In-Band ARP-based Man-in-the-Middle Attack Detection Using P4 Programmable Switches

Christian Garzón, Abdelkader Lahmadi, Jaime Vergara, Alexánder Leal, Juan Felipe Botero

In-Band ARP-based Man-in-the-Middle Attack Detection Using P4 Programmable Switches

Christian Garzón*, Abdelkader Lahmadi†, Jaime Vergara*, Alexánder Leal* Juan Felipe Botero*
*Faculty of Engineering, Universidad de Antioquia, Medellín, Colombia
†Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
christianc.garzon@udea.edu.co, abdelkader.lahmadi@loria.fr, jalberto.vergara@udea.edu.co, erwin.leal@udea.edu.co, juanf.botero@udea.edu.co

***Abstract*—Recently, P4-based Programmable Data Planes (PDP) have attracted significant interest in providing more fine-grained and customized packet processing, particularly for network security functions, including attack detection. This paper introduces a novel approach to detect Man-in-the-Middle (MitM) attacks by performing in-band processing of the Address Resolution Protocol (ARP) using Programming Protocol-independent Packet Processor (P4) programmable switches. To this end, we designed and developed an effective detection system for MitM based on real-time statistical data by measuring, in the data plane, the ARP messages for each connected host to the programmable switch that raise alarms according to configurable burst sizes. We conducted experiments to validate our approach and evaluated its effectiveness in attack detection times (DT) and false positive (FP) rates. The results show that at high rates our detection system achieves mean DT of 1.7 seconds for Network Discovery (ND) attack and 3.13 seconds for Spoofing-Poisoning (SP) attack. Also, at high rates, we obtained reduced FP alarms, 0.69% and 2.59% for ND and SP attacks, respectively. The results are consistent compared with a SDN existing approach.**

***Keywords*—SDN, P4, Attack, MitM, PDP, Detection, In-band.**

I. Introduction

Cyber threats, such as Man-in-the-Middle (MitM) attacks, can compromise one or more elements of the CIA triad (Confidentiality, Integrity, and Availability) [1]. In this malicious scheme, a third party intervenes in a private communication channel between two legitimate parties. Once the MitM attack is initiated, the third party gains control over the channel, enabling the malicious host to engage in activities such as reading, modifying, disrupting, or even replacing the packet's data, which allows for the execution of falsification and replay attacks by leveraging the address resolution protocol (ARP)-spoofing techniques. While traffic encryption methods (e.g., HTTPS, SSH, or VPN) and specialized security functions (e.g., IDS or Firewall) can mitigate such threats, they may impact network performance and are not always feasible for industrial control systems due to concerns like latency, traffic duplication, or data loss [2]–[4].

Notably, Software-Defined Networking's (SDN) principle of decoupling the control and data planes to manage them independently led to the introduction of pioneering controller-based cybersecurity applications that reached promising results regarding MitM attack detection techniques [5, 6]. However, due to the centralized architecture inherent to SDN, the controllers began to manifest certain limitations, like high latency and susceptibility to bottlenecks, which, in turn, posed single-point failure risks [7]. These limitations motivated the exploration of complete or hybrid approaches between the controller and the data plane as a natural next step in SDN development. The PDP concept introduces customizing packet processing behaviors in network devices and facilitating features such as line-rate processing, improved granularity, enhanced visibility, and in-band network telemetry. This customization is achieved by exposing low-level packet processing logic through high-level abstractions. Thus, an approach based on an in-band MitM detection technique through PDP, using the widely accepted P4 language [8], can support a controller-based approach leveraging early attack detection and reducing the need for continuous controller intervention.

This paper introduces an in-band detection mechanism based on real-time statistical data for layer 2 MitM attacks, focusing on ARP and encompassing ARP ND and ARP SP attacks. Our approach can individually detect the attacks that, when combined, could constitute ARP-based MitM attack. P4 external objects accompany the proposed technique, including meters, registers, and match-action tables. Notably, the detection process takes place exclusively within the data plane, without the dependency on extensive historical information stored in the switch's memory. The experiments show that the proposed strategy achieves a consistent DT and reduced FP percentage triggered alarms in presence of high traffic rates compared to other approaches in related work. The experimental platform used is the Bmv2 software switch, which effectively emulates multiple hosts connected to a production switch, along with a virtual network defined by Mininet, a synthetic traffic generator, and an external python-based controller that receives the individualized ND and SP generated alarms.

The remainder of this paper is as follows: Section II focuses on the previous work. Section III describes ARP operation and the MitM attack vector. Next, Section IV presents our detection approach and its implementation. Section V demonstrates the effectiveness of our detection technique through experimental results compared with related work. Finally, Section VI concludes this article and describes future work.

II. Related work

In classical networks, the predominant methodologies employed for detecting ARP spoofing attacks are Port Security and dynamic ARP inspection (DAI). These techniques are regularly integrated into specific hardware solutions provided by network vendors such as Cisco, Juniper, among others. Port security entails the association of MAC addresses with designated ports; otherwise, the packet is discarded [9]. On the other hand, DAI operates by examining ARP packets and comparing them with a valid MAC-to-IP address binding stored in a trusted database (a DHCP snooping binding database) [9]. However, it is essential to note that Port Security and DAI use pre-established MAC-to-Port or MAC-to-IP entries, making them difficult to scale in the data plane due to multiple table entries in each switch or in the control plane due to possible database query congestion.

Convincing Mechanism for MitM Detection (CMD), introduced in [5], leverages OpenFlow and the Floodlight controller, to identify various MitM spoofing attacks, including ARP, DNS, DHCP, ICMP, and SSL/TLS. By analyzing traffic patterns influenced by MitM, CMD observes fixed communication delays (ΔT) caused by channel interception and similar packet/byte ratios between victim and attacker nodes in both directions. Using OpenFlow artifacts, CMD compares time intervals and packet/byte ratios among nodes to detect MitM attacks. Besides, Girdler et al. [6] utilized OpenFlow to create a blacklist of MAC addresses that do not match between Ethernet frames and ARP payloads, detecting ARP spoofing.

Narayanan et al. [10] developed a complete data plane application that employs P4 match-action tables embedded in a BMv2 switch to compare ARP REPLY packets. If a match is found between a particular field in the ARP packet, namely the sender's MAC address, and an entry called "ARP cache table", the packet is forwarded; otherwise, it is dropped. Collaborative control and data plane applications enclose the redirection of packets from programmable switches to controllers for further analysis. The study presented in [11] employs match-action tables to handle ARP REQUEST packets within a specific range of IP addresses. These identified packets are then forwarded to a controller for further processing and the establishment of appropriate ARP REPLY packets to avoid MitM ARP-based attacks. It should be noted that exclusively ARP REPLY packets originating from the controller are selected for subsequent processing, thereby preventing SP attempts. Finally, ARP REPLY packets, potentially compromised by an SP, coming from end devices other than the controller are systematically discarded.

The primary limitation of the mentioned contributions lies in their reliance on external interventions in the data plane. These interventions, such as populating specific ARP match tables based on network traffic, introduce additional processing time. Furthermore, the previously discussed works predominantly focused on approaches dominated by the control plane or a combination of control and data planes. In contrast, our solution presents the following contributions:

- Implementation of a novel approach totally based on PDP capabilities with entirely in-band detection.
- Individualized early detection of ND and SP attacks without controller intervention.

III. ARP-based Attacks

This section will describe the regular operation of ARP and how a malicious third party can exploit it.

In a local area network, end devices contain ARP tables that establish the relationship between IP addresses (Layer 3) and MAC addresses (Layer 2) of other hosts. When an end device only knows the IP address to connect to another, it must initiate an ARP REQUEST as shown in Fig. 1 in which **h1** wants to connect to **h2**. This process is aimed at fulfilling the ARP tables on each end device. Consequently, the ARP REQUEST packet has broadcast destination Layer 2 addresses since the physical address has yet to be discovered. Given the network's collaborative nature and the message's broadcast nature, the requested device typically responds with an ARP REPLY message, as is shown in Fig. 1 when **h2** replies to the requirement of **h1**; on the contrary, **h3** ignores the requirement of **h1**, since it is not the ARP REQUEST original destination. Also, Fig. 1 illustrates the ARP header structure, which will be helpful in the following sections.

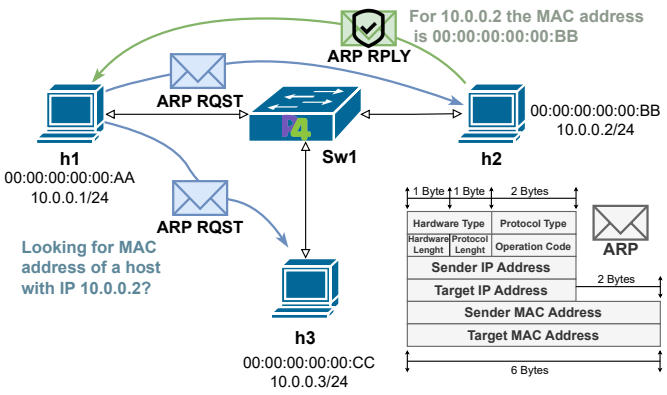

Fig. 1: ARP operation scheme and header fields.

A. ARP ND attack

Attackers often aim to gather information about active hosts or services within a network. By analyzing responses from other hosts, they can map out the network infrastructure [1]. For instance, a malicious actor might attempt to obtain MAC addresses for all hosts within a given network address range. This can be done by sending multiple ARP REQUEST packets across various IP directions within the network mask. Exploiting the operation of the ARP protocol which allows attackers to enumerate all active devices.

B. ARP SP attack

In an ARP spoofing attack, a malicious third party sends spoofed ARP REPLY messages to link its MAC address with

a legitimate IP address. If the attack succeeds, the attacker can eavesdrop or even modify data intended for the legitimate owner [1]. Indeed, the ARP tables have been poisoned because the legitimate information was spoofed, the end host relies on the collected information, and the malicious third party sends spoofed ARP REPLY messages periodically to maintain the attack over time.

C. MitM ARP attack

By exploiting the attacks mentioned above, an attacker can position himself between two or more genuine end devices. Initially, the attacker can seek to explore the network by issuing multiple ARP REQUEST packets in a ND attack. Subsequently, using information from the previous step or a priori network knowledge, the attacker launches an SP attack to impersonate two devices. The ARP REPLY packets sent employ a malicious third-party MAC address instead of the legitimate ones, so the attacker can seek to force legitimate hosts to communicate through a controlled malicious host. Finally, the above actions expose the potential use of ND and SP attacks in combination or autonomously, underscoring the imperative to develop a mechanism capable of independently detecting both attacks.

IV. Detection System Design and Implementation

As detailed in the Section III, the execution of a MitM attack can be leveraged with ND and SP attacks. Therefore, monitoring ARP traffic within a programmable switch enables the detection of MitM attacks on the network. To better understand the detection process of these attacks, Fig. 2 illustrates the steps to individually recognize the two types of ARP attacks of interest: ND and SP. These steps, marked as A, B, and C in the bottom right corner of Fig. 2, are explained below.

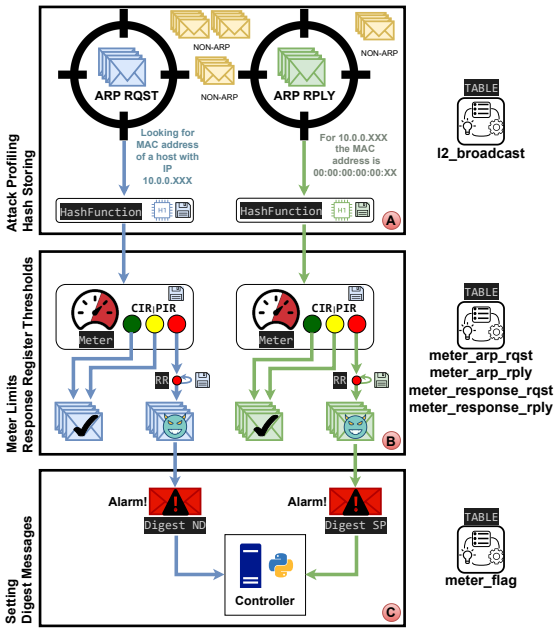

Fig. 2: Flowchart of the detection mechanism.

A. Attack profiling and hash storing

The P4 hash function maps a set of N keys (i.e. protocol headers), within a finite range into M elements or hashes, each represented by B-bits. The hash value's specific size, B, and its processing depend on the chosen hash algorithm and the architecture [8]. To create ARP hashes, we select specific ARP header fields along with the ingress_port as keys for the hash function, targeting ND and SP attacks (see Fig. 3).

In particular, the ND stage of a MitM attack can transmit multiple ARP REQUEST packets with identical field values. Notably, the distinguishing feature among these packets lies in the *header.arp.sender_IP_addr* field, as the attack may not vary this value across all the IP addresses defined by the network mask. Besides, the use of *standard_metadata.ingress_port* is necessary because the detection technique will report the switch ports within the raised alarms, and the main reason is that a MitM attacker can change the IP or MAC address but not the compromised physical port.

The same header fields used in the ND attack (ARP REQUEST) can be used for the SP attack. However, it is feasible to include *header.arp.target_IP_addr* field value because the malicious third party already knows the MAC and IP addresses of the victims; the attacker periodically sends spoofed ARP REPLY messages to keep the attack going.

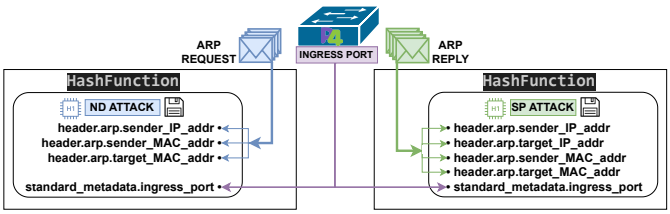

Fig. 3: Selection of hash keys for ND and SP attacks detection.

After detecting an ARP REQUEST or ARP REPLY packet, the specified field values are processed through a hash function. Additionally, there is a need to establish dedicated match-action tables inside the P4 language. Table keys specify data-plane values to look up for some entry (header fields or metadata). In addition, table actions refer to associated responses to matched lookup entries or even a default action for those without matches. Also, actions can request one, more than one, or no parameters. Hence, control plane or network programmers must feed the tables with entries [8].

A match-action table for forwarding is imperative, and it can use the destination MAC address to determine the outgoing port (Layer 2 forwarding). It is essential, however, for the specified table to also distinguish when a broadcast operation becomes necessary. The proposed match-action table **l2_broadcast** executes L2 forwarding and broadcast actions based on destination MAC address and ingress port, determining the assigned multicast group or egress port.

B. Defining meter limits and response register thresholds

Once the hash keys have been characterized, the next step is to define the meter limits and response register thresholds. To do this, statistical analysis must be performed on legitimate and malicious ARP traffic to accurately define these parameters. Indeed, one of the most widely adopted detection techniques in the P4 programming language, even using the BMv2 platform, involves the utilization of thresholds, including exclusive data plane detection or collaborative mechanisms with the control plane [12]–[14].

1) Meter limits - a legitimate traffic analysis: P4 meters classify statistical data (packets or bits per unit of time) coming into the switch, resulting in a subsequent action for the classified data flows. P4 meters are based on the RFC 2698 [15], which suggests using Two Rate Three Color Marker (trTCM) for classification. The two rates parameters are called Commited Information Rate (CIR) and Peak Information Rate (PIR), and the three colors are GREEN, YELLOW, and RED [8, 16]. Depending on data flow rates, the user or the control plane entity sets the CIR and PIR values, and the incoming packets can be marked RED if the rate exceeds the PIR, YELLOW if it exceeds CIR but not the PIR, otherwise the data is marked GREEN. P4 can work with two types of meters: *Direct*, which are associated to a flow and a table, and *Indirect* arrays of meters activated with different flows by one or more tables using an index. This work uses Indirect meters with activation indexes defined by hash values.

To analyze legitimate traffic at its maximum rates in a real experimental environment to define the meter parameters, we used the dataset CIC-IDS2017 [17]. It includes traffic processed by devices such as routers, firewalls, switches, servers, and hosts. Regarding ARP traffic, on a regular operation day (without attacks), it represents 0.4% of the total traffic. Of these ARP packets, 60% are ARP REQUEST packets, and the remaining are ARP REPLY packets. In addition, a statistical analysis was applied over the ARP REQUEST traffic using 3-second time windows (just as ND attack), 44 flows were found, using the keys of the hash function. Of these flows, only 2 exceeded 40 occurrences (packets with the same hash in the same time window) with 50 and 90 occurrences, respectively. On the other hand, for ARP REPLY within a 6-second time window (just as SP attack), 76 flows were found (avoiding outliers), but only 1 obtained a maximum of 3 occurrences along the time window. For v1model architecture [8] and RFC 2698, based on the last analysis and the proposed hashes, without any attack influence, we defined a CIR of 80 and a PIR of 90 for ARP REQUEST packet occurrences over three seconds, and a PIR of 5 and a CIR of 3 for ARP REPLIES over six seconds, the maximum beningn traffic rate in both cases. RED meter responses will be stored in registers to keep track of their number.

2) Response register thresholds - malicious traffic analysis: P4 registers represent stateful memory components within a programmable switch, allowing reading and writing values while processing a flow [8]. These registers store information over extended periods, in contrast to metadata, which is updated with the arrival of each new packet in the ingress. Each register has a fixed size of M elements and consists of B-bits. However, the memory available for storing registers may be limited due to the forwarding nature of programmable switches and the requirement for line rate operations [8]. Our detection technique incorporates a light **response register** (RR) of size two bits to process meter responses for ARP REQUEST and REPLY packets after reaching specified RED thresholds. The corresponding ND or SP attack alarm is triggered when an accumulation of RED marked packets has reached a threshold. Although the detection of attacks was initially confined to reading of RED meters responses, this approach led to numerous FPs. Consequently, it was necessary to complement the meter strategy with the RR strategy.

The attack analysis employed various MitM tools (e.g., ettercap, bettercap, and dsniff) which involve malicious rates for ND and SP attacks. In fact, these malicious rates are the based to complement meter limits (PIR/CIR values) with the use of RR thresholds for marking packets with RED in meter responses. The malicious rate and its RR threshold is defined as follows:

- **ND ATTACK:** 254 malicious packets in 3 seconds, and a RR threshold of 120 red marked packets.
- **SP ATTACK:** 10 malicious packets in 6 seconds, and a RR threshold 3 red marked packets.

The implementation of meter limits and RR thresholds involves creating match-action tables: **meter_arp_rqst** and **meter_arp_rply**. These tables initiate Indirect meters for distinguishing ARP REQUEST and REPLY hashes, respectively, with corresponding responses stored in metadata. The **meter_response_rqst** and **meter_response_rply** tables act as both a reader and writer of RR, capable of adding or resetting counters based on meter responses and recording alarm flags in metadata upon exceeding thresholds. These tables operate by matching ARP operation codes, with the **meter_response** tables also considering the meter color code response.

C. Setting digest messages

A digest message, defined in P4 code, is a data structure to communicate information from the data plane to the control plane, where the P4-based controller must be able to read it. Our approach utilizes two digest structures: one for ND attempt detection, and another one for SP attempt detection which includes the IP addresses potentially compromised under a MitM attack. Both digest messages also include the potential malicious port. Based on the metadata REQUEST and REPLY flags, the match-action table **meter_flag** fulfills the digest messages with the corresponding information to accomplish this last step. Next, the switch sends the alarm to the control plane to notify ND and SP attempts.

V. Experiments and results

A virtual network topology emulated in Mininet with the BMv2 software switch defined by the v1Model architecture [8] was used to implement the proposed detection technique (see

Fig. 4). This setup ran on a Core i7 10700 computer with 8 GB of RAM and Ubuntu 20.04.6 LTS. The potential expansion of hosts connected to Sw1 may scale comparably to production switches which work in the topology of the CIC-IDS2017 dataset [17]. The extent of this expansion depends on the number of defined hashes and the switch's memory capacity to prevent hash collisions. In our experiments, we implemented an ARP traffic generator that allows hosts to create ARP REQUEST and REPLY packets constantly. The added traffic is unitary and random: unitary for source ARP fields as local IP/MAC host addresses and random for source ARP fields as a choice of IP/MAC in a dictionary (between 29 members), which excludes the topology host tuples and emulates ARP traffic from extra hosts. Both types of traffic are constantly and simultaneously forwarded with the same maximum rate during the monitoring time. The added traffic stresses the switch and maintains heavy conditions along the experiments. Even in high traffic situations, the rates we emulate in this experiments for ARP traffic are not typical (see Section IV-B).

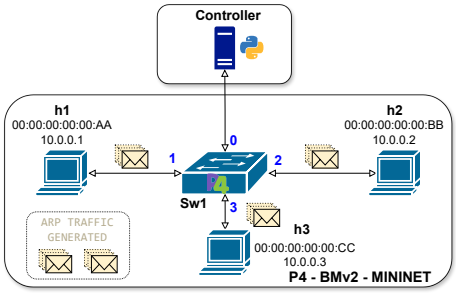

Fig. 4: Implemented experimental platform.

Experiments, with or without MitM influence, evaluate metrics carried out at six different added traffic rates, three rates per ARP message, including the maximum rates found in IV-B. Each rate is evaluated in five runs, each run is monitored during three time windows, and even during rare situations, since only one high rate occurred once a day and has no consecutive windows occurrences [17]. The traffic window for REQUEST packets takes 15 seconds and the traffic window for REPLY packets takes 18 seconds of total monitoring.

In this initial series of experiments, we evaluate the FP alarms without malicious traffic (non-MitM influence). Hence, it is necessary to review two exceptional cases: i) when the RR is not used (without RR), which will increase the FP alarms in heavy conditions since every RED metric response generates an alarm; ii) when the RR is used, which ideally should reduce the FP alarms to 0.0%. Table I shows the obtained results using mean FP (FP_M) and FP percentage (FP_%) according to number of packets per traffic rate in the monitoring window. Indeed, without RR analysis can trigger a lot of FPs, specifically 36.24% for REQUEST traffic and 2.78% for REPLY traffic. On the contrary, RR analysis highlights the absence of FP for mean rates and a reduced percentage in high rates, even without exceeding 1.5% for both cases, an expected result as the system is deliberately stressed by utilizing an anomalous rate across three consecutive time windows. Notably, a RR is more efficient in reducing FPs for REQUEST traffic. However, implementing the RR can reduce the FPs by almost half for REPLY traffic.

TABLE I: Experiment series results with non-MitM influence.

ARP REQUEST						
	90pkts/3s		**45pkts/3s**		**21pkts/3s**	
	FP_M	**FP_%**	**FP_M**	**FP_%**	**FP_M**	**FP_%**
without RR	587.1	36.24%	136.0	16.79%	0.00	0.00%
with RR	9.6	0.59%	2.2	0.27%	0.0	0.00%
Number of Packets	1620		810		378	
ARP REPLY						
	3pkts/6s		**2pkts/6s**		**1pkts/6s**	
	FP_M	**FP_%**	**FP_M**	**FP_%**	**FP_M**	**FP_%**
without RR	1.5	2.78%	0.0	0.00%	0.0	0.00%
with RR	0.6	1.11%	0.0	0.00%	0.0	0.00%
Number of Packets	54		36		18	

For MitM influence, the malicious host is randomly chosen using a uniform probability distribution in each run. The attacks are launched along the time windows in each traffic rate. In this new series of experiments, under MitM's influence, we aim to evaluate the DT measured in seconds, and FP alarms for ND and SP attacks. In addition, each DT includes the confidence interval (CI). Table II shows the obtained results for all runs; The results also hold the same number of packets per traffic rate in the monitoring window (see Table I), there are no FP alarms without added unitary traffic, and the DT are more stable. For instance, using the high rates under MitM influence, the percentages of FP alarms in both versions of the experiments are reduced to 0.69% for ND and 2.59% for SP. In comparison and based on Table I, the FP percentages of ND and SP attacks are significantly reduced with respect to those with no use of the RR (36.24% and 2.78%, respectively) and similar to those with the use of the RR (0.59% and 1.11%, respectively), leveraging the need for light RR use. Regarding CI, the results can determine the relation between traffic rates and DT, that is, the higher the rate, the lower the DT. In addition, the CI of each DT obtained at each rate indicates that the initial scan (ND attempt) detection exhibits shorter DT but broader intervals. In contrast, the intervals associated with detecting ARP poisoning (SP attempt) tend to be narrower, while the DT are longer. However, both the intervals and the DT show good consistency and small variability, indicating the robustness of the proposed approach.

Compared with experiments in [6], we obtained similar DT for ND attacks, even with our high added traffic rates. For SP attacks, our approach required more time to detect

TABLE II: ND and SP attacks experiment series results under MitM influence.

ND ATTACK					
90pkts/3s		**45pkts/3s**		**21pkts/3s**	
DT (s)	**FP (%)**	**DT (s)**	**FP (%)**	**DT (s)**	**FP (%)**
1.70 ± 0.15	0.69%	2.01 ± 0.28	0.52%	2.51 ± 0.36	0.16%
SP ATTACK					
3pkts/6s		**2pkts/6s**		**1pkts/6s**	
DT (s)	**FP (%)**	**DT (s)**	**FP (%)**	**DT (s)**	**FP (%)**
3.13 ± 0.01	2.59%	3.14 ± 0.02	1.11%	3.37 ± 0.38	0,00%

than [6], but we avoided the overhead of creating new rules in collaboration with the SDN controller or MAC addresses mapping. Nevertheless, our DT stays low and acceptable (including the controller link delay) to effectively mitigate a MitM attack, considering that a MitM attack typically requires a timeframe of months or years to gather information and execute actions [18]. Concerning FP alarms, the combined use of meters and RR, even at high rates, gives us similar results to Girdler et al. [6].

VI. Conclusion and future work

This paper presents a real-time MitM detection solution implemented exclusively in the data plane using P4 programming. Our design prioritizes real-time data processing, minimizes memory usage, and conducts in-network processing of ARP packets for enhanced MitM detection.

Experimental evaluations using Mininet and the BMv2 software switch demonstrate the effectiveness of our approach. Our solution consistently achieves reduced attack detection delays and very low FP alarms, even under high and sporadic traffic rates, enabling prompt notification to security operations center for further mitigation.

As part of our future work, we intend to implement this approach on physical switches and utilize higher volumes of network traffic, especially no-ARP traffic. In addition, we plan to improve the coupling between the control and data planes, dynamically changing the meter parameters and response thresholds to create an auto-configurable approach to improving the detection process. The network monitoring and telemetry capabilities facilitated by PDP can improve the detection, thus contributing to the implementation of more accurate mitigation strategies. However, new evaluation methods must be adjusted to create more realistic and reliable security applications. Additionally, we can implement machine learning techniques within a collaborative plane approach and calculate extra evaluation metrics such as true negatives, true positives, and false negatives.

Acknowledgment

The authors thank the ORION program for its contribution to funding Christian Garzón's research internship. This work has benefited from a government grant managed by the Agence Nationale de la Recherche with the reference ANR-20-SFRI-0009. This work has been partially supported by the French National Research Agency under the France 2030 label (Superviz ANR-22-PECY-0008). The development of this article has been supported by the General System of Royalties from Colombia with the reference BPIN code 2020000100381.

References

[1] National Institute of Standards and Technology, "Nist," 2024. Accessed: 2023-10-03.

[2] M. Panda, "Performance analysis of encryption algorithms for security," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, pp. 278–284, 2016.

[3] M. Cheminod, L. Durante, L. Seno, and A. Valenzano, "Performance evaluation and modeling of an industrial application-layer firewall," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2159–2170, 2018.

[4] D. Faquir, N. Chouliaras, V. Sofia, K. Olga, and L. Maglaras, "Cybersecurity in smart grids, challenges and solutions," *AIMS Electronics and Electrical Engineering*, 2021.

[5] K. Zhang and X. Qiu, "Cmd: A convincing mechanism for mitm detection in sdn," in *2018 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–6, 2018.

[6] T. Girdler and V. G. Vassilakis, "Implementing an intrusion detection and prevention system using software-defined networking: Defending against arp spoofing attacks and blacklisted mac addresses," *Computers Electrical Engineering*, vol. 90, p. 106990, 2021.

[7] A. Abuarqoub, "A review of the control plane scalability approaches in software defined networking," *Future Internet*, vol. 12, no. 3, 2020.

[8] P4 Language Consortium, "Behavioral model version 2 (BMv2)." https://github.com/p4lang/behavioral-model, 2015.

[9] C. Systems", "Security configuration guide, cisco ios xe release 3se (catalyst 3650 switches)." https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3650/software/release/3se/security/configuration_guide/b_sec_3se_3650_cg.html.

[10] N. Narayanan, G. C. Sankaran, and K. M. Sivalingam, "Mitigation of security attacks in the sdn data plane using p4-enabled switches," in *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1–6, 2019.

[11] T.-Y. Lin, J.-P. Wu, P.-H. Hung, C.-H. Shao, Y.-T. Wang, Y.-Z. Cai, and M.-H. Tsai, "Mitigating syn flooding attack and arp spoofing in sdn data plane," in *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 114–119, 2020.

[12] K. Friday, E. Kfoury, E. Bou-Harb, and J. Crichigno, "Towards a unified in-network ddos detection and mitigation strategy," in *2020 6th IEEE Conference on Network Softwarization (NetSoft)*, pp. 218–226, 2020.

[13] C. Hardegen, S. Rieger, and T. Geier, "Multi-step attack detection and mitigation enhancing in-network flow classification," in *2022 5th International Conference on Advanced Communication Technologies and Networking (CommNet)*, pp. 1–10, IEEE, 2022.

[14] V. Clemens, L.-C. Schulz, M. Gartner, and D. Hausheer, "Ddos detection in p4 using hyperloglog and countmin sketches," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–6, IEEE, 2023.

[15] D. J. Heinanen and D. R. Guerin, "A Two Rate Three Color Marker." RFC 2698, Sept. 1999.

[16] Y.-W. Chen, L.-H. Yen, W.-C. Wang, C.-A. Chuang, Y.-S. Liu, and C.-C. Tseng, "P4-enabled bandwidth management," in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1–5, IEEE, 2019.

[17] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, *et al.*, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.," *ICISSp*, vol. 1, pp. 108–116, 2018.

[18] S. Roy, N. Sharmin, J. C. Acosta, C. Kiekintveld, and A. Laszka, "Survey and taxonomy of adversarial reconnaissance techniques," *ACM Comput. Surv.*, vol. 55, dec 2022.