A Deep Study on Security Vulnerabilities in Virtualization at Cloud Computing

Nayeem Khan
Faculty of Computer Science & IT
University Malaysia Sarawak
94300, Kota Samarahan, Sarawak, Malaysia

Tariq Ahmad War
SP College, MA Road, Srinagar
J&K, India

ABSTRACT
Virtualization is the process of creating a virtual representation of architecture. It has advantages on five components; sharing, isolation, aggregation, dynamics, and ease of management. However, issues that rise due to the nature of virtualization, especially security issues has skyrocketed. To counter this, a number of solutions has been presented in multiple literature. In this paper, a new solution to improve the security of the system is proposed. In addition, the flaws of the current implemented system are discussed, and the advantages of the proposed system over the current implemented system are listed out.

General Terms
Security, Cloud Computing

Keywords
Virtualization, Cloud Computing, Hypervisor, Virtual machine, IaaS, PaaS, SaaS.

1. INTRODUCTION
Due to the advancement of the Internet, virtualization has significantly changed and enhance the architecture and safety of cloud computing. Virtualization has advantages on these five components; sharing, isolation, aggregation, dynamics, and ease of management [1]. In cloud computing, virtualization is an important element for sharing resources virtually and allowed the provider to share resources to its tenant without risking the security and integrity of the data. Nowadays, the use of virtualization has become popular because it allowed the user work in different type of environment or platform by using the same devices.

This virtualization in cloud computing however comes with many security concern such as the attack of hypervisor, network security loss, data loss and data leakage [2]. Some of the major contributors of security challenges in cloud computing is the vulnerability of hypervisor which often become the main target to virtualization attack, a new network standard known as Software-Defined Networking (SDN) which based on central control and the security vulnerabilities of virtualization in cloud computing.

Hypervisor is an important component in virtualization as it provides virtual sharing of resources and most of the virtualization architectures are exposed to different kind of runtime spaces, multi-tenant and shared environment, hence it is easy for the malicious attacker performing virtualization attack on the hypervisor using another runtime spaces by taking consideration of this advantages [3]. While, the Software-Defined Networking (SDN) is a new standard for network virtualization which is based on the concept of central control system that permits the access through specific network so any successful attack may lead to network security loss, data loss and network become more vulnerable after each attack. Besides, security is one of the most important component that need to be consider and one way of attacking is through uploading files online which have a risk of a breach which could occur anytime and it may affect the data confidentiality. Having hypervisor alone and not protected to manage all the shared hardware in cloud can be fatal because it will become the perfect goal of malicious user securing the hypervisor will ensure success access to all virtual machines (VM) shared resources. The new demands and trends boost the development of new standards regarding network virtualization is not a very suitable solution to help solving this problem because it is based on central control system permits thorough access through the network, any successful attacks can lead to network security loss and the network security will be much vulnerable after the each attack. While, the issues of security and scalability emerge from the multi-tenancy's environment and result of the two type of scalability which is the horizontal scalability and vertical scalability.

The suggested solution that will be discussed further in this paper is a solution that deployed firewall to each virtual machines that are in the virtual environment. The objective of this solution is to provide extensive protection to the hypervisor and virtual machines. The embedded firewall will prevent attacks from penetrating the hypervisor and the VMs. This allow for a better defense mechanism against rootkits and other malicious attacks.

This study focuses on the cloud computing vulnerabilities and its security concern before analyzing a few of existing approaches that can be used as the solutions to secure the virtualization in cloud computing. The first section of this paper is to present the background study of the security vulnerabilities of virtualization in cloud computing. In the rest of the section, a few method and approaches will be presented to help counter this virtualization vulnerabilities and improve the security of virtualization in cloud computing. Finally, this paper will be concluded along with a perspective of our future work.

2. TECHENICAL BACKGROUND

2.1 The architecture of cloud computing
Cloud computing enhances collaboration, agility, scale, availability, and cost reduction. More specifically, cloud describes the use of a collection of distributed services, applications, information and infrastructure comprised of pools of compute, network, information and storage resources [4]. These components can be manipulated, provisioned, implemented and decommissioned using an on demand utility-like model of allocation and consumption. Cloud service is often utilized in conjunction with virtualization technologies to provide dynamic integration, provisioning, orchestration, mobility and scale. Cloud service models

describe how cloud services are made available to clients. Most fundamental service models include a combination of Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). These service models can either be interdependent with each other, or have synergies between each other as shown in Figure 1.

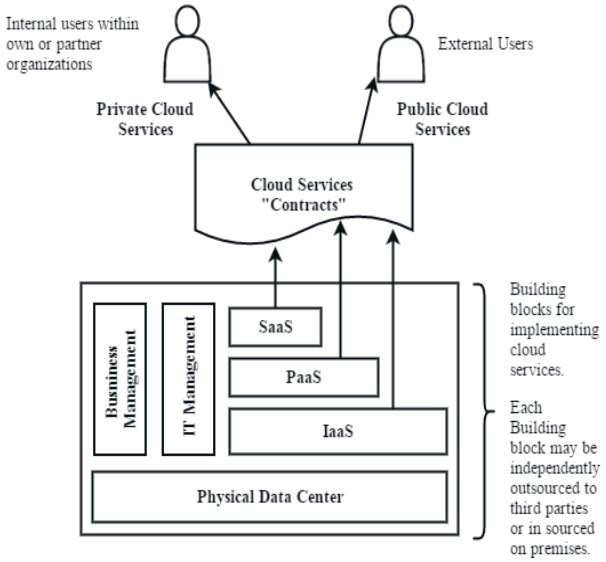

Fig. 1: The architecture of cloud computing

2.2 Software as a Service (SaaS)

This model allows user to use and access the services provided on the cloud infrastructure. In other words, a complete application is offered to the customer as a service on demand. However, the users do not manage or control the underlying cloud infrastructure, network, servers, and individual application capabilities, with the possible exception of limited user-specific application configuration settings. Google and Microsoft are two companies that offered SaaS to their consumers.

2.3 Platform as a Service (PaaS)

In this model, a layer of software or development environment is encapsulated and offered as a service. The user has the freedom to build his own applications, which run on the provider's infrastructure. Although users does not manage or control the underlying cloud infrastructure, network, servers, operating systems, or storage, but the control over the deployed applications and the application hosting environment configurations are under the users. To meet manageability and scalability requirements of the applications, PaaS providers offered a predefined combination of operating systems and application servers, such as LAMP (Linux, Apache, MySql and PHP) platform. Google's App Engine is an example of services that implements PaaS models.

2.4 Platform as a Service (PaaS)

In this model, a layer of software or development environment is encapsulated and offered as a service. The user has the freedom to build his own applications, which run on the provider's infrastructure. Although users does not manage or control the underlying cloud infrastructure, network, servers, operating systems, or storage, but the control over the deployed applications and the application hosting environment configurations are under the users. To meet manageability and scalability requirements of the applications, PaaS providers offered a predefined combination of operating systems and application servers, such as LAMP (Linux, Apache, MySql and PHP) platform. Google's App Engine is an example of services that implements PaaS models.

2.5 Platform as a Service (PaaS)

This model provides basic storage and computing capabilities as a standardized service over the network. Refer to Figure 2 for an example of IaaS. Servers, storage systems, networking equipment and data center space are pooled and made available to handle workloads. Users are allowed to rent processing, storage and other fundamental resources. The user has controls on the operating systems, storage, applications and some networking component. In contrast, the user does not have control on the underlying cloud structures. Some examples of IaaS are Amazon and GoGrid.

In an IaaS model, third-party provider hosts hardware, software, servers, storage and other infrastructure components on behalf of its users. IaaS providers also host users' applications and handle tasks including system maintenance, backup and resiliency planning. IaaS platforms offer highly scalable resources that can be adjusted on-demand. This makes IaaS well-suited for workloads that are temporary, experimental or change unexpectedly. In addition, IaaS users pay for the services on a pay per-use basis. Some providers charge users based on the space used.

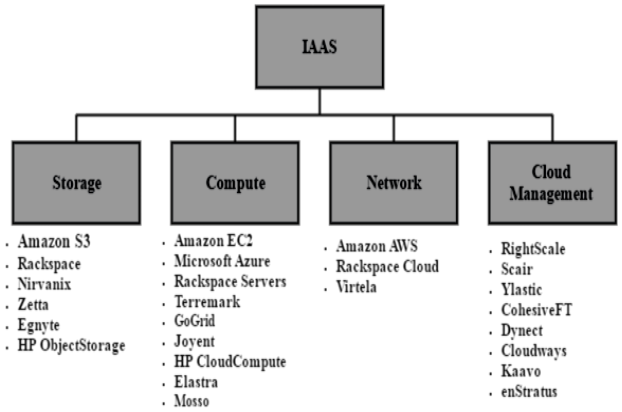

Fig. 2: The IaaS architecture

3. PREVIOUS WORKS

In this section, some existing solutions are discussed on their functionality and their flaws. There are two existing solutions that will be discussed in this section.

3.1 NoHype and HyperWall

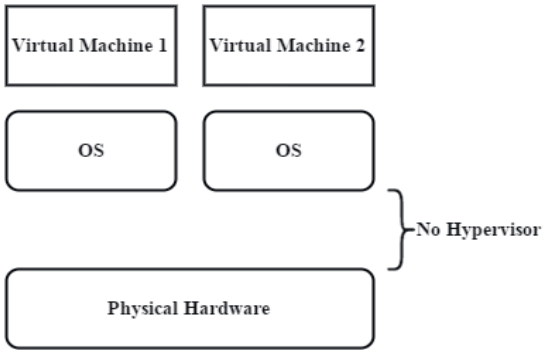

Fig. 3: The No-Hype architecture

There are two types of methods that can be used to defend the hypervisor from virtualization attack. These methods are called NoHype and HyperWall. NoHype architecture is an architecture that eliminates the hypervisor and fragile part in traditional virtualization architecture [5] while HyperWall provides security assurance to the guest user and generate security trust assurance during the VM's lifetime. The NoHype architecture as shown in Figure 3 removes the hypervisor in the computer architecture, rendering the attacks on the system pointless as there is nothing to attack. The system above can be considered complete, but it is still retains the needs of a virtualized cloud infrastructure. Therefore, the cloud infrastructure is used for virtualization because it supported the dynamic starting and/or stopping of the Virtual Machines (VMs).

However, these two solutions can't solve the issues of multi-tenant problems. The data can still be accessed even without the authorization of the user. This makes the solution insecure.

3.2 OpenADN

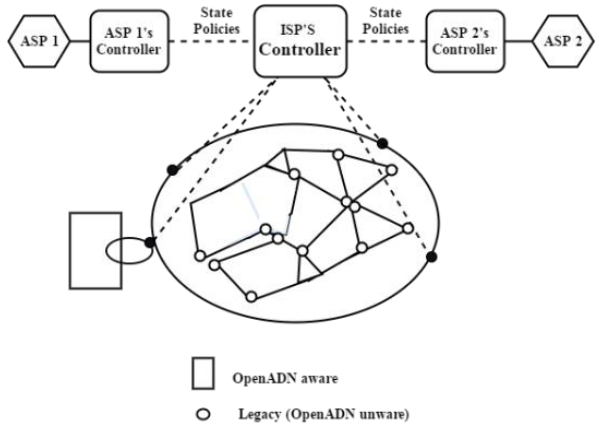

Fig 4: The openADN Architecture

The virtualization in both network and service provider are necessary to ensure the efficiency and future success in cloud computing. This is a critical issue that worries those who are considering to outsource their data storage and processes. The Software-Defined Networking (SDN) is one of the key developments of OpenADN. SDN allows the control and data planes to separated and differentiated. It also allows the control to be centralized and easy to be programmed [6]. This let multiple devices to be programmed. Figure 3 above shows how OpenADN and ASPs' controllers conveyed their policies. The design in network virtualization technique consists of five parts which are; Virtualization of NICs, Virtual LANs in Clouds, Virtualization for Multi-Site Data Centers, Virtualization of Switches and Network Function Virtualization.

In this approach, the hypervisor does not have any interaction with the execution of the VM, which is why, after the boot up, the architecture needs to be able to disengage the temporary hypervisor, used for starting up needs. To do this, a guest OS kernel module sends a hypercall [7], [8] to the handler to perform the disengagement function. It need to remove the VMs from a list of timers, and remove the cores from the online processor cores mask. The second step is to configure the hardware as the hardware has complete control over the core by defining settings in the virtual machine control structure and mappings in the extended age tables. Finally, it need to initialize the local APIC registers with written values to match the guest OS values. This is quite tedious, and takes a lot of resources and time.

4. PROPOSED SOLUTION

Existing solutions to counter issues of virtualization in cloud computing has been done in multiple literature. Therefore, this section is an elaborated explanation of our new solution which proposed for this work. The aim of this proposed solution is to provide an extensive protection to the hypervisor and VMs; prioritizing the VMs. The proposed solution is virtualizing firewall inside virtual machines which aimed to enforce high security measures by implementing firewall inside every virtual machine in the virtual environment.

Hypervisor is a critical component that plays the role of "traffic controller" that administers the processes of virtual environment to host environment. Attacks are usually aimed to hypervisor because the intruders desire to manipulate all the residing components in virtual environment as well as the host. Threat such as Rootkit or network threat [9] is injected into the hypervisor, and if the attempts of attack success, threat will progressively intrudes the VMs and then following the host. Once hypervisor failure happens, all components are unable to functions accordingly, resulting to disrupted system. This is where implementing firewall in every VMs can aid in blocking the threat. Figure 5 below shows the diagram for our approach.

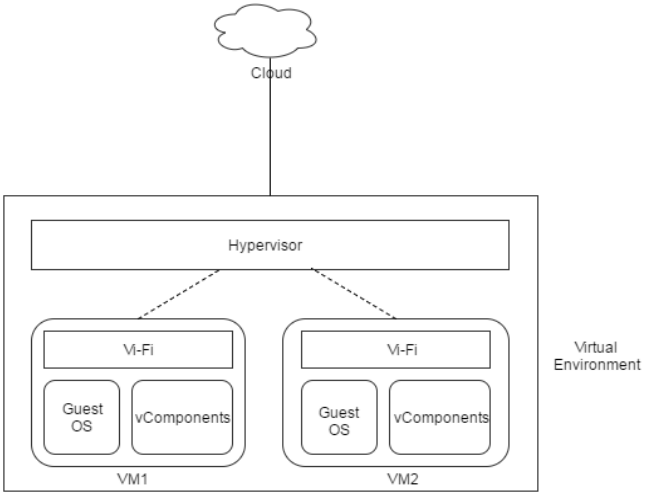

Fig. 5: Firewall virtualization inside the virtual machines

As proposed in the figure 5, deploying firewall in each VMs dynamically giving security enhancement to each of the VMs. In our reflection, embedding firewall inside every VMs will prevent any successful attack in hypervisor from seeping into the VMs, thus preventing the malicious entity from penetrating and exploiting the host system. This approach is a security measure of preventing the VM and host system from being damaged after the hypervisor has been attacked. Using Vi-Fi in the VMs is a step ahead by practicing early detection for every flow and the packets that pass through will be monitored. By virtualizing firewall inside every virtual machines, packet patrolling security is upped by a notch thus, halting attempts of attacks will be much easier.

Our motivation of this solution comes from recent research of a bug named VENOM. VENOM stands for Virtual Environment Neglected Operation Management and it is a vulnerability that will yield massive damage on QEMU legacy virtual floppy disk controller. It permits an intruder to escape from the guest VM after a success implantation of itself in

hypervisor, then may directly proceed to interfere the host hardware along with other VMs [10]. We believe that this vulnerability can be prevented from interrupting the VMs and the host by practicing the proposed solution. Vi-Fi will aid the virtual environment by elevating the security level; technically if the VMs is impenetrable, then the host system will not be exploited. Vi-Fi is able to protect its own VM by acting as a "security guard" that performs scanning on every packet data that went through its VM. We believe that it is almost impossible even for a hidden threat to be able to leak through as every packet data is being inspected strictly.

This solution emphasize on affordable solutions that works. Unlike other solutions, this method is not only cost-effective due to it being a network appliances compared to the usage of physical protection, it also monitors the data flow of each virtual machine, making the system sensitive to detect any abnormal data flow. Besides that, Vi-Fi also provides flexibility in terms of location of security monitoring by enabling remote placement of security control; for instance, in memory. Not only that, firewall virtualization is better than physical firewall in terms of redundancy because if hardware failure happens on one of them, virtual server is able to be migrated to another host automatically. Moreover, the usage of virtual firewall in each VMs independently will not bring restriction as virtual firewall is depending on the host resource instead of themselves.

5. PERFORMANCE ANALYSIS

In this section, the algorithm on firewall virtualization is analyzed. This algorithm [23] creates the explanation of pair-to-pair sessions, testing all the possible combinations between all the virtual machine and from virtual machine to the external location and vice-versa. Figure 6 below shows a piece of code which is an algorithm that act as an "evaluator" or "inspector" at determining whether the packet data that flows through every VMs are a legit data. A session contains a server and a client where the client sends TCP, UDP and ICMP packets to the server IP address on recognized ports or sub protocols with a specific 4-octet payload set in the packets. The server listens for all incoming packets and applies two filters which is to identify the particular payload and to capture packet address to it. The server machine may as well be a client of another session, but it can't act as two servers at a time [23] which consents at most a total number of parallel-running servers equal to the number of VMs. Figure 6 below is the algorithm of dynamic analysis of firewall virtualization inside VMs.

Still_Session← True

While Still_sessions do

Still_sessions←False, S←Ø

For n €1….Card(V)do

Session_found←False, vm1← 1

While vm1_<Card(V) and session_found do

vm2← 1

While vm2_Card(V)and session_found do

ip← 1

while M[vm1][vm2][ip]do ip++end while

if !M[vm1][vm2][ip]and S∩ {V[vm1],V[vm2], V[vm2] [ip]}== Ø then

still_session←True

run_server{V[vm1],V[vm2],V[vm2][ip]}

end if

vm2++

end while

vm1++

end for

for s € S do update_access(AM,s,t1) end for

end whle

s←Ø;

for vm € 1…Card(V)do

run_server(this);run_client(V[vm]);update_Access(AM, {V[vm],this,this_ip),t2)

end for

Dynamic analysis of firewall virtualization algorithm

Sleeping time (*t1*) between the launch of the server and the client is set to ensure that the server is ready to accept packets. Throughout the iterations of the main algorithm, three-dimension session array which contain 1st dimension: clients, 2nd dimension: servers, 3rd dimension: server IP address report the complete and uncompleted session [11]. All the value in the array is set to false, except when the server and the client are the same. The clients are monitored to recognize whether the packet sending operation has completed. If it has been completed, the servers can be interrupted and the result can be retrieved from them at the end of algorithm iteration. By running the VMs to VMs sessions then continued with the external location to VMs sessions (multithreaded) and finally the VMs to the external locations sessions sequentially, the algorithm can be started. Equation that can be used to evaluate the total amount of packet data flow between VMs and host systems.

NV as the total number of VMs

N I Pi as the total number of IP addresses of VM i.

$NS=\sum_{o<i<NV}(1+\sum_{o<i<NV} NIPj)+\sum_{o<i<}NV$

as the total number of sessions. Composed of the sessions from VMs to other VMs and the external location, and sessions from the external location to VMs.

NSi =N I Pi×(NV-1) the number of inter-VM sessions for a server i.d as the delay between packet sending.

N P as the number of packets to be sent in each session.

Equation to evaluate total amount of packet data flow between VMs and host

6. CONCLUSION

Virtualization has significantly changed cloud computing industry by enhancing its secureness. However, these improvements comes with a price. The security of virtualization has left many users reluctant and doubtful. Therefore, many solutions are being developed especially as a

counter measure for security issues. These solutions are found abundantly on multiple literature, each boast advantages over the other. However, while focusing the defense mechanism on one area, the areas was left vulnerable to attacks. A new solution was proposed, called Vi-Fi, or virtual firewall. Embedded firewalls in the virtual environment will prevent successful attacks, rendering all attacks useless. By taking into account of the disadvantages of the existing solutions in Section III, the new solution that was proposed in Section IV be able to cover all the flaws of the existing solution.

7. REFERENCES

[1] Garbacki, P., & Naik, V. K. (2007, May). Efficient resource virtualization and sharing strategies for heterogeneous grid environments. In Integrated Network Management, 2007. IM'07. 10th IFIP/IEEE International Symposium on (pp. 40-49). IEEE.

[2] Ali, M., Khan, S. U., & Vasilakos, A. V. (2015). Security in cloud computing: Opportunities and challenges. Information Sciences, 305, 357-383.

[3] Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2008). A break in the clouds: towards a cloud definition. ACM SIGCOMM Computer Communication Review, 39(1), 50-55.

[4] Gonzalez, N., Miers, C., Redigolo, F., Simplicio, M., Carvalho, T., Näslund, M., & Pourzandi, M. (2012). A quantitative analysis of current security concerns and solutions for cloud computing. Journal of Cloud Computing: Advances, Systems and Applications, 1(1), 11.

[5] Chiang, R. C., Rajasekaran, S., Zhang, N., & Huang, H. H. (2015). Swiper: Exploiting virtual machine vulnerability in third-party clouds with competition for i/o resources. IEEE Transactions on Parallel and Distributed Systems, 26(6), 1732-1742.

[6] Jain, R., & Paul, S. (2013). Network virtualization and software defined networking for cloud computing: a survey. IEEE Communications Magazine, 51(11), 24-31.

[7] Ahmad, R. W., Gani, A., Hamid, S. H. A., Shiraz, M., Yousafzai, A., & Xia, F. (2015). A survey on virtual machine migration and server consolidation frameworks for cloud data centers. Journal of Network and Computer Applications, 52, 11-25.

[8] Jain, R., & Paul, S. (2013). Network virtualization and software defined networking for cloud computing: a survey. IEEE Communications Magazine, 51(11), 24-31.

[9] Watson, M. R., Marnerides, A. K., Mauthe, A., & Hutchison, D. (2016). Malware detection in cloud computing infrastructures. IEEE Transactions on Dependable and Secure Computing, 13(2), 192-205.

[10] "Venom vulnerability could expose virtual machine on unpatched host systems", (2015, May 13). Retrieved from www.symantec.com on 2016

[11] Ho, M. H., Carminati, B., & Kuo, C. (2014). Network and System Security: 8th International Conference, NSS 2014, Xi'an, China, October 15-17, 2014.

[12] Khan, N., Abdullah, J., & Khan, A. S. (2015, August). Towards vulnerability prevention model for web browser using interceptor approach. In IT in Asia (CITA), 2015 9th International Conference on (pp. 1-5). IEEE.

[13] Khan, N., Abdullah, J., & Khan, A. S. (2017). Defending Malicious Script Attacks Using Machine Learning Classifiers. Wireless Communications and Mobile Computing, 2017.