Frequency-Minimal Moving Target Defense using Software-Defined Networking

Saptarshi Debroy, Prasad Calyam, Minh Nguyen, Allen Stage, Vladimir Georgiev
University of Missouri-Columbia; Humboldt State University; Southeast Missouri State University
Email: {*debroysa, calyamp*}*@missouri.edu, mhn2n8@mail.missouri.edu, ads361@humboldt.edu, vvgeorgiev3s@semo.edu*

Abstract—**With the increase of cyber attacks such as DoS, there is a need for intelligent counter-strategies to protect critical cloud-hosted applications. The challenge for the defense is to minimize the waste of cloud resources and limit loss of availability, yet have effective proactive and reactive measures that can thwart attackers. In this paper we address the defense needs by leveraging moving target defense protection within Software-Defined Networking-enabled cloud infrastructure. Our novelty is in the frequency minimization and consequent location selection of target movement across heterogeneous virtual machines based on attack probability, which in turn minimizes cloud management overheads. We evaluate effectiveness of our scheme using a large-scale GENI testbed for a just-in-time news feed application setup. Our results show low attack success rate and higher performance of target application in comparison to the existing static moving target defense schemes that assume homogenous virtual machines.**

I. Introduction

With the growing trend of hosting critical applications in the fields of finance and healthcare on cloud platforms, there is a need to protect these applications from the security threats of cyber attacks. Cyber attacks can lead to Loss of Availability (LOA) from Denial of Service (DoS) [1]. Lack of adequate protection against cyber attacks can impact reputation and cause millions of dollars in damages to cloud tenants.

The attack defense challenges within a cloud platform are more severe than traditional cyber security risks in two ways. Firstly, a cloud environment becomes a *vulnerability amplifier* to traditional cyber security threats due to the fully distributed and highly elastic nature of the infrastructure resources designed to serve a large population of consumers. Secondly, *new threats* exist that specifically target cloud environments in vulnerable areas of application multi-tenancy within a virtual machine (VM), and third-party broker services between the cloud service provider and the consumers. Consequently, extending traditional intrusion detection systems (IDS) [2, 3] to thwart such cloud-specific threats still remains a challenge.

Amongst the counter-strategies, the Moving Target Defense (MTD) mechanisms can be relatively more effective to protect critical cloud-hosted applications. This is due to MTD's inherent potential to be used to take proactive and reactive measures at the same time, and its amenability to leverage emerging Software-Defined Networking (SDN) [4] paradigms for dynamic network management. However, the design of a MTD strategy needs to minimize the wastage of cloud network/compute/storage resources and limit loss of availability, yet have effective proactive and reactive measures that can thwart attackers. The two fundamental questions that a truly proactive MTD strategy needs to answer to achieve the above design goals are: (i) What is the optimal frequency of proactive migration that protects the VM without consuming excessive cloud resources? (ii) What is the preferred VM location for migration using SDN that does not affect application performance?

In this paper, we address the above fundamental MTD protection design issues within SDN-enabled cloud platforms. Our MTD solution is dual-mode operational in the sense that it allows for proactive migration of target application in a VM for impending attacks, and reactive migration in the event of a LOA attack detection. Our solution novelty is in the frequency minimization and consequent location selection of target movement across heterogeneous virtual machines based on LOA attack probability, which in turn minimizes cloud resource wastage without affecting application performance. The core principle guiding our solution is that the ideal frequency of migration to avert LOA attack should be frequent enough to avoid vulnerability i.e., the frequency should minimize the probability of a VM being attacked before migration.

To realize this approach, we compare the attack probability and migration interval selection for different attack budgets; the higher the attack budget and the lower the attack probability, the more frequent migration is necessary. To counter LOA attacks, our scheme identifies the candidate VMs' key suitability factors, such as, candidate destination VMs' compute/storage capacity, network bandwidth between the candidate destination VMs and the VM hosting the target application, and reputation of the candidate destination VMs based on their attack history to identify the ideal VM to migrate. The reputation is based on rewards and penalties given to VMs on a longer time scale, based on their history of thwarting and falling prey to cyber attack threats. Once the target application is migrated to a new VM using our solution, all the application users are redirected to the chosen destination VM using an SDN controller directing OpenFlow [5] switches within the cloud infrastructure.

We evaluate our scheme using a large-scale GENI infrastructure [6] testbed that features cyber attack templates affecting a realistic just-in-time news feed application setup. The Just-in-time news feed application provides a unique target use case with both news feeds server and database being hosted on the cloud to serve large number of clients, thereby creating multiple vulnerability situations and migration complications. From the collected results, we show how our proactive scheme successfully performs VM migrations that protect the target news feed application from LOA attacks with very low attack success rate. We compare our optimal migration strategy with more static schemes which assumes homogenous VM pool to prove its optimality. We also show how our reactive scheme manages to reverse the performance degradations to the application consumer by timely migration to a chosen destination VM during a LOA attack.

This material is based upon work supported by the National Science Foundation under Award Number: CNS-1359125 and Thomson Reuters. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or Thomson Reuters.

II. Related Work

MTD based works are gaining momentum in tackling cloud based threats and among them [7–11] are notable. In [7], authors propose a shuffling technique of static IP addresses of attacked VMs. Authors in [8] propose a scheme to move around proxy servers to an application server in order to thwart attacks. Another notable work that applies MTD against cyber attacks on VMs is [9] where authors proactively copy the same service into multiple VMs with the consumers redirected to new VM whenever the current VMs running the critical application are attacked. In [10], the authors propose a MTD strategy to marginalize the attackers within a small pool of decoy VMs. Although this method successfully manages to misdirect the attackers, the scheme does not always guarantee regular consumer redirection. Although such works are valid efforts in proposing MTD based security strategies, most of them are reactive in nature assuming a homogenous VM pool.

There are a few notable work in SDN-enabled MTD for clouds, such as [11–13]. In [12], the authors propose a VM IP address mutation scheme instead of a proactive VM migration that uses OpenFlow to route cloud users to the target application using the updated IP address. Authors in [13] studied the benefits and overheads of SDN-enabled MTD schemes for VM migration and their scope does not involve proposing a novel VM migration strategy. The closest related work that uses a proactive security strategy using MTD similar to our scheme is [11], where the authors perform a live VM migration strategy by predicting impending attacks using attack traffic signature pattern recognition. However, they assume a homogenous VM pool and do not offer any reactive failsafes migration strategy in case an attack is detected. Thus, there is a need to propose a holistic proactive and reactive MTD strategy that captures the heterogeneity of different VMs and presents an elaborate trade-off analysis of 'where to migrate' and 'when to migrate', in order to minimize cloud resource wastage without affecting application performance.

III. System and Attack Model

In this section, we present the system and attack model used to describe our proposed MTD scheme.

A. System model

Our system model consists of a cloud application being hosted on a VM and connected to its consumers/clients through an OpenFlow controller as shown in Figure 1. The OpenFlow controller is connected to an authentication server which serves to authenticate and allow legitimate clients subscribed to that particular application. The OpenFlow controller is also connected with other candidate destination VMs which periodically share their status information such as, residual compute/storage capacity with the controller using the *control path*. We assume that the VM pool is heterogenous, i.e., the VMs have different levels of compute/storage capacity, connected to network with varying available bandwidth, and each possessing unique history of cyber attack/threat statistics.

As shown in Figure 1, the regular clients access the cloud hosted application through the OpenFlow controller along the *regular path* and the attacker attacks the target application, more specifically the VM hosting the application along the *attack path*. The IP address of the VM hosting the target application is hidden from the clients. The OpenFlow controller is responsible for managing and performing proactive and reactive VM migration where the current state of the application along with the associated database is migrated to the new VM along the *data migration path*, and the corresponding rerouting of consumers is performed using OpenFlow. For the reactive scheme, the controller is also responsible for intrusion detection, intruder identification, and the subsequent rerouting of only the regular clients of the application.

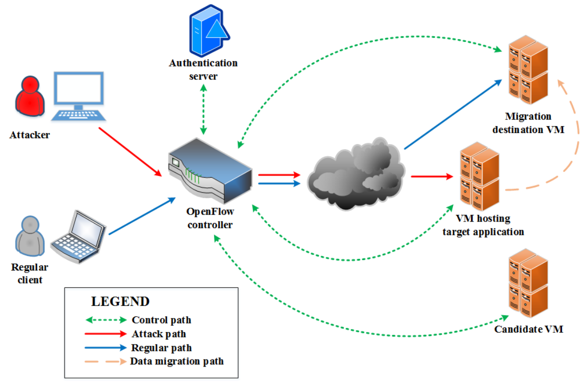


Fig. 1: Proposed MTD based VM migration technique against DoS attacks

B. Attack model

In order to model the DoS attacks on cloud services, we consider the commonly used exponentially distributed [14–16] attack duration on VMs depending on the 'attack budget' of the attackers. Using such a distribution, we model DoS attacks on a particular VM very similar to the way wireless base stations' transmission is modeled on a particular channel based on its power budget [17].

According to this model, every VM will experience two states in terms of being attacked by one or multiple attackers: *Attacked*, when the VM is under attack, and *Idle*, when there is no attack launched on the VM. The *Attacked* and *Idle* period durations are independent of each other and are exponentially distributed with parameters λ_a and μ_i. Thus, for any VM, the duration of *Attacked* period x is an exponentially distributed random variable with mean $T_a = \frac{1}{\lambda_a}$ and is given by

$$f_1(x) = \begin{cases} \lambda_a e^{-\lambda_a x} & \forall\ x \geq 0 \\ 0 & \forall\ x < 0 \end{cases} \tag{1}$$

Similarly, the duration of *Idle* period denoted by the random variable y with mean $T_i = \frac{1}{\mu_i}$ has the distribution,

$$f_2(y) = \begin{cases} \mu_i e^{-\mu_i y} & \forall\ y \geq 0 \\ 0 & \forall\ y < 0 \end{cases} \tag{2}$$

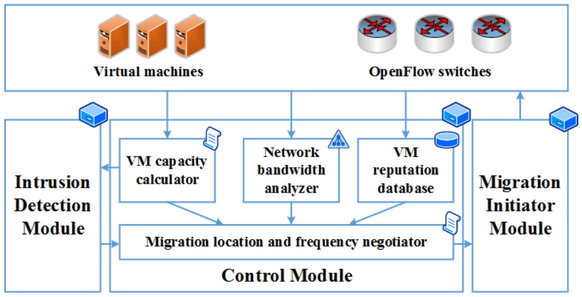

Virtual machines
OpenFlow switches
Intrusion Detection Module
VM capacity calculator
Network bandwidth analyzer
VM reputation database
Migration location and frequency negotiator
Control Module
Migration Initiator Module

Fig. 2: Software architecture of the SDN controller

C. MTD based VM migration strategy

Now we describe the proposed proactive and reactive MTD based VM migration scheme illustrated in Figure 2. The MTD strategy predominantly adopts a proactive scheme unless an attack is detected where it dynamically migrates the service application from one VM to another. The frequency of the VM migration is managed by the 'Migration Location and Frequency Negotiator' of the OpenFlow 'Control Module' as shown in Figure 2 and is adaptive to the statistical DoS attack pattern and probability.

We assume a heterogenous VM pool with all VMs being different in terms of storage/CPU capacity and cyber attack history. The 'Migration Location and Frequency Negotiator' is also responsible for new VM selection process. We argue that the 3 most important VM characteristics for its suitability to host the target application are: a) the new VM's compute/storage capacity; b) the available network bandwidth between the current and candidate destination VM; and c) the reputation of the candidate destination VM in terms of vulnerability to cyber attacks. Below we discuss these factors and their relative importance.

- **VM capacity:** The available computation/storage capacity is critical for successfully catering to client's requests. The new VM should have enough storage capacity to store all the necessary files and client related databases of all the current users and future users to the service. The 'VM capacity calculator' is responsible for calculating VM capacity and for passing it onto the 'Migration Location and Frequency Negotiator'.
- **Network bandwidth:** The available network bandwidth between the candidate destination VM and the VM hosting the target application plays an important role in performing VM migration. With less available bandwidth, it takes more time for the controller to perform VM migration, i.e., file copy between VMs which in turn results in increased threats of cyber attacks in case of proactive migration and extended service interruption to clients in case of reactive migration. The 'Network bandwidth negotiator' is responsible for analyzing the network bandwidth between the current VM and the candidate destination VMs.
- **VM reputation:** We argue that the previous history of a VM in terms of instances of cyber attacks launched against it is a critical factor in deciding the VM's suitability to be selected for migration. As the term 'previous history' is a subjective concept, we seek to translate this into a quantifiable 'VM reputation' which is an objective indicator of how robust a VM is to deter future cyber attacks. The 'VM reputation database' is responsible for calculating and keeping track of each VM's reputation.

Once the optimal VM migration frequency and ideal VM migration location are computed, the 'Control Module' initiates the migration process through the 'Migration Initiator Module' which performs VM snapshot and file transfer. As soon as the migration is complete, all the clients are re-routed to the chosen candidate VM using the OpenFlow switches initiated by the 'Migration Initiator Module'. The entire proactive migration process is preempted by detection of a DoS attack by the 'Intrusion Detection Module'. However, the ideal VM location computation follows the same process as proactive. Once the migration process is complete for such a reactive scheme, all the regular clients except for the attacker are re-routed to the chosen candidate VM via the OpenFlow switches.

IV. MTD Analysis and optimization

In this section we analyze our proposed proactive and reactive migration strategy and quantify optimal migration frequency and ideal migration location. The commonly used notations in the analysis are shown in Table I.

TABLE I: Notations used

λ_a	Random variable for average DoS attack frequency ($1/T_a$)
μ_i	Random variable for average idle period frequency ($1/T_i$)
T_m	VM migration frequency
C_p	Available capacity of any VM p
B_p^v	Network bandwidth between a part of VMs v and p
R_p	Reputation of a candidate VM p

A. Optimal migration frequency

The ideal frequency of VM migration will be such that it is not too infrequent to make the VM vulnerable to cyber attacks and at the same time not too often that it wastes valuable cloud network resources; thus generating an optimization problem. We approach to solve this optimization problem assuming that the interval between two consecutive migrations of a particular VM be T_m such that it is infinitely large if there is no cyber attack, thus minimizing the network resource wastage. However, due to threats of cyber attacks, T_m needs to be adjusted just enough so that it is less than the cyber attack inter-arrival rate. Thus, the modified optimization problem can be formulated as:

$$maximize(T_m)$$
$$s.\ t. \quad T_m \leq \text{cyber attack inter-arrival time} \tag{3}$$

Now, let us assume that the random variable representing the attack inter-arrival time be **z** which is the sum of two independent random variables for *Attacked* and *Idle* periods **x** and **y** respectively, i.e., **z** = **x** + **y**. Therefore, the distribution of attack inter-arrival time **z** is obtained as:

$$\begin{aligned} f_Z(z) &= f_X(x) * f_Y(y) \\ &= \int_{-\infty}^{+\infty} f_X(z-y) f_Y(y) dy \\ &= \begin{cases} \frac{\lambda_a \mu_i [e^{-\lambda_a z} - e^{-\mu_i z}]}{(\lambda_a - \mu_i)} & \forall\ \lambda_a \neq \mu_i \\ \lambda_a^2 z e^{-\lambda_a z} & \text{otherwise} \end{cases} \end{aligned} \tag{4}$$

In order to quantify the optimal T_m, we approach the problem by first calculating the probability of VM getting attacked before migration. Such probability is expressed as:

Prob{VM getting attacked before migration}

$$\begin{aligned} &= \text{Prob}\{\text{z} \leq T_m\} \qquad \text{(VM attack being memoryless)} \\ &= \int_{-\infty}^{T_m} f_Z(z) dz \\ &= \begin{cases} \int_0^{T_m} \frac{\lambda_a \mu_i [e^{-\lambda_a z} - e^{-\mu_i z}]}{(\lambda_a - \mu_i)} dz & \forall\ \lambda_a \neq \mu_i \\ \int_0^{T_m} \lambda_a^2 z e^{-\lambda_a z} dz & \text{otherwise} \end{cases} \\ &= \begin{cases} \frac{\mu_i (e^{-\lambda_a T_m} - 1) + \lambda_a (1 - e^{-\mu_i T_m})}{\lambda_a - \mu_i} & \forall\ \lambda_a \neq \mu_i \\ 1 - e^{-\lambda_a T_m}(\lambda_a T_m + 1) & \text{otherwise} \end{cases} \end{aligned} \tag{5}$$

Now in order to satisfy the condition in optimization Equation (3), probability of VM getting attacked before migration, i.e., Prob$\{z \leq T_m\}$ needs to be minimized. This reduces the optimization problem from Equation (3) to:

$$minimize\Big(\text{Prob}\{z \leq T_m\}\Big) \tag{6}$$

However, due to the asymptotic nature of exponentially distributed random variable **z**, the nature of Equation (5) is continuously increasing and asymptotic; and thus does not have any maxima or minima. Therefore, for a particular cyber attack scenario (i.e., with statistical λ_a and μ_i known), the optimal T_m can be evaluated by tuning the desired probability of VM getting attacked before migration, i.e., Prob$\{z \leq T_m\}$.

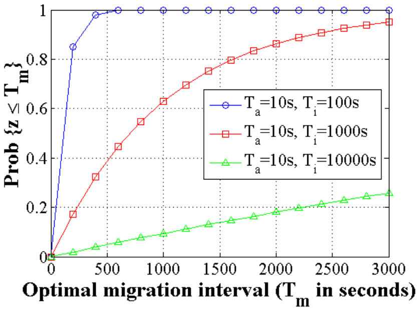


Fig. 3: Migration interval (T_m) optimization for different attack budget

We analyze the nature of Equation (5) against T_m in MATLAB which is shown in Figure 3. It shows how Prob$\{z \leq T_m\}$ increases and reaches maximum value 1 with larger value of migration interval T_m. However, the slope of the probability increase is dependent on the attack budget represented with mean attack and idle periods T_a and T_i respectively; thus making the T_m optimization adaptive to: a) attack budget, and b) the tolerable probability of cyber attack before migration. For example, if the attack budget is high, i.e., $T_a/T_i = 1/10$ and the system can tolerate 1 out of 10 chances of attack success, the optimal T_m should be around 50 seconds; whereas for low attack budget i.e., $T_a/T_i = 1/1000$, for the same tolerance limit of attack success, migrating VMs every 1000 seconds will suffice.

B. Ideal migration location

As mentioned in Section III-C, the constraints for finding an ideal VM migration location are: a) to minimize the client response time of service requests to the new VM, b) to minimize network cost of migration in terms of bandwidth consumed; and c) minimize the risk of future cyber attacks. Let us assume that suitability factor $\mathcal{S}_p^v$ for migrating a service from VM v to VM p represents the cumulative cost of all the above factors. Thus, our objective is to *maximize*$(\mathcal{S}_p^v)$ for the VM pair v and p.

1) VM capacity: The available computation/storage capacity C_p of a candidate VM p is critical for successfully catering to clients' requests. Available computational capacity in terms of CPU cycles is also important as the candidate VM p should be able to successfully process the incoming requests from all the clients within an acceptable response time.

2) Network bandwidth: The network bandwidth between the pair v and p (B_p^v) and the ensuing achievable throughput should be as high as possible in order to make the migration time as small as possible.

3) VM reputation: We argue that the previous history of a VM in terms of instances of cyber attacks is a critical factor in deciding the VM's suitability to be selected for migration. As the term 'previous history' is a subjective concept, we seek to translate this into a quantifiable 'VM reputation' which is an objective indicator of how robust a VM is to future cyber attacks.

We argue that the three fundamental 'Attack statistics' that define the 'VM reputation' in the context of its suitability of migration are:

- *Instances of successful attacks* (α): It is the number of times a VM gets attacked while the target service is being hosted causing DoS attacks to its consumers. It is the most important among the Attack Statistics as a successfully attacked VM is always prone to further cyber attacks as from attackers point of view the VM is an easier target and also the VM is a likely host of cloud services.
- *Instances of unsuccessful attacks* (β): It is the number of times a VM gets attacked after the target service being already migrated. The argument for unsuccessful attack being one of the fundamental Attack Statistics is the fact that once a VM is attacked without any major effect to the consumers makes the VM less vulnerable to future attacks as the attackers do not consider the VM to be a likely host of cloud services.
- *Instances of attack-free status* (γ): It is the number of times a VM does not get attacked with or without the target application. This Attack Statistic is the indicator of the robustness of the VM against cyber attacks which is also an indicator of its resilience against future attacks.

Using the above Attack Statistics, we propose a cumulative fair reputation model which is conservative in nature, i.e., the model penalizes a VM heavily for past successful attacks on it, but rewards incrementally with every unsuccessful attack and instances of attack-free status. The reputation model follows a modified version of the well known beta distribution [18] which is widely used for multivariate trust models and is calculated after every instance of VM migration. Under this model, the reputation R_p^j of any candidate VM p after jth instance is given by,

$$R_p^j = 1 - \frac{\alpha_p^j + \frac{\beta_p^j}{\beta_p^j + \gamma_p^j}}{\alpha_p^j + \beta_p^j + \gamma_p^j} \quad \forall\, p \in V \tag{7}$$

where V is the set of all candidate destination VMs. The rationale behind the modified beta distribution [18] assumption is that with each instance of successful attack (α), the VM becomes very susceptible to future attacks, however, an unsuccessful attack (β) or attack-free status (γ) does not necessarily mean that the VM will never be attacked; but of course the chances of future attack diminishes because from attacker's point of view that particular VM becomes less attractive. More explanation on how such Attack Statistics affect the overall VM reputation in the longer term is provided through testbed implementation results in Section V.

Finally, the overall suitability of any VM p to be chosen for migration of target application from v is expressed as:

$$maximize(\mathcal{S}_p^v)$$

$$\text{where} \qquad \mathcal{S}_p^v = w_c \times C_p + w_b \times B_p^v + w_r \times R_p^j \tag{8}$$

We represent $\mathcal{S}_p^v$ as a weighted function of the three fundamental factors discussed above. The relative and absolute values of the weights may depend on the specific MTD design. In Section V, we will discuss the specific values we choose for our evaluation and the corresponding rationale.

V. Performance evaluation

In this section, we describe the performance evaluation of our proposed scheme on a GENI [6] testbed. We compare the performance of our proposed proactive and reactive FM-MTD scheme against a more static MTD scheme that considers a homogeneous VM environment, such as SH-MTD [11], that is comparable to our work.

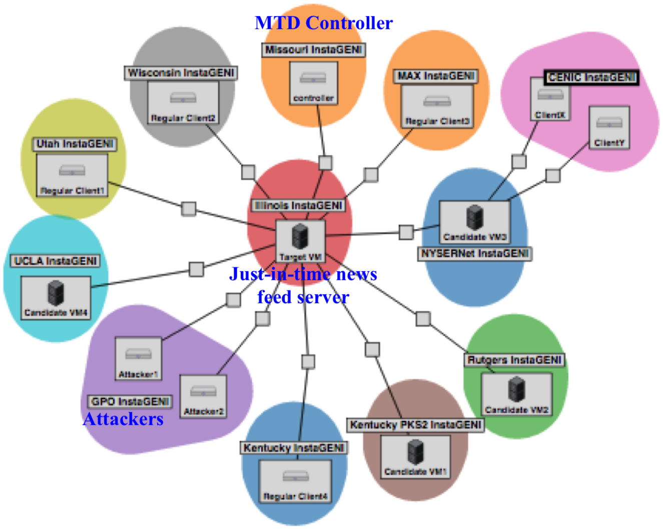


Fig. 4: GENI tesbed topology with VMs, clients, and attackers

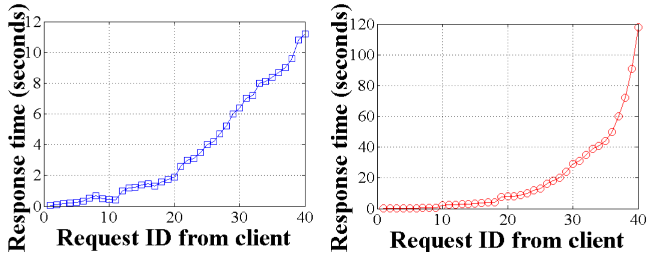
A. Experiment setup

The experiment setup on the GENI cloud testbed consists of the following components:

- One target VM under DoS attack at Illinois InstaGENI hosting Just-in-time news feed application and client database. The application supplies the latest RSS feeds when it receives HTTP GET requests from the clients.
- Four non-malicious clients of the target application at four different locations. The clients are created to simulate the client browsers, where they can send GET HTTP requests to the target VM hosting the target application, and receive the RSS feeds as response.
- Two attackers simulating regular client behavior where they keep sending a large number of GET HTTP requests to the target VM thereby blocking all the ports and creating a DoS attack.
- Upto 30 candidate VMs at different locations simulating varied scenarios with varied suitability factors (utilization, bandwidth, and reputation) discussed in Section IV-B. Figure 4 only shows 5 of them.
- The SDN controller at Missouri InstaGENI rack with software components of control module, IDS module, and migration initiator module as shown in Figure 2.

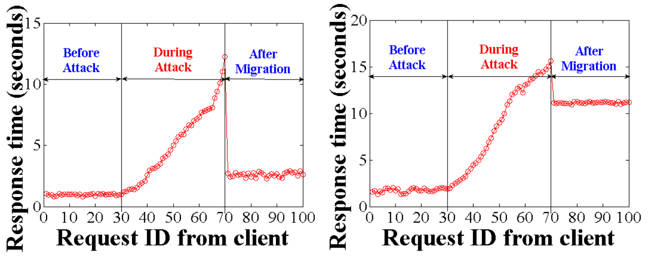
B. DoS attack impact

We make Attacker1 and Attacker2 send out continuous GET requests to slow down all VM resources to stall the news feeds service. While this attack is going on, Regular Client4 sends a regular request for accessing the resources. The response time for Regular Client4 with 1 attacker is shown in Figure 5(a), and the response time with 2 attackers is shown in Figure 5(b). We can clearly see that the response time for legitimate users increases exponentially with the number of attackers.



(a) DoS attack with one attacker (b) DoS attack with two attackers

Fig. 5: Response time degradation for different attack intensity



(a) Migration using proposed FM-MTD (b) Migration using SH-MTD

Fig. 6: Effect of VM utilization on migration performance

C. Impact of VM capacity

In Figures 6(a) and 6(b), we compare the results of reactive VM migration between FM-MTD and SH-MTD on the basis of candidate VM's capacity consideration. Figure 6(a) shows the performance of our proposed ideal VM location selection scheme by the SDN controller where the chosen destination VM location is Rutgers InstaGENI rack and the response time for Regular Client4 is almost equal to the old VM location at Illinois InstaGENI rack. Whereas, if the controller randomly selects a new VM location without considering VM utilization as in case of SH-MTD, the response time is almost as bad as being attacked. In this case the controller chooses NYSERnet InstaGENI rack which is already busy catering dummy ClientX and ClientY with some other dummy service and poorly serves Regular Client4 with Just-in-time news feeds due to lack of available capacity. The results show performance increase by a factor of four with FM-MTD over SH-MTD, and demonstrate that the residual capacity of candidate destination VMs is critical for application performance. For this experiment, we assumed $w_c > w_b > w_r$ as we argue that consumer response time after migration is the most important benchmark to measure success of cloud-based LOA attack defense.

D. Impact of network bandwidth

Figure 7 compares the effect of network bandwidth on migration performance and justifies the importance of network bandwidth as a factor in ideal VM migration location selection. In the experiment, we deliberately programmed the candidate VM at Kentucky PKS2 InstaGENI rack with similar features to Rutgers InstaGENI rack (the ideal choice for migration) except for the network bandwidth with Kentucky having half the achievable throughput (∼6 Mbps) than Rutgers (∼12 Mbps).

By varying the size of the entire application, particularly the database, we observe that the transfer takes longer for Kentucky as expected. This increased transfer time in turn affects the service interruption time of the clients during the attack and can be pronounced at a cloud-scale application delivery.

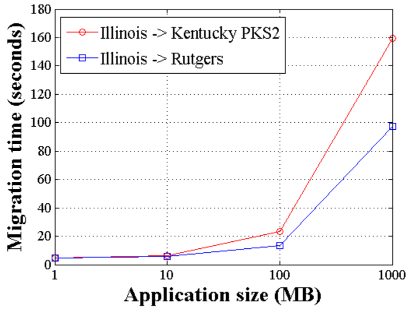


Fig. 7: Effect of network bandwidth on migration performance

E. *Performance of the cumulative reputation model*

In Figure 8, we show the performance of the cumulative reputation model through VM reputation evolution with different attack statistics. We program Attacker1 and Attacker2 to attack Illinois InstaGENI rack while it hosts the target application, and attack UCLA InstaGENI rack while it does not host the target application and never target Rutgers InstaGENI rack. Figure 8 shows sharp decline of Illinois InstaGENI rack reputation with instances of successful attacks (α) while UCLA and Rutgers slowly building reputation with β and γ respectively. It is interesting to note that Rutgers' reputation growth is steeper than UCLA which results Rutgers to be chosen over UCLA.

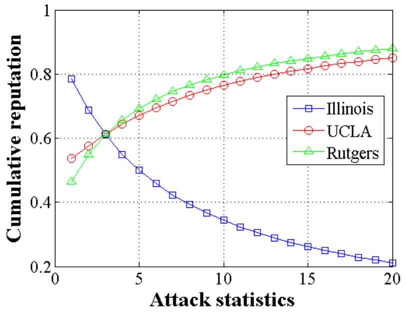
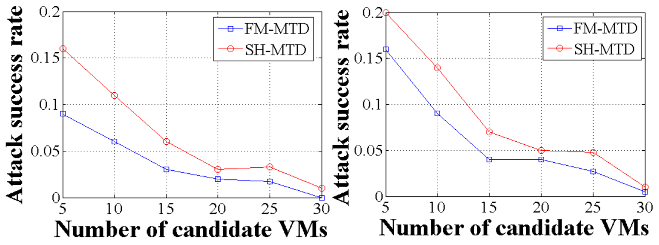


Fig. 8: Performance of the cumulative fair reputation model for different attack statistics

F. *Proactive migration performance*

Finally, we perform proactive migration of the Just-in-time news feeds service among different candidate VMs using the optimal T_m (from Figure 3) and ideal location schemes as proposed in FM-MTD. We vary the probability of attack dependent on the attack budget by varying the ratio $\frac{T_a}{T_i}$. We compare this performance against SH-MTD where migration interval is static (T_m is $\frac{T_i}{2}$). As shown in Figures 9(a) and 9(b), we can observe that for both values of attack budget, proactive migration strategy with optimal migration frequency performs better (roughly by 50% at lower ends) than static migration frequency in terms of attack success rate justifying the benefits of our proposed FM-MTD scheme.



(a) Migration comparison for $\frac{T_a}{T_i} = \frac{1}{10}$ (b) Migration comparison for $\frac{T_a}{T_i} = \frac{1}{100}$

Fig. 9: Migration performance comparison

VI. Conclusions and Future Work

In this paper, we proposed an intelligent MTD based proactive and reactive VM migration scheme to protect cloud based applications from LOA attacks, such as DoS. Our proposed scheme optimizes the frequency of migration in order to minimize wastage of network resources, yet limiting attack effects. The scheme also computes ideal migration location based on candidate VM's capacity, available network bandwidth, and VM reputation in terms of attack history. To the best of our knowledge, this work is the among the very first to propose a truly optimized proactive MTD based VM migration scheme that considers a heterogeneous VM pool and performs trade-off between cost of such migration and benefits in terms of enhanced protection. In future, we plan to minimize the cloud management cost of MTD based solutions to perform trade-offs between system obfuscation and resource management.

References

[1] P. Hershey and C. Silio Jr., "Procedure for detection of and response to distributed denial of service cyber attacks on complex enterprise systems," in *Systems Conference (SysCon), 2012 IEEE International*, March 2012, pp. 1–6.

[2] P. Stavroulakis and M. Stamp, *Handbook of information and communication security*. Springer-Verlag, 2010.

[3] A. Kartit, A. Saidi, F. Bezzazi, M. El Marakki, and A. Radi, "A new approach to intrusion detection system," vol. 36, no. 2, pp. 284 – 289, February 2012.

[4] I. Monga, E. Pouyoul, and C. Guok, "Software-defined networking for Big-Data science - architectural models from campus to the WAN," in *High Performance Computing, Networking, Storage and Analysis (SCC)*, Nov 2012, pp. 1629–1635.

[5] "Openflow switch specification," `https://www.opennetworking.org/sdn-resources/onf-specifications/openflow/`.

[6] "NSF GENI Infrastructure," `https://www.geni.net/`.

[7] T. Carroll, M. Crouse, E. Fulp, and K. Berenhaut, "Analysis of network address shuffling as a moving target defense," in *Communications (ICC), 2014 IEEE International Conference on*, June 2014, pp. 701–706.

[8] H. Wang, Q. Jia, D. Fleck, W. Powell, F. Li, and A. Stavrou, "A moving target DDoS defense mechanism," *Computer Communications*, vol. 46, pp. 10 – 21, 2014.

[9] 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*, 2014.

[10] Q. Jia, H. Wang, D. Fleck, F. Li, A. Stavrou, and W. Powell, "Catch me if you can: A cloud-enabled DDoS defense," in *Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on*, June 2014, pp. 264–275.

[11] R. Zhuang, S. Zhang, A. Bardas, S. DeLoach, X. Ou, and A. Singhal, "Investigating the application of moving target defenses to network security," in *Resilient Control Systems (ISRCS), 2013 6th International Symposium on*, Aug 2013, pp. 162–169.

[12] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "Openflow random host mutation: Transparent moving target defense using software defined networking," in *Proceedings of the Workshop on Hot Topics in Software Defined Networks*, 2012, pp. 127–132.

[13] P. Kampanakis, H. Perros, and T. Beyene, "SDN-based solutions for moving target defense network protection," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on a*, June 2014, pp. 1–6.

[14] L. Swiler, C. Phillips, D. Ellis, and S. Chakerian, "Computer-attack graph generation tool," in *DARPA Information Survivability Conference amp; Exposition II, 2001. DISCEX '01. Proceedings*, vol. 2, 2001, pp. 307–321 vol.2.

[15] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. Wing, "Automated generation and analysis of attack graphs," in *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*, 2002, pp. 273–284.

[16] M. E. Kuhl, J. Kistner, K. Costantini, and M. Sudit, "Cyber attack modeling and simulation for network security analysis," in *Proceedings of the 39th Conference on Winter Simulation: 40 Years! The Best is Yet to Come*, ser. WSC '07, 2007.

[17] S. Debroy, S. De, and M. Chatterjee, "Contention based multichannel MAC protocol for distributed cognitive radio networks," *Mobile Computing, IEEE Transactions on*, vol. 13, no. 12, pp. 2749–2762, Dec 2014.

[18] B. E. Commerce, A. Jsang, and R. Ismail, "The beta reputation system," in *Proceedings of the 15th Bled Electronic Commerce Conference*, 2002.