2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering

Evaluation for Combination of Shuffle and Diversity on Moving Target Defense Strategy for Cloud Computing

Hooman Alavizadeh[1], Julian Jang-Jaccard[1] and Dong Seong Kim[2]
[1] Institute of Natural and Mathematical Sciences,
Massey University, Auckland, New Zealand.
[2] Department of Computer Science and Software Engineering,
University of Canterbury, New Zealand.
Email: {h.alavizadeh, j.jang-jaccard}@massey.ac.nz and dongseong.kim@canterbury.ac.nz

***Abstract*—Moving Target Defense (MTD) has been recently proposed and is an emerging proactive approach which provides an asynchronous defensive strategies. Unlike traditional security solutions that focused on removing vulnerabilities, MTD makes a system dynamic and unpredictable by continuously changing attack surface to confuse attackers. MTD can be utilized in cloud computing to address the cloud's security-related problems. There are many literature proposing MTD methods in various contexts, but it still lacks approaches to evaluate the effectiveness of proposed MTD method. In this paper, we proposed a combination of *Shuffle* and *Diversity* MTD techniques and define evaluation criteria to compare MTD techniques. We investigate on the effects of deploying these techniques from two perspectives lying on two groups of security metrics (i) *system risk*: which is the cloud providers' perspective and (ii) *attack cost* and *return on attack*: which are attacker's point of view. Finally, we show that combining *Shuffle* and *Diversity* techniques can satisfy our evaluation criteria while individual technique cannot.**

***Index Terms*—Security analysis; Moving Target Defense; Cloud Computing; Security Metrics**

I. Introduction

Securing cloud computing has become a huge challenge for both cloud providers offering comprehensive service to their customers who can not trust on the security of this new paradigm. To this point, it is crucial to have novel security mechanism to improve the cloud security [1]. Generally, there are two main security mechanisms: Reactive and Proactive approaches. Reactive approaches are mostly synchronous and include recovery plan after intrusion detected on a compromised system. However, in many cases the recovery process is very difficult or even impossible as it also may cause damages to the system. Thus, using proactive security approaches which are actually based on prevention in advance rather then recovery can be more effective solution. As one of proactive approaches, Moving Target Defense (MTD) has been proposed as a part of emerging proactive approach which provides an asynchronous defensive strategies. Unlike traditional security solutions that focused on removing vulnerabilities, MTD makes a system more dynamic and unpredictable by continuously changing attack surface to confuse attackers.

MTD techniques can be categorized into three comprehensive categories [2] including: *Shuffle*, *Redundancy* and *Diversity*. MTD techniques can either be used independently or combined together to obtain more effective results. Many MTD strategies have been proposed [3]–[5], but it is still difficult to evaluate the *effectiveness* of the proposed MTD methods. It is important to assess the effectiveness of MTD techniques through security metrics (e.g. system risk and attack cost). Security analysis plays an inevitable role in evaluating the overall security-related perspectives of a system. Graphical Security Models (GSM) like Attack Trees (ATs) and Attack Graphs (AGs) are graphical models to formally analyse security of a given system using various security metrics [6], [7]. GSM can be utilized in analysing MTD techniques to determine how effective the MTD techniques or a combination of them are and can provide an optimal MTD technique solution before deploying any techniques which may cause extra cost for the cloud providers. In [8], Hierarchical Attack Representation Model (HARM) proposed which is a formal model based on hierarchical two-layered graph. HARM is more scalable and adoptable than other formal GSMs [8]. In this paper we used HARM to evaluate the effectiveness of our combined MTD techniques before deploying them.

Our main contributions are listed as below which, to the best of our knowledge, has not already been proposed:

- Evaluation of individual and combined MTD techniques, Shuffle (S), Diversity (D) and a combination of them (S+R) at virtualization layer of cloud through simulation;
- Defining an evaluation criteria based on three security-related metrics, System Risk (Risk), Attack Cost (AC), and Return on Attack (RoA) to consider both attackers' and cloud providers' perspectives;
- Analysing the correlation between Important Measures (IMs), *Betweenness* and *Closeness* with the result of deploying individual and combined MTD techniques and compare them with Exhaustive Search (ES) to investigate whether IMs can be used to find out the best deployment strategies satisfying our evaluation criteria or not.

2324-9013/18/31.00 ©2018 IEEE
DOI 10.1109/TrustCom/BigDataSE.2018.00087

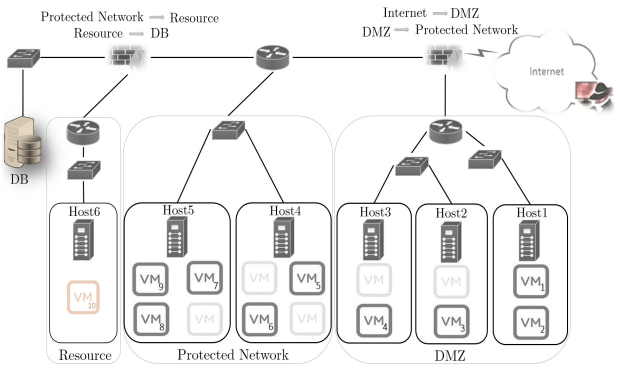

Fig. 1: A Cloud system example.

II. Preliminaries

A. System Setting and Configuration

We setup a Cloud system consisting of three subnets; Demilitarised Zone (DMZ), Protected Network (PN), and Resource Zone (RZ), as shown in Figure 1. Each subnet contains hosts, and each host holds up to four Virtual Machines (VM). Each VM includes two default and backup operating systems (OS). VMs hosted in the DMZ are installed with Windows 10, and VMs in PN and RZ are installed with Enterprise Linux OS. We assume that an attacker is outside the network and can exploit the vulnerabilities of those operating systems to gain access. The goal of the attacker is to compromise the Database (DB) in the RZ. The system's configurations, connectivities and constraints are assumed as follow (we use the following assumptions for the simplicity in description and models and these assumptions could be released):

- A VM either can migrate to another host or be launched with its backup VM
- A VM cannot migrate to other subnets
- Only VM_1 and VM_2 are connected to the Internet
- VM_{10} on $Host6$ cannot be migrated
- All VMs in $Host2$, $Host4$, and $Host5$ are interconnected, and VM_1, VM_4 are always connected to a VM in $Host2$ (if any)
- VM_7 and VM_8 are always connected
- Only VM_{10} (Target) can access to the Database (DB)

Table I shows the vulnerabilities for Windows 10 (W) and Linux (L) OS. Here, we only modelled the vulnerabilities that can bypass firewalls and authentications [9].

B. HARM Construction

We construct a two-layered HARM to model and analyse the cloud system example illustrated in Figure 1. Constructing the HARM, we calculate security-based metrics System Risk (R), Attack Cost (AC), and Return on Attack (RoA). Those metrics are used to assess the overall security of the network by considering both system administrator's and attacker's side. In the upper layer of the HARM, an AG is used, and in the lower layer an AT is used, such as in [2]. The upper layer can compute the reachability between VMs, and the lower layer captures the vulnerability information of each VM.

TABLE I: OS Vulnerabilities

OS_{ID}	CVE ID	CVE BS	Impact	Exploitability	Cost
$W10_{v0}$	CVE-2017-8530	5.8	4.9	0.86	4.2
$W10_{v1}$	CVE-2017-8495	6.0	6.4	0.68	4.0
$W10_{v2}$	CVE-2016-7247	7.5	3.6	0.39	2.5
$W10_{v3}$	CVE-2016-3209	5.0	2.9	1	5.0
$W10_{v4}$	CVE-2016-0019	9.3	10	0.86	0.7
$Linux_{v0}$	CVE-2016-7034	6.8	6.4	0.86	3.2
$Linux_{v1}$	CVE-2016-4278	5.0	2.9	1	5.0
$Linux_{v2}$	CVE-2016-10309	7.5	6.4	1	2.5
$Linux_{v3}$	CVE-2016-10307	10	10	1	0.1
$Linux_{v4}$	CVE-2016-10066	4.3	2.9	0.86	5.7

1) Importance Measures: Security analysis through GSM suffers from scalability problems [2], especially, when we use ES to find the optimal solution. To address this shortfall, we utilized two important Network Centrality Measures (NCM), *Betweenness* and *Closeness*. Using IMs we can find the most important nodes in the network and deploy the MTD techniques on a set of crucial VMs without using ES.

2) System Risk: System risk (Risk) is a security metric showing the overall risk of the network based on the vulnerabilities existing in each VM. In order to measure the system risk, we use both layers of HARM. We constructed the HARM for our networks according to vulnerabilities listed in Table I.

Let $P(vm)$ be the probability of an attack success for an specific VM, and I_{vm} be the impact of the successful attack on that VM, then we can define the risk value of the VM as $R_{vm} = p(vm) \times I_{vm}$. Equation 1 computes the risk value for an attack path (ap). Then, we can calculate the overall risk value of the system as equation 2.

$$R_{ap} = \sum_{vm_i \in ap} R_{vm_i}, \tag{1}$$

$$R_s = \sum_{ap \in AP_s} R_{ap}, \tag{2}$$

where APs is the list of all possible attack paths.

3) Attack Cost: The second security metrics we used in this paper is Attack Cost (AC) which, in here, can be defined as the cost of exploiting the vulnerabilities on a VM by an attacker and can be expanded to compute the overall attack cost of a system. The overall attack cost of a network can also be calculated through the upper layer of HARM. Table I lists the cost of exploiting a VM through vulnerabilities (AC_{vm}). Equation 4 shows the calculation formula for the overall attack cost value of a networked-system.

$$AC_{ap} = \sum_{vm_i \in ap} AC_{vm_i}, \tag{3}$$

$$AC_s = \sum_{ap \in AP_s} AC_{ap}, \tag{4}$$

where ap is a single attack path in the system and AP_s is the list of all possible attack paths in the network.

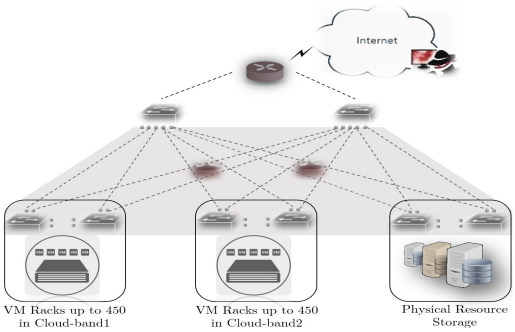

Fig. 2: A Cloud-band model consisting up to 900 VMs.

4) Return on Attack: Return on Attack (RoA) is another security metrics from attackers' perspective [10]. RoA quantifies the attack cost against benefits of attack. Higher value of RoA indicates higher probability that attacker can exploit those vulnerabilities. The ratio of risk value for a VM and attack cost determines the RoA value for a specific VM which are shown in equation 5. Then, the overall RoA of a system can be compute through equation 7.

$$RoA_{vm} = \frac{p(vm) \times I_{vm}}{AC_{vm}} \tag{5}$$

$$RoA_{ap} = \sum_{vm_i \in ap} RoA_{vm_i}, \tag{6}$$

$$RoA_s = \sum_{ap \in AP_s} RoA_{ap}, \tag{7}$$

III. Deploying and Analysing MTD Techniques

To evaluate the effectiveness of MTD techniques, we consider three criteria based on Risk, AC, and RoA metrics. Then, we define three criteria as the framework of evaluation and comparison of each MTD technique:

- C1: an appropriate MTD technique should decrease the overall system risk after deployment compared to none was deployed as in [2].
- C2: deploying MTD technique should make the attack to be more difficult, complicated, and costly for adversaries, for instance the adversary would need a lot more resources to attack.
- C3: MTD technique should satisfy both of the criteria mentioned in C1 and C2 by finding an appropriate threshold between attackers' and cloud providers' perspectives, for instance, the value of RoA.

In this paper, we choose to investigate on the effectiveness of i) VM live Migration (VM-LM) as a *shuffle*-based MTD technique, ii) VM OS diversification as a *Diversity*-based MTD technique, and iii) the combination of both VM-LM and VM diversification as our combined *Shuffle* and *Diversity* MTD technique. Apart from the cloud system example shown in Figure 1, we simulate a large Cloud-band model as demonstrated in Figure 2 to expand our investigation in a larger scale. This model includes two cloud-band nodes each

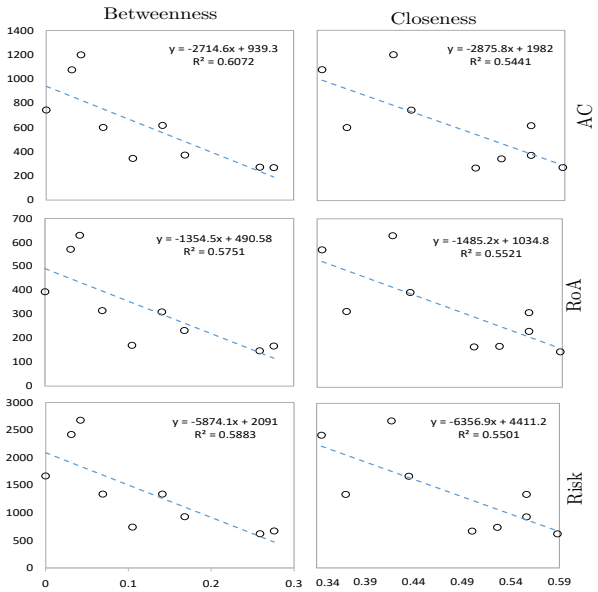

Fig. 3: Correlation analysis of shuffle: metrics vs IMs

of which can handle up to 450 VMs. We assume that only a few of VMs could be connected to the Internet as an entry point of the system (i.e., front-end servers). Moreover, We assume that attachers only can enter the system through entry points VMs connected to the internet. Attackers can exploit vulnerabilities on each VMs and, finally, get access to resource node. We also assume that all VMs in the cloud-band nodes use the same OS; and there is one backup OS for each VM in the network. Finally, we can apply VM-LM between cloud-band nodes as if there is an available space on each node.

A. Shuffle

We utilize VM-LM method to deploy *shuffle* in our simulation. Migration of each VM from a host to another one may affect the overall security of the system. Thus, before deploying *shuffle* technique, we investigate on shuffle deployment effects on three security metrics, Risk, AC, and RoA. We then use HARM to assess the effectiveness of deploying *shuffle* and calculate the security metrics. Shuffle technique only changes VM locations (i.e. from a physical server in cloud to another server); thus, it only affects the upper layer of HARM. Undoubtedly, if we consider all possible migration scenarios and analyze the effectiveness of each movement separately through an ES method, we can obtain an optimal solution. However, this evaluation is highly time consuming and is not applicable in the large sized networks. Alternatively, more effective approach is to use IMs to find out the most important nodes in the network. We calculate and analyse the correlation of each IMs, *betweenness* and *closeness*, with the result of deploying *shuffle* on each node. In this section we consider (i) a regression analysis between the results of deploying shuffle versus *Betweenness* and *Closeness* values, (ii) an investigation on the values of Risk, AC, and RoA for

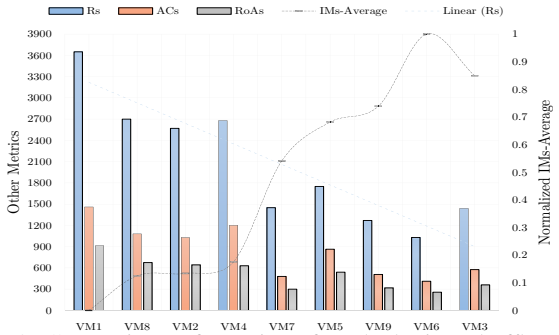

Fig. 4: Comparison of metrics after deploying shuffle technique for cloud system example (VMs in x-axis are sorted based on the IMs-average value).

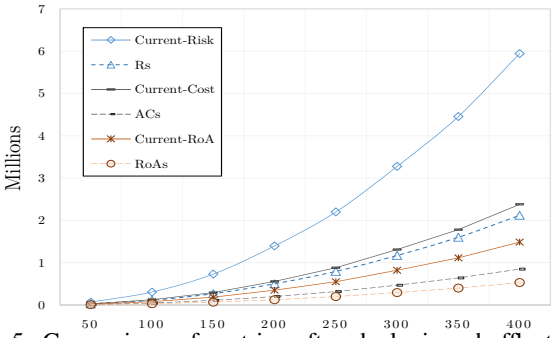

Fig. 5: Comparison of metrics after deploying shuffle technique for cloud-band example with various node size.

each VM. Figure 3 illustrates the correlation between IMs and three security metrics after deploying shuffle technique. Obviously, shuffle has a negative correlation with Risk, AC, and RoA. Figure 4 shows the values for Risk, AC, and RoA together with IMs-average (the average of *betweenness* and *Closeness*). We compare the results on the large cloud-band example shown in Figure 2 with various number of VMs. We consider the values of Risk, AC, RoA after deploying shuffle on the nodes with the highest IMs-average rate and compare them with the previous stage (before deploying shuffle technique), see Figure 5. The results show that (i) deploying shuffle technique on the VMs with higher IMs-average values reduces those three metrics linearly. (ii) shuffle technique decreases AC, but it also reduces Risk and RoA.

B. Diversity

Diversity can be considered as any technique replacing the variant of each component (which can be a VM, server, operating system, hardware, and *etc*.), while the system provides equivalent functionality with the previous state (before changing variant) [11], [12]. Unlike shuffle technique, deploying diversity has no effect on reachability of the VMs in the network; but it only varies vulnerability values which affects the lower layer of HARM. Deploying diversity technique introduces a new set of vulnerabilities to the attacker and may increase the attacker's time and effort. Hence, the attackers need to spend more cost and time to learn the methods in which they can exploit those new set of vulnerabilities. In order to evaluate *Diversity* we consider two scenarios, attacker's perspective (computing attack cost and return on attack metrics) and system administrator's view (computing system risk). In order to deploy diversity, we use OS diversity method which actually replaces the current OS of a VM with the a backup OS. Similar to shuffle technique explained in the previous section, in order to find the best diversity deployment scenario, we deploy diversity technique to all VMs in the system then we compute the Risk, AC, and RoA metrics through ES method. Then, we use IMs to investigate whether the best diversity deployment can be find through IMs. We calculate the correlation between IMs and the results of deploying diversity to see how those metrics correlate to each other. As it can be

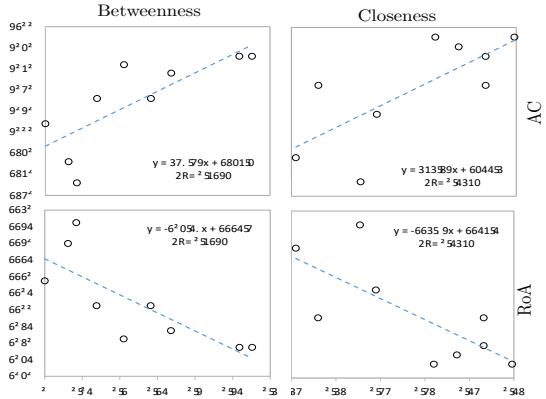

Fig. 6: Correlation analysis of diversity: metrics vs IMs

seen in Figure 6, there are positive correlations (about %77) between AC and IMs. It shows that deploying diversity on the VMs with higher value of IMs increases attack cost, while this scenario has negative correlation with return on attack metric which decreases RoA value. We normalized the values of Risk, AC, and RoA metrics and illustrated them in Figure 7(a). The results show that Diversity technique does not change system risk value. The value of AC goes up when IM increases, while the RoA value decreases. It shows that diversity can improve the security of a system in terms of making attack complicated and costly for attackers. Figure 7(c) shows the results of deploying diversity technique on the cloud-band example. The results shows that deploying diversity does not change system risk, but it decreases both AC and RoA.

C. Combination of Shuffle and Diversity

In Section III-A, we showed that deploying *shuffle* technique decreases system risk; moreover it also reduces AC value as well. Thus, shuffle techniques may increase the probability that an attacker devise a plan to penetrate to the system. However, deploying *diversity* increases AC, but it does not affect system risk. At this point, we combine shuffle and diversity technique to understand the benefit of using these techniques together. To combine both shuffle and diversity techniques, we utilize VM-LM as a shuffle and OS diversity as a diversity technique; then deploy both VM-LM and OS

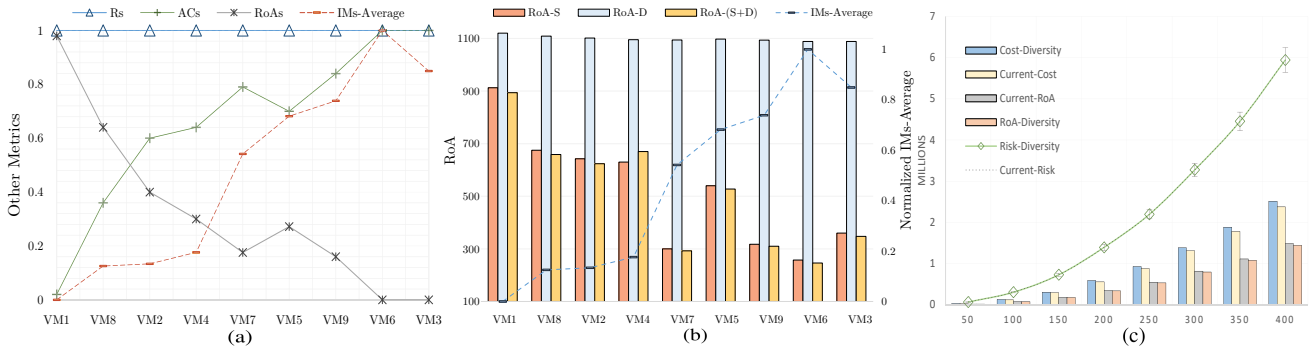

Fig. 7: Comparison of metrics after deploying: (a) diversity for cloud system example (normalised). (b) RoA metric after deploying S + D technique for cloud system example. (c) diversity for cloud-band example with various node size.

diversity at the same time for each VMs in the network. In this section, we show that the combination of shuffle and diversity can increase AC while it reduces RoA value. We used ES to find out the best deployment solution for S, D and $S + D$ for each VM. Figure 7(b) demonstrates the overall value of RoA after deploying combination of shuffle and diversity. As it is shown in the chart, the RoA value is a good indicator to choose the best $S + D$ deployment scenario, because RoA is actually a trade off value between system risk and attack cost; thus, the lowest RoA value is the best MTD deployment for an specific network (which in here VM_6 has the lowest RoA value while it has a high value of IMs). In this section, we showed that RoA metric is the the main key of interest in order to find a trade off between Risk value and AC and also assess how effective a combination of shuffle and diversity technique can be. Thus, by computing RoA value and comparing them with the results of deploying $S + D$ technique we can find out the most effective combined MTD technique which can satisfy C1-C3 in Section II. Moreover, we can compare individual shuffle and diversify techniques with a combination of shuffle and diversity through only analyzing RoA value; as in Figure 7(b), the best value of RoA obtained after deploying $S + D$ on VM6 while this VM has the highest value of IMs-average.

IV. Discussion and Limitations

We investigated on the results of deploying MTD techniques on metrics based on the evaluation criteria discussed in Section III. Table II shows whether each MTD technique can satisfy evaluation criteria in C1-C2 or not, as it can be seen, only the combination of $S + D$ can satisfy the desirable evaluation criteria. Experimental analysis in previous section III showed that the best *shuffle* technique (that minimise the *system risk*) can be found using the IMs. Deploying shuffle technique decreased the system risk, attack cost, and return on attack values. We also find out that shuffle technique has a good correlation with IMs, *betweenness* and *Closeness*. Although deploying shuffle decreases system risk, it also reduces attack cost and return on attack making it easier for attackers

TABLE II: Desirable criteria for each MTD technique

	Shuffle	**Diversity**	**S + D**
C1	✓	×	✓
C2	×	✓	✓
C3	✓	✓	✓

to exploit the vulnerabilities. However, deploying diversity increases both attack cost and return on attack metrics leading to more cost and effort for attackers, but diversity has no affect on system risk and keeps the system risk steady. Finally, to overcome those issues, combination of both shuffle and diversity can be used. Our experimental results in section III showed that by deploying $S+D$ we can find out an acceptable trade off between system risk and attack cost, and measure it through RoA metric. Thus, the lower value of RoA can be consider as an appropriate indicator for security in terms of considering both attacker's and system administrator's perspective. However, we only considered shuffle, diversity and combination of $S+D$ technique, while the other combinations of MTD techniques can also be modelled and investigated (like combinations of shuffle, Redundancy, and diversity). Moreover, we only used three metrics, system risk, attack cost, and return on attack as the main criteria for evaluating MTD techniques. We believe that other metrics should also be taken in to account for example reliability, availability, economical metrics and *etc.* Moreover, we investigate on deploying our MTD techniques to only one important VM in the network with highest rank of IMs, but we also can deploy our MTD techniques on a set of important components.

V. Related Work

Many studies worked on MTD techniques including frameworks, strategies, and applications [13]–[16]. Danev *et al* [17] proposed a shuffle technique for securing cloud infrastructure. They focused on consideration of Migrating VMs in cloud through a secure way. Other shuffle techniques proposed by researchers [18]. Azab *et al*. [19] proposed a diversity method in which it changes running programme's variants erratically.

The proposed method is based on dividing a large programme to smaller portion (cells or tasks) that can be performed with several variants (with the same functionality). However, they didn't consider overall security of their method using a security analysis model.

Most of current studies focused on the MTD techniques and implementation, like the method and strategies in which MTD can be implemented, finding a suitable time-period for applying MTD techniques, and so on [20]. However, there are very few works analysing the effectiveness of MTD techniques for the large networks and cloud environment through security analysis, and a few works combine the MTD techniques to obtain better and more effective results [3]. Peng *et al*, in [21], proposed a MTD techniques for securing cloud-based services with a heterogamous or dynamic attack surface. However, they did not use a formal security analysis model to evaluate the effectiveness of the deployed strategy. Hong *et al.* [2] analyzed the security changes when MTD techniques are deployed, by introducing a formal method to model *Shuffle*, *Redundancy*, and *Diversity* separately. We extended this work by (i) deploying a combination of *Shuffle* and *Diversity*, (ii) analyzing the effects of deploying MTD techniques on each VM and analyzing Risk, AC, and RoA through regression analysis, (iii) comparing the results of regression analysis with those obtained through using IMs (both *Closeness* and *Betweenness*) separately.

VI. Conclusion

MTD techniques have been proposed to enhance the cyber security by changing the network surface, therefore making the attack surface unpredictable for attackers. However, deploying individual MTD technique may affect on some of the security metrics but not satisfies other security metrics. Thus, combination of MTD techniques may overcome this issue. Moreover, it is also crucial to evaluate the effectiveness of MTD techniques using related security metrics before deploying them. In this paper we analyze the shuffle and diversity techniques individually, as well as a combination of both and assess the effectiveness of those techniques through three security metrics: Risk, AC and RoA. Analysing security metrics suffers from scalability problem, to address this issue we utilize a scalable GSM together with IMs to avoid using ES. We simulated a large cloud-band to conduct the results. Finally, our experimental results shows that we can find a trade off between security metrics by combining MTD techniques. For our future work, we will conduct experiments using a real testbed, which we are currently working on implementing a private cloud named Unitecloud [22], to evaluate our proposed methods in a real cloud infrastructure. Further, we will incorporate other combinations of MTD techniques to evaluate their effectiveness, as well as to incorporate more vulnerabilities from other layers in the system.

Acknowledgement

This paper was made possible by Grant NPRP 8-531-1-111 from Qatar National Research Fund (QNRF).

References

[1] Q. Jia, H. Wang, D. Fleck, F. Li, A. Stavrou, and W. Powell, "Catch Me if You Can: A Cloud-Enabled DDoS Defense," in *Proc. of the the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2014)*, Jun 2014.

[2] J. B. Hong and D. S. Kim, "Assessing the effectiveness of moving target defenses using security models," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 2, pp. 163–177, 2016.

[3] H. Alavizadeh, D. S. Kim, J. B. Hong, and J. Jang-Jaccard, "Effective security analysis for combinations of mtd techniques on cloud computing (short paper)," in *International Conference on Information Security Practice and Experience*. Springer, 2017, pp. 539–548.

[4] P. K. Manadhata, "Game theoretic approaches to attack surface shifting," in *Moving Target Defense II*. Springer, 2013, pp. 1–13.

[5] H.-q. Zhang, C. Lei, D.-x. Chang, and Y.-j. Yang, "Network moving target defence technique based on collaborative mutation," *Computers & Security*, 2017.

[6] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. Wing, "Automated Generation and Analysis of Attack Graphs," CMU, Tech. Rep., 2002.

[7] K. Kaynar and F. Sivrikaya, "Distributed attack graph generation," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 5, pp. 519–532, 2016.

[8] J. Hong and D. Kim, "HARMs: Hierarchical Attack Representation Models for Network Security Analysis," in *Proc. of the 10th Australian Information Security Management Conference on SECAU Security Congress (SECAU 2012)*, 2012, pp. 1–8.

[9] P. Mell, K. Scarfone, and S. Romanosky, "Common vulnerability scoring system," *IEEE Security & Privacy*, vol. 4, no. 6, 2006.

[10] M. Cremonini and P. Martini, "Evaluating information security investments from attackers perspective: the return-on-attack (roa)," in *WEIS*, 2005.

[11] J. Rohrer, A. Jabbar, and J. Sterbenz, "Path Diversification for Future Internet End-to-End Resilience and Survivability," *Telecommunication Systems*, pp. 1–19, 2013.

[12] A. Newell, D. Obenshain, T. Tantillo, C. Nita-Rotaru, and Y. Amir, "Increasing network resiliency by optimally assigning diverse variants to routing nodes," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 6, pp. 602–614, 2015.

[13] F. T. Sheldon and C. Vishik, "Moving toward trustworthy systems: R&d essentials," *Computer*, vol. 43, no. 9, pp. 31–40, 2010.

[14] D. Evans, A. Nguyen-Tuong, and J. Knight, "Effectiveness of moving target defenses," in *Moving Target Defense*. Springer, 2011, pp. 29–48.

[15] S. Venkatesan, M. Albanese, K. Amin, S. Jajodia, and M. Wright, "A moving target defense approach to mitigate ddos attacks against proxy-based architectures," in *Communications and Network Security (CNS), 2016 IEEE Conference on*. IEEE, 2016, pp. 198–206.

[16] B. Chatfield and R. J. Haddad, "Moving target defense intrusion detection system for ipv6 based smart grid advanced metering infrastructure," in *SoutheastCon, 2017*. IEEE, 2017, pp. 1–7.

[17] B. Danev, R. Masti, G. Karame, and S. Capkun, "Enabling Secure VM-vTPM Migration in Private Clouds," in *Proc. of the 27th Annual Computer Security Applications Conference (ACSAC 2011)*, 2011, pp. 187–196.

[18] E. Al-Shaer, "Toward network configuration randomization for moving target defense," *Moving Target Defense*, pp. 153–159, 2011.

[19] M. Azab, R. Hassan, and M. Eltoweissy, "Chameleonsoft: a moving target defense system," in *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2011 7th International Conference on*. IEEE, 2011, pp. 241–250.

[20] S. Vikram, C. Yang, and G. Gu, "NOMAD: Towards Non-Intrusive Moving-Target Defense against Web Bots," in *Proc. of the 1st IEEE Conference on Communications and Network Security (CNS 2013)*, 2013, pp. 1–9.

[21] W. Peng, F. Li, C.-T. Huang, and X. Zou, "A moving-target defense strategy for cloud-based services with heterogeneous and dynamic attack surfaces," in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 804–809.

[22] M. He, S. Pang, D. Lavrov, D. Lu, Y. Zhang, and A. Sarrafzadeh, "Reverse replication of virtual machines (rrvm) for low latency and high availability services," in *Proceedings of the 9th International Conference on Utility and Cloud Computing*. ACM, 2016, pp. 118–127.