# Analyzing Moving Target Defense for Resilient Campus Private Cloud

Minh Nguyen, Priyanka Samanta, Saptarshi Debroy
City University of New York
Emails: {*mnguyen, psamanta*}*@gradcenter.cuny.edu, saptarshi.debroy@hunter.cuny.edu*

***Abstract*—With the surge in data-intensive science applications, the campus cloud infrastructures are increasingly dealing with sensitive data that has strict security requirements. However, in most cases due to lack of sophisticated security frameworks and trained personnel, such campus private clouds (CPC) are not fully equipped to handle sophisticated integrity, availability, and confidentiality attacks. In this paper, we demonstrate the utility of a cost-effective, and implementationally simpler Moving Target Defense (MTD) based cloud resource adaptation approach that significantly reduces the probability of attack success. In particular, we propose a Bayesian Attack Graph (BAG) based threat assessment model. Our proposed model follows Common Vulnerability Scoring System (CVSS) impact evaluation recommendations. As a case study, We use our graph based threat assessment model to demonstrate the utility of MTD against attacks on City University of New York (CUNY) research network. The study involves unique scenarios with multiple confidentiality, integrity, and availability related vulnerabilities being exploited by attacks from different network locations. Finally, we simulate a CUNY research network in GENI environment to validate our BAG model by emulating attack scenarios and observing system resilience with and without MTD.**

***Index Terms*—Moving target defense, resilient private cloud, campus cyber infrastructure, Bayesian attack graph.**

## I. Introduction

Data-intensive science applications (e.g., in areas of high energy physics, bioinformatics, health-care) often require specialized instruments and cyber infrastructures (e.g., supercomputers, data repositories, high bandwidth data transfer) that do not always reside at the data generation sites on researcher labs [1]. They are often processed by virtualized supercomputers at university high performance computing facilities acting as Campus Private Clouds (CPC). Many data-intensive science applications requiring CPC infrastructures deal with highly sensitive data such as, Health Insurance Portability and Accountability Act (HIPAA) and the Family Educational Rights and Privacy Act (FERPA) that have strict integrity and confidentiality requirements. However, CPCs in many cases lack state-of-the-art cyber defense tools, softwares, and personnel: a) as they are less visible to the rest of the Internet and are relatively less attractive or lucrative targets of sophisticated cyber attacks, and b) due to overall operating budget constraints - thus making them ill-equipped to handle sophisticated attacks if and when they occur. Therefore, it is highly important that cost effective, simple to implement yet potent defense mechanisms are employed at CPCs that can not only react and recover but also take proactive measures to reduce the chances of such attack success.

Recently, the cloud security community is exploring 'Cyber Agility and Defensive Maneuver' mechanisms. These allow for real-time service restoration through agile cloud resource obfuscation/adaptation once an attack is detected, and also limit proliferation of detected attacks within the cloud environment through preventive maneuvers. Amongst these strategies, Moving Target Defense (MTD) [2] based virtual machine migration mechanisms are cost effective to implement in a CPC to protect critical cloud-hosted science applications [3]. MTD's amenability to leverage emerging Software-defined Networking (SDN) [4] paradigms makes it easier to implement to perform both (i) proactive migration, to detect an attack and act before it makes an impact, and (ii) reactive migration, to recover immediately upon attack detection. Although many works have proposed MTD based cloud defense strategy design for enterprise clouds, a theoretical study on the simple yet effective Virtual Machine (VM) based MTD mechanism against common attacks to CPCs is still warranted.

In this paper we first propose a Common Vulnerability Scoring System (CVSS) [5] driven Bayesian Attack Graph (BAG) model [6, 7]. This model is used to perform a dynamic threat/risk assessment for integrity, confidentiality, and availability attacks on CPC data residing in VMs. BAGs are a type of directed acyclic graph that can model cyber attack causal relationship and can enable adversary beliefs and attack evidences to be considered in assessing attack success likelihoods. We use CVSS [8] for level of 'Exploitation computation of a vulnerability to calculate the likelihood of an attack success in achieving targeted goals called 'achievements'. The overall representation of the BAG and intermediate likelihoods or probabilities help to determine the overall likelihood of the attack success in fully exploit the vulnerability. As for the vulnerabilities, we use relevant cyber-attack statistical data from Common Vulnerability and Exposures [11] for the proposed model. Using the proposed model, we perform a case study on City University of New York research network (CUNYNet) [9] to evaluate the likelihood of success of confidentiality, integrity, and availability attacks with and without MTD The BAG based theoretical results overwhelmingly outline the effectiveness of MTD in thwarting the attacks.

Finally, we demonstrate the utility of a SDN enabled MTD based proactive and reactive VM migration strategy using a GENI [12] based testbed implementation and evaluation. On GENI, we create a CUNYNet topology and generate five pre-configurations for the VMs to mimic the vulnerabilities to be exploited by cyber attacks. We then perform experiments on the simulated system by launching confidentiality, integrity, and availability attacks that exploit these pre-existing vulnerabilities. The experiment results corroborate with BAG based findings in establishing the utility of MTD based VM migration strategy in successfully minimizing attack impact and future attack success probability.

The rest of the paper is organized as follows. Section II discusses the related work. Section III presents the background and problem motivation. Section IV discusses BAG based threat assessment model. Section V outlines the CUNYNet case study. Section VI discusses the GENI testbed based performance evaluations. Section VII concludes the paper.

## II. Related Work

The related work in the area of MTD based cloud security solutions can be divided in to the following sub-areas.

### A. Large data-intensive science communities

Existing works pertaining to security and dependability of CPC resources for data-intensive science applications mostly deal with security measures and point solutions to counter Loss of Integrity (LoI), and Loss of Confidentiality (LoC) threats for 'data in motion' in large federated Big Data research communities. These point solutions include firewalls and Intrusion Detection/Prevention Systems (e.g. Snort [13] and Honeypot [14]). Whereas, exemplar solutions to Big Data transfer in a federated environment include Globus that provides the ability to use point solutions such as InCommon [15], OpenID [16] and X.509 [17] to access resources. The Large Synoptic Survey Telescope community on the other hand provides detailed guidelines for multi-domain cybersecurity compliance with a list of threat mitigating capabilities at involved domains [18].

### B. MTD-based cloud security solutions

In recent times, cloud based MTD works are gaining momentum in tackling cloud based threats, among these, [3, 19–23] are notable. In [19], authors propose IP addresses and proxy server randomization in order to thwart attacks. Whereas, in [20], the authors propose a MTD strategy to marginalize the attackers within a small pool of decoy VMs. Other notable work that applies MTD against cyber attacks on VMs is [3] where authors use VM duplication with consumers redirected whenever the VMs running the critical applications are attacked. Works such as [21–23] proposed SDN enabled MTD schemes where either VM IP address mutation schemes or VM migration schemes use OpenFlow to route cloud users to the target applications. The common theme in most of such SDN enabled MTD schemes are the ease of implementation in terms of application migration, user redirection, and service restoration with successful misdirection of the attackers.

### C. Bayesian Attack Graph based risk assessment

Among the most recent and notable works on attack graph based risk assessment [24, 25] are notable where non Bayesian approaches are used for graph modeling, information collection, and graph core building. Among Bayesian network inspired attack graphs, [26–28] are notable. In [26], the authors measure the network security risks using metrics produced by Bayesian attack graphs. In [27], the authors propose Bayesian network and truth table for attack graph modeling to justify the uncertainty (i.e. attackers' intention of attack) when analyze cyber-security. In [28], the authors propose game theoretic attack graphs to model attacker and defender interactions. Amount of BAG based works that assess the utility of a defense mechanism against cyber attacks in cloud are limited. *In this paper, we try to model CVSS inspired threats and vulnerabilities in CPC environment for the first time to not only understand their effects on critical data intensive applications but also to assess the utility of MTD based VM migration schemes in thwarting such attacks from a theoretical point of view.*

## III. Background and Motivations

In this section, we introduce the relevant concepts and fundamentals in order to motivate the problem and the proposed solution.

### A. CPC architecture and vulnerabilities

A typical CPC consists of a cyber infrastructure that includes virtual servers, such as, Web servers, Application servers, Database Management Systems (or Database servers), and File servers. In most cases, the CPC is within a Science DMZ infrastructure [29] that is designed to optimize performance for research applications by removing obstacles that traditional networks place on data transfer and other functions (e.g. firewalls). Unlike traditional servers, the data at CPC do not reside solely on some physical servers, rather on many virtual machines (VMs) to store and process the information more effectively as shown in Fig. 1 where the data can be accessed from within or outside or inside the DMZ.

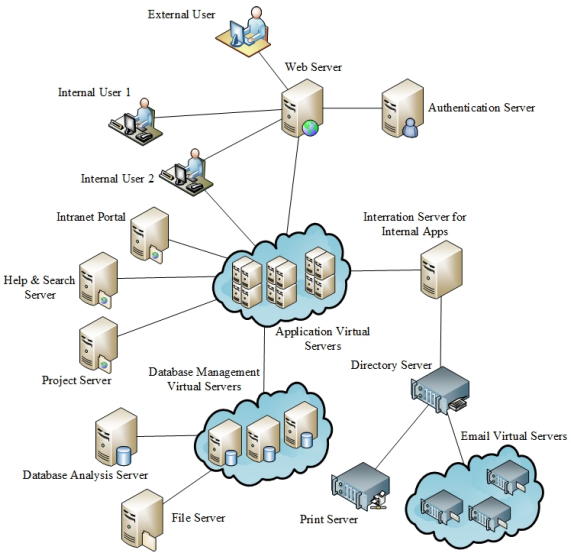



Fig. 1: An exemplary scenario of a campus private cloud with virtualized servers

Unlike enterprise clouds, in most CPC systems virtual servers share a common physical data center. Thus, compromising one can lead to the comprising the entire system. Moreover, coexistence of different types of virtualized servers results in multiple types of information and data to be vulnerable. For example, in Fig. 1, an attacker with access to the Application server can use vulnerability CVE-2017-0160 [11] on Microsoft .net framework and then using the inherent vulnerability of .net, the attacker can take over the entire database carrying sensitive HIIPA data by getting access to internal network. Similarly, by exploiting the vulnerability CVE-2015-4794 to execute an arbitrary code remotely to compromise Oracle DB on the Database server, attackers can achieve complete control over the server. Many of these existing vulnerabilities can become expensive to eradicate and need time, and manpower which CPCs many times can ill afford. At the same, such data-intensive science applications are handled by many people with varying privileges (e.g., lead researcher, postdocs, grad students). Sending the sensitive data outside the purview of the lab servers into the vulnerable CPC causes concerns over data confidentiality and integrity issues

which in turn acts as roadblock towards data dissemination and collaboration.

### B. Campus vs. enterprise cloud security

In comparison to the enterprise cloud service providers, campus clouds are smaller in both scale of available resources and amount of processed data; thus, some of the security concerns and corresponding solutions that are available at enterprise clouds, are not viable for CPCs. Compounded with CPCs budget limitations and lack of trained manpower, it is important that campus clouds find ingenious and low-cost ways of creating intelligent and effective security solutions. The traditional cyber defense mechanisms employed in CPCs have two basic shortcomings in defending against network infiltration based attacks within the cloud environment. Firstly, most traditional campus defense mechanisms remain static in their security composition and do not evolve with changing adversary behavior. This is particularly true for network infiltration based attacks due to adversary's enhanced visibility of the environment. Secondly, lack unpredictability and variance in the outcomes of low-cost enterprise cloud solutions resulting application performance degradations and in most cases easy circumventing by the adversary. Hence, dynamic solutions are warranted with 'at hand' resources that is easily implementable with the existence system and minimal footprint on the system performance.

### C. Moving Target Defense as the solution

Moving Target Defense (MTD) based cloud resource adaptation/obfuscation techniques are one of such solutions. According to Department of Homeland Security, MTD is defined as the "*concept of controlling change across multiple system dimensions in order to increase uncertainty and apparent complexity for attackers, reduce their window of opportunity and increase the costs of their probing and attack efforts*" [2]. As mentioned earlier, MTD based cloud resource adaptation/obfuscation can be manifested through various implementation including, IP address randomization, periodic VM spawning through decoys, and online and offline VM migration for reactive and proactive measures. The key of a MTD based cyber defense strategy that makes it an ideal solution for CPC environments are: 1) Intelligent but fast converging algorithms can be easily developed for both proactive and reactive maneuvers based on triggers for a wide range of global and local greedy optimization criteria; 2) Emerging network management technologies such as, SDN [4] and OpenFlow [30] can help implement and operationalize such dynamic and agile maneuvers with relative ease; and 3) Sophisticated dynamic maneuvers can be designed to create system obfuscation with very little overhead on an existing virtualized system.

Figure 2 shows a MTD based VM migration system that we assume for our work consisting of a data-intensive science application being hosted on the CPC and connected to its clients/users through an SDN/OpenFlow controller [3, 30]. The OpenFlow controller is connected to an authentication server which serves to authenticate and allow legitimate users subscribed to that particular application. The OpenFlow controller is also connected with other VMs that work as candidate destinations in case the application requires migration. These VMs periodically share their network, compute, and storage resource availability status information with the controller. As shown in Figure 2, the regular users access the application

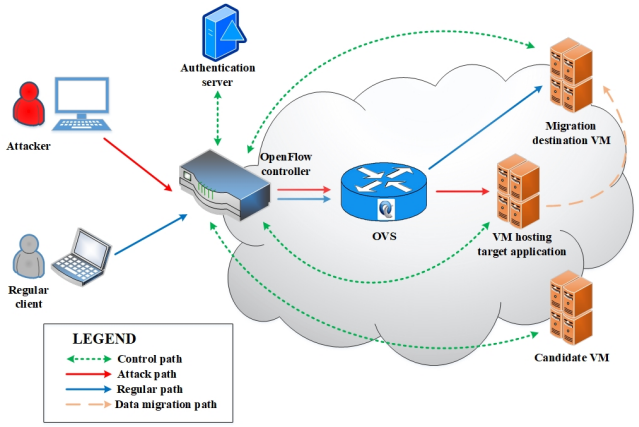



Fig. 2: MTD based proactive and reactive VM migration scheme with SDN controller and OpenFlow switches

through the controller and OVSs along the regular path and the attacker attacks the target application, more specifically the VM hosting the target along the attack path. The IP addresses of the VMs are hidden from the users. The controller is responsible for managing and performing proactive and reactive VM migration where the current state of the application is migrated to the new VM along the data migration path, and the corresponding redirection of the users is performed using OVSs. For the reactive scheme, the controller is also responsible for intrusion detection, identification, and the subsequent rerouting of only the regular users of the application.

## IV. BAG based Risk/Threat Assessment

To demonstrate the existing vulnerabilities of CPC network and the effectiveness of a MTD based VM migration approach showed in Figure 2 in reducing such vulnerabilities, we have proposed a Bayesian attack graph (BAG) model, capturing the relationships amongst vulnerabilities and exploitation routes for attackers. Our BAG model uses CVSS to compute attack likelihood parameters given a network implementation. Below we discuss the model components, followed by likelihood computation.

### A. Attack graph components

Our proposed BAG is an acyclic weighted graph $G(V, A, R, E)$ that captures the causality relationship between standard vulnerabilities and their impacts mentioned in CVSS list. Here $V$ represents vulnerabilities present in a particular CPC network and their impacts are represented as the achievement $A$ with a weighted edge $E$ with weight $P()$ which represents the probability of a vulnerability to be exploited successfully. $R$ is the relationship between vulnerability and the corresponding achievement. The components of the acyclic weighted graph $G(V, A, R, E)$ are defined below:

**Vulnerability ($V$):** Vulnerability $V$ is represented as vulnerability node (□), which is a combination of CVSS ID (e.g. CVE-2014-6567), precondition of that vulnerability (e.g. AV, PR, UI, AC), type of attacks (e.g. code execution), and target (e.g. applications and servers).

**Achievement ($A$):** Achievement $A$ is represented as achievement node (○), which is the post-condition of the vulnerability ($V$). It represents the achievement of the attackers if they can successfully exploit the vulnerability.

**Relationship ($R$):** Relationship $R$ is represented as

relationship node ($\diamond$), which contains 'AND' logic, where it combines two-or-more vulnerabilities/achievements gain newer achievements. 'AND' logic signifies that an achievement is gained if and only if the previous vulnerabilities or achievements are gained. A relationship can be influences by an external factor which impacts the outcome of such relationship.

**Edge ($E$):** An Edge $E$ represents as ($\longrightarrow$), is the direction from the Vulnerability ($V$) to the achievement ($A$) with a designated probability value ($P()$) as the edge weight, with which the Vulnerability is successful.

The structure of acyclic weighted BAG, in terms of the number of vulnerability and achievement nodes, relationship between vulnerabilities and achievement nodes, and the likelihood of successfully exploiting such vulnerabilities and attaining the achievements depends on the system and network architecture. The factors of the system and network architecture that determines the BAG stricture is a non-exhaustive list that includes firewall location and rules, attacker location in terms of network distance, softwares and applications installed and their inherent vulnerabilities etc. Below we first discuss the CVSS inspired likelihood of success vulnerability exploitation computation followed by a hypothetical and real case study to demonstrate the BAG construction process.

### *B. Likelihood of success computation*

The likelihood of success or probability $P$ of vulnerabilities are calculated based on the CVSS 'Exploitability'. In this work, we have only considered the events occurred before attack, in other words preconditions or pre-configurations; hence 'Impact' given CVSS has very little consideration. Exploitability of a vulnerability, in other words, the difficulty to exploit that vulnerability is calculated based on four metrics: Attack Vector, Privileges Required & Scope, User Interaction, and Attack Complexity. Exploitability score is inversely proportional to the difficulty level, hence higher the values of the above metrics' signifying higher exploitability, lesser is the level of difficulty to exploit that particular vulnerability. The factors and the corresponding weights are described below:

**Attack Vector (AV):** This metric reflects the media by which vulnerability exploitation is possible. The different values corresponding to different types of media includes: Network (0.85), which we do not consider in this paper since the Science DMZ is not exposed to the Internet; Adjacent (0.62), signifying that a vulnerability can be exploited via adjacent network such as Wi-Fi or Bluetooth; Local (0.55) signifying that the attacker has to exploit it via a local application or local login; and Physical access (0.2), which we also do not consider for CPC.

**Privileges Required (PR):** This metric describes the level of privileges an attacker must possess before successfully exploiting the vulnerability. Its values include 'None' (0.85), if no authentication privilege are required; 'Low' (0.62), for normal authentication; and 'High' (0.27) for root user authentication.

**User Interaction (UI):** This metric determines whether the vulnerability can be done solely at the will of the attacker, or whether a separate interaction/entity must participate through some means. Its values include 'No' (0.85) or 'Yes' (0.62).

**Attack Complexity (AC):** This metric reflects the required conditions for a successful exploitation depending upon whether additional information about the target is required, such as, system configuration settings or computational exceptions. This metric's value is largest for the least complex attacks, specifically, 'High' (0.44) and 'Low' (0.77).

Based upon these metrics, CVSS computes the overall exploitability score of a vulnerability as:

$$P(X) = 8.22 \times AV \times PR \times UI \times AC \tag{1}$$

This exploitability for our purposes servers as the likelihood or probability of success of an attack exploiting a particular vulnerability. The higher the value of $P(X)$, greater is the probability of a vulnerability to attain an achievement. Based on Eqn. (1), value of E ranges between [0.121, 3.887]. In order to normalize the range in to probabilistic region [0,1], we use Min - Max Normalization as:

$$\begin{aligned} P(X)_{norm} = & \frac{P(X) - P(X)_{Min}}{P(X)_{Max} - P(X)_{Min}} \\ & \times (P(X)_{NewMax} - P(X)_{NewMin}) \\ & + P(X)_{NewMin} \end{aligned} \tag{2}$$

### *C. BAG construction example*

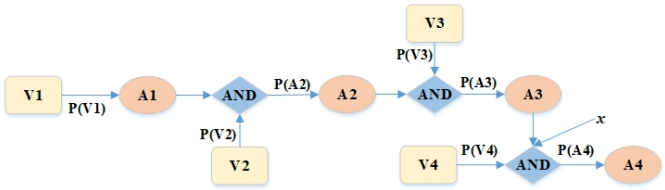



Fig. 3: Acyclic weighted Bayesian attack graph showing the causality behaviorial model of multiple vulnerabilities and achievements through conditional probabilities

Fig. 3 shows an exemplar an acyclic weighted BAG construction for a hypothetical CPC network architecture. As mentioned before, network architecture in terms of vulnerabilities $V$ present in the network, achievements $A$ and relationship $R$ amongst vulnerabilities and achievements are the key factor of designing such graph. Thus, Fig. 3 signifies that the CPC architecture has four vulnerabilities, e.g. $V1$, $V2$, $V3$, and $V4$ and achievements $A1$, $A2$, $A3$, and $A4$ can be attained by attackers through exploitation of the aforementioned vulnerabilities. According to the figure, the Bayesian precondition of achieving $A1$ is only through exploiting vulnerability $V1$ and thus $P(V1)$ is equal to $P(A1)$. Now if the attacker wants to achieve $A2$, then it needs to successfully exploit both vulnerabilities $V1$ and $V2$ as suggested by the 'AND' relationship. In such a case, the overall success probability of achieving $A2$ is calculated as,

$$P(A2) = P(A1) \times P(V2) = P(V1) \times P(V2) \tag{3}$$

In the same way, according to the BAG, the final achievement $A4$ can only be achieved through the simultaneous exploitations of vulnerabilities $V1$, $V2$, $V3$, and $V4$. The final 'AND' relationship is being influenced by an external factor with magnitude $x$ that impacts the outcome of the 'AND' relationship. The nature of such external factor although not required for BAG construction, can range from system configuration to human interaction. However, it is important to compute the factor magnitude which will work as a factor to the relationship outcome. Thus the probability of success of the final achievement $A4$ can be computed as,

$$\begin{aligned} P(A4) &= P(A3) \times P(V4) \times x \\ &= P(V1) \times P(V2) \times P(V3) \times P(V4) \times x \end{aligned} \tag{4}$$

It is to be noted that accordingly to the relationship, with any of the four vulnerabilities being exploited unsuccessfully or the factor with zero value, the probability of achieving $A4$ is 0.

In Section V, we will show concrete examples of BAGs constructed from existing vulnerabilities in the CUNY research network topology.

## V. Case Study: CUNY research network

We perform a case study on CUNYNet and CUNY CPC to demonstrate the utility of VM migration based MTD systems in reducing the probability of LoC, LoI, and LoA attack success using our proposed BAG model. The CUNY research network as shown in Figure 1 serves as unique case study candidate as unlike other campus research networks, CUNYNet is a city-wide distributed network consisting of senior research colleges (e.g., Hunter College, City College) and community colleges (Borough of Manhattan Community College, Brooklyn Community College) networks with all connecting to CUNYNet. The research colleges are connected through 100G links to facilitate data intensive science collaborations, whereas the community colleges are connected via 10G. CUNYs Science DMZ is a part of CUNY CPC located at Staten island that has a direct connection to New York State research network (NYSERNet) for intra-state and inter-state high speed science data transfer. The science data collaboration between CUNYNet and the Science DMZ is established via CUNYNet border router and Science DMZ OpenFlow vSwitch (OVS) controlled by an SDN controller. The CUNY CPC resources within the Science DMZ is virtualized as illustrated in Figure 4. The community colleges have access to the Science DMZ via edge routers under the scrutiny of firewalls. Whereas, research colleges can connect directly to Science DMZ via core routers and CUNYnet border router for friction free data movement.

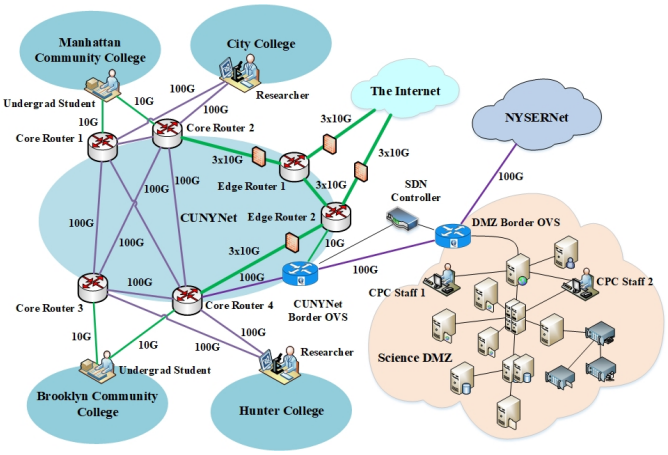



Fig. 4: CUNYNet topology consisting on distributed college campus networks and Science DMZ infrastructure connection with NYSERNet

For this case study, we will construct the BAG for different cases where the attackers launch the attack from different locations in the CUNYNet. These cases are:

**Case 1 - Community Colleges:** The attackers can only access to File Server with consent (authentication). The traffic goes through at least one core router, and one edge routers under the firewall's purview before reaching the CUNYNet border OpenFlow Switch. The traffic enters Science DMZ via OVS and Web server.

**Case 2 - Research Colleges:** The attackers can only access to File Server with authentication. However, their traffic only goes through at least one core router and directly into the CUNYNet border OpenFlow Switch on 100G links without firewall's surveillance. The traffic enters Science DMZ via OVS and Web server.

**Case 3 - Inside the CPC:** The attackers are inside the CPC's Science DMZ network with direct connection to Application server and can access to Database server without authentication.

For each of these cases, four virtual servers can be the attack targets: *Target A - File Server*, *Target B - Database Server*, *Target C - Email Server*, and *Target D - Application Server*. BAG will be designed for two distinct scenarios, one where the target virtual servers (Database and file servers) do not employ MTD based proactive VM migration (*Scenario I*) and another scenario where they do employ MTD (*Scenario II*). Below are the constructed BAG with explanations.

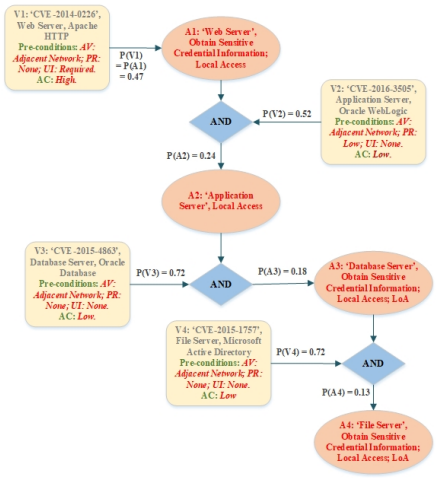



Fig. 5: Example BAG: Scenario 1 - Case 1 - Target A

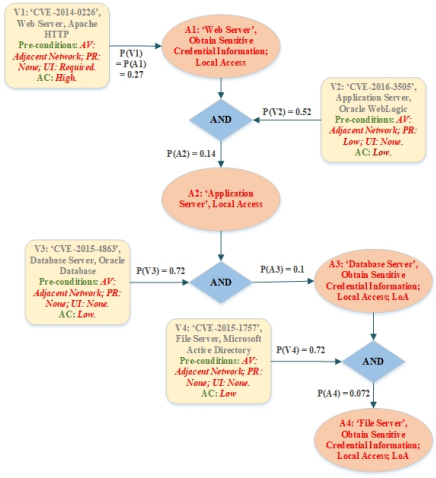



Fig. 6: Example BAG: Scenario 1 - Case 2 - Target A

Figure 5 represents the BAG when the attacker is launching the attack from one of the community colleges (Case 1) to target the CPC file server (Target A) with no MTD mechanism employed (Scenario I). Here we analyze a scenario where

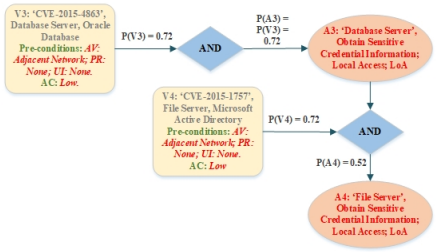



Fig. 7: Example BAG: Scenario 1 - Case 3 - Target A

the attacker could use vulnerability V1: CVE-2014-0226 to compromise Apache HTTP Server to obtain sensitive credential information as well as execute arbitrary code. Given the network architecture and situation, the vulnerability can be exploited from adjacent network (AV = 0.62), with special no privileges required (PR = 0.85), from inside the CUNYNet (UI = 0.62), and with high Complexity (AC = 0.44). Therefore, the overall exploitability of and normalized likelihood of success of $V1$ is computed as:

$$P(V1) = normalize(8.22 \times 0.62 \times 0.85 \times 0.62 \times 0.44) = 0.47$$

Here $P(A1) = P(V1) = 0.47$ since achievement $A1$ does not require any more vulnerabilities to be exploited beside $V1$. Similarly, we calculate $P(V2) = 0.52$; $P(V3) = P(V4) = 0.72$ for exploitability of $V2$. Here $A2$ can only be achieved if and only if $A1$ is already achieved along with $V2$ being exploited. Therefore $P(A2) = P(A1) * P(V2) = 0.47 \times 0.53 = 0.24$. The figure shows that Target A can only be fully compromised if achievement $A4$ is attained and the corresponding likelihood of success is $P(A4) = P(A3) \times P(V4) = 0.18 \times 0.72 = 0.13$. Thus, due to the complexity of the network and the significant network distance between the attacker and the target, probability of attack success $P(A4) = 0.13$ is significantly less than intermediate achievements/compromises, such as, obtaining login credentials from the web server.

Figures 6 and 7 demonstrate the constructed BAGs for the same target (Target A) without MTD (Scenario I), however with the attacks being launched from different parts of the CUNY research network in order to analyze their probabilities of success. In Figure 6, the attacker is at one of the community colleges' campus network. Therefore, the attack needs to penetrate through the firewall system unlike Case 1. According to CVSS guidelines, we assume that the presence of firewall will limit the probability of the vulnerability exploitation to 0.2. Hence, new $P(V1)$ is 0.27 instead of 0.47. We follow similar logic to Case 1 to calculate $P(A4) = 0.072$ which is significantly less than Case 1. Meanwhile, Figure 7 illustrates Case 3 where the attacker is at CPC internal network with direct connection to the application. Therefore, they can skip achievements $A1$ and $A2$ to jump directly to $A3$ by exploiting vulnerabilities $V3$ and $V4$. In this case $P(A4) = P(A3) \times P(V4) = 0.72 \times 0.72 = 0.52$ which is significantly greater than the previous two cases signifying that if the attacker has access to the CPC internal network, then launching a successful attack on the File server is easier.

Next, we analyze and construct BAGs for scenarios where the CPC employs MTD based VM migration technique on the database and file servers to randomize the target application location. In Figure 8, we specifically show an example of an attack launched from a community college campus (Case 1) on the file server (Target A) where the application is proactively moved among five candidate VMs. We assume that all the other services, such as, web server, application server, and database server are static in nature, i.e., being hosted on a particular VM. The figure shows that with all other likelihoods being the same as Figure 5, likelihood of $A3$ and $A4$ are one fifth of the static scenario resulting the magnitude of the external factor $x$ being 0.2. This results the overall probability of compromising the file server significantly reduced, i.e., $P(A4) = P(A3) \times P(V4) \times 0.2 = P(A2) \times P(V3) \times 0.2 \times P(V4) \times 0.2 = 0.005$. It is clearly evident that if all the intermediate servers are being proactively migrated between VMs, the overall likelihood of file server compromise would have been even less.

Figure 9 shows the table for all possible combinations of cases, target and scenarios. For all scenarios and targets, the difference between Case 1 and Case 2 is the probability $P(V1)$, due to the existence of firewall adding extra burden towards attaining the achievements. Where Case 3 consistently generates higher probability (wherever applicable) due to the attackers presence in the CPC network. Now for all cases and targets, if we compare the scenarios with and without MTD, then with MTD, the system resilience is consistently higher for same case and target without MTD (Different scenarios with same case and target are color coordinated).

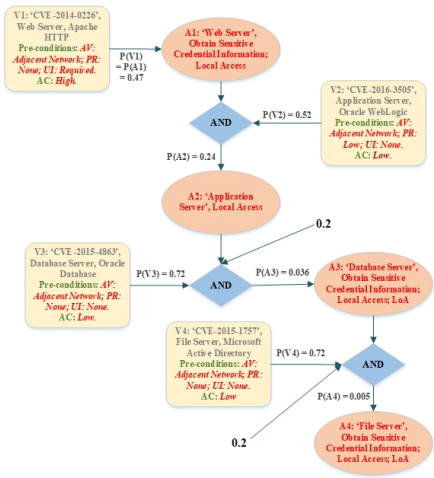



Fig. 8: Example BAG: Scenario II - Case 1 - Target A.

## VI. GENI-BASED IMPLEMENTATION AND EVALUATION

In this section, we describe the CUNY research network testbed implementation on GENI framework followed by system resilience performance evaluation with and without MTD implementation against confidentiality, integrity, and availability attacks.

### A. GENI topology and attack experiments

We create a scaled-down version of CUNYNet and CUNY CPC Science DMZ network topology in GENI as a testbed. The testbed includes one CUNYNet border OVS and DMZ OVS being controlled by an SDN controller. The virtualized database and email servers (not shown in the figure) work as the targets to the cyber attack experiments. For the cyber attacks, ten well known vulnerabilities are chosen that are relevant to LoC, LoI, and LoA attacks. Now for the vulnerabilities to be successfully exploited, we programmed the VMs with certain preconditions or pre-configurations.

| Scenario | Target | Case | Probability | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | P(V1) | P(V2) | P(V3) | P(V4) | P(A1) | P(A2) | P(A3) | P(A4) |
| I | A | 1 | 0.47 | 0.52 | 0.72 | 0.72 | 0.47 | 0.24 | 0.18 | 0.13 |
| | | 2 | 0.27 | 0.52 | 0.72 | 0.72 | 0.27 | 0.14 | 0.1 | 0.072 |
| | | 3 | NA | NA | 0.72 | 0.72 | NA | NA | 0.72 | 0.52 |
| | B | 1 | 0.47 | 0.52 | 0.72 | 0.72 | 0.47 | 0.24 | 0.18 | NA |
| | | 2 | 0.27 | 0.52 | 0.72 | 0.72 | 0.27 | 0.14 | 0.1 | NA |
| | | 3 | NA | NA | 0.72 | 0.72 | NA | NA | 0.72 | NA |
| | C | 1 | 0.47 | 0.52 | NA | NA | 0.47 | 0.24 | NA | NA |
| | | 2 | 0.27 | 0.52 | NA | NA | 0.27 | 0.14 | NA | NA |
| | | 3 | NA | NA | NA | NA | NA | 1 | NA | NA |
| | D | 1 | 0.47 | 0.52 | 0.37 | 0.72 | 0.47 | 0.24 | 0.12 | 0.07 |
| | | 2 | 0.27 | 0.52 | 0.37 | 0.72 | 0.27 | 0.14 | 0.07 | 0.04 |
| | | 3 | NA | NA | 0.37 | 0.72 | NA | NA | 0.39 | 0.22 |
| II | A | 1 | 0.47 | 0.52 | 0.72 | 0.72 | 0.47 | 0.24 | 0.036 | 0.005 |
| | | 2 | 0.27 | 0.52 | 0.72 | 0.72 | 0.27 | 0.14 | 0.02 | 0.003 |
| | | 3 | NA | NA | 0.72 | 0.72 | NA | NA | 0.144 | 0.021 |
| | B | 1 | 0.47 | 0.52 | 0.72 | 0.72 | 0.47 | 0.24 | 0.035 | NA |
| | | 2 | 0.27 | 0.52 | 0.72 | 0.72 | 0.27 | 0.14 | 0.02 | NA |
| | | 3 | NA | NA | 0.72 | 0.72 | NA | NA | 0.144 | NA |
| | C | 1 | 0.47 | 0.52 | NA | NA | 0.47 | 0.24 | NA | NA |
| | | 2 | 0.27 | 0.52 | NA | NA | 0.27 | 0.14 | NA | NA |
| | | 3 | NA | NA | NA | NA | NA | 1 | NA | NA |
| | D | 1 | 0.47 | 0.52 | 0.37 | 0.72 | 0.47 | 0.24 | 0.018 | 0.003 |
| | | 2 | 0.27 | 0.52 | 0.37 | 0.72 | 0.27 | 0.14 | 0.01 | 0.0014 |
| | | 3 | NA | NA | 0.37 | 0.72 | NA | NA | 0.074 | 0.011 |

Fig. 9: Table for all possible combinations of scenarios, cases, and targets

For example, vulnerability 'CVE-2016-0499' requires 'Low' access privilege, if the attacker does not have any privilege to the target VM, the exploitation will be unsuccessful resulting attack failure. For each VM, we randomly generate one of the following five pre-configurations that are relevant for the ten vulnerabilities. These are:

1) Application/Directory Server - *Connectivity:* Adjacent; *Priority:* None; *Interaction:* No.
   Database/Email Server - *Connectivity:* Local; *Priority:* None; *Interaction:* Yes.
2) Application/Directory Server - *Connectivity:* Adjacent; *Priority:* High; *Interaction:* Yes.
   Database/Email Server - *Connectivity:* Adjacent; *Priority:* None; *Interaction:* Yes.
3) Application/Directory Server - *Connectivity:* Adjacent; *Priority:* Low; *Interaction:* Yes.
   Database/Email Server - *Connectivity:* Local; *Priority:* High; *Interaction:* Yes.
4) Application/Directory Server - *Connectivity:* Adjacent; *Priority:* Low; *Interaction:* No.
   Database/Email Server - *Connectivity:* Adjacent; *Priority:* Low; *Interaction:* No.
5) Application/Directory Server - *Connectivity:* Local; *Priority:* None; *Interaction:* No.
   Database/Email Server - *Connectivity:* Adjacent; *Priority:* Low; *Interaction:* Yes.

The experiments results are compared with and without MTD where servers are proactively migrated between two candidate VMs for the cases with MTD based adaptations.

### B. LoC/LoI attack results with proactive MTD

Figures 10(a) and 10(b) show the outcome tables of LoC and LoI attacks launched on the database and email servers respectively with and without proactive MTD scheme. The tables show application server (AS), directory server (DS), database server (DBS), email server (ES) configurations along with the final outcome (FO). The table entries **S**, **F**, and **X** mean successful exploitation, failed exploitation, and the attacker cannot reach this step (i.e. failed in one of a previous step) respectively.

In Figure 10(a) for example, with pre-configuration 1 of the application server, although its 'Connectivity: Adjacent' satisfies the vulnerability attack vector (i.e., it can be reached via adjacent network) and its 'Interaction: No' satisfy vulnerability user interaction (i.e., the attacker does not have any extra interaction on the machine), its 'Priority: None' does not satisfy the vulnerability priority as the vulnerability requires at least 'Low' priority. Thus, the attack failed. On the contrary, for pre-configuration 2, the application server connectivity is 'High', signifying that the attacker has 'High' authentication information on the machine (e.g. root user information) which can be used to exploit the machine successfully. Thus the attack is successful.

The rate of attack success for scenarios with 'No MTD' is $13/50 = 0.26$ for the database server. However, combined with application servers, the final outcome (FO) in terms of attack success is $0.12$. Meanwhile for scenarios with 'MTD', the success rate is just $2/50 = 0.04$ for the database server, and the FO is $0.019$ which is significantly less than with 'No-MTD'. Similarly for email server shown in Figure 10(b), the FO for 'No MTD' and 'MTD' scenarios are $0.05$ and $0.005$ receptively, thus corroborating with the trends found in the theoretical results with BAG.

### C. LoA attack results with reactive MTD

Figure 11(a) illustrates the impact of LoA attack intensity on average response time of the database server. As expected, the response time increases with attack intensity as large number of packets sent by the attacker. Moreover, the number of concurrent users also play an important part of this outcome as the impact is more severe as the number of concurrent users increases. Figures 11(b), and 11(c) demonstrate the utility of different reactive MTD schemes upon reception of attack trigger. Both figure indicate sharp increase in response time during attack followed by return to normalcy upon MTD based VM migration. Figure 11(c) shows the performance benefits pf an intelligent MTD scheme where ideal VM location is selected by the SDN controller where the chosen destination. Whereas, if the controller selects a new VM location in a greedy manner (as shown in Figure 11(b)) without considering VM suitability, the response time improvement is not as good. However, the utility of MTD is evident for any underlying VM selection scheme.

## VII. Conclusions

In this paper, we demonstrated MTD as a cost-effective, easily implementable, and flexible solution for CPC to mitigate confidentiality, integrity, and availability attacks both theoretically and experimentally. For theoretical demonstration, we proposed a CVSS inspired BAG based threat assessment model and performed a case study on CUNY research network creating different cases, targets and scenarios. Experimentally, we use GENI framework to create a CUNY research network testbed where we launch cyber attacks and observe the system resilience with and without MTD based VM migration mechanism. Both theoretical and experimental results overwhelmingly support our claim of MTDs efficacy in improving system resilience. The results of this work can help campus cyber infrastructure engineers to utilize SDN programmability in employing MTD based cloud resource maneuvers and facilitate CPC adoption on university campuses and improve science researchers' confidence on campus wide data collaboration and dissemination.

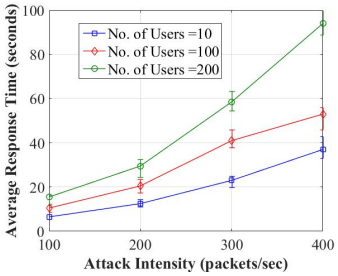
| Config. \ Vul. | | CVE-2016-3505 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| AS | 1 | No MTD | F | F | F | F | F | F | F | F | F | F |
| | 2 | No MTD | S | S | S | S | S | S | S | S | S | S |
| | 3 | No MTD | S | S | S | S | S | S | S | S | S | S |
| | 4 | No MTD | S | S | S | S | S | S | S | S | S | S |
| | 5 | No MTD | F | F | F | F | F | F | F | F | F | F |
| DBS | 1 | No MTD | X | X | X | X | X | X | X | X | X | X |
| | | MTD | X | X | X | X | X | X | X | X | X | X |
| | 2 | No MTD | F | F | F | F | F | S | F | F | F | F |
| | | MTD | F | F | F | F | F | F | F | F | F | F |
| | 3 | No MTD | S | S | S | S | S | S | S | S | S | S |
| | | MTD | F | F | S | F | F | F | F | F | F | F |
| | 4 | No MTD | F | F | F | F | F | F | F | F | S | S |
| | | MTD | F | F | F | F | F | F | F | F | F | F |
| | 5 | No MTD | X | X | X | X | X | X | X | X | X | X |
| | | MTD | X | X | X | X | X | X | X | X | X | X |
| FO | 1 | No MTD | F | F | F | F | F | F | F | F | F | F |
| | | MTD | F | F | F | F | F | F | F | F | F | F |
| | 2 | No MTD | F | F | F | F | F | S | F | F | F | F |
| | | MTD | F | F | F | F | F | F | F | F | F | F |
| | 3 | No MTD | S | S | S | S | S | S | S | S | S | S |
| | | MTD | F | F | S | F | F | F | F | F | F | F |
| | 4 | No MTD | F | F | F | F | F | F | F | F | S | S |
| | | MTD | F | F | F | F | F | F | F | F | F | F |
| | 5 | No MTD | F | F | F | F | F | F | F | F | F | F |
| | | MTD | F | F | F | F | F | F | F | F | F | F |

| Config. \ Vul. | | CVE-2015-5349 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DS | 1 | No MTD | F | F | F | F | F | F | F | F | F | F |
| | 2 | No MTD | F | F | F | F | F | F | F | F | F | F |
| | 3 | No MTD | S | S | S | S | S | S | S | S | S | S |
| | 4 | No MTD | S | S | S | S | S | S | S | S | S | S |
| | 5 | No MTD | F | F | F | F | F | F | F | F | F | F |
| ES | 1 | No MTD | X | X | X | X | X | X | X | X | X | X |
| | | MTD | X | X | X | X | X | X | X | X | X | X |
| | 2 | No MTD | X | X | X | X | X | X | X | X | X | X |
| | | MTD | X | X | X | X | X | X | X | X | X | X |
| | 3 | No MTD | S | S | S | S | S | S | S | S | S | S |
| | | MTD | F | F | F | F | F | F | F | F | F | F |
| | 4 | No MTD | F | F | F | F | F | F | F | F | F | S |
| | | MTD | F | F | F | F | F | F | F | F | F | F |
| | 5 | No MTD | X | X | X | X | X | X | X | X | X | X |
| | | MTD | X | X | X | X | X | X | X | X | X | X |
| FO | 1 | No MTD | F | F | F | F | F | F | F | F | F | F |
| | | MTD | F | F | F | F | F | F | F | F | F | F |
| | 2 | No MTD | F | F | F | F | F | F | F | F | F | F |
| | | MTD | F | F | F | F | F | F | F | F | F | F |
| | 3 | No MTD | S | S | S | S | S | S | S | S | S | S |
| | | MTD | F | F | F | F | F | F | F | F | F | F |
| | 4 | No MTD | F | F | F | F | F | F | F | F | F | S |
| | | MTD | F | F | F | F | F | F | F | F | F | F |
| | 5 | No MTD | F | F | F | F | F | F | F | F | F | F |
| | | MTD | F | F | F | F | F | F | F | F | F | F |

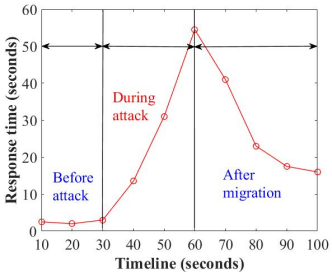
(a) Database server

(b) Email server

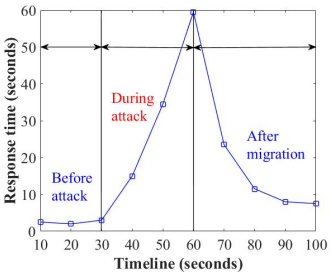
Fig. 10: Experiment results of LoI and LoC attacks with and without proactive MTD




(a) Characteristics against varying attack intensity




(b) Average response time before and after migration using greedy scheme




(c) Average response time before and after migration using intelligent scheme

Fig. 11: Average response time characteristics of database server with and without reactive MTD under availability attacks of different intensity

## REFERENCES


[1] R. Mount, D. Skinner, "Scientific collaborations for extreme-scale science workshop report," *US Department of Energy*, Tech. Rep, 2011.

[2] Cyber Security Division, "Project: Moving Target Defense," *US Department of Homeland Security - https://www.dhs.gov/science-and-technology/csd-mtd*.

[3] S. Debroy, P. Calyam, M. Nguyen, A. Stage, V. Georgiev, "Frequency-minimal moving target defense using software-defined networking," *Proc. of IEEE ICNC*, 2016.

[4] I. Monga, E. Pouyoul, C. Guok, "Software-defined networking for Big-Data science - Architectural models from campus to the WAN," *Proc. of IEEE SCC*, 2012.

[5] "CVSS Specification Document" - *https://www.first.org/cvss/specification-document*.

[6] J. Pearl, "Fusion, propagation, and structuring in belief network", *ACM Journal of Artificial Intelligence*, 1986.

[7] C. Liu, A. Singhal, D. Wijesekara, "Using Attack Graphs in Forensics Examination", *Proc. of IEEE ARES*, 2012.

[8] "CVSS User Guide" - *https://www.first.org/cvss/user-guide*.

[9] "Core Functions" - *http://www2.cuny.edu/about/administration/offices/cis/core-functions*.

[10] "CUNY High Performance Computing Center" - *https://cunyhpc.csi.cuny.edu*.

[11] "Common Vulnerabilities and Exposures" - *https://cve.mitre.org*.

[12] "NSF GENI Infrastructure" - *https://www.geni.net*.

[13] "Snort" - *https://www.snort.org*.

[14] L. Spitzner, "Honeypots tracking hackers", *Addison-Wesley*, 2002.

[15] "Incommon" - *https://www.incommon.org/*.

[16] "OpenID" - *http://openid.net/*.

[17] "X.509 Specification" - *http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0.pdf*.

[18] B. Baker, K. Borne, T. Handley, J. Kantor, J. Hughes, R. Lambert, C. L. Lee, H. Larrieu, R. Plante, "LSST data management cybersecurity draft plan," 2015.

[19] T. Carroll, M. Crouse, E. Fulp, K. Berenhaut, "Analysis of network address shuffling as a moving target defense," in *Proc. of IEEE ICC*, 2014.

[20] Q. Jia, H. Wang, D. Fleck, F. Li, A. Stavrou, W. Powell, "Catch me if you can: A cloud-enabled DDoS defense," in *Proc. of IEEE/IFIP DSN*, 2014.

[21] J. H. Jafarian, E. Al-Shaer, Q. Duan, "Openflow random host mutation: Transparent moving target defense using software defined networking," in *Proc. of ACM HotSDN*, 2012.

[22] P. Kampanakis, H. Perros, T. Beyene, "SDN-based solutions for moving target defense network protection," in *Proc. of IEEE WoWMoM*, 2014.

[23] R. Zhuang, S. Zhang, A. Bardas, S. DeLoach, X. Ou, and A. Singhal, "Investigating the application of moving target defenses to network security," in *Proc. of IEEE ISRCS*, 2013.

[24] I. Kotenko, M. Stepashkin, "Attack graph based evaluation of network security," in *Proc. of IFIP TC-6 TC-11*, 2006.

[25] K. Kaynar, F. Sivrikaya, "Distributed Attack Graph Generation," in *Proc. of IEEE TDSC*, 2016.

[26] M. Frigault, L. Wang, A. Singhal, S. Jajodia, "Measuring network security using dynamic bayesian network" in *Proc. of ACM QoP*, 2008.

[27] P. Xie, J. H. Li, X. Out, P. Liu, R. Levy, "Using Bayesian Networks for Cyber Security Analysis," in *Proc. of IEEE/IFIP DSN*, 2010.

[28] K. Durkota, V. Lisy, C. Kiekintveld, B. Bosansky, M. Pechoucek, "Case Studies of Network Defense with Attack Graph Games," in *IEEE Journal of Intelligent Systems*, 2016.

[29] E. Dart, B. Tierney, E. Pouyoul, J. Breen, "Achieving the Science DMZ," *Joint Techs*, 2012.

[30] "OpenFlow Switch Specification," *Open Networking Foundation*, 2015.