Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Design and implementation of a novel enterprise network defense system by maneuvering multi-dimensional network properties*

Yang CHEN, Hong-chao HU†‡, Guo-zhen CHENG

National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China

†E-mail: 13633833568@139.com

Received Aug. 30, 2018; Revision accepted Nov. 11, 2018; Crosschecked Jan. 22, 2019

Abstract: Although the perimeter security model works well enough when all internal hosts are credible, it is becoming increasingly difficult to enforce as companies adopt mobile and cloud technologies, i.e., the rise of bring your own device (BYOD). It is observed that advanced targeted cyber-attacks usually follow a cyber kill chain; for instance, advanced targeted attacks often rely on network scanning techniques to gather information about potential targets. In response to this attack method, we propose a novel approach, i.e., an "isolating and dynamic" cyber defense, which cuts these potential chains to reduce the cumulative availability of the gathered information. First, we build a zero-trust network environment through network isolation, and then multiple network properties are maneuvered so that the host characteristics and locations needed to identify vulnerabilities cannot be located. Second, we propose a software-defined proactive cyber defense solution (SPD) for enterprise networks and design a general framework to strategically maneuver the IP address, network port, domain name, and path, while limiting the performance impact on the benign network user. Third, we implement our SPD proof-of-concept system over a software-defined network controller (OpenDaylight). Finally, we build an experimental platform to verify the system's ability to prevent scanning, eavesdropping, and denial-of-service attacks. The results suggest that our system can significantly reduce the availability of network reconnaissance scan information, block network eavesdropping, and sharply increase the cost of cyber-attacks.

Key words: Intranet defense; Software-defined network; Multi-dimensional maneuvering
https://doi.org/10.1631/FITEE.1800516 **CLC number:** TP393

1 Introduction

Enterprise networks deploy a large number of high-value resources that are attackers' targets for penetration. Since the early days of network technology, enterprises have used perimeter security to protect and gate access to their internal resources. The perimeter security model is often compared to a castle with thick walls, surrounded by a moat, with a heavily guarded single point of entry and exit. Anything outside the wall is considered dangerous, while anything inside the wall is trusted. In this model, anyone making it past the drawbridge achieves ready access to the resources of the castle.

The perimeter security model worked well enough in the early days, especially when all employees were working exclusively in buildings owned by the enterprise. However, as companies have adopted

‡ Corresponding author

* Project supported by the Information Engineering University Emerging Direction Cultivation Fund, China (No. 2016610708), the Science and Technology Research Project of Henan, China (No. 172102210615), the National Natural Science Foundation of China (Nos. 61521003 and 61602509), and the National Key Research and Development Program of China (Nos. 2016YFB0800100 and 2016YFB0800101)

ORCID: Yang CHEN, http://orcid.org/0000-0001-7806-2066

mobile and cloud technologies, a secure perimeter is becoming increasingly difficult to safeguard. On the one hand, the means of attack have become more complex, e.g., advanced persistent threats (APTs), social engineering attacks, and zero-day exploits. These attack behaviors cannot be detected effectively by traditional security detection systems. On the other hand, the boundaries of the enterprise network have become blurred, especially in recent years. With the development of the mobile Internet, "bring your own device" (BYOD) (Miller et al., 2012; Flores et al., 2016; Escobedo et al., 2017) has become a new model for enterprises, governments, and other workplaces. It enables enterprises to allow their employees' own devices to connect to the internal networks. While this model reduces enterprise costs, it exposes enterprise intranets to significant security risks. Attackers can easily penetrate the internal resources through these devices. Therefore, there is an urgent need to study internal cyber threats.

Recently, researchers have been exploring many countermeasures to defend the enterprise network. A zero-trust architecture is considered to be one of the promising approaches. The zero-trust network model was created by Kindervag (2010, 2016). The central idea is that an enterprise should not automatically trust any person/thing/object inside or outside the network. It should verify any person/thing/object that attempts to access the enterprise system before authorization. In short, a zero-trust strategy trusts no one unless the network determines the identity of the user. However, zero-trust migration entails equipping employees with a new mindset, and building the system requires considerable manpower and material resources. Google spent two years creating a trusted library for users and devices before migrating from traditional corporate networks to BeyondCorp (Peck et al., 2017); it was a relatively long process.

Generally, existing cyber-attacks can be divided into seven stages (Hutchins et al., 2011): (1) reconnaissance; (2) weaponization; (3) delivery; (4) exploitation; (5) installation; (6) command and control; (7) actions on objectives, called the cyber kill chain. Only after completing all the steps can an attacker succeed.

In this study, we aim to intercept and cut the cyber kill chain and sharply increase an adversary's cost of attack. We build a zero-trust network environment through network isolation, and then maneuver multiple network properties so that the host characteristics and locations for identifying vulnerabilities cannot be located. Then we design and implement a software-defined proactive defense system based on the isolation and dynamic maneuvering. This defense can enable a system to maintain the integrity of the original network configuration and minimize operation management. In the enterprise network, we hide the real IP address (rIP) of the terminal, and assign a virtual IP address (vIP) and a virtual domain name (vDomain) to the terminal to achieve isolation between users. In the case where the network application is not affected, the vIP performs high-frequency maneuvering, the vDomain performs low-frequency maneuvering, the transmission path randomly maneuvers, and the external network implements port maneuvering. Altogether, this approach constitutes a multi-dimensional maneuvering system.

The main contributions of this paper are shown as follows:

1. We propose a software-defined proactive cyber defense solution (SPD) for enterprise networks, and design a general framework to strategically maneuver the IP address, network port, domain name, and path, while limiting the performance impact on benign network users.

2. We design and implement our SPD proof-of-concept system over a software-defined network (SDN) controller (OpenDaylight), with 20 000+ lines of developed code.

3. We build an experimental platform to verify the system's ability to prevent scanning, and its anti-eavesdropping and anti-denial-of-service (DoS) capabilities. Results suggest that our system can significantly reduce the availability of network reconnaissance scan information, block network eavesdropping, and sharply increase the cost of cyber-attacks.

2 Related work

Google's BeyondCorp initiative (Peck et al., 2017) designed a practical zero-trust network security model where access depends solely on device and user credentials, regardless of a user's network location. Duo Security's Duo Beyond (Duo, 2018) assumes a zero-trust environment for all devices by default. By deploying Duo certificates to a company-

managed device, it can help companies create and maintain accurate device lists including any personal device.

However, in addition to the complexity of a zero-trust network deployment, these technologies require the installation of appropriate software on all terminals, which affects the user experience. We prefer a solution that does not affect the user when the cyber defense system is deployed.

A great number of studies in the literature focus on network address maneuvering techniques, (e.g., APOD (Atighetchi et al., 2003), DyNAT (Kewley et al., 2001), and NASR (Antonatos et al., 2007)); i.e., they dynamically change the targeted host address so that it cannot be easily located by adversaries. However, none of these approaches provides an appropriate address hopping method that defends against network scanning attacks from inside or outside without changing the host configuration. Thus, Al-Shaer E et al. proposed another network address maneuvering technology based on a strategy of high-frequency mutation to present network characteristics in as uncertain a way as possible; it is referred to as random host mutation (RHM) (Jafarian et al., 2015). After that, Al-Shaer E et al. further improved RHM by exploring the newly emerged OpenFlow technology and designed the OpenFlow random host mutation (OF-RHM) (Jafarian et al., 2012) model. OF-RHM can globally manage network host addresses and efficiently enforce random address maneuvering with limited overhead. However, during the dynamic change of address, the IP address range will be limited to its corresponding subnet. Sharma et al. (2018) proposed a moving target defense technology called flexible random virtual IP multiplexing (FRVM) using the SDN environment, which enables the hosts to have multiple, random, time-varying virtual IP addresses to invalidate the information on the target system collected by an attacker.

In the field of path maneuvering technology, Talipov et al. (2006) proposed a path maneuvering method based on the reverse ad-hoc on demand distance vector (R-ADOV). It can adaptively jump to the available path during packet transmission to protect the data from intrusion by malicious nodes. Jafarian et al. (2013) proposed a random route mutation (RRM) (Duan et al., 2013) method. The selectable forwarding path was calculated by a satisfiability model theory. Lei et al. (2017) proposed an optimal maneuvering path generation method based on the safety capacity matrix based on RRM, and it selected the optimal combination of maneuvering path and maneuvering period to maximize the defense revenue. Zhou et al. (2017) proposed the node-centric path maneuvering method, which abstracts path maneuvering into a signature matching problem, modeled into the three-dimensional (3D) Earth mover's distance (EMD) model, and solved it by a binary branch and bound method.

In short, the existing research has focused on establishing appropriate theoretical models to analyze and evaluate the security and introduced an overhead of network properties (IP address and transmission path) mutations, and there are related studies on controller security and data processing (Guan et al., 2017; Li et al., 2018; Wu et al., 2018). However, there are few studies focused on how to design and implement a practical defense system that can be deployed in a production network. In this study, we solve several key issues about proactive defense system in practice. On the premise of not changing a host system and based on the centralized control of an SDN, we design and implement a proactive defense system with multi-dimensional attribute cooperative maneuvering.

3 Design of the system architecture

As companies adopt mobile and cloud technologies, an increasing number of enterprise employees use their own portable equipment within the enterprise building, i.e., BYOD. A large number of private devices and company assets can be easily accessed in the internal enterprise under current perimeter security models. Thus, many companies are migrating their network to a zero-trust environment; i.e., any access device is not trusted. Inspired by the zero-trust model, we isolate the enterprise network by allocating each host with a temporary vIP whereby each has a different prefix; then we can randomly change the vIP to build a dynamic enterprise network. Only the mutual trust regions can communicate with each other through the host domain name.

3.1 Design principles

Our dynamic defense system can disturb the dependency of the target's static properties on the enterprise network and increase the adversaries' cost

of attack. As a novel defense system based on dynamic network technology, it is necessary to consider the compatibility with traditional network devices and to consider the management cost introduced by our system. Therefore, a discussion of some design principles and the philosophy of a dynamic defense system is called for:

Compatibility: The system should be compatible with traditional communication equipment. The goal of the system is not to defend against all attacks; instead, the system uses dynamic technology to increase the level of difficulty in attacking. So, the system needs to be compatible with traditional security devices.

Manageability: Administrators should be able to easily configure the network. Management should be simple and the additional management overhead should be as low as possible.

Distribution: Generally, enterprises are cross-regional, and dynamic defense devices have scenarios for cross-regional deployment, so unified network management should be supported.

Customization: Users need different security levels for different network environments. For these network requirements, the system can provide on-demand capabilities, e.g., maneuvering time and IP address segment selection.

Based on the above design principles, we require a device that can centrally manage the network to achieve general control of the enterprise network. An SDN provides a flexible infrastructure for developing and managing networks, and it has the minimum operating overhead, which provides the possibility for system implementation. Based on these attributes, we design a dynamic defense architecture based on a software definition.

3.2 General framework

The design includes a data plane, a management plane, and a control plane. As shown in Fig. 1, the first two are responsible for data forwarding and configuration management, respectively; the latter is the core control unit for enterprise network related attribute assignment and maneuvering.

Data plane: This plane enforces packet forwarding according to the dynamic transform rules, such as change src_ip before the received packet is steered. We have to reduce the latency overhead caused by network property modification.

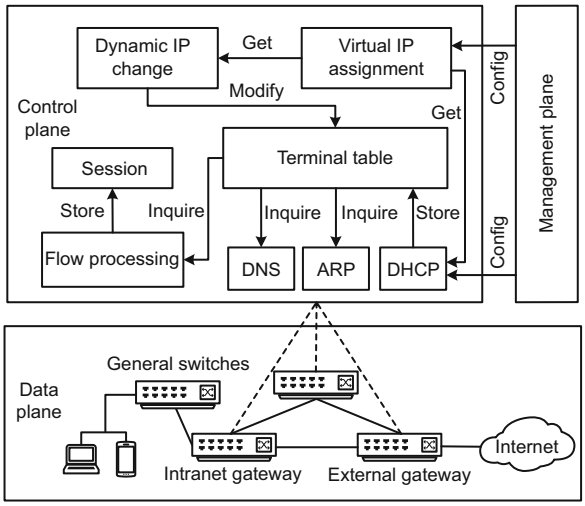

Fig. 1 System framework

Management plane: This plane is responsible for initial system configuration and network operational status.

Control plane: There are two main functions carried out in the control plane:

1. Virtual configuration: The control plane configures resources for the terminal and maintains a dynamic virtual configuration. A dynamic host configuration protocol (DHCP) module and a domain name system (DNS) module are included in this function. A terminal information table is configured for each network terminal in the network, including real configuration values (e.g., rIP) and virtual configuration values (e.g., vIP and vDomain), and the table is stored in the terminal information module.

2. Transmission path: The control plane establishes a session and a transmission path for communication, and maintains a dynamic external network port. A flow processing module and a dynamic maneuvering module are included in this function. When the Packet_In packet is sent, a forwarding path is generated for the packet and a corresponding packet processing flow is delivered. The virtual configuration is used to communicate between terminals in the network. The transmission path is dynamically modified during the maneuvering period.

In this study, we propose an enterprise network defense system. All terminals accessing the system can use only DHCP to obtain rIP. While acquiring rIP, the controller allocates a vIP and a vDomain for each terminal. The mutual access between the enterprise intranets can obtain only the vIP of the destination terminal at the current time through

DNS. In addition to the hyper text transfer protocol (HTTP)-based web page access, the enterprise intranet terminal user can use the command "nslookup (reverse) domain name" to query the vIP of the destination terminal to communicate. Users can obtain the domain name of the other party through the out-of-band method. The specific process is detailed in Section 4.2.

This method breaks the static characteristic of the configuration in a traditional network. By dynamically changing the configuration information in the communication network, the attacker cannot obtain real information about the network. Thus, this approach can effectively prevent attacks such as reconnaissance, greatly improving the network security capability.

3.3 Maneuvering mechanism

Maneuvering a single network property can defend only against some simple cyber-attacks, and it still does not work for some complex attacks, e.g., APTs. However, combining a variety of single attribute maneuverings can form a complex defense system, and the attacker's attack difficulty will rise linearly. We combine IP, port, domain name, and path maneuvering, and combines different maneuvering mechanisms to form a multi-dimensional maneuvering internal network dynamic defense system.

3.3.1 Two-level maneuvering

The rIP performs high-frequency maneuvering. The faster the rIP maneuvers, the more difficult it is for the attacker to scan.

The vDomain performs low-frequency maneuvering. As the communication basis for both terminals, once the vDomain is compromised, it can allow the attacker the opportunity to take advantage of the domain, so it needs to be changed periodically.

3.3.2 Partition maneuvering

Enterprise intranet IP maneuvering: The enterprise intranet IP is different from the public IP, and the company can customize it internally. The enterprise intranet IP maneuvering can effectively block an attacker's sniffing scan attack.

External port maneuvering: The traditional external network communication mode is based on network address translation (NAT), but NAT cannot change the port bound to a communication session. External network port maneuvering will prevent an attacker from simply restoring the communication content from the stolen data packets and locating the internal nodes.

4 Implementation of SPD

We need the global network view (such as the space and routing information) to enforce network property configurations and maneuvering. In traditional networks, we need to modify the current network equipment, and this method with high coupling is more expensive. The SDN architecture separates the control plane from the data plane. The control plane, i.e., the controller, can centrally manage the network devices and accurately program packet forwarding by the programming switches. The data plane can be programmed by the controller, so that we can change network behavior on demand. Therefore, the SDN-based dynamic network can be easily managed and configured by the network controller, and the OpenFlow switch forwards packets according to the flow rules. Therefore, the design does not require modification of other devices in the network.

4.1 Communication protocols

Our system makes sure that the changes to received packets are according to well-designed rules before the packets reach their destination; that is, it makes sure that each packet transmits over the network with a virtual source address and a real destination address. Such a design has the following advantages:

1. For network scanning, the destination IP is scanned. The SPD mechanism requires that users use only the vIP to access the target terminal. If the accurate destination vIP is not obtained, an attack on the target is difficult to establish. Thus, hiding only the source rIP is as good as the work of Jafarian et al. (2012, 2015) in which the source and destination are both hidden.

2. The packets from the terminal which are modified by the access end can be directly received by the destination terminal and do not need any change. Such an approach can improve the efficiency of switch packet processing.

As shown in Fig. 2, the communication process steps are as follows:

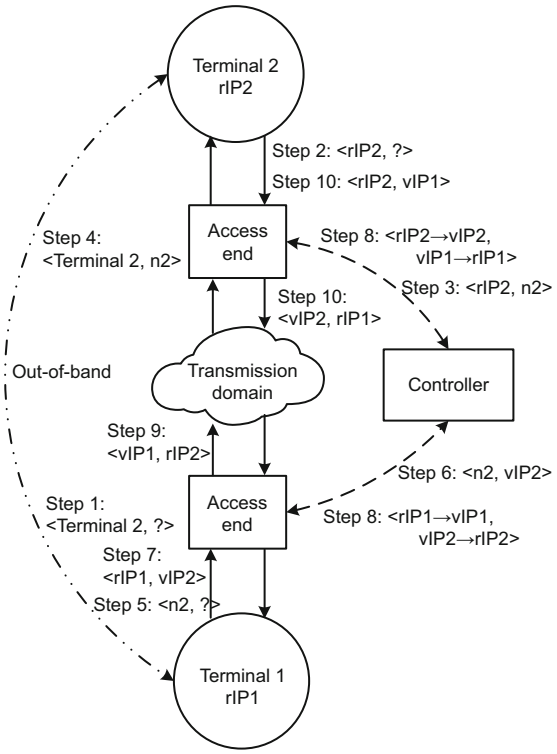

Fig. 2 Communication processing

Step 1: Terminal 1 initiates a request for terminal 2's enterprise intranet domain name in an out-of-band manner.

Step 2: Terminal 2 sends a message <rIP2, ?> to the DNS server of the controller to request the enterprise intranet domain name of terminal 2.

Step 3: The DNS server sends a message <rIP2, n2> to terminal 2 in response to the request of terminal 2, where n2 is the enterprise intranet domain name of terminal 2.

Step 4: Terminal 2 responds to the enterprise intranet domain name n2 requested by terminal 1 in an out-of-band manner.

Step 5: Terminal 1 sends a message <n2, ?> to the DNS server of the controller to request the vIP of terminal 2.

Step 6: The DNS server sends a message <n2, vIP2> to terminal 1 in response to the request of terminal 1.

Step 7: Terminal 1 sends a packet to terminal 2 using its rIP and terminal 2's current virtual address v2 in the format of <rIP1, vIP2>.

Step 8: When the packet passes through the access end of terminal 1, the switch replaces the source address of the packet with the current virtual address vIP1 of terminal 1, and the destination address is replaced with terminal 2's rIP. The message format is <vIP1, rIP2>.

Step 9: The packet is transmitted in the network in the format of <vIP1, rIP2> until reaching terminal 2.

Step 10: The packets of terminal 2 are returned with Steps 7, 8, and 9.

In this study, we establish a session for each communication process and store it. The session is an eight-element array:

$$S = \{\text{sRip, sVip, dRip, dVip, sPort, dPort, Protocol, TTL}\}, \quad (1)$$

where the elements represent the source's rIP, vIP, the destination's rIP, vIP, source port, destination port, communication protocol, and lifetime, respectively. The vIP in the array is the current virtual address of the terminals when the session is established. It will not change until the end of the session's lifetime TTL, even if the terminal's assigned vIP has been updated. The purpose of this is to ensure a good user experience and improve the efficiency of the use of network resources.

The selection of the session period is related to the size of the network and the quality of service (QoS) of the users. In a long-term stable transmission process, the deletion and reconstruction of the session will inevitably cause traffic on the network to be jittery. Therefore, the user experience requires a longer session period, but the long-term existence of the session will result in the accumulation of a large number of useless flow tables. The larger the size of the network, the more sessions, and the greater the load pressure on the controllers and switches caused by the flow backlog. Therefore, selecting the session cycle necessarily involves a compromise between network size and service requirements.

4.2 Packet classification

When an IP packet is sent to the controller, the controller mainly needs to determine that the packet belongs to one of these four types: (1) The sender-to-access packet, the next hop is the receiving end; (2) The sender-to-access packet, the next hop is not the receiving end; (3) The message from the non-access end, the next hop is the receiving end; (4) The message from the non-access end, the next hop is not the receiving end.

Corresponding to the above four types of packets, the switch that processes the packet should be classified into four processing types corresponding to the four cases in which the switch sends the flow.

The OpenDaylight controller processes only the packets uploaded by the access end. First, it needs to verify the source address. If it is the rIP, the controller establishes different communication sessions according to the purpose (intranet or extranet), and then generates path R_1. For each switch in R_1, the forwarding processing flow f_{i} and the return processing flow f_{b} are sequentially delivered. The four cases for the switch are differentiated according to the different locations in the path.

For enterprise intranet communication, the specific processing flow is shown in Algorithm 1. For example, for the access end switch, if the packet belongs to Type.A, the forwarding processing flow f_{i} is sent, which includes the action of modifying the source rIP to vIP, modifying the destination vIP to rIP, modifying the destination media access control (MAC) to the destination terminal MAC, and exiting port s.out. Flow f_{i} corresponds to Algorithm 1:

$$f_{\mathrm{i}}(h_i.\mathrm{rIP} \rightarrow h_i.\mathrm{vIP},\ h_j.\mathrm{vIP} \rightarrow h_j.\mathrm{rIP},\ \mathrm{dst\_mac} \rightarrow h_j.\mathrm{mac},\ \mathrm{output} : s.\mathrm{out}), \tag{2}$$

where "$\rightarrow$" means to modify, the previous content is matching, and "output" is the packet out port.

Finally, the return processing flow f_{b} is added. For the packet where the destination address is source vIP, and the source address is destination rIP, the switch modifies the destination address to the source rIP, the source address to the destination vIP, and the destination MAC to the source MAC. Flow f_{b} corresponds to Algorithm 1:

$$f_{\mathrm{b}}(h_j.\mathrm{rIP} \rightarrow h_j.\mathrm{vIP},\ h_i.\mathrm{vIP} \rightarrow h_i.\mathrm{rIP},\ \mathrm{dst\_mac} \rightarrow h_i.\mathrm{mac},\ \mathrm{output} : s.\mathrm{out}). \tag{3}$$

For external network communication, we need to change the source IP address of the packet to the external network IP address. The source port is mapped to the external network virtual port and forwarded through the external network gateway.

Algorithm 1 Communication processing

```
for all Packet p comes from the OF switch do
  if p.src is rIP and p is from the access end then
    if p.dst is vIP then
      R_1 ← the path from h_i to h_j
      for all switch s in R_1 do
        if s ∈ Type.A then
          f_i(h_i.rIP → h_i.vIP, h_j.vIP → h_j.rIP,
            dst_mac → h_j.mac, output : s.out)
          f_b(h_j.rIP → h_j.vIP, h_i.vIP → h_i.rIP,
            dst_mac → h_i.mac, output : s.in)
        else if s ∈ Type.B then
          f_i(h_i.rIP → h_i.vIP, h_j.vIP → h_j.rIP,
            dst_mac → h_j.mac, output : s.out)
          f_b(h_j.vIP, h_i.rIP, output : s.in)
        else if s ∈ Type.C then
          f_i(h_i.vIP, h_j.rIP, output : s.out)
          f_b(h_j.vIP, h_i.rIP, output : s.in)
        else if s ∈ Type.D then
          f_i(h_i.vIP, h_j.rIP, output : s.out)
          f_b(h_j.rIP → h_j.vIP, h_i.vIP → h_i.rIP,
            dst_mac → h_i.mac, output : s.in)
        end if
        flow_mod(f_i, f_b)
      end for
    end if
  end if
end for
```

4.3 IP allocation

The main goal of network configuration dynamization is to transform the system configuration according to the attacker's behavior and increase the system's unpredictability. The proposed technical solution must be able to solve both IPv4 and IPv6. The scarcity of IP addresses in IPv4 networks makes unused address spaces small and highly fragmented. Therefore, our main challenge is to ensure that the following requirements are met even in a limited and decentralized unused address space such that:

1. The size of the maneuvering subspace determines the degree of diversity of the configuration and its degree of unpredictability. Therefore, the IP maneuvering range of each terminal should be large enough.

2. For a specific communication network, the allocation strategy of the maneuvering subspace is related to the multi-dimensional factors, e.g., the number of network systems, security performance requirements, and network status. Therefore, it is necessary to balance the configuration resources with the security gain to ensure the balance of configuration allocation and that it is non-repetitive.

3. Furthermore, for communication networks, the allocation policy mechanism needs to adapt to

the needs of specific network environments, enabling flexible configuration of administrators.

There are r terminals in the communication network, and each terminal belongs to a subnet in set $S = \{S_1, S_2, \ldots, S_i\}$, where $|S| = I$, and a subnet is a group of terminals physically connected through an OF switch. Define $V = \{V_1, V_2, \ldots, V_j\}$ as a set of available vIP segments, where $V_j = \{v_1, v_2, \ldots, v_m\}$ and $|V| = J$. The process of assigning the virtual address set V to the network system S is a process of virtual mapping.

In the theoretical analysis in Section 5.1, the larger the jump address space is, the stronger the system's anti-sniffer capability is. Jafarian et al. (2012, 2015) chose the two-level maneuvering principle because of the distributed routing limitation, and solved the allocation problem of V to S by satisfiability modulo theory (SMT). Yet this allocation causes each subnet to jump only within the allocated limited space. Therefore, in this study, the shared pool allocation mechanism is adopted in system implementation, so that all terminals share the IP maneuvering pool to maximize the unpredictability of maneuvering.

The specific allocation algorithm follows Algorithm 2. Because IPv4 has a limited and scattered unused address space, the vIP segmentation will be stored in a two-dimensional array vIpR[l][2]. We first randomly select segments, and then randomly select vIP in the segment.

Algorithm 2 IP allocation

```
for each j < J do
    vIpR[j] ← {vIP's start, vIP's end}
end for
vIpPool ← 0
while true do
    i ← Random() mod J
    size ← vIpR[i][1] − vIpR[i][0]
    ip ← vIpR[i][0] + (Random() mod size)
    if vIpPool does not have ip then
        vIpPool.add(ip)
        return ip
    end if
end while
```

4.4 Path maneuvering

In the current network infrastructure, intradomain routing and forwarding policies usually adopt dynamic routing protocols (e.g., open shortest path first and intermediate system-to-intermediate system). They can dynamically change the transmission path of traffic according to network topology changes (e.g., links and node failures) or QoS policies. From the perspective of a dynamic defense, the "active transformation" strategy can be added to improve the dynamic nature of traffic transmission and improve the reliability of transmission in the traditional dynamic routing system. However, as a distributed routing protocol, each routing node advertises state changes to neighboring nodes and then updates to the entire network. If each node changes state too frequently, there is a problem of "route convergence." The centralized control architecture based on SDN can effectively avoid this problem. Dynamic paths for load balancing or reliability are often predictable and cannot handle eavesdropping or DoS attacks on specific nodes or links in the path. Thus, we propose an active random path maneuvering technology, based on the global view of an SDN, by actively and concurrently randomly transforming the paths of multiple streams to resist reconnaissance, eavesdropping, and DoS attacks.

The main challenge of SPD is to randomly change the path between a given source and destination address while considering the following limitations: (1) increasing unpredictability; (2) avoiding any link overload in the network (capacity limit); (3) meeting QoS constraints.

The physical network is denoted as a weighted undirected graph $G = (N, L)$, where N and L represent a collection of physical switches and physical links, respectively. For each physical link $l_i \in L$, there is one available bandwidth resource $B(l_i)$, delay $D(l_i)$, jitter $J(l_i)$, and packet loss rate $S(l_i)$.

Our main purpose is to build a security system. It is not required to satisfy the optimal QoS for the generated path, because considering the optimal solution will result in a selection bias, with which an attacker can easily find the hopping law. Therefore, path generation is required for better distribution and randomness, and the QoS requirement is limited to the minimum QoS standard for different services. Therefore, the weight selection criteria for the link are based on the available bandwidth $B(l_i)$, but the following constraints should be met for the generated arbitrary path $R_j (1 \le j \le K)$:

$$\min_{l_i \in R_j} B(l_i) \ge B, \tag{4}$$

$$\sum_{l_i \in R_j} D(l_i) \le D, \tag{5}$$

$$\sum_{l_i \in R_j} J(l_i) \le J, \tag{6}$$

$$\prod_{l_i \in R_j} S(l_i) \le S. \tag{7}$$

According to the service requirements, B, D, J, and S take different values. We use the recommended values of various service QoS parameters given by the asynchronous transfer mode, Diffserv, and the International Telecommunications Union — Telecommunication as the basis, as shown in Table 1.

In the path maneuvering mechanism of this study, when the controller needs to establish a path for both ends, the initial path R_1 is first obtained through the queue-optimized Dijkstra algorithm, then the flow is sent to establish a connection, and lastly a timer is set for the session. When the maneuvering period expires, based on the initial path and the current link situation, K paths are generated using the K shortest path algorithm. Finally, randomly select one as the next transmission path, and generate a corresponding flow for delivery. The specific path generation is shown in Algorithm 3.

Algorithm 3 Path maneuvering

Require: Physical topology $G(N, L)$

Ensure: Flow

1: **for all** Packet p comes from the OF switch **do**
2: $w[L] \leftarrow$ Calculate weights based on link
3: $R_1 \leftarrow$ Dijkstra$(G(N, L), w[L])$
4: flow $\leftarrow$ Generate a flow based on R_1
5: Discharge flow
6: **end for**
7: **if** maneuvering cycle **then**
8: $R \leftarrow$ Use K shortest path algorithm to obtain path group based on R_1
9: $R_j \leftarrow$ Randomly select a path
10: flow $\leftarrow$ Generate a flow based on R_j
11: Discharge flow
12: **end if**

When the flow is updated, the switch has four cases: (1) not belonging to the new and old paths; (2) belonging to the new path; (3) belonging to the old path; (4) belonging to the new and old paths. No consideration is required for case 1. For case 2, the message is forwarded according to the new flow. For case 3, after the new flow is delivered, and if the packet arrives, it is processed according to the old flow. If the packet does not arrive, the old flow is deleted after Idle_Timeout. For case 4, because the match is the same, the new flow will overwrite the old flow, so the message is processed according to the new flow. Therefore, in the flow update process, there are two cases of packet transmission paths: (1) where the packet is transferred to the new path transmission; (2) where the packet arrives at the destination according to the old path. In both cases, the packet will not be lost during the flow update process. The normal transmission of the message is guaranteed.

In the follow-up study, we will add early warning and trap functions to the network. The early warning function is implemented using Snort for abnormal traffic detection. The trap function is implemented using existing mature honeypot technology for malicious attack detection. All test results optimize the path maneuvering selection strategy by feedback.

Table 1 Performance indicator parameters of typical services

Typical service	Packet loss	Delay (ms)	Jitter (ms)	Upstream bandwidth (Mb/s)	Downstream bandwidth (Mb/s)
High-speed interactive service	<0.10%	<200	<30	2	2
Streaming media	<0.10%	<1000–2000	<1000	2	6
Internet protocol television	<0.10%	<150	<20	2	2–8
E-commerce, confidential service	<1.00%	<1000	–	–	–
Audio and video	<0.10%	<250	<10	–	–
Data file	<0.01%	<1000	–	–	–
Image	<0.01%	<1000	–	–	–
FTP, P2P	<0.10%	–	–	–	–
Real-time data	<0.01%	1–1000	–	0.512	2
Web	<1.00%	<1000	<1000	0.2	4
Telnet	<0.10%	<1000	–	–	–

5 Analysis

5.1 IP maneuvering analysis

To evaluate the IP maneuvering effect, we use the Urn models used by Carroll et al. (2014). For r terminals in the network, the allocated address space size is $v = m \times j$, and the attacker scan count is s. Since the address space is large enough, it is assumed that $v > s$.

5.1.1 Static IP

If the IP address is fixed, let X_s denote the number of terminals swept out in s scans. Then X_s is a hypergeometric distribution with

$$\Pr(X_s = x) = \binom{r}{x}\binom{v-r}{s-x}\Big/\binom{v}{s}. \tag{8}$$

Then the probability that at least one terminal is discovered is

$$\Pr(X_s > 0) = 1 - \Pr(X_s = 0) = 1 - \binom{v-r}{s}\binom{v}{s}. \tag{9}$$

Thus, the number of scans s required to discover an enterprise intranet terminal can be modeled as a negative hypergeometric distribution, so that Y represents the number of scans, and $Y \sim H^{-}(1, r, v)$. Then we have

$$E[Y] = \frac{v+1}{r+1}. \tag{10}$$

5.1.2 Dynamic IP

Optimal maneuvering means that the information obtained by an attacker from the current scan will not provide any useful information for the next scan. Then X_s is a binomial distribution:

$$\Pr(X_s = x) = \binom{r}{x}p^x(1-p)^{s-x},\ p = \frac{r}{v}. \tag{11}$$

Given s scans, the probability of success is

$$\Pr(X_s > 0) = 1 - \Pr(X_s = 0) = 1 - (1-p)^s. \tag{12}$$

Thus, Y obeys the geometric distribution:

$$E\left[Y\right] = \frac{1}{p} = \frac{v}{r}. \tag{13}$$

It can be seen that to increase the difficulty of the scan, it is necessary to increase the maneuvering space of the IP address. SPD allows all terminals to share a large IP pool, so the maneuvering space is large enough, and the difficulty of exploration increases. From the comparison of Eqs. (10) and (13), it can be found that the dynamic nature of the system does not improve the detection difficulty in the same maneuvering space, but SPD can block the attacker's use of the scan results. Even if an attacker can obtain terminal IP, SPD can provide a second-level maneuvering capability, where unless the attacker can break the target terminal within a few seconds, the scanned hit list is only expired IP information.

5.2 Path maneuvering analysis

For traffic eavesdropping in the network, assuming that the data stream transmitted during a period of time T is f, the number of link maneuverings in time T is r_d, the number of nodes per transmission path is $h_i\,(i = 0, 1, \ldots, r_d)$, and the total number of nodes is n.

First, we model the attacker before analyzing the effect. A successful attack is defined as obtaining a complete data stream f. We assume that the attacker has the ability to know whether a link is a target streaming link and has hopped. According to the link condition, the attacker can reselect the listening node according to the policy. The attacker can continuously scan for s times in time T.

5.2.1 Static path

For a static path, an attacker can acquire the entire data stream f as long as it detects one of the transmission paths. Let Z_s denote the number of sweeping target links in s scans, which can be obtained according to the model in Section 5.1.1:

$$\begin{aligned}\Pr(\text{win}) &= \Pr(Z_s > 0)\\ &= 1 - \Pr(Z_s)\\ &= 1 - \binom{n-h_0}{s}\Big/\binom{n}{s}.\end{aligned} \tag{14}$$

5.2.2 Dynamic path

For dynamic paths, the attacker must re-explore the new link of f during its maneuvering period after each maneuver; otherwise, the eavesdropping fails. Thus, the probability of an attacker's success is

$$\Pr(\text{win}) = \prod_{i=0}^{r_d}\left\{1 - \frac{\binom{n-h_i-\operatorname{sgn}(i)}{q}}{\binom{n-\operatorname{sgn}(i)}{q}}\right\}, \tag{15}$$

$$q = \left| \frac{s}{r_d} \right|, \tag{16}$$

where sgn(·) is a symbolic function and q is the maximum number of scans by the attacker during the path maneuvering period. When $0 < q < 1$, it indicates that the path maneuvering period is greater than the detection frequency, and the attacker success probability is 0. Considering the attacker's attack ability, this situation is generally not considered. Because the path maneuvering rate is very fast in this case, the performance requirements of the controller and the switch are very high, which will seriously affect the data stream transmission process. Thus, the general default is $q \geq 1$.

Fig. 3 can be obtained from Eqs. (14) and (15), where $n = 20$ and $h_i = 4$. We find that as the attacker's detection ability increases, the value of s increases, and the probability that there has been eavesdropping on the data is greater. When the system path maneuvering period increases, the probability of an attack gradually reduces. In the actual physical machine test, because of path maneuvering, the attacker's eavesdropping traffic is incomplete and SPD has a good anti-eavesdropping effect.

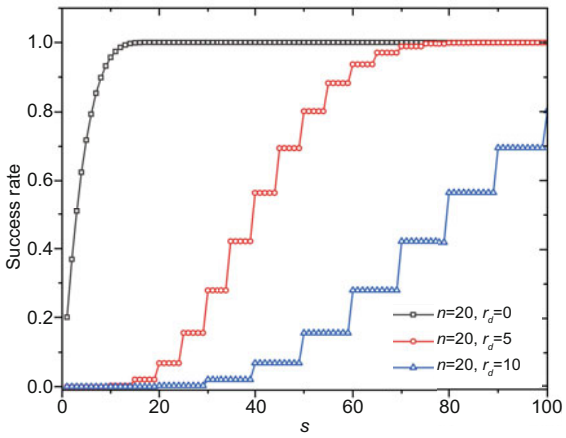

Fig. 3 Path maneuvering effect

6 Evaluation

In this section, we create a small physical network to evaluate our SPD system (Fig. 4). There are two Huawei servers (RH2288H V3), three Pica8 (P3297) switches, one H3C (S1048) switch, and three computers. Specifically, the two servers host services (e.g., file transfer protocol (FTP), mail, and web), and three terminals run on Windows 7, Ubuntu, and Kali (attackers) systems separately. To verify the effectiveness, we use Mininet to simulate a large topology with many virtual terminals.

6.1 Functional verification

The SPD system is based on the OpenDaylight controller and can seamlessly integrate traditional defense solutions, e.g., firewalls, intrusion detection systems (IDS), and other emerging defenses such as honeypots and sandboxes. This satisfies the compatibility principle as discussed in Section 3. Based on the SDN architecture, our controller can manage multiple programmable switches and simplify the management.

Fig. 5 shows the communication flow between two user terminals: H1 and H2. First, H1 has to inform H2, then H2 queries its own domain name through the reverse domain name, and then informs H1 through the out-of-band method. Finally, H1 obtains the rIP of H2 through a domain name search, so that communication can be completed. The flow information of the switch is shown in Table 2.

Table 2 Released flow

Match	Action
LLDP, DHCP, ARP, DNS	Controller
tcp, nw_src: 100.0.0.157, nw_dst: 123.235.121.63	mod_dl_dst: f8:0f:41:20:8c:42, mod_nw_src: 123.235.94.78, mod_nw_dst: 100.0.0.141, output: 1
tcp, nw_src: 100.0.0.141, nw_dst: 123.235.94.78	mod_dl_dst: 6c:0b:84:42:84:51, mod_nw_src: 123.235.121.63, mod_nw_dst: 100.0.0.157, output: 1

From Table 2, link layer discovery protocol (LLDP), DHCP, address resolution protocol (ARP), and DNS are handled by the controller. Intranetwork scanning based on broadcast messages has not been implemented. The controller changes the source rIP of the packet to vIP, the destination vIP to rIP, and the destination MAC address on the access end. In this way, the message needs to be modified only once, directly delivered to the destination.

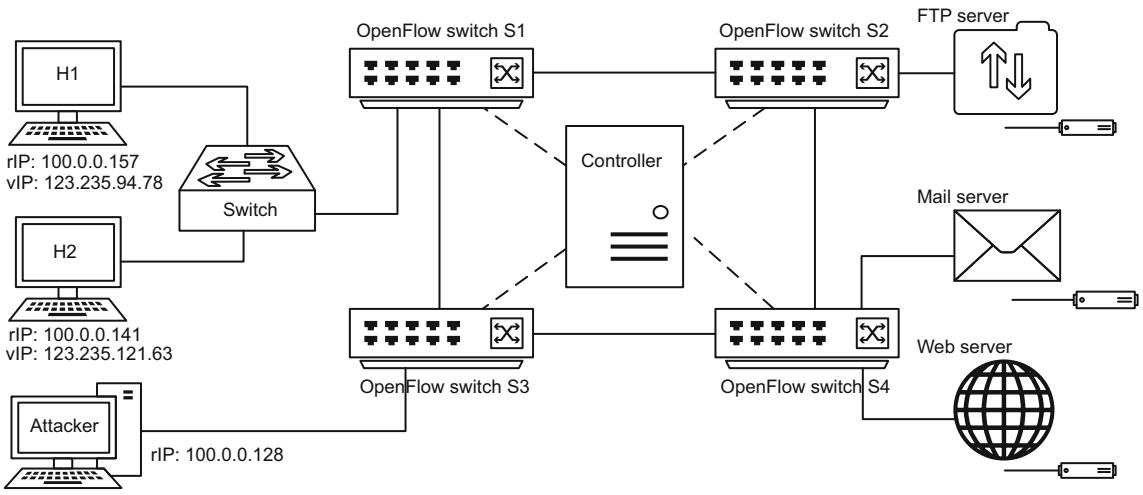

Fig. 4 Network topology

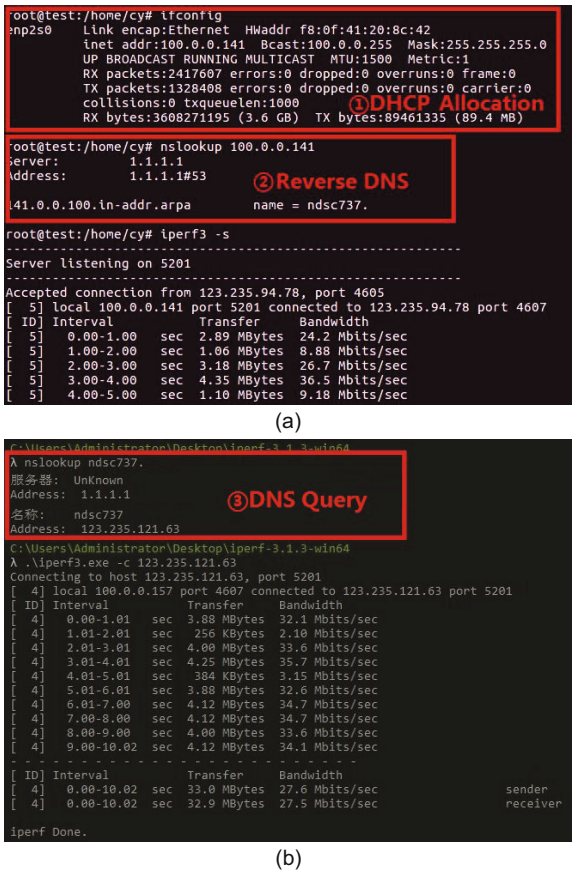

Fig. 5 Communication verification: (a) terminal H2; (b) terminal H1

6.2 Assessment of the effectiveness of the defense system

6.2.1 Anti-scan test

Scanning is usually a precursory step to an attack. Attackers often use scanning tools such as Nmap to discover active hosts in the target network and use them as a hit list. SPD can prevent hitlist-based attacks effectively because the IP addresses will be soon out-of-date.

To show the effectiveness of SPD against attacks, 150 online terminals were generated by Mininet, and 50 of them ran Nmap as an attacker to scan 100 target machines. The online terminal maneuvered in a class B network pool. The attacker scanned the network for 120 min using PING, transmission control protocol packet (TCP_SYN), and reverse domain name (DNS_PTR). As shown in Fig. 6, in any scan, the discovered terminal vIP did not exceed 4%, and subsequent attacks on the scanned target vIP revealed that all IP addresses had expired. Thus, the SPD could make all IP attacks based on IP scans invalid. This is related to the maneuvering space of the terminal, and if we continued to expand its hopping range, its vIP would be less likely to be scanned. We find that PING scans and reverse domain name scans have no results, because SPD will process only domain name pointer (PTR) that belongs to the requesting terminal, and will not process PING messages.

6.2.2 Anti-DoS attack experimental test

The maneuvering of network attributes can prevent DoS attacks. The maneuvering of IP, domain name, and port make the expired virtual information unavailable, which makes the destination terminal unreachable to defend against DoS attacks. Path hopping makes the traffic path unfixed and can defend against DoS attacks on the switch.

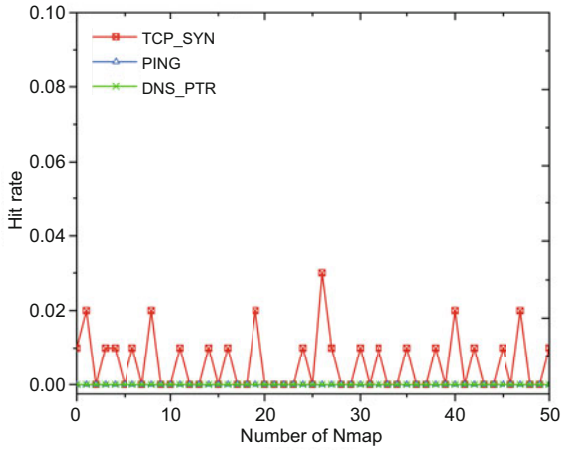

Fig. 6 Ratio of hits

To show the effect of SPD path maneuvering on the DoS attack, the sFlow software monitors the traffic of switch S1 in Fig. 4 and sends a large number of packets to the FTP server to simulate a DoS attack. As can be seen from Fig. 7, S1 enters a large amount of traffic at the beginning, but when the path is switched to S4, the flow pressure of S1 disappears. It can be seen that SPD can balance network traffic and effectively defend against DoS attacks.

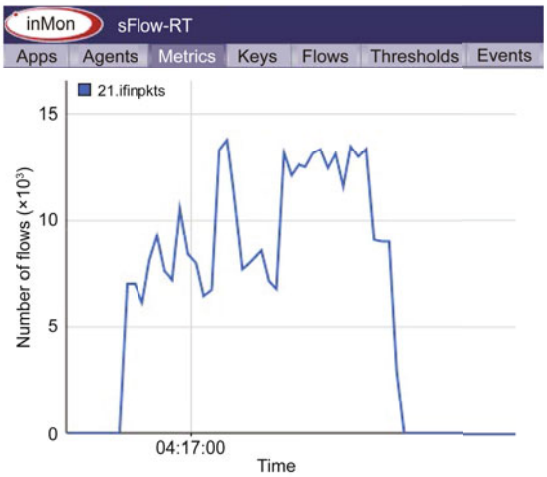

Fig. 7 Anti-denial-of-service (DoS) attack

6.2.3 K value selection

The selection of the number K of maneuvering paths is related to the network size. Define the overlap rate of the network node i as δ:

$$\delta = \frac{n_i}{N}, \tag{17}$$

where N is the number of maneuvers in time T and n_i is the number of occurrences of node i in time T.

In theory, under an ideal network topology, the overlap rate of the network will decrease as the K value increases. However, the actual deployed network topology will be some type of classic network topologies, e.g., FatTree (Al-Fares et al., 2008) and VL2 (Greenberg et al., 2009). Fig. 8 shows the variation in the maximum overlap rate $\delta_{\max}$ of the network over time with different K values under two standard network topologies in Fig. 9. It can be seen that the dynamic path maneuvers can effectively reduce the network overlap rate. Experimental results show that the larger the value of K is, the higher the path overlap rate will be. This is because as the value of K increases, the number of generated path nodes begins to increase, causing some key nodes to increase in frequency. Moreover, it shows that the difference between the effects of $K = 100$ and $K = 5$ is not large. Because calculating K needs to consume system resources, it is suggested through experimental results that the K value should not be greater than half of the number of generated path hops.

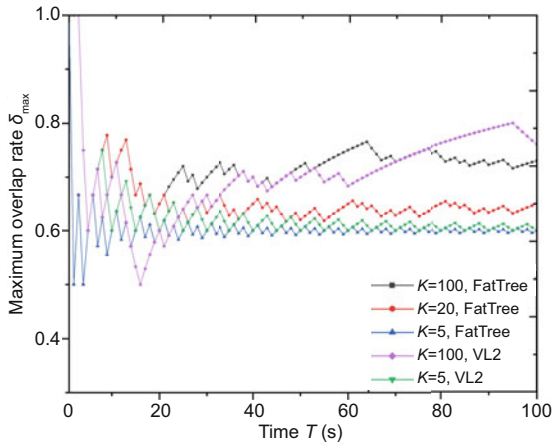

Fig. 8 Overlap rate of network

6.3 Overhead evaluation

6.3.1 Network performance overhead

SPD aims to achieve dynamic maneuvering of network attributes at the control level. Nothing happens to the client, so there is no additional network overhead for the user. For the controller, it is necessary to increase the overhead of the entire network topology and the dynamic maintenance of the access terminal maneuvering information. For the commercially tested OpenDaylight controller, these overheads are within control.

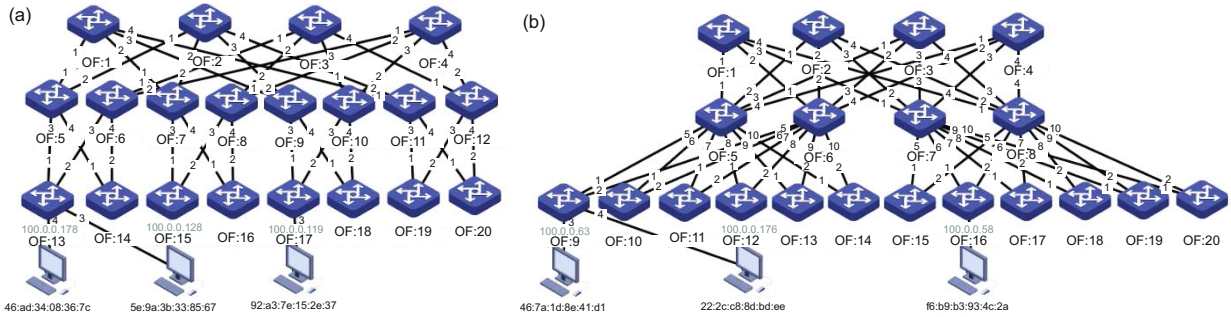

Fig. 9 Data center network architecture: (a) FatTree; (b) VL2. OF: OpenFlow

For the overhead of SDN switches, Fig. 10 compares the delay overhead caused by switch forwarding in RHM, SPD, and normal modes. As can be seen from Fig. 10, the delay overhead caused by SPD is less than RHM but higher than the normal mode. Since SPD will make one modification at the access end, the average delay is slightly higher than the normal mode. However, the MG gateway of the RHM modifies the packet twice, at the ingress and egress, and the performance overhead and packet delay overhead of the switch will be doubled compared to SPD. Fig. 10 shows that the SPD system has a delay peak when processing the first message. This is because in a session in the SPD system, the first packet is uploaded to the controller for processing, which will lead to the maximum delay in the system, about 1 s. Then the path is established, and the switch processes the message through the flow, and the average delay is slightly increased compared with the normal mode.

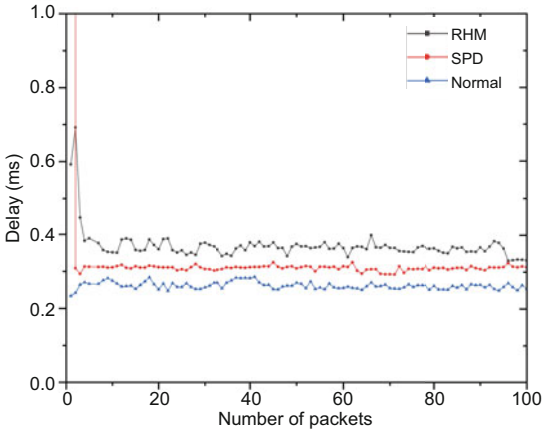

Fig. 10 Network delay caused by switch forwarding in RHM, SPD, and the normal modes

6.3.2 Flow overhead

Because SPD establishes a pair of flows for every session, the number of flows stored by the switch is very large. For n terminals, the maximum number of flows delivered is $C_n^2 + 4$, which is proportional to n_2. As shown in Fig. 11, when the switch is processing the sessions of 100 terminals at the same time, the delivery flows can reach up to 4954, which is a huge challenge for the performance of the switch. In comparison, under normal circumstances, the switch needs only 100 flows.

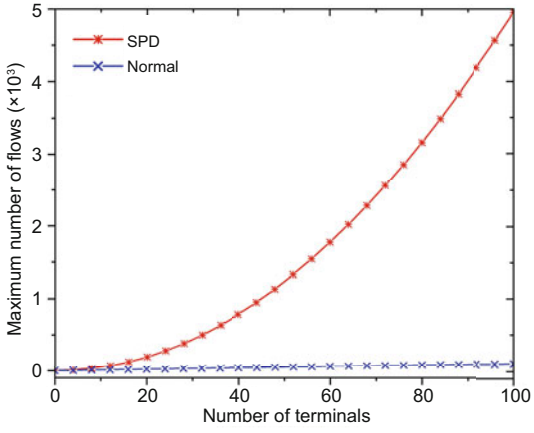

Fig. 11 Number of flows caused by switch forwarding in SPD and the normal modes

7 Conclusions and future work

In this paper, we have tried to defend traditional enterprise network against the potential threats inside. Our core idea is to cut off adversaries' cyber kill chain efficiently. We have designed and implemented a software-defined proactive defense system to prevent the availability of detection by randomly and unpredictably maneuvering more than one network property at the same time, e.g., IP addresses, domain names, ports, and paths. Experimental results showed that our SPD with an appropriate alternation strategy can prevent from almost all network

scanning attacks, and effectively intercept worms, DoS, and other unknown network attacks. However, the flow overhead introduced by the system needs to be further optimized.

In the future, we will deploy considerable decoys to some typical services in the enterprise network. For example, we will deploy Snort in the switch and add honeypot nodes to increase the system security.

References

Al-Fares M, Loukissas A, Vahdat A, 2008. A scalable, commodity data center network architecture. ACM SIGCOMM Conf on Data Communication, p.63-74. https://doi.org/10.1145/1402958.1402967

Antonatos S, Akritidis P, Markatos EP, et al., 2007. Defending against hitlist worms using network address space randomization. *Comput Netw*, 51(12):3471-3490. https://doi.org/10.1016/j.comnet.2007.02.006

Atighetchi M, Pal P, Webber F, et al., 2003. Adaptive use of network-centric mechanisms in cyber-defense. 6th IEEE Int Symp on Object-Oriented Real-Time Distributed Computing, p.183-192. https://doi.org/10.1109/ISORC.2003.1199253

Carroll TE, Crouse M, Fulp EW, et al., 2014. Analysis of network address shuffling as a moving target defense. IEEE Int Conf on Communications, p.701-706. https://doi.org/10.1109/ICC.2014.6883401

Duan Q, Al-Shaer E, Jafarian H, 2013. Efficient random route mutation considering flow and network constraints. IEEE Conf on Communications and Network Security, p.260-268. https://doi.org/10.1109/CNS.2013.6682715

Duo, 2018. Liftoff: guide to duo deployment best practices. https://duo.com/assets/pdf/Duo-Liftoff-Guide.pdf [Accessed on Oct. 18, 2018].

Escobedo V, Beyer B, Saltonstall M, et al., 2017. BeyondCorp 5: the user experience. *Login*, 42(3):38-43.

Flores DA, Qazi F, Jhumka A, 2016. Bring your own disclosure: analysing BYOD threats to corporate information. IEEE Trustcom/BigDataSE/ISPA, p.1008-1015. https://doi.org/10.1109/TrustCom.2016.0169

Greenberg A, Hamilton JR, Jain N, et al., 2009. Vl2: a scalable and flexible data center network. *ACM SIGCOMM Comput Commun Rev*, 39(4):51-62. https://doi.org/10.1145/1594977.1592576

Guan ZT, Li J, Wu LF, et al., 2017. Achieving efficient and secure data acquisition for cloud-supported Internet of Things in smart grid. *IEEE Internet Things J*, 4(6):1934-1944. https://doi.org/10.1109/JIOT.2017.2690522

Hutchins E, Cloppert M, Amin R, 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. In: Ryan J (Ed.), Leading Issues in Information Warfare & Security Research. Academic Publishing International Limited, London, UK, p.80-106.

Jafarian JH, Al-Shaer E, Duan Q, 2012. Openflow random host mutation: transparent moving target defense using software defined networking. 1st Workshop on Hot Topics in Software Defined Networks, p.127-132. https://doi.org/10.1145/2342441.2342467

Jafarian JH, Al-Shaer E, Duan Q, 2013. Formal approach for route agility against persistent attackers. 18th European Symp on Research in Computer Security, p.237-254. https://doi.org/10.1007/978-3-642-40203-6_14

Jafarian JH, Al-Shaer E, Duan Q, 2015. An effective address mutation approach for disrupting reconnaissance attacks. *IEEE Trans Inform Forensics Secur*, 10(12):2562-2577. https://doi.org/10.1109/TIFS.2015.2467358

Kewley D, Fink R, Lowry J, et al., 2001. Dynamic approaches to thwart adversary intelligence gathering. DARPA Information Survivability Conf and Exposition II, p.176-185. https://doi.org/10.1109/DISCEX.2001.932214

Kindervag J, 2010. Build security into your network's DNA: the zero trust network architecture. Technical Report, Forrester Research. http://www.ndm.net/firewall/pdf/palo_alto/Forrester-Build-Security-Into-Your-Network.pdf [Accessed on Nov. 5, 2010].

Kindervag J, 2016. No more chewy centers: the zero-trust model of information security. Technical Report, Forrester Research. http://crystaltechnologies.com/wp-content/uploads/2017/12/forrester-zero-trust-model-information-security.pdf [Accessed on Mar. 23, 2016].

Lei C, Ma DH, Zhang HQ, et al., 2017. Network moving target defense technique based on optimal forwarding path migration. *J Commun*, 38(3):133-143 (in Chinese). https://doi.org/10.11959/j.issn.1000-436x.2017056

Li GL, Wu J, Li JH, et al., 2018. Service popularity-based smart resources partitioning for fog computing-enabled industrial Internet of Things. *IEEE Trans Ind Inform*, 14(10):4702-4711. https://doi.org/10.1109/TII.2018.2845844

Miller KW, Voas J, Hurlburt GF, 2012. BYOD: security and privacy considerations. *It Prof*, 14(5):53-55. https://doi.org/10.1109/MITP.2012.93

Peck J, Beyer B, Beske C, et al., 2017. Migrating to BeyondCorp: maintaining productivity while improving security. *Login*, 42(3):49-55.

Sharma DP, Kim DS, Yoon S, et al., 2018. FRVM: flexible random virtual IP multiplexing in software-defined networks. 17th IEEE Int Conf on Trust, Security, and Privacy in Computing and Communications/12th IEEE Int Conf on Big Data Science and Engineering, p.579-587. https://doi.org/10.1109/trustcom/bigdatase.2018.00088

Talipov E, Jin DX, Jung J, et al., 2006. Path hopping based on reverse AODV for security. 9th Asia-Pacific Int Conf on Network Operations and Management: Management of Convergence Networks and Services, p.574-577. https://doi.org/10.1007/11876601_69

Wu J, Dong MX, Ota K, et al., 2018. Big data analysis-based secure cluster management for optimized control plane in software-defined networks. *IEEE Trans Netw Serv Manag*, 15(1):27-38. https://doi.org/10.1109/TNSM.2018.2799000

Zhou Y, Ni W, Zheng KF, et al., 2017. Scalable node-centric route mutation for defense of large-scale software-defined networks. *Secur Commun Netw*, 2017:4651395. https://doi.org/10.1155/2017/4651395