

A Proposal for IP Spoofing Mitigation at Origin in Homenet Using Software-Defined Networking

Manoel F. Ramos[1], Bruno L. Dalmazo[2(✉)], and Jeferson C. Nobre[2]

[1] University of Vale do Rio dos Sinos, São Leopoldo, Brazil
manoel@dropreal.com
[2] Federal University of Rio Grande do Sul, Porto Alegre, Brazil
{bldalmazo,jcnobre}@inf.ufrgs.br

Abstract. Computer networks are continually evolving, making the execution of security tasks increasingly complex. Also, the development of new networking environments, such as the Internet Engineering Task Force (IETF) Home Networking (Homenet), usually is not followed by advances in security mechanisms for these environments. In this context, IP Spoofing, the obfuscation of the actual network address of the attacker either to amplify or redirect communications responses to a given target, is an example of network attack that can be employed in several infrastructures. Considering Homenet, the utilization of IPv6 does not avoid such attacks since the Neighbor Discovery Protocol (NDP), which is responsible for neighborhood discovery in IPv6, does not have mechanisms of validation of network and link addresses in its packet header (i.e., source-address validation). The present work proposes a solution to mitigate the use of IP Spoofing attacks originated in a Homenet using Software-Defined Networking (SDN) features. The results from the experimental evaluation demonstrate that the proposed method has desirable properties to avoid such attacks without increasing the complexity of the Homenet architecture.

Keywords: IP Spoofing · Homenet · SDN

1 Introduction

Computer networks are constantly evolving, making their configurations increasingly complex. This provides a significant increase on the security challenges related to supported services [1]. In home networking (i.e., domestic networking), this complexity becomes an even more important challenge, since there are no network administrators to implement security policies and to ensure an adequate operational state of the networking infrastructure [7].

The Internet Engineering Task Force (IETF) *Home Networking* (Homenet) Working Group (WG) proposed a solution to allow self-management in home networks, including self-configuration, which can avoid human intervention and,

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11619, pp. 179–192, 2019.
https://doi.org/10.1007/978-3-030-24289-3_15

consequently, the need of technical knowledge from the home user [3]. In this context, it is also necessary to address security features. Homenet has vulnerabilities analogous to those found on traditional networks. Considering self-* properties, self-protection will be need to perform security tasks in an autonomic manner. However, this is not present in current Homenet specification.

Homenet is known to be vulnerable to several network attacks. One of this attacks is IP address spoofing [8]. Such attacks falsify source addresses contained in the headers of network packets, either to amplify/redirect communication responses to a particular target or to ensure the anonymity of the attacker. Also, IP spoofing is usually an initial step for various types of attacks. Fortunately, as the deployed IP addresses in a Homenet are controlled, network programmability can be used to validate source IP addresses.

Software-Defined Networking (SDN) has become a significant network programmability alternative to enhance the security of computer networks by providing the separation of data and control plane, and centralizing decision making over network traffic through the network controller [4]. In this context, SDN can help facing IP Spoofing at origin in home networks (such as IETF Homenet), avoiding that external targets are compromised by attacks originated in the home hosts.

The present paper presents a method for IP Spoofing mitigation in origin at a Homenet using SDN. The main contribution is the support of a solution based on the Homenet standards for the validation of source addresses. This solution employs to use control data in a programmatic fashion. We developed a proof of concept implementation to investigate the properties of the proposed solution.

This paper is organized as follows. In Sect. 2, the background is presented. In Sect. 3, it is described the proposed solution. Section 4 details the evaluation and the experiments performed, followed by the related work described in Sect. 5. Finally, Sect. 6 presents the conclusion about this study and future work.

2 Background

This section presents the central theoretical concepts needed to develop this work. We describe these concepts in two subsections. First, the IETF Homenet and its composing protocols are introduced. Then, this section closes with the presentation of fundamental concepts on SDN.

2.1 Homenet

Homenet is an IETF WG that aims to focus on the evolution of home networks by developing and providing a simple, self-configuring architecture to address IPv6 prefix configuration requirements for routers, network management, name resolution (Domain Name System - DNS) and network discovery services [3]. This architecture can be employed in several contexts, such as Internet of Things (IoT) [5]. Homenet employs two main protocols for its operation: the DNCP (Distributed Node Consensus Protocol) and the Home Networking Control Protocol (HNCP).

Homenet uses the DNCP to receive information from routers through discovery services, negotiations, and autonomous bootstrapping services. In addition, Homenet uses the HNCP with a DNCP profile for sharing information about the state of routers in Homenet. This use allows automatic discovery of gateways and receiving/delegating the prefixes for hosts and routers that have (or not) support from HNCP [3].

For IPv6 prefix assignment, Homenet uses the Distributed Prefix Assignment Algorithm that uses a flooding mechanism, which allows each node to advertise its directly connected prefixes. Also, the algorithm randomly creates an IPv6 /64 prefix and assigns the addresses to the hosts. Devices connected to Homenet will receive the IP addresses through the HNCP protocol. For non-HNCP hosts, the task is made through DHCPv6-PD or SLAAC [3].

2.2 Software-Defined Networking (SDN)

Some traits of SDN can be highlighted in the context of the present work. We will describe two of such traits. The first is the separation of the control plane from the data plane. The control plan decides how to handle network traffic and the data plane forwards traffic as the control plan decides. The second feature is that SDN consolidates the control plane; thus the programmable interface supports direct control over the state of the elements contained in the data plane, such as switches and routers.

SDN integrates the physical and virtual elements, allowing the controller to manage the network in an automated and centralized way. All traffic can be analyzed by the controller, which can decide what action to take on the passing flows. When the switches receive specific network packets, they will analyze their flow tables and, if the origin and destination are not known, the switches will forward the data to the controller [4].

3 Proposal for Mitigation IP Spoofing in Homenet at Origin Using SDN

The proposed solution is the use of an SDN application which uses the controller API to allow a global view of all network traffic through flow analysis. This application, called *SPOOFING_SRC_CONTROL*, is responsible for analyzing and handling the IPv6 addresses of each packet that comes out of Homenet. This verification is done by consulting the addresses provided by the HNCP protocol to the Homenet internal hosts. This section describes the proposed solution. First, the rationale of the solution is presented. Then, the algorithms of the proposed solution are depicted.

The proposed solution runs as an application on the Homenet router, and it can control the entire home network topology. SDN controller may be located on the edge router itself, where it analyses all network traffic and decides if packets can be added to the switch's flow table. In this context, decision making is done by comparing the MAC and IPv6 addresses entered in the switch flow table. If

the IPv6 addresses of a respective packet are not inserted in this table or the IPv6 address is already assigned to another MAC address, the switch forwards the packet to the SDN controller, which then decides what action to take.

The proposed solution can be abstracted in some steps as follows. First, this starts when a switch is unaware of the source and destination IPv6 address of an incoming packet to be forwarded to a particular destination. This packet is sent to the controller (step 1), which will then verify that the destination and source addresses are different from internal Homenet addresses. Then, the controller will query the database containing the IPv6 and MAC addresses of the Homenet hosts generated by the HNCP protocol. After this, the controller processes the address and sends the necessary action on the respective packet for the implied switch. The switch updates its flow table with the action defined by the controller.

Algorithm 1. MOD_CONTROL()

```
Input: IP_SRC, MAC_SRC, IP_DST, LOCAL_NETWORK_ADDRESS.
Output: SDN_STATUS.
if IP_DST == LOCAL_NETWORK_ADDRESS then
    if IP_SRC != LOCAL_NETWORK_ADDRESS then
        SDN_STATUS = ALLOWED
    end
    else
        SDN_STATUS = VERIFICATION
        ANL_IP_SRC = IP_SRC
        ANL_MAC_SRC = MAC_SRC
        MOD_VERIFY_IP-MAC()
        if SPOOFING_STATUS == NO then
            SDN_STATUS = ALLOWED
        end
        if SPOOFING_STATUS == YES then
            SDN_STATUS = BLOCKED
        end
        if SPOOFING_STATUS == FAIL then
            SDN_STATUS = BLOCKED
        end
    end
end
```

Three algorithms compose the proposed solution. In this context, SPOOFING_SRC_CONTROL consists of MOD_CONTROL(), MOD_VERIFY_IP-MAC(), and MOD_IPMAC_SEARCH() submodules. The MOD_CONTROL() module is responsible for deciding on the correct action to be taken on the previously analyzed network packet. It needs the MOD_VERIFY_IP-MAC() and MOD_IPMAC_SEARCH() submodules for its operation. The Algorithm 1 describes storing the packet source IP address information (in the IP_SRC

variable), the source MAC address (in the MAC_SRC variable), and the destination IP address of the packet (in the IP_DST variable). In addition, MOD_CONTROL() has the variable LOCAL_NETWORK_ADDRESS, which has the value corresponding to the IPv6 address of the Homenet local network.

LOCAL_NETWORK_ADDRESS is extracted from the database of the addresses assigned by HNCP to Homenet. The first action of MOD_CONTROL() is to analyze whether the value contained in the IP_DST variable corresponds to the same prefix of the value of the LOCAL_NETWORK_ADDRESS variable and if the source address (IP_SRC) is different from the local network. This process is done to verify if the packet destination is the local network or an external network (Internet), in addition to verifying if it is Homenet internal traffic. If the packet is destined for some internal Homenet host, it is released by the system through the SDN_STATUS function and no further verification is performed by closing the module.

The SDN_STATUS function is responsible for executing the system actions internally and, before the SDN controller, its values are: VERIFICATION that has the internal action of the proposed system to initiate the verification process of the source IPv6 address analyzed; ALLOWED that has the action to be taken by the SDN controller in which to release the flow in the switch; and BLOCKED, which has an action to be taken by the SDN controller in which it will block the flow in the switch.

Algorithm 2. MOD_VERIFY_IP-MAC()

```
Input: ANL_IP_SRC, ANL_MAC_SRC.
Output: SPOOFING_STATUS.
MOD_IP-MAC_SEARCH()
if DB_SERVER_STATUS != OK then
    SPOOFING_STATUS = FAIL
end
else
    if SEARCH_MAC == XX_XX_XX_XX_XX_XX then
        SPOOFING_STATUS = YES
    end
    else
        VERIFY_IP = SEARCH_IP
        if VERIFY_IP == ANL_IP_SRC then
            SPOOFING_STATUS = NO
        end
        else
            SPOOFING_STATUS = YES
        end
    end
end
```

The first value to be generated by the SDN_STATUS function is VERIFICATION, in which the system initiates the verification process by storing

the IP_SRC values in the ANL_IP_SRC variable and the MAC_SRC variable in the ANL_MAC_SRC variable. After this, the MOD_VERIFY_IP-MAC() submodule is executed, feeding the value of the SPOOFING_STATUS variable into MOD_CONTROL(). SPOOFING_STATUS has three types of values, **FAIL** that is generated when the MOD_VERIFY_IP-MAC() submodule did not obtain complete results that prove the existence of IP Spoofing, **NO** that is generated when the MOD_VERIFY_IP-MAC() submodule validated MAC and IPv6 addresses and did not identify the use of the IP Spoofing technique in the packet and **YES** that is generated when the MOD_VERIFY_IP-MAC() submodule validated the MAC and IPv6 addresses and identified the use of the IP Spoofing technique in the packet.

If the value of SPOOFING_STATUS is equal to FAIL, the proposed system could not analyze the addresses and prove the existence of the IP Spoofing technique. This incident may occur in cases of unavailability in the consultation process or in an attack on the database of the addresses assigned to Homenet. In this case, the system can enter two different types of values in the variable SDN_STATUS according to the security policy established by the administrator of the Homenet environment at the moment of configuring the solution. These values are: ALLOWED to release the packet stream; and BLOCKED to block packet flow. If the value of SPOOFING_STATUS is equal to NO, the variable SDN_STATUS receives the ALLOWED value, which indicates to the system that the packet must be released through the SDN controller because it was able to verify the addresses and did not identify the existence of the IP technique Spoofing. Finally, when the SPOOFING_STATUS value is equal to YES, the SDN_STATUS variable receives the value BLOCKED, which tells the system that the SDN controller should discard the packet because it has identified the use of the IP Spoofing technique.

The submodule MOD_VERIFY_IP-MAC(), described in Algorithm 2, works in conjunction with the MOD_IP-MAC_SEARCH() submodule and generates results for the completion of the MOD_CONTROL() module. MOD_VERIFY_IP-MAC() receives the values of ANL_IP_SRC and ANL_MAC_SRC generated by the MOD_CONTROL(). Your first action is to start the MOD module IP-MAC_SEARCH(). After this, the value of DB_SERVER_STATUS generated by the submodule MOD_IP-MAC_SEARCH() is analyzed.

DB_SERVER_STATUS may contain two distinct values, being they FAIL when the MOD_IP-MAC_SEARCH() submodule was not able to execute the address localization process and OK when the MOD_IP-MAC_SEARCH() submodule successfully executed the address localization process. If the value of DB_SERVER_STATUS is other than OK, SPOOFING_STATUS receives the FAIL value. If the value of DB_SERVER_STATUS is equal to OK, the verification process starts.

It is first checked whether the SEARCH_MAC value received by executing the MOD_IP-MAC_SEARCH() submodule is equal to XX_XX_XX_XX_XX_XX. If so, this indicates that the MAC address contained in ANL_MAC_SRC is not in

Algorithm 3. MOD_IP-MAC_SEARCH()

```
Input: DB_SERVER, ANL_IP_SRC, ANL_MAC_SRC.
Output: SEARCH_MAC, SEARCH_IP, DB_SERVER_STATUS.
Connection with DB_SERVER
if DB_SERVER is not accessible then
    DB_SERVER_STATUS = FAIL
end
else
    DB_SERVER_STATUS = OK
    Search ANL_MAC_SRC em DB_SERVER
    if ANL_MAC_SRC is not found then
        SEARCH_MAC = XX_XX_XX_XX_XX_XX
    end
    else
        SEARCH_MAC = ANL_MAC_SRC
        SEARCH_IP = IPv6 assigned to the MAC located in DB_SERVER;
    end
end
```

the database of the list of assigned IP addresses. Thus, the SPOOFING_STATUS value is defined as YES. If not, VERIFY_IP receives the SEARCH_IP value extracted from the module IP-MAC_SEARCH(). If the value of VERIFY_IP is equal to the value of ANL_IP_SRC, the source addresses are in compliance and SPOOFING_STATUS is given the value NO. Finally, if the value of VERIFY_IP is different from ANL_IP_SRC, the use of the IP Spoofing technique is identified and SPOOFING_STATUS receives the value YES, ending the execution of the submodule, giving sequence in the analysis process.

The submodule MOD_IP-MAC_SEARCH(), described in Algorithm 3, has the responsibility of locating the values of ANL_IP_SRC and ANL_MAC_SRC in the database of the list of IPv6 addresses assigned to Homenet. It is started through submodule MOD_VERIFY_IP-MAC().

4 Evaluation

In this section, the experiments carried out to evaluate the solution proposed in a Homenet are presented. First, the topology employed on the simulation (Experimental Scenario) is explained. Then, the implementation of the proposed solution is depicted. Finally, the performed experiments and their results are presented.

4.1 Experimental Scenario

The test infrastructure and its topology were implemented using a virtual environment in which it was possible to evaluate the proposed solution. Figure 1

shows the complete implementation topology. Routers A, B, and C (ISP) communicate through the OSPFv3 protocol and the automatic distribution of IPv6 addressing was used by the DHCPv6-PD service. Hosts B and C are ISP customers as well as Client A, the gateway of Homenet. Each of the hosts has a network interface connected to their respective gateways.

4.2 Development and Implementation of the Proposed Solution

SPOOFING_SRC_CONTROL was developed and implemented on the RYU controller. RYU was developed in the Python programming language, so SPOOFING_SRC_CONTROL was also developed in Python. The RYU driver has been compiled to support the proposed solution. The system is started by the controller when the switch does not know the source and destination (IPv6 or MAC) of the incoming packet because in its flow table there is no such information. With this, the switch sends the packet to the RYU controller which executes SPOOFING_CONTROL through its "*_packet_in_handler*" function before executing the add_flow function.

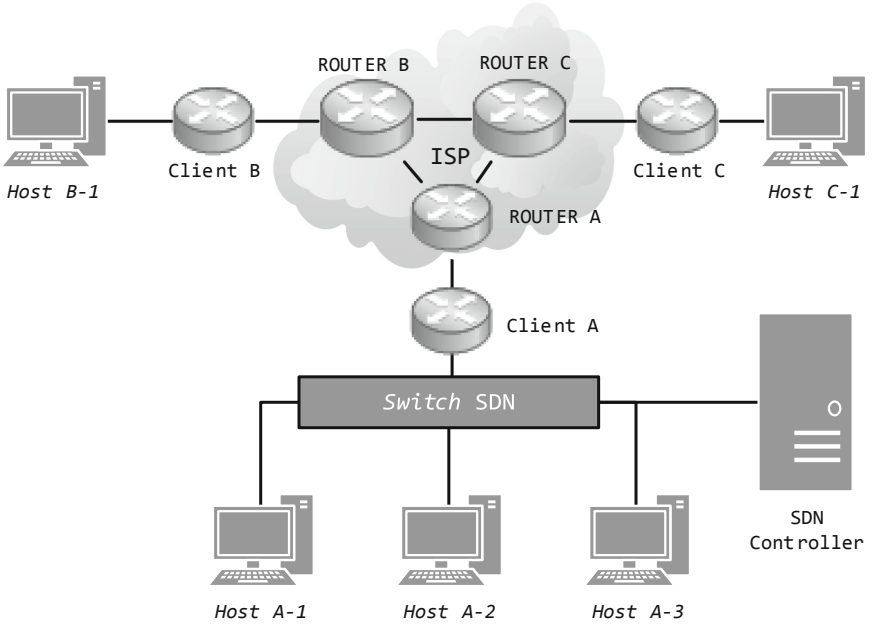

Fig. 1. Simulation infrastructure.

SPOOFING_SRC_CONTROL checks the Homenet source address according to the submodule MOD_VERIFY_IP-MAC(). SPOOFING_SRC_CONTROL has a function, developed according to the submodule MOD_IP-MAC_SEARCH(),

which is responsible for querying the prefixes and IPv6 addresses assigned to Homenet via HNCP. such queries check the source address in each packet. If the source IPv6 address is different from the Homenet or the source MAC address is different from that contained in the assigned address base and the destination address is also different from the Homenet, the use of the IP Spoofing technique is identified and the packet is discarded by the controller.

4.3 Experiments

As shown in Fig. 1, the communication for this experiment was made between Homenet hosts (Hosts A-1, A-2 and A-3) and the other external hosts, ISP clients (Host B-1 and Host C-1). The HNCP protocol assigned two IPv6 prefixes to the Homenet interface on its router and the internal Hosts A-1, A-2, and A-3). For these experiments, the Homenet environment with multihoming support was not implemented. For the use of the IP Spoofing and MAC Spoofing technique in IPv6 networks, the NMAP[1] software was used. The experiments were carried out 10 (ten) times to validate their results, and the observed variance was low.

Three experiments using the IP Spoofing technique were performed in the environment shown in Fig. 1. The experiment 1 validates the use of IP Spoofing at the source through the communication of Homenet hosts with external hosts (Host B-1 and Host C-1), comparing IPv6 addresses and MAC addresses; experiment 2 validates the source communication between Homenet's internal hosts (Hosts A-1, A-2 and A3), also comparing IPv6 addresses and MAC addresses; and experiment 3 validates the communication between the internal hosts in Homenet with the external communication, however, using only the validation of the Homenet source addresses through the assigned IPv6 prefix, not validating the MAC addresses of the Homenet hosts.

Results. The Fig. 2 shows the communication results between Hosts A-1, A-2, and A-3 with Host B-1 (Fig. 1) in a Homenet (experiment 1) without enabling the In the time period of 60 s, the NMAP was run, with IP Spoofing and MAC Spoofing enabled, and Ping6 on all Homenet hosts, successively, to Host B-1. found that Homenet is vulnerable to using the techniques of IP Spoofing and MAC Spoofing. The consumption of network traffic was measured by capturing the results from the output interface of the Homenet eth1 router through the use of the IFTOP software. NMAP totaled the average consumption of 5055 bytes/s (average of 1685 bytes/s per Homenet host) and Ping6 totaled the average consumption of 2412 bytes/s (average of 804 bytes/s per Homenet host) of the traffic generated by Homenet's internal hosts.

The Fig. 3 shows the result of the same communication of hosts A-1, A-2 and A-3 with host B-1 in a Homenet (experiment 1), however, with SPOOFING_SRC_CONTROL enabled on the SDN controller. In the first 20 s of execution, SPOOFING_SRC_CONTROL was executed to validate the host A-1

[1] NMAP - https://nmap.org/.

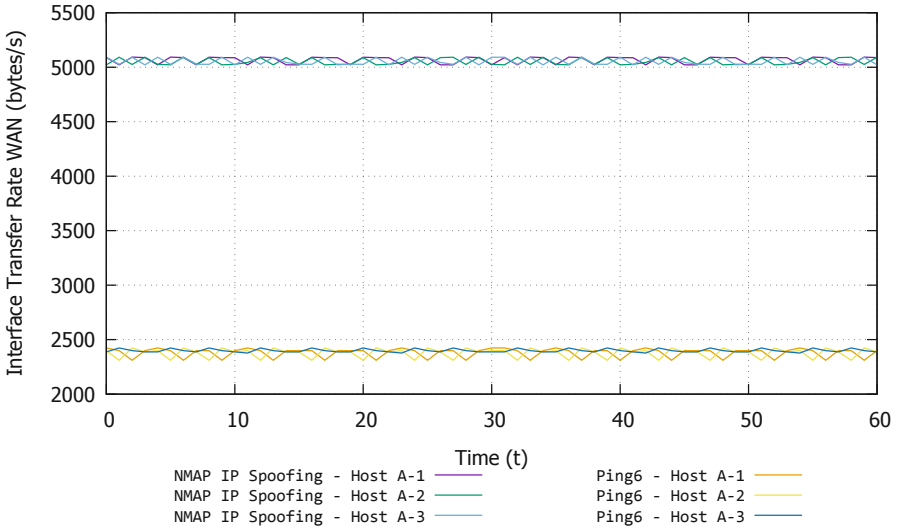

Fig. 2. Applicability of IP Spoofing and MAC Spoofing WITHOUT the proposed solution.

address, where communication of the same was terminated at IP Spoofing detection. In the next 10 and 20 s (30 and 40), the same process was executed on host A-2 and host A-3, terminating their communication after the detection of IP Spoofing. This same test procedure was performed for the validation of source MAC addresses, in which it obtained the same result and proving that the solution is also efficient to counter the use of the MAC Spoofing technique at the origin.

In experiment 2, the proposed solution was efficient in mitigating and blocking the use of the IP Spoofing and MAC Spoofing techniques at the origin. The results were captured at the interface of Mininet via the IFTOP software.

In experiment 3, only IP Spoofing was detected and blocked in communication with Homenet's external hosts, because in this experiment, it is not possible to compare the MAC addresses because the HNCP query base is related only to the IPv6 prefix assigned to the Homenet. This also impacted on the process of verifying the IP Spoofing internally generated in Homenet between its hosts.

Performance Analyzes. Performance analyzes were performed to identify the processing power and memory of the solution in the SDN controller. For this, HTOP software was used, installed directly in RYU controller.

The Fig. 4 shows the result of the processing and memory consumption of the RYU controller without the use of the proposed solution. Within the 60-s period, when the controller did not manipulate the switch's flow table, it had a variation between 0.3% and 0.7% in processing consumption and 1.3% in memory consumption. In seconds 11, 12, 13, 33, 34 and 35 the switch, which was unknown

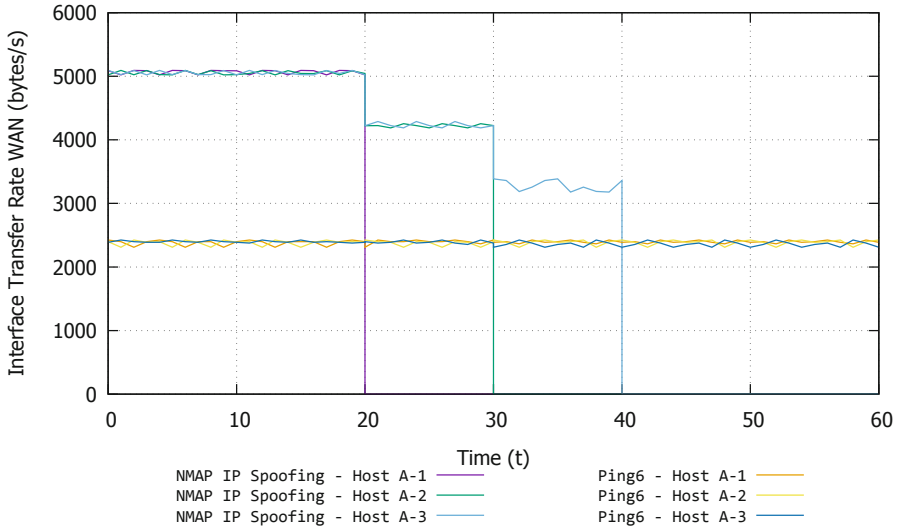

Fig. 3. Applicability of IP Spoofing and MAC Spoofing WITH the proposed solution.

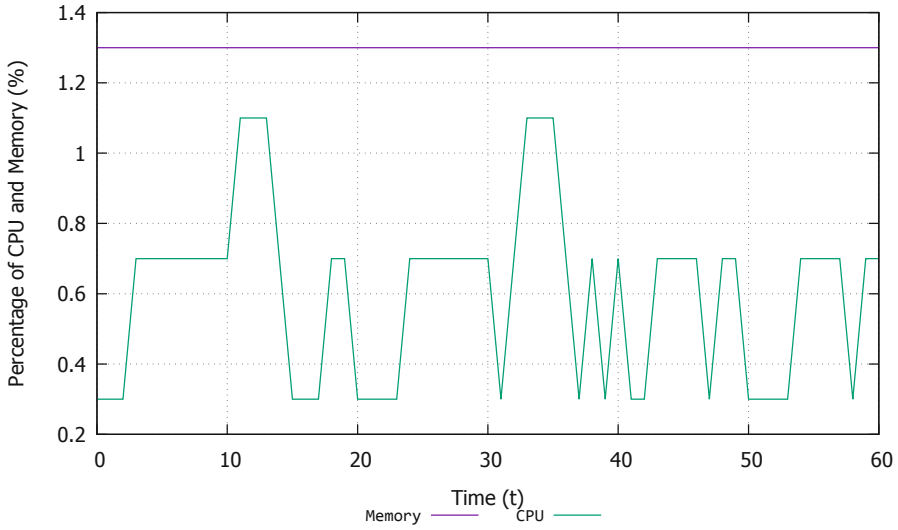

Fig. 4. Computational consumption WITHOUT use of proposed solution.

to the source and destination information of two network packets, forwarded them to the check by the RYU controller, in which it validated the network addresses of the packets and has added the information in the switch flow table. These checks generated a peak of 1.1% in each processing consumption. The RAM consumption was not changed and the same 1.3% consumption did not

oscillate throughout the analysis. Analyzing the controller log records, it was found that it updated the switch flow table in the time period of 0.1 ms.

Figure 5 displays the result of the controller's processing and memory consumption with the proposed solution feature enabled. Within the 60 s period, when the controller did not manipulate the switch flow table, it had a variation between 0.3% and 0.7% in processing consumption and 1.3% in memory consumption. In the second 17, the switch sent the addressing information to the controller which did not use the IP Spoofing technique and/or MAC Spoofing for the controller in which, during the process of validation of addresses, obtained a peak of 1.1% in the consumption of processing (until the second 17) and no variation in the consumption of memory occurred. In seconds 35, 36 and 37, the switch sent the addressing information of a packet containing the Spoofing technique and in seconds 47, 48 and 49, the addressing information of another packet containing the use of the MAC Spoofing technique to the controller. This resulted in seconds 35, 36, 37, 47, 48 and 49, at a consumption of 1.2% of processing and no change in memory consumption. By analyzing the logs of the controller, it was also found that it upgraded the switch flow table in the period of 0.1 ms.

The results proved that the proposed solution consumed the same percentage of the experiment shown in Fig. 4, when the package did not apply the IP Spoofing technique and/or MAC Spoofing and 0.1% more when it was identified. The time for the controller to update the switch's flow table was also not impacted.

5 Related Work

There are several studies related to the mitigation of IP Spoofing in IPv4 networks. On the other hand, few security mechanisms are proposed for prevention in IPv6 networks, in Homenet and mainly deal with the problem directly in its origin. In this section, we present the work related to the purpose of providing a basis for mitigating the use of IP Spoofing techniques.

Barbhuiya et al. [2] present an active IDS for the prevention of IP Spoofing in the NDP protocol exchange process. The authors state that the solution is effective for validating MAC addresses in IPv6 networks. The algorithms developed stand out for their simplicity in the process of identifying and comparing MAC and IPv6 addresses.

Yao et al. [10] contribute to the mechanism VAVE (Virtual Source Address Validation Edge), developed through the SAVI (Source Address Validation Improvement) framework, which uses the Openflow protocol to mitigate IP Spoofing in incoming traffic on a local network. One of the negative factors of this solution is that the SAVI protocol requires adaptations to the current protocols of the Internet, a negative factor to make it a standard protocol before the IETF.

Yan et al. [9] have developed an experiment to implement the SAVI protocol on a local network. In this contribution, the positive factor is that the authors consult DHCPv6 servers to carry out the validation process through NDP messages issued by SAVI.

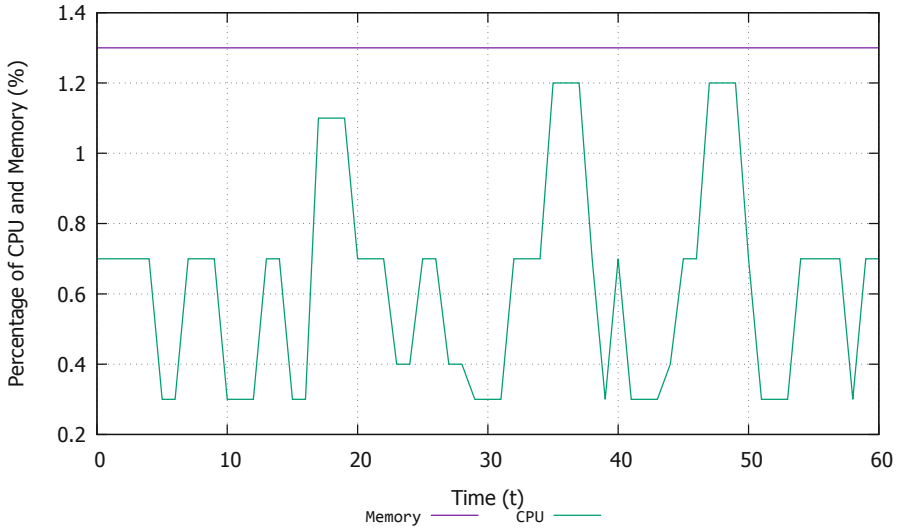

Fig. 5. Computational consumption WITH use of proposed solution.

Finally, Mowla et al. [6] propose a defense mechanism to IP Spoofing in the traffic of received data, validating the legitimate traffic and blocking the Spoofing. The solution is composed of SDN based on Content Distribution Network Interconnection (CDNi) technology, along with ALTO (Application Layer Traffic Optimization) technology. The purpose is to use SDN to detect IP Spoofing, following a mechanism to feed rules into SDN switches through the controller using the markup maps provided by the ALTO server.

6 Conclusions and Future Work

IPv6 networks are vulnerable regarding IP Spoofing techniques which consequently affects any network that uses IPv6, such as IETF Homenet. In the present paper, an efficient mechanism was proposed for the validation of IPv6 addresses at the origin in a Homenet. The results obtained highlighted the efficiency in the mitigation of IP and MAC Spoofing process when the packets are coming out from a Homenet. Besides that, the results also presented the low consumption of the computational resources consumed by the solution. Such consumption is constrained in the SDN controller.

Despite the encouraging results, there is also room for improvements. As future work, we intend to investigate the process of feeding the base of the network addresses assigned to Homenet should be a priority, extending the evaluation of the proposed solution in a Homenet environment using the multihoming feature in an ISP with MSP support and in conventional IPv6 networks.

References

1. Bannour, F., Souihi, S., Mellouk, A.: Distributed SDN control: survey, taxonomy, and challenges. IEEE Commun. Surv. Tutor. **20**(1), 333–354 (2018). https://doi.org/10.1109/COMST.2017.2782482
2. Barbhuiya, F., Bansal, G., Kumar, N., Biswas, S., Nandi, S.: Detection of neighbor discovery protocol based attacks in IPv6 network. Netw. Sci. **2**(4), 91–113 (2013). https://doi.org/10.1007/s13119.013.0018.2
3. Chown, T., Arkko, J., Brandt, A., Troan, O., Weil, J.: IPv6 home networking architecture principles. RFC 7368 (Informational), October 2014
4. da Costa Cordeiro, W.L., Marques, J.A., Gaspary, L.P.: Data plane programmability beyond OpenFlow: opportunities and challenges for network and service operations and management. J. Netw. Syst. Manag. **25**(4), 784–818 (2017)
5. Mars, D., Gammar, S.M., Lahmadi, A., Saidane, L.A.: Using information centric networking in internet of things: a survey. Wirel. Pers. Commun. **105**, 1–17 (2019)
6. Mowla, N., Doh, I., Chae, K.: An efficient defense mechanism for spoofed IP attack in SDN based CDNi. In: Proceedings of the International Conference on Information Networking (ICOIN), pp. 92–97, January 2015. https://doi.org/10.1109/ICOIN.2015.7057863
7. Sivaraman, V., Gharakheili, H.H., Vishwanath, A., Boreli, R., Mehani, O.: Network-level security and privacy control for smart-home IoT devices. In: 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 163–167, October 2015. https://doi.org/10.1109/WiMOB.2015.7347956
8. Xu, K., Wang, F., Egli, R., Fives, A., Howell, R., Mcintyre, O.: Object-oriented big data security analytics: a case study on home network traffic. In: Cai, Z., Wang, C., Cheng, S., Wang, H., Gao, H. (eds.) WASA 2014. LNCS, vol. 8491, pp. 313–323. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07782-6_29
9. Yan, Z., Deng, G., Wu, J.: SAVI-based IPv6 source address validation implementation of the access network. In: 2011 International Conference on Proceedings of Computer Science and Service System (CSSS), pp. 2530–2533, June 2011. https://doi.org/10.1109/CSSS.2011.5974125
10. Yao, G., Bi, J., Xiao, P.: Source address validation solution with OpenFlow/NOX architecture. In: 2011 19th IEEE International Conference on Proceedings of Network Protocols (ICNP), pp. 7–12, October 2011. https://doi.org/10.1109/ICNP.2011.6089085