Software Defined Stochastic Model for Moving Target Defense

Iman EL MIR[1], Ankur Chowdhary[2], Dijiang Huang[2],
Sandeep Pisharody[2], Dong Seong Kim[3], and Abdelkrim HAQIQ[1]

[1] Computer, Networks, Mobility and Modeling Laboratory
FST, Hassan 1^{st} University, Settat, Morocco
iman.08.elmir@gamil.com, ahaqiq@gmail.com
[2] Department of Computing, Informatics, and Decision Systems Engineering
Arizona State University, Tempe, AZ, USA
achaud16@asu.edu, Dijiang.Huang@asu.edu, spishar1@asu.edu
[3] Department of Computer Science and Software Engineering
University of Canterbury, New Zealand
dongseong.kim@canterbury.ac.nz

Abstract. Moving Target Defense (MTD) has emerged as a good solution to deal with dynamic attack surface. The goal is to make it difficult for an attacker to exploit network resources. But it is challenging to provide zero downtime guarantees when performing network rearrangement or when a physical host acts as a single point of failure for virtual servers. In this paper, we introduce Software Defined Networking (SDN) based continuous time modeling techniques to perform virtual machine migration and MTD techniques while maintaining high service availability and system security. This solution will not only increase attackers uncertainty but will also provide low downtime and high availability guarantee for the network.

Keywords: Moving Target Defense, Virtual machine migration, Cloud Computing, Virtualization, High Availability.

1 Introduction

As Cloud Computing has emerged as a prominent computing and storage tool, cloud data centers must be capable of supporting a large number of services, to cater to an ever-growing demand for computation, storage, and networking. Finding a power efficient method to migrate virtual machines from physical server to another physical server in cloud data center is becoming ever more challenging optimization problem in terms of downtime, cost, and performance. Virtual machine migration provides multiple benefits in terms of fault tolerance, high performance to deliver network services, flexibility, and improved manageability. With the advent of Software Defined Networking (SDN), cloud Data Centers have become more dynamic, with fluid boundaries, especially for virtual machines migration. Virtualization technology is recently becoming a core

mechanism in the implementation of cloud computing due to the benefits of abstraction of hardware resources by creating multiple instantiations of operating systems, running concurrently on a single physical server. In addition, it also provides the migration technique [1] which enables the possibility of migration of virtual machine from one physical server to another. The network operators can use SDN platform to have better visibility and control over the network. We plan to use Software Defined Networking (SDN) capable Open vSwitch in Data centre environment so that we can introduce OpenFlow-based command and control in Data Center environment. The network administrator will be able to enforce fine-grained policies for security enforcement using SDN controller. SDN provides several benefits :

- **Enhanced Configuration:** The configuration is one of the key functions in network management. SDN can automatically configure the network devices (i.e. switch, router, firewall, load balancer) from a single point. It offers a dynamic optimization and a programmable configuration of the entire network.
- **Improved Performance:** The centralized algorithms are effective in managing various challenging performance optimization problems including data traffic scheduling; Quality of service support and load balanced packet routing.
- **Reduced Downtime:** SDN based on virtualization of the physical networking devices makes it easy to do some modifications for one device rather than considering all devices. It simplifies the recovery from any failures that can occur by applying saved snapshot of the configuration.

Moving Target Defense is an emergent technique that provides enhanced security based on controlling change across multiple system dimensions. To deal with the limitations static defense, its concept is focused on creation a dynamic attack surface which decreases the likelihood of successful exploit as well as increases the effort required to penetrate the network and launch an attack. It's a potential game changer in cybersecurity concept that improves the resilience of the system against attacks. The SDN based MTD solution will provide benefits such as making reconnaissance difficult for an attacker through constant change of network topology. The change of network topology can induce network level policy conflicts such as a virtual machine migrated from a vulnerable subnet A to another subnet B, may gain access privileges that it is not supposed to have on reconfiguration. Another key feature that centralized command and control in our proposed solution provides is cost/intrusiveness analysis of reconfiguration using stochastic modeling techniques. Migration of a VM may introduce a delay in a network or bring down some critical services. SDN based solution ensures that availability/ downtime is optimized while deploying MTD based network reconfiguration. Most of the existing solutions are passive in nature and do not take into account the state transitions in real time on network reconfiguration, or the rate of change of security state of a particular network. The key contribution of this research work is as follows:

- A stable MTD solution for SDN based cloud data center environment using Continuous Time Markov Chain modeling technique to model system states.
- High availability and low downtime guarantee while implementing a Moving Target Defense solution in SDN environment.

The rest of the paper is organized as follows: Section 2, provides analysis of some Moving Target Defense solutions with special focus on SDN environment. We introduce our system model using CTMC modeling technique and use case analysis in Section 3. The performance parameters that optimize the fitness of cloud data center environment have been described in Section 4. The implementation of our solution in a simulated SDN environment and the Virtual Machine Migration algorithm have been described in Section 5. Finally, we conclude our paper in Section 6 and provide details of the extension to real network environment along with scope for future work.

2 Related Work

When hardware and software are coupled, the network becomes more expensive to maintain. Additionally, it stymies innovation by the users, and hinders application fine tuning by the administrators. For these reasons, SDN was proposed in order to provide more efficient configuration, better performance, and a high flexibility to manage network services through a central control point. These features of SDN allow it to be a good platform for Moving Target Defense solutions. MTD techniques have been devised as a tactic wherein security of a system is enhanced by having a rapidly evolving system with a variable attack surface; thereby giving defenders an inherent information advantage. An effective countermeasure used in MTD is network address switching which can be accomplished in SDN with great ease. In [8], the authors are focused on the problem of reducing the total migration time in SDN. They have proposed to determine the migration orders and transmission rates of virtual machines so as to reduce the total migration time. The migration of virtual machines has a high potential capability in data centers serving an uninterrupted services during the elapsed VM failover time. In [10], the authors described the advantages and disadvantages of some networked MTD techniques. Basically, they presented how to use SDN in order to implement the MTD migrations. Hence their experimental results showed the effectiveness of SDN by significantly increasing in the cost of an attack since the attacker must spend more time in configuration of the attack surface and traffic load. In [11], we have presented an analytic model for an Intrusion Tolerant Cloud Data Center. The model was implemented in SHARPE and the numerical results on system availability metrics were analyzed. The acquired results demonstrated that decreasing the exposure window will improve the intrusion tolerance of a SCIT-based cloud data center. In [12] and, [13] we have presented the previous security architectures, also discussed its advantages and inconveniences. In particular, we are focused on SCIT as an ITS architecture and using Semi Markov process we modeled the preventive maintenance process on top of the existing intrusion tolerance mechanism.

3 Proposed stochastic model based Markov chain

3.1 Model description

The proposed system shown in the Figure 1 represents the Cloud Data Center (CDC) which contains several physical servers . On each host, multiple virtual machines are running. We have also a Load Balancing server to forward the incoming requests users to a specific physical server respecting the size of the queue. The resources in the network are controlled and supervised by the SDN controller. This centralized controller maintains a global view of the network such the topology and the connectivity. It is characterized by its flexibility to install dynamically forwarding rules to investigate multiple forwarding possibilities from source to virtual machine destination. Hence, the network devices are switched through the forwarding tables and using the OpenFlow protocol, the SDN controller connects with the devices to generate a clear information about the network.

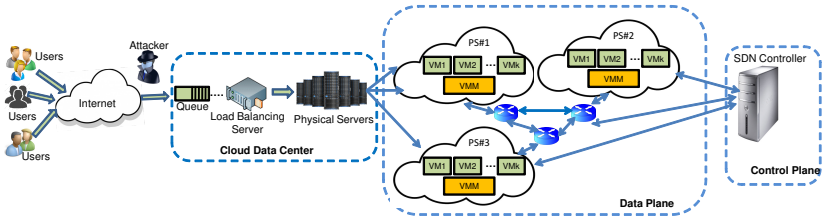

Fig. 1. Description of the system architecture.

3.2 Markov Chain Model

In this subsection, we dwell on the stochastic behavior of the proposed system. We formulate the proposed system with a mathematical model by means of the supplementary variable techniques and probability analysis. However, we consider three states such as R the running state, V the vulnerable state and M the migration state. The state transition diagram is $R \longmapsto V \longmapsto M \longmapsto R$. The virtual node running on the physical server is in a running state R, i.e. it is functioning normally and can protect itself from external malware. But after exposure to the Internet for a certain period of time with regular activity, it will be vulnerable. Thus, the virtual node moves from running state R to vulnerable state V with rate λ. In this case, virtual node transits from vulnerable state V to migration state M with rate β. The migration mechanism from the current physical server to another physical server will be occurred with rate δ.

Figure 2 illustrates the state transition diagram of the proposed system as Markov Chain Process. The steady-state balance equations for the model are as

follows where $\pi_{R_i}, \pi_{V_i}, \pi_{M_i}$ is the steady state probability respectively in Running, Vulnerable, Migration states :

$$\begin{cases} \lambda\pi_{R_1} = \delta\pi_{M_2} + \delta\pi_{M_3} + \delta\pi_{M_4} \\ \lambda\pi_{R_2} = \delta\pi_{M_1} + \delta\pi_{M_3} + \delta\pi_{M_4} \\ \lambda\pi_{R_3} = \delta\pi_{M_2} + \delta\pi_{M_1} + \delta\pi_{M_4} \\ \lambda\pi_{R_4} = \delta\pi_{M_2} + \delta\pi_{M_1} + \delta\pi_{M_3} \\ \beta\pi_{V_i} = \lambda\pi_{R_i}, \quad i = 1,2,3,4 \\ 3\delta\pi_{M_i} = \beta\pi_{V_i}, \quad i = 1,2,3,4 \end{cases} \quad with \quad \sum_{i=1}^{4}\pi_{R_i} + \sum_{i=1}^{4}\pi_{V_i} + \sum_{i=1}^{4}\pi_{M_i} = 1 \quad (1)$$

Solving the balance equations and the conservation equation, we determine

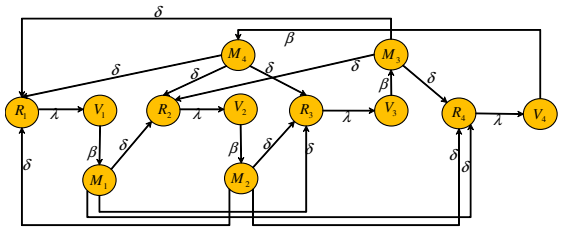

Fig. 2. Transition Diagram Model.

these results:

$$\pi_{R_i} = \frac{\lambda}{\beta}\pi_{V_i}; \quad \pi_{M_i} = \frac{\beta}{3\delta}\pi_{V_i}; \quad \pi_{V_i} = \frac{1}{(1 + \frac{\lambda}{\beta} + \frac{\beta}{3\delta})i} \quad (2)$$

4 Performance Parameters

4.1 Description of performance parameters

Availability : The steady-state availability A is defined as the probability that the system is in one of normal functioning states. Hence, we formulate the Availability as follows:

$$A = \lim_{t \to +\infty} A(t) \quad then \quad A = 1 - \left[\sum_{i=1}^{4}\pi_{M_i}\right] \quad (3)$$

Downtime : Is defined as the interval of time during which all services are unavailable [1]. This period starts from when the virtual machine stops running on the source machine to when it resumes on the destination. The synchronization mechanism has an impact on downtime. Hence it is normally performed in downtime. So, during $\Delta(t)$ interval the downtime function is formulated as follows:

$$Downtime(t) = \left[\sum_{i=1}^{4}\pi_{M_i}\right] * \Delta(t) \quad (4)$$

Downtime Cost : The cost of downtime depends on migration costs because when a system stops functioning, it can lead to expensive repair costs. Let C_M be the unit cost of the migration process. Consequently, during $\Delta(t)$ period of time:

$$DowntimeCost(t) = \left[C_M \sum_{i=1}^{4} \pi_{Mi} \right] * \Delta(t) \tag{5}$$

5 Implementation and Evaluation

5.1 Experimental Setup

For evaluating our Moving Target Defense solution we use a SDN based environment simulated using Mininet virtual network simulator. To create a virtualized environment similar to data centric network we are considering tree topology. The python based SDN controller POX is used to perform command and control over the Mininet environment as can be seen in the Figure 3. The POX has the capability to act as both SDN controller and switch. POX will communicate with TOR OpenFlow switches, and implement the MTD solution. All the experiments are performed on 64-bit Ubuntu Machine with an i7 processor and 8 Gb RAM.

5.2 Virtual machine Migration Algorithm

Algorithm 1 VM Migration Countermeasure Selection Algorithm

```
N : The number of physical servers
K : The number of VMs
VS : The list of vulnerability score for VMs
for i: 1 to N do
    SP ← PS_size Get physical server Size
    while SP > 0 do
        max ← k
        for j: k+1 to K do
            if VS[j] > VS[max] then
                max ← j
                VM_j has a higher vulnerability score VS_j
                VM_j should be migrated to another Physical server
                SV ← VM_size
                if SV < SP then
                    VM_j is in Migration
                    SP ← SP − SV
                end if
            end if
        end for
    end while
end for=0
```

In this algorithm, we consider two tables one for virtual machines and another for physical servers. Based on a designed Vulnerability score, we trigger the migration of one of the VMs. However, the score is the exponential average of CVSS Base Score from each vulnerability in the VM or a maximum 10. Let VS_i vulnerability score for VM_i is defined as [15]. We use the greedy algorithm below based on threshold CVSS score to perform migration pro-actively.

$$VS_i = Min\{10, \ln \sum \exp^{BaseScore(v)}\} \tag{6}$$

Let $PS = (PS_1, PS_2,...,PS_N)$; and $VM = (VM_1, VM_2,..., VM_K)$. The virtual machines that are running on the same tenant node, construct a VMCluster VC.

$$VC = (VC_1, VC_2, ..., VC_M) \;\; with \;\; N > M \;\; VM_j \in VC_i \;\; and \;\; VC_i \in VC \tag{7}$$

5.3 Experiment 1

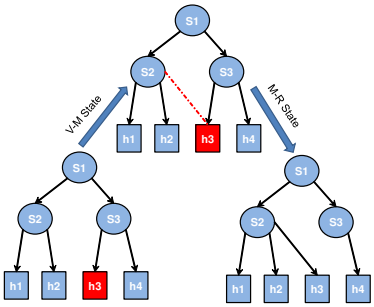

Fig. 3. MTD VM Migration - Tree Topology Network.

We consider a tree topology of depth 2 and fanout 2. Each of the Mininet VMs has a base score ranging from 0-10 which is assigned according to CVSS scoring metric. The cumulative base score of the VMs under a particular Open vSwitch is considered as total vulnerability score on a Physical Server.

The algorithm above checks for a period t = [0-10] seconds for change in base score of the system. We have randomly assigned base score increment value to be $\beta = 0.20$ per second which represents the rate of change of VM from Running State 'R' to Vulnerable state 'V'. As soon as the vulnerability score of a VM reaches over threshold 8, the machine changes to vulnerable state 'V', and the VM migration takes place according to the CTMC process described in section 3.2.

As can be seen from the table hosts h1 and h2 are connected to switch s2, and hosts h3 and h4 are connected to s3. Additionally, switch s1 is connected to both s2 and s3. Since h3 is having a vulnerability score above threshold 8, it is migrated according to CTMC algorithm and it reaches Migration State M.

Host	CVSS Score	Links
h1	7.60	h1-s2
h2	4.13	h2-s2
h3	8.18	h3-s3
h4	5.92	h4-s3

Table 1. Network Connections and Vulnerability Information.

After migration the host h3 is connected to switch s2 as can be seen in Figure 3. The VM h3 reaches Running State 'R' with a vulnerability score '0'.

5.4 Experiment 2

In the second experiment, we measure the performance of CTMC algorithm in the network before and after migration of VM for Tree topology of various depth and fanout. We used the time to transition from Running to Migration State (R-V + V-M) and in Migration to Running State (M-R) measured in seconds.

Depth	Fanout	Hosts	R-V (Ping Test secs)	M-R (Ping Test secs)
2	2	4	0.26	1.74
2	3	9	0.38	1.64
3	2	8	0.38	1.93

Table 2. Transition Time for R−M and M−R States.

The table shows that R-M state transition is faster than M-R migration, which is expected since the host needs to establish a new link on a different switch in M-R state. In all the cases shown in the table, it takes very small amount of time to perform VM migration (2.15 s). This includes R-M state transition time and M-R state transition time. This shows that there will not be any major performance impact on the network QoS because of MTD solution.

5.5 Experiment 3

We performed iperf test to check bandwidth between the host selected for migration and rest of the hosts in the network to check the effect of migration on bandwidth. The table below shows TCP bandwidth in the network from R-M state and M-R state between the host that is being migrated and all other hosts in the network. We tested for a network with depth = 3, fanout = 2.

There was no drastic reduction in the bandwidth during migration phase or after the migration. This shows that network services are not affected because of MTD solution. In addition to this we carried out Cbench test to test the controller behavior during VM migration. The performance remained same as in the case of the normal mode of operation.

Src Host	Dest Host	R-M (Gbps)	M-R(Gbps)
h8	h1	3.55	2.96
h8	h2	3.95	2.98
h8	h3	2.12	2.79
h8	h4	4.30	4.05
h8	h5	4.84	3.75
h8	h6	4.29	4.09
h8	h7	4.79	3.54

Table 3. TCP Bandwidth test for R−M and M−R States.

6 Conclusion and Future Work

In a multi-tenant SDN based cloud computing environment, there are various nodes and applications interacting with each other, and with the outside Internet. This makes it extremely difficult to maintain an environment where the VMs are trustable secure. We present a stochastic model to ensure that the VMs in an environment are below a vulnerability threshold and a methodology to recover the VM while minimizing downtime and preserving user experience. Most MTD techniques involve changing the attack surface so as to make the attacker expend more of their resources in doing reconnaissance and keep them on their toes. However, being proactive also means a greater memory, network and computation overhead for the data center tenants. Instead of random reshuffling for MTD, using our model will reduce system overhead since migration is contingent on VMs having crossed an exploitability threshold. Our future work will involve an HA service assessment that seeks to increase the Mean Time To Failure (MTTF) and minimize the Mean Time to Recover (MTTR). Optimal live migration of VMs and designing other HA methodologies is also being studied. Further, we seek to study how to address policy conflict issues that arise due to the migration of VMs from one physical server to another. In order to make this model more robust, we plan to further specify and evaluate methodologies for providing high availability that is suitable for our work. Further, we seek to quantify and enumerate vulnerability metrics and stipulate transition points from one state to another so as to optimize system resources. Finally, we plan on conducting migration experiments based on our model with real data center scenarios. For the purpose of full system implementation, we plan to test our solution in OpenStack based cloud environment with OpenDaylight controller.

Acknowledgments

This research was sponsored by NSF grant #1528099, and also supported by the NATO Science for Peace & Security Multi-Year Project (MD.SFPP 984425). The research work was conducted as part of visiting scholar - Iman EL MIR's visit to Arizona State University.

References

1. Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., Pratt, I., Warfield, A. : Live migration of virtual machines. In : the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2, pp. 273–286. USENIX Association (2005)
2. Hong, J., Kim, D. S. : Assessing the Effectiveness of Moving Target Defenses using Security Models. IEEE Transactions on Dependable and Secure Computing, 13, 163–177 (2016)
3. Jia, Q., Sun, K., Stavrou, A. : Motag: Moving target defense against internet denial of service attacks. In : 22nd International Conference on Computer Communications and Networks (ICCCN), pp. 1–9. IEEE, (2013)
4. Jafarian, J. H., Al-Shaer, E., Duan, Q. : Openflow random host mutation: transparent moving target defense using software defined networking. In : first workshop on Hot topics in software defined networks, pp. 127–132. ACM, New York (2012)
5. Rohrer, J. P., Jabbar, A., Sterbenz, J. PG. : Path diversification for future internet end-to-end resilience and survivability. Telecommunication Systems. 56, 49–67 (2014)
6. Cox, B., Evans, D., Filipi, A., Rowanhill, J., Hu, W., Davidson, J., Knight, J., Nguyen-Tuong, A., Hiser, J. : N-variant systems: a secretless framework for security through diversity. In : 15th USENIX Security Symposium, pp. 105–120, Vancouver (2006)
7. Yuan, E., Malek, S., Schmerl, B., Garlan, D., Gennari, J.: Architecture-based self-protecting software systems. In : 9th international ACM Sigsoft conference on Quality of software architectures, pp. 33–42. ACM, Vancouver (2013)
8. Wang, H., Li, Y., Zhang, Y., Jin, D., : Virtual machine migration planning in software-defined networks. In : Conference on Computer Communications (INFOCOM), pp. 487–495. IEEE, Hong Kong (2015)
9. Thompson, M., Evans, N., Kisekka, V. : Multiple OS rotational environment an implemented Moving Target Defense. In : 7th International Symposium on Resilient Control Systems (ISRCS), pp. 1–6. IEEE, Denver (2014)
10. Kampanakis, P., Perros, H., Beyene, T. : SDN-based solutions for Moving Target Defense network protection. In : 15th International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1–6. IEEE, Sydney (2014)
11. El Mir, I., Kim, D. S., Haqiq, A. : Security Modeling and Analysis of an Intrusion Tolerant Cloud Data Center. In : Third World Conference on Complex Systems (WCCS), pp. 1–6. IEEE, Marrakech (2015)
12. El Mir, I., Kim, D. S., Haqiq, A. : Security Modeling and Analysis of a Self-Cleansing Intrusion Tolerance Technique. In : 11th International Conference on Information Assurance and Security (IAS), pp. 110–116, IEEE, Marrakech (2015)
13. El Mir, I., Kim, D. S., Haqiq, A. : Cloud Computing Security Modeling and Analysis based on a Self-Cleansing Intrusion Tolerance Technique. Journal of Information Assurance and Security(JIAS). 11, 273–282 (2016)
14. Aziz, A., Sanwal, K., Singhal, V., Brayton, R. : Model-checking continuous-time Markov chains. ACM Transactions on Computational Logic (TOCL). 1, 162–170 (2000)
15. Chung, C. J., Khatkar, P., Xing, T., Lee, J., Huang, D. : NICE: Network intrusion detection and countermeasure selection in virtual network systems. IEEE Transactions on Dependable and Secure Computing. 10, 198–211 (2013)