Anti-reconnaissance Model of Host Fingerprint Based on Virtual Node

Tao Zhang\textsuperscript{1,2}, Bin Lu\textsuperscript{1,2}, Ding Li\textsuperscript{1,2}, Kang He\textsuperscript{1,2} and Yuefei Zhu\textsuperscript{1,2,*}

\textsuperscript{1} State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China
\textsuperscript{2} Information Engineering University, Zhengzhou, 450001, China

*Corresponding author: yfzhu17@sina.com

Abstract. Aiming at the problem of insufficient defense ability of fingerprint detection, the anti-reconnaissance model of host fingerprint based on virtual node is proposed. The model constructs periodically reconfigurable virtual nodes, dynamically camouflages the fingerprint information of the host to deceive the detector, and redirect attack traffic targeting virtual nodes to honeypots that can capture and analyze attack behavior. Honeypot, as an active defense technology, can effectively improve the model's defense capabilities. This paper introduces probabilistic models for the defense model to provide a deeper understanding of the theoretical effect their parameters have for cybersecurity, which quantifies the impact of different parameters on the probability of attack success, such as the number of probes, number of honeypot mapping rules, the virtual node deception rate, the honeypot detection rate and allowable losses. Furthermore, our prototype system using Software Defined Network (SDN) and Data Plane Development Kit (DPDK) verifies the effectiveness of the model against reconnaissance.

1. Introduction

With the development of network technology, attack methods are becoming more diverse which makes cybersecurity issues are increasingly prominent. Lockheed Martin has proposed a kill-chain model for network attacks, including reconnaissances, weaponization, delivery, utilization, deployment, command and control, completion. Reconnaissance is the first stage of most network attacks. The attacker needs to collect target information to discover vulnerability and available resources of the target, and then adopt a targeted attack method. Therefore, reconnaissance is an extremely important step in the kill-chain model, which accounts for more than 50% of attack time \cite{1}. Therefore, how to take defensive measures to prevent attacks during the detection phase is one of the research hotspots in the field of network security.

The host fingerprint is a collection of network information attracting the attention of attackers, which contains the characteristic information of the network services, the characteristic information of the operating system, etc. Fingerprint spoofing is a common technique for security. Researchers in \cite{2} proposed the IpMorph method to complete real-time traffic monitoring and data packet modification by implementing the user-mode TCP/IP protocol stack and deceive the prober. Using a game-theoretic approach, Rahman \cite{3} propose a selective and dynamic mechanism for counter-fingerprinting. Panos \cite{4} used SDN technology to forge operating system fingerprints to confuse attackers. This method only randomly modifies the fingerprint characteristics in the data packets, lacking a targeted
modification strategy. The above methods are based on the protection of operating system information by modifying the inbound and outbound traffic, and need to be deployed on each node in the target node. The deployment method is cumbersome and redundant, making it difficult to uniformly defend the entire network. Unprotected nodes are easily used by attackers as a foothold to compromise the target network.

Honeypot is a commonly used method to resist fingerprint detection [5]. Researchers in [6] addresses the problem of defending against attacks in honeypot-enabled networks by looking at a game-theoretic model of deception involving an attacker and a defender. HoneyMix, an SDN-based intelligent honeynet, leverages the rich programmability of SDN to circumvent attackers’ detection mechanisms and enables fine-grained data control for honeynet [7]. Fan proposed an SDN-based honeypot deployment method, which provided a high degree of programmability, and finally verified the feasibility and effectiveness of the method [8]. SDN-based honeypot technology enhances the flexibility of honeypot defense, but the above studies lack a formal analysis of defense effectiveness [9].

This paper proposes the anti-reconnaissance model based on virtual node in combination with the idea of moving target defense, generates dynamically changing virtual nodes within the target network to hide the real node fingerprints, and utilizes the rich programmability of SDN to redirect attack traffic to honeypots to reduce the probability of successful attacks. In Section 4, defense performance models are proposed, which is quantitatively evaluated by formal analysis of the probability of success. Then we analyze the impact of multiple parameters in performance models to provide a reference for how to improve the defense effect. Finally, we build a prototype system and verified the effectiveness and usability of the defense model.

2. Threat model
As shown in Figure 1, consider a simplified attack process. The attacker collects available resources with fingerprint detection. If the detection is successful, the attacker will take targeted measures to reach the goal. If not, the attacker will restart the detection. Defenders can adopt defense models to interfere with the attacker's acquisition of host fingerprints during the detection phase.

![Figure 1. A simplified process diagram for cyber attacks.](image)

External attackers obtain vulnerability resources by actively sending probes to fingerprint the target network and fingerprint detection can be divided into three steps [10].

- Host discovery: Attackers detects host IP address through ICMP protocol.
- Port scan: Attackers get the open ports set of the host through detection methods such as TCP CONNECT, TCP FIN, and TCP ACK.
- Operating system detection: By sending various types of probes including ICMP, TCP, UDP etc, the operating system type of the host is determined according to the different responses of the host's TCP / IP protocol stack to the probes.

3. Defense model
To express the model we make the following definition:

Definition 1 host fingerprints  \( F = \{ip, mac, ports, ostype\} \), where \( ip \) is the node's network address, \( mac \) is the node's physical address, the node's open port set is \( ports = \{port_1, port_2, ..., port_n\} \), and \( ostype \) is the node's operating system characteristic information.
Definition 2 Virtual node \( V = \{F_v, T\} \), \( F_v = \{ip_v, mac_v, ports_v, ostype_v\} \) is the host fingerprint information of the virtual node, and \( T \) is the reconfiguration period.

Definition 3 Honeypot mapping rules \( f : V \rightarrow H \) means that network traffic targeting the virtual node \( V \) will be redirected to the honeypot \( H \).

3.1. Virtual node defense model
The virtual node defense model (VNNDM) can hide the real node fingerprint and increase the attacker fingerprint detection address space by generating periodically reconfigurable virtual nodes, thereby reducing the probability of successful attacks. When the virtual node receives attack probes, the model constructs the response message according to the virtual node's fingerprint information \( F_v \). To be clear, when the IP address of the generated virtual node is the same as the real node, the virtual node will preemptively answer the attacker's probes to hide the real node fingerprint. The basic workflow of the VNNDM is shown in Figure 2.

- Virtual node generation: Generate the virtual node \( V = \{F_v, T\} \) based on configuration. After a reconfiguration period, the virtual node is set to \( V' = \{F_v', T\} \).
- Host discovery: the virtual node receives ICMP probes of the attacker and generates a response message according to the IP address and MAC address.
- Port scan: the virtual node receives port scanning probes and matches it with \( ports_v \). If it hits, a port response packet will be constructed based on the matching port and IP address.
- Operating system detection: The virtual node receives operating system scanning probes including TCP, UDP, ICMP and other protocols and constructs response messages according to \( ostype_v \). After receiving the response message, the attacker combines open ports of the virtual node to determine the operating system.

3.2. Virtual node defense model
The VNNDM can only delay but not prevent fingerprint detection. Honeypot is essentially a network spoofing technique that can capture and terminate attacks. By deploying decoy nodes, sensitive information or services, the attacker is tempted to carry out the attack, thereby capturing and analyzing the attack behavior. In the traditional deployment method, the honeypot and the target node are often in the same address segment, which causes the attacker to directly discover the honeypot through fingerprint identification. To solve the above problems, we proposed the virtual node honeypot defense model (VNHDM), which sets the honeypot mapping rules to redirect the attack traffic targeting the virtual nodes to the honeypot. There are two advantages, one is that the honeypot will not be directly accessed and identified by attackers, and the other is that the honeypot can be mapped to multiple virtual nodes reducing the cost of honeypot deployment. It must be noted that, the model does not support configuring rules for virtual nodes with the same IP address as the real node, which will cause normal traffic to be redirected to the honeypot.

Based on SDN technology, we have achieved fine-grained control of traffic, including address translation between honeypots and virtual nodes, and redirection of attack traffic. The honeypot and the target network are on different subnets and the attacker can successfully connect to the honeypot only by accessing the virtual node. When the attacker connects to a virtual node, which has a honeypot mapping rule, we convert its destination address and destination MAC to the IP address and MAC address of the honeypot and forward it. Conversely, the destination address and destination MAC of the honeypot's response message will be modified to the virtual node's IP and MAC address.

The basic workflow of the VNHDM is shown in Figure 3. The former four stages of the VNHDM are the same as the VNNDM, and then the attack traffic targeting virtual nodes will be redirected to the honeypot according to the honeypot mapping rules, which can record and analyze the actions of attackers.
4. Defense model

A primary objective of this section is the development of models, specifically the probability that an attacker will be successful, which describes the performance of detection defenses. We assume that the attacker has two attack success scenarios. One is the foothold attack whose target is to establish a foothold in the target network. When the target host is protected by firewalls, intrusion detection systems, and other protective measures, external attackers cannot directly launch attacks. Therefore, the attacker needs to control a node in the same local network as the target host as a foothold to bypass the protective measures. The other is control attack whose goal is not to gain a foothold, but to control a certain threshold number of nodes in the target network before the attack is considered a success. Typical scenarios are the formation of botnets to expand nodes [11].

Combining the martingale model with the multi-dimensional hypergeometric distribution model, consider a target network consisting of real nodes, virtual nodes. Principle hypothesizes are as follow.

- There are \( m \) real nodes in the network. Attackers can detect and obtain the fingerprint of the host when no protective measures are taken.
- The number of virtual nodes in the network is \( v \) where \( h \) virtual nodes are configured with honeypot mapping rules.
- The size of the address space that attackers can perform fingerprint detection is \( N = \max\{v, m\} \).
- The virtual node deception rate \( a \) is the probability that the fingerprint of the virtual node successfully deceives the attacker;
- The honeypot detection rate \( b \) is the probability that the attack is successfully captured by the honeypot;
- When the attacker obtains the fingerprint of more than one real node, the foothold attack is considered successful.
- When the attacker obtains the fingerprint of more than 50% real nodes in the target network, it is considered that the control attack is achieved.

Combined with the urn model to abstract the above scenario [12], the target network can be regarded as an urn containing \( N \) colored balls, and different types of nodes are identified by colors. \( m \) red balls represent real nodes, \( h \) yellow balls represent virtual nodes configured with honeypot mapping rules, and the rest are green balls. The non-repeating detection of attackers are equivalent to continuously draw \( k \) balls in the urn, and calculating the probability of successful attacks according to the probability distribution of the ball draw result.

4.1. Non-Defense Model

Consider this situation where the target network did not take any defensive measures. An attacker can discover all nodes by traversing the network address space, and then probe to obtain the host...
fingerprint of the real nodes. Therefore, when the attacker makes \( k \) detections, it is equivalent to drawing \( k \) balls in an urn with only red balls. Let \( X_k \) be a random number for drawing \( x \) red balls on \( k \) draws from the urn.

\[
P(X_k = x) = \begin{cases} 
1(x = k) \\
0(x \neq k) 
\end{cases} 
\]  

(1)

The mathematical expectation of obtaining \( x \) real nodes fingerprint is \( E(X_k) = k \).

4.2. Virtual-node Defense Model

Considering the virtual node defense model, \( v \) virtual nodes are generated in the target network to protect the fingerprint of real nodes. Therefore, the size of the address space that an attacker can detect is \( v \). When an attacker probes a real node protected by a virtual node, there is a probability of being deceived to obtain a virtual host fingerprint. Combined with the hypergeometric distribution, the probability of obtaining fingerprints of real nodes for \( k \) detection is.

\[
P(X_k \geq 1) = 1 - P(X_k = 0) = 1 - \sum_{i=\max(0,k+m-v)}^{\min(k,m)} \frac{C_{v-m}^{k-i} C_m^i (1-a)^i a^{k-i}}{C_v^k} 
\]  

(2)

The mathematical expectation of obtaining the host fingerprint information of the real node is.

\[
E(X_k) = k \frac{m(1-a)}{v} 
\]  

(3)

Given Equation 2, calculating the probabilities of the attacker success requires simple probabilistic manipulation. As Equation 4 shown, the foothold attack is considered successful to obtain more than one real node's fingerprint. As shown in Equation 5, the success of the control attack needs to detect more than half of the real node fingerprints.

\[
P(X_k \geq \frac{m+1}{2}) = \sum_{i=\frac{m+1}{2}}^{m} \sum_{j=\max(0,k+m-v)}^{\min(k,m)} \frac{C_{v-m}^{k-i} C_m^i C_v^j (1-a)^i a^{k-i}}{C_v^k} 
\]  

(5)

4.3. Virtual Node Honeypot Defense Model

By setting honeypot mapping rules, decoy nodes can be generated in the target network to reduce the probability of attackers obtaining the fingerprint of real nodes. When the attacker attempts to connect to a virtual node with a mapping rule, the attack will be captured and fails. In particular, the honeypot detection rate \( b \) will directly affect the attack success probability. If \( b = 0 \) then a honeypot will never find the attacker, and \( b = 1 \) means that a honeypot always catches the attacker. Combined with the urn model, it is necessary to consider the situation where red and yellow balls are drawn, which represent real nodes and virtual nodes with honeypot mapping rules, respectively. So consider using the multivariate hypergeometric distribution to calculate the attack success probability. Let \( X_k, Y_k \) be random numbers that follow the multivariate hypergeometric distribution for drawing \( x \) red balls and \( y \) yellow balls on \( k \) draws from the urn.

\[
P(X_k = x, Y_k = 0) = \sum_{i=\max(0,k+m-v)}^{\min(k,m)} \sum_{j=\max(0,k+m-h-v-i)}^{\min(k-m,h-k-i)} \frac{C_{v-m}^{k-i-j} C_m^i C_v^j (1-a)^i a^j C_h^k (1-b)^j}{C_v^k} 
\]  

(6)
The success probability of a foothold attack is the probability of drawing more than one red ball and no yellow ball for \( k \) draws which can be easily calculated using Equation 8. As shown in Equation 9, the probability of drawing more than half of the red ball without drawing the yellow ball for \( k \) draws can indicate the success probability of the control attack.

\[
P(X_k \geq 1, Y_k = 0) = P(Y_k = 0) = P(X_k = 0, Y_k = 0)
\]

\[
P(X_k \geq \left\lfloor \frac{m + 1}{2} \right\rfloor, Y_k = 0) = \sum_{x=\left\lfloor \frac{m + 1}{2} \right\rfloor}^{m} P(X_k = x, Y_k = 0)
\]

### 4.3.1. Allowable loss

According to the above description, the attacker is considered to have failed after being discovered by the honeypot. However, experienced attackers can accept the loss of multiple attack nodes. In other words, in a complete detection process, attackers can use multiple attack nodes for detection and allow \( Y \) attack nodes to be captured by the honeypot. In case of allowable loss, the success probability of the foothold attack is the probability of drawing more than one red ball and no more than \( Y \) yellow balls for \( k \) draws which can be easily calculated using Equation 10. As shown in Equation 11, the probability of drawing more than half of the red ball and no more than \( Y \) yellow balls for \( k \) draws can indicate the success probability of the control attack.

\[
P(X_k = x, Y_k = y) = \sum_{i=\max\{x,k+m-y\}}^{\min\{k,m\}} \sum_{j=\max\{x+m+k-h-v-i\}}^{\min\{h,k-i\}} \frac{C_{v}^{k-i-j} C_{m}^{i} C_{i}^{x} (1-a)^{i-x} a^{x} C_{h}^{j} C_{j}^{y} (1-b)^{j-y} b^{y}}{C_{v}^{k}}
\]

\[
P(X_k \geq 1, Y_k \leq Y) = \sum_{x=1}^{m} \sum_{y=0}^{Y} P(X_k = x, Y_k = y)
\]

\[
P(X_k \geq \left\lfloor \frac{m + 1}{2} \right\rfloor, Y_k \leq Y) = \sum_{x=\left\lfloor \frac{m + 1}{2} \right\rfloor}^{m} \sum_{y=0}^{Y} \sum_{i=0}^{\min\{k,m-h\}} \sum_{j=0}^{\min\{h,k-i\}} \frac{C_{v}^{k-i-j} C_{m}^{i} C_{i}^{x} (1-a)^{i-x} a^{x} C_{h}^{j} C_{j}^{y} (1-b)^{j-y} b^{y}}{C_{v}^{k}}
\]

### 5. Analysis of model

Section 4 establishes probability models of successful attack in the foothold attack and control attack scenarios to study the performance of defense models, which involves the parameters such as the number of probes, the virtual node deception rate, the honeypot detection rate, the number of honeypot mapping rules and allowable losses. This section aims to find the important factors that affect the performance of the defense model by analyzing the impact of the above parameters on the probability that the attacker will be successful.

#### 5.1. Number of probes

Consider the following scenario, there are 10 real nodes in the target network, 100 virtual nodes and 5 virtual nodes configured with honeypot mapping rules. The virtual node deception rate is 80%, and the honeypot detection rate is 90%. Figures 4 and figure 5 show the impact of the number of probes on the attack success rate in two attack scenarios. For foothold attack scenarios, as the number of probes increases, the probability of a successful attack increases even if VNNDM is adopted. As for the target network that adopts VNNDM, although the increase in the number of scans will lead to a higher chance of positioning and solid footing, it also increases the possibility of capture by the honeypot. Without considering allowable losses, 4 honeypot mapping rules are set up in the network, and the probability of successful attacks drops to less than 10%.
Figure 4. Relationship between the probability of successful attack and the number of probes of different models in the foothold attack scenario

Figure 5. Relationship between the probability of successful attack and the number of probes of different models in the control attack scenario

5.2. Number of honeypot mapping rules
The goal of foothold attacks is to successfully obtain fingerprints of more than one real node without being captured by any honeypots. Consider the following scenario, there are 10 real nodes in the target network, 100 virtual nodes, the virtual node deception rate is 80%, and the honeypot detection rate is 90%. As shown in Figure 6, we plot the relationship between the probability of successful attacks and the number of probes under different honey mapping rules. Even if only one rule is set, in the case of detecting half the address, compared with no rule, the probability of successful attacks is reduced by more than 20%. As the number of honeypot mapping rules increases, the probability of successful attack decreases. Therefore, setting rules can effectively protect the fingerprints of real nodes from being detected by attackers.

5.3. Allowable losses
In the previous discussion, we assumed that the attack failed if at least one honeypot captured the attacker’s detection. Consider a target network consisting of 10 real nodes and 100 virtual nodes, and the virtual node deception rate is 80%, the honeypot detection rate is 90%. The attacker can scan 50% of the address space. As shown in Figure 7, we draw a graph of allowable losses and the probability of successful attack under different numbers of honeypot mapping rules. The probability of successful attack increases as the allowable loss increases. Hence, the attacker who is willing to bear the loss of attack nodes has a high probability of successful attacks.

5.4. The virtual node deception rate
The virtual node deception rate has a great impact on the virtual node’s ability to protect fingerprints of real nodes. If the attacker traverses the entire address space, they can only obtain virtual fingerprints. As an example of the virtual node deception rate, consider a target network consisting of 10 real nodes and 100 virtual nodes. As shown in Figure 8, the higher the deception rate, the lower the probability of successful attack. Therefore, increasing the virtual node deception rate can effectively improve the model’s defense performance.

5.5. The honeypot detection rate
The honeypot detection rate is used to describe the ability of honeypots to capture attackers. As an example of the honeypot detection rate, consider a target network consisting of 10 real nodes and 100 virtual nodes whose deception rate is 90% and 5 virtual nodes configured with honeypot mapping rules. The probability of successful attacks as the honeypot detection rate increases is shown in Figure 9. We find that the higher the detection rate, the lower the probability of attack success under a given
number of probes from the plot. In the case where the attacker scans 20% of the address space, the probability of successful foothold attacks with \( b = 0.9 \) is reduced by about 3.5% compared with \( b = 0.7 \).

Figure 6. The effect of the number of honeypot mapping rules on the attack success probability

Figure 7. The effect of allowable losses on the attack success probability

Figure 8. The effect of virtual node deception rate on the attack success probability

Figure 9. The effect of honeypot detection rate on the attack success probability

6. Experiment and analysis

We designed and implemented the prototype system, which is implemented on the X86 platform using Intel’s DPDK framework. And then the experimental environment is built to give experimental results and analysis. Illustrated as figure 10, the IP address of the attack node is 172.14.96.181, running Windows 7 operating system, opening ports 22, 22, 3389. Host A runs CentOS 7 operating system, and the IP address is 172.14.96.177. The IP address of host B is 172.14.96.178, running Windows 10 operating system. We built a honeypot prototype system based on the KVM platform, deployed a highly simulated honeypot H which enabled SSH and Web services.

We generate 20 virtual nodes through the SDN switch, set their IP address, MAC address, operating system and open ports. The virtual node reconfiguration period is 5 minutes. Host A is protected by a virtual node but host B does not. In particular, we generated the virtual node C with an IP address of 172.14.96.179, ports of 22, 80, 3389, with an operating system kernel of Linux 3.10. Then, it is configured with honeypot mapping rules pointing to honeypot H. But beyond that, we do not reconfigure the virtual node C.
The attacker used the Nmap 7.0 to detect the fingerprint of the target network where hosts A and B are located. As shown in Table 1, to observe the change of the host fingerprint, we conducted a second detection at an interval of 7 minutes after the first detection. For the attacking node, the network distance is 0 hops, and the fingerprint is consistent with the above configuration result. Due to the lack of virtual node protection, the attacker accurately obtained the fingerprint of host B twice. In contrast, host A used a virtual node to protect its fingerprint. The attacker failed to obtain the accurate fingerprints in both detections, which proved the ability of virtual nodes to resist fingerprint detection. We notice that the MAC address, open ports, and operating system of the two probes of host A changed, which shows that the reconfiguration of virtual nodes greatly increases the uncertainty of the network environment.

Next, we simulated the attacker's actions by connecting to the ssh service on port 22 of virtual node C through a weak password attack. After successful login, we read and wrote some sensitive files. As shown in Figure 11, the attack was successfully captured by the honeypot recorded the file path, MD5 value and attack time, which proved that the system can effectively redirect the attack traffic from the virtual node to the honeypot which can analyze aggressive behavior.
7. Conclusion
In this paper, we proposed the anti-reconnaissance model of host fingerprint based on virtual-node and introduced a set of probabilistic modes that help defenders to take effective defensive measures. Generating periodically reconfigured virtual nodes within the target network can hide the fingerprints of real nodes and deceive attackers. Furthermore, combining virtual nodes with honeypots by setting honeypot mapping rules provides better anti-reconnaissance performance. According to our defense performance probability model in Section 4, even a small number of rules can greatly reduce the probability of successful attacks, whether it is the foothold attack scenario or the control attack scenario. Finally, we built a prototype system and verify the effectiveness and usability of the model through experiments.

References
[1] Zhuang R, DeLoach S A and Ou X 2014 Towards a Theory of Moving Target Defense Proceedings of the First ACM Workshop on Moving Target Defense - MTD ’14 the First ACM Workshop (Scottsdale, Arizona, USA: ACM Press) pp 31 – 40
[2] Prigent G, Vichot F and Harrouet F 2010 IpMorph: fingerprinting spoofing unification J Comput Virol 6 329 – 42
[3] Rahman M A, Manshaei M H and Al-Shaer E 2013 A game-theoretic approach for deceiving Remote Operating System Fingerprinting 2013 IEEE Conference on Communications and Network Security (CNS) 2013 IEEE Conference on Communications and Network Security (CNS) (National Harbor, MD, USA: IEEE) pp 73 – 81
[4] Kampanakis P, Perros H and Beyene T 2014 SDN-based solutions for Moving Target Defense network protection Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014 2014 IEEE 15th International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoWMoM) (Sydney, Australia: IEEE) pp 1 – 6
[5] Shi L, Li Y, Liu T, Liu J, Shan B and Chen H 2019 Dynamic Distributed Honeypot Based on Blockchain IEEE Access 7 72234 – 46
[6] La Q D, Quek T Q S, Lee J, Jin S and Zhu H 2016 Deceptive Attack and Defense Game in Honeynet-Enabled Networks for the Internet of Things IEEE Internet Things J. 3 1025 – 35
[7] Han W, Zhao Z, Doupé A and Ahn G-J 2016 HoneyMix: Toward SDN-based Intelligent Honeynet Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization - SDN-NFV Security ’16 the 2016 ACM International Workshop (New Orleans, Louisiana, USA: ACM Press) pp 1 – 6
[8] Fan W, Du Z, Smith-Creasey M and Fernandez D 2019 HoneyDOC: An Efficient Honeypot Architecture Enabling All-Round Design IEEE J. Select. Areas Commun. 37 683 – 97
[9] Bonfim M S, Dias K L and Fernandes S F L 2019 Integrated NFV/SDN Architectures: A Systematic Literature Review ACM Comput. Surv. 51 1 – 39
[10] Du M and Wang K 2020 An SDN-Enabled Pseudo-Honeypot Strategy for Distributed Denial of Service Attacks in Industrial Internet of Things IEEE Trans. Ind. Inf. 16 648 – 57
[11] Herwig S, Harvey K, Hughey G, Roberts R and Levin D 2019 Measurement and Analysis of Hajime, a Peer-to-peer IoT Botnet Proceedings 2019 Network and Distributed System Security Symposium Network and Distributed System Security Symposium (San Diego, CA: Internet Society)
[12] Crouse M, Prosser B and Fulp E W 2015 Probabilistic Performance Analysis of Moving Target and Deception Reconnaissance Defenses Proceedings of the Second ACM Workshop on Moving Target Defense - MTD ’15 the Second ACM Workshop (Denver, Colorado, USA: ACM Press) pp 21 – 9