Study the impact of topology-related attacks in Software Defined Network

Darius Cristian Ivascu
Supervisor(s): Mauro Conti, Chhagan Lal
EEMCS, Delft University of Technology, The Netherlands

27-6-2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

The Software Defined Network (SDN) is a relatively new paradigm that aims to tackle the lack of centralization in the existing network by separating the control centre from the programming data plane. The controller keeps an overview of the structure of the whole network, which makes it vulnerable to possible topology poisoning attacks. Topology attacks aim to disrupt the overview of the controller over the structure of the network in order to intercept or disrupt the transfer of the packages over the SDN network. In this paper, a survey on the state-of-the-art on topology attacks is conducted, followed by an analysis of the limitations of the existing solutions, and a comparison between the verification process of each solution and the number of known vulnerabilities are presented. Further, possible future research directions are proposed for improving these solutions and fixing the mentioned limitations and vulnerabilities.

1 Introduction

Software-Defined Network (SDN) is a relatively new paradigm in the networking sector [1][2][3], proposing an architecture that separates the control centre from the programming data plane [4] in a centralized manner. There are multiple improvements and deficits in this paradigm. One of the concerns regarding this new paradigm is the vulnerability to security threats [5]. The SDN paradigm has at its core the controller, which manages the whole network. The controller detects the structure of the network based on the Open Flow Discovery Protocol (OFDP), which is known to pose different security problems[6], such as the ability of attackers to poison the topology view of the controller over the network [7] by inserting fake links in it. Thus, if the controller would be compromised, the whole network would be hijacked, resulting in great concern for every connection to the network.

The security of a network is of utmost importance for the success of a networking paradigm since it is directly related to the reliability of the network. This research will investigate the impact of different types of topology attacks in the SDN scenario. First, a survey on the state-of-the-art attacks and proposed solutions are presented by identifying the different ways of launching a Topology attack and the impacted metrics. In the end, there will be an analysis of the limitations of the existing solutions proposed to solve the attacks in the state-of-the-art.

In the next section of the report, more in-depth background on the research topic will be presented, and will clearly state the existing related work on the subject of the research. The third section aims to present a state-of-the-art analysis from the literature review that was done on the related papers that were published in recent years. Then, a comparison of the current attacks and solutions proposed in the state-of-the-art will follow and possible future research directions on how to improve the information that exists on the topology attacks in the SDN will be given. The last section of the report incorporates the paper's conclusion, where the findings of the research will be summarised.

2 Related work

Multiple survey papers address the security issues related to the topology discovery protocol and the architecture of SDN. They are a great guideline to learn about the classic threats of this networking paradigm in terms of topology poisoning attacks and their solutions.

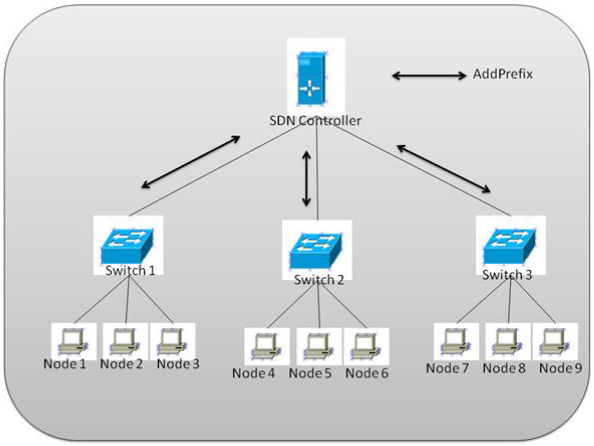

Figure 1: SDN Overview[8]

Figure 1, as it was presented in [8] represents an overview of the SDN topology structure. The paper [9] surveyed the general threats and solutions of topology discovery in the SDN. Security threats related to the Link Layer Discovery Protocol (LLDP) and how the Network Configuration Protocol (NETCONF) can fight against those vulnerabilities are discussed in [10]. Two novel attacks on the widely used Floodlight controller and six vulnerabilities in the famous countermeasure solutions for the classic attacks such as TopoGuard, TopoGuard+, SPV, and SecureBinder are discussed in [11]. The survey [12] discusses the state-of-the-art of the existing vulnerabilities of the SDN architecture and possible solutions to secure the SDN system. The paper [13] discusses the solutions proposed by the TopoGuard and Sphinx for the host-location hijacking and link fabrication attacks and proposes two new attacks that can bypass these solutions, namely Port Probing and Port Amnesia. In [14] the vulnerabilities of the OpenFlow Discovery Protocol (OFDP), such as discovery injection attacks, man-in-the-middle attacks, and topology discovery flood attacks are presented, and a new solution is proposed named Correlation-based Topology Anomaly Detection (CTAD). An analysis of link discovery service attacks on the controller layer is conducted in the paper [15]. The vulnerabilities of the OFDP discovery mechanism related to link spoofing attacks are discussed in the paper [16], and a solution to these attacks is proposed based on an HMAC authentication mechanism. An analysis of the fake link injection attacks and a countermeasure for them can be found in the paper [17] by comparing the number of packet losses on their new proposed model and the previous models that were designed.

3 Analysis of state-of-the-art

The state-of-the-art analysis was conducted by a thorough literature review regarding the topology attacks, the different ways through which one can be launched, and the metrics impacted, underlining the limitations of these attacks and the proposed solutions identified during the literature survey.

3.1 Attacks

Multiple vulnerabilities can lead to topology poisoning attacks on all levels of the network. As it was presented in the paper [9], the attacks can come from any plane of the network: the data plane can be attacked using malicious OF switches, malicious hosts in the data plane; the attacks on the control plane are the ones that are the most interesting for the attackers and can be conducted using malicious modules inside the controller, compromised controllers, or attack on management consoles. Figure 2 represents the overview of the topology of an SDN under the strike of a topology poisoning attack as represented in [18].

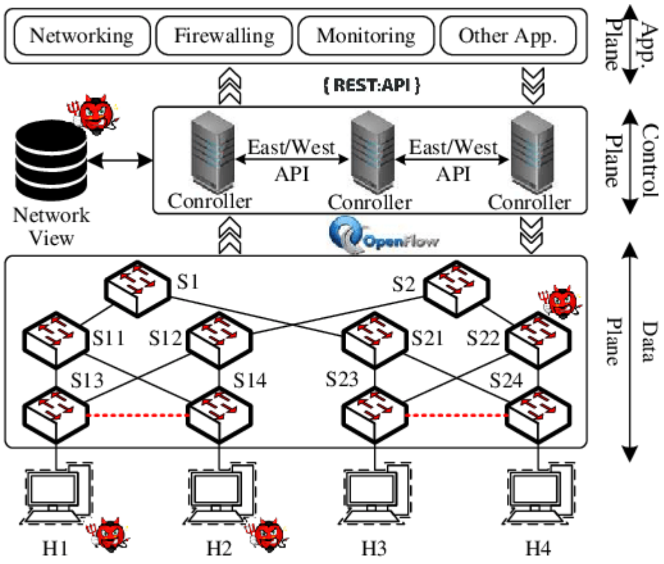

Figure 2: SDN overview under topology attack[18]

Silent Relay Attack

As described in [19], the Silent Relay Attack creates a fake link between a compromised port on a switch and another compromised port on a different switch in order to relay all incoming packets. This attack injects the fake link through the LLDP. It is hard to detect this type of topology attack because a compromised node can inject fake packets into the network or can relay the packets that are transferred over the LLDP without replying to any network probing since it behaves just as being part of a link.

Host-Location Hijacking

Similar to the Silent Relay Attack, the Host-Location Hijacking attack, as presented in [20], tricks the controller into believing that a victim's host was moved to a different location for as long as the host stays idle. Convincing the controller that the host was moved gives the opportunity to the attacker to send packets over the network using the IP and/or MAC address of the victim host.

Port Amnesia & Port Probing

These two attacks enable the classic Link Fabrication attacks and the Host-Location hijacking attacks to bypass the TopoGuard and SPHINX solutions[21].

Port Amnesia attacks rely on the fact that the attacker can control the OpenFlow messages by generating a Port-Down event to trick the TopoGuard security solution. This type of attack makes the controller forget the past classification of that specific port in the network structure. By using this technique, the Link Fabrication attacks can defeat the TopoGuard defence system.

The Port Probing attacks are used to enable the Host-Location Hijacking attacks. This type of attack is aimed at the host while it is transiting from one network location to another. During the transit period, the host's identifiers are not bound to any network. Therefore, the adversary can send various queries to a victim host and wait for the moment it goes offline; at that point, the Host-Location hijacking attack can be launched, dodging both the SPHINX and TopoGuard defence's system.

Topology Discovery Man-in-the-middle Attack

This type of attack requires two adversaries to be connected to the network as presented in [14]. The first attacker eavesdrops on the packets sent from the controller and then transfers them into a different format to the second attacker. The second attacker converts back the packages to the original format and forwards them to the switch, which will send a message to the controller to confirm it received it. By doing this, the adversary added a fake link between the two attackers. The effect of this attack is that the latency of the packets transported through the network increases, though if the adversary quickly plays back the packet, it is very hard to identify this type of attack.

Link Latency Attack

In the paper [22] the Link Latency Attack (LLA) is presented as follows: starting from the assumption that the adversary can compromise at least one host or virtual machine of the SDN through different methods, the attacker provides an out-of-band communication channel between two hosts to relay the LLDP packets. The attacker aims to provide a fake link between the switches in the network through the out-of-band channel. For this to result in a working attack that can bypass the existing solutions such as TopoGuard+, the adversary injects unwanted traffic in the LLDP to increase the processing time of the switches so he can add the fake link between switches. This type of attack impacts the Quality of Service of the SDN.

Persona Hijacking

The persona hijacking attack consists of two steps: IP takeover and flow poisoning [20]. The IP takeover step is achieved by convincing the DHCP server to provide the victim's IP address so the attacker can link his MAC address to it. The second step of the attack is required only if the DHCP server checks if the IP address is in use. Through this type of attack, the adversary aims to gain full access to the victim's identifiers in order to redirect all the traffic to himself.

3.2 Solutions

This subsection presents the proposed solutions for the attacks mentioned above, including the evaluation techniques used in the experiments proposed by the papers where they were introduced.

Silent Relay Detector

This solution was proposed to defend against the Silent Relay Attack proposed in [19]. This solution was built on top of the Floodlight controller, and its functionality is to listen to Topology updates of the SDN switches; in case the controller detects an inter-switch link, it will launch the verification process. The Silent Relay Detector is based on sending LLDP probe messages to verify the newly identified links that would result in the adversary generating significant time delays in relaying the packet but also making the attacker drop the large packets.

Since the solution was designed with this attack in mind, the fake link detection of the solution works on exploiting the adversary's capabilities. The idea of the solution is to make the adversary perform more offbeat than it would in order to trigger its detection by sending the probe LLDP probe messages aforementioned.

In the experiment there is an assumption that the MTU size of the host is 1500 bytes and that jumbo frames are not considered to be the default in the configuration of the network resulting that the large packets are being dropped by the specific host. It is also considered that the adversary will not be able to communicate with the network configuration to change it since it would unveil himself. The last assumption would be that the relay of a packet will induce extra latency since the packet is processed twice. For the experiment, the adversary was implemented as a raw socket program that aims to forward the packets received from a NIC through the fake link. The used implementation of the attack for the experiment was able to handle packets of up to 65536 bytes. The solution under test during the experiment was using the Floodlight controller in connection with Open vSwitches to form the data plane and a fake link between two switches.

In the implementation proposed in the paper, the latency of the LLDP probe messages was measured by the solution considering the time between the packet_out command and the packet_in request at the controller. There is a noticeable increase in the latency of packet passage between the regular and fake links directly connected to the size of the LLDP packet. The packet handler will drop the LLDP probe message when it exceeds the default MTU size. The Floodlight controller's log proves that the probe messages larger than its MTU will not reach the controller through the fake link but will do through a regular link. The limitation of the solution proposed is that once the adversary can handle a large-enough MTU, it is possible to relay the messages to the controller silently.

The impact of this solution is imposed on the Topology discovery process due to the fact that every time a new link is added, the solution will check whether it is a fake link or not. The check on the link should not significantly impact the time efficiency of the network because this check is conducted just the first time a new link is added to the network.

TopoGuard+

It appeared as an extension of TopoGuard in order to defend the system against port amnesia attacks as introduced in [13] and further discussed in [20]. TopoGuard+ also protects against link fabrication attacks based on relaying LLDP packets through the out-of-band channel. This solution protects against adversaries that have control over multiple hosts.

TopoGuard+ includes two new modules to the old TopoGuard system, namely the Control Message Monitor (CMM) and the Link Latency Inspector (LLI). The CMM verifies the traffic in case the Port-up/Port-down message is being transmitted during the progress of an LLDP packet making it safe against the port amnesia attacks. The LLI protects against fake link fabrication attacks by keeping track of the latencies of genuine links.

The CMM, as presented in [13] checks whenever an LLDP probe is in progress, and the controller detects this by receipt of any of the message types from a port involved in the LLDP probe, it will raise an alert. This check is applied retroactively to the receiving port for the period between the packet generation and receipt by logging the relevant messages in the controller. The Port-Up and Port-Down messages indicate a behavioural profile reset used by in-band port amnesia. The attacker ports repeatedly change their status from HOST to SWITCH to relay the LLDP traffic and originate data-plane traffic. An alert will be raised by bringing the interface down and up again. These checks stop the in-band port amnesia attacks, while the out-of-band port amnesia attacks are detected by the latency added in the relaying process that would not exist in a regular switch-to-switch connection. The adversary can still avoid CMM if it has access to an out-of-band channel through which it can relay the packets. However, it generates additional latency due to the signal propagation over the channel and the time needed for encoding and decoding the packets.

In the situation that the adversary manages to avoid the CMM, the LLI module of the TopoGuard+ should be able to identify the attack. LLI measures the latency of the switch-internal links during the LLDP propagations in order to flag the anomalies that may indicate the fabricated links. The latency between two target switches is measured by calculating the overall LLDP propagation time between them and the delays produced by the control links. The overall LLDP Propagation Delay is calculated using an extra optional field added to the LLDP packets during the link discovery procedure that stores the timestamp of the departure time of the packet in an encrypted format that later the controller can decrypt and use to compute the propagation delays. The Control Link Latency is calculated using echo messages that measure the round-trip delays between the controller and a switch by using packet-out messages to send probe messages to the target switch and set the next-hop to the controller. The controller computes the round-trip time delay by considering the elapsing time between the probe message being sent and the moment it arrived at the controller, and it is averaged over the last three measurements of the control links to minimize the variance. The link updates are verified considering the latency since it would increase abnormally if extra devices or channels are relaying the packets. In the situation that suspicious latency values of an internal switch link are found, the LLI raises an alert and may optionally block the topology update.

The evaluation of TopoGuard+ presented by [13] was a prototype of it in the Floodlight controller. More specifically, they extended the LinkManager application to inspect control messages during the propagation and measure the delays of the LLDP packets. Furthermore, they also added a new application to track real-time the latencies between the controller and the switches. They used Mininet to simulate the SDN testbed where all the data plane links were configured with 5ms latency and an out-of-band link between two compromised hosts with 10ms latency. The evaluation took place regarding security provided and the performance overhead introduced in the network.

In terms of the security evaluation, they first measured the latency of all the links, which proved consistent with the setup of the mock network environment to be at approximately 5ms. These low-latency measurements may have a consequence on the threshold value for detecting the fake links, which makes it easier to detect the fake ones. The maximum latency of those links provided micro-burst characteristics, which may introduce false positives for the solution, but it was considered to be tolerated in the controllers. At the startup of the controller, they recorded the measured link latencies and computed the threshold for the anomalous link detection. Then, they control the two compromised hosts to build up fake links through a side-channel one minute after the bootstrap of the controller. This experiment proves that TopoGuard+ effectively located all the fake links. The detection threshold was raised in the beginning because of the bootstrapping of the controller which added a significant delay to the measurement of the link latencies. However, this threshold converged once the controller reached a steady state, proving it could tolerate a small number of anomalous link latencies. In terms of the in-band port amnesia attacks, they launched multiple attacks, all of which were detected by TopoGuard+.

Considering the performance overhead added by the solution, they used the system.nanoTime Java API to measure the time stamps with the precision of 1 nanosecond. It was observed that the most significant overhead generated by TopoGuard+ was in the extra security inspections during the processing of the packets and not on any data plane operations. The paper shows that the overhead introduced by TopoGuard+ was of 0.299ms to the LLDP processing logic in the controller and of 0.134ms to the LLDP packet construction through the addition of the extra encrypted field containing the timestamp. This evaluation concluded that TopoGuard+ adds negligible overhead to the Floodlight controller.

The experiment proposed for the TopoGuard+ solution in [20] was run using a Mininet simulator built on the Raspberry Pi 3 Mold B switches acting as OpenFlow switches, and a controller ran on an Apple computer and several hosts. For the communication between the switches and the controller, they used the L2 Ethernet switch to forward the OpenFlow packets. The controller was based on the Floodlight controller because most topology defences were built on it while the switches were working on the Open vSwitch 2.5.5 LTS.

In the experiment of [20], there were identified two new vulnerabilities of the solution in the mechanism of tracking the latencies of the links but also in the LLDP packet generation. One of the exploits is based on overloading the switches in order for the latency between the inter-switch links to increase. This vulnerability can be present in the real-world SDN because the hardware switches in the SDN-enabled switches contain simple CPUs which restrict the capabilities of parsing and processing the packets. However, also, these switches have small flow table space that can accommodate up to a few thousand flow rules only, and their update rate is limited to 100-200 rules updates per second. These limitations are encountered because of the wire-speed packet processing achieved by the Ternary Content Addressable Memory (TCAM). The paper also proved that the hosts could overload the switches by triggering the controller into sending enough packets to overload the switches. The second limitation related to the generation of the LLDP packets is the lack of freshness in the generation of LLDP packets. The controller appends a MAC tag to all the LLDP packets, but the integrity of the packets proposed by the TopoGuard+ is not tackling the issue of reusing MAC tags for creating valid LLDP packets resulting in a vulnerability in stopping the port amnesia type of attack.

SecureBinder

This solution discussed in [20] and introduced by [23] is proposing a defence system for the port probing and persona hijacking attacks. It does it by modifying the authentication protocol in order to verify if the MAC addresses are valid. Authentication is now part of the controller's tasks. To verify the MAC addresses, the authenticator server connected to the controller contains a database that binds all the MAC addresses to their certificates. This solution also binds all the control traffic to the controller instead of being broadcast through the network, which results in a prevention measure against adversaries that try to sniff the packets.

According to [23] this solution leverages SDN and IEE 802.1x to target the facilitating factors of the persona hijacking attack. SecureBinder supports the SDN in separating the identifier binding control traffic from the regular data plane, protecting it from the adversaries, and creating a binding mediator which can perform additional security checks on identifier bindings. This approach enables the SDN to not trust the insecure protocols for identifier bindings anymore. It validates the changes of the identifier bindings by leveraging the global view of the network and the identifiers provided by the SDN, thus leveraging the distinguishment between the creation of new bindings and changes to existing ones unless there is a confirmation that the old binding is no longer active. This solution also prevents illegal binding changes at higher layers by using lower layer bindings to validate messages. It also protects against readily changed but supposedly unique identifiers by supporting IEEE 802.1x to provide a root-of-trust for network identifiers by binding the MAC address to the authentication process and eliminating the disconnected host race conditions.

The solution makes use of the multiple flow tables in OpenFlow 1.3 as follows: the first table (table 0) separates identifier binding traffic from regular data-plane traffic, and the rest of the tables are used for routing and other applications as usual. Then on table 0, egress filtering is applied by inserting flow rules into the table such that flows with expected source identifiers (MAC and IP addresses) are sent directly to table 1 to be routed normally, while all other traffic is rate-limited and sent to the controller. By doing this, SecureBinder is acting as a privileged SDN controller application which configured itself to handle all packet_in events before any other application and looks for packets sent to it as a result of the set rules in table 0. Any identifier binding traffic is validated and used to update the binding information and sent to the relevant applications. In contrast, any other packets sent to the controller based on the rules in table 0 will be logged and dropped.

The authors of [23] proposed an emulated SDN testbed network using Mininet 2.2.1 with Open vSwitch 2.4.0 switches on which three separate identifier binding attacks are being launched at a controller of type ONOS 1.5.1 with and without SecureBinder. They repeated the attacks ten times to ensure valid results. The topology used in the evaluation was having a single switch of three hosts (an adversary, a victim, and a user that wanted to reach out to the victim). The experiments were run in an Ubuntu 14.04.4 VM with 2 cores of an Intel i7 CPU 2.70GHz and 15GB of RAM. The ONOS 1.5.1 controller provides shortest-path routing, proxyARP, and DHCP. The attacks launched were: persona hijacking attack, ARP poisoning, and host location hijacking attack; SecureBinder identified all these three attacks, and an alert was thrown while the attacks were blocked immediately. The evaluation of this solution also indicated that it increases the latency of the host join, which refers to the latency for a host to join the network and the network link detection, DHCP negotiation, 802.1x authentication by 3 seconds due to the 802.1x authentication and additional flow rule insertions required by the solution. During the experiments, it was also noticed that the number of packet_in messages sent to the controller increased by 47%. The SecureBinder solution requires additional flow rules, which are a limited resource in OpenFlow switches; the number of additional flow rules is equal to: $26 + 13 * edge\_ports + internal\_ports$.

While in the discussion of the solution by [20] a different experiment was run where two new vulnerabilities were identified. Consider that the authors used the same setup for the system as the one presented in the discussion of TopoGuard+.

An issue of this solution would be that the authentication process takes place only when connecting a host to the network or a host changes its location disregarding the packets sent over the network. This vulnerability could result in a security breach considering that the solution is based on detecting the possible threats solely on the port-down and port-up messages. The attack proposed for exploiting this vulnerability could not be implemented using the Mininet simulation because it works differently than the real OpenFlow switches. Considering that the security of the solution is based on the port-down and port-up notifications, it would result in the future in a security breach which should be avoided because there is the opportunity of disconnecting an authenticated host and connecting a malicious one without the switch noticing the difference if the swap is happening fast enough.

The second vulnerability of this solution identified was related to the fact that the bindings between the MAC addresses and the host location in terms of connection to the switch and the port in the network are not taken into account in the verification process. The vulnerability is presented based on an attack model that aims to connect a new host replaying the victim host closer to the controller in order to convince the real host is at the closest location (malicious host's location). This attack would not only intercept all the traffic of the legitimate host but could also make the real host unreachable since the controller would think that the malicious host is the real one.

Correlation-based Topology Anomaly Detection

In [14] the Correlation-based Topology Anomaly Detection (CTAD) mechanism in the controller is proposed as a solution for the Topology Discovery Man-in-the-middle attack. This mechanism consists of three modules: Topology Management module, LLDP Handling module, and Correlation-based Topology Anomaly Detection module.

The Topology Management module monitors the LLDP packets received from the switches and maintains the real-time network topology information. In addition, this module is responsible for identifying injection attacks and the topology discovery flooding attacks.

The LLDP Handling module converts the LLDP packets into a specific format generating the verification information and encapsulating it into the packet that will be sent to the specific switch.

Lastly, the correlation-based Topology Anomaly detection module measures the time of the round trip time for each packet and analyzes the correlation of the traffic of the network for each link to identify the topology discovery man-in-the-middle attack.

The mechanism is composed of the three modules mentioned above, and the process would look like this: the Topology Management module converts the LLDP packets to link information and verifies the current topology; in the case that the link is already in the topology information, it is transferred to the Time-Difference Analyzer in the Topology Anomaly detection module; otherwise the mechanism is started for topology discovery attack analysis. Then, the time difference analysis and the correlation analysis are run on the network to identify whether there is a topology discovery man-in-the-middle attack. These checks are run as follows: the round-trip time of the transfer of a LLDP packet is calculated, and after that calculates the shortest path using the Dijkstra algorithm between two switches to only filter the round-trip time of the packets sent via the shortest path; followed by the correlation-analysis are run by sending a large number of random rate LLDP packets to the specific port of the switch collecting all the traffic of the ports of the specific switch, thus, by performing Spearman's rank correlation coefficient analysis on the collected traffic of the source port and all the target ports results in the correlation coefficient. If the obtained correlation coefficient is greater than 0.7 will indicate that there is a topology discovery man-in-the-middle attack.

The experiment proposed in the paper is based on a Mininet simulation of 8 Open vSwitches and 40 Host built on an ASUS/RS100-E9-PI2 server as the hypervisor, with a Ryu controller provided by Ryu SDN Framework Community that uses OpenFlow version 1.3. There will be two hosts that will simulate the adversaries for the experiment. The LLDP generator, TCPdump, and TCPreplay are used for the topology discovery attacks. In the experiment, they compared the CTAD to the statistical analysis of link latencies(SALL) solution proposed by [24] and KHMAC[7] authentication mechanism solution. After simulating the man-in-the-middle attack, the CTAD generated LLDP packets with different sending rates for being able to do the correlation analysis. The Correlation Analyzer proposed by the CTAD managed to detect this type of attack of high-speed replay LLDP packet starting from a 40ms replay time which is a great improvement compared to the detection time of SALL which was 100 ms, and of KHMAC with 900 ms.

4 Discussion and future research directions

4.1 Discussion on the solutions

Considering the identified types of attacks and solutions proposed in the state-of-the-art, Table 1 represents a comparison overview of the different types of topology attacks.

As it was previously presented, the attacks are aiming for different vulnerabilities of the SDN, thus the solutions proposed are performing checks on different parts of the SDN. Therefore, the presented known vulnerabilities in the table are not to be considered a complete list since further research is needed to establish the severity of the known vulnerabilities and identify possible new attacks. Furthermore, more research is needed to confirm that the two solutions considered not to have any vulnerabilities are entirely secure.

4.2 Future research

Considering the identified types of attacks and their proposed solutions in the state-of-the-art, multiple research directions can help in the prevention of serious topology poisoning threats in the SDN paradigm.

An excellent research opportunity would be further researching the impact of the Silent Relay Detector solution on the discovery process and how it can be reduced thus it could increase the usability of the solution in an extensive SDN.

Other points of possible future research would be regarding the vulnerabilities identified for TopoGuard+. One of the possible directions of the research would be identifying possible upgrades in the switches in order to convert the simple CPUs to more complex ones that can combat the processing limitations of the current switches. A secondary research could aim to identify a way to remove the old MAC tags from memory in order to prevent the reusability of the tags. A possibility to remove the old MAC tags would be that once the host connected to that MAC tag goes offline, the list of certified MAC addresses to be updated to contain only the active ones.

As mentioned in the solution subsection of the paper, the SecureBinder solution could benefit from identifying the impact on the memory consumed by the network by extending the list of bindings of the host to include as well the location of the connection (to which switch and port is the host connected). Considering that the impact of this addition to the solution would not increase by a substantial amount the memory needed to be allocated to the SDN it could help the solution to tackle one of the known vulnerabilities and make the SDN more secure.

A possible improvement to the existing solutions could be the result of combining the TopoGuard/TopoGuard+ with the SecureBinder in order to create a universal solution to tackle the different risks imposed by the topology poisoning attacks. Furthermore, this combination would make it easier for the SDN users to better protect their networks with a standard solution instead of having to implement multiple separate solutions and ensure they are keeping their security system up to date. This combination could be possible if a partnership is established between the developers since this should not be overly complicated to achieve since both solutions are already tackling the problem of the MAC address validation.

5 Responsible research

This paper aims to survey the different types of topology attacks in an SDN. The survey was conducted on state-of-the-art attacks and solutions proposed in reliable conference papers published and cited by different authors. By using reliable sources, there are clear descriptions of how the attacks and solutions proposed are functioning, making sure it offers the users of the SDN networks the opportunity to protect their system against the known types of attacks.

Even though this paper may offer an adversary a hand of help in further improving his attacks, it also offers the opportunity for the users to protect their system accordingly and also the researchers the possibility to continue the development of the solutions in order to cover the limitations of the existing solutions.

6 Conclusion

This paper surveyed the existing types of topology attacks and the proposed solutions to these attacks on the state-of-the-art. In the first part of the paper, background about the Software Defined Network paradigm was introduced and explained what the topology attacks influence on the network. Then, the identified solutions were compared based on the verifications they are processing in order to detect the specific attacks and the known vulnerabilities present. Finally, further research directions were presented to optimise the solutions' efficiency and solve the solutions' known vulnerabilities.

Solution	Attack(s) solved	Description	Checks conducted	Known vulnerabilities
Silent Relay Detector	Silent Relay Attack	It sends probe messages of different sizes on the newly identified links.	It verifies the latency of the transfer of the packets, more exactly between the packet_in and packet_out messages, but also possible drops of the large packets.	None known yet
TopoGuard+	Link fabrication attacks Port Amnesia Host-Location Hijacking	Extension of the TopoGuard solution aiming to protect against different types of attacks.	Verifies the traffic at the moment of receiving a Port-up/Port-down message during the transfer of a LLDP packet, and also constantly keeps track of the latencies of the links in the network.	Mechanism to track link latencies LLDP packet generation
SecureBinder	Port Probing Persona Hijacking Host-Location Hijacking ARP poisoning	Modifies the authentication protocol in order to verify the MAC addresses by binding them to their certificates.	Verifies the MAC addresses of the hosts that are connected to the SDN.	Possibility to disconnect a good host and connect a malicious one if quick enough Lack of detail in the bindings, more exactly the hosts location and the traffic not binded
Correlation-based Topology Anomaly Detection	Topology Discovery Man-in-the-middle attack Topology Discovery Injection attack Topology Discovery Flooding attack	Composed by three modules that aim to tackle the different attacks. It detects the attacks based on a correlation between the network traffic based on the links by using Spearman rank correlation coeficient.	Verifies the latency of the LLDP packet replays and makes a correlation with the network traffic between the links.	None known yet

Table 1: Comparison between the different solutions identified

References

[1] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[2] Y. Jarraya, T. Madi, and M. Debbabi, "A survey and a layered taxonomy of software-defined networking," *IEEE communications surveys & tutorials*, vol. 16, no. 4, pp. 1955–1980, 2014.

[3] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openflow: From concept to implementation," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2181–2206, 2014.

[4] E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, "Software-defined networking (sdn): Layers and architecture terminology," *RFC 7426*, 2015.

[5] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in software defined networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2317–2346, 2015.

[6] A. Azzouni, N. T. Mai Trang, R. Boutaba, and G. Pujolle, "Limitations of openflow topology discovery protocol," in *2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pp. 1–3, 2017.

[7] T. Alharbi, M. Portmann, and F. Pakzad, "The (in)security of topology discovery in software defined networks," in *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, pp. 502–505, 2015.

[8] T. Guesmi, A. Kalghoum, B. M. Alshammari, H. Alsaif, and A. Alzamil, "Leveraging software-defined networking approach for future information-centric networking enhancement," *Symmetry*, vol. 13, no. 3, 2021.

[9] S. Khan, A. Gani, A. W. Abdul Wahab, M. Guizani, and M. K. Khan, "Topology discovery in software defined networks: Threats, taxonomy, and state-of-the-art," *IEEE Communications Surveys Tutorials*, vol. 19, no. 1, pp. 303–324, 2017.

[10] S. Popic, M. Vuleta, P. Cvjetkovic, and B. M. Todorović, "Secure topology detection in software-defined networking with network configuration protocol and link layer discovery protocol," in *2020 International Symposium on Industrial Electronics and Applications (INDEL)*, pp. 1–5, 2020.

[11] S. Popic, M. Vuleta, P. Cvjetkovic, and B. M. Todorović, "Secure topology detection in software-defined networking with network configuration protocol and link layer discovery protocol," in *2020 International Symposium on Industrial Electronics and Applications (INDEL)*, pp. 1–5, 2020.

[12] K. Raghunath and P. Krishnan, "Towards a secure sdn architecture," in *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–7, 2018.

[13] R. Skowyra, L. Xu, G. Gu, V. Dedhia, T. Hobson, H. Okhravi, and J. Landry, "Effective topology tampering attacks and defenses in software-defined networks," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 374–385, 2018.

[14] L.-D. Chou, C.-C. Liu, M.-S. Lai, K.-C. Chiu, H.-H. Tu, S. Su, C.-L. Lai, C.-K. Yen, and W.-H. Tsai, "Behavior anomaly detection in sdn control plane: A case study of topology discovery attacks," in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 357–362, 2019.

[15] T.-H. Nguyen and M. Yoo, "Analysis of link discovery service attacks in sdn controller," in *2017 International Conference on Information Networking (ICOIN)*, pp. 259–261, 2017.

[16] T. Alharbi, M. Portmann, and F. Pakzad, "The (in)security of topology discovery in software defined networks," in *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, pp. 502–505, 2015.

[17] N. Kaur, A. K. Singh, N. Kumar, and S. Srivastava, "Performance impact of topology poisoning attack in

sdn and its countermeasure," in *Proceedings of the 10th International Conference on Security of Information and Networks*, SIN '17, (New York, NY, USA), p. 179–184, Association for Computing Machinery, 2017.

[18] A. Alimohammadifar, S. Majumdar, T. Madi, Y. Jarraya, M. Pourzandi, L. Wang, and M. Debbabi, *Stealthy Probing-Based Verification (SPV): An Active Approach to Defending Software Defined Networks Against Topology Poisoning Attacks: 23rd European Symposium on Research in Computer Security, ESORICS 2018, Barcelona, Spain, September 3-7, 2018, Proceedings, Part II*, pp. 463–484. 08 2018.

[19] P. Shrivastava, A. Agarwal, and K. Kataoka, "Detection of topology poisoning by silent relay attacker in sdn," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, MobiCom '18, (New York, NY, USA), p. 792–794, Association for Computing Machinery, 2018.

[20] E. Marin, N. Bucciol, and M. Conti, "An in-depth look into sdn topology discovery mechanisms: Novel attacks and practical countermeasures," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, (New York, NY, USA), p. 1101–1114, Association for Computing Machinery, 2019.

[21] R. Skowyra, L. Xu, G. Gu, V. Dedhia, T. Hobson, H. Okhravi, and J. Landry, "Effective topology tampering attacks and defenses in software-defined networks," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 374–385, 2018.

[22] S. Soltani, M. Shojafar, H. Mostafaei, Z. Pooranian, and R. Tafazolli, "Link latency attack in software-defined networks," in *2021 17th International Conference on Network and Service Management (CNSM)*, pp. 187–193, 2021.

[23] S. Jero, W. Koch, R. Skowyra, H. Okhravi, C. Nita-Rotaru, and D. Bigelow, "Identifier binding attacks and defenses in {Software-Defined} networks," in *26th USENIX Security Symposium (USENIX Security 17)*, pp. 415–432, 2017.

[24] D. Smyth, S. McSweeney, D. O'Shea, and V. Cionca, "Detecting link fabrication attacks in software-defined networks," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–8, IEEE, 2017.