Deception-based IDS against ARP Spoofing Attacks in Software-Defined Networks

Fabrice Mvah
Mathematics and Computer Science
University of Dschang
PO Box 67, Dschang, Cameroon.
fabricemvah@gmail.com

Vianney Kengne Tchendji
Mathematics and Computer Science
University of Dschang
PO Box 67, Dschang, Cameroon.
vianneykengne@yahoo.fr

Clémentin Tayou Djamegni
Mathematics and Computer Science
University of Dschang
PO Box 67, Dschang, Cameroon.
dtayou@yahoo.com

Ahmed H. Anwar
DEVCOM Army Research Laboratory
Adelphi, MD 20783, USA
a.h.anwar@knights.ucf.edu

Deepak K. Tosh
Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA
dktosh@utep.edu

Charles Kamhoua
DEVCOM Army Research Laboratory
Adelphi, MD 20783, USA
charles.a.kamhoua.civ@army.mil

***Abstract*—Address Resolution Protocol (ARP) spoofing is an important means for cyber adversaries to launch Denial of Service (DoS) or Man-In-The-Middle (MITM) attacks. These attacks can seriously impact system performance regarding confidentiality and data integrity. To overcome ARP spoofing attacks, several intrusion detection systems (IDSs) leverage the advantages of software-defined networking (SDN) to detect attackers. However, most of these approaches remain ineffective due to the use of a non-adaptive threshold and the lack of real-time information during attacker detection. To remedy these shortcomings, we propose a new deception-based IDS to efficiently detect attackers in SDN networks. This method deceives attackers to obtain real-time information to improve the detection system. Simulation results in the Mininet simulator show that the proposed method can significantly mitigate ARP spoofing attacks better than existing approaches.**

***Index Terms*—ARP Spoofing, Software-Defined Network, Cyber Deception.**

I. Introduction

Address Resolution Protocol (ARP) spoofing is a form of attack usually used by attackers to negatively impact system performance regarding confidentiality and data integrity [1]. These attacks come from the address resolution protocol, which is used to initiate communications in the network by providing a mapping between Internet Protocol (IP) and Media Access Control (MAC) addresses [2]. The ARP protocol suffers from several weaknesses such as lack of authentication (no mechanism for the receiving host to authenticate the packet sender) and its stateless nature (a host can send an ARP response without an associated ARP request) [3]. The exploitation of these weaknesses may allow attackers to impersonate other hosts and poison ARP caches. An attacker can poison an ARP cache by inserting multiple IP addresses for one MAC or multiple MAC addresses for one IP. To overcome ARP spoofing attacks, several approaches such as Naive Bayes [4], BSVR-ARP [5], E2BaSeP [6] and GaTeBaSeP [7] leverage the advantages of software-defined networking (SDN) to detect attackers. However, these approaches can be ineffective because they use a non-adaptive threshold to detect attackers. Indeed, the detection thresholds used do not depend on the attack frequency. Therefore, these approaches can lose their effectiveness when the attack frequency fluctuates. To overcome this drawback, we propose a new intrusion detection system (IDS) based on deception. This method deceives attackers to obtain real-time information (attack frequency, number of attacks at a given time, etc.) to improve the detection system. Our contributions can be summarized as follows:

- Propose a new dynamic detection threshold based on information collected from decoys;
- Propose a deception-based IDS that considers the attack frequency fluctuation during attacker detection.

The rest of this paper is organized as follows: Section II presents related works. Section III presents the network architecture. In Section IV, the threat model is described. Section V, presents the proposed approach against ARP spoofing attacks. Section VI validates this approach through simulations done in the Mininet simulator. Section VII concludes this paper and presents future directions.

II. Related Work

Several approaches based on flow graphs, traffic pattern analysis, and IP-MAC address binding have been proposed to defend ARP spoofing attacks [3]. For instance, a flow graph is a graph theory representation of switches (nodes) and flow metadata (edges) that provides a simple and easy mechanism to detect violations of network topology restrictions [8]. Approaches based on flow graphs proposed in [8], [9] assume that the OpenFlow messages sent by SDN switches are trustworthy. However, a compromised switch can add false information on various flows and distort the flow graph. Furthermore, if the topology is dynamic and changes frequently, the flow properties (links, paths followed, bytes transferred, etc.) may

979-8-3503-7099-7/24/$31.00 ©2024 IEEE

also change frequently and cause false alarms in the network. The main idea of approaches based on traffic pattern analysis is to use the controller in the SDN network to monitor and analyze ARP traffic. The controller monitors the traffic and detects an ARP spoofing attack when the ARP traffic analysis result meets a certain predefined traffic pattern [10], [11]. But, the network administrator must accurately identify predefined traffic patterns to avoid false alarms. Another way to handle ARP spoofing attacks is the IP-MAC address binding [6], [12], [13]. This method uses a global ARP mapping into the controller to answer ARP requests in the network. However, most of the approaches based on IP-MAC address bindings can be ineffective due to the use of a non-adaptive threshold and the lack of real-time information during attacker detection. In this paper, we address these shortcomings by using a deception-based method to efficiently detect attackers. Section IV describes the ARP spoofing threat model.

III. Network Architecture

We consider SDN because of its ability to collect information and program routing devices through applications, enabling proactive and intelligent security policies [14]. This architecture separates the control and data planes to represent the entire lower-level network infrastructure as an abstraction of the higher-level control and management functions implemented in the SDN controller. The SDN technology is recognized for its adaptability to large-scale networks and can enable global configuration across the network, facilitating overall network management [15]. Fig. 1 presents an SDN architecture in which the proposed approach is implemented. In this figure, SDN architecture comprises a controller, Open-

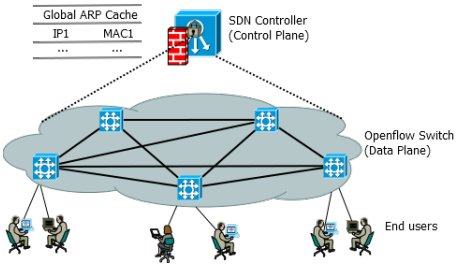
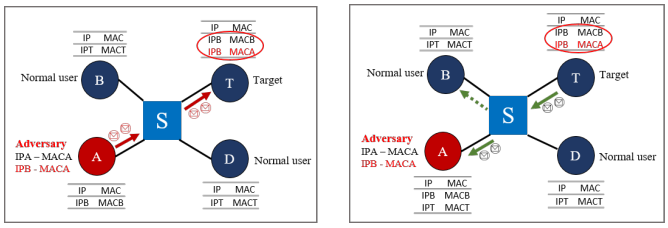


Fig. 1: SDN architecture [6]

Flow switches, and end users. The controller uses a global ARP cache to answer virtual machines' ARP requests. Virtual machines communicate with the controller through the OpenFlow protocol [6]. Section IV describes the ARP spoofing threat model.

IV. Threat Model

We consider the threat model where the attacker starts with the network reconnaissance phase before launching ARP spoofing attacks. Network reconnaissance is a preliminary step in which an attacker attempts to gather information (IP and MAC addresses, services, etc.) about a system using scanning tools such as Nmap and Xprobe2 [16]. This information is used by the attacker to target its victims. After reconnaissance, the adversary can launch ARP spoofing attacks to poison victims' ARP caches, as shown in Fig. 2. In this figure,



(a) Adversary A poisons host T's ARP cache.

(b) Packets from T destined to B are redirected to A.

Fig. 2: ARP spoofing attack scenario.

adversary A targets host T and poisons its ARP cache by impersonating host B (Fig. 2a). Therefore, traffic from T intended to B is redirected to A (Fig. 2b). A host is considered a non-attacker if it is identified by one and only one IP-MAC pair in the network. The attacker's characteristics are the following [6]:

- An attacker can send an ARP packet that leads to obtaining several IP addresses for a MAC address or several MAC addresses for an IP address in the network;
- The MAC address of an ARP packet header sent by an attacker is often different from the data link layer MAC address;
- The number of ARP requests sent by an attacker is often greater than the number of ARP responses received.

Section V presents the proposed defense mechanism against ARP spoofing attacks.

V. Proposed Approach Against ARP Spoofing Attacks

We consider a network comprising true and deception configurations. The true configuration includes the set of virtual hosts denoted by H. The deception configuration aims to hide valuable hosts using the set of decoy nodes denoted by $\tilde{H}$ ($H \cap \tilde{H} = \emptyset$). Decoy nodes collect information about users to enable the defender to determine the accurate attack frequency at a given time. As network transmission errors or virtual machine migrations may lead to modifying a normal packet to an attack one, we use the Bayes theorem to detect attackers.

A. Attacker detection algorithm

We consider P as the probability of being an attacker and $\overline{P}$ the probability of being a non-attacker ($P = 1 - \overline{P}$). Let S be the set of attacker characteristics. When the SDN controller detects the characteristic $s \in S$ from a packet of a node h, it computes the probability $P(h|s)$ using Eq. 1. We use probability to detect whether a host is an attacker because network configuration changes can modify a correct packet into an attack one.

$$P(h|s) = \frac{P(h) * P(s|h)}{P(h) * P(s|h) + P(\overline{h}) * P(s|\overline{h})} \quad (1)$$

Eq. 1 assumes that attacker characteristics appear independently of each other. In case some characteristics are mutually dependent, the SDN controller may compute the virtual host probability considering a subset of characteristics. Let $\theta = \{s_1, s_2\}$, the subset of two mutual dependent characteristics triggered by h. To compute the probability $P(h)$, Eq. 2 will be used, instead of Eq. 1.

$$P(h|\{s_1, s_2\}) = \frac{P(h) * P(\{s_1, s_2\}|h)}{P(h) * P(\{s_1, s_2\}|h) + P(\overline{h}) * P(\{s_1, s_2\}|\overline{h})} \quad (2)$$

Assuming that the characteristics are mutually independent, using Eq. 2 to calculate the probability $P(h)$ will provide the same result as Eq. 1. As we have shown how to compute the probability $P(h)$ that the virtual host h is an attacker, we need to define the threshold probability. This threshold considers real-time attack frequency from the decoys. Since attack frequency can vary, the detection threshold must adapt to this variation. Let λ be the attack frequency. We define the detection threshold based on the detected attackers (A), virtual machines recovered from the list of attackers (Ω), and the attack frequency (λ). Let δ be the period between two consecutive attacks. $\lambda = \frac{1}{\delta}$. Let us consider m as the number of decoy nodes in the network and N, the number of intervals between consecutive attacks. The average attack frequency can be defined as follows:

$$\lambda_t = \frac{1}{m}(\sum_{i=1}^{m}(\frac{1}{N}\sum_{j=1}^{N}\delta_j)) \quad (3)$$

Finally, the detection threshold is defined by Eq. 4.

$$\tau_t = \frac{A}{\lambda_t - \Omega} = \frac{A}{\frac{1}{m}(\sum_{i=1}^{m}(\frac{1}{N}\sum_{j=1}^{N}\delta_j)) - \Omega} \quad (4)$$

In Eq. 4, Ω refers to the virtual machines recovered from the list of attackers. We consider that there can be an error during the attacker detection. Thus, after connecting the attacker to the decoy through flow rules, the SDN controller continues to observe the interaction of the attacker with the decoy. If no malicious behavior is detected in the decoy information, the virtual host is reconsidered as a normal user. Ω is the set of previous attackers reconsidered as normal users.

B. Deception model description

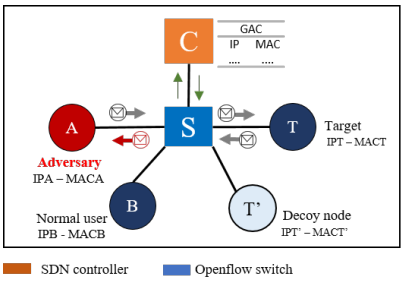
The deception model aims to obtain real-time information to improve the detection system. This improvement is achieved through the use of an accurate detection threshold based on the data collected from the decoy nodes. To deceive adversaries, the SDN controller intercepts scan packets and rewrites their headers by mutating real hosts' addresses with the decoy ones as shown in Fig. 3a. In this figure, the adversary A sends a scan request to the target T, and the latter receives this request and returns a scan response. However, the defender intercepts this response, rewrites its header by mutating the target's IP-MAC pair to that of the decoy, and forwards it to the adversary. Thus, if the adversary targets its victim based on the IP-MAC pair contained in this response, it will be deceived into believing that its attack succeeded. Similarly, the defender can also rewrite the scan request headers and forward them to decoy nodes as in Fig. 3b. This figure shows that the adversary A sends a scan request to the target T, the defender intercepts this request, rewrites its header, and forwards it to the decoy node T'. The latter receives this request and returns a scan response containing its $IP_{T'} - MAC_{T'}$ pair. Therefore, if the adversary uses this pair to launch an ARP spoofing attack, it will be deceived.

As for scanning requests, when the adversary launches an ARP spoofing attack by sending a fake ARP request to the victim, the defender intercepts this request and returns an ARP response as in Fig. 3c. This response includes the decoy's IP-MAC pair instead of that of a real host. In Fig. 3c, adversary A sends an ARP request to the target T by impersonating host B, and the defender intercepts this request and answers with an ARP response containing the decoy's IP-MAC pair. When the adversary receives this response, it believes that its attack was successful, whereas it has been deceived. Similarly, instead of directly answering the adversary's ARP requests, the defender can also rewrite an ARP request header by mutating the victim's IP address to the decoy's one and forward this request to the decoy node as in Fig. 3b. In this figure, the decoy node T' receives an ARP request and returns an ARP response containing its IP-MAC pair. When the adversary receives this response, it will update its ARP cache with the decoy's IP-MAC pair and will be deceived into believing that its attack succeeded. To prevent an attacker from suspecting a deception, the decoy node can mimic the target's traffic by sending packets to the adversary. Thus, when the target sends packets to the impersonated victim, the decoy node also generates fake packets and sends them to the adversary. In addition, the defender should avoid poisoning the opponent's ARP cache because, if the attacker finds multiple IP addresses for a MAC address, or multiple MAC addresses for an IP address in its ARP cache, it may suspect a deception.

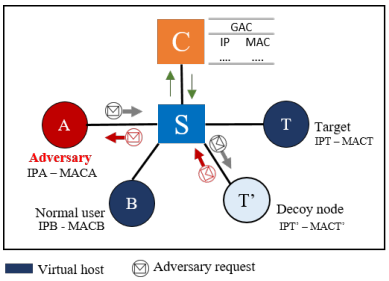
In the deception mechanism described previously, we have considered a network with four nodes, an OpenFlow switch, and the SDN controller. This network is chosen to show how the SDN controller redirects the adversary to the decoy. In reality, the network may have several switches, target nodes, decoy nodes, attackers, and normal nodes.

VI. Simulation

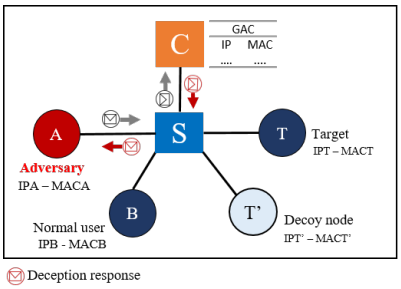
Simulations were performed in the Mininet 2.2.2 tool. This tool is useful because the source code of a network prototype developed in this tool can be reused in a network based on real hardware without any code modification. To implement the proposed approach, we created an SDN network comprising a Python-based controller (Pox), 10 OpenFlow switches, 100 hosts, and a DHCP server. The DHCP is configured from the service *isc-dhcp-server*. The set of hosts includes decoy nodes and real hosts. The SDN controller deceives detected attackers by installing flow rules to redirect their traffic to decoys. To simulate ARP spoofing attacks, we configure each virtual host



(a) Scan response header rewriting by the defender to include deception.



(b) Request header rewriting by the defender to redirect the attacker to a decoy node.



(c) The defender answers to the adversary's request by including deception in the response.

Fig. 3: Deception mechanism against network reconnaissance and ARP spoofing attacks.

to change its network configuration and send spoofed ARP requests to the SDN controller. A host sends ARP requests with a probability (γ) of triggering an attacker characteristic. We also used the tool *arpspoof* from the package Dsniff to enable a host to send fake ARP responses. Overall, we simulated attacker behaviors by sending spoofed requests and responses to the SDN controller. When virtual machines launch attacks, the SDN controller runs the proposed algorithm to detect attackers in the network. We compare this algorithm with existing approaches [4]–[6]. We choose the approaches Naive Bayes [4], BSVR-ARP [5], and E2BaSeP [6] as benchmarks because they are the most recent approaches that consider network configuration changes due to transmission errors or virtual machine migration. Indeed, network configuration changes can lead to the modification of a correct packet into an attack packet during data transfer. We consider these configuration changes to mitigate misjudgments in the network and improve detection accuracy. We compared the proposed method to other ones on the detection precision. The result obtained from this comparison is shown in Fig. 4. In these results, the proposed approach is called DbaD (Deception-based Defense). In this figure, the proposed approach has

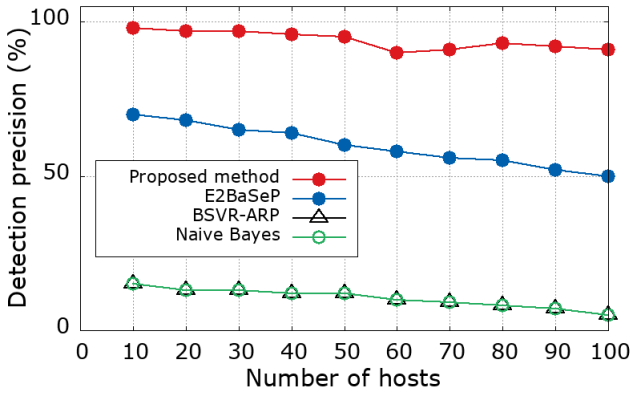


Fig. 4: Detection precision.

better detection precision than existing methods because the SDN controller recovers certain virtual nodes that were already in the list of attackers. This recovery process is based on the data obtained from the decoys. These data are also used to improve the accuracy of the detection threshold. Fig. 4 shows that the proposed model has an average precision of 96% compared to the E2BaSeP, which has a precision of 50%, and the BSVR-ARP and Naive Bayes whose precision is less than 15%. Indeed, the proposed method includes a dynamic threshold that can adapt to both high- and low-attack frequency environments. Using an adaptive threshold can also help to mitigate misjudgments in the network. We compared the false positive generated by the proposed model to that of the existing ones. Fig. 5 plots the result of this comparison. This figure

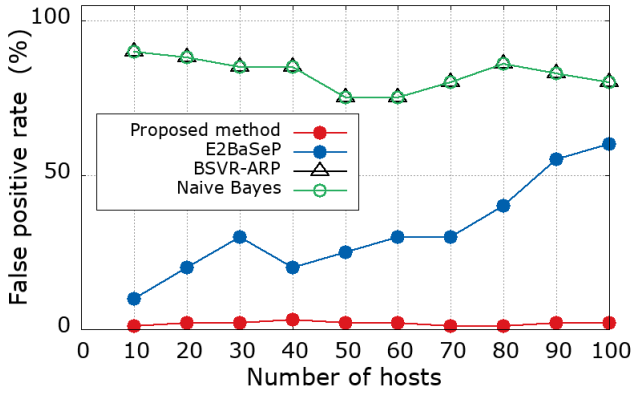


Fig. 5: False positive rate under attack scenarios related to multiple IPs for one MAC (s^*).

shows that the false positive rate of existing methods is greater than 50%, while that of the proposed approach is less than 5%. The deception method has a lesser false positive rate because it calculates the detection threshold based on real-time information collected from decoys.

We also evaluated the deceived adversary rate over the number of adversaries. The results obtained are given in Fig. 6. This figure shows that the proposed method has deceived more than 85% of adversaries. In the same way, the number of targeted real hosts has also been evaluated and the results are given in Fig. 7. This figure shows that the number of real

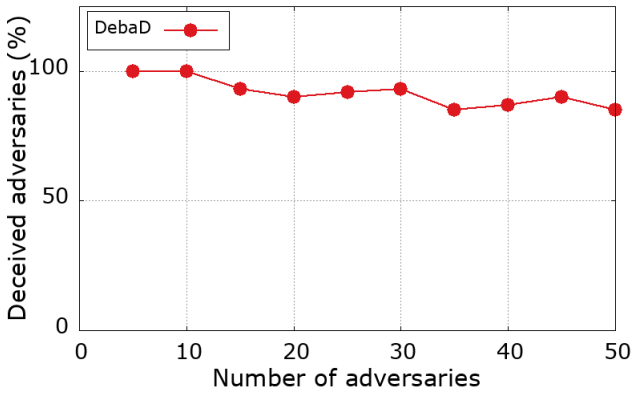


Fig. 6: Deceived adversary rate.

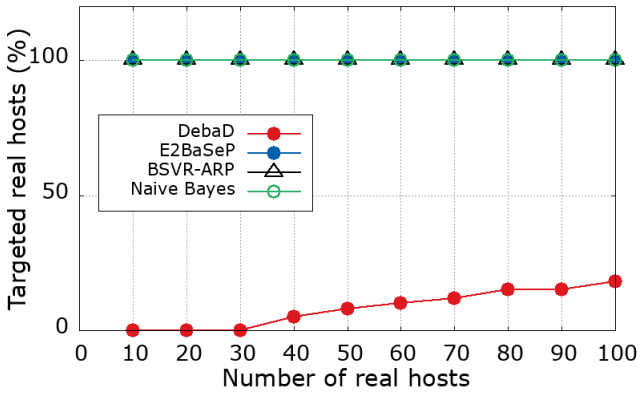


Fig. 7: Targeted real host rate.

hosts targeted by adversaries is less than 20%. Therefore, we conclude that the proposed model can prevent attackers from targeting real hosts in the network.

VII. Conclusion and Future Work

The aim of this paper was to propose an effective approach for software-defined network security against ARP spoofing attacks. The proposed method is a deception-based IDS that detects attackers based on an adaptive threshold. The results have shown that this approach can significantly mitigate ARP spoofing attacks by increasing detection precision and reducing misjudgments in the network. Future work will focus on adapting the proposed method for SDN multi-controller networks.

Acknowledgments

Research was sponsored by the Army Research Office and the Army Research Laboratory Cooperative Agreement, and was accomplished under Grant Number W911NF-21-1-0326 and cooperative agreement W911NF-23-2-0012. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] Weihua Gao, Yuhao Sun, Qingying Fu, Zhouzhe Wu, Xiao Ma, Kai Zheng, and Xin Huang. ARP poisoning prevention in internet of things. In *2018 9th International Conference on Information Technology in Medicine and Education*, pages 733–736, 2018. https://doi.org/10.1109/ITME.2018.00166.

[2] Sabah M. Morsy and Dalia Nashat. D-ARP: an efficient scheme to detect and prevent ARP spoofing. *IEEE Access*, 10:49142–49153, 2022. https://doi.org/10.1109/ACCESS.2022.3172329.

[3] Shah Zawar and Cosgrove Steve. Mitigating ARP cache poisoning attack in software-defined networking (SDN): a survey. *Electronics*, 8(10):1095, 2019. https://doi.org/10.3390/electronics8101095.

[4] Huan Ma, Hao Ding, Yang Yang, Zhenqiang Mi, and Miao Zhang. SDN-based ARP attack detection for cloud centers. In *IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and IEEE 12th Intl Conf on Autonomic and Trusted Computing and IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops*, pages 1049–1054, August 2015. https://doi.org/10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.195.

[5] Chakrala D. and Francis X. C. D. Security against ARP spoofing attacks using bayesian support vector regression. volume 8, May 2019.

[6] Vianney Kengne Tchendji, Fabrice Mvah, Clémentin Tayou Djamégni, and Yannick Florian Yankam. E2BaSeP: Efficient bayes based security protocol against ARP spoofing attacks in SDN architectures. *J. Hardw. Syst. Secur.*, 5(1):58–74, 2021. https://doi.org/10.1007/s41635-020-00105-x.

[7] Fabrice Mvah, Vianney Kengne Tchendji, Clémentin Tayou Djamegni, Ahmed H. Anwar, Deepak K. Tosh, and Charles Kamhoua. GaTeBaSep: game theory-based security protocol against ARP spoofing attacks in software-defined networks. *International Journal of Information Security*, pages 1615–5270, 2023. https://doi.org/10.1007/s10207-023-00749-0.

[8] Mohan Dhawan, Rishabh Poddar, Kshiteej Mahajan, and Vijay Mann. Sphinx: Detecting security attacks in software-defined networks. In *22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA*, pages 8–11, February 2015.

[9] Pradip Kumar Sharma, Saurabh Singh, Y. S. Jeong, and Jong Hyuk Park. DistBlockNet: A distributed blockchains-based secure SDN architecture for iot networks. *IEEE Communications Magazine*, 55:78–85, 2017.

[10] Nehra Ajay, Tripathi Meenakshi, and Gaur Manoj. FICUR: Employing SDN programmability to secure ARP. In *IEEE 7th Annual Computing and Communication Workshop and Conference*, pages 1–8, 01 2017. https://doi.org/10.1109/CCWC.2017.7868450.

[11] Sorin Buzura, Mihaiela Lehene, Bogdan Iancu, and Vasile Dadarlat. An extendable software architecture for mitigating ARP spoofing-based attacks in SDN data plane layer. *Electronics*, 11(13):1965, 2022. https://doi.org/10.3390/electronics11131965.

[12] Jamil Harun, Ali Abid, and Faisal Jami. Spoofing attack mitigation in address resolution protocol (ARP) and ddos in software-defined networking. *Journal of Information Security and Cybercrimes Research*, 5(1):31–42, June 2022. https://doi.org/10.26735/VBVS3993.

[13] Saritakumar N, Anusuya K V, and Sreehari Krishnakumar. Detection of ARP spoofing attacks in software defined networks. In *2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS)*, pages 422–426, 2023.

[14] Noe Marcelo Yungaicela-Naula, César Vargas Rosales, Jesús Arturo Pérez Díaz, and Mahdi Zareei. Towards security automation in software defined networks. *Comput. Commun.*, 183:64–82, 2022. https://doi.org/10.1016/j.comcom.2021.11.014.

[15] Thomas Girdler and Vassilios G. Vassilakis. Implementing an intrusion detection and prevention system using software-defined networking: Defending against ARP spoofing attacks and blacklisted MAC addresses. *Comput. Electr. Eng.*, 90:106990, 2021. https://doi.org/10.1016/j.compeleceng.2021.106990.

[16] Junchi Xing, Mingliang Yang, Haifeng Zhou, Chunming Wu, and Wei Ruan. Hiding and trapping: A deceptive approach for defending against network reconnaissance with software-defined network. In *38th IEEE International Performance Computing and Communications Conference, IPCCC 2019, London, United Kingdom, October 29-31, 2019*, pages 1–8. IEEE, 2019. https://doi.org/10.1109/IPCCC47392.2019.8958776.