A SDN Proactive Defense Scheme Based on IP and MAC Address Mutation

Liancheng Zhang(✉), Zhenxing Wang, Jiabao Fang, and Yi Guo

China National Digital Switching System Engineering and Technological Research Center, Zhengzhou 450002, China
lianchengl7@gmail.com, {wzx05,20l4xdfjb}@sina.com, nongfu@live.cn

Abstract. Existing address hopping technologies are hard to be deployed and implemented, at the same time, they only randomly hop IP address information of one communication node or both communication nodes, so they can't protect their identifications on data link layer. In order to deal with these problems, a SDN proactive defense scheme based on IP and MAC address mutation is proposed, which realizes IP and MAC address mutation along the transmission path by installing corresponding address mutation flow entries to intermediate OpenFlow switches. Theoretical analysis and experimental results show that this scheme can resist network interception and analysis attack with a relatively low transmission and processing costs.

Keywords: Address mutation · Address hopping · Software defined network Moving target defense · Proactive defense

1 Introduction

Network sniffer is the key part of information collection and is a significant threat to network security. Nowadays, the costs of network defenders and network attackers are extremely unequal. Network defenders are in a very passive situation, who usually have to add layers of security protection measures for the entire network, while network attackers are relatively active, who sometimes can break the network by only one flaw or vulnerability.

With standards and products of software defined network (SDN) becoming more and more mature and deployments and applications of SDN networks becoming more and more wide (such as cloud computing, data center, etc.) [1, 2], SDN network security issues become increasingly prominent [3, 4]. However, SDN security technologies largely adopt passive network protection and defense methods, such as firewall [5], intrusion detection system [6], denial of service (DoS) detection and defense [7, 8], security policy enforcement [9, 10], etc.

In order to change this passive situation, moving target defense (MTD), a new kind of proactive defense technology, is concerned. MTD's main idea is to make every node in the network becoming a dynamic target, thus can effectively resist network attacks.

As an important component of the MTD technologies, address hopping technologies have been payed more and more attentions by scholars. Address hopping technologies

M. Huang et al. (Eds.): WICON 2016, LNICST 214, pp. 51–60, 2018.
https://doi.org/10.1007/978-3-319-72998-5_6

[11] randomly change IP address information of one or both communication parties, so as to prevent from being discovered and attacked.

As the IP address information is the unique identification of a network node and the foundation of inter-node communication, address hopping technologies face many difficulties when deployed in network environment. Firstly, in order to realize random address hopping, address hopping technologies need to change the IP address configuration of related network nodes, which will add deployment complexity and inconvenience, and limit the hopping frequency of address hopping. Secondly, existing address hopping technologies only randomly hop IP address information of one or both communication nodes, so they can't protect their identifications on data link layer, as they don't hop MAC (media access control) address information.

To enhance SDN proactive defense capability on both network layer and data link layer, this paper presents a SDN proactive defense scheme based on IP and MAC address mutation (SPD-IMAM), where address mutation means just hopping address information along the path (except communication ends) during forwarding process.

2 Related Works

Address hopping technologies achieve proactive security protection by randomly hopping address information of one communication side or both communication sides.

To resist the man-in-middle attack, the authors of [12] put forward dynamic network address translation (DyNAT) mechanism, which changes IP address before communication packets enter the core network or public network. To resist the hitlist worm attack, the authors of [13] propose network address space randomization (NASR) scheme, which provides a random hopping strategy to update network address in the level of local area network (LAN) based on dynamic host configuration protocol (DHCP).

The authors of [11] propose an information hiding technology based on address hopping, which provides multiple paths for data transmission in the process of communication and improves the security of data transmission through the adoption of multiple routing. The authors of [14] put forward a kind of virtual port and address hopping scheme, which utilizes false addresses and ports in the corresponding fields of message packets in the process of communication to confuse attackers.

Motivated by frequency hopping for military communication, end hopping tactic is proposed in [15], which can mitigate DoS and eavesdropping threats greatly by pseudo-randomly changing the end information of port, address, timeslot, cryptographic algorithm or even protocol during end to end transmission.

The authors of [16] put forward random host mutation (RHM) technology by frequently and unpredictably changing IP addresses, which exploits MTC (moving target controller) and MTG (moving target gateway) to implement the transformation between the actual address rIP and the virtual address vIP. Subsequently, the authors of [17] put forward the OF-RHM (OpenFlow random host mutation) technology by using NOX controller to realize the MTG and MTG function of RHM.

Existing address hopping technologies have much interaction with communication sides in hopping process, as a result, they are hard to be deployed and implemented.

Besides, existing address hopping technologies only randomly hop IP address information of one communication side or both communication sides, so they can't protect their identifications on data link layer, as they don't hop MAC addresses.

To enhance SDN proactive defense capability on both network layer and data link layer, aiming at the problem that existing address hopping technologies are difficult to be deployed and implemented, this paper presents the SPD-IMAM scheme, which is independent of the sender and the receiver.

3 The SPD-IMAM Scheme

The SPD-IMAM scheme implements IP and MAC address mutation along the transmission path during forwarding process, as a result, it's independent of the sender and the receiver, which can further mislead attackers.

3.1 SPD-IMAM Framework

Figure 1 shows the overall framework of the SPD-IMAM scheme, where path generation module generates the required path for communication, address mutation module generates the required IP and MAC address information according to specific mutation slot, flow table maintenance module is responsible for installing flow entries into all OpenFlow switches along the transmission path based on particular mutation slot and timely removing expired flow entries on OpenFlow switches along the transmission path.

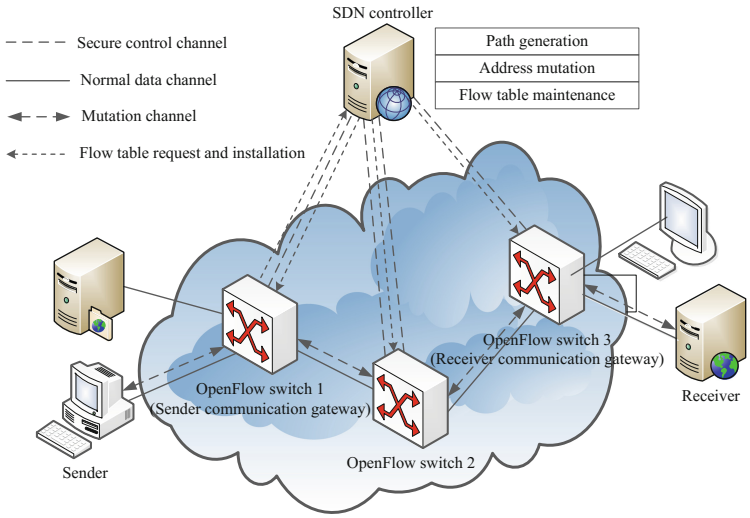

Fig. 1. SPD-IMAM framework.

The SPD-IMAM scheme transfers random hopping function from the sender and the receiver to the communication path, and IP and MAC addresses of every protected packet would be randomly changed by every OpenFlow switch (such as switch 1, 2 and 3 in Fig. 1) along the communication path, as a result, the hopping frequency of the SPD-IMAM scheme is much more faster than traditional address hopping schemes.

The process of address random mutation is shown as in Fig. 2, where rIPa, rIPb represent actual IP addresses, vIP1, vIP2, vIP3, vIP4, vIP5, vIP6 represent virtual IP addresses, rMACa, rMACb represent real MAC addresses, vMAC1, vMAC2, vMAC3, vMAC4 represent virtual MAC addresses.

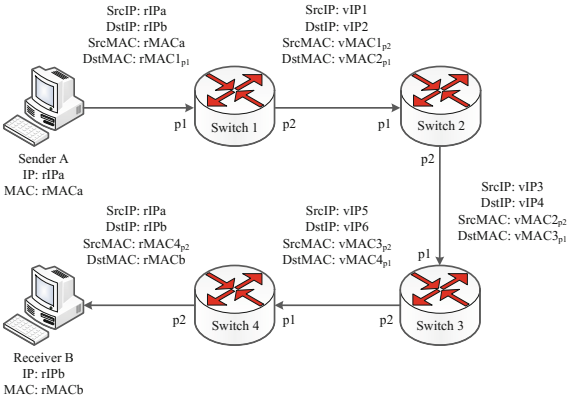

Fig. 2. IP and MAC address mutation.

Flow entries installed into OpenFlow switches (shown in Fig. 2) for address mutation along the transmission path is shown in Table 1.

Table 1. Flow entries for address mutation along the transmission path.

Switch no	Ingress port	SrcMAC	DstMAC	SrcIP	DstIP	Actions
Switch 1	p1	rMACa	$rMAC1_{p1}$	rIPa	rIPb	Set SrcIP=vIP1, DstIP=vIP2 Set SrcMAC=$vMAC1_{p2}$, DstMAC=$vMAC2_{p1}$ Forward to Port p2
Switch 2	p1	$vMAC1_{p2}$	$vMAC2_{p1}$	vIP1	vIP2	Set SrcIP=vIP3, DstIP=vIP4 Set SrcMAC=$vMAC2_{p2}$, DstMAC=$vMAC3_{p1}$ Forward to Port p2
Switch 3	p1	$vMAC2_{p2}$	$vMAC3_{p1}$	vIP3	vIP4	Set SrcIP=vIP5, DstIP=vIP6 Set SrcMAC=$vMAC3_{p2}$, DstMAC=$vMAC4_{p1}$ Forward to Port p2
Switch 4	p1	$vMAC3_{p2}$	$vMAC4_{p1}$	vIP5	vIP6	Set SrcIP=rIPa, DstIP=rIPb Set SrcMAC=$rMAC4_{p2}$, DstMAC=rMACb Forward to Port p2

3.2 Process of OpenFlow Switches

To randomly mutate the IP and MAC addresses of the sender and the receiver, the SPD-IMAM scheme utilizes SDN controller to install corresponding flow entries to every OpenFlow switch on the transmission path, which is shown in Fig. 3.

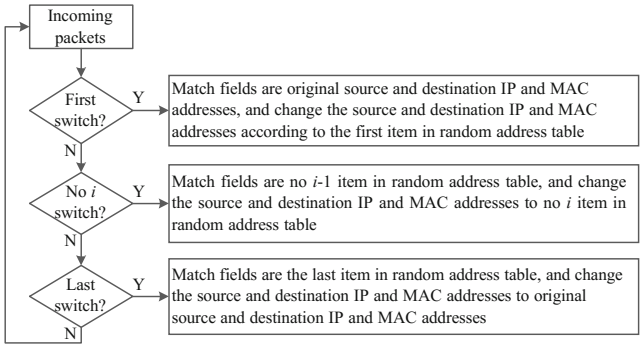

Fig. 3. IP and MAC address mutation process on different switches.

When a packet arrives an OpenFlow switch, the switch will try to search for a matched flow entry according to different match fields. If the switch finds one match, it will perform the actions in the matched flow entry. If the switch finds no match, it will ask SDN controller for a decision.

Before installing corresponding flow entries, SDN controller will generate a random address table, which contains a random source IP address table, a random destination IP address table, a random source MAC address table and a random destination MAC address table. According to these tables, SDN controller can then install corresponding flow entries into these OpenFlow switches along the transmission path.

4 Proactive Defense Ability Analysis of the SPD-IMAM Scheme

To network interception and analysis attacker, we suppose that the attacker only eavesdrop particular links or nodes in certain time and the attacker can intercept all the interactive packets in the network.

Suppose that the protected packets of the SPD-IMAM scheme are $\{p_1, p_2, p_3, \ldots, p_n\}$ and the packets intercepted by the attacker are $\{k_1, k_2, k_3, \ldots, k_m\}$ $(m > n)$. The attacker need to filter out the protected n packets from all m packets, and analyze their real order of the n packets.

Suppose that C_i is the cost of packet interception, C_f is the cost of packet comparison and filtering, and C_r is the cost of obtaining these real ordered packets. Then, the total cost C_{all} can be expressed as $C_{all} = C_i + C_f + C_r$.

Suppose that C_a is the cost for analyzing a single packet. When these compared and filtered packets contain all real ordered communication packets, the value of C_r is minimum, which is:

$$C_{r_{\min}} = \sum_{i=1}^{n} C_a = nC_a = O(n) \tag{1}$$

At the same time, when these compared and filtered packets contain all communication packets and their order is reverse, the value of C_r is maximum, which is:

$$C_{r_{\max}} = \sum_{i=m-n+1}^{n} iC_a = O(n^2) \tag{2}$$

Form above analysis, we can conclude that the overall cost of the attacker is $C_i + C_f + O(n) \leq C_{all} \leq C_i + C_f + O(n^2)$. In general, the value of C_i and C_f are relatively fixed, as a result, the overall cost of the attacker is between $\Omega(n)$ and $O(n^2)$

It is worth noting that even if the network interception and analysis attackers can capture and obtain the real IP address information of real communication nodes, but they can't find the real MAC addresses used by real communication nodes, as a result, this scheme can effectively resist data link layer based network interception and analysis.

5 Experimental Results and Analysis

To test the SPD-IMAM scheme, we use Mininet network simulator [18] and Open vSwitch (OVS) software switches to simulate and set up SDN test network (which is similar with Fig. 1) and utilize Floodlight as SDN controller to be responsible for address mutation function.

Mininet simulator, OVS switches, Floodlight controller, the sender, the receiver and the attacker are deployed on different computers with Intel i7-4790 quad-core 3.6 GHz CPU and 4 GB memory.

5.1 IP and MAC Address Mutation

When the sender pings the receiver (IP addresses of the sender and the receiver are 192.168.1.2 and 192.168.1.5 respectively, MAC addresses of them are 44:37:E6:1A:27:4A and 78:2B:CB:EB:79:D6 respectively), firstly, the sender need to know MAC address of the receiver for starting normal communication, so the sender will send ARP (address resolution protocol) broadcast packets to ask for MAC address of the receiver at first, when the sender get the ARP's reply, it will send an ICMP (Internet control message protocol) echo request packet to the receiver. At this time, flow entries in OpenFlow switches installed by SDN controller is shown in Table 2.

Table 2. Flow entries installed by Floodlight controller.

Switch no	Match fields	Actions
Switch 1	SrcIP: 192.168.1.2 DstIP: 192.168.1.5 SrcMAC: 44:37:E6:1A:27:4A DstMAC: 78:2B:CB:EB:79:D6	SrcIP: 219.120.169.32 DstIP: 36.124.113.28 SrcMAC: 00:13:46:65:BC:7D DstMAC: 00:1E:37:52:5E:EC
Switch 2	SrcIP: 219.120.169.32 DstIP: 36.124.113.28 SrcMAC: 00:13:46:65:BC:7D DstMAC: 00:1E:37:52:5E:EC	SrcIP: 26.116.138.27 DstIP: 116.204.35.16 SrcMAC: 78:2B:CB:13:78:20 DstMAC: 00:13:46:20:EC:1B
Switch 3	SrcIP: 26.116.138.27 DstIP: 116.204.35.16 SrcMAC: 78:2B:CB:13:78:20 DstMAC: 00:13:46:20:EC:1B	SrcIP: 119.20.108.24 DstIP: 29.30.104.29 SrcMAC: 44:37:E6:10:5C:BD DstMAC: 00:1E:37:13:CE:3D
Switch 4	SrcIP: 119.20.108.24 DstIP: 29.30.104.29 SrcMAC: 44:37:E6:10:5C:BD DstMAC: 00:1E:37:13:CE:3D	SrcIP: 192.168.1.2 DstIP: 192.168.1.5 SrcMAC: 44:37:E6:1A:27:4A DstMAC: 78:2B:CB:EB:79:D6

As shown in Table 2, source and destination IP and MAC addresses of the packets which are transmitted between the first and the last OpenFlow switches are random generated IP addresses and random generated MAC addresses, as a result, the SPD-IMAM scheme can realize IP and MAC address mutation to achieve the purpose of concealing real IP and MAC addresses of communication nodes, which can not only protect their address information, but also improve SDN network's proactive defense capacity of resisting network layer based and data link layer based network intercept and analysis attacks.

5.2 Average Transmission Delay

The average transmission delay can reflect the processing performance of the SPD-IMAM scheme. As a result, we evaluate the average transmission delay (shown in Fig. 4) within different route hops when using and not using address mutation.

From Fig. 4, we can see that the transmission delay of the SPD-IMAM scheme is higher than that of normal communication (non-mutation). But the extra delay consumed by the SPD-IMAM scheme is not heavy (approximately 10^{-6} s) and acceptable.

5.3 Overhead of Floodlight Controller's CPU Load

To test the extra processing load of Floodlight SDN controller brought from the SPD-IMAM scheme, we evaluate the influences on Floodlight controller's CPU load by utilizing packets of different length, and the results are shown in Fig. 5.

From Fig. 5, we can see that extra processing load of Floodlight controller is not heavy, which is lower than 4.8%. In order to protect communication process with address mutation, this extra overhead is in an acceptable range.

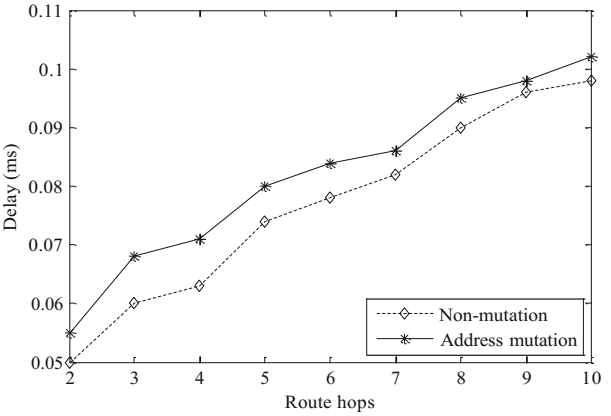

Fig. 4. Forwarding delay difference between address mutation and non-mutation.

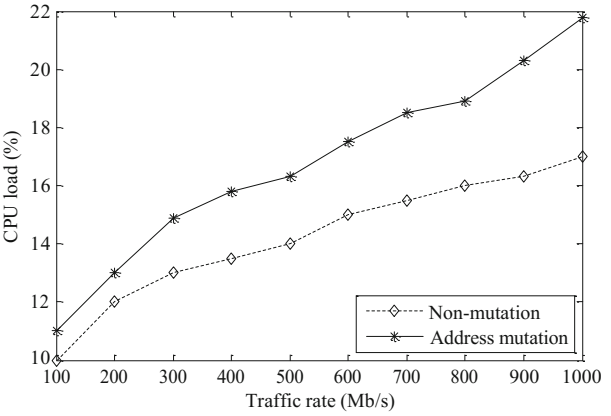

Fig. 5. Floodlight controller's CPU load influenced by the SPD-IMAM scheme.

6 Conclusions

Address hopping technology provides a new way for SDN proactive defense. On this basis, this paper puts forward an IP and MAC address mutation scheme. Based on SDN's features, such as flexible network architecture, logically centralized control and network programmable, this scheme randomly changes source and destination IP and MAC addresses of the protected flow along the communication path, so that attackers can't capture, analysis and restore the protected flow accurately. At the same time, this proposed scheme is independence of source and destination nodes, which can be easily adopted and deployed in SDN networks as well as traditional networks.

For future work, on the basis of this proposed SPD-IMAM scheme, similar idea can be applied to port hopping/mutation, end hopping/mutation, etc. In addition, multiple SDN controllers can be utilized to increase the scalability of the SPD-IMAM scheme.

Acknowledgments. This work was supported in part by National Natural Science Foundation of China under grant 61402526, 61402525 and 61502528, and in part by National High Technology Research and Development Program of China under grant 2012AA012902.

References

1. Xia, W., Wen, Y., Foh, C.H., Niyato, D., Xie, H.: A survey on software-defined networking. IEEE Commun. Surv. Tutorials **17**, 27–51 (2015)
2. Farhady, H., Lee, H., Nakao, A.: Software-defined networking: a survey. Comput. Netw. **81**, 79–95 (2015)
3. Akhunzada, A., Ahmed, E., Gani, A., Khan, M.K., Imran, M., Guizani, S.: Securing software defined networks: taxonomy, requirements, and open issues. IEEE Commun. Mag. **53**, 36–44 (2015)
4. Alsmadi, I., Xu, D.: Security of software defined networks: a survey. Comput. Secur. **53,** 79–108 (2015)
5. Hu, H., Han, W., Ahn, G., Zhao, Z.: FlowGuard: building robust firewalls for software-defined networks. In: ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, pp. 97–102. ACM Press, Chicago (2014)
6. Giotis, K., Argyropoulos, C., Androulidakis, G., Kalogeras, D., Maglaris, V.: Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments. Comput. Netw. **62**, 122–136 (2014)
7. Wang, B., Zheng, Y., Lou, W., Hou, Y.T.: DDoS attack protection in the era of cloud computing and software-defined networking. Comput. Netw. **81**, 308–319 (2015)
8. Wang, H., Xu, L., Gu, G.: FloodGuard: a DoS attack prevention extension in software-defined networks. In: Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, pp. 239–250. IEEE Press, Rio de Janeiro (2015)
9. Shin, S., Yegneswaranz, V., Porrasz, P., Gu, G.: AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks. In: 20th ACM Conference on Computer and Communications Security, pp. 413–424. ACM Press, Berlin (2013)
10. Kreutz, D., Ramos, F.M.V., Verissimo, P.: Towards secure and dependable software-defined networks. In: Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, pp. 55–60. ACM Press, Hong Kong (2013)
11. Sifalakis, M., Schmid, S., Hutchison, D.: Network address hopping: a mechanism to enhance data protection for packet communications. In: Proceedings of 40th Annual IEEE International Conference on Communications, pp. 1518–1523. IEEE Press, Seoul (2005)
12. Kewley, D., Lowry, J., Fink, R., Dean, M.: Dynamic approaches to thwart adversary intelligence gathering. In: Proceedings of DARPA Information Survivability Conference and Exposition II, pp. 176–185. IEEE Press, Anaheim (2001)
13. Antonatos, S., Anagnostakis, K.G.: TAO: protecting against hitlist worms using transparent address obfuscation. In: Leitold, H., Markatos, E.P. (eds.) CMS 2006. LNCS, vol. 4237, pp. 12–21. Springer, Heidelberg (2006). https://doi.org/10.1007/11909033_2
14. Atighetchi, M., Pal, P., Webber, F., Jones, C.: Adaptive use of network-centric mechanisms in cyber-defense. In: 6th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, pp. 183–192. IEEE Press, Hakodate (2003)
15. Shi, L., Jia, C., Lv, S.: Research on end hopping for active network confrontation. J. Commun. **29**, 106–110 (2008)

16. Al-Shaer, E., Duan, Q., Jafarian, J.H.: Random host mutation for moving target defense. In: Keromytis, Angelos D., Di Pietro, R. (eds.) SecureComm 2012. LNICST, vol. 106, pp. 310–327. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36883-7_19
17. Jafarian, J.H., Al-Shaer, E., Duan, Q.: OpenFlow random host mutation: transparent moving target defense using software defined networking. In: Proceedings of the 1st Workshop on Hot Topics in Software Defined Networks, pp. 127–132. ACM Press, Helsinki (2012)
18. Oliveira, R.L.S., Schweitzer, C.M., Shinoda, A.A., Prete, L.R.: Using mininet for emulation and prototyping software-defined networks. In: 2014 IEEE Colombian Conference on Communications and Computing, pp. 1–6. IEEE Press, Bogota (2014)