Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2023) 31: 1187 – 1205

doi:10.55730/1300-0632.4042

LSAV: Lightweight source address validation in SDN to counteract IP spoofing-based DDoS attacks

Ali KARAKOÇ*, Fatih ALAGÖZ
Department of Computer Engineering, Faculty of Engineering, Boğaziçi University, İstanbul, Turkiye

Received: 21.02.2023 • **Accepted/Published Online:** 11.08.2023 • **Final Version:** 30.11.2023

Abstract: In this paper, we propose a design to detect and prevent IP spoofing-based distributed denial of service (DDoS) attacks on software-defined networks (SDNs). DDoS attacks are still one of the significant problems for internet service providers (ISPs) and individual users. These attacks can disrupt customer services by targeting the availability of the system, and in some cases, they can completely shut down the target infrastructure. Protecting the system against DDoS attacks is therefore crucial for ensuring the reliability and availability of internet services. To address this problem, we propose a lightweight source address validation (LSAV) framework that leverages the flexibility of SDN architecture in ISP networks and employs a lightweight filtering mechanism that considers the cost of operation to maintain high performance. Our setup for the proposed mechanism reflects client–server communication through an ISP SDN, and we use the entry points to eliminate malicious user requests targeting the systems. We then propose a novel algorithm on top of this setup to introduce a new and more efficient approach to existing mitigation methodologies. In addition to filtering the traffic against IP spoofing-based DDoS attacks, LSAV also prioritizes low resource consumption and high performance in terms of delay and bandwidth. With this approach, we believe that ISPs can effectively defend against IP spoofing-based DDoS attacks while still preserving low resource consumption for the infrastructure and high-quality internet services for their customers.

Key words: Software-defined network, source address validation, IP spoofing prevention, security, DDoS mitigation, DoS mitigation

1. Introduction

Software-defined networks (SDNs) are a type of networking architecture that separates the control plane from the data plane. The control plane determines how data are transmitted and the data plane responsible for forwarding the data to their destination. This separation of control and data planes allows for more flexible and efficient network management, since the control plane is centrally manageable and the data plane can be configured for specific requirements.

SDNs have numerous advantages, including the ability of configuring and reconfiguring networks in an easier way, enhanced network visibility and monitoring, and minimal complexity and low cost. The usage of SDNs is increasingly being adopted by service providers and enterprise networks to improve network performance and agility for specific needs.

Using SDN architectures for the detection and prevention of DDoS attacks is a trending topic today because of the flexibility and visibility of the environment. DDoS attacks are a common type of cyberattack

*Correspondence: ali.karakoc@boun.edu.tr

that aims to disrupt the availability of a target service by creating a flood or exhaustive resource consumption. These attacks may have significant impacts on businesses, as they can result in loss of revenue and reputation. In addition to businesses, DDoS attacks are a particularly noteworthy threat to ISPs and their customers.

Mitigation of DDoS attacks is one of the most significant aspects of the network security domain. One technique for achieving this is through source address validation (SAV), which checks the validity of incoming traffic from a legitimate source. One of the initial methods for preventing DDoS attacks was to use access control lists (ACLs) with IP network prefixes [2]. However, these solutions do not offer a high level of granularity while filtering traffic, as the IP network prefixes contain a set of IP addresses. While more granular filtering mechanisms exist, they have some significant challenges [3]. Most of these mechanisms are based on a binding database, which can be memory-intensive when IPv6 networks are considered. Additionally, their implementation may be difficult on complex systems, and it can be problematic in terms of performance and scalability. Other methods include the examination of the source path of incoming traffic via border gateway protocol (BGP) announcements at intermediate nodes, or using statistical methods and artificial intelligence to detect and prevent DDoS attacks based on traffic metadata. Unfortunately, even primitive SAV methods are not widely implemented, and studies have shown that a significant proportion of autonomous systems (ASes) are not able to utilize SAV properly. One study showed that 25% of the ASes tested had problematic or inadequate SAV implementation [4].

In this paper, we propose a SDN-based algorithm to detect and prevent IP spoofing-based DDoS attacks. Our design simplifies SAV by eliminating the use of a binding database and preventing disclosure of symmetric keys to the client side, as it performs validation through traffic characteristics. This implementation reduces the memory requirements of legacy solutions, which might create problems due to the high number of IPv6 addresses, while also decreasing key distribution complexity and enhancing the security of the keys. Furthermore, it provides completely granular filtering. We also implement an existing and widely used solution in conjunction with LSAV and demonstrate their effective implementation together in our SDN system to showcase the compatibility of our solution. LSAV offers several key advantages, including:

- Complete granular validation of source IP addresses,
- A minimal memory requirement,
- Authentication on a per-packet basis,
- A CPU consumption level comparable to traditional algorithms,
- The ability to take advantage of the flexibility of SDN,
- And a high level of performance for end users.

2. Related work

To effectively combat IP spoofing-based attacks, it is essential to consider both legacy mitigation techniques and the advantages of implementing SDN architecture. In this paper, we conduct a comprehensive review of the existing works in these domains, including different approaches for combating IP spoofing attacks and the utilization of SDN to improve agility and flexibility.

2.1. IP spoofing mitigation

DDoS attacks are a widespread form of attack that aim to interfere with the accessibility of a target service. These attacks can be observed at any layer of the Open Systems Interconnection (OSI) model and can manifest in different types. In the context of this research, our primary focus includes IP spoofing-based DoS and DDoS attacks and the detection and prevention methods to mitigate them. Over the years, a plethora of approaches have been introduced to safeguard against these types of attacks. In this section, we will examine the primary well-established and effective methodologies to mitigate IP spoofing-based attacks.

2.1.1. Source address validation

IP spoofing is a technique used in many types of cyberattacks, in which the attacker sends malformed packets with randomly chosen or predetermined source addresses that belong to other parties. According to statistics, nearly 42% of all web applications are vulnerable to spoofing-based malicious activities by hackers [1]. It is quite difficult to trace the actual source of the attack, and that makes it harder to mitigate them [2]. To address this problem, various SAV techniques have been introduced to validate the authenticity of incoming traffic. These techniques can effectively mitigate the impact of IP spoofing-based DDoS attacks. However, their efficiency and effects can be reduced by more sophisticated attacks.

2.1.2. Ingress filtering

One of the earliest and most straightforward techniques to mitigate IP spoofing attacks was introduced in [2]. According to this mechanism, an ingress router uses an ACL and only incoming packets with source IP addresses that belong to the known IP prefix of the ISP are allowed to pass through, while the other ones are dropped. While this method is quite simple and efficient, it has some limitations. For instance, an attacker may still be able to spoof an IP address within the same subnet prefix as the ISP. Additionally, manual operations to update the router ACLs may not be practical, and forgetting to update the ACL could cause the legitimate traffic to be dropped. As a result, implementing this technique for legacy systems may be difficult.

2.1.3. Reverse path forwarding

Another way to combat IP spoofing attacks is to use the technique introduced in RFC 3704 [5]. This technique entails checking the source address of incoming packets against the routing table of the source interface of the intermediate routers to verify if there is a best path to the source from there. If the packet passes this test, it is forwarded with respect to the destination routing table on the exit interface. This method is called strict reverse path forwarding (strict RPF). There are several other types of RPF, including feasible path reverse path forwarding (feasible RPF), which searches for alternative routes, and loose reverse path forwarding (loose RPF), which accepts any kind of routes including the default route or ignores the default route and checks the other possible routes.

2.1.4. Source address validation improvement (SAVI)

The SAVI framework introduced in RFC 5210 addresses IP spoofing problems in access network, inter-autonomous system (Inter-AS), and intra-AS scenarios [6]. In the access network, in the case of IP spoofing, the filtering mechanism is implemented for the same source prefix. If a host connects to the network and it is authenticated, the session key information is used as a binding anchor. Otherwise, the physical port is used.

Besides this, the edge router implements intra-AS filtering using the same approach as in RFC 2827. For inter-AS cases, there are two approaches. If the ASes are connected and compatible in terms of SAV implementation, they can share a validation table and perform filtering based on the IP prefixes in the table. If the ASes are not directly connected, another method involves tagging the IP packet with a common messaging key determined by a central alliance authority. If the packet is received at the destination AS, the tag is omitted and the source IP is validated. This approach allows performing effective IP spoofing prevention in different network scenarios.

The first-come first-served (FCFS) source IP binding technique was introduced in [7] and aims to validate the source IP address via setting the source port as a binding anchor. However, this implementation has some possible conflicts if used in conjunction with other SAV methods, such as DHCP-based SAV [3]. For the IPv6 case, the secure neighbor discovery (SEND) approach, introduced in [8], provides a granular SAV mechanism. This approach can only be implemented on IPv6-enabled devices. In [9], another approach for creating a SAV binding table was proposed. This method involves getting information about the IP address via a DHCP server and using the binding anchor of the source as a SAV lookup table. Usage of multiple SAV techniques in a single node, such as FCFS, DHCP, and SEND, and their potential conflicts, were analyzed in [10]. To address inconsistencies between different SAV methods, that study introduced a range of prioritization-based approaches.

The enhanced feasible path uRPF (EFP uRPF) method, proposed in [11], aims to solve the inconsistencies of the unicast RPF (u-RPF). However, this solution may cause false negative events. To achieve more accurate SAV, the distributed source address validation (DSAV) methodology was proposed for intra-AS and inter-AS SAV in [12]. This technique uses a special protocol message, through which source and destination IP information is propagated along the path to create a more precise SAV operation.

MANRS, or Mutually Agreed Norms for Routing Security, is a global organization that aims to enhance the security and resilience of the internet's routing infrastructure. One of the key actions that participating parties crucially commit to is the assurance of SAV. This means that IP packets are only sent from legitimate source IP addresses, which helps prevent IP spoofing and the potential for DDoS attacks [13]. The organization enforces the use of the RFCs mentioned above in this section to provide SAV.

In [14], the introduced protocol, RISAV, allows for secure communication between ASes using the RPKI identity system. This protocol ensures that all participating systems validate the source IP address. Another study introduced an approach called the spoofing prevention method (SPM) for filtering spoofed IP packets [15]. This method enables routers closer to the destination IP of a packet to verify the authenticity of the source IP address, in contrast to standard ingress filtering, which works at ingress routers. In [16], the authors presented the SAVE protocol, which provides routers information to validate source address. SAVE messages spread valid source address information from the source to all destinations, allowing each router to construct an incoming table that links the incoming interface and a set of valid source address blocks. In [19], a hash-based IP traceback technique was presented that generates audit trails for the traffic and can trace the origin of a single IP packet in the recent past. In [20], the presented technique aims to prevent attackers from launching attacks outside the IPv6 edge network using malformed source addresses at fine granularity. In [17], the authors introduced a technique that involves node sampling and marked packets using distance-based probabilities. Another study described IP spoofing-based attacks and evaluated the existing proposed methods for detection and prevention [21]. That work introduced a new algorithm, inspired by the hop count filtering (HCF) technique, which alters the learning phase of HCF to incorporate all possible hop count values. In [22], a trust-based approach was

proposed using a Bayesian inference model to evaluate the trustworthiness of access routers while forwarding packets without modifying their source IP addresses. The authors introduced a mechanism to protect against spoofing threats for IPv6 tunnels [23]. This mechanism allows using the optional field in IPsec's ESP frame, specifically the padding area, to mark the IPv6 source address before encapsulating the IPv6 packet into an IPv4 datagram. On the receiving end, the device verifies the IPv6 source address after decapsulation of the packet. In [24], a novel IP spoofing filtering technique called virtual anti-spoofing edge (VASE) was introduced, which filters IP spoofing-based attacks by sampling and on-demand configuration to minimize unnecessary overhead during peace time.

2.1.5. SDN-based SAV

There have been several proposals for implementing SAV mechanisms in SDN. In [25], the authors analyzed the features, limitations, and gaps of SAV techniques and identified potential opportunities for future research. The SAVSH mechanism aims to maximize SAV accuracy while minimizing the number of SDN switches required in a traditional AS [26]. However, the solution performs filtering on the prefix level, which does not provide granular SAV. The virtual source address validation edge (VAVE) architecture uses OpenFlow and a NOX controller to perform a centralized decision-making process for validation rules [27]. This solution uses a holistic view to detect flow change at intermediate devices. The dynamic framework for SAVI (D-SAVI) implements a dual-level architecture and priority-based validation [28]. Since the solution uses a binding table, in the case of a high number of IP addresses, the lookup operation may create performance issues. To evaluate the performance of different SAV techniques in the SDN architecture, a testbed called TestSDN was proposed in [29]. In [30], a solution for the internet of Things (IoT) was presented that endeavors to address the issue of ARP spoofing attacks through the implementation of a secure SDN-based architecture. In the work presented in [31], an architecture for a SDN-based CDN network was introduced to detect spoofed IP addresses and create mitigation against them. As another approach, CAuth autonomously blocks spoofed queries while authenticating legitimate queries [32]. Using manipulation of OpenFlow control messages, CAuth enables collaborative work between the client and server network. The study in [33] presents a new model for mitigating the effects of DDoS attacks on NFV. It was designed specifically for individual users, particularly gamers and online streamers. The work in [34] involves a design for HyPASS, a hybrid-SDN solution for host discovery, flow entry configuration during handshaking, and detection and prevention of forged payloads.

2.2. DoS and DDoS mitigation

In [35], the authors shed light on how to utilize the benefits of SDN to mitigate DDoS attacks in cloud computing environments and how to secure SDNs from possible DDoS attacks. In [18], the authors made use of the matching pursuit algorithm to identify resource depletion DDoS attacks. In [36], the authors presented a framework for a detection and prevention mechanism against DDoS attacks in SDN environments. They introduced the deployment of a DDoS detection trigger mechanism on the data plane, followed by a machine learning algorithm combining K-means and KNN to analyze a SDN-based DDoS defense mechanism, named efficient and low-cost DDoS defense (ELD). In another paper, the authors proposed an algorithm named SYN-Guard for detecting and preventing SYN flooding in SDN networks [38]. The implementation of SYN-Guard on the SDN controller checks incoming TCP connection requests and creates forwarding rules for verified SYN requests.

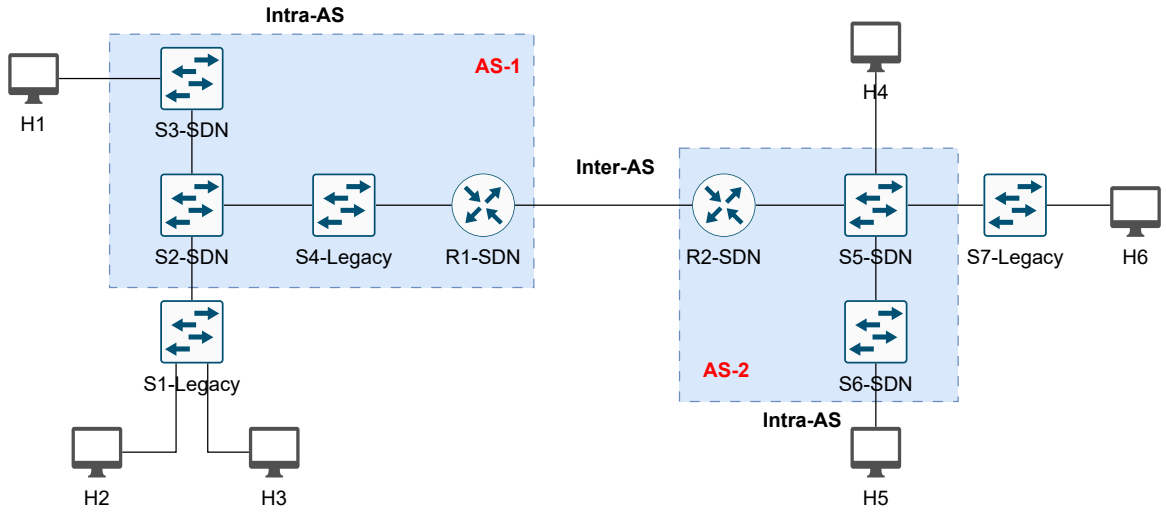

Figure 1. Protection perimeter.

3. SDN-based SAV framework

Our IP spoofing mitigation framework is based on an OpenFlow application that filters the traffic against IP spoofing attacks. The application filters the malformed traffic by comparing a precalculated hash string and hash resulting from the traffic characteristics of the incoming packet. A quite basic architecture is taken as a reference for our implementation in the following subsection.

3.1. Protection perimeter and use cases

To show where the LSAV will work in the system, a sample architecture is shown in Figure 1. In the topology, there are two different ASes and each AS has legacy and SDN-supported components. The colored areas represent the protection perimeter of the LSAV algorithm.

3.1.1. Ingress interface

The ingress interface is the entry point of the traffic from the source host to the protection perimeter. The most efficient approach is eliminating the bad traffic close to the user as much as possible [3]. Therefore, the LSAV is implemented on the ingress devices to eliminate the spoofed traffic as early as possible.

3.1.2. Intra-AS interfaces

The spoofed traffic can be eliminated in the intermediate routers of the AS network like in [5]. However, this technique was not widely adopted by ISPs due to the complexity and potential workload. Therefore, the traffic passing through the intermediate network components will not be analyzed after the ingress component. Those components will be considered inside the protection perimeter. For instance, in the given architecture in Figure 1, traditional switches located between the SDN switches are also within the protection perimeter. Given that it is not connected directly to a host and that ingress filtering is performed by the ingress SDN switch, it can be deemed to be situated within the protection perimeter.

3.1.3. Inter-AS interface

Traffic validation between ASes poses a significant challenge due to the varying technologies and structures of different ISPs. In the literature, there are several existing methods for inter-AS validation, as outlined in [6] and [14]. However, these methods may not provide granular verification of source IPs when the spoofed IP originates from a legitimate IP prefix. While the LSAV algorithm is not specifically designed for use at this interface, it can be utilized in conjunction with existing inter-AS validation techniques to enhance granularity.

3.2. LSAV: Lightweight source address validation

In this work, we aim to mitigate IP spoofing-based attacks in a SDN environment by carefully analyzing the interfaces and entry points where the solution is to be applied. Previous researchers proposed various solutions for different types of interfaces, such as access networks, inter-AS, and intra-AS in traditional networks. Our focus is specifically on ingress points, and we present a comprehensive filtering mechanism that combines existing algorithms from relevant RFCs with a new algorithm developed by us to provide efficient and granular protection against IP spoofing-based DDoS attacks. As depicted in Figure 2, our filtering mechanism can be used with legacy algorithms that comprise two parts: known methods to mitigate IP spoofing followed by LSAV to provide the highest granularity for filtering traffic.

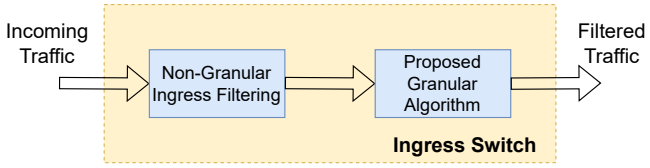

Figure 2. Combination of legacy algorithm and LSAV.

3.2.1. Nongranular ingress filtering

The nongranular ingress filtering mechanism is an adoption of RFC2827 to the SDN environment. Since IP spoofing relies on the change of the source address, the mechanism in RFC2827 aims to reduce the effect of this via ACLs. Since each ISP provides the IP addresses to the customers, the information of the subnet prefixes of each switch can be used to filter the traffic. For example, if the source IP address of the incoming traffic at the ingress switch is not in the ACL that the ISP keeps for the customer for that region, the traffic drops. Each source IP should come from the corresponding region that the ISP assigned on the switch. Even if it seems very effective, there are problems with this solution. First, in conventional systems, the IP prefixes for the corresponding switch are not updated regularly. That may lead to dropping the legitimate traffic. In our structure we provide this service by using a SDN controller and that gives more flexibility and control over the ACL.

The other problem of the method is the allowed IP ranges. Since the mechanism compares the incoming IP addresses with the list of IP prefixes, the attacker can still use the IP addresses in the allowed list and this method cannot provide filtering for those types of attacks. Therefore, to achieve high granular filtering, we propose another approach. While our approach can be used on its own, it can also be used with this nongranular mechanism to increase the performance. In our system, we use this nongranular mechanism as an initial filter to reduce the amount of the bad traffic to increase the performance of LSAV. Since our solution checks each

incoming packet by cryptographic calculation, subnet-based filtering reduces the amount of bad traffic in the simplest way before our solution does this operation.

3.2.2. Granular SAV (LSAV algorithm)

One of the ways to mitigate DDoS attacks in legacy networks is using granular SAV techniques. These techniques involve creating a binding table that associates a unique source IP with a binding anchor, such as a physical port or MAC address, on the edge switch. When incoming traffic is received, the source IP and switch port tuple is checked against the binding table. If the tuple is valid, the traffic is forwarded; otherwise, it is dropped. This approach provides a granular level of SAV and can be effective in mitigating IP spoofing-based DDoS attacks. However, it requires the connection of each customer to the edge switch through a unique physical port and that may not be feasible in all cases [3]. The well-known layer-2 unique properties to be used as binding anchors are defined in [3] as follows:

- IEEE extended unique identifiers (EUI-48 or EUI-64)
- Physical port of the host
- Wireless link association
- MAC address and customer relationship
- PPP session identifier, virtual ATM channel, L2TP session identifier
- Tunnels (IP-in-IP, GRE, or MPLS label-switched path)

In LSAV, we use the optional IP headers to put extra information for the authenticity of the incoming traffic. To do this, there are two different types of IP headers that can be utilized. For IPv4 networks, the options field can be used to carry custom information. The options header in IPv4 is a part of the IP packet header that can be used to carry additional information or options that are specific to a particular implementation or protocol. The options header is optional and is not used for every packet. The options header can be used to carry a variety of different types of information, including router alerts, time stamps, security, and record routes.

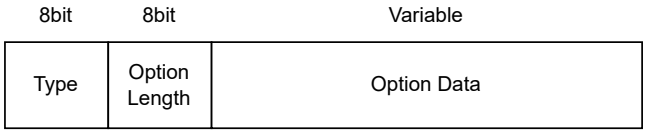

Figure 3. Options header.

The IPv4 options header is used to provide additional information or functionality beyond the basic header fields in an IPv4 packet, such as source routing and time stamping. However, there are also unregistered option types that can be used for other purposes [39]. Despite this, the use of the options header is uncommon, and typically it is only used for troubleshooting purposes. The size of the options header can vary, depending on the options that are included. If the options header is not used, it is simply left blank. The layout of the IPv4 options header is shown in Figure 3. The first 8 bits represent the option type, the next 8 bits indicate

the length of the options value, and the remaining portion has a variable length as specified in the previous section. For the LSAV algorithm, the options header will be utilized for transmitting a 32-byte SHA256 HMAC payload. The maximum size for the options header is 40 bytes. For IPv6 traffic, instead of the options header, another multipurpose header hop-by-hop extension header is used to transport the same 32-byte SHA256 HMAC payload [40].

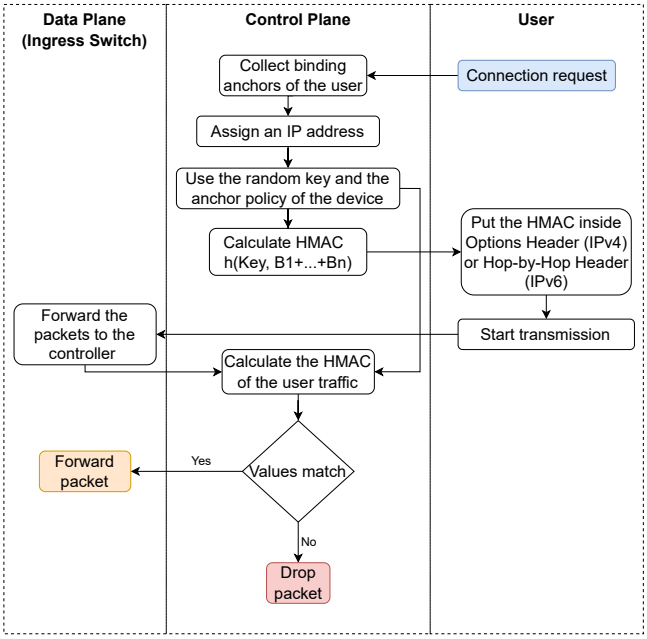

Figure 4. Granular SAV for IPv4 and IPv6 at ingress switch.

In Figure 4, the flow diagram of LSAV algorithm is shown. The SDN application generates a random key with a lifetime and binding anchor policy for each network device. Once the key's lifetime has expired, a new key is generated within the device. Since the key never leaves the device, there is no key distribution scheme. Each device only retains one secret key and binding anchor policy that are used for all source traffic registered to the device.

When a client attempts to connect to the network, the ISP authenticates the user by collecting layer-2 binding anchor information based on the binding anchor policy and the assigned source IP address of the user. It then uses the device's random key to create a unique HMAC using the following equation:

$$HMAC = h(Key, B_1 + B_2 + ... + B_n), \tag{1}$$

where h() is the HMAC function, Key is the secret key, and B_i is a binding anchor. Once the HMAC value is calculated, it is sent to the customer through an out-of-band channel. This process may involve the use of an agent on the customer's home router. As long as the binding anchor policy remains the same and the key has a valid lifetime, the customer is expected to include this HMAC value in the options header when sending the packets. If the binding policy or key is altered, the device calculates a new HMAC for the user with the new key or binding anchor policy and sends it to the user.

Once the authentication is complete, the users include the HMAC value in each IP packet they send. When the packet reaches the ISP's edge device, LSAV uses the secret key and its binding anchor policy to

Table 1. Simulation parameters.

Simulation parameter	Values
Mininet version	2.3.0
ONOS version	2.7.0
User traffic protocol	TCP
User traffic type	On-Off Poisson traffic
Lambda	10
User traffic rate	10 MBps
User payload size	1 MB
Binding anchor	Physical port

recalculate the HMAC from the traffic itself and compare it to the HMAC value provided in the options header. If the values match, the packet is forwarded; otherwise, it is dropped.

Previous solutions outlined in recent RFCs require the use of a binding table at the device, resulting in an increased memory requirement. This can become problematic when dealing with large numbers of IP addresses, as is the case with IPv6. Additionally, some of these solutions involve distributing unique keys to each user to perform HMAC calculations, which can be risky as the keys are not kept within the device and increase the complexity of the process.

LSAV eliminates the need for a binding table at the device and keeps the secret key within the device, making the authentication process more secure and efficient. Additionally, this solution is designed to be used in conjunction with SDN, but can also be implemented on legacy devices.

4. Experimental setup

LSAV was simulated using a SDN environment and the parameters specified in Table 1, as explained in the following subsections.

4.1. Simulation setup

Our simulation setup is based on a straightforward SDN design that includes three ASes. The simulation topology is illustrated in Figure 5.

The data plane is composed of OpenFlow switches and routers simulated via Mininet [41]. The topology includes two legitimate users that perform normal traffic and one attacker who initiates an IP spoofing-based DoS attack [42]. One of the legitimate users is connected to the same switch as the attacker, while the other is connected to a switch in a different AS. The switches and routers are all controlled by a central controller.

The control plane includes an ONOS controller and applications built on top of it [43]. In this simulation, we only apply the LSAV algorithm to ingress switches S1 and S2. When a packet arrives at the ingress interface of these switches, it is sent to the controller, where the applications running on top of it perform filtering on the traffic.

4.2. Performance metrics

To evaluate and compare the performance of LSAV in handling IP spoofing-based DDoS attacks, the following key performance metrics are utilized: CPU and memory usage, packet latency, packet loss, total traffic volume,

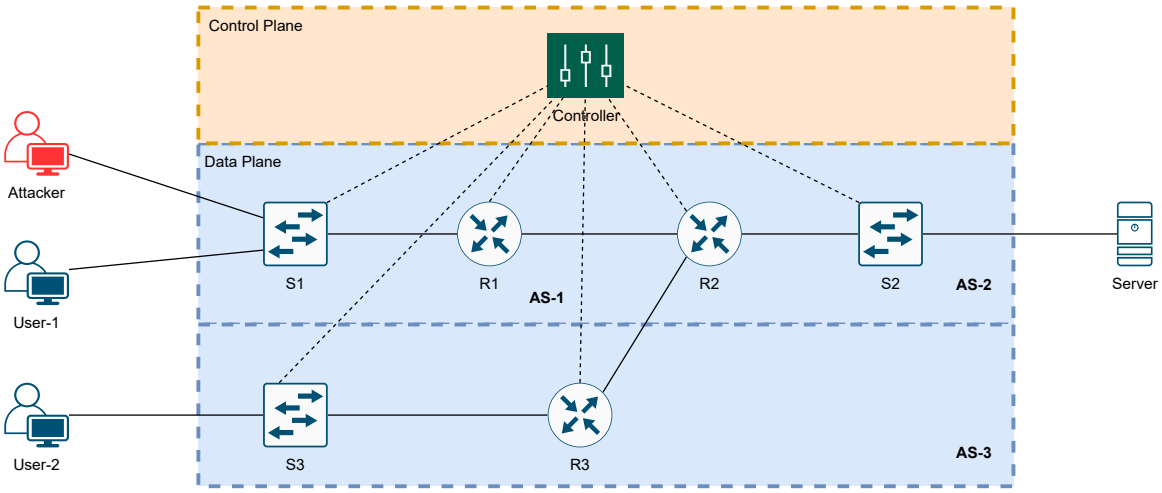

Figure 5. Test network used in the simulations.

Table 2. Attacks and tools used in the simulations.

Attack type	Attack	Tool
DoS (IP spoofing)	SYN Flood	Hping3
DoS (IP spoofing)	UDP Flood	Hping3
DoS (IP spoofing)	ICMP Flood	Hping3

service availability, and IP spoofing prevention rate.

4.3. Simulation scenario

In order to evaluate the effectiveness of LSAV, simulations are conducted in which User-1 and User-2 are legitimate clients who engage in normal activity while the attacker is a malicious actor who uses IP spoofing-based DoS attacks. The target of the attack is a server hosting a TCP server. During these scenarios, the attacker will launch IP spoofing-based DDoS attacks against the server while the legitimate users engage in TCP traffic with the server. To create a realistic simulation, traffic patterns generated by the clients are modeled using a Poisson distribution. The performance of the algorithm is assessed using various types of DoS attacks, including L3-L4 attacks, and measuring its efficiency against each attack type separately. This is done by utilizing the tools and the attack types listed in Table 2.

For the LSAV algorithm, we calculated our HMAC with physical port information to keep the environment simple. To evaluate the efficiency of our solution, two simulation sets were run using legacy algorithms. First, LSAV was compared to the ingress filtering mechanism used in real-world environments, as described in RFC 2827. To compare the best-case scenario of the legacy algorithm, simulations only had one subnet in the ACL. This comparison aimed to assess the resource consumption and performance of our algorithm against the best case of the legacy algorithm and determine its practicality in real-life scenarios. In addition to comparing the two algorithms individually, we also evaluated the combination of the legacy algorithm and LSAV algorithm (legacy algorithm is first and LSAV algorithm is second in series). To conduct the comparisons, the algorithms

were tested against the three different attacks mentioned in a previous section (SYN flood, ICMP flood, and UDP flood).

For the second simulation set, we compared the LSAV algorithm with a legacy algorithm that maintains a binding anchor table. To make the comparison, we used the legacy algorithm outlined in RFC 5210, as the mechanisms are similar. The algorithms were evaluated under the SYN flood attack type for varying attack rates and sizes of the binding anchor table. This comparison aimed to assess the efficiency of the algorithm in different traffic density scenarios and with different sizes of the lookup table.

5. Simulation results

In this section we first show the performance of the LSAV algorithm compared to the legacy algorithm. We then show end-to-end delay measurements for the different scenarios.

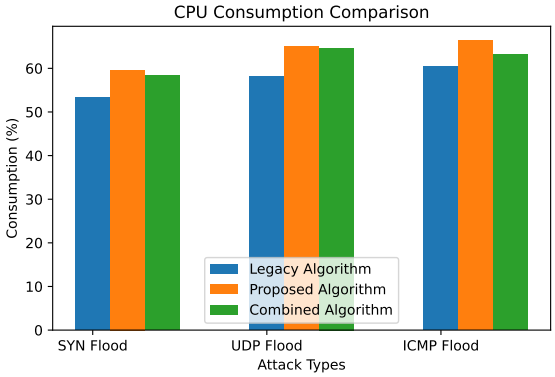

Figure 6. Comparison of CPU consumption of LSAV algorithm, legacy algorithm (RFC 2827), and combination of two algorithms.

5.1. Evaluation of the performance metrics

5.1.1. CPU usage

Figure 6 displays the CPU utilization of different algorithms during IP spoofing-based DoS attacks. As per the simulation, the legacy algorithm (RFC 2827) performed better than the LSAV algorithm for all three types of attacks, with a difference of approximately 8% in total CPU usage. However, when both algorithms were combined, it was evident that the CPU usage decreased. In the simulation, the IP prefix lookup table of the legacy algorithm had only one entry. It can be inferred that as the number of entries increases, the processing requirements will also increase. Therefore, it can be concluded that the LSAV algorithm and the legacy algorithm have similar usability and processing requirements in terms of resource consumption for the best-case scenario of the legacy algorithm.

The left figure in Figure 7 compares the CPU performance of the LSAV algorithm and the legacy algorithm (RFC 5210) under different attack rates and sizes of the binding anchor table. As shown, after reaching 10,000 packets per second, all algorithms exhibit poor performance due to network saturation and the switch's inability to handle the high traffic volume. This results in the CPU consumption of the controller being around 60%. However, for lower attack rates, it can be seen that as the number of entries in the legacy algorithm increases,

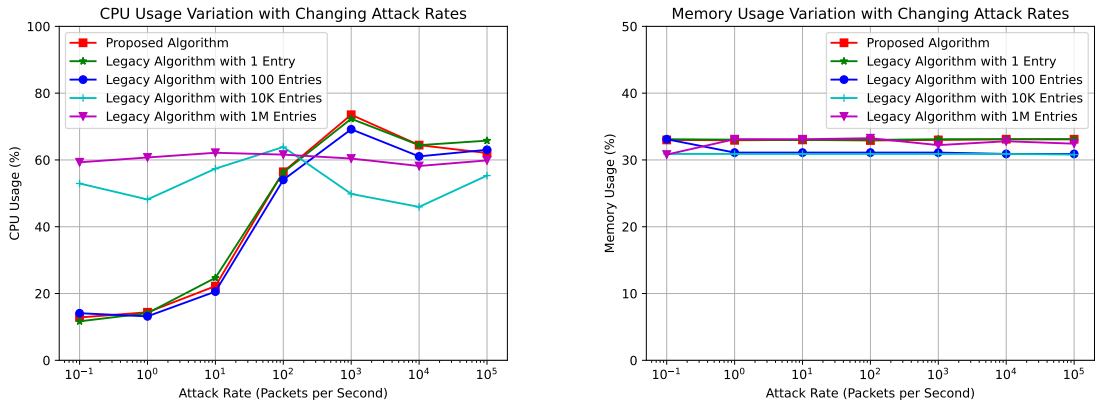

Figure 7. CPU and memory usage variations of LSAV algorithm and the legacy algorithm (RFC 5210) with different sizes of lookup table under different SYN flood attack rates.

its performance deteriorates. Since the LSAV algorithm does not use a lookup table, an increase in the number of users does not impact its CPU performance, making it more scalable and suitable for environments with high user density.

5.1.2. Memory usage

For the first simulation set, all three algorithms demonstrated similar memory consumption of about 29% as the lookup table of the legacy algorithm (RFC 2827) was kept small, which included all operating system processes of the controller. However, if the size of the lookup table of the legacy algorithm increases, it is expected to lead to higher memory consumption.

Moreover, as the number of users grows, the legacy algorithm (RFC 5210) using binding anchor tables will require increasing amounts of memory. The right graph in Figure 7 shows similar results for all algorithms. Despite the binding anchor lookup table of the LSAV algorithm having 1 million entries, it does not significantly impact the controller's 4 GB RAM due to the O(1) memory requirement of the LSAV algorithm compared to the O(n) requirement of the legacy algorithm that uses 10 MB of additional memory for 1M entries.

5.1.3. Delay and packet loss

In the first set of simulations, to evaluate the delay during the attack, legitimate users (User-1 and User-2) sent ping requests to the server. Table 3 presents the delay measurement results of this scenario. It is evident from the results that, for all three attack types, the legitimate user (User-1) connected to the same switch as the attacker experienced more delay than the other legitimate user (User-2) who was connected to a different AS. In the case of the UDP flood attack, User-1 did not receive any ICMP response and all packets were dropped due to the limited resources of the switch in the simulation, which was a VM-based environment. On the other hand, User-2 did not lose any packets during the entire attack scenario and the delay values for this user were relatively low. The same characteristics were observed for the legacy algorithm (RFC 2827) as well, which confirms the effectiveness of the LSAV algorithm in real-world settings.

The second simulation set compared the legacy algorithm from RFC 5210 and the LSAV algorithm during various intensities of SYN flood attacks. The bottom figure in Figure 8 displays the delay values for each algorithm for different attack rates. After an attack rate of 100 packets per second, all algorithms experienced an increase in delay. This effect is particularly noticeable as the number of binding anchors in the legacy algorithm's lookup table increases, leading to a delay of about 20 s, even at low attack rates, when the lookup table has 1 million entries.

Table 3. Delay and packet loss table for LSAV algorithm, legacy algorithm (RFC 2827), and the combined algorithm under different attack types.

Algorithm	Delay (%)						Packet loss (%)					
	SYN flood		UDP flood		ICMP flood		SYN flood		UDP flood		ICMP flood	
	User-1	User-2	User-1	User-2	User-1	User-2	User-1	User-2	User-1	User-2	User-1	User-2
Legacy	861.5	56.8	-	204.7	556.5	41.0	75.9	0	100	0	91.6	0
LSAV	1228.5	46.5	770.9	443.3	2032.5	159.4	89.6	0	80.7	0	82.2	0
Combined	699.1	50.3	-	123.4	797.5	30.3	81.8	0	100	0	86.6	0

Table 4. Traffic and packet loss table for LSAV algorithm, legacy algorithm (RFC 2827), and the combined algorithm under different attack types

Algorithm	Total transfer (KB)						Packet loss (%)					
	SYN flood		UDP flood		ICMP flood		SYN flood		UDP flood		ICMP flood	
	User-1	User-2	User-1	User-2	User-1	User-2	User-1	User-2	User-1	User-2	User-1	User-2
Legacy	732	9979	206	9453	348	1097	93	15.9	96.6	28.4	91.8	13,1
LSAV	1285	12237	226.1	9520.8	645	7454.6	94.8	6.6	96.3	28.5	96.9	53
Combined	337.6	12257.2	218.8	11252.6	436.9	12380.2	92.3	1.7	96.2	19.3	92.6	1.9

5.1.4. Traffic rate and packet loss

For the first simulation set with the legacy algorithm proposed in RFC 2827, Table 4 shows the simulation results for regular TCP traffic from legitimate users to the server. In the presence of the three attack types, the total data transferred from User-1 decreased significantly, while the traffic from User-2 remained high. The packet loss values aligned with the results of the ping simulation. Since all the network devices were connected to the same controller, User-2 experienced some packet loss with all three algorithms. This suggests that controller task allocation or using separate edge devices as controllers should be considered.

The LSAV algorithm in the bottom figure in Figure 8 was compared to legacy algorithm RFC 5210 with varying numbers of binding anchor entries and SYN flood attack rates. The total traffic transfer decreased significantly when the attack volume surpassed 100 packets per second. However, the legacy algorithm with 10K entries or more experienced a greater drop even at low attack rates, indicating its limitation when dealing with high numbers of users. The LSAV algorithm performed well with the available resources. In the right figure in Figure 8, the same effect can be observed for packet loss. All algorithms started to perform poorly once the rate reached 100 packets per second due to the switch's network bandwidth consumption. This effect was more pronounced for the legacy algorithms with 10K or more entries, even at low attack rates.

5.1.5. Prevention rate and availability

The LSAV algorithm completely blocked all spoofed packets as it only forwarded authenticated packets by design. However, if a legitimate user were a part of an infected botnet, none of the source IP validation

algorithms would be effective in preventing a DDoS attack, as it falls outside the scope of SAV capabilities. If there is no options header or the options header is used for another reason for the legitimate user, the LSAV algorithm may block the legitimate user, which is a negative aspect of LSAV. However, that is unlikely to happen.

Despite being an efficient and straightforward algorithm, the legacy ingress algorithm RFC 2827 lacks the capability for granular filtering. As a result, it allows traffic from whitelisted IP prefixes in the case of the IP spoofing-based DoS attacks in the simulation. In addition to achieving a full prevention rate, the LSAV algorithm also demonstrated comparable performance and resource consumption to the legacy algorithm, making it highly practical. While the legacy algorithm in RFC 5210 offers detailed filtering, its resource consumption grows as the number of users increases.

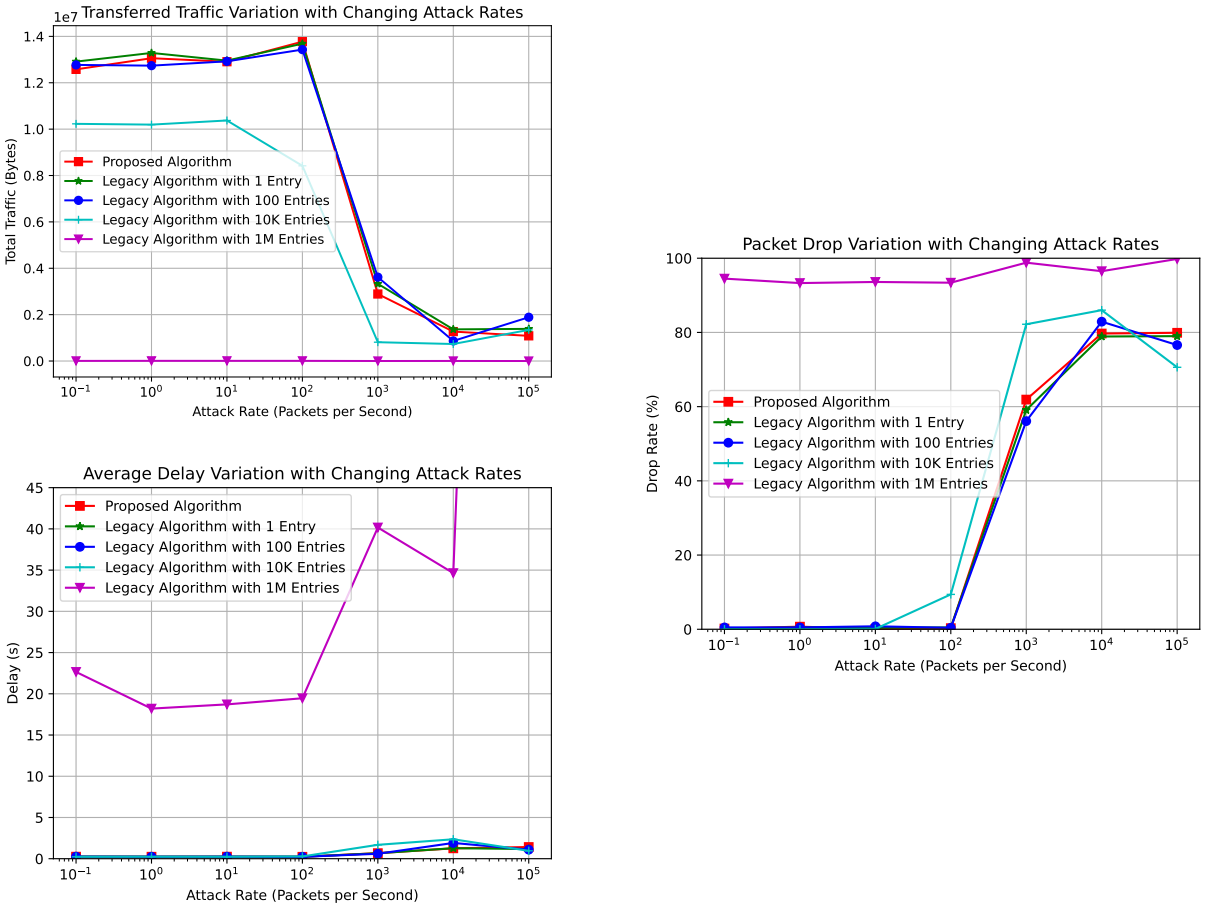

Figure 8. Traffic characteristic variations of the LSAV algorithm and legacy algorithm (RFC 5210) with different sizes of lookup table under different SYN flood attack rates.

5.2. Benefits and limitations of LSAV

LSAV offers an option type and value that may not be recognized by network devices that are not compatible with the solution. However, as the LSAV mechanism is within the control of the ISP and a part of the SDN, this

issue can be prevented by removing the options header from the packet after authentication, before forwarding it to another domain.

Another challenge is providing the HMAC values to users and requiring them to use them for each packet. This would require a user agent on the home router for all ISP customers, which may not be feasible in real-world usage.

Legacy SAV algorithms rely on ACLs or binding tables to function. However, when a high volume of users are connected to the same edge device, this data storage can cause problems with resource usage and delay. This problem is expected to be exacerbated by the increase in number of users caused by the transition to IPv6. The LSAV solution does not rely on any binding tables or ACLs, making it less affected by an increase in number of clients in terms of resource consumption and delay.

5.3. Security and privacy considerations

Our solution prioritizes privacy and security by keeping the key on the device at all times. According to RFC 5210, session keys for packet authentication are distributed to clients. However, if the key is leaked to attackers, unauthorized users could use spoofed IP addresses to authenticate their packets. To mitigate this, our solution only sends HMAC values to the users. Even if an attacker were to obtain the HMAC, it would be useless as the validation is done on the edge device and the HMAC is compared to the user's actual traffic. Spoofed traffic does not possess the correct layer-2 unique characteristics, so the calculated HMAC at the edge device will not match the HMAC provided by the attacker in the options header.

The key used in LSAV is an ephemeral key, meaning that it has a limited lifespan. Even if attackers were to obtain the key or binding anchor policy, they would not be able to validate packets with them, as the spoofed traffic fingerprint cannot be altered. The only way for an attacker to obtain a valid HMAC is through a brute force attack. If an attacker were to find a correct HMAC value through brute force, it would only be valid until the secret key's lifetime expires. However, this is highly unlikely due to the strength of the SHA256 HMAC algorithm and the changing nature of the ephemeral key.

If legitimate users are compromised, the attacker could launch a DoS or DDoS attack without using IP spoofing. In these scenarios, LSAV would not be effective, as it is designed to specifically address spoofing mitigation.

6. Conclusion and future directions

In this study, we have proposed a comprehensive solution for combating IP spoofing attacks in SDN networks. We began by outlining a basic SDN ISP architecture and highlighted the shortcomings of traditional methods in SDN environments due to their lack of precision and high resource requirements in filtering. LSAV combines new and established techniques. Our approach is efficient, with minimal resource usage. Additionally, the cryptographic operations in the LSAV algorithm has a negligible impact on packet delay. LSAV also takes into account potential security and privacy risks.

Our objective is to enhance the system and provide solutions for all architectural interfaces. Detecting IP spoofing is a difficult task and ideally should be done as close to the source user as possible. However, as not all ISPs have the capability to control at the ingress interfaces, it is crucial to develop solutions that cater to both intra-AS and inter-AS scenarios.

References

[1] Tekerek A. A novel architecture for web-based attack detection using convolutional neural network. Computers & Security 2021; 100. https://doi.org/10.1016/j.cose.2020.102096

[2] Ferguson P, Senie D. Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing. BCP 38, RFC Editor; 2000.

[3] Wu J, Bi J, Bagnulo M, Baker F, Vogt C. Source Address Validation Improvement (SAVI) Framework. RFC 7039, RFC Editor; 2013.

[4] Luckie M, Beverly R, Koga R, Keys K, Kroll JA et al. Network Hygiene, Incentives, and Regulation: Deployment of Source Address Validation in the Internet. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security; New York, NY, USA; 2019. pp. 465-480.

[5] Baker F, Savola P. Ingress Filtering for Multihomed Networks. BCP 84, RFC Editor; 2004.

[6] Wu J, Bi J, Li X, Ren G, Xu K et al. A Source Address Validation Architecture (SAVA) Testbed and Deployment Experience. RFC 5210, RFC Editor; 2008.

[7] Nordmark E, Bagnulo M, Levy-Abegnoli E. FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses. RFC 6620, RFC Editor; 2012.

[8] Bagnulo M, Garcia-Martinez A. SEcure Neighbor Discovery (SEND) Source Address Validation Improvement (SAVI). RFC 7219, RFC Editor; 2014.

[9] Bi J, Wu J, Yao G, Baker F. Source Address Validation Improvement (SAVI) Solution for DHCP. RFC 7513, RFC Editor; 2015.

[10] Bi J, Yao G, Halpern J, Levy-Abegnoli E. Source Address Validation Improvement (SAVI) for Mixed Address Assignment Methods Scenario. RFC 8074, RFC Editor; 2017.

[11] Sriram K, Montgomery D, Haas J. Enhanced Feasible-Path Unicast Reverse Path Forwarding. RFC 8704, RFC Editor; 2020.

[12] Li D, Huang M, Qin L, Geng N. Distributed Source Address Validation (DSAV) Framework. Tech. Rep., RFC Editor; 2022.

[13] "Mutually Agreed Norms for Routing Security," MANRS. Accessed 24 April 2023.

[14] Xu K, Wu J, Guo Y, Schwartz BM. An RPKI and IPsec-based AS-to-AS Approach for Source Address Validation. draft- xu-risav-02, 2022.

[15] Bremler-Barr A, Levy H. Spoofing prevention method. In: Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies; Miami, FL, USA; 2005. pp. 536-547.

[16] Li J, Mirkovic J, Wang M, Reiher P, Zhang L. SAVE: Source address validity enforcement protocol. In: Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies; New York, NY, USA; 2002. pp. 1557-1566.

[17] Akyuz T, Sogukpinar I. Packet marking with distance based probabilities for IP traceback. In: 2009 First International Conference on Networks & Communications; Chennai, India; 2009. pp. 433-438.

[18] Erhan D, Anarim E. Hybrid DDoS detection framework using matching pursuit algorithm. IEEE Access 2020; 8: 118912-118923. https://doi.org/10.1109/ACCESS.2020.3005781.

[19] Snoeren AC, Partridge C, Sanchez LA, Jones CE, Tchakountio F et al. Hash-based IP traceback. In: Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications; New York, NY, USA; 2001. pp. 3-14.

[20] Xie L, Bi J, Wu J. An authentication based source address spoofing prevention method deployed in IPv6 edge network. Lecture Notes in Computer Science 2007; 4490: 801-808. https://doi.org/10.1007/978-3-540-72590-9_121

[21] Mukaddam A, Elhajj I, Kayssi A, Chehab A. IP spoofing detection using modified hop count. In: 2014 IEEE 28th International Conference on Advanced Information Networking and Applications; Victoria, BC, Canada; 2014. pp. 512-516.

[22] Gonzalez JM, Anwar M, Joshi JB. A trust-based approach against IP-spoofing attacks. In: 2011 Ninth Annual International Conference on Privacy, Security and Trust; Montreal, QC, Canada; 2011. pp. 63-70.

[23] Ahmed AS, Hassan R, Ali ZM. Eliminate spoofing threat in IPv6 tunnel. In: 2012 8th International Conference on Information Science and Digital Content Technology; Jeju, Korea; 2012. pp. 218-222.

[24] Yao G, Bi J, Xiao P. VASE: Filtering IP spoofing traffic with agility. Computer Networks 2013; 57 (1): 243-257. https://doi.org/10.1016/j.comnet.2012.08.018

[25] Meena RC, Bundele M, Nawal M. Appraisal of source IP validation techniques in SDN. In: 2019 4th International Conference and Workshops on Recent Advances and Innovations in Engineering; Kedah, Malaysia; 2019. pp. 1-5.

[26] Chen G, Hu G, Jiang Y, Zhang C. SAVSH: IP source address validation for SDN hybrid networks. In: 2016 IEEE Symposium on Computers and Communication; Messina, Italy; 2016. pp. 409-414.

[27] Yao G, Bi J, Xiao P. Source address validation solution with OpenFlow/NOX architecture. In: 2011 19th IEEE International Conference on Network Protocols; Vancouver, BC, Canada; 2011. pp. 7-12.

[28] Zhou Q, Yu J, Li D. A dynamic and lightweight framework to secure source addresses in the SDN-based networks. Computer Networks 2021; 193: 108075. https://doi.org/10.1016/j.comnet.2021.108075

[29] Meena RC, Bundele M, Nawal M. RYU SDN controller testbed for performance testing of source address validation techniques. In: 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things; Jaipur, India; 2020. pp. 1-6.

[30] Aldabbas H, Amin R. A novel mechanism to handle address spoofing attacks in SDN based IOT. Cluster Computing 2021; 24 (4): 3011-3026. https://doi.org/10.1007/s10586-021-03309-0

[31] Mowla NI, Doh I, Chae K. An efficient defense mechanism for spoofed IP attack in SDN based CDNi. In: 2015 International Conference on Information Networking; Cambodia; 2015. pp. 92-97.

[32] Sahri N, Okamura K. Protecting DNS services from IP spoofing: SDN collaborative authentication approach. In: CFI '16, Association for Computing Machinery; New York, NY, USA; 2016. pp. 83-89.

[33] Singh AK, Jaiswal RK, Abdukodir K, Muthanna A. ARDefense: DDoS detection and prevention using NFV and SDN. In: 2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops; Brno, Czech Republic; 2020. pp. 236-241.

[34] Meena RC, Bhatia S, Jhaveri RH, Cheng L, Kumar A et al. HyPASS: Design of hybrid-SDN prevention of attacks of source spoofing with host discovery and address validation. Physical Communication 2022; 55: 101902. https://doi.org/10.1016/j.phycom.2022.101902

[35] Yan Q, Yu FR, Gong Q, Li J. Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: a survey, some research issues, and challenges. IEEE Communications Surveys Tutorials 2016; 18 (1): 602-622. https://doi.org/10.1109/COMST.2015.2487361

[36] Tan L, Pan Y, Wu J, Zhou J, Jiang H et al. A new framework for DDoS attack detection and defense in SDN environment. IEEE Access 2020; 8: 161908-161919. https://doi.org/10.1109/ACCESS.2020.3021435

[37] Wang YC, Wang YC. Efficient and low-cost defense against distributed denial-of-service attacks in SDN-based networks. International Journal of Communication Systems 2020; 33 (14): e4461. https://doi.org/10.1002/dac.4461

[38] Mohammadi R, Conti M, Lal C, Kulhari SC. SYN-Guard: An effective counter for SYN flooding attack in software-defined networking. International Journal of Communication Systems 2019; 32 (17): e4061. https://doi.org/10.1002/dac.4061

[39] Internet Protocol Version 4 (IPv4) Parameters. IANA.

[40] Deering S, Hinden R. Internet Protocol, Version 6 (ipv6) Specification; 1970.

[41] Keti F, Askar S. Emulation of software defined networks using Mininet in different simulation environments. In: 2015 6th International Conference on Intelligent Systems, Modelling and Simulation; Kuala Lumpur, Malaysia; 2015. pp. 205-210.

[42] S RR, R R, Moharir M, G S. SCAPY- A powerful interactive packet manipulation program. In: 2018 International Conference on Networking, Embedded and Wireless Systems; Bangalore, India; 2018. pp. 1-5.

[43] Berde P, Gerola M, Hart J, Higuchi Y, Kobayashi M et al. ONOS: Towards an open, distributed SDN OS. In: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking. Association for Computing Machinery; New York, NY, USA; 2014. pp. 1-6.

[44] Shi L, Lu Z, Qin P, Yao H. OpenFlow based spatial information network architecture. In: 2015 International Conference on Wireless Communications Signal Processing; Nanjing, China; 2015. pp. 1-5.

[45] McPherson D, Baker F, Halpern J. Source Address Validation Improvement (SAVI) Threat Scope. RFC 6959, RFC Editor; 2013.