2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC) | 978-1-7281-8535-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/IMCEC51613.2021.9482133

MTDCD: an MTD Enhanced Cyber Deception Defense System

Chungang Gao[1,2], Yongjie Wang[1,2], Xinli Xiong[1,2], Wendian Zhao[1,2]
1. College of Electronic Engineering, National University of Defense Technology, Anhui, China
2. Anhui Province Key Laboratory of Cyberspace Security Situation Awareness and Evaluation, Anhui, China
gangchungaog9432@nudt.edu.cn, wangyongjie17@nudt.edu.cn, xxlyx25@hotmail.com, zwd19@nudt.edu.cn
Corresponding Author: Yongjie Wang Email: wangyongjie17@nudt.edu.cn

Abstract—Advanced persistent threat (APT) attackers usually conduct a large number of network reconnaissance before a formal attack to discover exploitable vulnerabilities in the target network and system. The static configuration in traditional network systems provides a great advantage for adversaries to find network targets and launch attacks. To reduce the effectiveness of adversaries' continuous reconnaissance attacks, this paper develops a moving target defense (MTD) enhanced cyber deception defense system based on software-defined networks (SDN). The system uses virtual network topology to confuse the target network and system information collected by adversaries. Also Besides, it uses IP address randomization to increase the dynamics of network deception to enhance its defense effectiveness. Finally, we implemented the system prototype and evaluated it. In a configuration where the virtual network topology scale is three network segments, and the address conversion cycle is 30 seconds, this system delayed the adversaries' discovery of vulnerable hosts by an average of seven times, reducing the probability of adversaries successfully attacking vulnerable hosts by 83%. At the same time, the increased system overhead is basically within 10%.

Keywords—network reconnaissance attack; cyber deception; moving target defense; software-defined network

I. INTRODUCTION

Enterprises and governments integrate various information systems, data resources, Internet resources into an information management platform through portal websites, and establish information channels for external customers or internal personnel, to release various information stored internally or externally. For attackers, these portals become their entrance to the intranet. Traditional network defense technologies (such as intrusion detection, firewall, etc.) are static. Advanced persistent threat (APT) attackers can perform repeated vulnerability analysis and penetration testing on the inherent vulnerabilities of the target for a long time until they reach the final goal[1]. Therefore, security personnel began to focus on active defense technology, cyber deception was proposed as one of them.

Cyber deception is a defense mechanism evolved from the idea of honeypots. By deploying scams in one's network information system, interferes with the adversary's perception of one's network information system to achieve the purpose of detecting, delaying, or blocking the adversary's activities. However, the current network deception defense faces two problems. First, after deploying cyber deception, the network system configuration is still static, so sophisticated adversaries can still bypass the defense mechanism after careful detection and analysis. Second, the deployment of network deception is complicated in traditional networks.

To solve the first problem, we utilize the dynamic and random nature[2] of MTD[3] to enhance the defensive effectiveness of the network deception system. To solve the second problem, we utilize the flexibility and programmability of SDN to build an SDN-based network deception defense system.

Therefore, the goal of this paper is to mitigate the intrusion of APT attackers into the intranet by combining cyber deception and MTD to establish an MTD enhanced cyber deception defense system based on software-defined networks. First, the system uses the method of rewriting the data packet header to change the IP address of the host in the network and generate a large number of decoy nodes to construct a virtual network topology, so that the adversary spends more time on false resources, increasing the adversary's time cost. However, APT attackers can identify real hosts and decoy nodes based on the responses after interacting with nodes in the network. Therefore, this paper integrates IP randomization technology based on virtual network topology. Although the attacker can collect some information of the network system within a period of time, this information will become invalid after the deployment of IP randomization, so the attacker must re-probe the network. By integrating cyber deception with moving target defense, goals that they cannot achieve alone can be accomplished, effectively resisting continuous network reconnaissance attacks.

The rest of this paper is organized as follows. In the second section, we discuss the previous work related to the method in this paper. We construct the threat model in the third section, describe the detailed work of the dynamic cyber deception defense system in the fourth section, besides analyzing the effectiveness and system performance of the method in the fifth section. The sixth section summarizes the full text.

II. RELATED WORK

A. Cyber Deception

Akiyama et al.[4] proposed a cyber deception system based on honey marks. By setting user name, password, server address, or other information in the sandbox environment, when an attack is discovered, these honey marks will be transmitted to the attacker to log in to the preset Honeypot server. Achleitner et al.[5] assumed that the adversary has successfully entered the intranet and at least one host has been infected with some kind of malware. By deceiving the adversary's view, the information obtained by the adversary is confused, thereby prolonging the time for the adversary to identify the real host. However, the normal interaction between the two has caused a greater impact. Zhan et al.[6] discussed how to improve the effectiveness of cyber deception to reinforce FTP services to deal with APT attacks, instantiate a new FTP file system through logical constraints to ensure the consistency of spoofing, besides passing the Turing test to find that The probability of a participant identifying a deceptive environment is close to random guessing. Julian et al.[7] described a deception defense method that resides in the operating system. The core idea is to invalidate the malware that infects the target host by displaying fake I/O devices in the computer system. Rubio-Medrano et al.[8] proposed a highly interactive, scalable, and highly disguised industrial honeypot to collect malicious data samples for future analysis, which can effectively deceive attackers. It is identified as a real device by a variety of widely used detection tools.

B. Moving Target Defense

The moving target defense at the network layer changes network elements such as IP addresses or ports through diversified, dynamic, and randomized methods, which increases the difficulty for attackers to conduct network detection, network eavesdropping, or denial of service attacks[9]. At present, the framework of the moving target defense system at the network layer is mostly implemented based on SDN. The network layer moving target defense based on SDN uses the characteristics of SDN to optimize moving target defense technology. The highly programmable SDN can effectively increase the uncertainty and complexity of attacks or cost[10]. RHM[11] prevents and frustrates the attack process by analyzing and describing the behavior of the attacker and reconfiguring the network unpredictably and adaptively. The OpenFlow controller frequently assigns a random virtual IP to each host, while the real IP remains unchanged, so the IP mutation is completely transparent to the end host, ensuring the availability of the system. DESIR[12] deploys a large number of decoy nodes outside the server, besides uses IP randomization to dynamically change the IP addresses of the server and decoy nodes, also besides uses a seamless migration mechanism to ensure the transparency of the IP conversion to the host. The legitimate user communicates by initiating a service request to the authentication server to obtain the real-time IP address of the server. AEH-MTD[13] adopts a frequency hopping synchronization strategy. The client and the controller calibrate the local clock through the synchronization module and use the source address entropy and flow method to detect the network state. According to the detection result, the time adaptive and spatial adaptive methods are used to control the endpoints information, it is more dynamic and flexible than the previous method. To solve the single-point failure problem of the MTD system of a single SDN controller, Narantuya et al.[14] proposed an MTD architecture based on multiple SDN controllers to ensure the performance and safety of the MTD system in a large-scale network.

Both MTD and cyber deception have their advantages and disadvantages. Cyber deception can achieve the purpose of protecting important resources by attracting adversaries to attack the bait, but it lacks dynamics and uncertainty. It happens that MTD can make up for this shortcoming. However, it is not easy to combine these two technologies to better utilize their advantages. If they cannot be deployed reasonably, they may even be inferior to the defensive effect of a single technology. To our best knowledge, there is no research to achieve this goal. So how to better integrate MTD with cyber deception technology to maximize the effectiveness of defense is very important to research.

III. THREAT MODEL

Enterprises and governments integrate various information systems, data resources, and Internet resources into an information management platform through portals, which facilitates the interaction with external and internal personnel. However, portals have also become the entrance for attackers to enter the intranet.

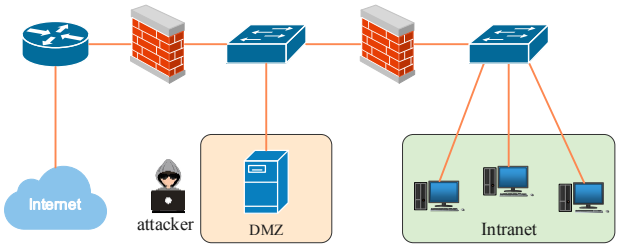

Fig. 1. threat model

As shown in Fig. 1., the portal website is deployed on a server in the DMZ. Both external and internal personnel can access the portal, but external personnel cannot directly access the internal network. The adversary has used the vulnerability on the website to obtain the administrator rights of the server where the website is located and uses this as a springboard to probe the intranet to prepare for further intrusion.

The attacker will adopt the most efficient strategy when conducting network reconnaissance. Generally, the attacker first scans the entire network segment to scan out all the surviving hosts in the network. It then further probes the surviving hosts, analyzes the possible vulnerabilities, and finally attacks the vulnerable hosts. However, this strategy is not suitable for network systems deployed with

IP randomization. When an attacker scans out all the surviving hosts in the network and prepares to further detect and exploit the surviving hosts, their IP addresses are likely to be out of date. Therefore, for a network system with IP randomization deployed, the adversary can only analyze and exploit vulnerabilities immediately after detecting a surviving host.

Through the above analysis, we put forward three assumptions about the threat model.

(1) The adversary has enough patience to invade the intranet;

(2) The adversary has robust vulnerability analysis and vulnerability exploitation capabilities, so it can successfully exploit every vulnerability found;

(3) When the adversary realizes that the IP randomization mechanism has been deployed, it will randomly detect the IP address and attack immediately after finding the vulnerability.

IV. SYSTEM ARCHITECTURE

This paper implements a dynamic cyber deception system based on SDN. Fig. 2. shows the system architecture, which consists of three main components, including the virtual network topology module, the IP randomization module, and the deception server.

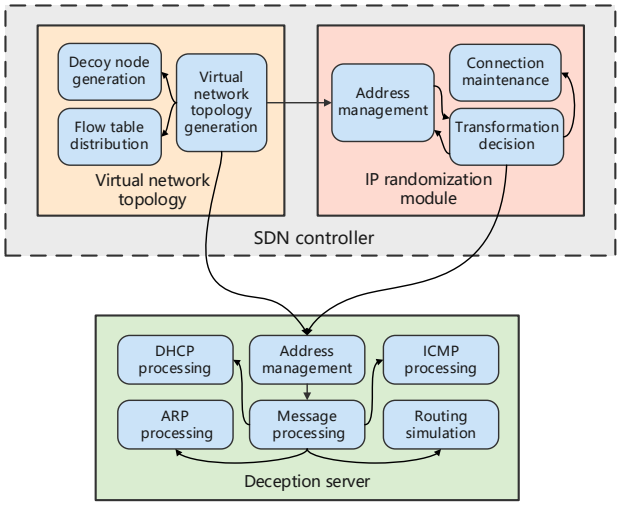

Fig. 2. system architecture

A. Virtual network topology module

The virtual network topology module is mainly responsible for generating the virtual network topology and distributing the flow table according to the specifications of the virtual network topology, including three sub-modules of the virtual network topology generation module, the decoy node generation module, and the flow table distribution module.

The virtual network topology generation module provides specifications for virtual network components and their connectivity, that is, describing the real and virtual address information of the host and decoy nodes, as well as the connectivity between them, including the real IP address, the virtual IP address, the real MAC address, the switch port, and the virtual path information.

The decoy node generation module generates a large number of decoy nodes according to the virtual network topology specification. The system uses one-to-many mapping to generate decoy nodes, assigns multiple virtual IPs to a honeypot host in the virtual network topology configuration file, and responds to scanning detection, the SDN controller reverses the virtual IP to its parent honey.

The flow table distribution module listens to the PackIn message from the switch, dynamically generates a specific flow table according to the specifications of the virtual network topology, and pushes it to the SDN switch to control network transmission.

Algorithm 1 gives the algorithm realization of the virtual network topology module. The virtual network topology module monitors PackIn messages from the switch in real-time, analyzes the source and destination addresses of the data packets as well as the type of the data packet, generates a specific flow table according to the virtual network topology specification, and adds flow table actions, including rewriting the packet header to complete the real IP and virtual IP conversion and designated forwarding outlets.

Algorithm 1 Virtual network topology algorithm

```
input: PackIn messages, virtual network topology
output: PackOut message
if PackIn.src = target.addr then
  if PackIn.type = ARP then
    OutFlow.action is outport =server.switchport
    InFlow.action is outport = ingressport
  end if
  if PackIn.type = IP then
    OutFlow.action is srcIP = target.rIP and dstIP = dst.rIP
    InFlow.action is srcIP = dst.vIP and dstIP = target.vIP
  end if
end if
else then
  if PackIn.type = ARP then
    OutFlow.action is outport = server.switchport
    InFlow.action is outport = ingressport
  end if
  if PackIn.type = IP then
    if PackIn.dst = target.rIP then
      OutFlow.action is srcIP = src.vIP and dstIP = target.vIP
      InFlow.action is srcIP = target.rIP and dstIP = src.rIP
    end if
    else then
      OutFlow.action is outport = dst_node.switchport
      InFlow.action is outport = src_node.switchport
  end if
```

B. IP randomization module

To further increase the defensive effectiveness of the cyber deception system, we have established the IP

randomization module. The IP randomization module is responsible for coordinating the address conversion between the host and the decoy node in the network. It includes three sub-modules: the address management module, the conversion decision module, and the connection maintenance module.

The address management module is responsible for real-time statistics of the hosts and decoy nodes in each subnet, the unused IP addresses in each subnet, as well as the allocation of IP addresses for the hosts and decoy nodes in the subnet to ensure that they will not interact with each other in the IP address allocation.

The transformation decision module is responsible for setting the period of IP address randomization and the construction method of the virtual network topology, including the size of the virtual network topology, the number of subnets, the number of decoy nodes in each subnet, and the location of the host in the network.

The connection maintenance module is responsible for the end-to-end connection will not be interrupted when the address change occurs to ensure that the address change is transparent to the user. The connection is maintained by setting the idle survival time of the flow table.

Algorithm 2 presents the enhanced algorithm implementation of the IP randomization module. First, determine the transformation period, the subnets in the network, the hosts and decoy nodes in each subnet, as well as the unused IP addresses in each subnet. During each change, for each node in each subnet, randomly select one of the unused IP addresses in the subnet as the virtual IP address of the node, and update the virtual network topology information.

Algorithm 2 IP randomization algorithm

```
input: T, subnetlist, hostdict, ipdict
output: ipdict, hostdict, virtual network topology
if time = T do
   for net in subnetlist do
      for host in hostdict[net] do
         ip = net + random(start,end)
         if ip in iplist do
            host.vIP = ip
            update ipdict
            update hostdict
            update virtual network topology
         end if
      end for
   end for
end if
```

C. Deception server

The deception server is responsible for making responses based on the specifications of the virtual network view to deceive malicious scanners, including the address management module, the message processing module, the DHCP processing module, the ARP processing module, the ICMP processing module, the routing simulation module.

The address management module is responsible for ensuring that the same virtual network topology specifications are maintained with the SDN controller. The message processing module is responsible for parsing the received data packet and sending it to the corresponding module for processing according to the type of the data packet.

DHCP processing module, the ARP processing module, the ICMP processing module, and the routing simulation module are responsible for deceptively responding to specific requests from malicious scanners. We take the routing simulation module as an example to introduce them to achieve deception.

(1) The malicious scanner uses *traceroute* to send detection packets to the node, and the TTL of the packets is incremented from 1.

(2) The virtual network topology module forwards the data packet to the deception server, and the message processing sub-module sends it to the routing simulation sub-module for processing.

(3) The virtual network topology specification describes the virtual routing information between two hosts in the network. The routing simulation module sends an *ICMP timeout message* to the source host according to the virtual routing information to prove that the data packet has passed the virtual routing.

(4) The routing simulation module generates a *port unreachable message* according to the virtual IP of the destination host to prove that the data packet has reached the destination.

Through the above process, the routing simulation module simulates the multi-hop path between two nodes.

V. CASE STUDY AND ANALYSIS

In this section, we will evaluate the effectiveness of the cyber deception system and the overhead it generates on the network. The evaluation experiment is carried out under the Ubuntu 20.04.1 operating system environment, the virtual machine memory is 4GB, the SDN-based network system is built using *mininet*, the SDN controller is the *POX* controller, and the switch is *Open vSwitch*.

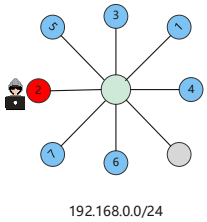

Fig. 3. real network topology

The real network topology is shown in Fig. 3. The red node is the host that has been occupied by the attacker, the blue node is six ordinary clients, the gray node is the honeypot, and the green node is the SDN switch. The network segment address is 192.168.0.0/24, and the common clients and honeypots in the network all have exploitable vulnerabilities.

TABLE I Parameters of the virtual network topology

The parameter name	Parameter	
Subnet number	3	
Number of real hosts per subnet	192.168.10.0/24	2
	192.168.11.0/24	2
	192.168.12.0/24	2
Number of decoy nodes per subnet	192.168.10.0/24	17
	192.168.11.0/24	18
	192.168.12.0/24	16

The parameters of the virtual network topology (VNT) are shown in TABLE I. The virtual network topology includes three subnets, The segment addresses are 192.168.10.0/24, 192.168.11.0/24, and 192.168.12.0/24. Common clients are evenly distributed in the address space.

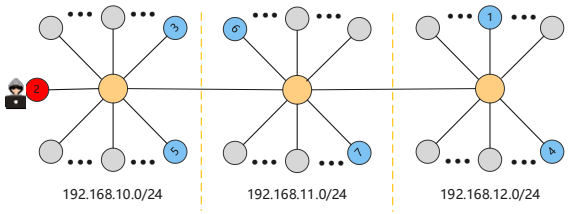

Fig. 4. virtual network topology

The deployed virtual network topology is shown in Fig. 4. The red node is the host that has been occupied by the attacker, the blue node is the ordinary client, the gray node and ellipsis are the honeypot, and the green is the virtual router. The structure of the virtual network topology is completely different from the real network topology.

TABLE II Parameters of IP address randomization

Type	The parameter name	Parameter
DVNT_30	Conversion period	30 seconds
	Conversion method	random
DVNT_10	Conversion period	10 seconds
	Conversion method	random

The IP address randomization parameters are shown in TABLE II. Two types of IP address enhanced randomization virtual network topologies are deployed in the experiment. They are a dynamic virtual network topology with an address conversion cycle of 30 seconds (DVNT_30) and a dynamic virtual network topology with an address conversion cycle of 10 seconds (DVNT_10).

A. Effectiveness

First, we evaluate the effectiveness of deploying virtual network topologies to extend the time for attackers to discover vulnerable hosts. The experiment simulates that the attacker uses Nmap to detect vulnerable hosts in the network. The scanning method is random scanning, and the IP addresses are evenly distributed, that is, each IP address is scanned only once, and a total of 100 rounds of scanning detection are performed.

TABLE III Time to scan out vulnerable hosts under different networks(seconds)

Network	Minimum time	Maximum time	Average time
NO VNT	15	155	44
VNT	15	1102	341

TABLE III shows the comparison of the time taken by the attacker to scan for vulnerable hosts under the network without the deployment of the virtual network topology (NO VNT) and VNT. In both cases, the shortest time to scan for vulnerable hosts is the same, but the longest and average time VNT is seven times that of NO VNT.

By deploying a virtual network topology, the time for the adversary to discover vulnerable hosts is greatly extended, and the time cost of the adversary is increased. But as long as there is sufficient time, the adversary can still compromise all the hosts in the network. Therefore, the address randomization is added based on the virtual network topology, and it is evaluated that the number of hosts attacked by the adversary changes over time under NO VNT, VNT, and DVNT_30. To be close to the real situation, suppose that the time from an attacker detecting a vulnerable host to launching an attack is a random time within 10 to 60 seconds.

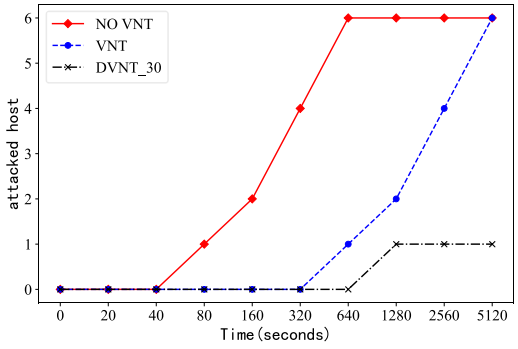

Fig. 5. The number of hosts successfully attacked under different networks varies with time

Fig. 5. shows the change in the number of hosts attacked by the attacker over time in NO VNT, VNT, and DVNT_30. It can be seen from the experimental results that the deployment of a virtual network topology prolongs the adversary's time to attack a vulnerable host by an average of 8 times. Because there are a large number of decoy nodes in the virtual network topology, adversaries waste more time analyzing and exploiting the vulnerabilities of decoy nodes. When a dynamic virtual network topology is deployed, the probability of an adversary attacking a vulnerable host is greatly reduced. This is because when an attacker exploits the vulnerability

of a vulnerable host, its virtual IP address is likely to be out of date. If the frequency of address change is fast enough, it is expected that the probability of an attacker's successful attack can be reduced to zero.

B. System overhead

Use *Netperf* to measure the impact of the system on network latency, and compare the performance under four different system configurations: NO VNT, VNT, DVNT_30, DVNT_10. Run the *Netperf* server on the target host, run the *Netperf* client on the client, use *TCP_RR* to simulate the client and server request/response mode, the client sends a query packet to the server, the server receives the request and returns the result data, and measures the network Delay, the data size ranged from 8KB to 1024KB, and ten sets of repeated experiments were carried out.

TABLE IV Network latency under different networks(us)

Network type	Transmission data size (KB)				
	8	32	128	512	1024
NO VNT	56.5	63.9	94.6	205.7	359.7
VNT	58.6	69.4	96.7	212.2	369.5
DVNT_30	59.8	70.6	98.0	212.3	370.6
DVNT_10	60.9	71.3	98.8	213.1	374.9

TABLE IV shows the comparison of network delays under different networks. Analysis of experimental results shows that the network latency of deploying virtual network topology has increased by 2.2% to 8.6%. The network delay of deploying dynamic virtual network topology with an address conversion period of 30 seconds has increased by 3% to 10.5%, the network delay of the dynamic virtual network topology with the address conversion cycle of 10 seconds increased by 3.6% to 11.6%. In a certain period time, the higher the frequency of address conversion, the more times the flow table is reinstalled, and the greater the additional time overhead caused.

VI. CONCLUSIONS

This paper proposes an MTD-enhanced dynamic cyber deception defense system to defend against intranet detection. The system's defense object is an attacker who has exploited the vulnerabilities of the portal website to invade a host in the intranet and penetrate the intranet horizontally. The virtual network topology that is completely different from the real network is displayed to the adversary, thereby increasing the adversary's time to detect the real vulnerable host. At the same time, the IP address of the node in the virtual network topology is dynamically changed to reduce the probability of the adversary's successful attack. Finally, a prototype cyber deception system is implemented based on *Mininet* and POX controllers, and the system is evaluated through comparative experiments. By deploying a virtual network topology, the time for an adversary to scan to a vulnerable host can be delayed by seven times. After deploying IP randomization, the probability of a host being successfully attacked is further reduced. In terms of system overhead, the deployment of a virtual network topology increases latency by 2.2% to 8.6%. Also besides, the network delay increases as the frequency of address changes increases. Experimental evaluation results show that the system can effectively defend against network reconnaissance attacks, and has acceptable system performance overhead.

REFERENCES

[1] K. D. Bowers, M. V. Dijk, R. Griffin, A. Juels, A. Oprea, R. L. Rivest, and N. Tri, "Defending against the unknown enemy: Applying flipit to system security," in International Conference on Decision and Game Theory for Security, 2012.

[2] J. Xu, P. Guo, M. Zhao, R. F. Erbacher, M. Zhu, and P. Liu, "Comparing different moving target defense techniques," ACM, pp. 97–107, 2014.

[3] NITRDSubcommittee. National cyber leap year summit 2009 co-chairs' report. 2009. https://www.nitrd.gov/nitrdgroups/index.php?title=Category:National_Cyber_Leap_Year_ Summit_2009.

[4] M. Akiyama, T. Yagi, T. Hariu, and Y. Kadobayashi, "Honeycirculator: distributing credential honeytoken for introspection of web-based attack cycle," International Journal of Information Security, 2018.

[5] S. Achleitner, T. L. Porta, P. Mcdaniel, S. Sugrim, and R. Chadha, "Cyber deception: Virtual networks to defend insider reconnaissance," in the 2016 International Workshop, 2016.

[6] S. Zhan and G. Yan, "Ensuring deception consistency for ftp services hardened against advanced persistent threats," in the 5th ACM Workshop, 2018.

[7] J. L. Rrushi, "Nic displays to thwart malware attacks mounted from within the os," Computers Security, vol. 61, pp. 59–71, 2016.

[8] E. López-Morales, C. Rubio-Medrano, A. Doupé, S. Yan, and G. J. Ahn, "Honeyplc: A next-generation honeypot for industrial control systems," in CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, 2020

[9] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati, "A survey of moving target defenses for network security," 2019.

[10] Y. Yang and L. Cheng, "An sdn-based mtd model," Concurrency and Computation: Practice and Experience, vol. 31, 2018.

[11] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "An effective address mutation approach for disrupting reconnaissance attacks," IEEE Transactions on Information Forensics Security, vol. 10, no. 12, pp. 2562–2577, 2015.

[12] J. Sun and K. Sun, "Desir: Decoy-enhanced seamless ip randomization," IEEE, 2016.

[13] Z. Liu, Y. He, W. Wang, S. Wang, X. Li and B. Zhang, "AEH-MTD: Adaptive Moving Target Defense Scheme for SDN," 2019 IEEE International Conference on Smart Internet of Things (SmartIoT), 2019.

[14] J. Narantuya, S. Yoon, H. Lim, J. H. Cho, and F. Nelson, "Sdn-based ip shuffling moving target defense with multiple sdn controllers," in 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks – Supplemental Volume (DSN-S), 2019.