electronics

MDPI

Article

A Port-Hopping Technology against Remote Attacks and Its Effectiveness Evaluation

Jiajun Yan, Ying Zhou * and Tao Wang

School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen 518107, China; yanjj3@mail2.sysu.edu.cn (J.Y.)
* Correspondence: zhouying5@mail.sysu.edu.cn

Abstract: Traditional network defense approaches are insufficient to deal with new types of network threats. Active defense approaches based on software-defined networks helps to solve this problem, which includes random port-hopping technology. Existing port-hopping approaches have problems such as the inability to completely hide the service port and the complicated hopping mechanism. What is more, there is no strict demonstration of the security effectiveness evaluation of random port hopping and its influencing factors. In this paper, a hidden services port-hopping approach and several models are proposed to solve these existing problems. Firstly, the algorithm, protocol , and flow update process of the method are presented. Secondly, according to the conceptual model of network attack and the network attack and defense model, the mathematical model of network attack is proposed to evaluate the security effectiveness of random port hopping. Furthermore, the resource layer and attack surface are redefined and the conceptual model of random port hopping is proposed to reveal the security mechanism of random port hopping more figuratively. After that, the factors that influence the security effectiveness of random port hopping are analyzed. Finally, both experiments and theoretical analysis show that hidden services port hopping is an effective active defense technology and the factors that influence the probability of a successful attack include the time interval of port hopping, the size of port-hopping space, and the number of vulnerable ports.

Keywords: port hopping; moving target defense; software-defined network; hidden services; security effectiveness evaluation

Citation: Yan, J.; Zhou, Y.; Wang, T. A Port-Hopping Technology against Remote Attacks and Its Effectiveness Evaluation. *Electronics* **2023**, *12*, 2477. https://doi.org/10.3390/electronics12112477

Academic Editors: Leandros Maglaras, Helge Janicke and Mohamed Amine Ferrag

Received: 11 April 2023
Revised: 19 May 2023
Accepted: 26 May 2023
Published: 31 May 2023

1. Introduction

With the rapid progress of information technology, the security protection capability of the network system is also improving [1,2]. However, various network intrusion events still emerge one after another. The fundamental reason is that the passive defense idea of traditional defense technology creates an attack and defense asymmetry between attackers and defenders. Firstly, there is asymmetry of time. Various configurations and attributes of the network system remain unchanged for a long time, and attackers have enough time to complete the entire process of the attack chain. Secondly, the asymmetry of space means that defenders need to protect the entire system comprehensively, while attackers only need to find and reasonably utilize a system vulnerability, which may defeat the entire network system. In addition, there is information asymmetry. Attackers can utilize a variety of means to obtain network information without being found, while defenders can hardly track the attacker's behavior, state, and purpose.

Traditional security defense methods (such as firewalls, antivirus software, intrusion detection equipment, etc.) are no longer sufficient to deal with various new security threats. Facing the defensive dilemma of traditional network security technology, moving target defense (MTD) adopts a proactive network security defense idea and is committed to building a dynamic, heterogeneous, and uncertain network to greatly increase the complexity and cost to the attacker [3–7]. From the perspective of the system level, MTD can be divided

into network layer MTD, system layer MTD, and application layer MTD. Port-hopping technology is a typical network layer MTD. By randomly changing the port numbers of both communication parties, attackers cannot accurately scan and attack the real ports, which is of great significance for improving the security protection capabilities of clients and servers.

In traditional networks, Lee et al. [8] proposed a port-hopping technology, which frequently changes the User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) ports according to the time and key shared by the server and the client. Luo et al. [9] proposed a TAP-based port and address-hopping technology, which realizes the hopping and synchronization of addresses and ports through the algorithm in the driver and the shared initial key. Chavez et al. [10] realized port randomization on each host in the network with the help of the netfilter kernel module. Badishi et al. [11,12] proposed a random port-hopping protocol that makes it difficult for attackers to efficiently craft packets that pass the filters, which can mitigate the adversaries' DoS attacks. It can be seen that the implementation of port-hopping technology in the traditional network can improve the system security, but there are also problems such as complex synchronization mechanisms, being affected by network delay, and easy interception and analysis of data packets in the synchronization process. The software-defined network (SDN) centralizes the control software of various devices to a unified computing resource, which can grasp the status of the entire network and finally make the best decision. Different from traditional networks, SDN has the characteristics of plane separation, centralized control, and programmable network. These characteristics provide natural convenience for solving the problems of port-hopping deployment in traditional networks.

Corresponding to the traditional security defense technology, the current mainstream network security effectiveness evaluation technology is still focused on the security evaluation of traditional static security defense systems, including evaluation methods based on vulnerability detection rules [13], mathematical evaluation methods based on Bayesian networks [14], and model evaluation methods based on attack graphs [15], etc. These network security evaluation technologies have their applicable values, but they are difficult to apply to dynamic and active network attack and defense games, and cannot effectively solve the problem of dynamic network defense security effectiveness evaluation. In the research field of dynamic network defense technology, the research on security effectiveness evaluation is still an important research field.

In this paper, firstly, the hidden services port-hopping approach (HSPH) is proposed based on SDN architecture to enhance the ability to resist remote attacks. HSPH uses the characteristics of a software-defined network (plane separation, centralized control, and network programming) to deploy a new port-hopping approach on the SDN controller, which realizes the synchronization of port hopping, is transparent to the terminal, and does not require additional hardware. At the same time, in order to hide the real port of the server, we access the server by using the port name. HSPH not only hides the real port on the transmission path, but also hides the real port from the source access node, which further improves the security of the network system. Secondly, this paper quantitatively analyzes the security effectiveness of random port hopping against attacks' entire attack processes. To start with, in order to evaluate the security effectiveness of random port hopping, the conceptual model of network attack and the network attack and defense model are proposed. Then, the security effectiveness is quantitatively analyzed from the perspective of the attacker's attack success probability in both static port and port-hopping cases. Furthermore, in order to reveal the security mechanism of random port hopping more figuratively, the attack surface and resource layer are redefined and the conceptual model of random port hopping is proposed. After that, the factors that influence the security effectiveness of random port hopping are analyzed. Thirdly, the proposed HSPH and theoretical analysis are verified by experiments.

In our previous work [16], we proposed a hidden services port- hopping approach to enhance network security. This work is an extension of our previous work with the following contributions:

(1) The update mechanism of flow entries is optimized to ensure the continuity of communication.
(2) The formula of network security effectiveness evaluation parameters of HSPH is derived through the mathematical model of network attack.
(3) The security defense mechanism of random port hopping is intuitively described through the proposed dynamic rope-threading model.
(4) The factors that influence the security effectiveness of random port hopping are obtained through quantitative analysis of network security effectiveness evaluation parameters.
(5) An experimental network of HSPH has been built and deployed to verify the correctness of the theoretical analysis.

The rest of this paper is organized as follows. Related works are discussed in Section 2. Section 3 describes the basic structure and communication protocol of HSPH. Section 4 describes the conceptual model and the mathematical model. Section 5 presents network deployment and simulation experiments. Section 6 concludes the paper.

2. Related Work

Hopping communication is a dynamic and random MTD technology, which aims to break the static configuration of traditional network attributes by changing the attack surface of the system, increasing the cost of attackers to obtain network configuration, and then improving the security of the system. Port-hopping technology is a kind of hopping communication. At present, researchers have proposed many different port hopping technologies. Hari et al. [17] developed an improved random port-hopping algorithm that can change the port number even if the communication between the sender and the receiver fails. Lin et al. [18] proposed a new distributed time stamp synchronization method and established a model to improve synchronization capabilities. Shi et al. [19] proposed a new synchronization scheme based on timestamps, and a prototype of port and address hopping was implemented. The methods described above have their advantages. However, all these methods bring about problems in the synchronization mechanism. Furthermore, some methods need to deploy software and add hardware devices at the terminal, which brings an additional overhead.

In the traditional network, the implementation of port-hopping technology has many problems caused by the synchronization mechanism. However, the appearance of SDN brings a new method for the implementation of port-hopping technology. Zhang et al. [20] proposed a timestamp-feedback-based hopping synchronization method, combining the timestamp synchronization method with the ACK synchronization method to achieve port-hopping synchronization. Sharma et al. [21] proposed an SDN-based random host and service multiplexing technology, which uses multiplexing (or demultiplexing) to dynamically change and remap from all the virtual IPs of the host to the real IP or the virtual ports of the services to the real port. Yoon et al. [22] proposed an asset-criticality-aware MTD technique that shuffles a host's network configurations (e.g., MAC/IP/port addresses) to enhance the security defense capability. Zhang et al. [23] proposed a transparent synchronization-based port mutation scheme in the SDN network, which changes communication ports along the transmission path. Luo et al. [24] developed a new random port and address-hopping mechanism, which can effectively defend against internal and external attacks and threats through source hopping, service hopping, and temporal hopping. However, the current hopping synchronization mechanisms [20] are difficult to deploy. Moreover, putting the port-hopping module on the host easily increases the overhead of the host and weakens the function of the controller [21]. Finally, only changing the ports along the transmission path [22–24] affects the security performance of the gateway, and it is easy for an attacker to intercept the real port of the server at the access gateway.

Random port hopping achieves the purpose of confusing attackers by dynamically mapping service ports to ports in the virtual port space, which is a typical moving target defense mechanism. The moving target defense mechanism improves the security of the system by dynamically changing the attack surface of the network. At present, many researchers study the mechanism of moving target defense based on the attack surface, but there is still no clear unified definition for the attack surface. Howard et al. [25] describe a system's attack surface as three abstract dimensions, including attack targets and enablers, channels and protocols, and access rights. Zhuang et al. [26] believed that the attack surface consists of system resources exposed to attackers and compromised network resources that can be used to further penetrate the system. Peng et al. [27] define the attack surface as the sum of externally accessible resources. Bopche [28] et al. define the attack surface as a subset of vulnerable connections and exploitable vulnerabilities. Badishi et al. [11] proposed a random port-hopping protocol and proved its security effectiveness under DOS attack. Hari et al. [17] analyzed the communication success rate of random port hopping under DOS attack through simulation experiments and proposed an improved RPH algorithm. Luo et al. [29] verified the effectiveness of random port hopping and its influencing factors under the condition of perfect port hopping by introducing the urn model. To sum up, the current definition of attack surface has the following deficiencies: (1) The attack surface is only placed in a resource set, and the relationship between resources cannot be described. (2) The description of the attack surface changes is not precise, comprehensive, and concrete enough. (3) There is no strict demonstration of the security effectiveness of random port hopping and its influencing factors. (4) The security effectiveness analysis of random port hopping is limited to specific offensive and defensive scenarios.

3. HSPH: Hidden Service Port Hopping

The architecture of the proposed approach is shown in Figure 1. Each client host and service host are connected to its corresponding OpenFlow switch. Each OpenFlow switch is connected to the controller which manages the whole network.

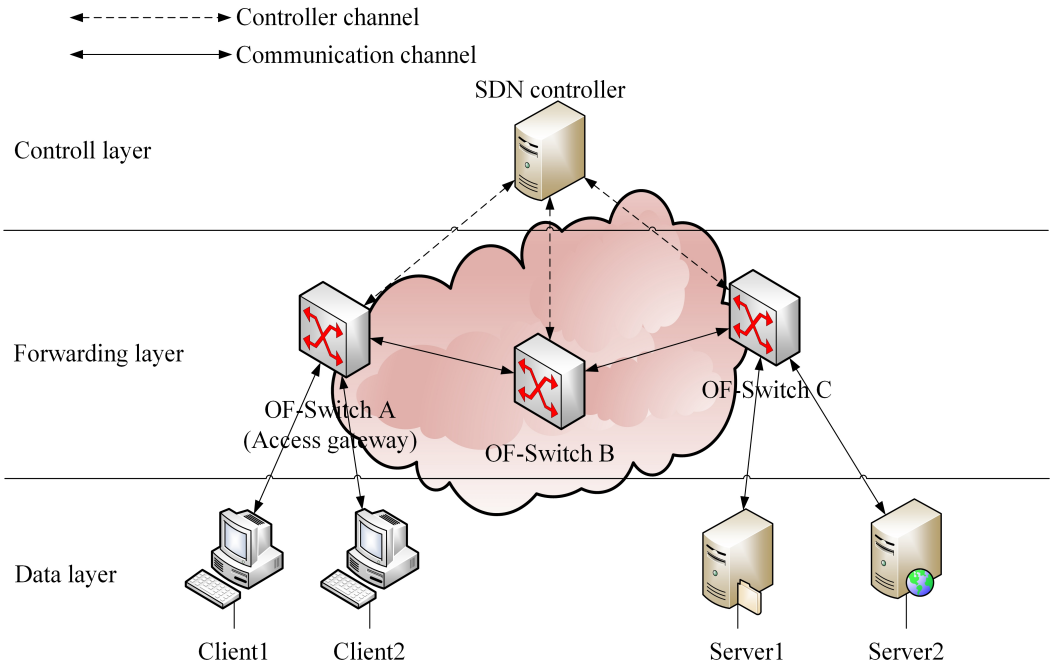

Figure 1. The architecture of HSPH.

As shown in Figure 1, HSPH is designed to be implemented in SDN networks and deployed in the SDN controller. The key modules of HSPH include the topology monitoring module, route calculating module, network address configuring module, port configuring

module, and route distributing module. The topology monitoring module is designed to monitor the real-time topology of the entire forwarding network. The route calculation module is designed to calculate the route of the entire forwarding network. The network address configuration module is designed to configure the IP address corresponding to each host. The port configuration module is designed to dynamically configure ports and establish the mapping relationship between port names and ports. The route distributing module is designed to deploy corresponding routes and related operation instructions of HSPH in the forwarding device for each data flow.

3.1. Port-Hopping Space

In the traditional network, the port number is divided into the port number used by the server (0∼49,151) and the port number used by the client (49,152∼65,535). The port numbers used by the server are divided into well-known port numbers (0∼1023) and registered port numbers (1024∼49,151). There is no correlation between the same port numbers in different computers. In a communication process, two computers need to establish a link relationship. $IC = (MAC_{src}, MAC_{dst}, IP_{src}, IP_{dst}, PORT_{src}, PORT_{dst})$ is defined as the connection relationship representing a communication process. Considering that the port pool $P = \{P_1, P_2, ...P_n\}$, $P_n \in random(0, 65535)$, the space of a connection relationship can be expressed as

$$S_{IC} = \{(MAC_{src}, MAC_{dst}, IP_{src}, IP_{dst}, PORT_{src}, PORT_{dst}) \mid PORT_{src}, PORT_{dst} \in P\} \quad (1)$$

3.2. HSPH Algorithm in SDN Controller

The proposed HSPH algorithm is presented in Algorithm 1, which is designed to be deployed in the POX controller. The NOX controller is the first industry's OpenFlow controller and the foundation of SDN research and development projects, while the POX controller is the Python implementation version of NOX, supporting the rapid development of controller prototype functions [30].

The key modules for evaluating computational complexity include the route calculating module, port configuring module, and route distributing module. The route calculating module uses the Floyd–Warshall algorithm with a computational complexity of $O(n^3)$. The algorithms for the port configuring module and route distributing module are shown in Algorithm 1 with a computational complexity of $O(n)$. Therefore, the overall computational complexity of our proposed techniques is $O(n^3)$.

At the beginning of the algorithm, five mapping tables are created: F_{pr}, which is used to map the port name of the server to the real port of the server; F_{ps}, which is used to map the real ports (rport) to the virtual ports (vport) for server at different times; F_{pc}, which is used to map the real ports to the virtual ports for the client at different times; T_{ie}, which is used to map the IP addresses to corresponding mac addresses; and T_{ip}, which is used to map the IP addresses to corresponding incoming ports of switches. Then, all packets entering the network will be processed by the OpenFlow switch and SDN controller according to Algorithm 1.

Algorithm 1 POX controller algorithm.

create port name to real port(rport) mapping table F_{pr}
create real port to virtual port (vport) mapping table for server F_{ps}
create real port to virtual port (vport) mapping table for client F_{pc}
create IP address to Ethernet address mapping table T_{ie}
create IP address to incoming port mapping table F_{ip}
for all packets p from OF-Switches **do**
  if p is a TCP or UDP packet **then**
    if $srcport(p)$ is in F_{pc} or F_{ps} **then**
      install a flow with
      action hwsrc(p) = Mac-OF-switch
      action $srcport(p) = F_{pc}(srcport(p))$ or $srcport(p) = F_{ps}(srcport(p))$
      action $output\_port = F_{ip}(dstIP(p))$
    end if
    if $dstport(p)$ is in F_{pr} or F_{pc} **then**
      install a flow with
      action $hwdst(p) = T_{ie}(dstIP(p))$
      action $dstport(p) = F_{ps}(F_{pr}(dstport(p)))$ or $dstport(p) = dstport(p)$
      action $output\_port = F_{ip}(dstIP(p))$
    end if
    if $dstport(p)$ is not in F_{pr} or F_{pc} **then**
      drop
      return
    end if
    if reach the hopping intervals T **then**
      generate new vport
      update F_{pc}, F_{ps}
      update the flow tables of OF-Switches
    end if
  end if
end for

3.3. HSPH Protocol

3.3.1. The Communication between Client and Server

Figure 2 shows the common handling process of a service request message. Client1 is the client and Server2 is the server. OF Switch A, OF Switch B, and OF Switch C are OpenFlow switches in the forwarding path. The MAC address and IP address of Client1 are m_1 and r_1, respectively. The MAC address and IP address of Server2 are m_2 and r_2, respectively. The MAC addresses of OF Switch A, OF Switch B, and OF Switch C are m_a, m_b, and m_c, respectively. The real port of an application service in Server2 is rp_2. The real port allocated when Client1 accesses an application service of Server2 is rp_1.

At some specific time, Client1 sends a service request message to query the service corresponding to Server2. Client1 sends an ARP packet to query the MAC address of Server2 (r_2). OF Switch A captures the ARP request message and sends it to the POX controller through the control channel. After receiving the ARP query packet, the SDN controller constructs an ARP response packet with m_a as the response and forwards it to Client1 from the corresponding port of OF Switch A. At the same time, Client 1 queries the port of Server 2 through the port name, OF Switch A forwards the query request to the SDN controller, and the SDN controller returns the virtual port (vp_2) of Server2 by looking up the F_{pr} table and the F_{ps} table. After Client1 receives the ARP response and virtual destination port(vp_2), it will encapsulate the request message with m_1 as the source mac address, m_a as the destination mac address, r_1 as the source IP address, r_2 as the destination IP address, rp_1 as the source port, and vp_2 as the destination port. It then sends the data flow to OF Switch A. Assume that the request data flow is $flow_{12}$, and the reply data flow is $flow_{21}$. After receiving $flow_{12}$, OF Switch A submits a processing request of $flow_{12}$ to

the SDN controller through the OpenFlow control channel. When receiving the processing request from OF Switch A, the SDN controller sends the relevant flow table entries to the OF Switch A, OF Switch B, and OF Switch C for processing the port hopping and routing of $flow_{12}$ and $flow_{21}$, respectively. After $flow_{12}$ passes through OF Switch A, the source real port address rp_1 is modified to the source virtual port address vp_1, the source MAC address m_1 is modified to m_a, and the destination MAC address m_a is modified to m_b; when $flow_{12}$ passes through OF Switch B, the source MAC address m_a is modified to m_b, and the destination MAC address m_b is modified to m_c; after $flow_{12}$ passes through OF Switch C, the destination virtual port address vp_2 is modified to the destination real port address rp_2, the source MAC address m_b is modified to m_c, and the destination MAC address m_c is modified to m_2.

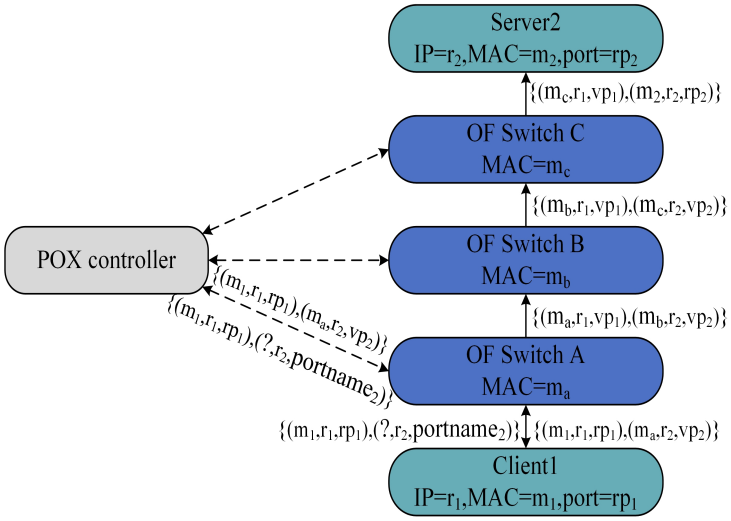

Figure 2. Handling process of a service request message.

Figure 3 shows the common handling process of a service reply message. After the Server2 receives the $flow_{12}$, it will respond to the corresponding $flow_{21}$. Server2 will encapsulate the reply message with m_2 as the source mac address, m_c as the destination mac address, r_2 as the source IP address, r_1 as the destination IP address, rp_2 as the source port, and vp_1 as the destination port, and then sends the data flow to OF Switch C according to the flow entry sent by the SDN controller. After $flow_{21}$ passes through OF Switch C, the source real port address rp_2 is modified to the source virtual port address vp_2, the source MAC address m_2 is modified to m_c, and the destination MAC address m_c is modified to m_b; after $flow_{21}$ passes through OF Switch B, the source MAC address m_c is modified to m_b, and the destination MAC address m_b is modified to m_a; after $flow_{21}$ passes through OF Switch A, the destination virtual port address vp_1 is modified to the destination real port address rp_1, source MAC address m_b is modified to m_a, and destination MAC address m_a is modified to m_1. So far, Client1 and Server2 have completed a request and reply process. If Client1 and Server2 continue to interact with data, the OF Switch will continue to handle the data flow using the flow entry previously sent by the SDN controller.

When the port-hopping interval T is reached, the flow table entries in the OF Switch, table F_{pc}, and table F_{ps} need to be updated. If Client1 wants to maintain communication with Server2, the process shown in Figures 2 and 3 must be repeated.

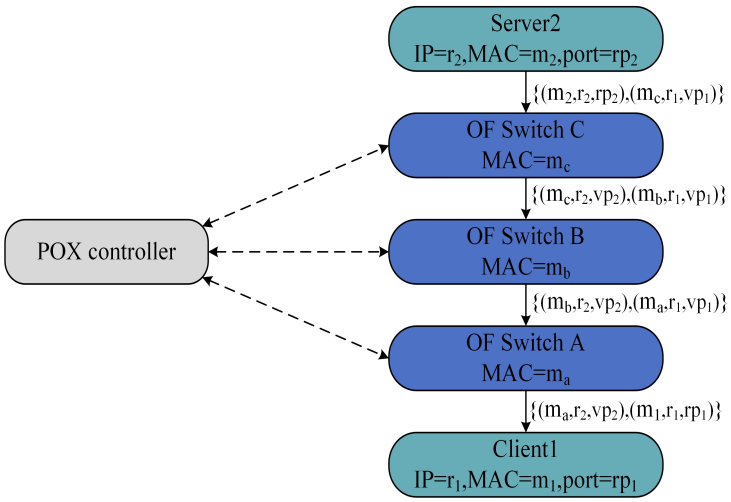

Figure 3. Handling process of a reply request message.

3.3.2. Flow Entries Update

In the communication process, SDN controls the transmission of data packets through the flow table. In order to ensure the continuity of data transmission and no loss of data packets in the process of port hopping, this paper adopts the flow table update mechanism of "sequential addition and delayed deletion". Taking Figure 1 as an example, the specific steps for the establishment and update of the flow table are as follows:

(1) When Client1 initiates communication with Server1, the controller sends flow tables along OF Switch A, OF Switch B, and OF Switch C according to the routing calculation results.
(2) When a hopping interval is reached, the controller sends a new flow table to OF Switch B and OF Switch C, and then sends a modification instruction to modify the old flow entries in OF Switch A as new flow entries.
(3) When the maximum communication delay between Client1 and Server1 is reached, the controller sends an instruction to delete the old flow entries in OF Switch B and OF Switch C in turn.

4. Model

4.1. Mathematical Model of Network Attack

4.1.1. Conceptual Model of Network Attack

As shown in Figure 4, the process of a remote network attack can generally be divided into four steps: reconnaissance, fingerprint, vulnerability analysis, and attack execution. The main task of reconnaissance is to detect the number and IP address of online hosts in the network. The main task of the fingerprint is to extract the fingerprinting information of the online hosts, including service port numbers. The main task of vulnerability analysis is to select specific attack targets based on the fingerprint of the host, analyze the vulnerability of the target system, and develop and test corresponding attack tools or methods. The main task of attack execution is to use the developed tools or methods to attack the target host.

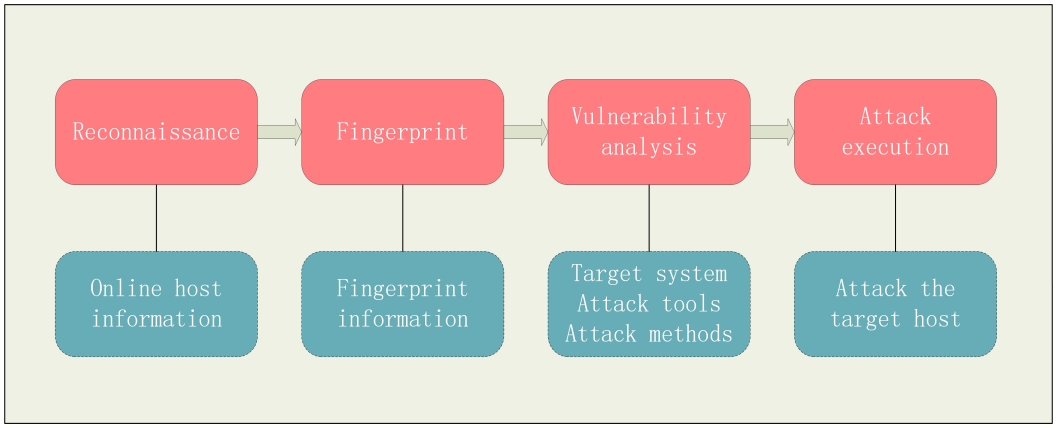

Figure 4. Conceptual model of network attack.

4.1.2. Network Attack and Defense Model

In order to compare the security effectiveness of HSPH and static port technology and derive the mathematical model of a network attack, we propose a network attack and defense model M-1.

M-1 includes a remote attacker A_1 and a target attack network N_1. N_1 includes routers, switches, hosts, and other network devices. The hosts can communicate with each other and are connected to the internet. In addition, N_1 has the conditions and capabilities to deploy traditional static port and random port hopping. The process of remote network attack against a host can be divided into four steps: reconnaissance, fingerprint, vulnerability analysis, and attack execution. A_1 has the conditions and capabilities to attack the vulnerable service ports in the host of N_1. In order to reduce the complexity of the analysis under the premise of ensuring rationality, it is assumed that the time required for the attacker to reconnaissance each address is equal, and the time required to extract fingerprints for each port is also equal. The parameters and descriptions included in the network attack and defense model are shown in Table 1.

Table 1. Parameters related to security effectiveness evaluation of random port hopping.

Parameter	Description
Q_h	The number of hosts in the network
Q_{hs}	The size of address space
Q_p	The number of open ports in the network
Q_{ps}	The size of the port-hopping space
Q_m	The number of tasks that an attacker can complete in one reconnaissance or fingerprint, $Q_m < Q_h Q_{ps}$
Q_v	The number of vulnerable service port in the whole network
T_n	The time interval of random port hopping
T_h	The time required to reconnaissance a single address
T_f	The time required to fingerprint a service port
T_z	The time required for vulnerability analysis
T_k	The time required for attack execution

4.1.3. Network Security Effectiveness Evaluation Parameters

According to the conceptual model of network attack proposed in Section 4.1.1 and the network attack and defense model M-1 proposed in Section 4.1.2, suppose P is the evaluation parameter of security effectiveness, which is the attack success probability of the attacker completing the entire attack process, which can be expressed as

$$P = \prod_{i=1}^{4} p_i \tag{2}$$

p_1, p_2, p_3, and p_4 are the attack success probability for the attacker to complete the four steps of reconnaissance, fingerprint, vulnerability analysis, and attack execution, respectively.

4.1.4. Analyzed Scenarios

The analysis scenarios of network security effectiveness evaluation include deploying traditional static port and random port hopping in N_1, respectively.

(a) Static Port

Reconnaissance: After the network system is deployed and configured, its network attributes will remain unchanged for a long period of time. Once the attacker has infinite time to reconnaissance the network, the target reconnaissance step must be successful, $p_1 = 1$.

Fingerprint: When the attacker has infinite time to finish the fingerprint steps, the number of fingerprint steps can be infinite. Assuming that the first $(n-1)$ times of fingerprint steps failed, then the failure probability of the nth fingerprint steps is $p(n)$, then

$$p(n) = \frac{\begin{pmatrix} Q_p - (\mathrm{n}-1)Q_m - Q_v \\ Q_m \end{pmatrix}}{\begin{pmatrix} Q_p - (\mathrm{n}-1)Q_m \\ Q_m \end{pmatrix}}, n \geq 1 \tag{3}$$

When $Q_p - (\mathrm{n}-1)Q_m - Q_v < Q_m$, the attacker will definitely identify the vulnerable port when performing the fingerprint steps, the failure probability of fingerprint is 0, and the success probability of fingerprint steps is $p_2 = 1 - p(n) = 1$.

Vulnerability analysis: When various network attributes in the network system remain unchanged for a long time, after completing the reconnaissance and fingerprinting steps, the attacker has sufficient time to perform the vulnerability analysis steps. Finally, the vulnerability analysis steps must succeed, $p_3 = 1$.

Attack execution: Similarly, when various network attributes in the network system remain unchanged for a long time, attackers can use various tools and methods developed in the vulnerability analysis steps to attack the target network. Finally, the attack execution steps must succeed, $p_4 = 1$.

In summary, when the traditional static port is deployed in N_1, the probability of the attacker's successful attack is $P = 1$.

(b) Random Port Hopping

When the random port hopping is deployed in the target network N_1, since the network service port number is randomly hopping, the attack success probability is affected by the time interval T_n of random port hopping. Assuming that T_r and T_e denote the time spent by the attacker on reconnaissance and fingerprint steps, respectively, then

$$T_r = \frac{Q_{hs}}{Q_m} \times T_h \times \eta, \eta \in [0, 1] \tag{4}$$

$$T_e = \frac{Q_h Q_{ps}}{Q_m} \times T_f \times \varepsilon, \varepsilon \in [0, 1] \tag{5}$$

ε and η are the time coefficient, which is affected by the length of the attack time, the time interval of random port hopping, and the number of vulnerabilities, etc.

(1) $T_n \leq Q_{hs}/Q_m \times T_h + T_f$

The attacker has completed the reconnaissance and fingerprint steps at most but cannot complete the vulnerability analysis and attack execution steps. The probability of the attacker's successful attack is $P = 0$.

(2) $Q_{hs}/Q_m \times T_h + T_f < T_n \leq Q_{hs}/Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f$

The attacker completed the reconnaissance steps but was not sure whether the fingerprint, target analysis, and attack execution steps could be completed.

Reconnaissance: The attacker has enough time to complete the reconnaissance of the entire target network; the reconnaissance steps must be successful, $p_1 = 1$.

Fingerprint: Assuming that the first $(n-1)$ times of the fingerprint step failed, the success probability of the nth fingerprint steps is $p(n)$, then

$$p(n) = \frac{\begin{pmatrix} Q_{ps}Q_h - (\mathrm{n}-1)Q_m - Q_v \\ Q_m \end{pmatrix}}{\begin{pmatrix} Q_{ps}Q_h - (\mathrm{n}-1)Q_m \\ Q_m \end{pmatrix}}, n \geq 1 \tag{6}$$

Assuming that the maximum number of fingerprint steps is N_{max} and the number of fingerprint steps conforms to the equal probability distribution, then the probability of each value of the number of fingerprint steps is $p(x) = 1/N_{max}$. Assuming that the mathematical expectation of the number of fingerprint steps is N, then

$$N = \sum_{x=1}^{N_{\max} = \left\lfloor (T_n - \frac{Q_{hs}}{Q_m} \times T_h)/T_f \right\rfloor} x p(x) \tag{7}$$

$$p_2 = 1 - \frac{\begin{pmatrix} Q_{ps}Q_h - (N-1)Q_m - Q_v \\ Q_m \end{pmatrix}}{\begin{pmatrix} Q_{ps}Q_h - (N-1)Q_m \\ Q_m \end{pmatrix}} \tag{8}$$

Vulnerability analysis and attack execution: Assuming that the time expectation of the target analysis and attack execution steps is T, then

$$T = T_n - \frac{Q_{hs}}{Q_m} \times T_h - \frac{Q_h Q_{ps}}{Q_m} \times T_f \times \varepsilon \tag{9}$$

Assuming that the success probability of both the vulnerability analysis and the attack execution steps are p_z, then

$$p_z = p_3 p_4 = \frac{T}{T_Z + T_K} = \frac{T_n - \frac{Q_{hs}}{Q_m} \times T_h - \frac{Q_h Q_{ps}}{Q_m} \times T_f \times \varepsilon}{T_Z + T_K} \tag{10}$$

In summary, the probability of the attacker's successful attack can be expressed as

$$\begin{aligned} P &= p_1 p_2 p_3 p_4 \\ &= \left(1 - \frac{\begin{pmatrix} Q_{ps}Q_h - (N-1)Q_m - Q_v \\ Q_m \end{pmatrix}}{\begin{pmatrix} Q_{ps}Q_h - (N-1)Q_m \\ Q_m \end{pmatrix}} \right) \left(\frac{T_n - \frac{Q_{hs}}{Q_m} \times T_h - T_e}{T_Z + T_K} \right) \end{aligned} \tag{11}$$

(3) $Q_{hs}/Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f < T_n \leq Q_{hs}/Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f + T_z$

The attacker completed the reconnaissance and fingerprint steps but was not sure whether the target analysis and attack execution steps could be completed.

Reconnaissance: The attacker has enough time to complete the reconnaissance of the entire target network; the reconnaissance steps must be successful, $p_1 = 1$.

Fingerprint: The attacker has enough time to complete the fingerprint of the entire target network; the fingerprint steps must be successful, $p_2 = 1$.

Vulnerability analysis and attack execution: Assuming that the time expectation of the target analysis and attack execution steps is T_m, then

$$T_m = T_n - \frac{Q_{hs}}{Q_m} \times T_h - \frac{Q_h Q_{ps}}{Q_m} \times T_f \tag{12}$$

Assuming that the success probability of both the vulnerability analysis and the attack execution steps are p_m, then

$$p_m = p_3 p_4 = \frac{T_m}{T_Z + T_K} = \frac{T_n - \frac{Q_{hs}}{Q_m} \times T_h - \frac{Q_h Q_{ps}}{Q_m} \times T_f}{T_Z + T_K} \tag{13}$$

In summary, the success probability of the attacker's attack can be expressed as

$$P = p_1 p_2 p_3 p_4 = \frac{T_n - \frac{Q_{hs}}{Q_m} \times T_h - \frac{Q_h Q_{ps}}{Q_m} \times T_f}{T_Z + T_K} \tag{14}$$

(4) $Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f + T_z < T_n < Q_{hs}/Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f + T_z + T_k$

The attacker completed the reconnaissance, fingerprint, and target analysis steps but was not sure whether the attack execution steps could be completed.

Reconnaissance: The attacker has enough time to complete the reconnaissance of the entire target network; the reconnaissance steps must be successful, $p_1 = 1$.

Fingerprint: The attacker has enough time to complete the fingerprint of the entire target network; the fingerprint steps must be successful, $p_2 = 1$.

Vulnerability analysis: The attacker has enough time to complete the vulnerability analysis steps; the vulnerability analysis steps must be successful, $p_3 = 1$.

Attack execution: Assuming that the time of the attack execution steps is T_4, then

$$T_4 = T_n - \left(\frac{Q_{hs}}{Q_m} \times T_h + \frac{Q_h Q_{ps}}{Q_m} \times T_f + T_z \right) \tag{15}$$

Assuming that the success probability of the attack execution steps is p_4, then

$$p_4 = \frac{T_4}{T_K} = \frac{T_n - \left(\frac{Q_{hs}}{Q_m} \times T_h + \frac{Q_h Q_{ps}}{Q_m} \times T_f + T_z \right)}{T_K} \tag{16}$$

In summary, the probability of the attacker's successful attack can be expressed as

$$P = p_1 p_2 p_3 p_4 = \frac{T_n - \left(\frac{Q_{hs}}{Q_m} \times T_h + \frac{Q_h Q_{ps}}{Q_m} \times T_f + T_z \right)}{T_K} \tag{17}$$

(5) $T_n \geq Q_{hs}/Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f + T_z + T_k$

It is equivalent to deploying a static port in the network, and the probability of the attacker's successful attack is $P = 1$.

4.2. Conceptual Model of Random Port Hopping

In order to solve the problems existing in the previous definition of the attack surface, we propose a new attack surface definition and a conceptual model associated with it, which realizes the characterization of the relationship between resources and the concrete description of the attack surface change.

Definition 1. *The resource layer consists of a certain system resource and its parameter set. System resources refer to one of the network resources that may be attacked, such as software, communication IP, and communication port numbers. The parameter set refers to software attributes, IP addresses, service ports, etc. corresponding to network resources.*

Considering the scalability of system resources in cyberspace as well as the disabling and opening of various resources in the process of network attack and defense, suppose a certain system resource with the possibility of attack is S_t. At the time t, a certain resource layer of the system can be expressed as

$$RL(t) = \{S_t, U_t\} \tag{18}$$

Suppose the parameter set corresponding to S_t is U_t, it can be expressed as

$$U_t = \{u_{t1}, u_{t2}, ..., u_{tk}\} \tag{19}$$

u_{tk} represents the kth parameter value corresponding to the resource parameter set U_t at time t.

Definition 2. *Attack surface refers to the set of parameters in a certain resource layer that has the possibility of attack and is a subset of the parameter set corresponding to a certain resource, such as software attribute vulnerabilities in the software resource layer, vulnerable IP addresses in the communication IP resource layer, and vulnerable service port numbers in the communication port number resource layer. Correspondingly, the attack surface of the system can be expressed as*

$$V_t = \{v_{t1}, v_{t2}, ..., v_{tn}\} \tag{20}$$

v_{tn} represents the nth attack possibility parameter value corresponding to the attack surface V_t at time t,$V_t \subseteq U_t$.

According to the definition of the resource layer and attack surface, in order to reveal the security mechanism of random port hopping more figuratively, we propose a conceptual model of random port hopping. The conceptual model we propose is called the dynamic rope-threading model (DRM), shown in Figure 5a. This model consists of blinds, hard panels, and holes in the hard panel, representing the attacker, resource layer (communication port numbers), and attack surface (vulnerable service port numbers), respectively. In order to make the dynamic rope-threading model more similar to the real network attack and defense process, it is assumed that the blind person cannot determine the position of the hole by touch. At the time T_1, when the blind person can use the rope to pass through the hole on the hard panel, it means the rope threading process is successful. This process means that an attacker with certain attack capabilities can find the vulnerable service port numbers in the port-hopping space and successfully launch an attack.

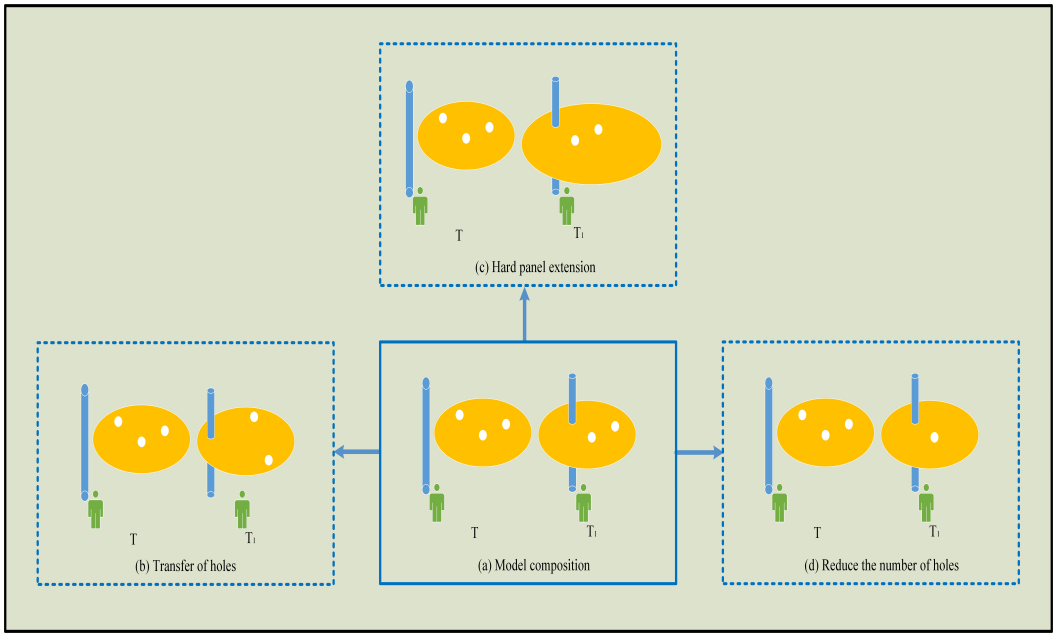

Figure 5. Dynamic Rope-threading Model.

Assuming that a blind person can only complete one rope-threading attempt per unit of time, in the DRM, measures to increase the difficulty of rope threading are as follows:

(1) Transfer of holes. As shown in Figure 5b, when other conditions remain unchanged, the transfer of holes on the hard panel will reduce the probability of successful rope threading in a unit of time. The speed of hole transfer affects the probability of successful rope threading. In particular, rope threading must fail when the holes are transferred on the hard panel faster than one rope threading attempt.
(2) Hard panel extension. As shown in Figure 5c, when other conditions remain unchanged, increasing the area of the hard panel reduces the probability of successful rope threading in a unit of time.
(3) Reduce the number of holes. As shown in Figure 5d, when other conditions remain unchanged, reducing the number of holes reduces the probability of successful rope threading in a unit of time.

Correspondingly, according to the above definition and discussion, the measures taken by the defender to increase the difficulty of the attacker's attack include expanding the port-hopping space, reducing the number of vulnerable service ports, and the time interval of random port hopping.

5. Deployment and Simulation Experiment

5.1. Simulation Experiment

In order to prove the feasibility and effectiveness of HSPH, we used mininet [31] and Open vSwitch (OVS) to establish an experimental network similar to Figure 1. The proposed algorithm is deployed on the POX controller. In terms of the scanning method, we chose the most common half-blind scanning strategy [11]. As a relatively similar port-hopping technology, TS-PM [23] is used as a comparison scheme of the experiment.

5.2. Effectiveness Analysis

According to the conceptual model of network attack and the network attack and defense model M-1, the mathematical model of a network attack is proposed, which can guide us to improve the defense effectiveness of random port hopping in practical applications. When HSPH is deployed in the network, combined with the discussion of the DRM in Section 4.2, it can be seen that the main influencing factors of the security

effectiveness evaluation include the time interval of random port hopping, the size of the port-hopping space and the number of vulnerable service ports.

5.2.1. The Time Interval of Random Port Hopping (T_n)

Random port hopping is deployed in the target network N_1. Assuming T_n is a variable, other parameters are quantitative. From the analysis in Section IV-C, it can be seen that

(1) $Q_{hs}/Q_m \times T_h + T_f < T_n \leq Q_{hs}/Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f$

The attacker completed the reconnaissance steps, $p_1 = 1$. From Formulas (6)–(8), with the decrease of T_n, the success probability of the fingerprint step will also decrease. In addition, according to Formulas (9) and (10), under the same attack and defense conditions (ε remains unchanged), as T_n decreases, the attacker will have less time to complete the target analysis and attack execution steps, so that the attack success rate of the target analysis and attack execution steps decreases. Finally, according to Formula (2), with the decrease of T_n, the probability of the attacker's successful attack also decreases.

(2) $Q_{hs}/Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f < T_n \leq Q_{hs}/Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f + T_z$

The attacker completed the reconnaissance and fingerprint steps, $p_1 = 1$ and $p_2 = 1$; Furthermore, according to Formulas (12) and (13), under the same attack and defense conditions ($\varepsilon = 1$), as T_n decreases, the attacker will have less time to complete the target analysis and attack execution steps, so that the attack success rate of target analysis and attack execution steps decreases. Finally, according to Formula (2), with the decrease in T_n, the probability of the attacker's successful attack also decreases.

(3) $Q_{hs}/Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f + T_z < T_n < Q_{hs}/Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f + T_z + T_k$

The attacker completed the reconnaissance, fingerprint, and target analysis steps, $p_1 = 1$, $p_2 = 1$, and $p_3 = 1$. Furthermore, according to Formulas (15) and (16), under the same attack and defense conditions ($\varepsilon = 1$), as T_n decreases, the attacker will have less time to complete the target analysis and attack execution steps, so that the attack success rate of target analysis and attack execution steps decreases. Finally, according to Formula (2), with the decrease in T_n, the probability of the attacker's successful attack also decreases.

To sum up, within a certain range, with the decrease in T_n, the probability of the attacker's successful attack also decreases.

In the experiment, we set $Q_{ps} = 30{,}000$ and $Q_v = 2$. The influence of T_n on the attack success probability is shown in Figure 6. On the one hand, in a static network, the probability of the attacker's successful attack reaches 1 after 152 s of an attack. On the other hand, when HSPH and TS-PM are deployed in the network, the probability of the attacker's successful attack will not exceed 0.22. Furthermore, during the experiment, we set T_n to 15 s, 20 s, and 25 s, respectively. As T_n decreases, the probability of the attacker's successful attack also decreases. The main reason is that as T_n decreases, the time for attackers to complete target analysis and attack execution steps will decrease. Ultimately, it will lead to a decrease in the probability of the attacker's successful attack. The experimental results are in good agreement with the theoretical analysis.

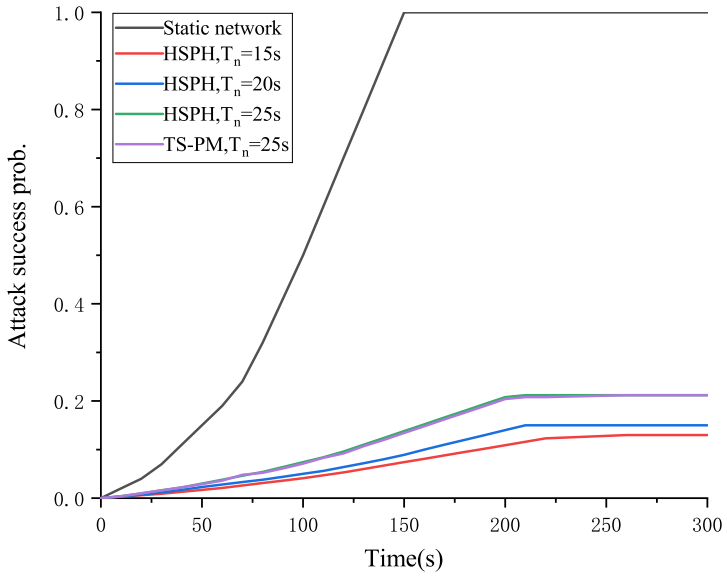

Figure 6. The influence of T_n on attack success probability.

5.2.2. The Size of the Port-Hopping Space (Q_{ps})

Random port hopping is deployed in the target network N_1. Assuming Q_{ps} is a variable, other parameters are quantitative. From the analysis in Section IV-C, it can be seen that

(1) $Q_{hs}/Q_m \times T_h + T_f < T_n \leq Q_{hs}/Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f$

The attacker completed the reconnaissance steps, $p_1 = 1$. From Formulas (6)–(8), with the increase of Q_{ps}, the success probability of the fingerprint step will decrease. In addition, according to Formulas (9) and (10), under the same attack and defense conditions (ε remains unchanged), with the increase of Q_{ps}, the attacker will spend more time in fingerprint steps, so that the attack success rate of target analysis and attack execution steps decreases. Finally, according to Formula (2), with the increase in Q_{ps}, the probability of the attacker's successful attack decreases.

(2) $Q_{hs}/Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f < T_n \leq Q_{hs}/Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f + T_z$

The attacker completed the reconnaissance and fingerprint steps, $p_1 = 1$ and $p_2 = 1$. Furthermore, according to Formulas (12) and (13), under the same attack and defense conditions ($\varepsilon = 1$), with the increase of Q_{ps}, the attacker will spend more time in fingerprint steps, so that the attack success rate of the target analysis and attack execution steps decreases. Finally, according to Formula (2), with the increase in Q_{ps}, the attacker's attack success probability decreases.

(3) $Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f + T_z < T_n < Q_{hs}/Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f + T_z + T_k$

The attacker completed the reconnaissance, fingerprint, and target analysis steps, $p_1 = 1$, $p_2 = 1$ and $p_3 = 1$. Furthermore, according to Formulas (15) and (16), under the same attack and defense conditions ($\varepsilon = 1$), with the increase of Q_{ps}, the attacker will spend more time in fingerprint steps, so that the attack success rate of target analysis and attack execution steps decreases. Finally, according to Formula (2), with the increase in Q_{ps}, the probability of the attacker's successful attack decreases.

To sum up, within a certain range, with the increase in Q_{ps}, the probability of the attacker's successful attack decreases.

In the experiment, we set T_n = 25 s and Q_v = 2. The influence of Q_{ps} on the attack success probability is shown in Figure 7. When HSPH and TS-PM are deployed in the network, the probability of the attacker's successful attack will not exceed 0.27. Furthermore, during the experiment, we set the Q_{ps} to 10,000, 20,000, and 30,000, respectively. With the increase in Q_{ps}, the probability of the attacker's successful attack decreases. The main reason is that as Q_{ps} decreases, attackers will spend more time on fingerprint steps, thereby reducing the

success rate of the target analysis and attack execution steps. Ultimately, it will lead to a decrease in the probability of the attacker's successful attack. The experimental results are in good agreement with the theoretical analysis.

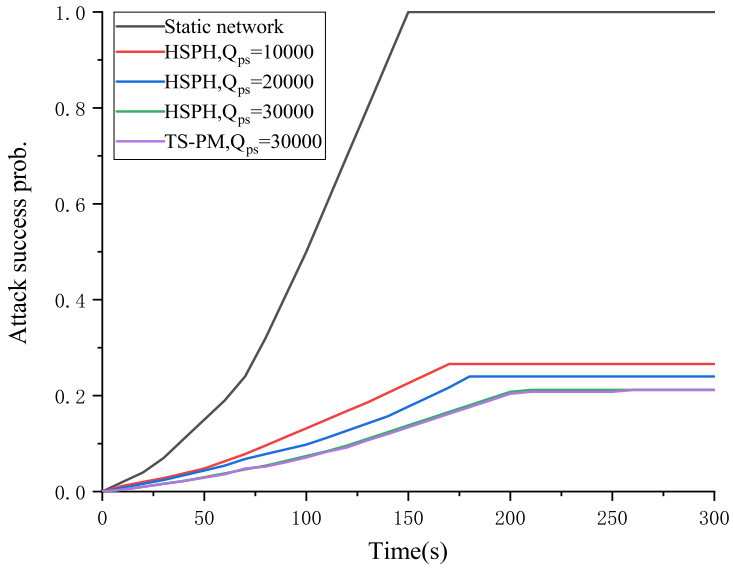

Figure 7. The influence of Q_{ps} on attack success probability.

5.2.3. The Number of Vulnerable Service Ports (Q_v)

Random port hopping is deployed in the target network N_1. Assuming Q_v is a variable, other parameters are quantitative. From the analysis in Section IV-D, it can be seen that

(1) $Q_{hs}/Q_m \times T_h + T_f < T_n \leq Q_{hs}/Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f$

The attacker completed the reconnaissance steps, $p_1 = 1$. From Formulas (6)–(8), with the decrease of Q_v, the success probability of fingerprint will decrease. In addition, according to Formulas (9) and (10), under the same attack and defense conditions (ε remains unchanged), with the decrease in Q_v, the attack success rate of the target analysis and attack execution steps unchanged. Finally, according to Formula (2), with the decrease in Q_v, the probability of the attacker's successful attack decreases.

(2) $Q_{hs}/Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f < T_n \leq Q_{hs}/Q_m \times T_h + Q_h Q_{ps}/Q_m \times T_f + T_z$

The attacker completed the reconnaissance and fingerprint steps, $p_1 = 1$ and $p_2 = 1$. Furthermore, according to Formulas (12) and (13), under the same attack and defense conditions ($\varepsilon = 1$), with the decrease in Q_v, the attack success rate of the target analysis and attack execution steps unchanged. Finally, according to Formula (2), with the decrease in Q_v, the attacker's attack success probability is unchanged.

To sum up, within a certain range, with the decrease in Q_v, the probability of the attacker's successful attack decreases.

In the experiment, we set T_n = 25 s and Q_{ps} = 30,000. The influence of Q_v on the attack success probability is shown in Figure 8. When HSPH and TS-PM are deployed in the network, the probability of the attacker's successful attack will not exceed 0.31. Furthermore, during the experiment, we set the Q_v to 2, 4, and 6, respectively. With the decrease in Q_v, the probability of the attacker's successful attack will decrease. This is because under the same conditions, as Q_v decreases, the probability of an attacker discovering a system vulnerability in the fingerprint steps decreases, and ultimately the probability of the attacker's successful attack decreases. The experimental results are in good agreement with the theoretical analysis.

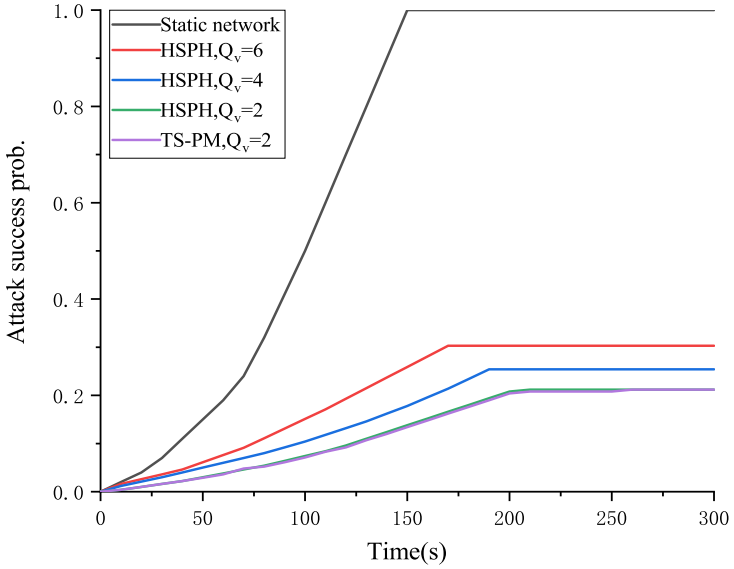

Figure 8. The influence of Q_v on attack success probability.

5.3. Performance Analysis

5.3.1. Evaluation of Data Transmission Time

We deploy static port technology, HSPH, and TS-PM, respectively, in the experimental network. In the experiment, the bandwidth of all connections is set to 10 Mbit/s. In the beginning, two hosts are randomly selected to run the iPerf program, one of which is in TCP server mode and the other is in customer service mode. Then, the customer sends 10 MB, 20 MB, 30 MB, 40 MB, 50 MB, and 60 MB files to the iPerf server in turn. The experimental results are shown in Figure 9. It can be seen that the data transmission time of HSPH is shorter than TS-PM, slightly longer than the static port, less than 0.4%. This is because compared to static port technology, the additional time cost of port hopping mainly includes updating the virtual port mapping table, updating the flow table entries, and processing port number changes. As the packet size increases, the transmission time also increases, and once the time interval for port hopping is reached, additional time overhead will be added. Compared to HSPH, TS-PM incurs more additional time overhead due to port hopping at each hop.

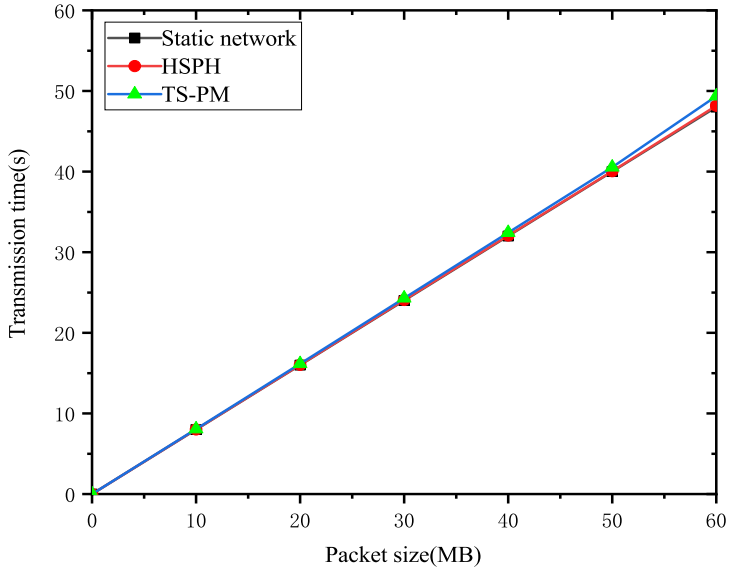

Figure 9. Performance of forwarding.

5.3.2. Overhead of CPU

In this section, we discuss the impact of different port-hopping technologies on CPU load. We used the same experimental steps as Section 5.3.1 to observe the changes in CPU load. The experimental results are shown in Figure 10. As the packet size increases, the transmission time will increase. When the time interval for port hopping is reached, compared to static port technology, updating the virtual port mapping tables, updating the flow table entries and processing port number changes can cause an increase in CPU load. Compared to HSPH, TS-PM incurs more CPU load due to port hopping at each hop. The experimental results show that compared with static ports, the CPU load increase in HSPH is less than 13%, while the increase in TS-PM can reach 47.8% at most. Combining Figures 6–8, HSPH can achieve similar defense performance as TS-PM with less overhead.

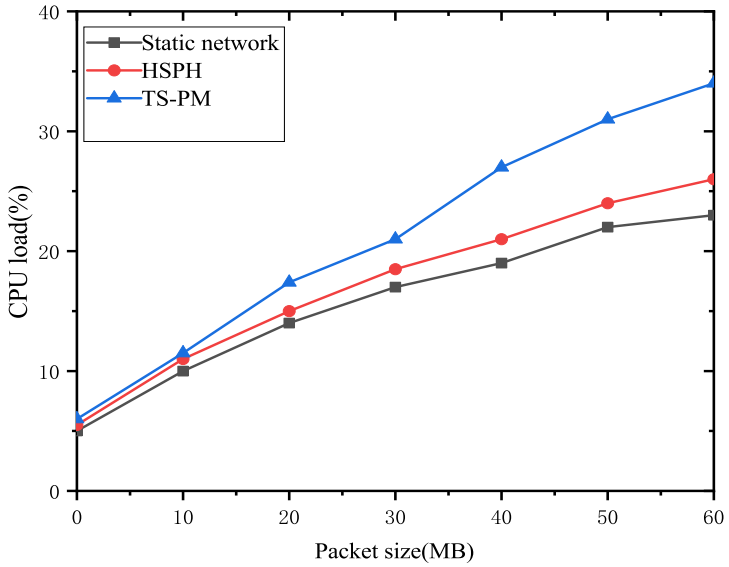

Figure 10. Performance of overhead.

6. Conclusions

In this paper, a hidden services port-hopping approach for moving target defense is proposed. HSPH effectively realizes the complete hiding of the real port of the server and the synchronization of port hopping. On this basis, in order to evaluate the security effectiveness of HSPH against remote attacks, the conceptual model and the mathematical model of network attack are proposed. Then, the resource layer and attack surface are redefined and the conceptual model of random port hopping is proposed, which reveals the security mechanism of random port hopping more figuratively. Finally, experimental and theoretical analysis results show that HSPH can be used to defend against regular remote attacks. Compared with static port technology, the increase in data transmission time of HSPH does not exceed 0.4%, and the increase in CPU load is less than 13%, both within an acceptable range and better than TS-PM. In actual deployment, the defender can reduce the probability of the attacker's successful attack by changing the time interval of port hopping, the size of the port-hopping space, and the number of vulnerable ports.

Author Contributions: Conceptualization, J.Y. and Y.Z.; methodology, J.Y. and Y.Z.; software, J.Y.; validation, J.Y., Y.Z. and T.W.; writing—original draft preparation, J.Y.; writing—review and editing, J.Y., Y.Z. and T.W.; supervision, Y.Z. and T.W.; project administration, J.Y., Y.Z. and T.W.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the Natural Science Foundation of Guangdong Province under grant number 2021A1515011910, and by the Shenzhen Science and Technology Program under Grant No.KQTD20190929172704911.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Anajemba, J.H.; Yue, T.; Iwendi, C.; Chatterjee, P.; Ngabo, D.; Alnumay, W.S. A Secure Multiuser Privacy Technique for Wireless IoT Networks Using Stochastic Privacy Optimization. *IEEE Internet Things J.* **2022**, *9*, 2566–2577. [CrossRef]
2. Anajemba, J.H.; Tang, Y.; Iwendi, C.; Ohwoekevwo, A.; Srivastava, G.; Jo, O. Realizing Efficient Security and Privacy in IoT Networks. *Sens. Rev.* **2020**, *20*, 2609. [CrossRef] [PubMed]
3. Lei, C.; Zhang, H.Q.; Tan, J.L.; Zhang, Y.C.; Liu, X.H. Moving target defense techniques: A survey. *Secur. Commun. Netw.* **2018**, *2018*, 3759626. [CrossRef]
4. Zheng, J.; Namin, A.S. A survey on the moving target defense strategies: An architectural perspective. *J. Comput. Sci. Technol.* **2019**, *34*, 207–233. [CrossRef]
5. Cai, G.L.; Wang, B.S.; Hu, W.; Wang, T.Z. Moving target defense: State of the art and characteristics. *Front. Inf. Technol. Electron. Eng.* **2016**, *17*, 1122–1153. [CrossRef]
6. Cho, J.H.; Sharma, D.P.; Alavizadeh, H.; Yoon, S.; Ben-Asher, N.; Moore, T.J.; Kim, D.S.; Lim, H.; Nelson, F.F. Toward proactive, adaptive defense: A survey on moving target defense. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 709–745. [CrossRef]
7. Sengupta, S.; Chowdhary, A.; Sabur, A.; Alshamrani, A.; Huang, D.; Kambhampati, S. A survey of moving target defenses for network security. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1909–1941. [CrossRef]
8. Lee, H.C.; Thing, V.L. Port hopping for resilient networks. In Proceedings of the IEEE 60th Vehicular Technology Conference, Los Angeles, CA, USA, 26–29 September 2004; VTC2004-Fall 2004; IEEE: Piscataway, NJ, USA, 2004; Volume 5, pp. 3291–3295.
9. Luo, Y.B.; Wang, B.S.; Wang, X.F.; Hu, X.F.; Cai, G.L. TPAH: A universal and multi-platform deployable port and address hopping mechanism. In Proceedings of the 2015 International Conference on Information and Communications Technologies (ICT 2015), Xi'an, China, 24–26 April 2015.
10. Chavez, A.R.; Stout, W.M.; Peisert, S. Techniques for the dynamic randomization of network attributes. In Proceedings of the 2015 International Carnahan Conference on Security Technology (ICCST), Taipei, Taiwan, 21–24 September 2015; IEEE: Pisvataway, NJ, USA, 2015; pp. 1–6.
11. Badishi, G.; Herzberg, A.; Keidar, I. Keeping denial-of-service attackers in the dark. *IEEE Trans. Dependable Secur. Comput.* **2007**, *4*, 191–204. [CrossRef]
12. Badishi, G.; Herzberg, A.; Keidar, I. Keeping denial-of-service attackers in the dark. In Proceedings of the Distributed Computing: 19th International Conference, DISC 2005, Cracow, Poland, 26–29 September 2005; Proceedings 19; Springer: Berlin, Germany, 2005; pp. 18–32.
13. Barrere, M.; Badonnel, R.; Festor, O. Vulnerability assessment in autonomic networks and services: A survey. *IEEE Commun. Surv. Tutor.* **2013**, *16*, 988–1004. [CrossRef]
14. Wang, J.; Fan, K.; Mo, W.; Xu, D. A method for information security risk assessment based on the dynamic bayesian network. In Proceedings of the 2016 International Conference on Networking and Network Applications (NaNA), Hakodate, Japan, 23–25 July 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 279–283.
15. Kumar, S.; Negi, A.; Prasad, K.; Mahanti, A. Evaluation of network risk using attack graph based security metrics. In Proceedings of the 2016 IEEE 14th International Conference on Dependable, Autonomic and Secure Computing, 14th International Conference on Pervasive Intelligence and Computing, 2nd International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), Auckland, New Zealand, 8–12 August 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 91–93.
16. Yan, J.; Zhou, Y.; Qin, G.; Wang, T.; Bin, R. A Hidden Services Port Hopping Approach for Moving Target Defense. In Proceedings of the ISCTT 2022 7th International Conference on Information Science, Computer Technology and Transportation, Xishuangbanna, China, 27–29 May 2022; pp. 1–5.
17. Hari, K.; Dohi, T. Sensitivity analysis of random port hopping. In Proceedings of the 2010 7th International Conference on Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing, Xi'an, China, 26–29 October 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 316–321.
18. Lin, K.; Jia, C.; Weng, C. Distributed timestamp synchronization for end hopping. *China Commun.* **2011**, *8*, 164–169.
19. Shi, L.; Jia, C.; Lü, S.; Liu, Z. Port and address hopping for active cyber-defense. In Proceedings of the Intelligence and Security Informatics: Pacific Asia Workshop, PAISI 2007, Chengdu, China, 11–12 April 2007; Springer: Berlin, Germany, 2007; pp. 295–300.
20. Zhang, L.; Guo, Y.; Yuwen, H.; Wang, Y. A port hopping based DoS mitigation scheme in SDN network. In Proceedings of the 2016 12th International Conference on Computational Intelligence and Security (CIS), Wuxi, China, 16–19 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 314–317.
21. Sharma, D.P.; Cho, J.H.; Moore, T.J.; Nelson, F.F.; Lim, H.; Kim, D.S. Random host and service multiplexing for moving target defense in software-defined networks. In Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
22. Yoon, S.; Cho, J.H.; Kim, D.S.; Moore, T.J.; Free-Nelson, F.; Lim, H. Attack graph-based moving target defense in software-defined networks. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 1653–1668. [CrossRef]

23. Zhang, L.; Wang, Z.; Gu, K.; Miao, F.; Guo, Y. Transparent synchronization based port mutation scheme in SDN network. In Proceedings of the 2016 5th International Conference on Computer Science and Network Technology (ICCSNT), Changchun, China, 10–11 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 581–585.
24. Luo, Y.B.; Wang, B.S.; Wang, X.F.; Hu, X.F.; Cai, G.L.; Sun, H. RPAH: Random port and address hopping for thwarting internal and external adversaries. In Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA, Helsinki, Finland, 20–22 August 2015; IEEE: Piscataway, NJ, USA, 2015; Volume 1, pp. 263–270.
25. Howard, M.; Pincus, J.; Wing, J.M. *Measuring Relative Attack Surfaces*; Springer: Berlin, Germany, 2005.
26. Zhuang, R.; Zhang, S.; DeLoach, S.A.; Ou, X.; Singhal, A. Simulation-based approaches to studying effectiveness of moving-target network defense. In Proceedings of the National Symposium on Moving Target Research, Annapolis, MD, USA, 11 June 2012; Citeseer: Princeton, NJ, USA, 2012; Volume 246.
27. Peng, W.; Li, F.; Huang, C.T.; Zou, X. A moving-target defense strategy for cloud-based services with heterogeneous and dynamic attack surfaces. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, NSW, Australia, 10–14 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 804–809.
28. Bopche, G.S.; Mehtre, B.M. Graph similarity metrics for assessing temporal changes in attack surface of dynamic networks. *Comput. Secur.* **2017**, *64*, 16–43. [CrossRef]
29. Luo, Y.B.; Wang, B.S.; Cai, G.L. Analysis of port hopping for proactive cyber defense. *Int. J. Secur. Its Appl.* **2015**, *9*, 123–134. [CrossRef]
30. Patel, R.; Patel, P.; Shah, P.; Patel, B.; Garg, D. Software Defined Network (SDN) Implementation with POX Controller. In Proceedings of the 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 20–22 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 65–70.
31. De Oliveira, R.L.S.; Schweitzer, C.M.; Shinoda, A.A.; Prete, L.R. Using mininet for emulation and prototyping software-defined networks. In Proceedings of the 2014 IEEE Colombian conference on communications and computing (COLCOM), Bogota, Colombia, 4–6 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–6.