Evaluating Performance and Security of a Hybrid Moving Target Defense in SDN Environments

Minjune Kim[1], Jin-Hee Cho[2], Hyuk Lim[3], Terrence J. Moore[4], Frederica F. Nelson[4], Ryan K. L. Ko[1], and Dan Dongseong Kim[1]

[1]The University of Queensland, Australia
[2]Virginia Tech, USA
[3]Korea Institute of Energy Technology (KENTECH), Republic of Korea
[4]DEVCOM Army Research Lab., USA
mj.kim@uq.edu.au, jicho@vt.edu, hlim@kentech.ac.kr,
{terrence.j.moore.civ, frederica.f.nelson.civ}@army.mil, {ryan.ko, dan.kim}@uq.edu.au

***Abstract*—As cyberattacks are rising, Moving Target Defense (MTD) can be a countermeasure to proactively protect a networked system against cyber-attacks. Despite the fact that MTD systems demonstrate security effectiveness against the reconnaissance of Cyber Kill Chain (CKC), a time-based MTD has a limitation when it comes to protecting a system against the next phases of CKC. In this work, we propose a novel hybrid MTD technique, its implementation and evaluation. Our hybrid MTD system is designed on a real SDN testbed and it uses an intrusion detection system (IDS) to provide an additional MTD triggering condition. This in itself presents an extra layer of system protection. Our hybrid MTD technique can enhance security in the response to multi-phased cyber-attacks. The use of the reactive MTD triggering from intrusion detection alert shows that it is effective to thwart the further phase of detected cyber-attacks. We also investigate the performance degradation due to more frequent MTD triggers.**

This work contributes to (1) proposing an ML-based rule classification model for predicting identified attacks which helps a decision-making process for security enhancement; (2) developing a hybrid-based MTD integrated with a Network Intrusion Detection System (NIDS) with the consideration of performance and security; and (3) assessment of the performance degradation and security effectiveness against potential real attacks (i.e., scanning, dictionary, and SQL injection attack) in a physical testbed.

***Keywords*—Intrusion Detection, Moving Target Defense, Performance, Rule classification, SDN**

I. Introduction

Cyber-attacks are becoming more sophisticated and protecting systems with traditional defense mechanisms remains a challenge. Attackers have asymmetric advantages in terms of time and resources and conventional defenses may provide security controls in order to mitigate ongoing attacks after detection. The reactive responses in defense may be able to cause significant financial and systemic losses as a result of attacks.

In comparison to static defense mechanisms, Moving Target Defenses (MTDs) are symmetric defense mechanisms that increase the complexity of a system, so that attackers can only obtain partial knowledge about targeting systems, which impedes the attacker's efforts to gain total access to the system. In order to compromise targets, adversaries expend higher attack costs to discover a target or its vulnerabilities. As many previous MTD-related works [1], [2] have shown, security benefits of MTD adoption are evident compared to conventional defense services (e.g., firewalls or IDS) because dynamic changes in system configuration lead to the rise of uncertainty for adversaries to use their resources or time in order to compromise their targets.

As one of the MTD techniques, various network address shuffling techniques have been proposed to enhance security benefits against the reconnaissance phase of an attack in the Cyber Kill Chain (CKC) [3]. A virtual IP (vIP) shuffling technique called Flexible Random Virtual IP Multiplexing (FRVM) [4] was proposed to increase the complexity of network address in a Software-Defined Network (SDN) environment. Their work was evaluated on a theoretical basis using the metric of attack success probability that showed the success probability of discovering virtual IPs of targets would be low when the size of the network space is considerably larger than a network space that can be scanned by an attacker. Most previous work on MTD techniques proposed the use of a single MTD technique on a regular basis and support the assessment of either performance or security, with consideration that a system cannot be secured without compromising performance. Using a single timeliness-based MTD is not sufficient to protect an SDN-based system from the security perspective [5].

Therefore, this work first proposes a Machine Learning (ML)-based model for rule classifications that can be used to identify attack types from malicious traffic in a realistic system. The proposed model aims to predict the types of attacks involved in miscellaneous groups and then is used to apply adjustable MTD configuration (i.e., MTD actions) against potential cyber-attacks. A few previous work [6], [7] proposed some methods to identify attack types or patterns extracted from a set of existing rules of an Intrusion Detection System (IDS). The objectives of their work were to cluster attack attributions into groups characterized by alert messages. However, their work had limitations to proposing and evaluating the models only in a theoretical manner. In comparison to this work, our work proposes not only an ML-based rule classification method but also applies the proposed model to

2693-9177/22/$31.00 ©2022 IEEE
DOI 10.1109/QRS57517.2022.00037

our physical system in order to identify the types of ongoing attacks based on alerts involved in miscellaneous (MISC) groups. Our proposed model can help to decide appropriate MTD parameters against an identified ongoing attack, which leads to an increase in security effectiveness in the response to events.

Second, we propose a realistic hybrid MTD that can be integrated with a Network Intrusion Detection System (NIDS). We then apply the MTD mechanism with a feasible web service into a physical SDN-based network. The combination of an event-based and time-based defense mechanism (i.e., hybrid MTD) aims to enhance security benefits against current ongoing attacks. An event-driven solution in our hybrid system can reactively respond to the event created by malicious payloads in order to reduce risks based on alerts, whereas a time-based service can proactively protect a system against any further known and unknown attacks. Among the configurations in a time-based mechanism, an MTD interval of 300 s is determined for the purpose of minimizing performance overhead (e.g., throughput) when compared with conventional defense techniques [8].

Third, we evaluate the performance overhead as well as the security benefits of a system adopting a hybrid MTD technique in a physical SDN-based testbed. Most previous work assessed either performance or security in their analytic, simulation, or emulation models. Only one study evaluated both security and performance in a system: Dishington *et al.* [9] under a Mininet emulator with a few metrics only. However, the previous work had limitations with regard to evaluating MTD adoption in practice, and there are few studies that examined the performance and security of MTD systems in a physical environment. In comparison to these work, we propose to assess the performance and security requirements of a system adopting a hybrid MTD in a real SDN-based testbed. In this evaluation, one of goals will be to show implications and applicability of the hybrid MTD with security and performance by adjusting MTD operations.

Our key contributions are summarized as follows:

- We develop an ML-based rule classification model and apply the proposed model into a real system for identifying attack types;
- We develop a hybrid MTD model with an NIDS (i.e., Snort) in a realistic network system that can decide adjustable MTD actions in the decision-making;
- We evaluate both performance degradation and security advantages of a hybrid MTD system in a physical SDN-based environment.

The rest of this paper is structured as follows. Section II describes the related work regarding MTDs and rule classification/clustering. Section III describes our proposed approach. Section IV presents rule classification for attack identification. Section V describes the design and implementation of our proposed hybrid MTD system. In Section VI presents the experimental results and our discussion. Lastly, we conclude this paper in Section VII.

II. Related Work

Moving Target Defense: As a proactive defense mechanism, an MTD protects a surrounding system by dynamically changing attack surface. Many MTD-related solutions (e.g., network shuffling, software diversity) in various layers [10]–[19] have been proposed to demonstrate their security advantages compared to conventional defense systems. Hong and Kim [20] categorized ongoing MTDs into three techniques such as shuffle, diversity, and redundancy. Among them, shuffling techniques in networks represented dependable benefits against attacks in the reconnaissance phase of the CKC, which leads to the rise of complexity or uncertainty of network configurations. Jafarian *et al.* [21], Antonatos *et al.* [22], and Sharma *et al.* [4] proposed network shuffling techniques that randomly changed network addresses. In their theoretical and empirical research, they found that dynamic changes in network configurations increased an attacker's workload, requiring them to extend more time and effort in exploiting or penetrating the targeted system only during a given period of time.

Identification of Attacks: Using a collection of rules published by the Snort community, Turner and Joseph [6] proposed a rule clustering approach to characterize common attributions based on network protocols (e.g., ICMP). Scarabeo *et al.* [7] identified attack patterns extracted from IDS rules that could be mapped into a defense strategy. They found an optimal security mechanism against multiple attacks heading to a target by computing attack costs [23]. Since there are few studies that deal with rule classifications in order to identify potential attack types, our work proposes appropriate defense algorithms that are based on the attack cost of specific attack types.

Hybrid-MTD: As cyber-attacks grew more complicated and intelligent, a single defense system was not sufficient so the combination of multi-defense mechanisms were proposed to help systems to enhance security benefits as reactively detecting and proactively preventing adversaries. There were some discussions of integrating an MTD system with other defense countermeasures in order to increase system security. Cho and Ben-Asher [24] introduced a Stochastic Petri Nets model of an integrated IDS with various defense solutions (e.g., MTD or honeypot) for assessing the performance and security. They described that an IDS with MTD could provide better security benefits than the conventional defense system (i.e., IDS only). Zhuang *et al.* [25] proposed an event-driven MTD technique by using adaptive intervals and evaluated the model in a simulation-based environment. There are limitations in these studies as they showed the effectiveness of security only in simulation-based models, and some gaps of experimental results in terms of performance and security could be found between simulation-based and practical models.

Performance and Security Evaluation: There are several studies using analytic models, simulation models, or emulation models that aimed to evaluate either security or performance of a system when a MTD technique is adopted [4], [8], [19], [25]–[28]. Mendonça *et al.* [8] developed a Deterministic and

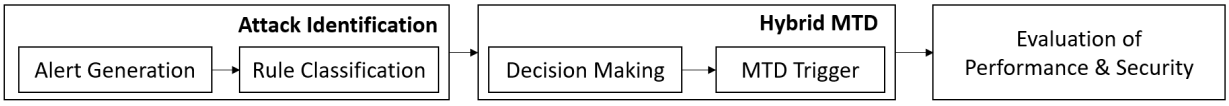


Figure 1: Our Proposed Approach

Stochastic Petri Nets (DSPNs) model to evaluate performability of a time-based MTD adoption. Connell *et al.* [26] proposed a Continuous Time Markov Chains (CTMC)-based model to measure performance of a system and to find an optimal reconfiguration rate. Sharma *et al.* [4] evaluated security benefits of a virtual IP shuffling technique in an analytic manner. These works showed to evaluate either performance or security in a system compared to static defense systems when a time-based or event-driven MTD algorithms were adopted. There are few related works that take into account security and performance in MTD systems and only Dishington *et al.* [9] proposed to assess both requirements by using a few metrics. However, experiments in this work were also carried out under constrained environments of an emulation called Mininet.

III. Proposed Approach

The figure 1 presents the overview of our proposed approach about how to design and evaluate a hybrid MTD system in terms of performance and security. We first propose attack identification and a hybrid MTD system based on alerts. Then we propose the methodology to evaluate performance degradation and security effectiveness.

A. Attack Identification

1) Alert Generation: Alert generation is to raise alerts to an administrator when any malicious traffic/behaviors are detected in a network/system. An IDS provides a monitoring service that can detect any suspicious traffic or malicious activities across a network and then alert to an administrator/system controller. When it comes to the intrusion detection, there are two methods that are common, anomaly-based or signature-based approaches. In contrast to anomaly-based IDSs, signature-based IDSs consist of an internal database that is informed by previously studied attacks. Therefore, alerts generated by signature-based IDSs represent attributions (e.g., attack patterns or severity) about the current ongoing attacks. In response to alert information, a system administrator can decide security controls that take action in order to stop ongoing attacks and to prevent future attacks. The ability to detect malicious behaviors in our alert generation is dependent on the set of rules that is used to monitor the potential attacks on a network or system and their accuracy. In order to classify and categorize real attacks on the basis of characteristics, a collection of practical rules should be followed.

2) Rule Classification: A rule classification is a method for categorizing rules involved in miscellaneous groups into identified classifications. In a signature-based IDS, the message field of rule options contains information or patterns about malicious traffic being monitored. Upon receiving alert messages, a defender may take an action against identified attacks by adopting an appropriate countermeasure and then mitigating those attacks. The alert messages in a set of rules should be well categorized in order to help identify a type of attack. There are, however, some ambiguous rules that may lead to the wrong decision being taken. As a result of improper defense policies, current ongoing attacks may not be thwarted or future attacks will not be prevented. Therefore, our rule classification model is proposed to predict miscellaneous rules for the correct decision in the decision-making process. Our model is trained and validated from well-classified practical rules as a collection of labeled data for ground truth and its performance is also assessed by using 10-fold cross-validation method.

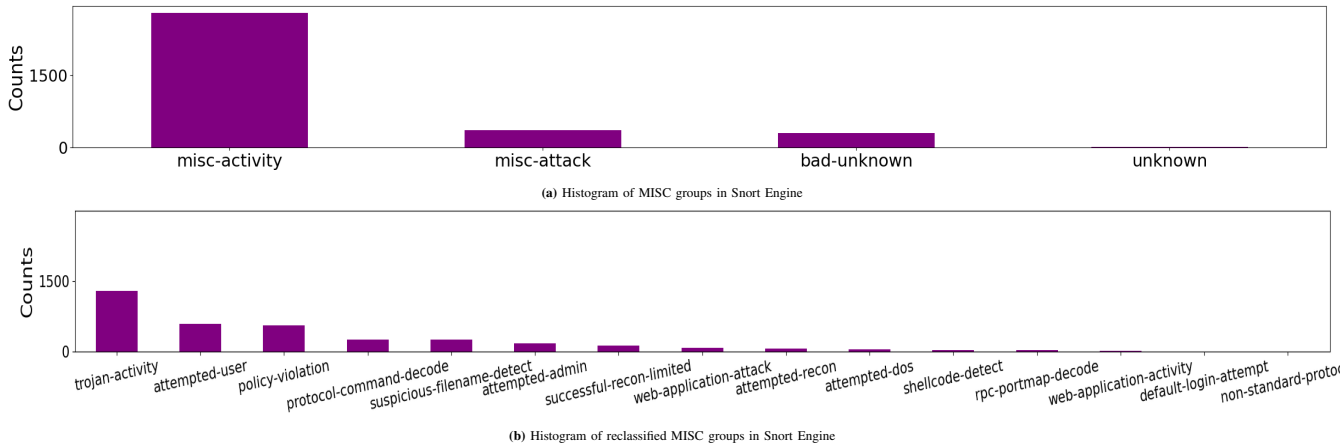
B. Hybrid MTD

1) Decision-Making: It is the process of gathering information from an alert message, identifying the type of attack, and deciding MTD policies in a given system that constitutes decision-making. This process aims to provide a system with a defense solution that has appropriate operation policies that are capable of mitigating current attacks in progress. The algorithm implements several steps to determine what operations or policies should be carried out on the system based on the relevant information in the received alert message(s). The function of decision-making has an internal database of preliminary information to assist with what options can be applied into a system when types of attacks can be detected.

2) MTD Trigger: A hybrid MTD can be a countermeasure against attack traffic that combines time-based and event-based methods. It is possible to trigger changes in an attack surface periodically, in response to events, or both. Our proposed MTD trigger policies can be determined based on the knowledge of alert information or on a regular basis, and an MTD controller determines when changes are made to the system configuration of a system (when to move). A hybrid MTD is capable of securing a system in the event that malicious behavior is detected. The proposed hybrid MTD utilizes adjustable MTD trigger mechanisms to reactively respond to identified attacks while taking both performance and security into considerations.

C. Evaluation

The objectives of this work include the proposal of a hybrid MTD system and the measurement of performance overhead as well as security benefits when the hybrid MTD is deployed in a system. As a state-of-art defense mechanism, many previous studies have already demonstrated the security benefits of MTD techniques, however evaluating both performance and



(a) Histogram of MISC groups in Snort Engine

(b) Histogram of reclassified MISC groups in Snort Engine

Figure 2: k-NN supervised machine learning of classifications in Snort rules. 4 MISC groups at (a) can be predicted into a set of more identified classifications at (b)

security is rare and is a necessary consideration to take into account the pros and cons of MTD adoptions. The performance and security metrics are proposed for quantitative analysis and evaluated by using benign and attack traffic across a system. The benign or attack workloads generate an amount of network packets that lead to an increase of overhead in a network or server(s). We model the benign and attack workloads to follow the Poisson distribution with a request rate and an attack rate, respectively. Measured metrics in a physical testbed are used to compare with baseline of systems under the constraints of system environments and selected attack scenarios.

IV. Attack Identification

In this section, we describe the detail of our proposed attack identification. First, we present the design of our alert generation mechanism. Then, we present our ML-based rule classification and show the accuracy of the proposed model including cross-validation.

A. Alert Generation

As part of the system model, we deploy a signature-based Network Intrusion Detection System (NIDS) called Snort [29]. Snort is one of the NIDS open software solutions in practice and provides a collection of practical rules for the generation of alerts. Our detection system is based on a set of rules with the version of '29171_210630', that is published by the Sourcefire Vulnerability Research Team (VRT) [29]. Snort in the promiscuous mode on a machine can monitor and analyze all network packets passing across a network and then raise alerts when any suspicious network activities are detected. The alerts from a Snort engine includes the information about malicious traffic such as information of source and destination, priority, alert messages, and so on. Our alert generation also manages an amount of duplicated alerts due to repeatable attack traffic or invalid traffic (e.g., invalid IP/port) that can be ignored and then sends only valid alerts to the decision-maker.

B. Rule Classification

We utilize a supervised machine learning algorithm called a k-Nearest Neighbors (k-NN) in order to create a taxonomy of miscellaneous and ambiguous classifications. The message option in rules are vectorized by using word2vec [30] and then our trained model is used to reclassify a set of miscellaneous rules into one of the classifications that is well-categorized.

1) Training: The official Snort rules are classified into 38 default classifications and four of them (i.e., unknown, bad-unknown, misc-activity, and misc-attack) excluding 'deleted' have not been well-classified. We characterize the message option in a set of Snort rules and cluster it into 33 default classifications excluding the miscellaneous groups and 'deleted'. Some of these rules (36030 samples) are used for training our model, considered as ground truth, and the text-based messages are vectorized by the Natural Language Processing (NLP)-based technique called word2vec. The word2vec algorithm learns word associations from a context of alert messages and produces a representative vector space. The NLP model uses the architecture of continuous bag-of-words (CBOW) because the order in the messages does not affect the prediction of rules and the vector size of 100.

TABLE I: Top 5 of classifications and their accuracy of our ML-based model in Engine

Classification	Precision	Recall	F1-Score
attempted-user	0.86	0.91	0.89
trojan-activity	0.94	0.96	0.95
attempted-admin	0.75	0.71	0.73
protocol-command-decode	0.98	0.91	0.94
web-application-attack	0.74	0.79	0.77

2) Validation: We train and validate our rule classification model by splitting our data set into 80:20 ratio, that is 36030 (8/10 of samples) for training and 9008 (2/10 of samples) for testing are being used. Each validation score from rules in 33 default classifications are assessed by the accuracy such as precision, recall and F1-score and some of them can be seen in Table I. The weighted average of our training model are also calculated into precision of 0.84, recall of 0.85, and F1 score of 0.84. In addition to the validation of our model, we also conduct k-fold cross validation by splitting the training data into 10 folds ($k = 10$). We use 9 folds for training and

1 fold for the testing and cycles through so each fold is used for testing once. The average of cross-validation results goes to 0.8254, whereas its minimum and maximum are computed into 0.8176 and 0.8313 respectively.

3) Prediction: Classifications of rules for miscellaneous groups are predicted from our trained model and their new predicted classifications are used to replace their prior ambiguous label in the Snort alerts. As can be seen in Figure 2, the histogram of the predicted classification represents an amount of 3467 rules involved in ambiguous groups assigned by the engine and they are reclassified into the other classifications. As a result of our rule classification model, we can predict that rules involved in miscellaneous groups are clustered into 'trogan-activitiy' at the highest reclassification total, followed by 'attempted-user' in second, whereas the majority of training samples for the model is 'attempted-user' and 'trogan-activity', respectively. The prediction is operating when any alerts involved in miscellaneous groups are raised and the MTD controller will use new classified information for decision making in order to apply appropriate defense operations against multi-phase attacks.

V. HYBRID MTD

A hybrid MTD combines time-based and event-driven MTD triggering mechanisms, resulting in proactive and reactive responses to attacks. In this work, we propose an integrated MTD system with a signature-based NIDS (i.e., Snort) as a hybrid MTD mechanism in a physical network environment. An NIDS in alert generation raises alerts including alert classifications and a decision-making process in a hybrid MTD decides MTD actions in the response to alerts. Decision making in this work is constructed to identify the types of attacks and then take MTD actions.

A. Decision Making

Our proposed decision makers (locally for the NIDSs and centrally for the MTD controller) play a role of analyzing the generated alert messages and identifying their potential threats across a system. A local decision maker aims to identify characteristics of malicious traffic and reducing bulky duplicated alert messages from malicious packets. A central decision maker aims to verify alert messages that have potential threats against the system and makes a decision of MTD action(s) such a immediately triggering the mechanism or shortening the time-based MTD interval. From predicting an attack type, we may estimate the attack cost of the ongoing attack. The attack time of the identified attack becomes a key feature to decide MTD action(s) on our hybrid MTD system because it can determine when a vIP shuffling technique is operating in order to change an assigned vIP address across a network or to disconnect any ongoing connections between a web server and an attacker.

1) MTD Actions: We propose three potential MTD actions that are applicable to a vIP shuffling technique against cyber-attacks. The MTD actions determine when an MTD is operating and how it adjusts current configuration. Their functions are as follows:

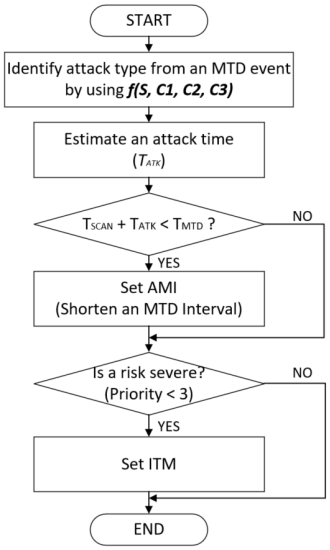


Figure 3: Decision making process for taking MTD action(s)

- *Adjust an MTD Interval (AMI):* Decision making can decrease an MTD interval by a scaling factor (e.g., half) for a time-based MTD when any potential threats are detected across the SDN network, whereas it increases the interval back to the previous value if any further threats are not detected during a given time (e.g., current MTD interval $\times$ 2).
- *Immediately Trigger an MTD (ITM):* Decision making triggers an MTD operation as soon as the NIDS detects any potential threats. It expects to intentionally disconnect any connections between an attacker and a targeting host so that it can hide a new vIP address from an attacker.
- *Both AMI and ITM:* In order to maximize the security effect, decision making can perform the combination of AMI and ITM together when any threats are discovered. This will help not only reduce the time that a vIP is exposed from a potential attacker, but also disconnect a connection between an attacker and a targeting host.

2) Decision-Making Process: The process is used to determine one of the MTD actions based on the estimated attack time associated with an identified attack type. Within the constraints of the experiments, we measure the attack time of the selected attack scenarios in a physical SDN testbed and then the data is stored in the internal database of the decision-maker. Figure 3 describes the decision-making process that takes an MTD action using alert information derived from detected malicious traffic. The attack type can be identified by:

$$Attack\ Type = f(S, C1, C2, C3),$$

where S is the severity (the inverse of priority in Snort), $C1$ represents one of the 67 category options in Snort, $C2$ denotes one of the 33 default classification in Snort and $C3$ is

a classification predicted by our ML-based rule classification model. The severity option refers to the level of each specific vulnerability as measured by industry standard vulnerability metrics such as the Common Vulnerability Scoring System (CVSS). Table II presents the examples of identified attack type based on criteria and Table III shows their measured attack time on a physical SDN-based testbed.

TABLE II: Examples of criteria in the decision-making process against three types of potential attacks

Attack Type	S	C1	C2	C3
Scanning	1/2	-	attempted-recon	-
Dictionary	1/3	-	unknown	attempted-recon
SQLi	1	sql	web-application-attack	-

TABLE III: Attack time measured in an SDN testbed. This empirical attack cost is used for the condition of taking an MTD action

Attack Type	Success Time [s]	T_{SCAN} + T_{ATK}
Scanning	145.7287	145.7287
Dictionary	68.4155	214.1442
SQLi	116.6489	262.3776

B. MTD Trigger Mechanism

An MTD is triggered on a regular basis if no ongoing attacks are detected, whereas an event-based MTD is triggered when an alert is raised by an NIDS after detecting any malicious traffic across a network.

1) Time-based Trigger: As a proactive defense mechanism, a time-based MTD is applied to periodically alter a vIP address of a host that is temporarily valid during a certain period of time (i.e., an MTD interval). A legitimate client can resolve a vIP address of a server from a DNS service, whereas an attack cannot access it. An MTD mechanism running on an Open Network Operating System (ONOS) as an SDN platform updates flow tables of connected SDN switches and manages network traffic and paths based on a current matrix of a real IP address and a virtual IP address. From a security point of view, a shorter MTD interval provides better security benefits. However, it accordingly leads to higher performance degradation such as increasing the number of abrupt disconnections between a client and legitimate requests [1]. In this work, a time-based MTD system runs using a default MTD interval (T_{MTD}) of 300 s, so that a virtual IP address of a connected host is changed every T_{MTD}. The MTD interval is adjustable so that the interval is halved when an attack is detected by an NIDS or doubled when no additional attacks are detected over a given time frame (e.g., twice the current MTD interval). To ensure performance and security, the MTD interval should be determined at a minimum of 60 s and a maximum of 300 s.

2) Event-based Trigger: As a reactive defense mechanism, an event-based MTD is applied to alter a vIP address of a host in response to events such as alerts by an NIDS. When our NIDS detects any malicious activities in a system, its intrusion alerts are sent to the MTD controller in order to change the vIP assigned to a host. We assume that the NIDS and the MTD controller are securely connected in a distinct network separated from the internet so that an attacker cannot access it. This event-driven MTD operation is designed to reactively thwart attackers' acts towards a targeted server.

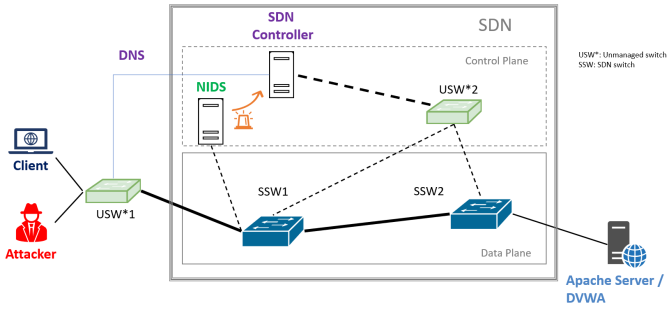


Figure 4: Network Architecture of Our Physical Testbed

TABLE IV: Key design parameters and their default values

Type	Variable	Description	Value
Inputs	λ	HTTP Request rate per 1 min.	20 [#/min.]
	ATK_{λ}	Attack rate per 1 min.	0.6 [#/min.]
System Param.	T_{MTD}	MTD interval	300 [s]
	Web_s	Size of a static web page	2 [Mbytes]
	N	Number of possible vIP network addresses	4096
	M	Possible number of vIP scanned by an attacker	4096
	BW	Bandwidth of inter-switch links	10 [Gbps]
	WKR_{max}	Max number of workers	400

C. Experimental Setup and Evaluation

1) System: We propose a hybrid MTD system that works on the combination of a time-based and an event-based MTD in a physical SDN-based testbed. The SDN testbed consists of a MTD controller, SDN-based switches (Aruba 2920F), practical web servers, a vulnerable server, and a legitimate client. With the purpose of enhancement of security, the MTD system is interacting with Snort version 2.9.17 in order to detect malicious behaviors by an attacker across a network. In addition to the hybrid MTD system, a web server is running an Apache web service for the performance evaluation as well as a Damn Vulnerable Web Application (DVWA) for the security evaluation on it as can be seen in Figure 4. The MTD system updates a newly assigned vIP address of a web server to a DNS server.

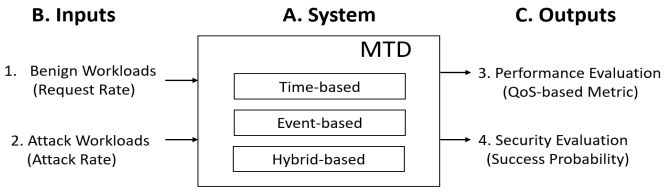


Figure 5: Evaluation Framework

2) Inputs: Benign and attack workloads generate an amount of network packets based on their request rates as an input parameter that obey a Poisson distribution. Benign samples send web-based requests to a server that can be considered baseline traffic. The baseline traffic requests increase workloads on a web server or a network, that can be used to measure performance. Probabilistic attack workloads also generate malicious traffic across a network that aims to discover information of a target and to compromise it.

- Benign Workloads: Probabilistic benign workloads generate HTTP requests with a request rate (λ) for measuring performance overhead in a system due to MTD adoptions. An Apache benchmark tool [31] is used as a legitimate process to send requests by the HTTP GET method to a server and the server is delivering 2 MBytes of static web data over a connected network. The requested web data is delivered by an amount of segmented network packets and they lead to the increase of workloads on a network or a web server. To measure the system performance, we are running an Apache web server that is one of the popular web service in practical systems and using a fixed request rate (λ) of 20 [#/min.] along with realistic attacks such as scanning, dictionary or SQL Injection (SQLi) attacks.
- Attack Workloads: The attack traffic can be used for evaluating the security impact on a system due to our proposed defense system. Using classic but practical penetration tools in Kali Linux, we are generating realistic and probabilistic attack workloads for evaluating security in an SDN network adopting a vIP shuffling. The attack workloads follow a Poisson distribution with an attack attempt rate of 0.6 [#/min.] and the total attempts in each attack scenario were homogeneously conducted up to 100. We assume that adversaries have knowledge of the network space in service (e.g., 10.0.0.0/20) but they are always required to discover a vIP assigned to a target during the first phase of the CKC because a host's vIP address keeps being changed. We evaluate three attack scenarios in this work and their objectives are as follows: (i) a detectable XMAS Scanning attack aims to discover a vIP address of a target in phase 1; (ii) Either a dictionary or a SQLi attack with undetectable scanning in phase 2 aims to crack a user's credential by using 'Patator' or 'Sqlmap' respectively. With the purpose of supporting the assessment of phase 1 & 2 of the CKC [3], two different options of scanning (-sX or -sT) are being used. dictionary and SQLi attacks are selected because they all are one of the most critical security threats towards practical web servers and ranked in the Open Web Application Security Project (OWASP) top 10 2021 [32].

3) Outputs: The outputs are performance and security metrics that are used for supporting the assessment of performance degradation and security effectiveness in a system adopting MTD mechanisms. With the comparison to a baseline, experimental results are collected and analyzed against a series of attack scenarios for quantitative analysis.

- Performance Evaluation: Adopting a vIP shuffling technique accordingly leads to an amount of abrupt disconnections between a web server and legitimate requests due to time-based or event-driven network reconfiguration. In this work, a Quality of Service (QoS)-based performance metric called 'Fraction of Failed Jobs (FFJ)' is used and can be computed by:

$$FFJ = \frac{\#\ of\ failed\ jobs}{\#\ of\ total\ HTTP\ GET\ requests},$$

  where failed jobs denote HTTP requests that do not complete and do not receive the requested web data. A job is considered a success when all requested data is completely delivered to a client, otherwise it is considered a failure.
- Security Evaluation: To compare security benefits between MTD systems, we compute the success probability representing if an attacker meets its objective toward a target. The security metric called 'Attack Success Probability (ASP)' can be computed by:

$$ASP = \frac{\#\ of\ attack\ successes}{\#\ of\ total\ attack\ attempts},$$

  where attack success represents if an attacker can meet its goal. At a scanning attack, an attack is considered a success if an attacker can discover a vIP in Phase 1. A dictionary attack or an SQLi attack is considered a success when an attacker can discover a vIP address of a target in the first phase and then crack a user's credential in the second phase.

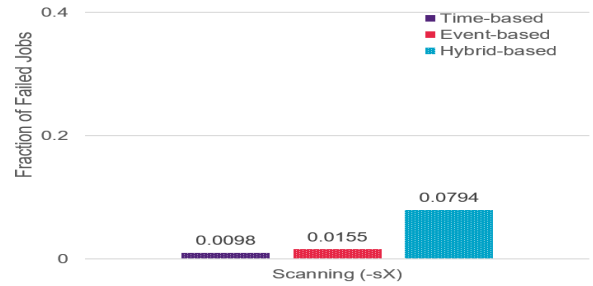
VI. Evaluation of Performance and Security

This section presents the experimental results to assess performance and security in our physical SDN-based testbed. We first show performance degradation in a system adopting three MTD systems during three types of attack scenarios. Then, we present security advantages of a hybrid MTD compared with other MTD systems. Next, we analyze advantages and disadvantages of MTD adoption in a system based on our experimental results. Lastly, we discuss the limitations of this work.

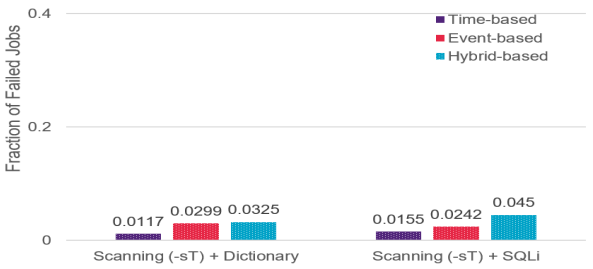
A. Performance

Figure 6 presents the performance metric of FFJ that measures performance degradation caused by abrupt disconnections during one of the attacks. In overall, it shows that the event-based MTD as well as a hybrid MTD results in more performance degradation than a time-based MTD system. As the shorter attack time leads to a higher number of vIP shuffles in order to defend against an attack, it accordingly causes more abrupt disconnections between a connected legitimate client and a server.

At Figure 6(a), FFJ of the event-driven MTD almost doubles that of a time-based MTD against a detectable scanning attack, whereas 0.0794 of a hybrid MTD is the highest. It is because a scanning attack continuously attempts to scan the

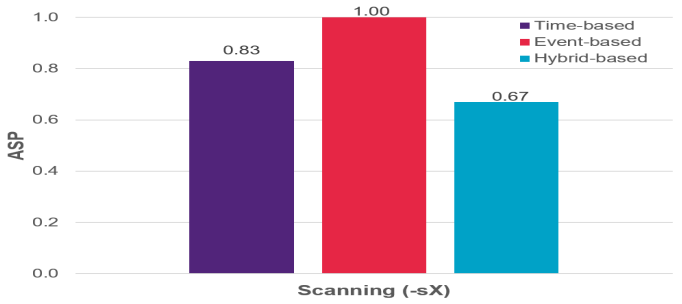


(a) Fraction of Jobs (FFJ) when phase 1 is detected by NIDS

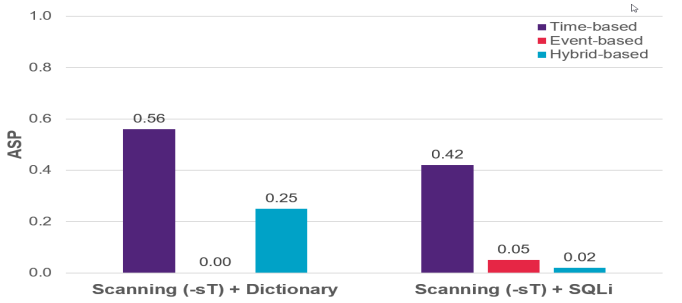


(b) Fraction of Failed Jobs (FFJ) when phase 1 is not detected and only phase 2 is detected by the NIDS.

Figure 6: Performance degradation when the hybrid MTD is applied



(a) It represents FFJ when the 1st phase of attacks (i.e., scanning) is detected by NIDS and MTD action(s) are taken based on the alerts



(b) It shows ASP when the NIDS detects the 2nd phase of attacks (i.e., exploitation) and MTD action(s) are taken based on the alerts. Undetectable scanning attacks in the phase 1 do not lead to trigger MTD action(s).

Figure 7: Security effectiveness of the hybrid MTD adoption

range of its targeted network space and the NIDS accordingly keeps generating alert messages within the internal time window. In this work, a scanning attack keeps sending malicious packets to disclose a vIP address of a target until a default Round-Trip Time (RTT) timeout for a packet response.

Figure 6(b) shows the FFJ of both the event-based and hybrid MTD in the response to alerts in phase 2 causes additional performance overhead in the system when compared to the time-based MTD. In these experiments, we assume that Snort raises alerts not against scanning attempts in the first phase but against exploitation in the second phase. As an alert is raised and an MTD is more often triggered, either the event-based or hybrid MTD results in the increase of performance degradation. It is because we assume that the attack time of 100s ($1/ATK_\lambda$) is shorter than T_{MTD} of 300 s in the time-based MTD. In short, the hybrid MTD may be more reactive to the occurrence of ongoing attacks than the time-based MTD. If it assumes the attack time was larger than an MTD interval, the gap of performance degradation in the hybrid was reduced.

B. Security

Our study compares the security benefits between three types of MTD systems (i.e., time-based, event-based and hybrid MTDs) by using the metric called ASP. We assume two phases of attack scenarios that an adversary first attempts to discover a current vIP of a target within an SDN network at least in the first phase and during the first phase, the preliminary is either detected or undetected by a NIDS. As part of the security experiments, we use two different scanning options (-sX or -sT) for scanning attempts, which are either detectable or undetectable respectively.

As can be seen in Figure 7(a), MTD with the event-based approach does not offer any additional protection against a detectable scanning attack when compared with a time-based MTD. This is due to the fact that an attacker can already discover a vIP address and proceed to the second phase even though the NIDS identifies malicious traffic. Our measured success probability at Figure 7(a) appears to be relatively higher because we are intentionally using a small network space size (i.e., 4096) and a relatively long (for that size) MTD interval of 300 s. Security effectiveness of a network shuffling technique is highly dependent on the size of the network space and the length of the MTD interval. Using a smaller network space for vIP addresses, we aim to measure multi-phases of an attacker that can proceed to the second phases, which leads to a higher success probability for an attacker.

In comparison with a detectable scanning attack, Figure 7(b) describes that the event-based MTD provides much better security effectiveness than the time-based MTD against dictionary and SQLi attacks because the defense system can react to the attacker's ongoing behaviors. According to our results, a time-based would appear to provide better security benefits than an event-based MTD in the case of a scanning attack. This is somewhat deceptive. When an attacker discovers a vIP address of a host under the time-based MTD scheme, the attacker has the time remaining in the current MTD interval to initiate an exploit like a dictionary attack or SQL injection before the vIP is changed by the time-based MTD. This isn't the case for an event-based MTD mechanism. When a host vIP is discovered by the attacker and an NIDS alert is generated, the event-based MTD may change the vIP of the host prior to execution of the exploit. That is, the time the vIP information is valid for the attacker will tend to be less under

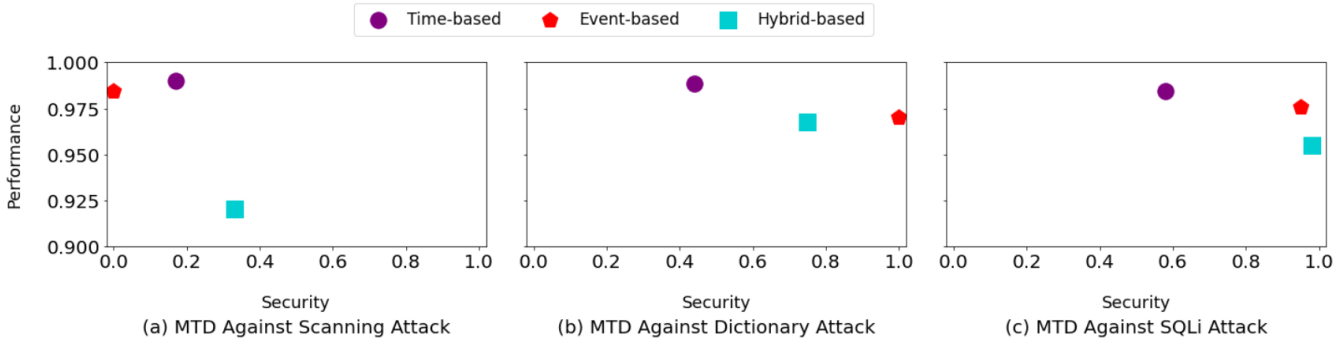


Figure 8: Security vs Performance in MTD systems

the event-based MTD scheme than under the time-based MTD scheme. While the time-based MTD is better at preventing the attacker's awareness of a targeted host or service (Figure 7(a)), the event-based MTD is better at preventing the attacker's exploitation of the host or service (Figure 7(b)).

The results also demonstrate that a hybrid MTD may enhance security when compared to either a time-based or an event-based MTD because it could proactively and reactively protect a system against both phases of attacks (phase 1 and phase 2). The success probability in the hybrid MTD is nearly 20% lower than that of the time-based MTD against a scanning attack, whereas the success probability of 0.02 in the hybrid MTD is smaller than 0.05 in the event-based MTD against an SQLi attack. However, even through the hybrid MTD provides more security benefits than the time-based MTD against a dictionary attack, the event-based MTD represents better security effectiveness than the hybrid system. It is due to the fact that the severity of a dictionary attack is ranked at 1/3 in a decision-making process in the hybrid system, which leads to taking an MTD action, AMI.

C. Performance vs. Security

There are advantages and disadvantages associated with using the timeliness-based MTDs within a physical SDN-based system in terms of performance and security. As can be seen in Figure 8, a networked system that adopts MTD techniques may thus result in a trade-off between performance and security due to the change in vIP addresses. The security on the x-axis is calculated by $1-ASP$, while the performance on the y-axis is computed by using $1-FFJ$.

When it comes to scanning attacks, the time-based MTD has better performance and security benefits than other types of MTDs as can be seen in Figure 8(a). By means of the time-based MTD, a vIP of the server is proactively changed on a regular basis, thereby limiting the amount of time that the attacker has to discover a target. The event-based mechanism may instead operate to change a vIP address as a result of the detection, which leads to late response to the reconnaissance acts. In contrast to a scanning attack, the hybrid MTD systems can provide better security advantages against the second phase of attacks (i.e., exploitation) as can be seen in Figure 8(b) and (c). When the hybrid MTD is utilized against the dictionary attack, its security benefit increases about 55% in comparison with the time-based MTD, and when the hybrid MTD is utilized against SQLi attacks, the security benefit increases about 60% in comparison with the event-based MTD. However, as a drawback of the hybrid MTD adoption, it results in an increase in performance loss compared to the other MTDs. For example, FFJ of the hybrid MTD against a dictionary attack has an increase of 0.0295 when compared to the time-based MTD and an increase of 0.0208 when compared to the event-based MTD. Of course, it is due to the fact that this work assumes one of the worse cases that ATK_λ is shorter than T_{MTD}. However, it is obvious that the MTD trigger not only enhances security against multi-phase attacks, but also increases performance overhead due to abrupt disconnections as a result of the MTD adoption.

D. Discussion

- DNS caching: The shuffling of vIP addresses may affect the performance of DNS caching mechanisms when multiple HTTP requests are made. However, It is not the scope of this study to measure its impact.
- System parameters: In order to assess performance and security, we used the determined system configurations and parameters.
- Optimization problems: This work first focuses on evaluating performance and security of a hybrid system in comparison with a time-based and an event-based MTD. We will be able to solve the optimization problems in a hybrid MTD adoption.

VII. Conclusions

We have presented an ML-based model for rule classifications for decision-making and evaluation of performance and security in an SDN adopting a hybrid MTD mechanism. We have proposed a k-NN model to predict the classifications of *miscellaneous* rules and then used them for decision-making for security enhancement. Experimental results in a physical SDN testbed have shown that a hybrid-based MTD system can enhance security against multi-phase of cyber-attacks when compared with other MTD techniques (i.e., time-based or event-based MTD). The combined functions have provided better security effectiveness in terms of Attack

Success Probability (ASP) against both reconnaissance and exploitation of attacks. However, we have also presented in our experiments that a hybrid MTD may cause more performance degradation by using the metrics including Fraction of Failed Jobs (FFJ) in an SDN when an attack time is shorter than MTD intervals of a time-based MTD.

ACKNOWLEDGMENT

This material is based upon work supported by the International Technology Center Indo-Pacific (ITC IPAC) under Contract No. FA520920C0022. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ITC IPAC.

The authors would like to acknowledge the UQ Cyber Device Testing Lab for equipment and lab support.

REFERENCES

[1] J. H. Cho, D. P. Sharma, H. Alavizadeh, S. Yoon, N. Ben-Asher, T. J. Moore, D. S. Kim, H. Lim, and F. F. Nelson, "Toward Proactive, Adaptive Defense: A Survey on Moving Target Defense," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 1, pp. 709–745, 2020.

[2] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati, "A survey of moving target defenses for network security," *IEEE Communications Surveys and Tutorials*, pp. 1–33, 2020, Less practical survey.

[3] E. M. Hutchins, M. J. Cloppert, R. M. Amin, *et al.*, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research*, vol. 1, no. 1, p. 80, 2011.

[4] D. P. Sharma, D. S. Kim, S. Yoon, H. Lim, J. H. Cho, and T. J. Moore, "FRVM: Flexible Random Virtual IP Multiplexing in Software-Defined Networks," in *Proceedings - 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and 12th IEEE International Conference on Big Data Science and Engineering, Trustcom/BigDataSE 2018*, 2018, pp. 579–587.

[5] T. Moghaddam, M. Kim, J.-H. Cho, H. Lim, T. J. Moore, F. F. Nelson, and D. D. Kim, "A practical security evaluation of a moving target defence against multi-phase cyberattacks," in *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, IEEE, 2022, pp. 103–110.

[6] C. Turner and A. Joseph, "A statistical and cluster analysis exploratory study of snort rules," It proposed 3 groups of rule clustering based on the protocol field such as TCP, UDP, ICMP using Snort rulesets and validated it by silhouette coefficient at 0.84, vol. 114, Elsevier B.V., 2017, pp. 106–115.

[7] N. Scarabeo, B. C. Fung, and R. H. Khokhar, "Mining known attack patterns from security-related events," *PeerJ Computer Science*, vol. 2015, 10 2015.

[8] J. Mendonça, J. H. Cho, T. J. Moore, F. F. Nelson, H. Lim, A. Zimmermann, and D. S. Kim, "Performability analysis of services in a software-defined networking adopting time-based moving target defense mechanisms," in *Proceedings of the ACM Symposium on Applied Computing*, 2020, pp. 1180–1189.

[9] C. Dishington, D. P. Sharma, D. S. Kim, J.-h. Cho, T. J. Moore, and F. F. Nelson, "Security and performance assessment of ip multiplexing moving target defence in software defined networks," *2019 IEEE International Conference on Trust, Security And Privacy In Computing And Communications*, 2019.

[10] T. E. Carroll, M. Crouse, E. W. Fulp, and K. S. Berenhaut, "Analysis of network address shuffling as a moving target defense," in *2014 IEEE International Conference on Communications (ICC)*, IEEE, Jun. 2014, pp. 701–706. DOI: 10.1109/ICC.2014.6883401.

[11] N. O. Ahmed and B. Bhargava, "Mayflies: A moving target defense framework for distributed systems," in *Proceedings of the 2016 ACM Workshop on Moving Target Defense*, 2016, pp. 59–64.

[12] B. C. Ward, S. R. Gomez, R. Skowyra, D. Bigelow, J. Martin, J. Landry, and H. Okhravi, "Survey of cyber moving targets second edition," MIT Lincoln Laboratory Lexington United States, Tech. Rep., 2018.

[13] D. C. MacFarland and C. A. Shue, "The sdn shuffle: Creating a moving-target defense using host-based software-defined networking," in *Proceedings of the Second ACM Workshop on Moving Target Defense*, ser. MTD '15, Denver, Colorado, USA: Association for Computing Machinery, 2015, pp. 37–41, ISBN: 9781450338233. DOI: 10.1145/2808475.2808485.

[14] H. Shacham, M. Page, B. Pfaff, E.-J. Goh, N. Modadugu, and D. Boneh, "On the effectiveness of address-space randomization," in *Proceedings of the 11th ACM conference on Computer and communications security*, 2004, pp. 298–307.

[15] G. S. Kc, A. D. Keromytis, and V. Prevelakis, "Countering code-injection attacks with instruction-set randomization," in *Proceedings of the 10th ACM conference on Computer and communications security*, 2003, pp. 272–280.

[16] N. O. Ahmed and B. Bhargava, "Bio-inspired formal model for space/time virtual machine randomization and diversification," *IEEE Transactions on Cloud Computing*, 2020.

[17] S. Vikram, C. Yang, and G. Gu, "Nomad: Towards non-intrusive moving-target defense against web bots," in *2013 IEEE Conference on Communications and Network Security (CNS)*, IEEE, 2013, pp. 55–63.

[18] P. Larsen, A. Homescu, S. Brunthaler, and M. Franz, "Sok: Automated software diversity," in *2014 IEEE Symposium on Security and Privacy*, IEEE, 2014, pp. 276–291.

[19] Q. Jia, H. Wang, D. Fleck, F. Li, A. Stavrou, and W. Powell, "Catch me if you can : A cloud-enabled ddos defense," IEEE, 2014.

[20] J. Hong and D. S. Kim, "HARMs: Hierarchical attack representation models for network security analysis," *Proceedings of the 10th Australian Information Security Management Conference, AISM 2012*, pp. 74–81, 2012.

[21] J. Jafarian, E. Al-Shaer, and Q. Duan, "Openflow random host mutation: transparent moving target defense using software defined networking," ... *Topics in Software Defined Networks*, pp. 127–132, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2342467.

[22] S. Antonatos, P. Akritidis, E. P. Markatos, and K. G. Anagnostakis, "Defending against hitlist worms using network address space randomization," *Computer Networks*, vol. 51, no. 12, pp. 3471–3490, 2007. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1389128607000710.

[23] H. Zhang, K. Zheng, X. Wang, S. Luo, and B. Wu, "Efficient strategy selection for moving target defense under multiple attacks," *IEEE Access*, vol. 7, pp. 65 982–65 995, 2019.

[24] J. H. Cho and N. Ben-Asher, "Cyber defense in breadth: Modeling and analysis of integrated defense systems," *Journal of Defense Modeling and Simulation*, vol. 15, pp. 147–160, 2 Apr. 2018.

[25] R. Zhuang, S. Zhang, A. Bardas, S. A. DeLoach, X. Ou, and A. Singhal, "Investigating the application of moving target defenses to network security," IEEE Computer Society, 2013, pp. 162–169.

[26] W. Connell, D. A. Menasce, and M. Albanese, "Performance Modeling of Moving Target Defenses with Reconfiguration Limits," c, 2018.

[27] J. Fonseca, M. Vieira, and H. Madeira, "Evaluation of web security mechanisms using vulnerability & attack injection," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, pp. 440–453, 5 2014.

[28] Y. Zhou, Y. Hu, G. Cheng, Y. Zhao, S. Jiang, and Z. Chen, "A cost-effective shuffling method against ddos attacks using moving target defense," *Proceedings of the 6th ACM Workshop on Moving Target Defense*, pp. 57–66, 2019.

[29] Cisco, *Snort*, [Online]. https://www.snort.org. Accessed on 13/08/2022., 2022.

[30] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[31] The Apache Software Foundation, *Ab - apache http server benchmarking tool*, [Online]. https://httpd.apache.org/docs/2.4/programs/ab.html. Accessed on 13/08/2022., 2022.

[32] The OWASP Foundation, *Owasp top 10*, [Online]. https://owasp.org/www-project-top-ten/. Accessed on 13/08/2022., 2022.