Virtual Firewall Security on Virtual Machines in Cloud Environment

Gladman Jekese, R.Subburaj Professor, Chiedza Hwata

Abstract —Virtualization is revolutionizing how information technology resources and services are used and managed and has led to an explosive growth in the cloud computing industry, illustrated by Google's Cloud Platform and Amazon's Elastic Cloud. It brings unique security problems such as virtual traffic, denial of service and intrusion, resulting in penetration of virtual machines, which is disastrous for the enterprise, the user and the cloud provider. Virtual traffic between virtual machines may never leave the physical host hardware; making traditional physical firewalls hopeless to monitor and secure it. This paper proposes a virtual firewall which allows managing the network security of the virtual infrastructure per-virtual machine basis, defining network traffic rules, and hardening the security of the virtual environment. A private cloud is designed using open source solutions and to manage the firewall rules, we implement a Tree-Rule firewall technique which filters packets in a tree-like way based on their attributes such as IP address and protocols. The speed of filtering and processing packets on virtual firewall is highly improved to avoid overload of the firewall in the particular case. It permits to log and analyze network traffic logs for each of the monitored virtual machines. The virtual firewall will provide the power to control the bandwidth utilization of each virtual machine in the infrastructure, preventing overutilization and denial of service to critical applications.

Index Terms— virtual firewall, hypervisor, virtualization, virtual machine, tree-rule firewall, stateful firewall, virtual traffic

1 INTRODUCTION

IN cloud computing multi tenancy is problematic, especially with public clouds. For example, co-locating two competing companies in the same physical server can raise some privacy concerns unless tenants are properly isolated from each other [1]. Also hybrid clouds introduce inter-cloud communications a security threat that has to be properly authenticated and protected to avoid abuse. In this paper, we propose a virtual firewall to improve security for virtual machines (VMs) in cloud environment.

Virtualization permits not only to reduce costs (electrical, space, hardware) by lowering the number of physical machines, but it also eases the management of an ever-growing number of computers and servers. VM systems can be categorized into two groups: Type I VMMs or Type II VMMs as shown in Figure 1. A Type I VMM runs directly on the physical hardware and a Type II VMM runs as an application in a normal operating system. However, virtualization is both an opportunity and a threat [2], [3]. The process of collapsing multiple servers into a single one, with several VMs inside, result in eliminating firewall and other protections in existence prior to the virtualization. Traditional physical security measures literally become blind to traffic between VMs since the virtual network traffic may never leave the physical host hardware [4], [5].

- *Gladman Jekese is an M.Tech student in Information Technology at SRM University, Chennai, India, E-mail: jgman86@gmail.com*
- *Dr. R Subburaj is a Professor and Consultant in the Department of Information Technology at SRM University, Chennai., India, E-mail: subburaj.spr@gmail.com*
- *Chiedza Hwata is an M.Tech student in Information Technology at SRM University, Chennai, India, E-mail: chiedza11@gmail.com*

One way to secure and monitor VM-to-VM traffic is through routing the virtualized network traffic out of the virtual network and onto the physical network via Virtual Local Area Networks (VLANs), and hence into a physical firewall already present. The VLAN traffic could be monitored and filtered by the physical firewall and then passed back into the virtual network and on to the target virtual machine. This process also presents risks to the hypervisor so it might be more efficient to keep the traffic entirely within the virtualized environment and secure it from there [6], [7]. The solution to this issue is the use of virtual firewall.

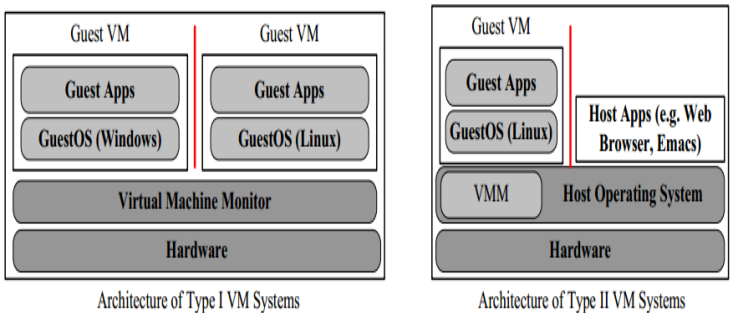

Fig 1. Type 1 and Type II VMM Architecture

1.1 Traditional Firewalls

A firewall puts a barrier that controls the flow of traffic among domains, hosts and networks. A firewall is usually placed between the public internet and a private and trusted network. They are a combination of hardware (network switches and routers) and software that deals with network packets according to a given set of rules (firewall policy) that is the security policy. Conceptually there are three types of firewalls which are static, dynamic and application-layer firewalls that generally apply different filtering rules. There have been many studies about firewall rule conflicts (anomalies) that occur within rule sets. The traditional

firewalls filter packets by comparing their header information against a set of pre-defined firewall rules. Packets are filtered in a sequential order, starting from the first rule until there is a matching rule. If there is no matching rule found, the packet is processed by the default rule. This process operates rule by rule until a specific condition is reached. The rules in traditional firewalls are a list of condition statements followed by actions which are shown in Table 1 [8].

Table. 1 An example of rule in Traditional Firewalls

Rule	Proto.	Source_IP	Destination_IP	Destination_Port	Action
1	TCP	192.168.1.1	54.251.1.1	80	ACCEPT
2	TCP	192.168.1.2	54.251.1.1	80	DENY
3	TCP	192.168.1.*	54.251.1.1	80	DENY
4	TCP	192.168.1.3	54.251.1.1	80	ACCEPT
5	TCP	192.168.2.*	54.251.2.3	80	DENY
6	TCP	192.168.2.4	54.251.2.*	80	DENY
7	TCP	192.168.3.*	54.251.3.5	80	ACCEPT
8	TCP	192.168.3.6	54.251.3.*	80	DENY
9	Any	Any	Any	Any	DENY

Basically there are key issues with the traditional firewalls such as:

- Security problem, caused by potential shadowed rules and the change of meaning of the rule policy due to rule repositioning,
- Functional speed problem, raised by shadowed rules redundant rules and sequential rule matching, and
- Difficulty in rule design, in which one needs to carefully choose the proper positions for the firewall rules in order to avoid listing 'bigger rules' before 'smaller rules'[8].

1.2 Virtual Firewalls

Virtual firewalls are the linchpin of enterprise security in cloud environment and are the most widely adopted technology for protecting virtual private networks. An error in a firewall policy either creates security holes that will allow malicious traffic to sneak into a virtual private network or blocks legitimate traffic and disrupts normal business processes, which, in turn, could lead to irreparable, if not tragic, consequences.

A virtual firewall is a firewall service running in a virtualized environment and providing the usual packet filtering and monitoring services that a physical firewall would provide. Virtual firewalls enable the use of network access controls between VMs and other points in virtual and physical environments. Virtual firewalls are deployed within the fabric of the virtualization environment. A firewall can be implemented on cloud environment as a service or appliance. If a software firewall is applied, the firewall position is as shown in Fig. 1(a) [9]. It can be a purpose-built virtual security appliance and the firewall position differs from what is shown in Fig. 1(a) [10] and the firewall is not in the hypervisor although the levels of network security remain the same. The firewall positioning in Fig. 1(a) has its own benefits because it does not consume much resource as only a single firewall computer is required. However, this positioning still faces security problems because a virtual machine (VM) behind the firewall may be attacked by other situated in the same domain. Therefore, to upgrade the security level, one proposal is to reposition the firewalls as shown in Fig. 1(b) [9]. However, this positioning consumes much more resource (e.g., disk space, Random Access Memory (RAM), and hypervisor's Central Processing Unit (CPU)). To resolve this problem, the firewall can be a managed kernel process running within the host hypervisor that sits atop all VM activity as shown in Fig. 1(c) [9] and this proposal consumes less resource but requests more rules in the firewall than the one shown in Fig. 1(a). The added rules are the ones for preventing the attacks between VMs. To overcome the problems, a virtual firewall is presented to support the third model (Fig. 1(c)) and evaluate its performance.

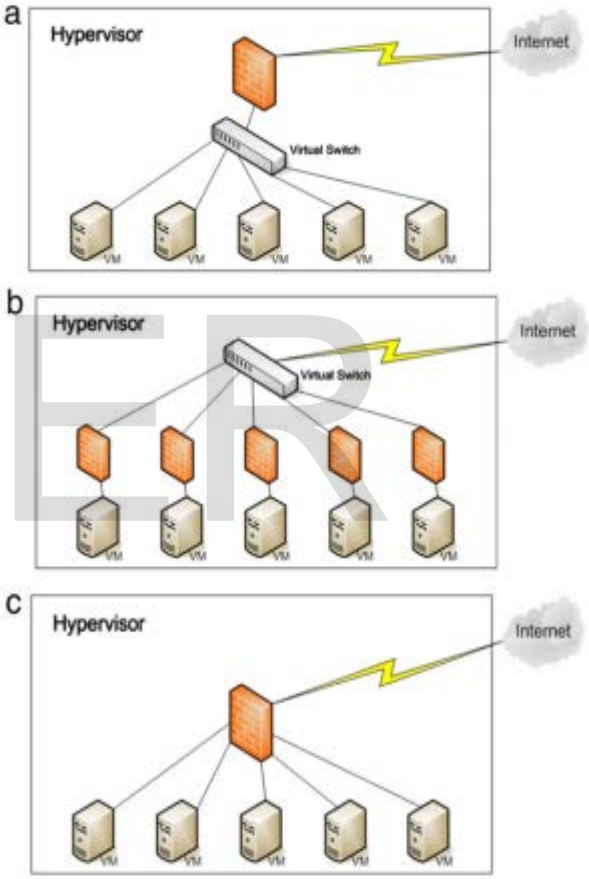

Fig 2. Firewall models in cloud environment

The rest of this paper is organized as follows. In Section 2 we review previous studies on (virtual security) and firewall optimization mainly from a theoretical view of point. In Section 3 we show the methodology and design of the firewall from a practical view of point and section 4 the implementation details of the virtual firewall. Finally in Section 5 we conclude with an outline of possible improvements.

2 Related Work

As a recommendation to understand virtual firewalls, [11],

[12], provides an overview of virtual firewalls and firewall policy focusing on both network service access policy and firewall design policy. They did not look at the implementation of the firewalls, nor do they use any particular products or security architectures or models. Their overlooked the performance of the virtual firewall in the cloud environment an issue that we are going to look at in this paper.

R Schwarzkopf suggested the use of software updates and scanning of VMs in either virtualized Grid or Cloud computing environments for known security vulnerabilities through the use of an Update Checker and an Online Penetration Suite [13]. An image management system, called Mirage, was presented by Wei et al. that address security concerns of a virtual machine image publisher, customer and administrator [14].

Denz and Taylor state that the use of a large number of nearly identical processors in open source cloud computing acts as a vulnerability amplifier: a single vulnerability being replicated thousands of times throughout the computing infrastructure. Malware prevention and detection, secure virtual machine managers, and cloud resilience are the ways being used but these approaches detect known threats rather than mitigation of new or zero-day threats, which are often left undetected. They proposed leveraging the strengths from each technique in combination with a focus on increasing attacker workload which would make malicious operation time consuming and deny persistence on mission time-scales. This could be accomplished by incorporating migration, non-determinism, and resilience into the fabric of virtualization [15]. Through incorporating their security mechanism with the virtual firewall, security can be improved in the cloud.

CloudVisor uses nested virtualization to deal with the compromise of the hypervisor [16]. In this technique a secure hypervisor is introduced below the traditional hypervisor and the interactions between the traditional VMM and VMs are monitored by the secure hypervisor. However, the compromise of VMM can impact the operation of the VMs. The technique proposed in [17] allocates a separate privileged domain for each tenant. However the model can become more complex as different tenant VMs can be hosted on the same physical server. Furthermore, such models cannot deal with the case of malicious tenants that misuse the cloud resources to generate attacks on other hosts. Also a secure hypervisor alone cannot guarantee security without the help of other protection measures such as virtual firewall.

Currently there is significant interest to develop security tools based on virtualization technology [18], [19], [20]. Dunlap et al., [19] proposed ReVirt architecture for secure logging by placing the logging tool inside the VMM. ReVirt logs detailed information such as user inputs and system calls, which enables the administrator to replay the execution of virtual machine. Garfinkel [18] proposed a Livewire intrusion detection system which makes use of the virtual machine monitor to analyze the state of VMs and detect attacks. Lycosid [20] detects hidden process in the VMs by comparing the implicit guest view with the VMM image. However attacks can be generated by non-hidden processes and these virtualization techniques cannot be directly applied to the cloud environment due to the semantic gap problem. As the semantic gap increases the number of false alarms increases.

Some public cloud providers have provided ways to apply a set of static firewall rules to a cloud instance when it is provisioned. For example the IBM SmartCloud Enterprise (SCE) allows for such firewall definitions to be included in the "parameters.xml" [21] that is associated with the cloud image that is used to create the instance. The Amazon Elastic Compute Cloud (EC2) goes a step further in providing Security Groups [22]. The security groups provide firewall access control at a network level above the actual running instances, and the policies can be modified via APIs and thereby allow for policies to be changed beyond instance creation time. With that, EC2 does not currently allow for instances to be included in more than one security group nor can the security group defined at creation be changed later. Further the notion of authenticated client registration is missing in both of these examples.

Another white paper calls for a novel virtual network framework required in order to improve security of the inter-communication among VMs. [23]. Security architecture is proposed that provides a security as a service model that a cloud provider can offer to its multiple tenants and customers of its tenants [24]. The security as a service model while offering a baseline security to the provider to protect its own cloud infrastructure also provides flexibility to tenants to have additional security functionalities that suit their security requirements. The paper described the design of the security architecture and discussed how different types of attacks are counteracted by the proposed architecture [24].

3 METHODOLOGY AND DESIGN

This section presents the design and implementation of the newly proposed virtual firewall. The basic design is presented and illustrated. First, the cloud infrastructure choice and why it was chosen is outlined. Then the virtual firewall used in this cloud infrastructure is described along with the basic set of filtering rules. We will analyze the benefits of the proposed firewall and improve the basic design and analyze the performance.

3.1 Cloud Infrastructure

In order to assess a virtual firewall in a cloud environment, it is required to choose a cloud platform where the evaluation will take place. Open source software can be used as it is quite easy to install and set up the cloud platform without having to worry about the cost or copyright issues. It also allows any future work or improvement based on this paper much easier. We have used Oracle VM VirtualBox as the virtual machine monitor and OpenNebula as a cloud toolkit used to manage and build the cloud infrastructure.

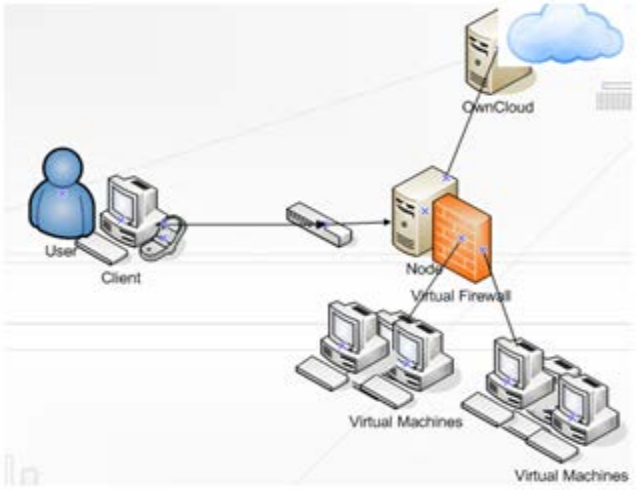

Fig 3. A small size network showing the cloud infrastructure to be implemented

Oracle VM VirtualBox is the popular cross-platform virtualization software that enables multiple operating systems to run on one desktop. It allows teleportation of running VMs between hosts without interruption and support for massive workloads of up to 32 virtual CPUs. VirtualBox is not only an extremely feature rich, high performance product for enterprise customers, but it is also a professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2 [25]. The machine running the VirtualBox hypervisor contains three components: Oracle VM VirtualBox Hypervisor, Domain0, the privileged domain and DomainU, the unprivileged domain guest.

OpenNebula is an open source cloud toolkit that permits to build and manage any type of cloud Infrastructure and is simple to install, update and operate by the admins, and use by end users It can interact with multiple different hypervisor such as Xen, KVM, or VMWare ESX. OpenNebula integrate and works with existing technologies such as MySQL, Ceph, LVM, GlusterFS, Open vSwitch, Ceph, LDAP and this allows delivering a light, flexible and robust cloud manager For ease of evaluation, it has been used to implement a private cloud so both clients and virtual machines will be localized on the same site. In order to implement a private cloud infrastructure, two machines will be required: a **frontend** and a **node**. The frontend is used to install the cloud toolkit (OpenNebula) while the node is used to install the Oracle VM VirtualBox hypervisor in charge of the VMs. The operating system of these machines will be Windows as it fully supports Oracle VM VirtualBox and OpenNebula. In terms of hardware requirements, the node should present high CPU performance in order to support virtualization. Also, its processor should support virtualization (AMD-V or Intel-VD processor). Once the platform is in place, Windows images are installed on the node using Oracle VM VirtualBox and deployed for users using OpenNebula.

3.2 Virtual Firewall

The design of the proposed 'Virtual firewall' is shown in Fig. 3. This design can avoid the limitations of static (current) firewall. In this subsection, we will explain the advantages of virtual firewall including:

- Dynamic (virtualization-aware) operations
- Ease of management
- Multi-tenant support
- Cost-effectiveness

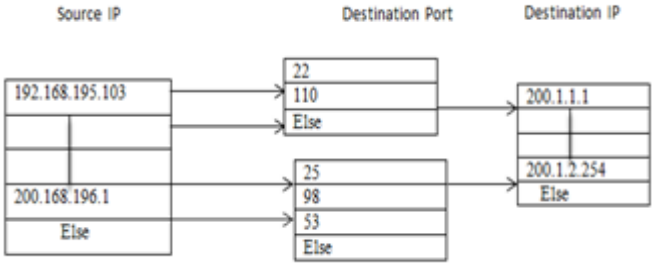

Fig 4 Basic virtual firewall structure

The virtual firewall is a new kind of firewall in which the rules are dynamically presented instead of statically or manually presented. Once the Hypervisor is installed, the privileged domain (dom0) is not only responsible of the guest management but also of bridging the different unprivileged domain (domU) to it. For each new domU instance, Hypervisor creates a new pair of connected virtual Ethernet interfaces with one end in dom0 and the other in domU. The firewall uses IP addresses, protocols and ports to filter network traffic; it can also track the state of a connection of the flow that pass through it and therefore is considered as a stateful inspection firewall.

This firewall will read attribute information from packet and compare packet's attribute with the data in the root nodes in the firewall. After that, the firewall will check packet's other attributes in order by searching only on relevant nodes at the corresponding levels. As a result, the packet will be decided quickly with a specific action. For example, when packets arrive at the virtual firewall, the firewall will consider destination IP address, Source IP address, destination mac address, respectively in order until packets' access decisions are made by predefined actions. Actually, an attribute within the root node can be Source IP, source mac address, or any attribute suitable to work with the firewall rules. Users can select attributes that they want for each column before creating firewall rules. For example, if we focus on permissions for users, we should specify Source IP to be the root node to allow where (destination IP addresses) they (Source IP addresses) want to go. We have created a Graphic User Interface (GUI), a rule editor, where users (administrators) can specify attributes for each column easily. In this paper, we use source IP (for the permission of users) as the root node and destination IP for server protection since they are important source of information [26].

3.3 Improvement of the basic design

Usually computers in the same network need to be protected with the same policy. In the basic design of firewall there is the use of a single IP address corresponding to the number of user's computers. Each line is linked to some sub-trees with repeated (the same) data. Based on the tree-rule firewall design [26], we add another column in the tree structure which is the Mac Address. The firewall has to check the source IP, source mac address, port number, and the destination IP address. This will increase the processing time of the firewall since an extra attribute has been added. In the Tree-rule firewall design the time complexity is in the order log (N). So for the Tree-Rule firewall in Fig. 3 (where Dest IP has 4 lines, Dest Port has 4 lines (on average), Source IP has 2 lines (on average)and Source Mac has 2 lines (on average)) will take a time less than K × Log 4 + K × Log 4 + K × Log 2 + K × Log 2 = K × (2 + 2 + 1 + 1) = 6K. Note that all 'Log' values above are of base 2.

To get rid of the above-mentioned problems and to improve the basic design, the 'Single IP Address' design (as shown in Fig. 3) is replaced by 'IP Address Range' design and the additional use of Mac address (as shown in Fig. 4). For example IP Addresses between 192.168.195.103–192.168.195.254 apply the same rules. Using the improved design, the memory spaces that were previously requested in the basic design can now be saved. Also to improve the security of the firewall access privileges will be introduced so that users will not access unauthorized data.

We have shown that the improved design can save memory spaces. We now show that the rule searching time increases only a bit because of the change from a single number design to a range design. It is found that the searching time within the node slightly increased as shown below.

$$t \in O(\log_2 N) \text{ is changed to } t \in O(1 + \log_2 N), \quad (1)$$

where N is the row number within a node. A new attribute called 'in-the-line' is added to the data structure of the firewall to help search the data using Binary Search algorithm. For example the firewall will check 'Dest Port' based on the link indicated in the algorithm.

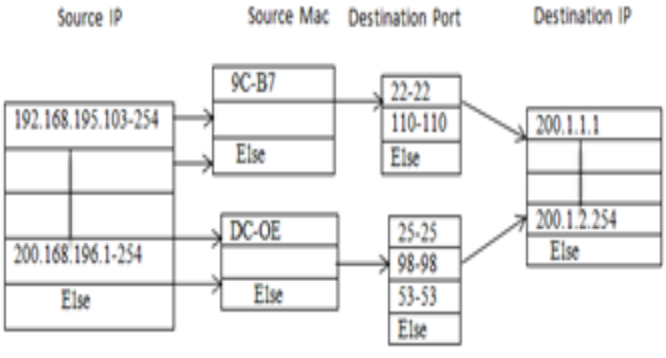

Fig 5 Improved design of the firewall using IP ranges

4 IMPLEMENTATION DETAILS

This section shows the cloud infrastructure implementation, virtual firewall implementation and the client side implementation used to access data in cloud environment.

4.1 Cloud implementation

As outlined in the design section, the node is implemented with the Oracle VM VirtualBox Hypervisor to create and manage VMs. The characteristics of the node machine are the following: 2.66 Ghz Intel Quad Core i3 processor, 64-bit platform and 3 GB of RAM.

The Oracle VM VirtualBox is installed on top of the Windows 7 host operating system. Once the VMM is installed it is now possible to create VMs directly from the node, however because the idea is to implement a private cloud, we want the creation of VMs to be managed by the frontend with OpenNebula. A 60Gb virtual hard disk is created as well as a virtual CD-ROM drive pointing to a Windows 7 installation image. With this, we will be able to copy the virtual hard disk image during the Frontend implementation so we can easily deploy VMs from there. Once the node is ready to run VMs, it is important to setup a network bridge so these VMs can communicate on the network. On the client side users will connect to their VMs using http service in order to access data in the cloud environment.

4.2 Virtual Firewall implementation

The virtual firewall implementation is conducted on Oracle VM VirtualBox with the virtual firewall operating in the Kernel space. Similar to IPTABLES, we focus on the network firewall that verifies the requests or packet forwarding between the network interfaces. The algorithm includes three programs as follows:

- Virtual firewall. It is written with Visual C# language on Windows in order to detect the packets and make a decision whether the packets or request should be accepted or dropped. This program runs on the Kernel Space.
- Rule Sender. This is written with C# language on Linux in order to receive rules from GUI (running on the user's Windows XP/7/8). Rule Sender would send the rules to Virtual Firewall through 'procfs Virtual File System', a specific memory for the data exchange between the regular software and the software functioning on Kernel. This Rule Sender runs on the User Space (not the Kernel Space).
- GUI. It is written with C# language on Windows in order to communicate with administrators so that each administrator could create a rule. After creating and editing the rules, the user can either save or send/apply the rule to the firewall so that the rule can be functioned. GUI will communicate with the 'Rule Sender' on the firewall.

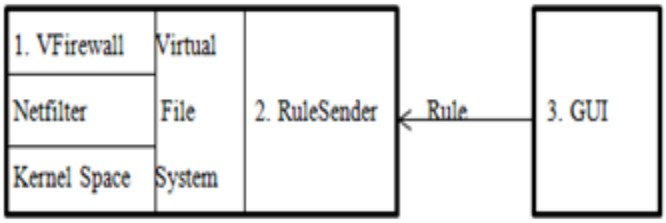

Fig 6 Implementation of Virtual Firewall

5 CONCLUSION AND FUTURE WORK

In this paper, a brief summary of virtual firewalls and security mechanism implemented in different cloud environment is given. The basic design of the virtual firewall which includes cloud infrastructure and the virtual firewall as well as the deployment of virtual machines and implementation process is outlined. The virtual firewall will be deployed in the VMM, and the filtering procedure will be based on packet attributes and host information. In the next study we will increase the filtering procedures, protocols and evaluate the performance of the firewall.

REFERENCES

1. M. Komu, M. Sethi, R. Mallavarapu, H. Oriola, R. Khan, S. Tarkoma, "Secure Networking for Virtual Machines in the Cloud", Department of Computer Science and Engineering, Aalto University, 2011.
2. X. Zhao, K. Borders, A. Prakash, "Virtual Machine Security Systems", Advances in Computer Science and Engineering, pp. 340-341, 2009.
3. K. J. Higgins. VMs create potential risks. Technical report, dark READING, 2007.
4. http://www.darkreading.com/document.asp? doc_id=117908.
5. Shields, Greg, "Why Hyper-V virtual networks are less secure than physical networks". TechTarget SearchNetworking, 2009.
6. Rosenberg, David, "Security considerations for virtual environments". Cnet News Nov 2009.
7. "Software-Based Access Management Protects Mixed Networks of Virtual and Physical Machines without Complex Rule Sets and High IT Overhead" Apani Inc. Aug 2008.
8. "Secure Virtualized Hosting" Altor Networks Inc.
9. T Chomsiri, X He, P Nanda, Z Tan, "A Stateful Mechanism for the Tree-Rule Firewall", Center for Innovation in IT Services and Applications (iNEXT), IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 122-123, 2014.
10. Virtual firewall appliances: trust misplaced 2012. http://blog.cloudpassage.com/2012/01/24/virtual-firewall-appliances-trust-misplaced/.
11. VMware virtual switch isolation, 2013. http://serverfault.com/questions/186735/vmware-virtual-switch-isolation.
12. Overview of Virtual Firewalls on VBlock Infrastructure Platform, June 2012.
13. R Hunt, "Internet/Intranet firewall security—policy, architecture and transaction services", Department of Computer Science, University of Canterbury, New Zealand, Computer Communications 21 pp. 1107–1123, 1998.
14. R Schwarzkopf, M Schmidt, C Strack, S Marti, B Freisleben, "Increasing virtual machine security in cloud Environments", Journal of Cloud Computing: Advances, Systems and Applications2012,1:12 http://www.journalofcloudcomputing.com/content/1/1/12. Downloaded on
15. Wei J, Zhang X, Ammons G, Bala V, Ning P, "Managing Security of Virtual Machine Images in a Cloud Environment", In Proceedings of the ACM Workshop on, Cloud Computing Security, CCSW, 2009.
16. R Denz and S Taylor, "A survey on securing the virtual world", Journal of Cloud Computing: Advances, Systems and Applications, 2013. http://www.journalofcloudcomputing.com/content/2/1/17.
17. F. Zhang,et al., "CloudVisor: retrofitting protection of VMs in multi-tenant cloud with nested virtualization", in Proc. Symposium Operating System Principles, 2011.
18. C. Yu,et al., "Protecting the security and privacy of the virtual machine through privilege separation," in Proc. Int. Conf. Comput. Sci. Electron. Eng., 2013.
19. T. Garfinkel and M. Rosenblum, "A virtual machine introspection based architecture for intrusion detection", Network Distribution System Security Symposium, 2003.
20. G. W. Dunlap, "ReVirt: Enabling intrusion analysis through virtual-machine logging and replay", inProc. Operating Syst. Des. Implementation, 2002.
21. S. T. Jones,et al., "VMM-based hidden process detection and identification using lycosid," in Proc. ACM Virtual Execution Environments, 2008.
22. IBM SmartCloud Enterprise Release 2.1, Customizing images and software bundles, Operation.
23. Amazon Elastic Compute Cloud User Guide (API Version 2012-08-15).
24. http://docs.amazonwebservices.com/awsec2/latest/userguide/usingnetwork.
25. H. Wu, Y. Ding, C Winer, L Yao, "Network Security for Virtual Machine in Cloud Computing", National Natural Science Foundation of China, 2009.
26. V Varadharajan, U Tupakula, "Security as a Service Model for Cloud Environment", IEEE Transaction on Network and Service Management, Vol.11, No.1, pp.60-75, 2014.
27. https://www.virtualbox.org/.
28. X. He, T. Chomsiri, P. Nanda, Z. Tan, "Improving cloud network security using the Tree-Rule firewall", School of Computing and Communications, Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia, 2013.