestine
[2] itakouna@gmail.com

***Abstract*—Cloud Computing represents a new computing model that poses many demanding security issues at all levels, e.g., network, host, application, and data levels. The variety of the delivery models presents different security challenges depending on the model and consumers' Quality of Service (QoS) requirements. Confidentiality, Integrity, Availability, Authenticity, and Privacy are essential concerns for both Cloud providers and consumers as well. Infrastructure as a Service (IaaS) serves as the foundation layer for the other delivery models, and a lack of security in this layer will certainly affect the other delivery models, i.e., PaaS, and SaaS that are built upon IaaS layer. This paper presents an elaborated study of IaaS components' security and determines vulnerabilities and countermeasures. Finally, as a result of this research, we propose a Security Model for IaaS (SMI) to guide security assessment and enhancement in IaaS layer.**

I. Introduction

Software, Platform, and Infrastructure as a Service are the three main service delivery models for Cloud Computing. Those models are accessible as a service over the Internet. The Cloud services are made available as pay-as-you-go where users pay only for the resources they actually use for a specific time, unlike traditional services, e.g., web hosting. Furthermore, The pricing for cloud services generally varies according to QoS requirements [1]. The cloud deployment models, based on their relationship to the enterprise, are classified to private, public, and hybrid. Public Cloud services are sold as Utility Computing, while private Cloud refers to internal datacenters of an enterprise which are not available to the general public. Examples of emerging Cloud Computing Platforms include Microsoft Azure[1], Amazon EC2[2], and Google App Engine[3].

The confusion between Cloud and Service Oriented Architecture (SOA) has prompted us to discuss this issue and offer a brief comparison between them. SOA and Cloud Computing can be considered complementary services sharing common characteristics. Hence, if SOA is a set of principles and methodologies designed to facilitate systems integration and communication regardless of development languages and platforms, Cloud Computing, on the other hand, is designed to enable companies to utilize massive capacities instantly without having to invest into new infrastructure, train new staff, or license new software. Cloud Computing allows small and medium-sized businesses to completely outsource their datacenter infrastructure, as well as large companies that need huge load capacities without building larger expensive datacenters internally. Cloud Computing employs the virtualization technology to offer a secure, scalable, shared, and manageable environment. In short, regardless of the difference in designing purposes and the dependency of Cloud Computing on virtualization technology, Cloud Computing might intersect with SOA in Components as a Service, e.g., SOA via Web Service standards. Therefore, Cloud Computing and SOA can be pursued independently, or concurrently as complementary activities to provide an outstanding business.

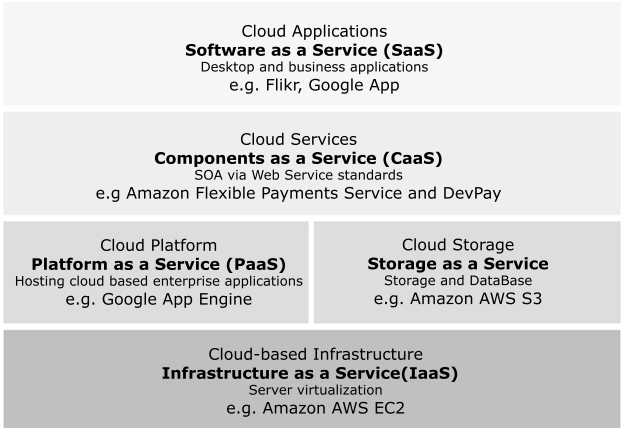

Fig. 1. Cloud delivery models

Cloud Computing depends primarily on IaaS layer to provide cheap and pay-as-you-go processing power, data storage, and other shared resources. This paper presents a detailed and precise study of IaaS security and privacy concerns. We have investigated security for each IaaS component: Service Level Agreement (SLA), Utility Computing (UC), Platform Virtualization, Networks & Internet Connectivity, and Computer

[1]http://www.microsoft.com/windowsazure/
[2]http://www.amazon.com/ec2
[3]http://code.google.com/appengine/

Hardware. Furthermore, Cloud software's security that impact on IaaS and on the whole Cloud Computing is presented. We are interested in the IaaS delivery model because it is the foundation of all other delivery models, and a lack of security in this layer affects the other delivery models that are built upon IaaS layer.

The rest of the paper is organized as follows. Section II analyzes IaaS components vulnerabilities and security challenges in addition to the recent proposed solutions. Section III presents a Security Model for IaaS (SMI), and how that model could enhance security for IaaS against the vulnerabilities of the current used technologies. Finally, Section IV concludes the paper and outlines areas for future work.

II. IaaS Components

IaaS delivery model consists of several components that have been developed through past years, nevertheless, employing those components together in a shared and outsourced environment carries multiple challenges. Security and Privacy are the most significant challenges that may impede the Cloud Computing adoption[4]. Breaching the security of any component impact the other components' security, consequently, the security of the entire system will collapse. In this section we study the security issue of each component and discuss the proposed solutions and recommendations.

A. Service Level Agreement (SLA)

Cloud Computing emerges a set of IT management complexities, and using SLA in cloud is the solution to guarantee acceptable level of QoS. SLA encompasses SLA contract definition, SLA negotiation, SLA monitoring, and SLA enforcement [2]. SLA contract definition and negotiation stage is important to determine the benefits and responsibilities of each party, any misunderstanding will affect the systems security and leave the client exposure to vulnerabilities. On the other hand, monitoring and enforcing SLA stage is crucial to build the trust between the provider and the client. To enforce SLA in a dynamic environment such Cloud, it is necessary to monitor QoS attributes continuously [2]. Web Service Level Agreement (WSLA) framework [3] developed for SLA monitoring and enforcement in SOA. Using WSLA for managing SLA in Cloud Computing environment was proposed in [4] by delegating SLA monitoring and enforcement tasks to a third-party to solve the trust problem. Currently, cloud clients have to trust providers' SLA monitoring until standardizing Cloud Computing systems and delegating third-parties to mediate SLA monitoring and enforcement.

B. Utility Computing

Utility Computing is not new concept; it played an essential role in Grid Computing deployment. It packages the resources (e.g., computation, bandwidth, storage, etc...) as metered services and delivers them to the client. The power of this model lies in two main points: First, it reduces the total cost, i.e., instead of owning the resources, client can only pay for usage time (pay-as-you-go). Second, it has been developed to support the scalable systems, i.e., as an owner for a rapid growing system you need not to worry about denying your service according to a rapid increase of users or reaching peak in demand. Obviously, Utility Computing shapes two of the main features of the Cloud Computing (e.g., scalability, and pay-as-you-go). The first challenge to the Utility Computing is the complexity of the Cloud Computing, for example, the higher provider as Amazon must offer its services as metered services. Those services can be used by second level providers who also provide metered services. In such multiple layers of utility, the systems become more complex and require more management effort from both the higher and the second level providers. Amazon DevPay[5], an example for such systems, allows the second level provider to meter the usage of AWS services and bill the users according to the prices determined by the user. The second challenge is that Utility Computing systems can be attractive targets for attackers, so an attacker may aim to access services without paying, or can go further to drive specific company bill to unmanageable levels. The provider is the main responsible to keep the system healthy and well functioning, but the client's practice also affects the system.

C. Cloud Software

There are many open source Cloud software implementations such as Eucalyptus [5] and Nimbus [6]; Cloud software joins the cloud components together. Either Cloud software is open source or commercial closed source. We can't ensure the vulnerability and bugs in available software, furthermore, cloud service providers furnish APIs (REST, SOAP, or HTTP with XML/JSON) to perform most management functions, such as access control from a remote location [6]. For example, client can use the Amazon EC2 toolkits, a widely supported interface, to consume the services by implementing own applications or by simply using the web interfaces offered by the provider. In both cases, user uses web services protocols. SOAP is the most supported protocol in web services; many SOAP-based security solutions are researched, developed, and implemented [7]. WS-Security, a standard extension for security in SOAP, addresses the security for web services. It defines a SOAP header (Security) that carries the WS-Security extensions and determines how the existing XML security standards like XML Signature and XML Encryption are applied to SOAP messages. Well known attacks on protocols using XML Signature for authentication or integrity protection [8] would be applied to web services consequently affecting the Cloud services. Finally, an extreme scenario in [9] showed the possibility of breaking the security between the browser and the clouds, and followed by proposal to enhance the current browsers security. Indeed, these attacks belong more to the web services world, but as a technology used in Cloud Computing, web services' security strongly influences the Cloud services' security.

[4]IDC Enterprise Panel, August 2008.

[5]http://aws.amazon.com/devpay/

[6]http://workspace.globus.org/clouds/nimbus.html

D. Platform Virtualization

Virtualization, a fundamental technology platform for Cloud Computing services, facilitates aggregation of multiple standalone systems into a single hardware platform by virtualizing the computing resources (e.g., network, CPUs, memory, and storage). Hardware abstraction hides the complexity of managing the physical computing platform and simplifies the computing resources scalability. Hence, virtualization provides multi tenancy and scalability, and these are two significant characteristics of Cloud Computing.

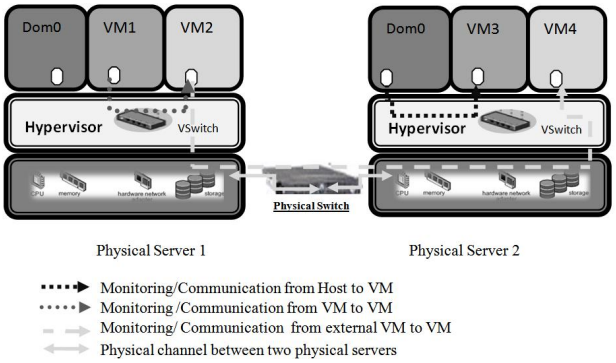

Fig. 2. The different types of interactions between VMs themselves and Host

As the hypervisor is responsible for VMs isolation, VMs could not be able to directly access others' virtual disks, memory, or applications on the same host. IaaS, a shared environment, demands an accurate configuration to maintain strong isolation. Cloud service providers undertake a substantial effort to secure their systems in order to minimize the threats that result from communication, monitoring, modification, migration, mobility, and DoS. In this section, we discuss virtualization risks and vulnerabilities that affect particularly IaaS delivery model in addition to the recent proposed solutions to guarantee security, privacy, and data integrity for IaaS.

1) Security threats sourced from host: The threats sourced from host (i.e., privilege domain) are result from monitoring, communication, or modification processes to VMs. These threats and the proposed solutions are discussed as follows:

Monitoring VMs from host : Monitoring is considered an important procedure which includes control actions (e.g., start, shutdown, pause, restart the VMs), and VMs' resources modification. Unfortunately, the sysadmin or any authorized user who has privileged control over the backend can misuse this procedure. Xenaccess [10] is a tool allows sysadmin to run a user level process in Dom0 (i.e., a privileged domain in Xen) to access the memory of a customer's VM at run time. Here, it is important to know that Xenaccess is developed to run on Xen which was adopted by some of the initiative Cloud Computing providers (e.g. Amazon EC2 and Citrix[7] are Xen-based).

Communications between VMs and host: Communications between VMs and host flow between VMs through shared virtual resources (e.g., virtual network). Fig. 2 shows that all network packets coming from or going to a VM pass through the host, so the host is generally able to monitor network traffic of its hosted VMs. Attackers might exploit some useful features in virtual machine such as shared clipboard that allows data to be transferred between VMs and the host to exchange data between cooperating malicious program in VMs [11]. Nevertheless, the worst case occurs when a host is compromised, this puts all VMs in risk. After discussing the threats that originate from a host, we present and discuss the proposed solution that prevent or mitigate these threats and vulnerabilities.

Terra [12] is an architecture presenting closed box execution environment for VMs to be protected from a user with full privileges (e.g., sysadmin), so VMs would not be inspected or modified by another VM running on the same platform even by a user with full privileges. Unfortunately, Terra is not suitable to be deployed in a complex dynamic environment like IaaS which comprises several hundreds of machines networked together. In IaaS environment, VMs are created and scheduled to dynamically run. Furthermore, serving huge number of consumers turns IaaS more vulnerable and less trusted.

To overcome the drawbacks in traditional trusted platforms such as Terra, Trusted Virtual Datacenter (TVDc) [13],[14] technique is proposed to addresses both infrastructure and management security issues. (TVDc) manages the security in datacenter virtualization by enforcing a control access schemes to the networked storage based on security labels and by implementing management prototypes that demonstrate the enforcement of isolation constraints and integrity checking.

Similarly, Trusted Cloud Computing Platform (TCCP) [15] is proposed for ensuring the confidentiality and integrity of computations that are outsourced to IaaS services, the TCCP provides the abstraction of a closed box execution environment for a customer's VM. Like Terra, TCCP prevents the privileged administrator from inspecting or tampering VMs contents, on other words, TCCP is a solution for inside attacks that gain full privilege on systems. TCCP allows a customer to reliably and remotely determine whether the service backend is running in trusted environment before requesting the service to launch a VM. Moreover, this capability extends the notion of attestation to the entire service, and thus allows a customer to verify if its computation will run securely. Unfortunately, TCCP authors have not implemented a fully functional prototype to evaluate the performance of TCCP yet. Unlike Terra, TCCP and TVDc are proposed to work in complex environment to provide IaaS service.

Using VLANs [16] to strengthen network isolation and enhance systems management capabilities was implemented by TVDs [13] and [14]. However, [17] described a technique to strengthen grid security by using Trusted Platform Module (TPMs) [18],[19]. TPMs was proposed by The Trusted Computing Group (TCG) to provide cryptographic credentials, remote attestation, and integrity protection. It also could be employed in Cloud Computing to enable remote attestation like in Trusted Platforms [20] and [21]. Reference [22] pro-

[7]http://www.citrix.com

posed a VM verifier technique for building comprehensive runtime integrity proofs for general purpose systems in distributed computing systems. This technique is for enforcing integrity on applications that based on VMs and reporting their integrity to remote VM verifier (VMV).

Practically, to improve security in Xen, a lot of effort has been done, and we summarize the significant contributions as follows: First, "Improving Xen Security through Disaggregation" [21] was proposed to strengthen the Trusted Computing Base (TCB) of a Xen-based system where Xen's TCB cannot guarantee the integrity of a virtual machine, and controlling dom0 gives a full control over the other domains. Removing Dom0 user-space from the TCB to enforce local closed box protection would improve Xen's TCB security against a malicious sysadmin [21]. Disaggregated Xen-based system was evaluated for confidentiality and integrity and against two sources of attacks: the administrator of the physical platform and the other unprivileged VMs on the same physical host. Additionally, an inter-VM communication technique is developed to bypass networking stack during communicates between Xen guest VMs. Second, sHype [23], implemented to the Xen hypervisor, provides a security reference monitor interface in the hypervisor to enforce information flow constraints between virtual machines. It improves hypervisor resource-level isolation to include access control on virtual resources by labeling the VMs into groups and controls the access to the local resources (e.g. local VLAN and virtual disks) and the distributed shared virtual resources (e.g., VLANs which spanning multiple hypervisors) by enforcing a Mandatory Access Control (MAC) policy. Finally, Intel proposed "LaGrange Technology (LT)" to Enhance Xen Security [24]. LT provides a protected memory for better isolation with enhanced Xen I/O spaces, protected launch anchors Xen start-up in hardware with fewer elements in the trust chain, and protected peripherals (e.g., graphics, keyboard, and mouse) to build blocks for enhanced trusted path.

2) Security threats sourced from other VM: In this section, we present threats that result from communication, monitoring, and modification of VM by another VM or by external machine. As Dom0 provides backend/frontend I/O paravirtualized architecture [25] , the backend driver runs in privileged domain (usually Dom0), and frontend drivers run in unprivileged domain (DomU), accordingly, although packets sniffing and IP spoofing are common attacks in conventional networks, they can't risk the confidentiality or integrity of the network in virtual environment when they run from unprivileged VM.

Monitoring VMs from other VM: As mentioned earlier, monitoring VMs could violate security and privacy, but the new architecture of CPUs, integrated with a memory protection feature, could prevent security and privacy violation. The hypervisor uses this to prevent a VM from monitoring the other VMs memory resources, and access other VMs' virtual disks allocated in the host. On the other hand, physical networking machines are connected by physical dedicated channel. However, in virtual networking, VMs are linked to the host machine by a virtual switch. Unfortunately, in both cases, packets sniffing and ARP poisoning could be occurred between machines. An encryption scheme, such as transport layer security (TLS) [26] or secure Internet protocol (IPSec) [27] would be used to protect confidentiality. To modify packet data, it would be necessary to modify the Dom0 kernel code that controls the software bridge. As kernel code is part of the TCB , any modification reflects integrity measurements. However, encryption provides an additional defense against attacks on the network integrity, and using Virtual Private Networking [28] software in the guest VM would be sufficient to protect the confidentiality and integrity of the network from Dom0-admin. From these scenarios, we notice that the protection given to virtual network connections is equivalent to physical network when connected to an untrusted network. However, using traditional IDS tools [29],[30] in conventional networks would solve such these problems, but using them in cloud environment would not be an appropriate solution to identify suspicious activities due to the characteristics of cloud such as dynamicity, Self-service, and self-managed platform.

Communication between VMs: The threats against the communication between VMs depend on how those machines will be deployed (e.g., Sharing a physical computer between multiple organizations). Sharing resources between VMs might expose security of each VM, for instance, collaboration using some applications such as a shared clipboard allows data to be transferred between VMs and the host assisting malicious programs in VMs to exchange data by which violate security and privacy. A malicious VM can potentially access other VMs through shared memory, network connections, and any other shared resources without compromising the hypervisor layer. The critical risks of such network environments motivated researchers to provide protecting solutions and techniques for securing communication between VMs.

First, TVDc technique in [13] was extended in [14] to provide customer workloads isolation from each other to prevent data leakage, thus, it prevents VMs from spreading viruses and other malicious. Additionally, to prevent or mitigate the incidence of failed configuration management tasks, [14] proposed an isolation management policy for competing datacenter workloads and a continuous audit for dynamic cloud environment.

Second, [31] proposed an IDS for Grid Computing and Cloud Computing environment. The proposed approach applies two intrusion detection techniques to the collected data from the cloud: I- behavior-based method to verify user's actions correspond to known behavior profiles. II- knowledge-based method to verify security policy violations and known pattern attacks. However, according to implemented prototype results, applying the both techniques together achieved a higher level of security and a lower rate of false positives and negatives. Unfortunately, this approach works for intrusion at the middleware layer only (i.e., PaaS), and it deserves a more detailed investigation to be extended to IaaS.

Third, a Security virtual machine (SVM) provides analysis of all virtual network traffic using Intrusion Prevention System (IPS) [32]. IPS, an advanced version of Intrusion Detection

Systems, is capable of detecting and preventing both known and unknown attacks. Rootkits [33] originally have been implemented as regular applications to assist in gaining control of a failing or unresponsive system, but recently, they are used as malware to help intruders to access systems while avoiding detection. An anti-rootkit approach was proposed in [34] for automated detection and containment of user level as well as kernel-level rootkit attacks and other malwares that use rootkits for hiding.

Finally, Anti-DDoS Virtualized Operating System (ADVOS) [35] was recently proposed to secure networked computers against DDoS attacks. ADVOS integrates anti-DDoS capabilities in operating systems by performing packet filtering at the source computer itself to classify malicious traffic. Furthermore, the anti-DDoS was moved outside the host into independent domain to protect anti-DDoS from misbehaving by malicious code. ADVOS was not proposed to be used in Cloud, but we believe that ADVOS would be a feasible and an effective solution for DDoS in any virtualization environment especially for IaaS.

Virtual machines Mobility: Mobility is an advantage feature that allows VMs to be transferred to other physical machines where the contents of the virtual disk for each VM are stored as a file. Mobility is essential for systems maintenance and load balancing, but it would be source of security risk (e.g., VM file can be stolen without physical theft of the host machine). The integrity of an offline VM might be compromised if the host is not secure and protected. For instance, Offline attacks might be occurred by copying an offline VM over the network or to a portable storage media and access or corrupt data on their own machine without physically stealing a hard drive [36]. On the other hand, live virtual machine migration might be a sever threat to VM, where live migration techniques [37],[38] usually design by copying the memory pages of the VM across the network from the source VMM to destination VMM. In [39], three classes of live migration attacks against live VM were explored empirically to show the significance of securing migration process. Furthermore, [39] demonstrated how a malicious party can exploit the latest versions of Xen and VMware virtual machine monitors and proposed a mutual authentication between source and destination VMMs to protect migration process through the network as mentioned earlier. LoBot [40] is an architecture for secure provisioning and migration of virtual machines within the cloud. LoBot provides many other security features for Private Virtual Infrastructure (PVI) such as environmental monitoring, tamper detection and secure shutdown.

Denial of Service (DoS): Denial of Service (DoS) [41] attacks in virtual environment are a critical threat to VMs. These attacks can be an outcome of a hypervisor's misconfiguration that allows a single VM to consume all available resources, thus starving any other VM running on the same physical machine and avoiding network hosts to function appropriately due to the hardware resources shortage. However, Hypervisors prevent any VM from gaining 100% usage of any shared hardware resources, including CPU, RAM, network bandwidth, and graphics memory [11]. Additionally, an appropriate hypervisor's configuration enables extreme resource consumption detection to take the suitable solution, e.g., automatically restart the VM, nevertheless, restarting the VM has a smaller effect than restarting a physical machine, where VMs can usually be initialized much faster than physical machines because there is no need to initialize and verify hardware.

E. Networks & Internet Connectivity

To maintain availability and performance, cloud infrastructure spans multiple geographical sites to reduce the latency and the damage of unpredicted disasters. Each site connected locally as local area network is connected with the other sites by high speed Internet connections. These sites in total compose the cloud infrastructure which serves remote clients through the Internet. Thus, Cloud bequeaths both the conventional vulnerabilities of Internet and computer networks. IaaS model is vulnerable to DDOS, MITM, IP Spoofing, and Port Scanning. For instance, a web-based code hosting service that uses both EC2 and the Amazon's Elastic Block Storage reported 19 hours of downtime as a result of a DDoS attack[8]. In addition to the external attacks (from the Internet), IaaS is exposure to internal attacks initiated from internal VMs against internal services. The internal attacks can be more severe than external attacks due to the system administrator's privileges on VMs that allow him to install and run any malicious applications. Furthermore, the dynamicity of IaaS environment (e.g., creating, removing, migrating VMs) adds more challenges to build defense plans against any attacks. Obviously, IaaS is more vulnerable to networks attacks than any other networked system; however, some of practical solutions and techniques for eliminating these attacks or reducing their impacts are listed as follows:

Logical network segmentation: A restrictive and a well planned network configuration should be applied in IaaS environment beside the hypervisor isolation power. VLAN offers isolated segments to prevent the external VMs from sniffing or monitoring internal traffic; for instance, bridges forward unicast, multicast, and broadcast traffic on a VLAN segment only to VMs which have a virtual interface in that segment. Administrator have to choose the best connection model, i.e., Routing, NAT, or simple bridging between VLANs. Thus, virtual networks avoid wasting bandwidth and offer more flexibility, performance, and security [16].

Firewalls implementing: using firewalls enforce the organization's security policy by implementing rules to control the traffic based on protocol type, service port, and source IP address. Traditional three-tiered web applications architecture advised by Amazon AWS[9] , could be a secure architecture for deploying applications such as in the next scenario. Port 80 (HTTP) and/or port 443 (HTTPS) should be accessible to the world and port 8000 should be accessible only to the web

[8]http://www.theregister.co.uk/2009/10/05/amazon_bitbucket_outage/

[9]AWS Security Whitepaper 2008-2009

servers' group meanwhile port 3306 will be accessible only to the application servers' group. A well configured firewall for VMs instances level also is recommended to prevent all traffic except the required traffic.

Traffic encryption: To access the outsourced infrastructure on the clouds, clients need secure channels to ensure privacy and integrity of the transferred data. VPNs provide encrypted tunnel between the client and the provider using Layer 2 Tunneling Protocol (L2TP) or Point-to-Point Transfer Protocol (PPTP), but, since these protocols are point to point, they cannot secure user's traffic inside the cloud. Thanks to WS-Security, it became possible to maintain a secure context over a multi-point message path[10], but because WS-Security provides security at message level, it requires all the entities within the cloud to support the web services and to communicate using SOAP messages.

Network monitoring: In IaaS model, providers are responsible for network monitoring to sustain acceptable level of QoS. The monitoring process includes malicious activity, fault detection, and troubleshooting. In cloud, Network monitoring is not simple compared with traditional networks because cloud is geographically distributed and depends significantly on resources sharing. Furthermore, cloud infrastructure is a public environment containing multiple monitoring records refer to anonymous (users rent some resources for specific time then left). MapCenter [42] and NetSaint [43] are two examples of Grid monitoring systems. However, NetSaint, the "official choice" of INFN-Testbed, is able to notify the events into selected users or groups and to handle event to fix problems automatically. As NetSaint relies on external programs (plug-ins) to do all the monitoring activities, it could be deployed for cloud environment preventing providers from installing sensors on the clients' VM instances. However, this should be studied to show its feasibility and performance in cloud environment.

F. Computer Hardware

IaaS offers an interface to a pool of distributed physical resources (e.g., Network Components, CPUs, and Storage Devices) and delivers a shared business model to serve multiple consumers. Virtualization, as seen previously, can keep a secure share of the computer resources and a controlled communication on hardware and network level. Eventhough the private organizations used to move the hardware components into locked rooms accessible only by the authorized and trusted persons to protect the resources, a study showed that over 70% of all attacks on organizations' sensitive data and resources occurred internally (i.e., from inside the organization itself) [44]. Physical resources include networks components, computing resources, and storage resource. As the security of the network components as hardware is out of this paper scope, we will only discuss the physical security of computing and storage resource:

Computing resources: As discussed earlier, we consider that an attacker is able to access the machine physically. Depending on the goal of the attacker, we have multiple scenarios. First scenario is denying the service by turning off the machine or by removing any of hardware resources. This is not a common attack, but it can hurt the company's reputation. Hence, IaaS providers must carefully control the access to physical resources. Second scenario is accessing the physical machine to get or corrupt data for specific company benefit. Eventhough TCCP proposed a solution to protect the running VMs and their contents, storage resources still vulnerable to physical access attacks that will be discussed next.

Storage resources: IaaS providers play an essential role in protecting clients' data. Whatever the level of the data security, it can be part of retired or replaced storage devices. Usually, companies don't have restrictive policy to manage the retired devices that could be accidentally devolved to untrusted people. Each organization is supposed to assure clients' data security along its life cycle. Encryption would be a good solution, but it might prevent the other users' accessibility to the data. To support multi parties' accessibility to encrypted storage, [45] propose architecture to mange encryption keys. Nevertheless, this approach increases traffic and degrade performance. Transparent cryptographic file systems (NAS_CFS) [46] provides a high security by using session ID and user ID for key management. Implementing (NAS_CFS) [46] in-kernel file system level does not cause a significant degradation in the performance. IBM [11] has introduced first self-encrypting enterprise tape drive (TS1120) and full disk encrypting drives (DS8000 & DS5000) supported by key management software system to prevent access to the encrypted data stored on replaced or retired hard disks without the original encryption drive.

Finally, after a detailed discussion for IaaS threats and challenges and the proposed solutions, we summarize them in Table 1.

III. Security Model for IaaS

As a result of this research, we propose a Security Model for IaaS (SMI) as a guide for assessing and enhancing security in each layer of IaaS delivery model as shown in Fig. 3. SMI model consists of three sides: IaaS components, security model, and the restriction level. The front side of the cubic model is the components of IaaS which were discussed thoroughly in the previous sections. The security model side includes three vertical entities where each entity covers the entire IaaS components. The first entity is Secure Configuration Policy (SCP) to guarantee a secure configuration for each layer in IaaS Hardware, Software, or SLA configurations; usually, miss-configuration incidents could jeopardize the entire security of the system. The second is a Secure Resources Management Policy (SRMP) that controls the management roles and privileges. The last entity is the Security Policy Monitoring and Auditing (SPMA) which is significant to track the system life cycle. The restriction policy side specifies the level of restriction for security model entities. The level of

[10]http://msdn.microsoft.com/en-us/library/ms977327.aspx

[11]http://www.ibm.com

TABLE I
THREATS AND SOLUTIONS SUMMARY FOR IAAS

IaaS Component	Threats / Challenges		Solutions	
Service Level Agreement (SLA)	Monitoring and enforcing SLA. Monitor QoS attributes.		Web Service Level Agreement (WSLA) framework. SLA monitoring and enforcement in SOA.	
Utility Computing	Measuring and billing with Multiple levels of providers On-demand billing system availability.		Amazon DevPay.	
Cloud Software	Attacks against XML. Attacks against web services.		XML Signature and XML Encryption. SOAP Security Extensions.	
Networks & Internet connectivity	DDOS Man-In-The-Middle attack (MITM). IP Spoofing. Port Scanning. DNS security.		Logical Network segmentation and Firewalls. Traffic encryption. Network monitoring. Intrusion Detection System and Intrusion Prevention System (IPS).	
Virtualization	Security threats sourced from host: • Monitoring VMs from host. • Communications between VMs and host. • VMs modification.	Security threats sourced from VM: • Monitoring VMs from other VM. • Communication between VMs. • Virtual machines Mobility • Resources Denial of Service (DoS). • VMs provisioning and migration.	Security threats sourced from host: • Trusted Cloud Computing Platform • Terra • Trusted Virtual Datacenter (TVDc) • Mandatory Access Control MAC	Security threats sourced from VM: • IPSec. • Encryption. • VPN. • Xen Security through Disaggregation. • LoBot architecture for secure provisioning & migration VM
Computer Hardware	Physical attacks against computer hardware. Data security on retired or replaced storage devices.		High secure locked rooms with monitoring appliances. Multi-parties accessibility to encrypted storage. Transparent cryptographic file systems. Self-encrypting enterprise tape drive TS1120.	

restriction starts from loose to tight depending on the provider, the client, and the service requirements. Nevertheless, we hope SMI model be a good start for the standardization of IaaS layers. This model indicates the relation between IaaS components and security requirements, and eases security improvement in individual layers to achieve a total secure IaaS system.

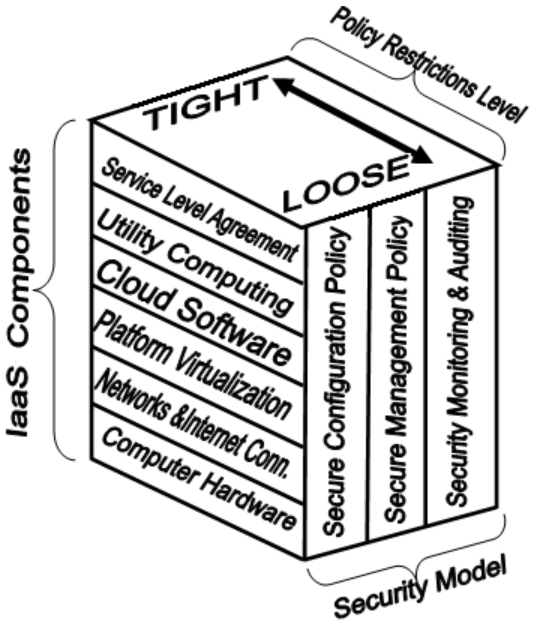

Fig. 3. Security Model for IaaS

IV. CONCLUSION AND FUTURE WORK

IaaS is the foundation layer of the Cloud Computing delivery model that consists of multiple components and technologies. Each component in Cloud infrastructure has its vulnerability which might impact the whole Cloud's Computing security. Cloud Computing business grows rapidly despite security concerns, so collaborations between Cloud parties would assist in overcoming security challenges and promote secure Cloud Computing services.

In this paper, we investigated the security challenges that associated with IaaS implementation and deployment. The security issues presented here concern the security of each IaaS component in addition to recent proposed solutions. However, some of these components take a part in other computing technologies such as Grid, but virtualization is the key of IaaS model. Hence, we presented a majority of the security issues concern the host and VM; furthermore, we considered Utility Computing as one of IaaS components and showed how SOA and Cloud Computing are related. Finally, an IaaS security model was proposed as a guide for assessing and enhancing security in each layer of IaaS delivery model.

Our future research vision will focus on two directions to provide confidentiality, integrity, and secure infrastructure management for IaaS service. First, extending techniques such as proposed in TCCP into IaaS layer to improve confidentiality and integrity of VMs. Second, integrating TCCP with secure resources management schemes to get more controlled isolation environment. Finally, a prototype will be implemented to demonstrate the system feasibility and performance.

REFERENCES

[1] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities," *Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications*, p. 9, August 2008. [Online]. Available: http://arxiv.org/abs/0808.3558

[2] SLA Management Team, *SLA Management Handbook*, 4th ed. Enterprise Perspective, 2004.

[3] G. Frankova, *Service Level Agreements: Web Services and Security*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 4607.

[4] P. Patel, A. Ranabahu, and A. Sheth, "Service Level Agreement in Cloud Computing," *Cloud Workshops at OOPSLA09*, 2009. [Online]. Available: http://knoesis.wright.edu/aboutus/visitors/summer2009/PatelReport.pdf

[5] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The Eucalyptus Open-Source Cloud-Computing System," *Cluster Computing and the Grid, IEEE International Symposium on*, vol. 0, pp. 124–131, 2009.

[6] T. Mather, S. Kumaraswamy, and S. Latif, *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance*, 1st ed., 2009. [Online]. Available: http://books.google.com/books?id=BHazecOuDLYC&pgis=1

[7] R. Kanneganti and P. Chodavarapu, *SOA Security*. Manning Publications, 2008. [Online]. Available: http://www.amazon.com/SOA-Security-Ramarao-Kanneganti/dp/1932394680

[8] M. McIntosh and P. Austel, "XML signature element wrapping attacks and countermeasures," *Workshop On Secure Web Services*, 2005.

[9] M. Jensen, J. Schwenk, N. Gruschka, and L. Lo Iacono, *On Technical Security Issues in Cloud Computing*. IEEE, 2009.

[10] B. D. Payne, "Xenaccess." [Online]. Available: http://doc.xenaccess.org/

[11] J. Kirch, "Virtual machine security guidelines," 2007. [Online]. Available: http://www.cisecurity.org/tools2/vm/CIS_VM_Benchmark_v1.0.pdf

[12] T. G. Ben, B. Pfaff, J. Chow, M. Rosenblum, and D. Boneh, "Terra: A Virtual Machine-Based Platform for Trusted Computing." ACM Press, 2003, pp. 193–206.

[13] S. Berger, R. Cáceres, D. Pendarakis, R. Sailer, E. Valdez, R. Perez, W. Schildhauer, and D. Srinivasan, "TVDc: Managing security in the trusted virtual datacenter," *ACM SIGOPS Operating Systems Review*, vol. 42, no. 1, p. 7, 2008.

[14] S. Berger, R. Caceres, D. Pendarakis, R. Sailer, E. Valdez, R. Perez, W. Schildhauer, and D. Srinivasan, "Security for the cloud infrastrcture: trusted virtual data center (TVDc)." [Online]. Available: www.kiskeya.net/ramon/work/pubs/ibmjrd09.pdf

[15] N. Santos, G. P. Krishna, and R. Rodrigues, "Towards Trusted Cloud Computing," *HotCloud'09*, 2009. [Online]. Available: http://www.usenix.org/event/hotcloud09/tech/full_papers/santos.pdf

[16] V. Rajaravivarma, "Virtual local area network technology and applications," *Proceedings The Twenty-Ninth Southeastern Symposium on System Theory*, pp. 49–52, 1997.

[17] W. Mao, A. Martin, H. Jin, and H. Zhang, *Security Protocols*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, vol. 5087.

[18] "Property-Based TPM Virtualization," *Lecture Notes In Computer Science; Vol. 5222*, 2008.

[19] V. Scarlata, C. Rozas, M. Wiseman, D. Grawrock, and C. Vishik, "TPM Virtualization: Building a General Framework," pp. 43 – 56, 2007.

[20] S. Berger, R. Cáceres, K. A. Goldman, R. Perez, R. Sailer, and L. van Doorn, "vTPM:virtualizing the trusted platform module," *USENIX Security Symposium*, 2006.

[21] D. G. Murray, G. Milos, and S. Hand, "Improving Xen security through disaggregation," *ACM/Usenix International Conference On Virtual Execution Environments*, p. 9, 2008.

[22] J. Schiffman, T. Moyer, C. Shal, J. Trent, and P. McDaniel, "Justifying Integrity Using a Virtual Machine," *25th Annual Computer Security Applications Conference (ACSAC)*, 2009. [Online]. Available: http://www.patrickmcdaniel.org/pubs/acsac09c.pdf

[23] R. Sailer, E. Valdez, T. Jaeger, R. Perez, L. van Doorn, J. L. Griffin, and S. Berger, "sHype: Secure hypervisor approach to trusted virtualized systems," NewYork, p. 12, 2005.

[24] C. Rozas, "Intels security vision for Xen," 2005. [Online]. Available: www.xen.org/files/XenSecurity_Intel_CRozas.pdf

[25] J. Matthews, E. M. Dow, T. Deshane, W. Hu, J. Bongio, P. F. Wilbur, and B. Johnson, *Running Xen: A Hands-on Guide to the Art of Virtualization*, 2008. [Online]. Available: http://books.google.com/books?id=XS-Jj7s2nhYC&pgis=1

[26] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1." 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4346.txt

[27] S. Kent and K. Seo, "Security Architecture for the Internet Protocol." 2005. [Online]. Available: http://www.ietf.org/rfc/rfc4301.txt

[28] E. Rosen and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)," 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4364.txt

[29] S. Axelsson, "Intrusion Detection Systems: A Survey and Taxonomy," 2000.

[30] K. A. Jackson, "Intrusion Detection Systems (IDS): Product Survey," *Los Alamos National Laboratory*, 1999.

[31] K. Vieira, A. Schulter, C. Westphall, and C. Westphall, "Intrusion Detection Techniques in Grid and Cloud Computing Environment," *IT Professional*, vol. 99, no. PrePrints, 2009.

[32] X. Zhang, C. Li, and W. Zheng, "Intrusion Prevention System Design," *CIT*, 2004.

[33] S. King and P. Chen, *SubVirt: Implementing malware with virtual machines*. IEEE, 2006.

[34] A. Baliga, L. Iftode, and X. Chen, "Automated containment of rootkits attacks," *Computers & Security*, vol. 27, no. 7-8, pp. 323–334, 2008. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0167404808000382

[35] S. Garg and H. Saran, "Anti-DDoS Virtualized Operating System," *ARES*, p. 7, 2008.

[36] T. Garfinkel and M. Rosenblum, "When virtual is harder than real:security challenges in virtual machine based computing environments," *Proceedings of the 10th conference on Hot Topics in Operating Systems - Volume 10*, 2005.

[37] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2*, 2005.

[38] J. G. Hansen and E. Jul, "Self-migration of operating systems," *ACM SIGOPS European Workshop*, 2004.

[39] J. Oberheide, E. Cooke, and F. Jahanian, "Empirical exploitation of live virtual machine migration," *In Proc. of BlackHat DC convention*, 2008.

[40] F. J. Krautheim and D. S. Phatak, "LoBot: Locator Bot for Securing Cloud Compupting Environments," *ACM Cloud Computing Security Workshop*, 2009.

[41] S. Lin and T. Chiueh, "A Survey on Solutions to Distributed Denialof Service Attacks." [Online]. Available: http://www.ecsl.cs.sunysb.edu/tr/TR201.pdf

[42] F. Bonnassieux, R. Harakaly, and P. Primet, "MapCenter: An Open Grid Status Visualization Tool," in *proceedings of ISCA 15th International Conference on parallel and distributed computing systems*, 2002, pp. 2–3. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.11.7294

[43] R. Barbera, P. L. Re, G. Sava, and G. Tortone, "Grid monitoring with NetSaint," *Bologna-Datagrid WP7 meeting*, 2002. [Online]. Available: http://www.cnaf.infn.it/~ferrari/infn-grid-wp5/task-dg/task3/datagridwp7-netsaint.pdf

[44] E. Markatos, "Large Scale Attacks on the Internet Lessons learned from the LOBSTER project," Crete, Greece. [Online]. Available: http://www.ist-lobster.org/publications/presentations/markatos-attacks.pdf

[45] L. Seitz, J.-M. Pierson, and L. Brunie, "Key Management for Encrypted Data Storage in Distributed Systems," *SISW*, 2003.

[46] H. Jianzhong, X. Changsheng, and C. Bin, "Research and Implement of an Encrypted File System Used to NAS," *SISW*, 2003.