Symbiosis

Research Article | Journal of Computer Science Applications and Information Technology | Open Access

Received: June 16, 2017
Accepted: July 18, 2017
Published: August 11, 2017

Automated Live Migration in OpenStack:A Moving Target Defense Solution

Fahad Polash and Sajjan Shiva*

Computer Science Department, The University of Memphis, Memphis, USA

***Corresponding author:** Sajjan Shiva, Professor, Computer Science Department, 335 Dunn Hall, The University of Memphis, Memphis, TN 38152-3240, USA, Tel: (901) 678-5667; Fax: (901) 678-1506, E-mail: sshiva@memphis.edu

1 Abstract

Cloud computing has become one of the most promising fields in information technology in recent years. Many big players such as Amazon, Google, and Microsoft are playing vital roles in the proliferation and adoption of cloud technologies. Besides them, the availability of free open source Cloud platform is also helping the spread of cloud computing adoption greatly. OpenStack is a free open source Cloud computing software originally released by Rackspace and NASA. Security issues have always been concerns for the cloud customers. Live migration of virtual machine instances from one physical machine to a different physical machine could be a potential for security threats in cloud computing. The live migration technology in the domain of OpenStack cloud can be regarded as Moving Target Defense (MTD) as the targeted physical machine is offloading its resident virtual machine instances to a different physical machine. Our contribution in this paper is the implementation of automated live migration in OpenStack cloud environment triggered by an Intrusion Detection System (IDS) and its performance analysis. We also analyze how it will affect the Service Level Agreement (SLA) between customers and service providers. Our future work is centered on monitoring the migration aspects to detect and prevent hacking.

2 Keywords

Cloud computing; Live migration, OpenStack; Moving target defense (MTD);

3 Introduction

Cloud computing has become a prominent paradigm in recent years. It has gained popularity among the IT world due to its ability to transfer the capital expenditure to operational expenditure [1]. A company can get access to high-end computing infrastructure of clouds by only paying for the duration of usage. It has other advantages as well: On-demand self-service, broad network access, resource pooling, rapid elasticity, etc. [2]. However, the cloud computing has also brought some vulnerabilities in addition to the existing security risks in traditional IT technology. According to Cloud Security Alliance (CSA) report, the number of cloud vulnerability incidents has increased from 33 in 2009 to 71 in 2011 [3]. This security concern responsibility is divided among the cloud users, the cloud vendors and the third party vendor involved in ensuring secure sensitive software or configurations. If the application level security is the responsibility of the cloud user, then the provider is responsible for the physical security and also for enforcing external firewall policies. Security for intermediate layers of the software stack is shared between the user and the operator. In this paper, we mainly focus on the security measures that are taken by the service providers to ensure the customers' data security and service availability as per the SLA.

Cloud Service Providers (CSPs) have become increasingly active in implementing aggressive measures to address the security concerns of the customers. Various types of IDS systems have been implemented and used successfully on high-volume networks to monitor and record activities in order to detect potential intrusions, malicious activities, or policy violations. In our implementations, we have deployed Snort which is an open source network-based intrusion detection system and capable of analyzing IP packets in real- time. Any anomaly detected by Snort will trigger the migration of all virtual machines from the victim physical server.

Live migration itself is not secure enough in cloud computing environment. Duncan, et al. [4] has identified migration attack performed by the insider attacker as a very real threat to the security and integrity of customers' data. However, if live migration is performed in a trusted secured environment, it could be considered as a moving target defense strategy against several types of attacks (e.g. DoS attack). The rest of the paper is orga-

Citation: Polash F, Shiva S (2017) Automated Live Migration in OpenStack:A Moving Target Defense Solution. J Comp Sci Appl Inform Technol.2(3):1-5. DOI: 10.15226/2474-9257/2/3/00119

nized as follows: Overview of Live Migration gives an overview of live migration, Moving Target Defense Strategies describes moving target defense strategy, Implementation of Mtd in Openstack describes the implementation of MTD in OpenStack environment, Performance analysis section analyses the performance of MTD and finally conclusion section concludes and provides direction for future research.

4 Overview of Live Migration

Live migration of virtual machine is the process of moving virtual machines from one physical machine to another without any impact on virtual machine availability to users. The main purposes of live migration include: 1) Load balancing: In networked environment, if any server machine is becoming overloaded with virtual machines, then some of the virtual machines can be migrated from it to reduce the load. 2) Maintenance: If any maintenance activity is required in the physical machine which demands down time (e.g. Operating System upgrade, network configuration changes etc.) then, the existing virtual machines running on that machine could be moved to another physical machine and thus it can ensure the availability of those virtual machine to users. 3) Power Management: If some of the physical machines are being underutilized, then the virtual machines running on them could be live migrated to another machines and the previous machines could be shut down to reduce power consumption. 4) Fault Tolerance: if at any point of time, the server starts malfunctioning, the running virtual machines can be migrated to another machine and the dysfunctional server can be investigated to find the reason of its misbehavior. In this way, the users of those virtual machines remain unaffected.

Live migration of virtual machine requires the transfer of complete state of VM from source to destination. The complete state comprises all the resources the VM uses in the source machine. The resources include: permanent storage, volatile storage, connected devices (e.g. LAN Cards), internal state of the virtual CPUs (VCPU). Normally, in data centers, the permanent storage is provided by network-attached storage. Thus, it is not required to move the permanent storage during the migration of VM. The internal states of the VCPUs are only a few kilobytes of data, so it does not take considerable amount of time to be transferred. Longer period of time is required to transfer the volatile memory contents, and thus it affects the performance of live migration process. So, more attention is given to improve the transfer of volatile memory from the source to the destination.

There are two prevalent algorithms for migrating virtual machines: Pre-copy and Post-copy algorithm. An overview of both of these algorithms is given below:

• **Pre-copy:** The pre-copy technique first transfers the entire contents of the VMâĂŹs volatile storage to the destination host. Then it iteratively transfers the dirty pages to the destination host. It continues to do so until the number of dirty pages falls below a threshold or the pre-defined maximum number of iterations has been reached. After that, the VM is stopped and along with the remaining dirty pages, the state of VCPUs and other devices are sent to the destination host. Then the VM is resumed in the destination. The performance of this algorithm depends on how frequently the pages get dirtied in the source while it is getting prepared to be migrated. In case, the VM is running write-intensive application, the VM migration time will be longer and the user will get prolonged down time of their services. But for read-intensive application, this algorithm performs better.

• **Post-copy:** In post-copy technique, the VM at the source is stopped first. Then a minimal subset of the execution state of the VM (CPU state, registers and, optionally, non-pageable memory) is transferred to the target. After that, the VM is restarted in the destination host. At the same time, the source host starts pushing the remaining memory pages to the destination host. This is called pre-paging. If the VM at the destination tries to access a page which is yet to be transferred to the destination, a page-fault occurs. The VM waits for the page from the source. If the VM contains a read-intensive application, then there would be too many page faults and the customers would experience a significant degradation in their services. That's why, post-copy algorithm is better for write-intensive application, and worse for read-intensive application.

The performance of live migration of VM is measured by four metrics. They are as below:

• **Downtime:** It denotes the time period during which the VM is completely shut down.

• **Total migration time:** It is the time period between the start of live migration until the resource of the source host are released.

• **Time-to-responsiveness:** represents the time span after the resume phase has ended until the VM achieves a certain guaranteed minimal utilization.

• **The amount of transferred data:** It measures the amount of data received at the destination host from the different sources However, the live migration of virtual machine has other costs as well. For example, energy consumption increases during the live migration.

5 Moving Target Defense Strategies

Information technology systems are built to operate on static environment. This static nature gives attackers the advantage of time. That is, the attackers get ample time to study the target system. Then they can perform the attack in a good time so that they can go unnoticed. The attackers maintain back doors without being discovered for a long period of time. However, to overcome these situations, a new defense strategy has been adopted in information technology, which is regarded as Moving Target Defense (MTD). The concept came from the strategy of battle field. In case of battle field, sometimes we need to change our place so that the enemies get confused and find difficulties to attack us. Similarly, in information technology systems, if you move the target system, then it will increase the uncertainty and complexity for the attackers.

Different techniques and strategies are applied in moving target defense [7]. Some of them are discussed below:

• **Software-based Diversification:** Software diversification indicates addition or deletion of non-critical functionalities in the software. This technique will eliminate the vulnerabilities or limit

Citation: Polash F, Shiva S (2017) Automated Live Migration in OpenStack:A Moving Target Defense Solution. J Comp Sci Appl Inform Technol.2(3):1-5. DOI: 10.15226/2474-9257/2/3/00119

their exposures. Input validation and rectification is also regarded as the diversification of software.

• **Runtime-based Diversification:** Many defense techniques mitigate attacks via introducing diversification in runtime environments. Address Space Layout Randomization (ASLR) is a widely known example of Runtime-based Diversification techniques. In ASLR, the addresses of code, data, stack and heap are assigned randomly. This randomization will make the attackers difficult to go to their desired address.

• **Communication Diversification:** This technique is mainly for combatting the network related attacks. For example, changing the database schema or changing the database table title would make attacks difficult for the attackers. Again, changing the IP address of the target system and changing the port number and then share the changed IP address and port number will also make attacks difficult for the attackers. The attackers will not be able to scan or exploit the systems easily.

• **Dynamic Platform Techniques:** Dynamic Platform Techniques (DPT) change platform techniques to stop attacking processes. For example, if any hypervisor is running multiple VMs and if it detects any anomaly or attack, it can transfer all the VMs to a safer machine. Sometimes, there might be redundant VMs so that the hypervisor can rotate the services running in a VM to an idle VM. Saidane, et al. [8] propose the deployment of redundant servers with diverse software in a web system.

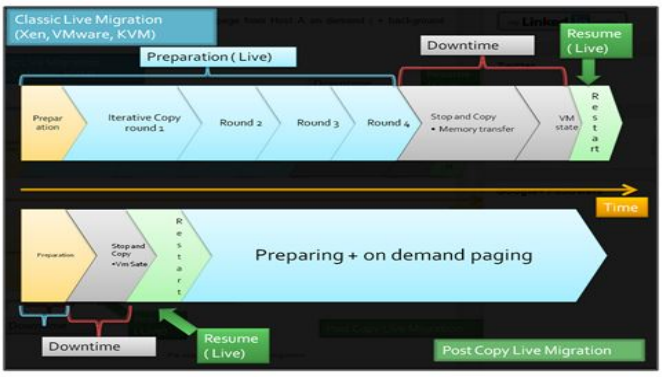

Figure 1: Pre-copy (Classic Live Migration) and Post-copy Live Migration [6]

6 Implementation of MTD in Openstack

In order to perform the experiments, we have set up a cloud environment with OpenStack. In our test bed, we had one controller node and two compute nodes. All the nodes are Dell Optiplex 960 machine with 2GB RAM, 160GB Hard Disk, and 3.00GHz processor. Each of the nodes has two gigabit Ethernet card. One of the Ethernet network interface is connected to all other nodes via switch. The other Ethernet interface is connected to internet with public IP address. The network diagram of our implementation is given in Figure 2. All the nodes are running on the operating system Ubuntu 12.4 LTS server. The IP address plan we have used for our experiments is given in Table 1.

In order to implement MTD, we have implemented live migration of VM with OpenStack as directed by the guidelines [9]. Net-

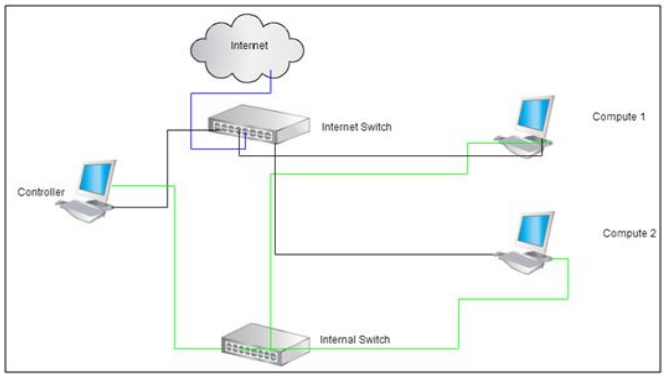

Figure 2: Network Diagram of Test bed

Table 1 IP address plan for Test Bed

Node	Public IP Address	Private IP Address
Controller	141.225.11.199	10.0.0.11
Compute 1	141.225.11.190	10.0.0.31
Compute 2	141.225.10.88	10.0.0.32
Instances	141.225.11.191-198	NA

work Attached Storage was implemented among the three nodes via the local area network. In our OpenStack implementation, the software does not support automated live migration. Only the user with admin privilege can issue the command for live migration either from command prompt or from the horizon dashboard webpage. So, in order to automate the live migration we have written a Java application which will be running in controller node, and any time it gets any notification from any of the compute nodes for possible anomaly detection, it will automatically give instruction to the victim compute node to live migrate all the running VMs to other compute node. The anomaly will be detected by Snort, an intrusion detection system. We can configure different types of rules in Snort to trigger alert in compute nodes. The alert will be logged in a file in compute node. Our client monitor application will be installed in compute node which will monitor the alert file generated by Snort in the compute node. Once it detects an alert, it will immediately inform the controller. In our implementation (Figure 3), the client application will be continuously monitoring the alert file every two seconds. However, we can vary it as per demand.

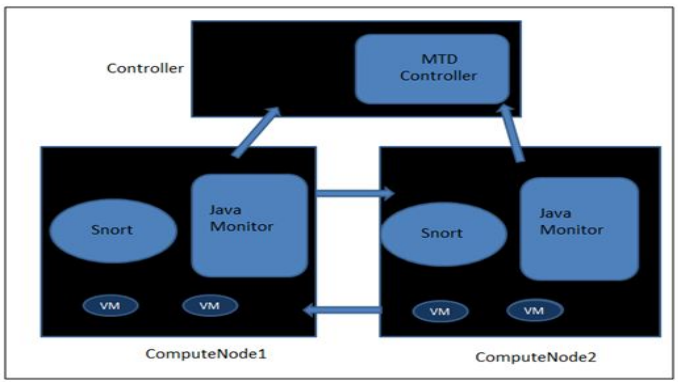

Figure 3: MTD in cloud environment

Citation: Polash F, Shiva S (2017) Automated Live Migration in OpenStack:A Moving Target Defense Solution. J Comp Sci Appl Inform Technol.2(3):1-5. DOI: 10.15226/2474-9257/2/3/00119

7 Performance Analysis

The OpenStack cloud has five different flavors of instances. They are: 1) Tiny, 2) Small, 3) Medium, 4) Large and 5) Extra-large. These flavors differ in RAM, Hard Disk and CPU capacity. In our experiments, we have measured three metrics for the performance analysis: down time, total migration time and total amount of data transferred. The corresponding graphs are given in Figure 4, Figure 5 and Figure 6 respectively.

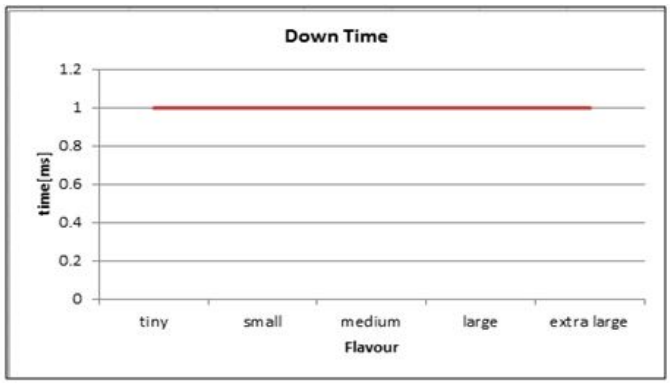

Figure 4: Down time

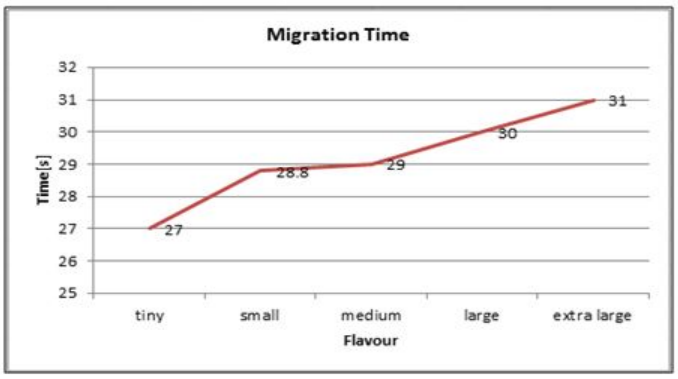

Figure 5: Migration time

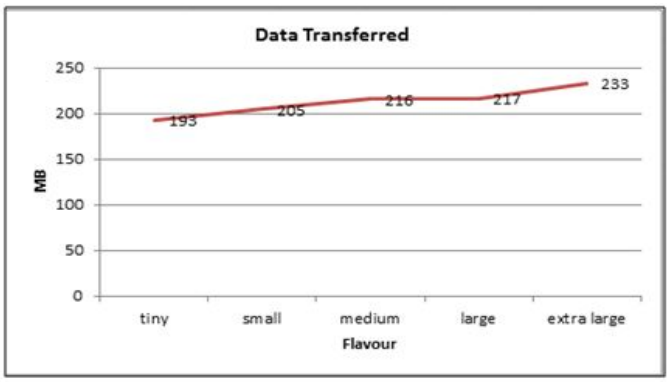

Figure 6: Amount of data transferred

From Figure 5, we can observe that the migration time increases with the size of the VM. This is obvious because in case of large VM, it will take longer time to transfer the memory. The same statement is true for the amount of data transferred over the network. So, certainly the live migration will have impact on SLA enforcement. It will affect the availability of services for customers. However it will also have positive impact in case any anomaly activity really takes place in the network. So, whenever the cloud service provider will sign the SLA with the customers, they should consider the down time of live migration and also the extra load on network for transferring the VM.

From Figure 4, we observed that the downtime is similar for all flavors. Actually, we were pinging the VM while it was being migrated, but we did not find any packet loss. It is because the VMs were not loaded and the network connectivity between the VMs was not congested and the speed was high enough to move the VM without any downtime. But in reality, it is not possible to get a migration with zero downtime. So, we have considered it as one millisecond for our experiments. In our implementation we have two compute nodes. So, whenever the client monitor detects any anomaly, the VMs are being migrated to other compute node. But in real field, there might be many more compute nodes. In that case, it would be challenging to choose the physical machine to transfer the VMs of the affected machine. Also, instead of only Snort, we can apply game theory to detect anomaly. Shandilya, et al. [10] have presented a game theoretic model to detect anomaly in the network. We can also apply game theory to choose the best destination physical machine.

8 Conclusion

In this paper, we have discussed the applicability of MTD in combatting the security threats in the OpenStack cloud environment. The MTD is a new strategy in the field of cyber security defense techniques. We have proposed to live migrate the virtual machine automatically to a safer physical machine in order to perform MTD. However, live migration in OpenStack is still immature and not secure. If live migration could be made in a secure environment, then it would be a great defense technique in cyber security space. So, additional research attention is required to make the live migration secure in OpenStack and other environments.

We are now investigating the use of multiple monitors during the migration to retain the system security [11]. In future, we would also like to look into the possibilities to apply game theory to choose the best physical machine and mechanisms for migrating VMs and analyze the corresponding performance issues [12].

9 References

1. Srinivasan S. Building Trust in Cloud Computing: Challenges in the Midst of Outages. Proceedings of Informing Science & IT Education Conference (InSITE). 2014:305-312.

2. Mell P, Grance T. The NIST definition of cloud computing. NIST publication. 2011:800-145.

3. Rashid F. The Dirty Dozen: 12 Cloud Security Threats. Infoworld. 2016.

Citation: Polash F, Shiva S (2017) Automated Live Migration in OpenStack:A Moving Target Defense Solution. J Comp Sci Appl Inform Technol.2(3):1-5. DOI: 10.15226/2474-9257/2/3/00119

4. Duncan A, Creese S, Goldsmith M, Quinton JS. Cloud Computing: Insider attacks on virtual machines during migration. 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. 2013:493-500. DOI: 10.1109/TrustCom.2013.62

5. Jo C, Egger B. Optimizing live migration forvirtual desktop clouds. IEEE 5th International Conference on Cloud Computing Technology and Science. 2013;1:104-111. doi:10.1109/CloudCom.2013.21

6. Reflections of the Void. 2012. Available from:http://www.reflectionsofthevoid.com/2012/01/kvm-post-copy-live-migration-with.html

7. Xu J, Guo P, Zhao M, Erbacher RF, Zhu M, Liu P. Comparing Different Moving Target Defense Techniques. In Proceedings of the First ACM Workshop on Moving Target Defense. 2014:97-107. Doi: 10.1145/2663474.2663486

8. Saidane A, Nicomette V, Deswarte Y. The design of a generic intrusion-tolerant architecture for web servers. IEEE Transactions on Dependable and Secure Computing. 2009;6(1):45-58. DOI: 10.1109/TDSC.2008.1

9.http://docs.openstack.org/icehouse/install-guide/install/apt/content/

10. Ko SR, Lee S, Rajan V. Cloud computing vulnerability incidents:A statistical overview. 2013.

11. Mahfouz A, Rahman L, Shiva S. Secure Live Virtual Machine Migration through Runtime Monitors. IEEE Tenth International Conference on Contemporary Computing. 2017.

12. Shandilya V, Shiva S. On A Generic Security Game Model. International Journal of Communications, Network and System Sciences. 2017;10(7):142-172. DOI: 10.4236/ijcns.2017.107008.

13. ACM-CODASPY 2015. 5th ACM Conference on Data and Application Security and Privacy. San antonio Texas. 2015.

Citation: Polash F, Shiva S (2017) Automated Live Migration in OpenStack:A Moving Target Defense Solution. J Comp Sci Appl Inform Technol.2(3):1-5. DOI: 10.15226/2474-9257/2/3/00119