Journal of Telecommunication, Electronic and Computer Engineering
ISSN: 2180 – 1843 e-ISSN: 2289-8131 Vol. 17 No. 3 (2025) 7-18
jtec.utem.edu.my
DOI: https://doi.org/10.54554/jtec.2025.17.03.002

Reward Based Metrics for Assessing the Effectiveness of Shuffled Based Moving Target Defense

Abba Hali*, P. B. Zirra
Department of Computer Science, Faculty of Science, Federal University of Kashere, Gombe State, 760222, Nigeria.

Article Info

Article history:
Received Jan 8th, 2025
Revised Jun 13th, 2025
Accepted Jun 16th, 2025
Published Sep 30th, 2025

Index Terms:
Attack surface
Effectiveness
Dynamic network changes
Security metrics

Abstract

The research evaluates the effectiveness of shuffle-based Moving Target Defense (MTD) on host and network systems using a temporal graph-based security model, T-HARM. A novel dynamic security metric, the Reward-Based Metric (RBM), is introduced to assess the impact of MTD from the defender’s perspective, capturing changes in system resilience and attacker effort. The study involves implementing shuffle-based MTD techniques, defining and integrating the new metric with existing ones, and conducting simulation-based experiments to analyze security posture over time. The results show that the dynamic metric more accurately reflects real-time security changes, and that shuffle-based MTD significantly increases attack complexity and delays system compromise, thereby enhancing overall network defense.

*Corresponding Author: abbahali@fukashere.edu.ng

I. INTRODUCTION

Cyber security threats have become increasingly sophisticated and pervasive in today’s interconnected and digital world. Traditional static defense mechanisms, such as like firewalls and intrusion detection systems, are no longer sufficient to counter the dynamic and evolving cyber-attacks [1]. To address these limitations, Moving Target Defense (MTD) has emerged as a proactive cybersecurity approach with significant potential. MTD has the potential to dynamically alters systems configurations, thereby making it difficult for attackers to exploit vulnerabilities [2]. Similarly, Moving Target Defense (MTD) is an innovative approach aimed at addressing the limitations of traditional defense mechanisms by implementing dynamic and constantly changing elements into the system's architecture, configuration, or operation [3][4].

The concept of MTD was developed in response to the recognition that conventional cybersecurity strategies focused primarily on strengthening defenses, which often left critical assets vulnerable once protections were breached. MTD offers a transformative approach designed to cope with the increasing complexity and unpredictability of modern cyber threats [5]. It can be implemented at three major levels, namely Application-based, Host-based, and Network-based levels [1]. The application-based MTD can be described as a technique that dynamically alters application code to reduce exposure to analysis and vulnerabilities, thereby hindering attackers. The Host-based MTD (Host-based and Software-based) involves modifying host or platform properties, such as operating is described as techniques that can modify host or platform properties, such as OS version, OS instance, or CPU architecture, to enhance the uncertainty the attacker. Network-based MTD is described as a technique that continuously changes network properties, such as IP addresses or port numbers to disrupt and mislead attackers.

Despite growing interest and adoption of MTD, there remains a critical need to evaluate its effectiveness across different levels (host and network) and techniques. Among these techniques, MTD strategies are typically categorize into shuffle, diversity, and redundancy approaches. This study specifically focuses on the shuffle-based technique and emphasizes the importance of comprehensive assessments using appropriate security metrics [6][7].

The study addressed key gaps identified in the literature, such as the lack of standardized metrics and limited comparative evaluations [2][8]. The primary objectives include implementing a shuffle-based MTD technique, developing new security metrics to assess the effectiveness of this strategy from a defender’s perspective, and performing comparative evaluations between existing and newly proposed security metrics.

The remainder of the paper is organized as follows, Section 2 presents related work; Section 3 explains the methodology with design and implementation scenarios; Section 4 discusses results and discussion; Section 5 presents a comparative analysis; and Section 6 concludes the research work with suggestions for future research.

II. RELATED WORKS

Numerous studies have been carried out to evaluate the effectiveness of MTD using security metrics. These research efforts often involve simulations, real-world experiments, or theoretical analysis. Ref. [9] explores the need for more robust methods to assess the effectiveness of defense mechanisms in the face of evolving cyberattacks.

Traditional evaluation approaches are often insufficient for capturing the complex interactions between attackers and defenders in dynamic environments. To address this, the article proposes a simulation-based approach to model realistic attack-defense scenarios. This allows for analysis of defense strategies in terms of adaptability, performance, and resilience, ultimately providing insights to improve dynamic defenses and support better decision-making in real-world network protection.

A recent study by [7] examines the balance between the security benefits and economic costs of implementing MTD strategies in cloud environments. While MTD techniques offer promising improvements in mitigating cyber threats, their impact on operational costs and resource consumption remains underexplored. The study aims to assess the effectiveness of MTD in enhancing cloud security while also evaluating its economic implications, such as increased costs and potential return on investment. It offers practical guidance for cloud service providers and organizations seeking to deploy MTD effectively while balancing security gains and financial feasibility.

In a related research, Ref. [10] addresses the need for effective cybersecurity measures in cloud environments, which are frequently targeted by cyberattacks. The article highlights the limitations of traditional security mechanisms in providing adequate protection in shared and variable cloud infrastructures. It focuses on evaluating the performance and effectiveness of specific (MTD techniques, namely shuffle and redundancy, in mitigating threats within these settings. This study aims to analyze their performance impacts to assess their feasibility for real-world deployment, generate empirical data for comparison, and provide recommendations for integrating these techniques into existing cloud security frameworks to enhance resilience against evolving cyber threats.

Another significant contribution by [11] highlights the importance of automating the assessment of MTD strategies in dynamic cloud environments. Manual evaluation processes are time-consuming, prone to errors, and insufficient for large-scale, reliable security assessments. This study proposes the development of an automated framework that streamlines deployment, monitoring, and evaluation of MTD strategies in cloud infrastructures. The framework aims to offer comprehensive insights into MTD effectiveness, considering performance, resilience, and scalability, while enabling practical, automated evaluation and implementation.

Ref. [12] critiques the limitations of traditional security metrics in evaluating MTD techniques, which requires continuous adaptation in dynamic environments. Existing metrics often fail to reflect the real-time performance and evolving threat landscape associated with MTD strategies. The article proposes the development of dynamic security metrics that can accurately assess the performance of MTD techniques across various attack scenarios. These metrics aim to establish a standardized framework for evaluation and comparison, supporting data-driven decision-making and the advancement of more robust MTD strategies based on empirical data and performance assessments.

Ref. [13] presents a novel approach to evaluating MTD effectiveness based on the System Attack Surface (SAS). This model is designed to accommodate enterprise-scale topologies and multi-layered Moving Target (MT) techniques. It addresses issues of inaccurate performance assessment caused by improper characterization of attacker and defender behaviors. By capturing how system resources are affected by both sides, the SAS model enables accurate parameter measurement and provide practical validation for evaluating the dynamic and systematic aspects of MTD.

Another research was conducted by [4] to explore various dimensions of MTD, including key roles, design principles, categorization, common attack scenarios, fundamental strategies, and areas of practical applications. The article provides an analytical overview of the benefits and drawbacks associated with each aspect of MTD, offering a holistic understanding of the approach.

Research by [14] was also conducted to address the challenge of evaluating network security in complex environments, where traditional metrics focus narrowly on specific aspects such as availability or confidentiality. The study proposes composite metrics that integrate multiple security dimensions to provide a more comprehensive assessment. Its objective is to develop a systematic framework that allows security analysts to quantify and compare the overall security posture of networks, enabling better decision-making. These metrics aim to adapt to the dynamic nature of threats and provide actionable insights for enhancing network security.

Ref. [15] addresses the limitations of traditional static defense mechanisms in protecting endpoints within SDN environments. The researchers highlight the vulnerability of predictable network configurations to sophisticated attacks, such as zero-day exploits and lateral movements. To counter these threats, the article proposes a shuffle-based Moving Target Defense (MTD) approach that dynamically alters network configurations. This strategy creates unpredictability, making it more difficult for attackers to exploit system weaknesses. The objective is to enhance endpoint security by reducing attack success rates and improving the overall resilience of SDN systems through continuous modification of the attack surface.

A similar research by [16] highlighted the limitations of traditional static defenses in combating sophisticated and evolving cyber-attacks. The authors point out that many current evaluations of MTD strategies rely on overly simplistic attack models, thereby limiting their practical value. To address this, the article recommends evaluating MTD techniques against more realistic and complex attack scenarios. By doing so, it aims to accurately assess the adaptability and real-world effectiveness of MTD strategies, offering deeper insights into their strengths, weaknesses, and practical applications in defending against sophisticated threats.

Ref. [17] addresses the vulnerability of static IP addresses, which can make network systems easy targets for scanning and exploitation attacks. The study proposes an adaptive IP hopping strategy that continuously changes IP addresses to increase unpredictability and security. To ensure the practicality of this method, a lightweight Convolutional Neural Network (CNN)-based detector is integrated for real-time detection of malicious activities, minimizing computational overhead. The goal is to evaluate this IP hopping method's effectiveness in preventing attacks while ensuring low resource usage, offering a scalable solution for enhancing network security in dynamic environments.

Ref. [18] explores the challenges of integrating the dynamic nature of MTD into traditional attack modeling frameworks, which are typically designed for static environments. The dynamic nature of MTD makes it difficult to accurately representing and reasoning about shifting attack surfaces in these models. To overcome this, the article proposes the development of formal methods that incorporate MTD into these models. This would allow security analysts to predict the impact of MTD on attack success rates and system vulnerabilities more effectively. The goal is to enable more precise security evaluations and support informed decision-making by understanding how MTD disrupt attacker strategies.

Furthermore, Ref [19] addresses the growing threat of Distributed Denial of Service (DDoS) attacks, which overwhelm networks and disrupt services. The study highlights the limitations of static defenses in responding to such attacks and proposes a comprehensive framework that leverages MTD techniques. By dynamically altering network configurations, the proposed framework seeks to enhance network resilience and hinder attackers' ability to sustain DDoS attacks. The article also evaluates the framework's performance in mitigating different DDoS attack types, emphasizing its effectiveness in preserving legitimate traffic and maintaining overall network performance.

A more recent study by [20] explores the challenges in assessing the effectiveness of advanced cybersecurity techniques, including deception and MTD, due to the static and predictable nature of traditional security approaches. The authors advocate for development of a network attack simulation environment to test these defense mechanisms under dynamic, real-world attack scenarios. This environment would provide empirical data to assess the adaptability and performance of deception and MTD mechanisms, supporting their integration into operational security frameworks and enhancing overall network resilience.

Despite advancement in the field, existing work lacks a metric that quantitatively reflects the defender's gain from deploying shuffle-based MTD strategies. This research addresses this gap by introducing a novel security metrics called the Reward-Based Metric (RBM). RBM is designed to capture security changes in the network resulting from the implementation of shuffled-based MTD and assess its effectiveness from the defender's perspective. The metric is demonstrated through simulations and case examples to highlight its practical applicability.

Additionally, the study also presents a comparative evaluation of the shuffle MTD approach using both the proposed RBM and conventional security metrics. This dual-layered analysis offers comprehensive insights into the performance and benefits of dynamic defense, with a particular focus on host- and network-level implementations.

The scope of this research concentrates on the implementation and effectiveness assessment of the shuffle-based MTD approach, using appropriate security metrics. The study specifically targets hosts and network-level MTD approaches and examines their impact on selected security metrics.

The research aims to evaluate the effectiveness of shuffle-based MTD technique compared to traditional static defense mechanisms. The significance of this study lies in its potential to advance the body of MTD research, guide cybersecurity decision-making, and support the establishment of industry best practices, ultimately contributing to stronger and more adaptive cybersecurity defenses.

III. RESEARCH METHODOLOGY

This study utilizes a temporal graphical security model called T-HARM by [21] to identify and evaluate the security characteristics of dynamic networks. The methodology follows a structured approach: first, the design and implementation of the shuffle-based MTD technique are outlined; second, the definition and construction of the T-HARM model are presented; and third, the MTD security metrics, along with examples of their computation are presented.

A. Design and Implementation

Implementing a shuffled-based MTD technique on a real network to evaluate its effectiveness involves several steps and technical considerations. The overall methodology used to achieve the objectives of the study is shown in Figure 1 below.

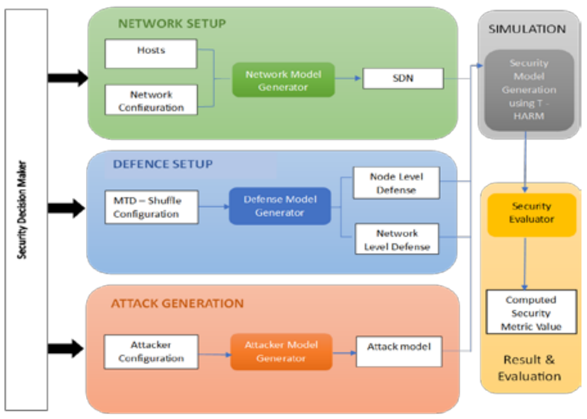

Figure 1. System framework and workflow

1) Network Setup

The network setup involves configuring the infrastructure and devices that constitute the experimental environment. The network architecture is built around an SDN framework, where an SDN controller manages OpenFlow switches connected to various hosts running different operating systems (e.g. Linux, Windows, and FreeBSD). The SDN controller can dynamically alter the flow table in OpenFlow switches, enabling real-time reconfiguration of the logical network topology.

It is assumed that the presence of at least two or more hosts with different OS variants to foster network diversity. The goal is to increase complexity of potential attackers by optimizing the number of variations within communication paths between hosts. Thus, it increases the difficulty for attackers by constantly altering the network's attack surface. For example, changing a host from Windows to Linux maintains the same functionality but forces an attacker to adapt to a new set of vulnerabilities in order to compromise the system. This setup allows for centralized control over network policies and configurations. Ports are configured to segment traffic, which enhances both security and manageability. A firewall is strategically placed between the SDN controller and the rest of the network to monitor and control traffic entering and exiting the network, acting as a barrier against unauthorized access.

The network setup for the configuration of the SDN network is shown in Figure 2 and the reachability of the SDN network is illustrated in Figure 3. Meanwhile, the operating system running on the host within the network is detailed in Table 1, and the firewall rules for the network are presented in Table 2.

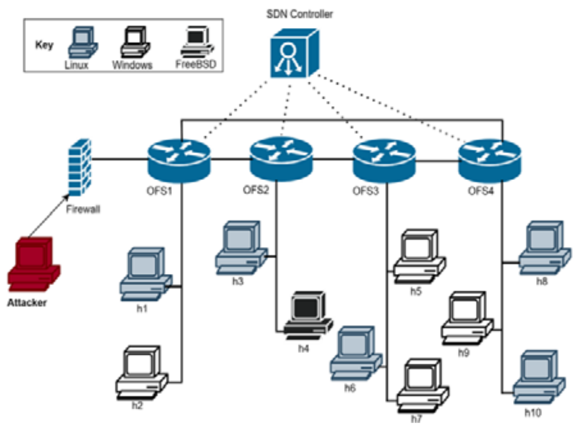

Figure 2. Configuration of an SDN network

From Figure 2, the attacker's reachability of the SDN network at different time intervals is shown in Figure 3.

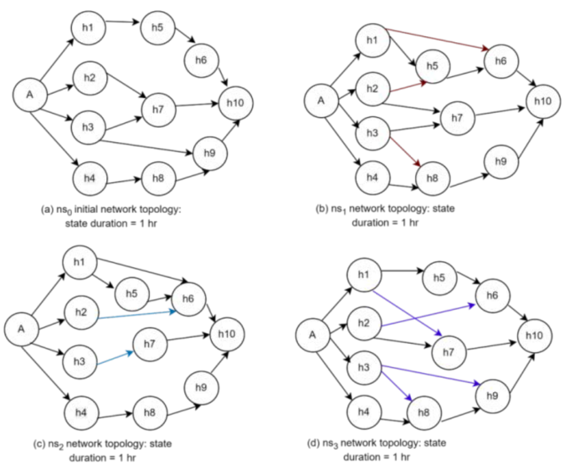
A
h1
h2
h3
h4
h5
h6
h7
h8
h9
h10
(a) ns0 initial network topology: state duration = 1 hr
(b) ns1 network topology: state duration = 1 hr
(c) ns2 network topology: state duration = 1 hr
(d) ns3 network topology: state duration = 1 hr

Figure 3. Reachability of an SDN network at different time intervals

Table 1
Operating System installed on the hosts

Host	Operating System	Service
h_1	Ubuntu 22.04	Google Chrome standalone64
h_2	Windows 10	Microsoft Word
h_3	Linux Ubuntu 22.04	Microsoft Edge
h_4	FreeBSD	Email server
h_5	Windows 11	Mozilla Firebox 31.1.0
h_6	Linux Ubuntu 22.04	Google Chrome standalone64
h_7	Windows 11	Mozilla Firebox 31.1.0
h_8	Linux Ubuntu 22.04	Mozilla Firebox 31.1.0
h_9	Windows 11	Google Chrome standalone64
h_{10}	Linux Ubuntu 22.04	Oracle Database Server

Table 2
Firewall Rules for the Network

Host	Allows connections from
h_1	Internet
h_2	Internet
h_3	Internet
h_4	Internet
h_5	h_1
h_6	h_5
h_7	h_2, h_3
h_8	h_4
h_9	h_3, h_8
h_{10}	h_6, h_7, h_9

2) *Attack Generation*

Ensuring the security of a system can be highly intricate due to the diverse architectures and software components requiring assessment. Furthermore, as systems grow larger, the process of assurance and testing becomes more time-consuming and costly. Therefore, it is crucial to prioritize critical points or components for protection and identify potential attack routes that adversaries could exploit. Failure to do so may result in inadequate protection, leaving the system vulnerable to exploitation by attackers.

Attack models represent different frameworks that incorporate security-related information such as attacks, exploits, threats, assets, and vulnerabilities. These models usually offer a comprehensive overview of a system or component's security posture, enabling further analysis and mitigation planning. They help security professionals better understand potential threats and identify the most likely targets for adversaries.

In the network, users can access various services, but it also open the door to potential attacks. An attacker may remotely compromise network hosts with relative ease. In this scenario, a directed attack is considered, where the attacker targets h10, which contains sensitive information. Possible entry points for the attacker (A) include hosts h1, h2, h3, and h4, as shown in Figure 3.

This scenario assumes a single attacker originating from outside the network (e.g., from the Internet or a partner network linked to the local/enterprise network, such as customers or vendors). The attacker's goal is to elevate privileges from a basic user account to administrator rights and ultimately steal sensitive data. The target host runs the Linux operating system and contains a known vulnerability identified as CVE-2014-5270 ID, which has an associated CVSS base score.

It is assumed that the attacker cannot directly compromise the target host. Instead, after successfully breaching an initial host, the attacker escalates to root privileges and uses the compromised host as a pivot point to move laterally through the network, repeating the process until reaching the final target. Additionally, since the SDN environment includes IP shuffling, the attack scenario must adapt to these dynamic changes.

In practice, potential real-world attack scenarios and attacker actions are often represented using attack graphs. These graphs illustrate various paths an attacker might take to reach their objective, such as exploiting multiple vulnerabilities within the SSH library [22]. Attack graphs serve as a foundational framework for more detailed attack model, which may be enhanced with additional elements such as the incorporation of conditional variables, probabilities, states, and transitions.

Furthermore, this methodology also emphasizes attack trees, which have a hierarchical structure with a root node representing the attack goal, followed by intermediate nodes representing sequential steps in the attack. Each intermediate node can branch into multiple child nodes, reflecting different options or method available to the attacker at each step. To establish connections and relationships between these nodes, various logic gates are utilized to construct attack chains. For example, an OR gate indicates that any one of several vulnerabilities can be exploited, while an AND gate signifies that all listed conditions must be fulfilled for the attack to succeed. The leaf nodes of the tree represent the initial actions the attacker must undertake to begin progressing toward the objective. Details of this structure are presented in Figure 4 in section 3.1.7.

3) Defense Setup

Defense setup involves implementing measures to prevent and mitigate attacks on a network by deploying appropriate security solutions and following best practices. This strategy considers both patchable vulnerabilities, which can be addressed through updates, and non-patchable vulnerabilities, for which no patches are currently available [23]. To protect IT assets from cyber-attacks, even when vulnerabilities are present, defense mechanisms such as reconfiguration and shuffling are employed.

In an SDN environment, shuffling can involve changing routing tables or IP addresses, disrupts attacker activities. This method alters the network structure without necessarily removing vulnerabilities, thereby delaying reconnaissance and impeding ongoing attacks. For example, in the SDN network shown in Figure 2 (Section 3.1.1), each host typically has a single IP address used to connect to other hosts. Here, the SDN controller, which manages data flows in the network, shuffled the IP addresses at every x second. As a result, the OpenFlow switches are updated to redirect network flows and maintain communication consistency. When attackers scan for IP addresses, the constant shuffling causes them to lose track of previously identified victim hosts, effectively reducing their chances of launching a successful attack.

4) Simulation

The simulation employs a temporal graphical security model, specifically the Temporal Hierarchical Attack Representation Model (T-HARM), which incorporates MTD techniques to capture the changes when a shuffled-based MTD technique is deployed. Shuffle-based security metrics are incorporated into the T-HARM model to assess the effectiveness of the shuffle-based MTD technique, as illustrated in Figure 4 (Section 3.1.7).

It is assumed that the attacker is positioned outside the network and is attempting to compromise a host within the network. The attacker must follow a defined sequence of actions, including reconnaissance and exploitation, to escalate privileges and breach the target host. Additionally, it is assumed that each host in the simulation has one vulnerability, denoted as *vi*, where *i* represents the host number.

Importantly, if the sequence of privilege escalation is disrupted, (for example, through a shuffling technique that redirects service routes), the attacker forfeits any previously gained privileges and reverts to the last accessible host in the sequence.

To enable this simulation, the T-HARM model is expanded to include MTD techniques by capturing the modifications made to the network.

The assumptions made in the simulations are:

i) The Dynamic Host Configuration Protocol (DHCP) automatically assigns IP address settings to hosts joining the network.
ii) Users are allowed to install software on this host, which may contain vulnerabilities.
iii) A security administrator can deploy defense measures, such as MTD techniques, update firewall rules, or make other security changes.
iv) Network states are recorded whenever changes are detected in the network**.**

5) Definition of T-HARM

Definition 1 from [21]: T-HARM is a three-part structure denoted as (N, H, V), where N = {n_{t0}, n_{t1}, . . ., n_{tn}} is a collection of HARM models associated with every network state s_{ti} at time t_i for i ∈ N, where each nti is associated with sti. H = {h0, h1, ..., hx} represents a finite set of all hosts in the network, and V = {v_0, v_1, ..., v_y} denotes a finite set of all vulnerabilities present in the network. Further, H_{ti} ⊆ H is the set of hosts present in the network state at time t_i, and similarly, V_{ti} ⊆ V represents the set of vulnerabilities in the network state at time t_i. The HARM for each network state is structured in two layers: the upper layer uses an Attack Graph to model host reachability, and the lower layer uses Attack Trees [12] to represent host-specific vulnerabilities. The individual HARM recorded at each time t_i can be defined as follows:

Definition 2 from [21]: A 2-layered HARM is a three-part structure denoted as a 3-tuple n_{ti} = (U_{ti}, L_{ti} , C_{ti}), where U_{ti} represents the upper layer utilizing an Attack Graph that captures host reachability and potential attack paths. L_{ti} serves as the lower layer, consisting of a set of Attack Trees that exclusively captures the vulnerability information for each host identified in the upper layer. C_{ti} represents the mapping between the hosts in the upper layer and their corresponding Attack Tree model in the lower layer (i.e., each host is associated with one Attack Tree). Finally, the definitions for the upper and lower layers of the HARM are defined as follows:

Definition 3 from [21]: The upper layer of the HARM is represented by a 2-tuple U_{ti} = (H_{ti}, E_{ti}) at time t_i, where H_{ti} is a finite set of hosts in the network, and $E_{ti} \subseteq H_{ti} \times H_{ti}$ refers to a set of edges, representing reachability between hosts.

Definition 4 from [21]: The lower layer of the HARM is composed of a set of Attack Trees, denoted as L_{ti}={l_{h1},l_{h2},…,l_{h10}} at time t_i, with each Attack Tree instance linked to a specific host from the upper layer $h_j \in H_{t_i}$ is a 4-tuple $l_{h_j} = \left(A_{t_i}, B_{t_i}, c_{t_i} root_{t_i}\right)$ with $l_{h_j} \in AT_{t_i}$ where $A_{t_i} \in V_{t_i}$is a collection of vulnerabilities, B_{ti} =$\left\{b_{t_i}^j \,\middle|\, b_{t_i}^j \in \{AND, OR\}\right|\Big\}$ is a collection of logical gates, $c_{t_i} \subseteq \left\{b_{t_i}^j \rightarrow e_k\right\} \forall b_{t_i}^j, e_k \in A_{t_i} \cup B_{t_i}$ It represents a mapping that links gates to vulnerabilities and other gates, where $root_{ti} \in A_{ti} \cup B_{ti}$ is the root (i.e., the root node, which can be either a vulnerability or a gate that connects vulnerabilities and other gates). Based on the definitions provided, the reachability of SDN, as illustrated in Fig. 3, is denoted as follows.

Example 1 T − HARM eg = (N_{eg}, H_{eg}, V_{eg}), where N_{eg} = {$n_{eg, t1}$, $n_{eg, t2}$}, H_{eg} = {A, h_1, h_2, . . ., h_{10}}, and V_{eg} = {v_1, v_2, . . ., v_{10}}. Further, $H_{eg, t1}$ = $H_{eg, t2}$ and the total number of $H_{eg, t1}$ = 14 while $H_{eg, t2}$ = 16 with V_{t1} = V_{t2} As shown in Fig. 4.

Example 2 The 2-layered HARM for the example at time t_1 is $n_{eg, t1}$ = ($U_{eg, t1}$, $L_{eg, t1}$, $C_{eg, t1}$).

Example 3 The upper layer of the HARM $n_{eg,t1}$ is $U_{eg,t1}$ = ($H_{eg,t1}$, $E_{eg,t1}$), where $H_{eg,t1}$ = H, and $E_{eg,t1}$ = {{A, h_1 }, {A, h_2 }, {A, h_3 }, {A , h_4 }, {h_1 , h_5 }, {h_2 , h_7 }, {h_3 , h_7 }, {h_3 , h_9 }, {h_4 , h_8 }, {h_5 , h_6 }, {h_6, h_{10}}, {h_7, h_{10}}, {h_8, h_9}, {h_9, h_{10}}} is a set of edges.

Example 4 lower layer of the HARM $n_{eg,t1}$ is a collection of Attack Trees $L_{eg,t1}$ = {l_{h1} , l_{h2} , . . ., l_{h10} }. For example, an Attack Tree for a host h_1 is l_{h1} = ($A_{h1,t1}$, $B_{h1,t1}$, $C_{h1,t1}$, $root_{h1,t1}$), where $A_{h1,t1}$ = v_1, $B_{h1,tt}$ = {OR_1}, $c_{h1,tt}$ = {(OR_1, v_1)}, and $root_{h1,t1}$ = OR_1. The HARM structure model represents the security posture of each network state, enabling the calculation of attack paths and the evaluation of vulnerability exploitability using the CVSS Base Score (BS) [12][21]. Building on the T-HARM definition, the next section will integrate Moving Target Defense (MTD) techniques based on their specific characteristics.

6) Shuffled-based MTD technique characteristic and definition

Shuffle-based MTD techniques modify current network configurations at different levels. The examples include migrating virtual machines (VMs) by [3], network path reconfiguration by [24], random host mutation by [25] at the network layer, and address space layout randomization by [26]. These techniques modify the network topology and/or the connectivity of hosts, enabling the capture of resulting topological changes and the creation of new instances at defined time intervals t_i. This process is referred to as T-HARM, as explained below:

Definition 5: Given a pair {h_x, h_y} (where $h_x \in H$, $h_y \in H$) to be updated (i.e., added or removed) at time t_{i+1} due to the shuffle-based MTD approach, the upper layer of the HARM in the model will reflect the changes in the network topology and host connectivity. n_{ti+1} changes to (H_{ti+1}, E_{ti+1}), where the collection of hosts remains unchanged (i.e., H_{ti+1} = H_{ti}), and E_{ti+1} E_{ti} ⊕ { (h_x , h_y) }.

Example 5: For the path reconfiguration demonstrated in the example SDN between $n_{eg,t1}$ and $n_{eg,t2}$ the transition is documented from $U_{eg,t1}$ to $U_{eg,t2}$ such that $H_{eg,t1}$ = $H_{eg,t2}$ = H (i.e., no changes to the hosts as previously defined), and E_{t2} = E_{t1} ⊕ { (h_1, h_6), (h_2, h_5), (h_3 , h_8), (h_3, h_9) }

7) Construction of the T-HARM

T-HARM is employed to track network modifications over distinct time intervals. A customizable time frame can be defined, allowing adjustments to different durations (e.g., 1 hour, 2 hours, or 1 day), which provides different perspectives for evaluating security over time. Figure 4 presents an example of the T-HARM for the network shown in Figure 2, demonstrating the scenario where a new workstation is added to the network.

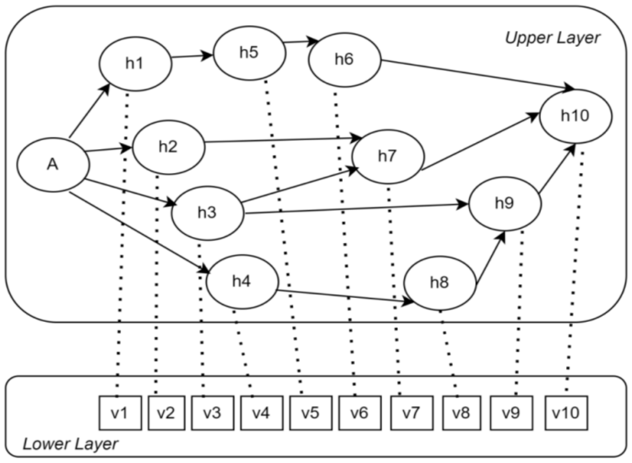

Figure 4. Example of T-HARM for attacker Model (Reachability of an SDN at t_1)

B. MTD Security Metrics

In this section, we describe the security metrics used to assess the effectiveness of the shuffle-based MTD technique which includes;

i) Attack Path Variation
ii) Attack Path Number
iii) Attack Path Exposure
iv) Attack Cost Exploitability

In addition, we proposed a new security metric named reward-based metrics (RBM), and this metric is described in Section 3.2.5

1) Attack Path Variation

A key component of attack surface lies in the attack paths. By monitoring changes in these paths, we can assess the increased effort required by attackers. In a static network, the potential attack routes remain unchanged unless the underlying vulnerabilities are altered. Therefore, the attack path variation metric is used to measure changes in potential attack paths that can be used to exploit vulnerabilities within a network. This metric quantifies how attack paths evolve as the network is modified or when MTD techniques are implemented. According to the definition provided, the change in attack paths between between two network states (i and i-1) is denoted as ΔAP (AP_i,AP_{i-1}) and is calculated using Equation 1.

$$\Delta AP_{i,\,i-1} = \frac{|AP_i - AP_{i-1}|}{|AP_i|} \tag{1}$$

Where AP_i represents the collection of the attack paths identified in ith network state. This formula measures the difference between the current and previous sets of attack paths. A low value suggests the most attach paths remain unchanged, reducing the effectiveness of MTD technique. A high value indicates that new paths have emerged due to network changes, reflecting the impact of MTD on attacker uncertainty.

The overall Attack Path Variation (APV) metric across all successive network states is computed using Equation 2.

$$APV = \frac{\sum_{i-1}^{|S|} \Delta AP_{i,\,i-1}}{|S| - 1} \tag{2}$$

Where [S] is the total number of network states. This metric provides a quantifiable measure of how attack paths shift overtime with the implementation of shuffle-based MTD. Consequently, no arbitrary values are required to compute the APV.

2) Attack Path Number (APN)

The Attack Path Number (APN) metric demonstrates the vulnerability of dynamic network states by measuring the number of paths available for potential attacks. As the number of attack paths increases, the likelihood of a successful attack also increases, as attackers gain more options to exploit, especially during scanning and reconnaissance phases. Therefore, it is crucial to monitor variations in the number of attack paths during network adjustments to understand how network security evolves in response to MTD implementation. The change in the number of attack paths between two consecutive network states is calculated using Equation 3.

.

$$\frac{\left|AP_{ns_{t_i}} - AP_{ns_{t_{i-1}}}\right|}{\left|AP_{ns_{t_i}}\right|} \tag{3}$$

This formulation means that if the number of attack paths remains unchanged, the metric yields 0. Conversely, if no attack paths existed in the prior state of the network, the metric yields 1. For cases where the number of attack paths decreases (assuming the attack effort remains constant), the metric can be computed using Equation 4

$$\frac{\max\left(\left|AP_{ns_{t_i}}\right| - \left|AP_{ns_{t_{i-1}}}\right|, 0\right)}{\left|AP_{ns_{t_i}}\right|} \tag{4}$$

The Attack Path Number (APN) refers to the total number of potential paths available for exploitation across network states. This metric provides insights into the overall complexity and vulnerability of the network's security posture by indicating the number of paths available for potential attacks. The aggregated and normalization APN metric is calculated using Equation 5

$$APN = \frac{\frac{\sum_{i=1}^{|NS|} \max\left(\left|AP_{ns_{t_i}}\right| - \left|AP_{ns_{t_{i-1}}}\right|, 0\right)}{\left|AP_{ns_{t_i}}\right|}}{|S|-1} \tag{5}$$

The APN metric does not require the assigning of an arbitrary value; it is based solely on recorded counts of attack paths across all network states (NS).

3) Attack Path Exposure

This metric calculates the duration that attack paths remain exposed in dynamic networks. The underlying assumption is that an attacker can effectively plan and execute an attack if the exposure time is extended. Therefore, increasing the duration of attack path exposures increase the attacker's opportunity to construct a successful attack, as also noted by [12]. Therefore, to enhance security, it is crucial to minimize the exposure time of each attack path, which in turn increases the effort required by attackers to success.

Attack Path Exposure refers to the degree to which potential paths for cyber-attacks are accessible or vulnerable within a network. It measures the susceptibility of a network to be exploited through various attack paths, highlighting the potential risks and weaknesses that attackers can exploit to infiltrate the network and compromise its security. In other words, the main purpose of this metric is to quantify the exposure time of each attack path. The function t(ap$_i$) calculates the exposure time for an attack path, as shown in Equation 6, and is normalized by the total number of attack paths and network states.

$$APE = \frac{\sum_{i=0}^{|NS|} t\left(ap_j\right)}{|AP| \times t\left(AP_{ns_{t_i}}\right)} \forall ap_j \in AP_{ns_{t_i}} \tag{6}$$

In this metric, if the original set of attack paths remains vulnerable across all network states without the appearance of new paths, the APE value will be 1. Conversely, a lower APE value indicates that an attack path is exposed in only a single network state, making a lower APE value more desirable. Therefore, a lower APE is more desirable, as it reflects reduced attacker opportunity and improved network security through decreased path exposure.

4) Attack Cost Exploitability (ACE)

This metric outlines the key constraints that influence the decision-making processes of both adversaries and protectors. The attack cost is evaluated based on the complexity of leveraging vulnerabilities, using the Common Vulnerability Scoring System (CVSS)[12]. Specifically, this study uses CVSS version 2, focusing on the exploitability score from the CVSS Base Equation, which quantifies the difficulty of exploiting a given vulnerability. The attacker's level of knowledge is also essential in determining the capacity to exploit vulnerabilities, especially those with lower vulnerability scores.

To determine the cost of exploiting vulnerability, each potential attack route is evaluated. Subsequently, the feasibility of an attack route is determined by multiplying the exploitability probabilities of all the vulnerabilities along that path. This yields the overall exploitation cost of specific states, ACns$_{ti}$ as defined in Equation 7

$$AC_{ns_{t_i}} = \prod_{j=1}^{\left|AP_{ns_{t_i}}\right|} \left(1 - \prod_{k=1}^{|ap_j|} Ep(vk)\right) \tag{7}$$

where: where v_(k)∈ vuls(ap_j)∀ap_(j)∈ AP_i

Using the above equation, the average attack cost exploitability across all network states is calculated using Equation 8.

$$ACE = \frac{\sum_{i=0}^{|NS|} ACns_{t_i}}{|NS|} \tag{8}$$

This formulation uses inner product operation to assess the exploitability of each attack path and integrates them using disjoint set theory. This approach is applied across all network states.

5) Reward-based Metric

This metric assesses the effectiveness of a shuffle-based security strategy from the defender's perspective. It quantifies the defender's efforts in deploying a shuffle defense technique across the network by assigning a reward value. This reward reflects the overall effort involved in the deployment and serves as an indicator of how well the strategy contributes to enhancing the network's security posture. In this context, the security administrator has the flexibility to define reward values based on organization's priorities and requirements. For the purpose of this research, reward values were normalized between 0 to 1, ensuring consistency and comparability across different scenarios. The formula for calculating this metric is provided in Equation 9.

$$RBM = \frac{\sum_{ns_{t_i} \in NS} \sum_{j=0}^{|ns_{t_i}|} r_j(ns_{t_i})}{|NS|} \tag{9}$$

In the above metric, r_j (ns_{ti}) represents a set of shuffles for a network state and [NS] is the total number of network states evaluated. By using the above metric, one can calculate the reward of all the network states to get the total reward of mitigation measures.

C. Example of Computation of Security Metrics

This section explains the application of the proposed security metrics and shows how both existing and new introduced security metrics are computed when shuffle-based MTD is deployed in the network. The SDN network and attacker model (i.e., host reachability) are shown in Figure 2 and 3 in Section 3.1.1 respectively.

Time intervals are assigned to network topologies for illustrative purposes, with durations (for example in hours) arbitrarily chosen. In practice, the actual duration of each network state may be defined by a reconfiguration schedule. The goal of the attacker is to reach the target host, h10, via privilege escalation, assuming that each host has a remote-to-root vulnerability with a uniform exploitability value of 0.2 for all operating systems. Additionally, it is assumed that the time required to update the set of edges for each host is based on the total number of edges added or removed compared to the previous network state. Considering this assumption, the security metrics APV, APN, APE, ACE, and RBM can be computed using the equations in Section 3.2.1 and Figure 3 in Section 3.1.1

1) APV Computation

From Figure 3, the APV metrics captures the attack path scenarios across the network states.

Example, AP_{ns0} = {(A, h_1, h_5, h_6, h_{10}), (A, h_2, h_7, h_{10}), (A, h_3, h_7, h_{10}), (A, h_3, h_9, h_{10}), (A, h_4, h_8, h_9 h_{10}). Is a collection of attack paths in ns_0 with a cardinality value |S| = 4 and a value of AP_0 = 5 = ns_0, while ns_1, ns_2, ns_3 have 7, 5, 7 respectively.

When $AP_{i=0}$ the value = 0;
$AP_{i=1}$, the value = 3
$AP_{i=2}$, the value = 1, and
$AP_{i=3}$, the value = 4

Therefore, by substituting these values in Equation 2, the researchers obtained the APV of 0.4. Thus,

$$\text{APV} = \left(\frac{\frac{3}{7} + \frac{1}{5} + \frac{4}{7}}{3} \right) = 0.4 \tag{10}$$

From the result, the APV value indicates the variations in attack paths across all network states within the SDN network. A lower APV value shows that the attack paths remain relatively stable. An APV value of 0.4 signifies the expected proportion of variations in attack paths across every network state observed.

2) APN Computation

From Figure 3, the APN can be computed as shown below.

When $AP_{ns_{ti=0}}$ the value = 0;
$AP_{ns_{ti=1}}$, the value = 2,
$AP_{ns_{ti=2}}$, the value = 0 and
$AP_{ns_{ti=3}}$, the value = 2

Therefore, substituting these values in Equation 5, the researchers obtained the APN value of 0.1905. Thus,

$$APN = \left(\frac{\frac{2}{7} + \frac{0}{5} + \frac{2}{7}}{3} \right) = 0.1905 \tag{11}$$

From the result, if the APN value moves closer to 1, it signifies a rise in the number of attack paths as the network shifts between different states. Conversely, if the APN value close to zero, it indicates a reduction in attack paths during transitions between network states. Therefore, it is preferable to maintain or reduce the number of attack paths rather than increase them.

3) APE Computation

From Figure 3, the APE metric assesses the extent of attack path exposure across all network states. Considering all network states, there are 10 potential attack paths. The exposure of each path can be computed as shown below.

When $AP_{i=0}$, the value = 5;
$AP_{i=1}$, the value = 7,
$AP_{i=2}$, the value = 5,
$AP_{i=3}$, the value = 7 with total AP = 10 and $t(ns_{t_i})$ = 19

Therefore, substituting these values in Equation 6 the researchers obtained the APE value of 0.1263. Thus

$$APE = \left(\frac{1(5)+1(7)+1(5)+1(7)}{10 \times 19} \right) = 0.1263 \tag{12}$$

From the results, when attack paths remain exposed across all network states without the emergence of new ones, the APE value tends toward 1. Therefore, a lower APE value is desirable, as it indicates that an attack path is exposed in only one or a limited number of network states.

4) ACE Computation

From Figure 3, the ACE metric evaluates how exploitable a target is for an attacker. The attack cost for exploiting vulnerabilities across all network states can be calculated as demonstrated below.

When $AP_{i=0}$, the value = 0.9731;
$AP_{i=1}$, the value = 0.9699,
$AP_{i=2}$, the value = 0.9731,
$AP_{i=3}$, the value = 0.9637 with number of network states (NS) = 4 and Vulnerability score v_k = 0.2.

Therefore, substituting these values in Equation 8, the researchers obtained the ACE value of 0.9699. Thus,

$$ACE = \left(\frac{0.9731+0.9699+0.9731+0.9637}{4}\right) = 0.9699 \qquad (13)$$

From the results, if the vulnerabilities have a high exploitability value (e.g., 1), the ACE value tends toward zero. Given that the exploitability of the assigned operating systems is considered low, the ACE metric value in this case was calculated to be near one. This indicates that the changes in the configuration decrease the difficulty of exploitation, thereby decreasing the attack efforts. Therefore, the network administrator has to increase the complexity of the network configuration to make the exploitation more difficult, aiming for the ACE value to trend towards zero (0).

5) RBM Computation

From Figure 3, the reward-based metric computes a reward for the defender after each shuffle (with a specific color of the arrow representing the shuffle) in a network state. Here, we assumed a reward value of 0.1 for each of the shuffles in a network state, and the reward for the defender's effort in each shuffle can be calculated as shown below.

For ns$_0$, the $r_j\left(ns_{t_i=0}\right)$ value = 0;
ns$_1$, the $r_j\left(ns_{t_i=1}\right)$ value = 0.3,
ns$_2$, the $r_j\left(ns_{t_{i=2}}\right)$ value = 0.2 and,
ns$_3$, the $r_j\left(ns_{t_3}\right)$ value = 0.4 with number of network states (NS) = 4.

Therefore, substituting these values in equation 9 the researchers obtained the RBM value of 0.225. Thus

$$RBM = \frac{0+0.3+0.2+0.4}{4} = 0.225 \qquad (14)$$

From the result, the more frequently the network changes, the higher the security level within the network, with a reward value equal to 1 representing mitigation effectiveness. Therefore, we should always be aiming for a higher reward value, and this is determined by an organization that wants to deploy shuffle.

6) Method of Data Collection

Relevant data for this research were collected through computer-based simulations designed to evaluate the efficacy of the shuffle-based MTD technique under controlled experimental conditions. The simulations allowed for precise manipulation of network configurations and attacker behaviors to model realistic scenarios. All graphs and visualization outputs were generated using MATLAB R2024a.

IV. RESULTS AND DISCUSSION

A. Introduction

This section presents the results, discussions, and key findings of the research from three (3) different simulations conducted using the configuration of the SDN network shown in Figure 2 (Section 3.1.1). The initial network state was used as the basis for the experimental analysis, which was conducted through simulations using the T-HARM framework. The simulations using developed using HARM codes and run in Python 3.10.

The purpose of the experiments is to demonstrate the effectiveness of both existing and newly proposed security measures. For the simulations, the CVE-2014-5270 vulnerability was used, including its CVSS Base Score. However, while demonstrating the effectiveness of the Shuffle-based MTD technique, three factors are considered (i) Number of Host; (ii) Number of Network States; and (iii) Number of Vulnerabilities. The results are presented as line graphs generated in MATLABR2024a. In the graphs, the horizontal axis represents the progression of time (t), while the vertical axis shows the normalized metric values, which range from 0.0 to 1.0.

B. Results

Figure 5 shows a graphical representation of the result generated by varying the number of hosts using the equations in Section 3.2

Figure 6 displays a graphical representation of the result generated by changing the number of network states using the equations in Section 3.2.

Figure 7 shows a graphical representation of the result generated by varying the number of hosts using the equations in Section 3.2.

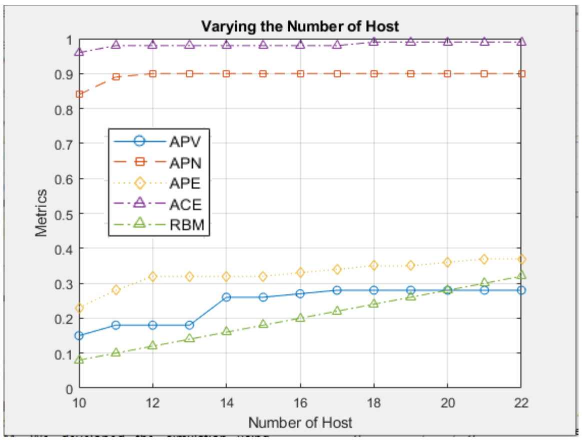

Figure 5. The Result of changing the number of hosts

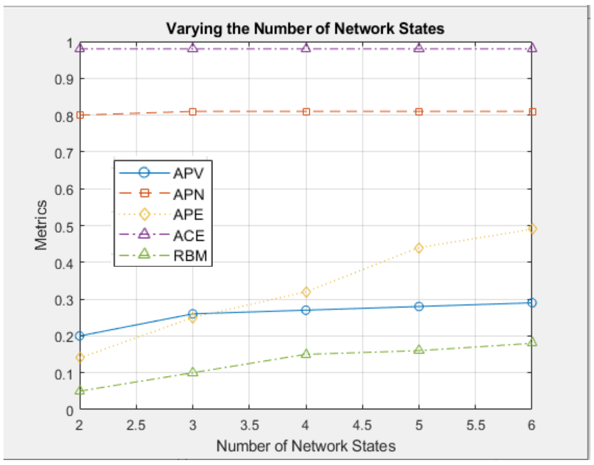

Figure 6. The Result of varying the number of network state

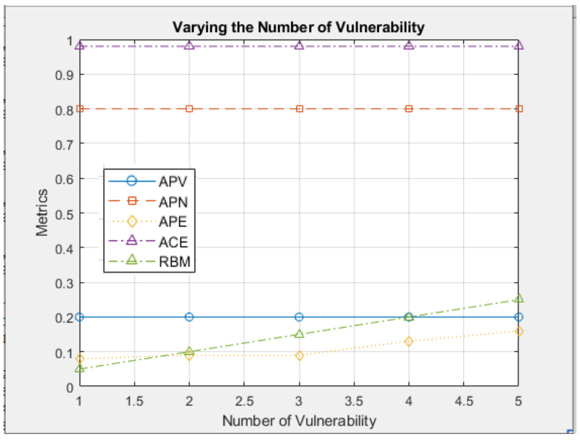

Figure 7. The Result of varying the number of vulnerabilities

C. Discussion

From Figure 5, 13 different experiments were conducted. Each data point utilized five distinct network states. Using the, equations in Section 3.2, different numbers of hosts to capture the changes that occur when shuffle MTD is deployed on the Network. The results show that as the number of hosts increases, both APV and APE metrics increase. This suggests that more hosts lead to more dynamic attach paths, increasing the effort required from attackers. Since the defender is constantly reconfiguring the network, attack paths change frequently, disrupting the attacker's reconnaissance.

The APN metric increases initially and then becomes static, indicating that as more hosts are added to the network, new attack paths emerge until a saturation point is reached. This is consistent with the findings in [12], which says that as more hosts are added to the network, the APN and attack surface are likely to increase and, by implication, decrease the attack effort by providing more options for attacks.

For ACE, the values increase at first and then become static. This suggests that more hosts introduce more opportunities to exploit vulnerabilities, thereby increasing the overall exploitability score. This finding contradicts a statement [12], which says that as the number of hosts grows, the attack cost diminishes.

For the RBM, the effort from the defender's perspective increases as expected because the more shuffle is deployed, the more effort of the defender is reflected with the reward increasing each time there is a deployment.

From Figure 6, five experiments were conducted. The first experiment used two network states to observe changes during a transition. Additional network states were introduced incrementally in subsequent experiments, maintaining the same number of hosts, except for a few changes in the topology. The findings show that as the number of network states increases, APV and APE also increase. This indicates that greater variation in the network leads to an increase in attack efforts in the sense that the attack scenario needs to be altered.

RBM increases accordingly, reflecting the increased effort and proactive behavior of the defender, which indicates the network security level keeps increasing as more network states are added with the shuffle deployment.

APN increases slightly from the first experiment to the second one and then becomes static, which shows that adding more network states introduces a limited number of new paths. This slight increase provides attackers with more choices, potentially reducing their effort.

The value of ACE consistently trends towards one (1), suggesting that more network states provide more opportunities for exploitation, thereby increasing the overall exploitability score to the maximum.

From Figure 7, two (2) network states with ten (10) hosts were maintained, while the number of vulnerabilities was varied from one (1) to five (5). Each experiment introduced one new vulnerability. The result clearly shows that APV, APN, and ACE, remain unaffected by the change in the number of vulnerabilities. This is because the overall structure of the attack paths remains unchanged. APE, however, increases, indicating that more vulnerability extends the time or frequency with which paths are exposed; thus, increasing the attacker's opportunity to exploit them. RBM also increases as the number of vulnerabilities are added. This reflects the additional defensive effort required and the increased reward associated with managing more vulnerabilities through shuffle deployments.

V. RESULT COMPARISON

From the discussion of the results presented in Section 4.3, the metrics APV, APE, and RBM consistently increase across all scenarios. This indicates their reliability in reflecting both attacker and defender efforts. However, APN and ACE show context-dependent behavior, which may sometimes contradict expected attacker effort trends (e.g., more available attack paths may reduce attacker effort rather than increase it). Meanwhile, RBM effectively captures defensive effectiveness and associated costs, especially under shuffle deployment. Figure 8 below shows the comparative results of the security metrics across the experiments.

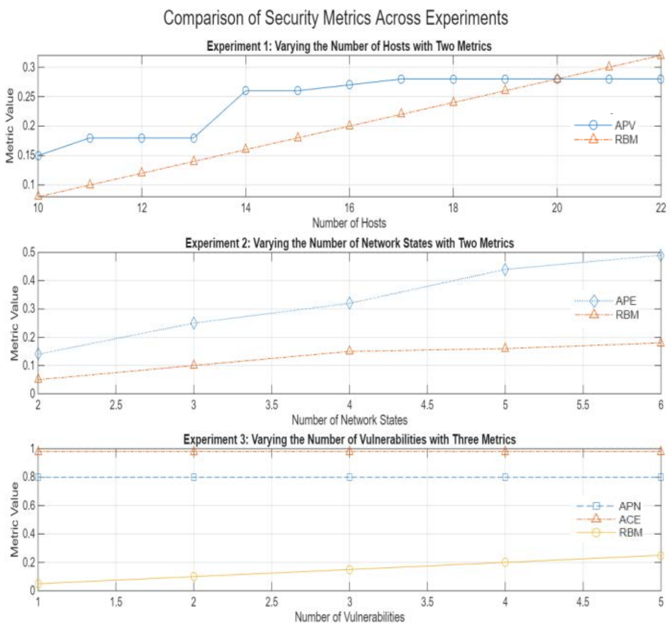

Figure 8. The result comparison of security metrics across the experiments

A. Model Comparison

HARM is a security modeling framework that combines two types of security modeling techniques: Attack Trees and Attack Graphs. The upper layer represents a simplified Attack Graph and uses Attack Trees to model the attacker's objectives, while the lower layer is the T-HARM model and uses an Attack Graph to show how specific vulnerabilities can be exploited. T-HARM has been compared with other model in prior studies [27] [28] [29].

Notably, T-HARM is a hybrid model that is particularly effective in dynamic security environments, where both attack paths and system configurations change over time [30] [31]. In this research, the shuffled-based MTD computation is applied at the upper layer of T-HARM. This dynamic configuration does not change the attack trees in the lower layer. The objective is to continuously modify the attack surface and system configurations (e.g., IP addresses, services, and routing paths), thereby preventing attackers from reliably predicting or maintaining knowledge of the current system state.

VI. CONCLUSIONS

The study provides a comprehensive assessment of the effectiveness of MTD in altering attack surfaces and increasing the complexity of network attacks. By implementing shuffle-based MTD in the T-HARM model and conducting simulations, the research demonstrates that MTD can significantly increase the effort required by attackers, thereby improving overall network security. The findings offer valuable insights for advancing MTD research, the development of standardized security metrics, and the formulation of best practices for robust cybersecurity defense strategies.

The research was limited to implementing and assessing the effectiveness of the shuffle-based MTD approach, with a focus on hosts- and network-level MTD approaches and their impact on both the existing and newly proposed security metrics.

The following suggestions are proposed to improve the existing literature and offer opportunities for future researchers:

i) Currently there is no single metric that can comprehensively assess all security changes introduced by MTD techniques. Future work should focus on developing integrated security metrics capable of evaluating different aspects of MTD-induced changes, which would greatly benefit network defender.
ii) The deployment of security mechanism such as shuffling can affect network availability and performance. Therefore, it is essential to explore methods for balancing MTD deployment with the need to maintain consistent network performance and accessibility.
iii) Future research should integrate multiple MTD techniques within the T-HARM framework to better capture dynamic changes network behavior, and compare the effectiveness of these techniques under different conditions.
iv) There is a need to create a framework capable of measuring critical aspects such as attack surface reduction, system performance, and resilience, providing a more holistic view of MTD effectiveness.
v) Future studies should include cost-benefit analyses to access the economic feasibility of the deploying MTD techniques. This would help decision-makers develop clear, informed adoption strategies.

ACKNOWLEDGEMENTS

All praise is to Almighty Allah (SWT) for allowing me to do this research work.

I am sincerely thankful to my supervisor, Prof. P. B. Zirra, for his invaluable support and contributions throughout this research. His concern, encouragement, and support have been crucial to this paper.

My heartfelt appreciation also goes to Dr. Simon Yusuf Enoch, who had confidence in me during this research. His expertise, particularly in using HARM codes and Python programming for experimental analysis, greatly benefited this work. I thank him for his constant support, patience, and insightful contributions.

Moreover, I am grateful to the NCS Jim Ovia Scholarship program for their financial assistance.

CONFLICT OF INTEREST

The authors state that they have no conflicts of interest concerning the publication of this paper.

AUTHORS CONTRIBUTION

Development and design of the study: A. Hali, Prof. P. B. Zirra; data collection: A. Hali; analysis and interpretation of findings: A. Hali, Prof. P. B. Zirra; Manuscript draft preparation: A. Hali. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

[1] M. Khosravi-Farmad, A. A. Ramaki, and A. G. Bafghi, “Moving target defense against advanced persistent threats for cybersecurity enhancement,” in Proc. of 8th International Conference on Computer and Knowledge Engineering (ICCKE), 2018, pp. 280-285, https://doi.org/10.1109/iccke.2018.8566531

[2] D. Kim, "Moving Target Defense (MTD): Recent Advances and Future Research Challenges," 2022 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Charlotte, NC, USA, 2022, pp. xxxi-xxxi, doi: 10.1109/ISSREW55968.2022.00020.

[3] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati, “A survey of moving target defenses for network security,” IEEE Communication Surveys & Tutorials, vol. 22, no. 3, pp. 1909–1941, 2020, https://doi.org/10.1109/comst.2020.2982955

[4] J. Cho et al., “Toward proactive, adaptive defense: a survey on moving target defense,” IEEE Communications Surveys & Tutorials, vol. 22, no. 1, pp. 709–745, 2020, https://doi.org/10.1109/comst.2019.2963791

[5] R. Sun, Y. Zhu, J. Fei, and X. Chen, “A survey on moving target defense: intelligently affordable, optimized and self-adaptive,” Applied Sciences, vol. 13, no. 9, 2023, https://doi.org/10.3390/app13095367

[6] B. V. Leeuwen, W. Stout, and V. Urias, “MTD assessment framework with cyber-attack modelling,” in Proc. of 2016 IEEE International Carnahan Conference on Security Technology (ICCST), 2016, https://doi.org/10.1109/ccst.2016.7815722

[7] H. Alavizadeh, S. Aref, D. S. Kim, and J. Jang-Jaccard, “Evaluating the security and economic effects of moving target defense techniques on the cloud,” IEEE Transactions on Emerging Topics in Computing, vol. 10, no. 4, pp. 1772-1788, 2022, https://doi.org/10.1109/tetc.2022.3155272

[8] C. Lei, H. Zhang, J. Tan, Y. Zhang, and X. Liu, “Moving target defense techniques: a survey,” Security and Communication Networks, vol. 2018, no. 1, pp. 1–25, 2018, https://doi.org/10.1155/2018/3759626

[9] A. Bajic, “Simulation-based evaluation of dynamic attack and defense in computer networks,” Ph.D dissertation, Freie University Berlin, Germany, 2021.

[10] H. Alavizadeh, J. B. Hong, D. S. Kim, and J. Jang-Jaccard, “Evaluating the effectiveness of shuffle and redundancy MTD techniques in the cloud,” Computers & Security, vol. 102, 2021, https://doi.org/10.1016/j.cose.2020.102091

[11] H. Alavizadeh, H. Alavizadeh, D. S. Kim, J. Jang-Jaccard, and M. N. Torshiz, “An automated security analysis framework and implementation for MTD techniques on cloud,” in International Conference on Information Security and Cryptology - Lecture Notes in Computer Science, 2020, pp. 150–164, https://doi.org/10.1007/978-3-030-40921-0_9

[12] J. B. Hong, S. Y. Enoch, D. S. Kim, A. Nhlabatsi, N. Fetais, and K. M. Khan, "Dynamic security metrics for measuring the effectiveness of moving target defense techniques," Computers & Security, vol. 79, pp. 33–52, 2018, https://doi.org/10.1016/j.cose.2018.08.003

[13] X. Xiong, L. Yang, and G. Zhao, "Effectiveness evaluation model of moving target defense based on system attack surface," IEEE Access, vol. 7, pp. 9998–10014, 2019, https://doi.org/10.1109/access.2019.2891613

[14] S. Y. Enoch, J. B. Hong, M. Ge, and D. S. Kim, "Composite metrics for network security analysis," Cryptography and Security, 2020, https://doi.org/10.48550/arXiv.2007.03486

[15] M. F. Hyder, M. U. Farooq, U. Ahmed, and W. Raza, "Towards enhancing the endpoint security using moving target defense (shuffle-based approach) in software defined networking," Engineering, Technology & Applied Science Research, vol. 11, no. 4, pp. 7483-7488, 2021.

[16] A. Brown, T. W. Lee, and J. B. Hong, "Evaluating moving target defenses against realistic attack scenarios," in Proc. of 2023 IEEE/ACM 4th International Workshop on Engineering and Cybersecurity of Critical Systems (EnCyCriS), 2023, pp. 1-8.

[17] X. Xu, H. Hu, Y. Liu, H. Zhang, and D. Chang, "An adaptive IP hopping approach for moving target defense using a light-weight CNN detector," Security and Communication Networks, vol. 2021, no. 1, 2021, https://doi.org/10.1155/2021/8848473

[18] G. Ballot, V. Malvone, J. Leneutre, and E. Borde, "Reasoning about moving target defense in attack modeling formalisms," in Proc. of the 9th ACM Workshop on Moving Target Defense, 2022, pp. 55-65.

[19] B. M. Amro, S. Salah, and M. Moreb, "A comprehensive architectural framework of moving target defenses against DDoS attacks," Journal of Cyber Security and Mobility, pp. 605-628, 2023.

[20] D. Reti, D. Fraunholz, K. Elzer, D. Schneider, and H. D. Schotten, "Evaluating deception and moving target defense with network attack simulation," in Proc. of the 9th ACM Workshop on Moving Target Defense, 2022, pp. 45-53.

[21] S. E. Yusuf, M. Ge, J. B. Hong, H. K. Kim, P. Kim, and D. S. Kim, "Security modelling and analysis of dynamic enterprise networks," in Proc. of 2016 IEEE International Conference on Computer and Information Technology (CIT), 2016, pp. 249-256.

[22] I. Pekaric et al., "A systematic review on security and safety of self-adaptive systems," Journal of Systems and Software, vol. 203, 2023, https://doi.org/10.1016/j.jss.2023.111716

[23] S. Y. Enoch, "Dynamic cybersecurity modelling and analysis," Ph.D dissertation, University of Canterbury, New Zealand, 2018.

[24] J. B. Hong, S. Yoon, H. Lim and D. S. Kim, "Optimal network reconfiguration for software defined networks using shuffle-based online MTD," in Proc. of 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS), 2017, pp. 234-243, https://doi.org/10.1109/SRDS.2017.32

[25] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "Openflow random host mutation: transparent moving target defense using software defined networking," in Proc. of the 1st Workshop on Hot Topics in Software Defined Networks (HotSDN '12), 2012, pp. 127-132, https://doi.org/10.1145/2342441.2342467

[26] D. Evans, A. Nguyen-Tuong, and J. Knight, "Effectiveness of moving target defenses," in Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats, New York: Springer, 2011, pp. 29-48.

[27] J. B. Hong, and D. S. Kim, "HARMS: Hierarchical attack representation models for network security analysis," in Proc. of 10th Australian Information Security Management Conference, 2012, pp. 74-81.

[28] J. B. Hong, and D. S. Kim, "Scalable security analysis in hierarchical attack representation model using centrality measures," in Proc. of 2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W), 2013, pp. 1-8.

[29] K. Sowka, V. Palade, X. Jiang, and H. Jadidbonab, "Towards the generation of hierarchical attack models from cybersecurity vulnerabilities using language models," Applied Soft Computing, vol. 171, 2025.

[30] S. Y. Enoch, M. Ge, J. B. Hong, and D. Seong Kim, "Model-based cybersecurity analysis: past work and future directions," in Proc. of 2021 Annual Reliability and Maintainability Symposium (RAMS), 2021, pp. 1-7, https://doi.org/10.1109/RAMS48097.2021.9605784

[31] K. Zenitani, "Attack graph analysis: an explanatory guide," Computers & Security, vol. 126, 2023, https://doi.org/10.1016/j.cose.2022.103081