TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 04/18 pp39–46
DOI: 10.26599/TST.2021.9010077
Volume 28, Number 1, February 2023

Defense Against Software-Defined Network Topology Poisoning Attacks

Yang Gao and Mingdi Xu*

Abstract: Software-Defined Network (SDN) represents a new network paradigm. Unlike conventional networks, SDNs separate control planes and data planes. The function of a data plane is enabled using switches, whereas that of a control plane is facilitated by a controller. The controller learns network topologies and makes traffic forwarding decisions. However, some serious vulnerabilities are gradually exposed in the topology management services of current SDN controller designs. These vulnerabilities mainly exist in host tracking and link discovery services. Attackers can exploit these weak points to poison the network topology information in SDN controllers. In this study, a novel solution is proposed to defend against topology poisoning attacks. By analyzing the existing topology attack principles and threat models, this work constructs legal conditions for host migration to detect host hijacking attacks. The checking of the Link Layer Discovery Protocol (LLDP) source and integrity is designed to defend against link fabrication attacks. A relay-type link fabrication attack detection method based on entropy is also designed. Results show that the proposed solution can effectively detect existing topological attacks and provide complete and comprehensive topological security protection.

Key words: Software-Defined Network (SDN); topology discovery; topology poisoning attacks

1 Introduction

Software-defined networks (SDNs), originating from the campus network of Stanford University, were proposed to solve the bloated and inefficient problems of traditional networks. Through the separation of the data forwarding and routing control of the traditional Internet in SDNs, the centralized control and distributed forwarding of these networks can be realized. The form of programming provides an interface to the outside world[1, 2]. The dynamic and flexible characteristics of SDNs have attracted widespread attention from academia and industries. Researchers in many fields actively use SDNs to build new systems for solving problems, such as the limited scalability of traditional architectures, wireless sensor networks[3], the Internet of Things[4], and optical networks[5]. The recent studies on the security of the SDN topology discovery mechanism mainly involve three aspects: design of a security framework[6–9], construction and addition of a new protocol[10–12], and encryption authentication[13]. TopoGuard[14] is a security extension of the SDN controller, and it detects attacks on the SDN network topology view by fixing the security vulnerabilities in the controller. However, TopoGuard is unable to detect switch-based link fabrication attacks. PolicyTopo[9] proposes a solution to determine the link status on the basis of the information entropy of the network delay. Topology attacks are distinguished by the threshold when the network delay is low. When the network delay is high, PolicyTopo detects attacks by secure ports. The disadvantage is that the state only depends on the linear relationship of adjacent entropy values and the definition of secure ports brings further burden to the network. Reference [8] proposed a defense scheme on the basis of a statistical analysis of link delays to detect relay-type link fabrication attacks, but this mechanism

• Yang Gao and Mingdi Xu are with the Platform Research and Development Department, Wuhan Institute of Digital Engineering, Wuhan 430073, China. E-mail: Shirley_Stefani@163.com; mingdixu@163.com.
* To whom correspondence should be addressed.
Manuscript received: 2021-09-30; accepted: 2021-10-13

can be bypassed by modifying the timestamp of the Link Layer Discovery Protocol (LLDP) frame. Azzouni et al.[11] improved the topology discovery method by constructing a new protocol, reducing the controller load, and improving efficiency and safety. However, with the maturation of the "open-flow discovery protocol", the promotion of a new protocol is bound to affect network deployment, application, standardization, etc. The existing research work in the field of SDN topology security has achieved progress, but most studies aim at a certain type of security threat and are thus limited in terms of the availability of systematic defense methods. Some security solutions need to be modified or added with existing mechanisms during deployment. Consequently, system integration becomes particularly difficult. In addition, the defense methods against relay-type topology attacks rely heavily on the LLDP frame delay threshold, which may thus become invalid when the network delay is high.

On the basis of existing research results, the current work proposes a novel solution to solve the security problems that occur in the topology discovery process of SDNs. This work detects host hijacking attacks by constructing legal conditions for host migrations and designs source and integrity checks for LLDP frames to defend against link fabrication attacks. Then, a relay-type link fabrication attack detection method based on entropy is proposed. This method uses the LLDP frame transmission threshold and entropy threshold to detect network anomalies. Finally, the SDN simulation environment for the design experiment is built through the Mininet and Floodlight controllers. The results show that the proposed solution offers effective defensive against mainstream topology attacks and comprehensive topology security protection.

2 Threat Model

2.1 Host hijacking attacks

Assume that an attacker has read and written permissions to data packets in a network, the attacker can generate data packets and send them to the SDN, and the controller works in passive mode. In Fig. 1a, Host h2 displayed in gray is the infected host. The attacker constructs a data packet whose source address is Host h1 and sends it to Switch s2 by using Host h2. This approach can make the controller think that Host h1 has migrated to Switch s2. Then, all the data packets that need to be sent to Host h1 are forwarded to Host h2 to achieve host hijacking.

2.2 Link fabrication attacks

Link fabrication attacks are divided into forgery and relay-type attacks according to the LLDP frame generation method. Forgery means that the attacker initiates an attack by forging LLDP frames. In Fig. 1b, h2 is the infected host, and the attacker learns and forges s1. According to the default flow rules, s2 is forwarded to the controller, and the controller generates a false link from s1 to s2 after identifying the LLDP data packet. Relaying means that the attacker initiates an attack by relaying real LLDP frames. Figures 1c and 1d show the relay-type link fabrication attacks of hijacking the host and hijacking the switch, respectively. In Fig. 1c, h1 and h2 are the infected hosts. The attacker collects the LLDP frames of s1 in h1 and then uses h2 to send the frames to s2 to generate a false link from s1 to s2. In Fig. 1d, s2 is the infected switch. s2 forwards all LLDP frames to s3, which then forwards the frames to the controller according to the default flow rules, thereby forming a false link from s1 to s3.

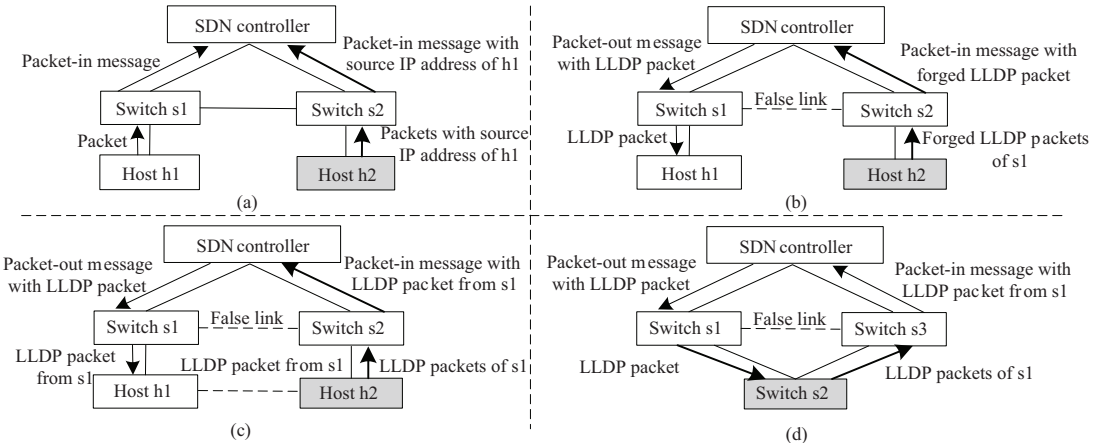

Fig. 1 Attack scenarios considered in this work.

3 Defense Strategy Design

This study proposes an SDN security topology defense solution, which aims to effectively detect network attacks against the topology and provide accurate and effective protection for the SDN topology view. Therefore, this study constructs security strategies for different types of attacks to improve the flexibility and scalability of security mechanisms. The following is an explanation of different topological attacks.

3.1 Strategy design for host location hijacking attacks

A host location hijacking attack is carried out, such that the host tracking service cannot provide real-time verification for host migration while the SDN controller cannot verify the legitimacy of the host location information. Therefore, the key defense measure is to verify the legitimacy of the host migration and the switch ports that are directly connected so as to prevent port reuse. The analysis of the host migration process of the SDN reveals that two conditions are generated when the host is migrated. First, the SDN controller receives the port-down message from the data plane before the host is migrated. Second, after the host migration is completed, the previous location should no longer be accessible. This study verifies the legitimacy of the host migration on the basis of these two conditions. To facilitate verification, we maintain a host mapping table, which binds host location information and switch port information. This table can be used to check port reuse and set modification permissions to prevent illegal host migration.

To ensure the source of the LLDP frame, we check the port on which the switch accepts the LLDP frame. As a feature of SDN link discovery is that the host does not forward LLDP frames by default, once the LLDP frame comes from the switch port directly connected to the host, the corresponding LLDP frame can be regarded as successfully relayed. Therefore, this study checks whether the switch port receiving the LLDP frame is connected as a host and judges the validity field of LLDP frames.

3.2 Strategy design for link fabrication attacks

The main causes of link fabrication attacks can be summarized as follows. First, for a link fabrication attack, the integrity and source of the LLDP frame cannot be ensured during the topology discovery process. Second, for a relay-type link fabrication attack, the compromised host can interfere with the transmission path of LLDP frames. The following defense strategies are discussed accordingly.

To ensure the integrity of LLDP frames, we implement a defense mechanism by adding the "Verification TLV" field to the LLDP frames. The field is calculated using the DPID, port number, and sending time of the LLDP frame to ensure the unforgeability and integrity of the package. The expanded frame format is shown in Fig. 2, where the "Verification TLV" field is used for integrity checking.

To ensure the source of the LLDP frame, we check the port on which the switch accepts the LLDP frame. As a feature of SDN link discovery is that the host does not forward LLDP frames by default, once the LLDP frame comes from the switch port directly connected to the host, the corresponding LLDP frame can be regarded as successfully relayed. Therefore, this study checks whether the switch port receiving the LLDP frame is connected as a host and judges the validity field of LLDP frames.

To solve relay-type link fabrication attacks, existing research usually introduces a link delay to identify any interference in the transmission path of the LLDP frame. However, the instability of link delays can easily cause misjudgment, resulting in a relatively high rate of false positives. To reduce the false positive rate, this study introduces information entropy to check the distribution of destination IP addresses in LLDP frames in the SDN.

3.3 Relay-type link fabrication attack detection strategy based on information entropy

The attacker constructs a false link for the SDN network by relaying LLDP frames, which mainly cause relay-type link fabrication attack. This action has two aspects. On the one hand, the relay will increase the transmission delay of the corresponding LLDP frame. On the other hand, the false link generated by the relay-type link fabrication attack will make the IP address of the victim host appear more frequently within a certain range. For the former, this paper constructs a delay threshold σ' for detection, and then the paper calculates the information entropy of the destination IP and detects relay attacks

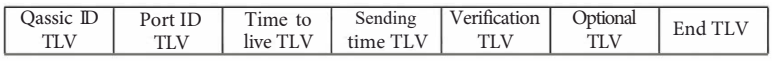

Fig. 2 Extended LLDP frame format.

based on the basis of the entropy threshold σ.

This study sets the LLDP frame transmission delay threshold σ', and the threshold calculation formula is as follows:

$$\sigma' = 2 \times \frac{\sum_{i=1}^{W} T_i - T_{\max} - T_{\min}}{W-2},$$

where $T_{\max}$ is the maximum transmission delay, and $T_{\min}$ is the minimum transmission delay. To eliminate the influence of network fluctuations on the delay threshold, we remove the maximum and minimum values to calculating the average value. Then, we increase the average value by 2 times while considering the network delay W, which is the number of packets through the preset window.

Entropy was originally used to measure the disorder of a physical system and was later described as a measurement of system uncertainty. In 1948, Shannon introduced the concept of entropy into information theory and referred to it as Shannon entropy.

As entropy can measure the randomness of random variables, this article counts the frequency of occurrences of the destination IP address in the flow entries. By marking the destination IP address as X, the number of occurrences as x_i, and the probability of occurrence as p_i, we derive the calculation formula for the entropy value of the destination IP in all the traffic table statistics item in the preset window W is as follows:

$$H(x) = -\sum_{i=1}^{W} p_i \log_2 p_i,$$

$$p_i = \frac{x_i}{W}.$$

According to the above equations, we calculate the entropy value every time W data packets are passed. Next, we discuss the value of W. If the value of W is relatively high, the number of entropy calculations decreases, and the time for each calculation of entropy increases. If the value of W is relatively low, the entropy calculation time decreases while the number of updates increases. On the basis of Ref. [15], this study sets the value of W to 50. According to the statistical analysis, when all data packets are sent to the same destination address, the probability of the corresponding event is the largest, the randomness is the smallest, and the entropy value is the smallest. When all the destination addresses occur the same number of times, all events have the same probability of occurrence, the randomness is the largest, and the entropy value is the largest. Therefore, this study realizes the detection of relay-type link fabrication attacks by calculating the corresponding entropy value and comparing it with the attack threshold. A 95% confidence level is used herein to calculate the entropy threshold. The calculation is as follows:

$$a = \bar{x} + z\frac{s}{\sqrt{N}}, \quad b = \bar{x} - z\frac{s}{\sqrt{N}},$$

$$\sigma = \left(\frac{H_N^{\max} - H_A^{\min}}{H_N^{\max}} - \frac{H_N^{\min} - H_A^{\max}}{H_N^{\min}}\right) \times \overline{H}_A + H_A^{MAX},$$

where a and b represent the maximum and minimum values of the confidence interval, respectively, $\bar{x}$ is the average value of the entropy, z is the sample variance, s is the variance of X, and N is the total value of the entropy. For z is the statistic corresponding to a certain confidence level, which can be obtained by looking up the table, the z value is 1.96 at the 95% confidence level. In the calculation formula of the entropy threshold σ, $H_N^{\max}$ represents the upper confidence limit of the normal network, and $H_A^{\min}$ represents the lower confidence limit of the normal network. Meanwhile, $H_A^{\max}$ represents the upper confidence limit of the attacked network, and $H_N^{\min}$ represents the lower confidence limit of the attacked network, $\overline{H}_A$ represents the average entropy value of the attacked network, and H_A^{MAX} represents the maximum entropy value of the attacked network.

To avoid misjudgment caused by a single entropy value (for example, because the host enters the network), this study introduces the cumulative entropy value of the abnormal queue counter C_θ and abnormal queue counter θ. The principle is that when the entropy value of θ consecutive windows is lower than the entropy threshold, a relay-type link fabrication attack is deemed to have occurred.

4 Defense Module Design

The defense modules in this work are embedded in the SDN controller. They are independent of other functional modules of the controller. The architecture is shown in Fig. 3. The host verification module and link verification module are designed herein and used to deal with host hijacking attacks and link fabrication attacks. The two modules are described in the following sections.

4.1 Host verification module design

The host verification module saves a host location mapping table L_h, which binds the host location and the connected switch in the form of a hash table to ensure that the verification of the host location information is completed in a short time. As the link verification module also needs L_h to verify the source of the LLDP

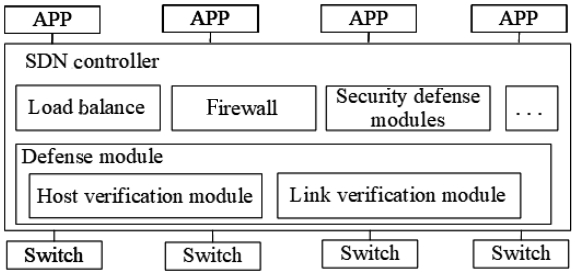

Fig. 3 Location diagram of defense module.

frame, the search object is the switch information at this time. For the key value structure of the hash table, the time complexity of using a key query is $O(1)$, and the time complexity of using a value query is $O(n)$. Therefore, some researchers[16] chose to add another switch mapping table L_s. However, maintaining an information table separately for the two modules results in serious consistency problems, which can easily cause network errors. Therefore, this study only constructs a host location mapping table, and the key uses switch information. The host location structure is shown in Table 1. The key of the mapping table is a complex object composed of the switch's unique identifier DPID and the port number. The mapping table contains the host's MAC address[17].

The dynamic maintenance of the host mapping table involves three events, namely, adding, modifying, and removing. When the SDN is initialized or a new host is added to the network, a new record is constructed and added to L_h. When the host is migrated, the host verification module searches and modifies L_h according to the port status message generated by the switch. When the host or switch leaves the network, the host verification module searches for and deletes the related records in L_h.

In a host hijacking attack, when the controller receives the packet-in message sent by the data plane, the host verification module extracts the host location information in the message to match it with L_h. The three matching results are as follows:

(1) If the MAC address of the host matches the information of the connected switch, then the message is legal.

(2) If the MAC address of the host matches but the connected switch does not match, then check whether the original path recorded by L_h is reachable. If it is reachable, then the message is illegal; otherwise, the message is legal, and L_h is modified at the same time.

(3) If the MAC address of the host does not match the information of all connected switches, then the other security modules (security policies) in the controller can be called to check the legitimacy of the host.

Table 1 Host location mapping table structure.

Key	Value
DPID, port	MAC

4.2 Link verification module design

The link verification module mainly includes three functions: verifying the integrity of the LLDP frame, verifying the source of the LLDP frame, and detecting whether a relay-type link fabrication attack occurs. For the integrity check, this study modifies the LLDP frame format by adding the "Verification TLV" field. The field value is calculated by the md5 algorithm on the basis of the DPID, port number, and sending timestamp of the LLDP frame. For the source verification, the source address of the LLDP frame is searched in the host location mapping table L_h to check whether it is a switch. If it is a switch, then the source of the current frame is legal; otherwise, the frame is discarded, and a warning is generated. In addition, the link verification module stores a hash table L, an abnormal queue counter C_θ, and an abnormal queue counter threshold θ. The hash table L is used for the number of occurrences of the destination IP in the network, and the abnormal queue counter C_θ is used to accumulate entropy. The inspection process is as follows:

Step 1: The switch uses the data packet to match the flow table. If it succeeds, then it records the destination IP address in the data packets as X and forwards it according to the flow rule; otherwise, it sends a Packet-In message to the controller.

Step 2: The controller extracts the destination IP address of the data packet as X. If X exists in the hash table, then L modifies x_i to $x_i + 1$; otherwise, it records the number of occurrences x_i as 1. Then, the sum of L is calculated and compared with W. If L is greater than W, then an entropy $H(x)$ calculation is performed in the window; otherwise, the next step is initiated.

Step 3: The link verification module checks whether the data packet is an LLDP frame. If it is not, the controller calls other modules to complete the subsequent processing; otherwise, the link verification module performs verification on the LLDP frame. First, to check the integrity, we calculate the signature by using the DPID, port number, and timestamp of the LLDP frame. Second, we compare the calculation result with the "Verification TLV". If they are different, then

the frame is an illegal LLDP frame and is discarded by the link verification module, and Step 6 is initiated; otherwise, the next step of verification is performed.

Step 4: The link verification module verifies the source of the LLDP frame. It extracts the switch DPID of the LLDP frame and the entry port number record to match the host location mapping table L_h. If the match is successful, then the LLDP frame is an illegal data packet from the host, the link verification module discards the LLDP frame, and Step 6 is initiated; otherwise, the next check is performed.

Step 5: The link verification module checks whether the LLDP propagation path has artificially interfered. First, it compares the transmission delay of the LLDP frame with the delay threshold σ'. If the delay is less than σ', then the LLDP frame is not relayed; otherwise, the relationship between the calculated entropy value of Step 2 and the entropy threshold is checked. If the calculation result of Step 2 is greater than σ, then the module continues to wait for the arrival of the next packet and sets the abnormal queue counter C_θ to 0; otherwise, it adds 1 to C_θ. If C_θ reaches the abnormal queue counter threshold θ, then a relay-type fabrication attack is deemed to have occurred, and Step 6 is initiated to locate and handle abnormal nodes; otherwise, we continue to execute other SDN functional modules.

Step 6: When the LLDP frame is found to be illegal, the link verification module locates the abnormal node by obtaining the entry port of the illegal data packet. To avoid the bandwidth resource consumption problem caused by the flooding attack, the SDN controller generates the flow rules to discard the data packet and reduces the number of illegal packet-in messages within the specified time.

5 Experiment and Discussion

In this work, Floodlight is selected as the SDN controller, and defense modules are installed in the control layer. The data layer uses Mininet to simulate the network environment. The switch uses the virtual machine switch OVS. All experiments are conducted on a virtual machine configured with 4 GB memory and an operating system of Ubuntu 14.04. The host is configured with a 3.6 GHz CPU and 16 GB memory. Scapy is used to construct data packets. The test environment is shown in Fig. 4, and the test topology is shown in Fig. 5.

In the host hijacking attack experiment, Host h4 is the infected host. The attacker first obtains the IP address and MAC address of Host h1 by using the ARP request and then uses Scapy to construct a data packet whose source address is Host h1. The attacker subsequently sends the data packet to Switch s2 by using Host h4. This process can simulate a host hijacking attack that attempts to change the location of Host h1 to achieve the purpose of hijacking traffic. The experimental results are shown in Fig. 6. This migration does not meet the preconditions for host migration, and the host location information in the data packet does not match the host mapping table. Thus, the host migration fails.

In the link fabrication attack experiment, Host h4 is the infected host. The attacker intercepts and learns the LLDP frame sent by target Switch s1. After forging an LLDP data packet received by Switch s1 from the SDN controller, the attacker uses Host h4 to send the data packet to Switch s2. Then, Switch s2 forwards the fake packet to the controller according to the default flow rule. The experimental result is shown in Fig. 7. The integrity check of the LLDP frame succeeds, but the source check fails. Thus, the frame is discarded.

In the relay-type link fabrication attack experiment, two relay-type attacks are constructed. Hosts h1 and

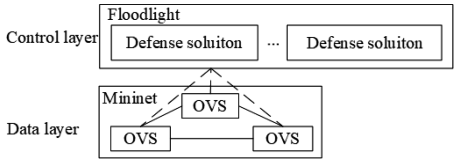

Fig. 4 Diagram of experimental environment structure (OVS means Open vSwitch).

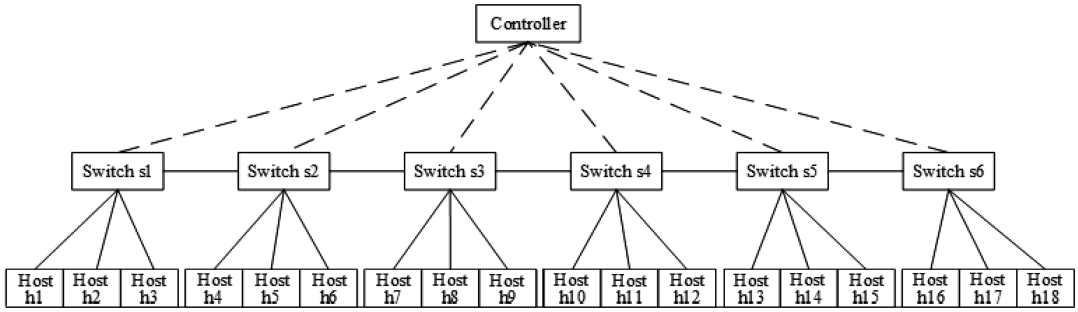

Fig. 5 Test topology.

```
r] Controller will automatically deserialize all Ethernet packet-in messages
r] Controller role set to Active
PortManager] Host h1 location migrated
PortManager] From Switch[h1][p1] to Switch[s2][p1]
PortManager] No Port-Down event was generated
PortManager] Host location mapping table matching failed
```

Fig. 6 Results of host hijacking attack detection.

```
yManager] The LLDP frame from Switch[h1][p1] to Switch[s2][p1] is faked!
yManager] The LLDP frame integrity check successful
yManager] The LLDP frame source check failed
yManager] The LLDP frame is dropped
```

Fig. 7 Results of link fabrication attack detection.

h13 are the infected hosts, switch s2 is an infected switch, and the internally installed malicious flow rule to forward LLDP frames received from Switch s1 to Switch s3. The window size W is set to 50, and the attacker relays the LLDP frame of Switch s1 to Switch s5 to simulate a relay-type link fabrication attack initiated by the hijacking switch.

First, we determine the delay threshold σ'. The experiment launches a host hijacking link fabrication attack on the test topology. Figure 8 intercepts the transmission delay of 15 LLDP data frames in a certain window. We observe a significant increase in the LLDP frame transmission after the link fabrication attack occurs.

Second, we determine the entropy threshold σ. The experiment launches two types of relay-type link fabrication attacks on the test topology. Before launching attacks, we set the hosts to communicate with each other to simulate a normal network environment. Figure 9 shows the relationship between the threshold and the entropy according to the entropy value and threshold calculation formula of the SDN under attack traffic and normal traffic. The abscissa is the number of windows, and the ordinate is the corresponding entropy value under the window. The results show that the entropy value drops significantly from the 20th window and is lower than the threshold. At this time, the attacker initiates relay-type link fabrication attacks.

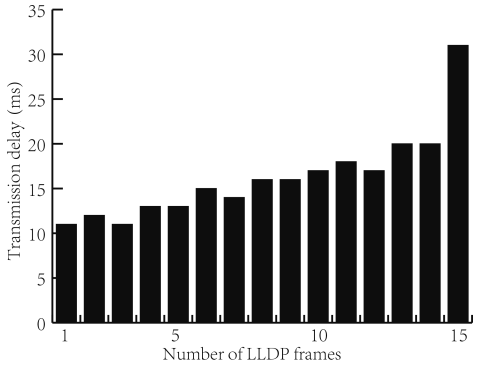

Fig. 8 LLDP frame transmission delay.

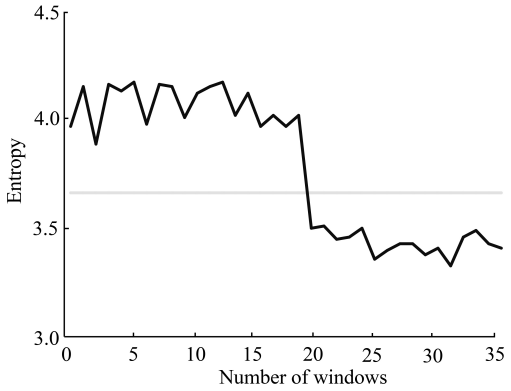

Fig. 9 Test result of the target IP entropy of the victim host.

Third, we determine the value of the abnormal queue counter threshold θ. The traffic statistics and entropy calculation of a single window cannot accurately determine whether it is caused by a relay attack. Hence, multiple consecutive windows should be judged, and the appropriate abnormal queue counter threshold θ should be identified. The threshold needs to consider two factors, namely, false positive rate and detection performance. The experiment uses Scapy to construct data packets to increase the communication simulation hotspot access between the hosts. A total of ten tests are conducted, and the results are shown in Table 2.

According to the experimental results, the false positive rate is relatively high when the threshold is equal to 1 and 2. Meanwhile, the threshold drops to below 50% when the threshold is 3. The false positive rate is 0 when the threshold increases to 5. The increase in threshold brings a decrease in false positive rate and an increase in the detection time. In summary, we set the threshold to 3.

Finally, the effectiveness of the defense is verified. The experimental results are shown in Fig. 10. The LLDP frame transmission delay timeout and the abnormal

Table 2 Impact of abnormal queue counter threshold on false positive rate and detection time.

θ	False positive rate (%)	Average detection time (s)
1	70	2.8
2	60	5.5
3	40	8.3
4	20	11.0
5	0	13.8
6	0	16.5

```
yManager] Sending LLDP packets out of all the enable ports
yManager] LLDP packets timeout
yManager] Exception queue counter is 3, more than threshold
yManager] Relay-type attack occurred
```

Fig. 10 Relay-type link fabrication attack detection results.

queue counter exceed the threshold. Therefore, a relay-type link fabrication attack is deemed to have occurred.

6 Conclusion

To solve the security problem in which the global topology view in the SDN controller is easily tampered with by attackers, this study designs a security solution. First, the existing principles and threat models of topology attacks are analyzed, and the legal condition detection for host migration is constructed to defend against host hijacking attacks. Second, LLDP source check and integrity check are designed to defend against link fabrication attacks. Third, a relay-type link fabrication attack detection method based on entropy calculation is defined to construct LLDP frames. Finally, an SDN simulation environment is built through the Mininet and Floodlight controllers. The results verify the effectiveness of our solution against mainstream topology attacks and indicate its capability of providing complete and comprehensive topology security protection.

References

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, OpenFlow: Enabling innovation in campus networks, *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[2] X. Mingdi and G. Yang, Distributed deception defense system based on SDN, (in Chinese), *J. Commun.*, vol. 39, no. S2, pp. 54–60, 2018.

[3] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, A survey on software-defined wireless sensor networks: Challenges and design requirements, *IEEE Access*, vol. 5, pp. 1872–1899, 2017.

[4] S. B. Zhang, G. J. Wang, M. Z. A. Bhuiyan, and Q. Liu, A dual privacy preserving scheme in continuous location-based services, *IEEE Internet Things J.*, vol. 5, no. 5, pp. 4191–4200, 2018.

[5] A. Aguado, E. Hugues-Salas, P. A. Haigh, J. Marhuenda, A. B. Price, P. Sibson, J. E. Kennard, C. Erven, J. G. Rarity, M. G. Thompson, et al., Secure NFV orchestration over an SDN-controlled optical network with time-shared quantum key distribution resources, *J. Lightw. Technol.*, vol. 35, no. 8, pp. 1357–1362, 2017.

[6] A. R. Abdou, P. C. Van Oorschot, and T. Wan, Comparative analysis of control plane security of SDN and conventional networks, *IEEE Commun. Surv. Tutor.*, vol. 20, no. 4, pp. 3542–3559, 2018.

[7] S. Hong, L. Xu, H. P. Wang, and G. F. Gu, Poisoning network visibility in software-defined networks: New attacks and countermeasures, in *Proc. of the 22nd Annu. Network and Distributed System Security Symp.*, doi: 10.14722/ndss 2015.23283.

[8] D. Smyth, S. McSweeney, D. O'Shea, and V. Cionca, Detecting link fabrication attacks in software-defined networks, in *Proc. 26th Int. Conf. on Computer Communication and Networks*, Vancouver, Canada, 2017, pp. 1–8.

[9] Y. Q. Lu, Z. S. Mao, Z. Cheng, J. C. Qin, D. Z. Jin, and W. Q. Pan, Research on SDN topology attack and its defense mechanism, (in Chinese), *J. South China Univ. Technol. (Nat. Sci. Ed.)*, vol. 48, no. 11, pp. 114–122, 2020.

[10] F. Pakzad, M. Portmann, W. L. Tan, and J. Indulska, Efficient topology discovery in software defined networks, in *Proc. 8th Int. Conf. on Signal Processing and Communication Systems* (*ICSPCS*), Gold Coast, Australia, 2014, pp. 1–8.

[11] A. Azzouni, R. Boutaba, N. T. M. Trang, and G. Pujolle, sOFTDP: Secure and efficient OpenFlow topology discovery protocol, in *Proc. 2018 IEEE/IFIP Network Operations and Management Symp.*, Taipei, China, 2018, pp. 1–7.

[12] A. Azzouni, N. T. M. Trang, R. Boutaba, and G. Pujolle, Limitations of OpenFlow topology discovery protocol, in *Proc. 16th Annu. Mediterranean Ad Hoc Networking Workshop* (*Med-Hoc-Net*), Budva, Montenegro, 2017, pp. 1–3.

[13] T. Alharbi, M. Portmann, and F. Pakzad, The (in)security of topology discovery in software defined networks, in *Proc. 40th IEEE Conf. on Local Computer Networks*, Clearwater Beach, FL, USA, 2015, pp. 502–505.

[14] S. Q. Xiang, H. B. Zhu, L. L. Xiao, and W. L. Xie, Modeling and verifying TopoGuard in OpenFlow-based software defined networks, in *Proc. 2018 Int. Symp. on Theoretical Aspects of Software Engineering* (*TASE*), Guangzhou, China, 2018, pp. 84–91.

[15] Z. Y. Zhao, Research and application of DDoS attack detection and protection technology based on SDN, Master thesis, Beijing University of Posts and Telecommunications, Beijing, China, 2019.

[16] Z. S. Mao, Research on topology security based on SDN, Master thesis, South China University of Technology, Guangzhou, China, 2020.

[17] J. Ren, J. Li, H. Liu, and T. Qin, Task offloading strategy with emergency handling and blockchain security in SDN-empowered and fog-assisted healthcare IoT, *Tsinghua Science and Technology*, vol. 27, no. 4, pp. 760–776, 2022.

Yang Gao received the BEng degree from Northeastern University, China in 2016, and the MEng degree in computer science from Wuhan Institute of Digital Engineering, China in 2019. She is currently an assistant engineer at Wuhan Institute of Digital Engineering. Her research interests include Software-Defined Network (SDN) and cyberspace security.

Mingdi Xu received the BEng, MEng, and PhD degrees from Wuhan University, China in 2002, 2006, and 2009, respectively. He is currently a researcher and the department director at Wuhan Institute of Digital Engineering. His research interests include trusted computing and cyberspace security.