https://doi.org/10.26682/sjuod.2024.27.1.5
Journal of University of Duhok., Vol. 27, No.1 (Pure and Engineering Sciences), Pp 65-75, 2024

# DYNAMIC HONEYPOT DEPLOYMENT IN SDN: INTEGRATING MOVING TARGET DEFENSE AND DECEPTION MECHANISMS FOR ENHANCED CYBERSECURITY

**H**ARMAN **Y. I. K**HALID[*] and **N**AJLA **B. I. A**LDABAGH[**]
[*]Dept. of Computer Science, College of Science, University of Duhok, Kurdistan Region-Iraq
[**]Dept. of Computer Science, College of Computer Science and Mathematics,
University of Mosul-Iraq

***(Received:, July 14, 2024; Accepted for Publication: July 18, 2024)***

## ABSTRACT

**Deception mechanisms such as honeypots proved to be effective security mechanisms that lure cyber attackers to fake services away from real services , log their behavior to be analyzed in order to extract knowledge about their operation. Honeypots have been adopted by the research community since they can detect passive scanning attacks, which attackers usually perform to collect knowledge about the current network to prepare for larger attacks. However, the static deployment of honeypots makes them easier to be exposed by skilled attackers. Therefore, it is necessary to make honeypot deployments in the network dynamically and proactively. To overcome this shortcoming, Moving Target Defense (MTD) is a solution technology that changes network configuration parameters efficiently, either reactively or proactively, to falsify the details collected by attackers in order to disrupt the intended cyber attack. In this paper, we present a review of the current work on MTD techniques in Software Defined Networking (SDN) environment and highlight some important requirements for MTD applications and their issues. Moreover, we present a theoretical model of a cyber security mechanism combining MTD with a deception mechanism implemented as an SDN controller.**

***KEY WORDS:*** Moving Target Defense (MTD), IP Shuffling, Topology Shuffling, Software Defined Networking (SDN), Cybersecurity, Deception, Honeypot, Reconnaissance, Intrusion Detection.

## 1. INTRODUCTION

Even though there are several security measures, such as intrusion detection systems and firewalls, to identify and stop attackers from accessing vital resources, these measures are still not effective forms of defense. The cyber deception technique is a proactive defense mechanism intended to draw adversaries and divert them from prospective targets and vulnerabilities, as opposed to the reactive defense method of waiting for attackers to breach the network system before swiftly blocking them. In cybersecurity research, deception techniques were first proposed in the late 1980s to trace intruders, and the concept of honeypot followed (Zhou et al., 2021) . Defensive deception tactics offer proactive defense services by offering an additional layer of protection to more conventional security solutions like firewalls, intrusion detection systems, or endpoint anti-virus software (Ge et al., 2022). As a cyber-trap, honeypots are used to fool hackers into thinking they are legitimate network resources, they are placed around the network to consume attackers' time and effort (Duy et al., 2022). In order to create efficient countermeasure defenses, the collected honeypot log files are examined further to understand the attackers' tactics and the means by which the network was breached. Depending on the amount of interaction, honeypots can be categorized as either High Interaction Honeypots (HIH) or Low Interaction Honeypots (LIH). A LIH only simulates certain network elements, whereas an HIH is the entire operating system. A decision must be made before installing either solution since an HIH is costly to build and maintain because both the OS and the application (the attacking target service) are built on actual hardware. A LIH, on the other hand, implements the OS and the application as a virtual environment. Hybrid honeypots are a hybrid between HIH and LIH (Valdovinos et al., 2021).

There are two types of honeypot deployment strategies: static and dynamic. Conventional honeypot systems primarily rely on physical devices or virtual machines, and need complex configuration and deployment

procedures (Y. Gao et al., 2021). A large number of network systems in use today are static. They are created, configured once, and then left alone for a long period of time (Jalowski et al., 2022). From the standpoint of functional requirements, this strategy makes sense: network architecture and topology are created to satisfy needs and minimize costs if use cases are well specified (Jalowski et al., 2022). As a result, it is more likely to choose static deployment tactics, which are inflexible and do not scale well (Y. Gao et al., 2021).

However, because modern networks are static, potential attackers have an unfair advantage because they can watch a network for extended periods of time and gather information about network hosts or traffic patterns. The same problem arises when a honeypot is deployed statically, which can be extremely advantageous to attackers in terms of how much time and effort they can put into their attacks. Attackers can conduct reconnaissance on the target system, identify any potential weaknesses, distinguish the honeypot from genuine nodes, and select the most effective configuration to launch a devastating, large-scale cyberattack with the help of this static configuration (Steinberger et al., 2018). The acquired intelligence regarding the system setup may be valid for a very long time due to the static nature of current networks (Jalowski et al., 2022). Therefore, dynamic deployment tactics have steadily gained traction as virtualization technology and Software Defined Networking (SDN) have advanced. These tactics will modify the honeypot deployment scenario at various phases based on the attackers' current activities (Y. Gao et al., 2021).

In this paper, we present a theoretical model of deploying Moving Target Defense (MTD) techniques to solve the issue of honeypot static deployment from our previous work in

(Yousif Khalid et al., 2024) . This paper is organized as follows: In Section 2, we introduce an overview of MTD defense, its techniques, and the present requirements for MTD application development in SDN. In Section 3, we first briefly summarize related work on the implementation of MTD in SDN. In Section 4, a theoretical model of the proposed MTD mechanism is presented. In the last section, we conclude this paper and discuss future directions.

## 2. MOVING TARGET DEFENSE

### *2.1 Background*

The primary security issue with traditional networks is that the attacker always has the upper hand when it comes to defense. An attacker can carefully plan their actions in advance by conducting network reconnaissance and preparing the necessary tools. By forcing network elements to move, Moving Target Defense (MTD) mitigates the issue of traditional defense measures, which arise from the static nature of network services and configurations, and lowers an attacker's ability (Ryapukhin et al., 2022). MTD mechanisms are carried out from either proactive or reactive viewpoints. In proactive defense, attack prevention occurs prior to major damage, and this method is very useful for passive attacks. While the reactive defense involves a defensive response after an attack occurs or when an attack is detected (Galadima et al., 2022). There are benefits and drawbacks to both MTD and cyber deception. By drawing enemies to attack the bait, cyber deception can accomplish the goal of safeguarding crucial resources, but it lacks dynamic and unpredictable elements. Fortunately, MTD has the ability to compensate for this deficiency. Goals that MTD and cyber deception cannot accomplish together can be successfully resisted by cyber deception and persistent network reconnaissance attacks (C. Gao et al., 2021).

The basic idea, behind MTD is to defend against attackers by continuously changing attack surfaces (e.g., system/network configurations) to increase attack complexity, cost, and time, as well as increase attacker uncertainty and/or confusion, which invalidates the system intelligence collected by the attackers (Ge et al., 2022) (Sharma et al., 2018). MTD alters the attack surfaces arbitrarily and continuously in order to disrupt the Cyber Kill Chain (CKC) (Zhou et al., 2021), which shortens the attacker's window of opportunity and raises the attack efforts (Luo et al., 2019). Frequent alterations to the attack surface heighten the attacker's uncertainty regarding the characteristics of the target and complicate the process of identifying and exploiting system vulnerabilities

(Cho et al., 2018).

To identify an intruder, the stationary security system employs attack detection methods in the application layer or security devices like intrusion detection systems (IDS). Rather than looking for insiders, MTD uses persistent network alterations

to trick the attacker instead of detecting them, in contrast to stationary security measures. Establishing a framework for MTD implementation in traditional networks is challenging because of the nature of MTD and its management challenges. Such a framework is possible thanks to SDN, which enables administrators to set up and put into practice the best movement strategies and carry out network operations dynamically (Valdovinos et al., 2021). A number of SDN features, like globe network view, logical network control centralization, and network programming capabilities, are beneficial for creating and administering MTD applications in an effective and adaptable manner (Luo et al., 2019). SDN is getting popular in designing security solutions. The centralized controlling attributes enable greater ease for designing the MTD solution
(Faraz Hyder et al., 2021) (Ryapukhin et al., 2022).

### *2.2 MTD Techniques*

Moving Target Defense (MTD) can be accomplished by using three general classes: diversity, redundancy, and shuffling. The main goal of all three of these strategies is to modify the system's characteristics, either on a regular basis or in response to specific events. Shuffling's aim is to confuse attackers in the reconnaissance stage by changing network/system configurations such as network addresses (e.g., IP addresses, MAC addresses, or port numbers), software migration, or network topology. The redundancy strategy, as illustrated in Figure 1, creates duplicates of services or apps with the same functionality in order to increase resources with the same abilities to trick attackers (Zhou et al., 2021). In order to maintain high system reliability, it dynamically uses numerous clones of system components inside a network to provide security measures (Ge et al., 2022). The diversity strategy uses different kinds of system components or platforms (e.g., software, operating systems) that offer the same functionalities, which raises attack complexity. Examples include altering the webserver software or DBMS platform. Current MTD methods additionally take advantage of the combination of these strategies.

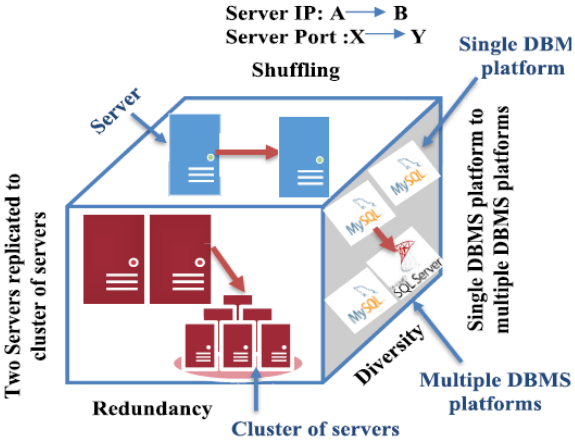


**Fig (1):** Main categories of MTD techniques (Hyder et al., 2021)

.

Several MTD techniques that fall into two primary categories—network-level and host-level—have been published in recent years. At the network level, rearranging network attributes, such as the IP address, MAC address, or port number, is a frequent MTD strategy. The host-level MTD focuses on modifications to the operating system and host (such as name and configuration) or altering system configurations concerning the run-time configuration of instruction sets (Sharma et al., 2018) (Steinberger et al., 2018). The most popular MTD strategy is network address shuffling, which aims to invalidate the address information attackers have gathered based on the present network IP setup by dynamically and regularly changing the IP address of a target system (Sharma et al., 2018).

## 3. LITERATURE REVIEW

### *3.1 Related Work*

The author in (Sharma et al., 2018) proposes flexible random virtual IP multiplexing (FRVM) which defends against network reconnaissance and scanning attacks. FRVM enables a host machine to have multiple, random, and time-varying virtual IP addresses that are multiplexed to the real IP address of the host. They used a DNS server to resolve the domain name of the server host and return the IP address. In another similar work, (Galadima et al., 2022) designed an MTD reactive and proactive approach to network address shuffling against DDoS attacks in an IoT environment based on SDN. The used method of multiplexing where single real IP address where mapped to several randomized virtual IP address. In (Luo et al., 2019), they proposed MTD and

SDN-based honeypots to defend against DDoS attacks in the IoT to prevent fingerprinting by attackers through MTD shuffling based on IP address mutation. They used several scanning tools, and no real IP address was discovered; only some virtual IP addresses were discovered, which will be useless a few time later. In another work that utilized Honeypot, the author in (Ryapukhin et al., 2022) designed a new MTD approach that includes a set of system configurations and their security mechanisms that change system configurations when an attack occurs.

SDN-based MTD for Control and Data Plan Security (SMCDS) was proposed by the author in (Hyder et al., 2021). By responding to malicious probe traffic with shadow controllers rather than actual controllers, they used the distributed controller idea to safeguard the control plane. When reconnaissance traffic is directed at the controller, the shadow controller generates a controlled response. They used the MTD technique of IP and port shuffling on a periodic basis to take a proactive and reactive strategy for servers connected to the data plane. Moreover, they used Snort IDS in proposed security mechanism and deception controllers. In another work (Duy et al., 2022), they proposed a mechanism that redirects suspicious actions to decoys through instructing flow rules in OpenFlow switches. They used Honeypot to lure attackers and deployed IP shuffling to hide network details from attackers. They periodically change the type of honeypot and deployment platform.

In the work of (Belalis et al., n.d.), they proposed a deception mechanism that combined MTD with honeypot. Their method presents a virtual network topology to mislead intruders to hide the real network, along with possible vulnerabilities that attackers could exploit by presenting falsified information the intruder collects. The controller sends ARP, ICMP, and UDP packets from the intruder to the packet handler module to create a response packet according to the virtual network and send it back. Intruder packets are considered that are designated to honeypots or hosts that send multiple SYN messages from one source to multiple destination ports of another network host. In (Ge et al., 2022), the author presented an integrated defense mechanism for proactive MTD by rearranging the IoT network's topology, which includes both decoy and actual nodes. This approach maximizes attacker complexity while decreasing defense costs for carrying out MTD actions. They used three methods of shuffling genetic algorithms: decoy attack path based optimization and random, and four methods of temporal shuffling: fixed, random, adaptive, and hybrid. In another work, the author of (C. Gao et al., 2021) employed a technique that involved exploiting virtual network topology to trick opponents into thinking that hosts and a decoy node were part of the system. Additionally, they used virtual topology-based IP randomization to make it difficult for attackers to distinguish between genuine and phony nodes.

In (Zhou et al., 2021), they established a hybrid proactive defense mechanism combining MTD and cyber deception to spread falsified information to confuse attackers. They propose DDoS mitigation mechanism in IoT while maintaining high performance with acceptable overload by using multistage signaling game model in which defender thwart attacker using proactive defense technique. To simulate the interplay between the attacker and the defender, game theory has been established. It is difficult for the defense to determine if messages they have received are malicious or from legitimate users. As a result, signaling games more effectively represents players' dynamic activity and takes proactive steps to defend against DDoS attacks.

**Table (1):** Work Implemented MTD in SDN Environment.

| Article | Publish Year | Attack Type | Environment | IoT | Deception Mechanism | IDS | MTD Technique | MTD Level | Defense Mode | Metric Used |
|---|---|---|---|---|---|---|---|---|---|---|
| (Sharma et al., 2018) | 2018 | Scanning | LAN | X | X | X | IP Shuffling | Network level | Proactive | -Attack Success Probability vs. the number of scans and the number of discovered hosts |
| (Steinberger et al., 2018) | 2018 | DDoS | ISP | X | X | X | IP Shuffling | Host and Network Level | Proactive | -Bandwidth Availability<br>-Probability of Successful Attack |
| (Luo et al., 2019) | 2019 | DDoS | LAN | X | X | X | IP Shuffling | Network Level | Proactive | -Number of Scanned Devices<br>-TCP Connection Time Delay |
| (Belalis et al., n.d.) | 2020 | Scanning | LAN | X | ✓ | X | IP, MAC and Topology Shuffling | Network Level | Reactive | -NMAP Scan Time<br>-Number of Device Detected |
| (Hyder et al., 2021) | 2021 | Scanning | LAN | X | ✓ | ✓ | IP and Port Shuffling | Network Level | Proactive and Reactive | -Attacker Effort<br>-Defender Cost<br>-Number of Scans vs. Defender Success |
| (C. Gao et al., 2021) | 2021 | Scanning | LAN | X | ✓ | X | IP and Topology Shuffling | Network Level | Reactive | -Scan Time<br>-Number of Attacked Hosts<br>-Network Latency |
| (Ge et al., 2022) | 2021 | Scanning + Data Exfiltration | LAN | ✓ | ✓ | X | Topology Shuffling | Network Level | Proactive | -Number of Attack Paths Toward Decoy Target<br>-Mean Time to Security Failure<br>-Defense Cost |
| (Zhou et al., 2021) | 2021 | DDoS | LAN | ✓ | ✓ | X | IP Shuffling | Network Level | Proactive | -Average Survival Rate of the Production System<br>-CPU Load of the controller<br>-Packet Loss Rate<br>-RTT |
| (Faraz Hyder et al., 2021) | 2021 | Scanning | LAN | X | X | ✓ | IP Shuffling + MTD Diversity | Host and Network Level | Proactive | -IP Address Discovered in Unit Time. |
| (Ryapukhin et al., 2022) | 2022 | DDoS | LAN | X | X | X | IP Shuffling | Network Level | Reactive | -Load of Attacked Host (KB/s) |
| (Galadima et al., 2022) | 2022 | DDoS | LAN | ✓ | X | X | IP Shuffling | Network Level | Proactive and Reactive | -Latency (RTT)<br>-Bandwidth<br>-Jitter. |
| (Duy et al., 2022) | 2022 | Scanning | - | X | ✓ | ✓ | IP Shuffling + Honeypot Diversity | Host and Network Level | Proactive | -Scanning Time vs. Discovered Hosts |

### *3.2 MTD Requirement and Consideration in SDN*

In this section, we introduce the requirements of any defense model using MTD strategies that should be considered to provide effective and efficient security mechanisms in an SDN environment:

**1. Dedicated controller for shuffling-based MTD**

As discussed in the previous section, applying efficient MTD techniques has become possible thanks to SDN because of its features, which facilitate its management process. The controller should be at the center of managing MTD because of its global view, centralized control, and programmability. SDN switches must not participate in the defense mechanism since they must remain the only forwarding device in the OpenFlow protocol state. Although SDN controller is the optimal location to implement MTD applications, as shown in the literature, most research articles deployed MTD in controllers, but it creates overhead issues if the network is with a single controller, especially when MTD techniques such as IP shuffling happen frequently. The SDN controller is considered a point of failure of the system; if it fails, then the whole system shuts down (Ibrahim et al., 2020) (Yousif et al., 2024). The solution to this issue is by using more than one controller with collaboration between them in the SDN environment and dedicating a special controller to managing the MTD defense, similar to the work done in
(Hyder et al., 2021), where shadow controllers were dedicated to responding to reconnaissance attacks. Moreover, using a load balancer to deal with each probe packet by selecting a controller from a pool will increase attacker uncertainty.

**2. When to perform MTD**

When to perform MTD: The timing to execute a proactive MTD defense system or take action is very critical. Constant timing also faces issues, such as using a short time period, which will lead to executing MTD more frequently, which may degrade the performance. Performance will unavoidably suffer from frequent MTD modifications (Zhou et al., 2021). While using a long period may not lead to an effective deception defense mechanism since the attacker could get valuable information about the hosts, services, and system vulnerabilities during this period. Moreover, attackers monitoring the network state can learn about the deception defense system and the constant timing of execution. Then attackers could recognize the MTD technique, and those details could be useful for later bypassing the system. The solution to this issue could be:

A. Using random timing for MTD execution. Instead of constant time interval, we define range of time (*T-min and T-max*) and the defender need to select randomly time value between this range.

B. Using a reactive MTD approach by reacting to events triggered by the detection of an attack, the unavailability of a server or a channel, or depending on a threshold value in the system to exceed to execute as in (Chowdhary et al., 2016).

**3. MTD combining with deception defense**

MTD combining with Deception Defense: Some works in the literature implement deception mechanisms such as LIH along with MTD. However, the issue with LIH is that they could be recognized by advanced attackers, thus minimizing the opportunity to get more details from them. The solution could be engaging HIH in the defense system such that MTD performs traffic migration of malicious traffic from LIH to HIH to make attackers spend more time without noticing. However, MTD performing traffic migration should provide reliability and consistent service without degrading performance.

**4. Unpredictability**

Unpredictability: The attacker and normal user must not predict the adjustments and changes made to the chosen network configuration. MTD strategies must occur smoothly, consistently, and transparently to the end user. The user as well as the attacker will not feel the changes made by MTD. Moreover, the defense model must make sure that after configuration changes, the real services will not be affected by the MTD mechanism or disrupt active connections.

**5. Follow OpenFlow protocol**

Follow OpenFlow Protocol: The model implemented in the SDN environment should not invoke the SDN switch in the defense solution since it must remain the only forwarding device following OpenFlow protocol standards (KHALID et al., 2019).

**6. Combining multiple MTD stratigies**

Combining Multiple MTD strategies: MTD defense strategies increase attackers cost and uncertainty. The defense solution should support multiple MTD strategies combined together to create more robust security system.

**7. System Security policies**

System Security Policies: The system that establishes a framework for the implementation of effective MTD solutions must guarantee that

the solutions are in accordance with the overall security policy and do not result in unexpected side effects due to conflicts with other flow rules in the system.

**8. Network latency and system overhead**

Network Latency and System Overhead: The likelihood of a host being successfully attacked is further diminished by MTD mechanisms, such as IP shuffling. As the frequency of address changes rises, the network latency also increases (C. Gao et al., 2021). The solution could be using a methodology that decreases the frequency of system changes.

**9. Scalability issue**

The majority of research conducted in MTD in an SDN environment was evaluated on a simulator or in small-scale local network implementations. Additional study is required to investigate the uses of MTD approaches in large-scale networks (Jalowski et al., 2022). As the network size increases, the MTD cost and service latency increase during reconfiguration of the network, which eventually leads to performance degradation.

**10. Lack of metrics**

Lack of Metrics: As shown in the literature, the MTD is proven to be an efficient security method, but there have been no substantial commercial applications of MTD techniques (Jalowski et al., 2022). The absence of measures to evaluate the efficacy of particular MTD solutions, as shown in Table 1, is one of the reasons that they are still not mature technologies and need to be standardized.

Motivated by the literature as well as the hypothesis we come up with for MTD requirements and consideration, we propose a theoretical model. This theoretical model has two purposes: to fix the problem of the dynamic deployment of honeypots we had in earlier work by using different MTD strategies, and to decrease attackers uncertainty about their knowledge about the network nodes.

## 4. PROPOSED MODEL

### *4.1 Theorotical Model*

In our previous (Yousif Khalid et al., 2024) , to detect attacker who scan the network for vulnerabilities, honeypot were used for luring attackers to trap as well as they are used to trigger the detection module to check the network for malicious traffic. In general, the deception defense suffer from its static deployment, which lead to be identified with time by sophisticated attackers whom bypass the defense mechanism after network analysis. Dynamic deployment of deception defense becomes hot topics in current research community, which change the honeypot location dynamically, and frequently according to predefined policies. To overcome the drawback of our previous work and make honeypot deployment dynamically, we propose a theoretical model utilizing the dynamic and random nature of MTD to be extended to our detection mechanism to enhance the defensive effectiveness of the network deception system. MTD is regarded as a technology that protects against network spying and scanning threats. A customized suite of software tools is frequently employed by an attacker to scan the target system in order to find information such as operating system kinds, IP addresses, port numbers, running services, protocols, network topology, and/or exploitable vulnerabilities.

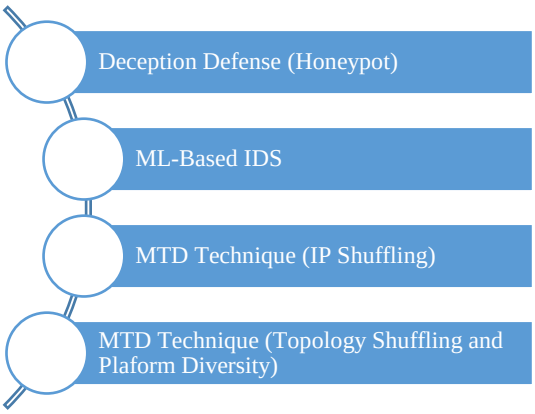


**Fig (2):** Multi-layer Defense approach.

This work, in conjunction with our previous work, provides a four-layer protection against insider reconnaissance, as depicted in Figure 2. In the first line of defense, honeypot is used as a deception mechanism to lure attackers as well as a trigger detection mechanism when there is contact with honeypot. The second line of defense is a machine learning-based IDS located in the SDN controller as a checking module to sense the traffic, classify whether it is malicious or normal, and block malicious insiders generating malicious reconnaissance traffic. The third line of defense is the MTD IP shuffling technique, which proactively changes the IP addresses of real and decoy nodes in the network to prevent them from being recognized. Fourth line of defense is topology shuffling MTD technique, which dynamically inject new decoy nodes and relocate all decoy nodes as well as provide host-level MTD

by presenting different services thus provide diversity as well as redundancy. Honeypot can provide different platforms for servers, and within these platforms, many versions are implemented to heighten the level of uncertainty for the attackers. The insider will get a variety of platform information when they conduct reconnaissance to gain information about the servers in the network.

The internal reconnaissance is strongly countered by these four approaches, which also guarantee that the information obtained by the insider during the reconnaissance is inaccurate due to its potential for change after a period. The issue of applying MTD in traditional network as explained in previous section such as management issue as well as operational cost as well as hardware requirement were luckily solved by utilizing the flexibility and programmability of SDN to build an SDN-based network deception defense system (C. Gao et al., 2021). The lightweight framework we have proposed will be deployed as apps on top of the SDN controller, and it can significantly reduce the overhead imposed by MTD techniques and eliminate excessive defense costs (Zhou et al., 2021).

In the proposed work, we have integrated a defense system that is equipped with IDS using machine learning, deception using honeypots, and MTD. When costs are a concern or a temporary delay of an assault might give significant benefits to a system (e.g., the need to delay the attacker's infiltration into the system until MTD is deployed), deception techniques are used. In addition, IDS is beneficial when deception or MTD strategies are unable to prevent intrusions as a result of performance degradation, deployment overhead, or highly sophisticated attackers who can readily discover the deception or identify the MTD strategy. Nevertheless, the attacker may be able to evade IDS by conducting passive scanning assaults or may be difficult to detect due to their constant patterns. That is, each defense mechanism has its own strengths and weaknesses, which can be enhanced by the integration of several defense techniques in order to provide a solid defense against attacks and a light-weight defense.

### *4.2 System Architecture*

In this section, we describe each module of the proposed system as depicted in Figure 3, as well as their tasks in the defense system. The components of the proposed system architecture include Honeypot Management and MTD Application modules.

#### *4.2.1 Honeypot Management Module:*

The network topology shuffling that can successfully disrupt the attack actions was not considered in the majority of research in the literature. Moreover, as previously mentioned in the previous section, the integration of two MTD approaches will result in a stronger security mechanism. The MTD-based network shuffling can not only confuse the attacker by altering connections among network devices but also create a false impression of the network and divert the attacker from genuine valuable devices through the deployment of decoys. This can effectively increase the attack effort and cost while decreasing the chances of real devices being compromised. Therefore, we implemented IP shuffling along with network topology shuffling to dynamically create decoy nodes and relocate them to different system points to provide a fake view for the attacker that can effectively and efficiently mitigate the attack effect. As seen in Figure 3, honeypot management module is mainly responsible for generating the virtual network topology, including decoy node generation, and dynamically decoy node deployment. Moreover, this module is responsible for managing different services to ensure diversity and redundancy in MTD. Our topology is made of several hosts and servers as real nodes, along with several honeypots as deception nodes, to construct a virtual network topology. The honeypot deployment mechanism, which is a part of the topology shuffling strategy as well as the IP shuffling mechanism, will occur proactively and reactively. If there is no attack, it will follow a variable time interval; otherwise, when an attack has been detected, both mechanisms will execute after attack mitigation. Honey log files provide some valuable information about attacker behavior that could be extracted and become useful, such as the location of the attacker, the goal of the attacker, the services he tries to exploit, etc. Based on this information, the honeypot management module will create more decoy nodes around the same node that was exploited before in case it is compromised again and keep real nodes away from it. The honeypot management module is responsible for the services that honeypots will provide to the attacker after a scanning attack.

#### *4.2.2 MTD Application Module:*

In order to enhance the defensive capabilities of the cyber deception system, we have implemented the MTD application module, which executes the entire IP shuffling operation.

The address conversion between the host and the decoy node in the network is coordinated by this module. We incorporate IP shuffling technology that is based on the virtual network topology, in which the IP addresses of both the genuine node and the decoy node are regularly updated. The attacker must reprobe the network, as the information they have collected will become invalid following the implementation of IP shuffling, despite the fact that they can obtain some knowledge about the network system within a period of time.

### *4.3 Defense System Operation Scenario*

Each host in the network will have a single rIP address and a set of several vIP addresses. The MTD application module is responsible for determining the optimal set of new vIPs for hosts. As shown in Figure 3 , Host 1 tries to get the IP address of Host 2 through a DNS query by domain name to send traffic. Named hosts are reachable via the virtual IP addresses acquired via DNS. The edge switch that receives a request from Host 1 will send the request to the controller in the MTD module. The SDN controller will perform the following tasks:

1- The controller contacts DNS to get the destination rIP address of Host 2, then sends it to the MTD module to find its corresponding vIP, creates a DNS response, and sends it back to host 1, including the Host 2 vIP.

2- The controller will install the necessary flows in all switches in the paths between both hosts.

3- The controller will install special flow rules in both edge switches between sender and receiver to redirect the traffic packets to the controller, then forward them to the MTD module to translate between real and virtual IP addresses.

Host1 will receive only a randomly selected vIP of the destination host, mapping it to the destination host, thinking it is a real one, and then sending the traffic to the network. The first and last switches in the path between two hosts are involved in our mechanism and will have two modes of operation, as follows:

1- Forwarding mode: In this mode, when the first traffic packet is received by the first switch in the path, which is the edge switch, it is forwarded to the controller to replace the sender rIP with vIP. However, this method may create controller overhead since each packet must be forwarded to the controller for translation. A future work could be using a dedicated controller for the translation process, where the main controller instructs the edge switches to send packets to the dedicated controller.

2- Smart mode: In this mode, the edge switch of the first switch in the path has the least intelligence to perform translation without sending packets to the controller. However, this method is against the OpenFlow protocol specification, where they are only considered a forwarding device and no other task should be given.

In both modes, after the packet header is modified and sent through the path, each switch in the middle of the path will simply forward to the next, following the flow rules installed previously by the controller. Once the final switch before the destination host receives the traffic packets, it will again follow one of the two previous modes, and the vIP of the destination host will be replaced with the rIP. The source and destination IP addresses of traffic between two hosts will be only virtual addresses, and the process will be transparent to users.

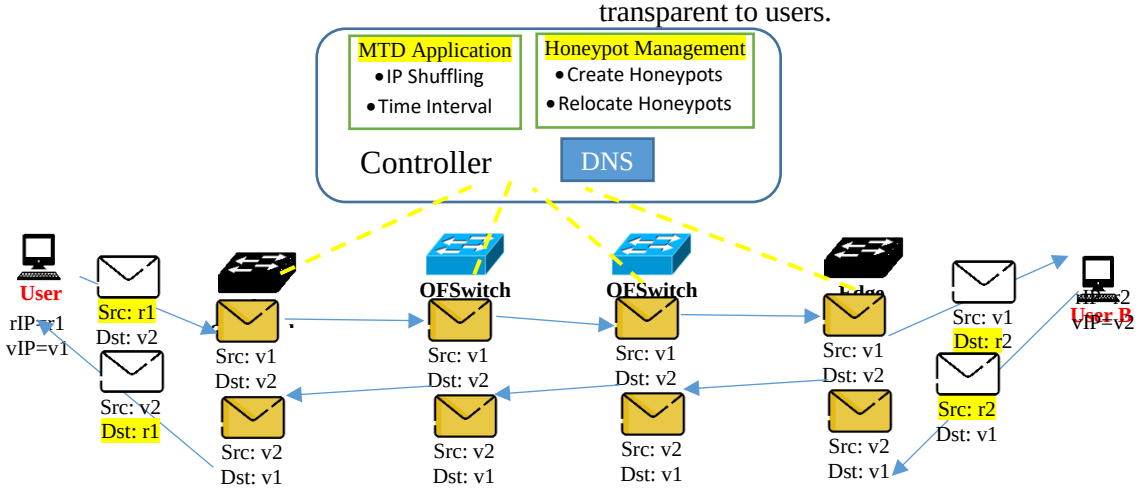


**Fig (3):** Proposed MTD Framework

At each interval period, a random vIP is associated with each rIP. The high unpredictability and mutation rate of virtual IP addresses is designed to maximize the defense of attackers' probing about the active hosts and network architecture, as well as raise the deterrence of attack planning. This is achieved by the short and random lifetime of virtual IP addresses. It is a good idea to make a variable short time to live for each host to create more complexity for attackers as well as to preserve active connections by making IP shuffling occur when the connection to the host is idle. After a specific time interval when shuffling occurs, the MTD module, through the SDN controller, dynamically changes the vIP, maps each host's real IP with a newly generated vIP, and updates the flow tables in each switch. The controller must make sure the active connection does not fail because of the shuffling process and update configurations in real-time while preserving network operation integrity. Our system provides a framework to implement effective MTD solutions while ensuring that they do not cause unexpected side effects because of conflicts with other flow rules present in the system. The system must make sure that no collision occurs in vIP assignment; i.e., a vIP must not be assigned to two or more hosts simultaneously. The MTD mechanism adds some operational delay due to the mapping and remapping of IP addresses (Sharma et al., 2018), as well as adding overhead on flow table updates in switches. This overhead is directly related to IP shuffling frequency, while it can be mitigated by optimizing the frequency of performing shuffling. However, the higher the frequency, the greater the attacker's uncertainty, while if the shuffling frequency decreases, the attacker's opportunity increases to successfully scan the network and find vulnerabilities. There must be a suitable trade-off between MTD accuracy and cost.

## 5. CONCLUSION AND FUTURE WORK

Despite honeypot capabilities of luring attackers and gathering intelligence about them through providing fake services, their static deployment remains an open challenge, which makes them vulnerable to detection easily by advanced attackers and bypassed. MTD addresses this limitation by dynamically altering network configurations, thereby disrupting attackers' efforts to gather accurate information. Integration of MTD with deception mechanisms, such as honeypots, within a SDN environment offers a promising approach to enhancing cybersecurity. After reviewing current works that implement MTD in SDN, we highlighted some requirements for implementing MTD applications in SDN environments. Following those requirements, a theoretical model was proposed to overcome this issue by using different MTD techniques. IP address shuffling was used, which proactively and reactively changed the IP addresses of both decoy and real nodes. Moreover, network shuffling MTD applied through creating and relocating various honeypot nodes in network that provide different services making honeypot deployment dynamic which increase the attacker effort and falsify previous information abstained on network hosts and services. Future work for this work will be implementing distrusted SDN controllers and dedicating one for MTD applications to reduce the load on the main controller, especially with large-scale networks. The shuffling timing is still an open challenge, and a lot of research work needs to consider the methodology of the selected optimal time interval. Moreover, some security services, such as access list services, will be disrupted because of IP shuffling, which may lead to system failure. Therefore, there is a need to implement solutions that stay consistent with the overall security policy. Finally, a unified standard of metrics is essential to measure the efficiency of MTD so that the research community can measure the effectiveness of their work compared with others.

## 6. REFERENCES


Belalis, I., Kavallieratos, G., Gkioulos, V., & Spathoulas, G. (n.d.). *Enabling Defensive Deception by Leveraging Software Defined Networks*.

Cho, J. H., & Ben-Asher, N. (2018). Cyber defense in breadth: Modeling and analysis of integrated defense systems. *Journal of Defense Modeling and Simulation*, *15*(2), 147–160. doi: 10.1177/1548512917699725

Chowdhary, A., Pisharody, S., & Huang, D. (2016). SDN based scalable MTD solution in cloud network. *MTD 2016 - Proceedings of the 2016 ACM Workshop on Moving Target Defense, Co-Located with CCS 2016*, 27–36. doi: 10.1145/2995272.2995274

Duy, P. T., Hoang, H. Do, Khoa, N. H., Thu Hien, D. T., & Pham, V. H. (2022). Fool Your Enemies: Enable Cyber Deception and Moving Target Defense for Intrusion Detection in SDN. *2022 21st International Symposium on Communications and Information Technologies, ISCIT 2022*, 27–32. doi: 10.1109/ISCIT55906.2022.9931208

Faraz Hyder, M., & Umer Farooq, M. (2021). Towards Countering the Insider

Reconnaissance Using a Combination of Shuffling and Diversity Moving Target Defense Techniques. In Technology & Applied Science Research (Vol. 11, Issue 6). Retrieved from www.etasr.com

Galadima, H., Seeam, A., & Ramsurrun, V. (2022). Cyber Deception against DDoS attack using Moving Target Defence Framework in SDN IOT-EDGE Networks. *2022 3rd International Conference on Next Generation Computing Applications (NextComp)*, 1–6. doi: 10.1109/NextComp55567.2022.9932172

Gao, C., Wang, Y., Xiong, X., & Zhao, W. (2021). MTDCD: An MTD Enhanced Cyber Deception Defense System. *IMCEC 2021 - IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference*, 1412–1417. doi: 10.1109/IMCEC51613.2021.9482133

Gao, Y., Zhang, G., & Xing, C. (2021). A Multiphase Dynamic Deployment Mechanism of Virtualized Honeypots Based on Intelligent Attack Path Prediction. *Security and Communication Networks*, *2021*. doi: 10.1155/2021/6378218

Ge, M., Cho, J. H., Kim, D., DIxit, G., & Chen, I. R. (2022). Proactive Defense for Internet-of-things: Moving Target Defense with Cyberdeception. *ACM Transactions on Internet Technology*, *22*(1). doi: 10.1145/3467021

Hyder, M. F., & Ismail, M. A. (2021). Securing Control and Data Planes from Reconnaissance Attacks Using Distributed Shadow Controllers, Reactive and Proactive Approaches. *IEEE Access*, *9*, 21881–21894. doi: 10.1109/ACCESS.2021.3055577

Ibrahim, H. Y., Ismael, P. M., Albabawat, A. A., & Al-Khalil, A. B. (2020). A Secure Mechanism to Prevent ARP Spoofing and ARP Broadcasting in SDN. *2020 International Conference on Computer Science and Software Engineering (CSASE)*, 13–19. doi: 10.1109/CSASE48920.2020.9142092

Jalowski, Ł., Zmuda, M., & Rawski, M. (2022). A Survey on Moving Target Defense for Networks: A Practical View. *Electronics (Switzerland)*, *11*(18). doi: 10.3390/electronics11182886

KHALID, H., ISMAEL, P., & AL-KHALIL, A. (2019). EFFICIENT MECHANISM FOR SECURING SOFTWARE DEFINED NETWORK AGAINST ARP SPOOFING ATTACK. *The Journal of the University of Duhok*, *22*(1), 124–131. doi: 10.26682/sjuod.2019.22.1.14

Luo, X., Yan, Q., Wang, M., & Huang, W. (2019). Using MTD and SDN-based Honeypots to Defend DDoS Attacks in IoT. *2019 Computing, Communications and IoT Applications, ComComAp 2019*, 392–395. doi: 10.1109/ComComAp46287.2019.9018775

Ryapukhin, A. V., Karpukhin, E. O., & Zhuikov, I. O. (2022). Method of Forming Various Configurations of Telecommunication System Using Moving Target Defense. *Inventions*, *7*(3). doi: 10.3390/inventions7030083

Sharma, D. P., Kim, D. S., Yoon, S., Lim, H., Cho, J.-H., & Moore, T. J. (2018). *FRVM: Flexible Random Virtual IP Multiplexing in Software-Defined Networks*. Retrieved from http://arxiv.org/abs/1807.09343

Steinberger, J., Kuhnert, B., Dietz, C., Ball, L., Sperotto, A., Baier, H., Pras, A., & Dreo, G. (2018). DDoS defense using MTD and SDN. *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 1–9. doi: 10.1109/NOMS.2018.8406221

Valdovinos, I. A., Pérez-Díaz, J. A., Choo, K. K. R., & Botero, J. F. (2021). Emerging DDoS attack detection and mitigation strategies in software-defined networks: Taxonomy, challenges and future directions. In Journal of Network and Computer Applications (Vol. 187). Academic Press. doi: 10.1016/j.jnca.2021.103093

Yousif, H., Khalid, I., Badie, N., & Aldabagh, I. (2024). A Survey on the Latest Intrusion Detection Datasets for Software Defined Networking Environments. *Technology & Applied Science Research*, *14*(2), 13190–13200. doi: 10.48084/etasr.6756

Yousif Khalid, H., & Badie Aldabagh, N. (2024). Exploring Honeypot as a Deception and Trigger Mechanism for Real-Time Attack Detection in Software-Defined Networking. *International Journal of Computing and Digital Systems*, *15*(1), 951–960. doi: 10.12785/ijcds/160169

Zhou, Y., Cheng, G., & Yu, S. (2021). An SDN-Enabled Proactive Defense Framework for DDoS Mitigation in IoT Networks. *IEEE Transactions on Information Forensics and Security*, *16*, 5366–5380. doi: 10.1109/TIFS.2021.3127009