IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 May 2024, accepted 25 July 2024, date of publication 5 August 2024, date of current version 14 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3438538

RESEARCH ARTICLE

A Method for DDoS Attacks Prevention Using SDN and NFV

MOHAMMAD JAVAD SHAYEGAN AND AMIRREZA DAMGHANIAN
Department of Computer Engineering, University of Science and Culture, Tehran 1461968151, Iran
Corresponding author: Mohammad Javad Shayegan (shayegan@usc.ac.ir)

ABSTRACT Distributed Denial-of-Service (DDoS) attacks are among the most common security attacks in enterprise networks. DDoS attacks are designed to disrupt networks by sending many false requests. With the introduction of Network Functions Virtualization (NFV), a new paradigm has been created for network management and design. The NFV architecture allows network functions to be defined dquite dynamically. A dynamic definition of network functions will provide the most effective support for organizational environments. This research aims to prevent DDoS attacks using NFV and Software-Defined Networking(SDN) platforms. Moving Target Defense (MTD) is used in this research to alter the routing and location of particular detection packets in the network. This MTD technique effectively hinders attackers from targeting real network topologies. A significant innovation introduced in this research is the selection of MTD types based on the processing resources of overlay networks. The results indicate that the proposed method will save these resources and reduce the time required to check network packets.

INDEX TERMS Network functions virtualization (NFV), virtualization of network functions (VNF), network security functions (NSF), virtualization, denial of service attack, moving target defense (MTD), DDos attacks.

I. INTRODUCTION

Today's companies range from small operations to multinational corporations, whose main priority is secure communication. Security is a critical prerequisite related to the service type, infrastructure (hardware and software), extent, and scale. DDoS attacks are network denial-of-service attacks created to disturb services. This attack occurs when the desired service becomes unavailable and disrupted. Despite decades of research, defending against DDoS attacks remains exceedingly challenging [1], [2]. These attacks are orchestrated based on the ingenuity of the attackers [3]. Attackers sometimes utilize the Internet of Things (IoT) for low-rate attacks [4].

As DDoS attack techniques evolve, the defense mechanisms for protecting against such attacks have also advanced significantly. Virtualization has played a pivotal role in this evolution, facilitating the implementation of numerous defense methods by transitioning from hardware-based to software-based platforms. The aim of this paper is to present a method for preventing DDoS attacks using Software-Defined Networking (SDN) and Network Functions Virtualization (NFV).

NFV introduces a novel approach to infrastructure design and network configuration. A NFV may contain one or more Virtual Machines (VMs) running different software and processes. The versatility of NFV allows for implementation across diverse conditions, presenting a new and economically advantageous prospect for enhancing network security [5]. An NFV network is implemented using the software network features, creating beneficial conditions and optimizing the network resources [6].

Furthermore, SDN technology enables automation and programmability by separating the control and management aspects of the network architecture. OpenFlow is the communication protocol that allows a SDN controller to manage the forwarding behavior of network devices directly, enabling flexible network configuration and dynamic traffic control.

SDN and NFV has incorporated advanced capabilities within its platform to detect and mitigate DDoS

The associate editor coordinating the review of this manuscript and approving it for publication was Alessio Giorgetti.

attacks [7], [8]. The convergence of SDN and NFV establishes a network built, operated, and managed entirely through software. SDN and NFV technology have provided various defense methods, one of which is Moving Target Defense (MTD) to increase the uncertainty and apparent complexity for attackers. Regarding network security components, we can refer to the virtual shadow network, a general term for providing various fake responses (honeypots) for security purposes. MTD mechanisms change the configuration and structure of networks both during and before an attack, making it very difficult for attackers to identify the actual network topology for a potential DDoS attack. The MTD strategy alters the routes for specific detection packets, preventing attackers from identifying the network topology and launching a potential DDoS attack.

Prior research has primarily assessed the effectiveness of MTD strategies based on their ability to prevent attacks. These evaluations often overlook the impact of MTD on overlay network infrastructure and processing resource consumption.

This research proposes a selective MTD model that dynamically adjusts defense mechanisms based on the available bandwidth of virtual machines (VMs) within the overlay network and the type of MTD strategy employed. Our model prioritizes the free bandwidth of VMs as a key criterion. Systems with greater bandwidth availability can accommodate more resource-intensive MTD techniques, such as frequent IP address changes, employ decoy servers and minimizing potential delays for clients. The simulated attack operates at the network layer and disturbs the server by sending packets. Additionally, network forensics is important in strengthening network security and identifying attacks on private servers. A virtual name collector point is used for traffic processing, network forensics, NFV hosting, and calculation of possible routes. In this research, the Virtual name Collector Point (VCP) serves as the executive server.

In summary, this paper's key contribution is a MTD method designed specifically for SDN and NFV environments. This approach prioritizes minimizing processing resource consumption, thereby reducing delays associated with defense activation. By leveraging real-time resource state information, the MTD response dynamically adapts, leading to a significant reduction in system overhead.

II. RELATED WORKS

Previous articles have traditionally discussed network defense and security in the context of DDoS attacks. These studies have considered virtualization criteria and software-defined networks. Various research has investigated methods to uphold security and advance defense mechanisms utilizing NFV and SDN. In [9], the authors investigated methods for DDoS attack defense utilizing NFV and SDN. In [10], a robust security framework was introduced for threats in 5G networks. The presented algorithm employs entropy to classify suspicious packets as normal or malicious based on their characteristics. Chowdhary et al. [11] introduced a framework designed to augment the complexity faced by potential attackers seeking to exploit vulnerabilities within the cloud network. The Moving Target Defense Assisted Security Framework (MASON) method systematically assesses the influence of moving target defense on intrusion detection system alerts, employing a threat score metric for comprehensive analysis.

In [12], Liu et al. proposed a novel DDoS defense algorithm based on NFV using a fuzzy system and Virtual Private Network (VPN) to detect and mitigate DDoS attacks effectively. The proposed method dynamically reroutes suspicious traffic and disconnects it from the network. Aydeger et al. introduced MTD mechanisms in [6], which dynamically alter the network's structure and configuration, making it more challenging for attackers to identify and exploit vulnerabilities. These mechanisms respond to attacks in real-time and proactively protect against potential attacks. Additionally, they introduced various network forensic mechanisms to identify the source and type of attacks, enabling a more comprehensive defense strategy. Rawski [5] presented the concept of MTD using topology mutation. This method involves identifying and characterizing hosts within the network, obtaining their topology information, and dynamically altering the network's structure to disrupt attackers' attempts to exploit known vulnerabilities. Singh et al. [13] presented ARDefense, a novel model that leverages NFV to mitigate DDoS attacks. ARDefense effectively defends online services such as websites by virtualizing network functions and implementing specific algorithms. Bringhenti et al. [14] introduced a novel automated approach to determine the optimal layout, placement, and configuration of virtual firewalls based on a set of predefined security requirements. Their method sought to optimize the placement of firewalls for enhanced network security and threat mitigation. Alhebaishi [15] proposed a concept that utilizes a virtual network architecture with dynamically configurable virtual segments and dynamically rearranges the network's logical structure. This approach strives to increase the complexity for attackers to identify and exploit network vulnerabilities by constantly altering the network's topology. The goal was to thwart attackers' attempts to pinpoint target systems and diminish their overall efficacy in launching attacks.

A comprehensive survey of MTD techniques, their key classifications, design dimensions, and attack behaviors considering existing moving target defense approaches was conducted by Cho et al. [16]. In [17], Bulbul and Fischer proposed a DDoS attack mitigation plan leveraging a machine learning algorithm to uncover DDoS attack patterns. Chen [18] discussed the architecture and detailed design of SDNShield, a defense system to counter DDoS attacks at the data control layer. SDNShield is a linear defense system coordinated by the SDN controller. Agrawal et al. [19] presented an algorithm that delegates network control layer routing decisions and oversees the entire network utilizing a centralized network controller. Rangisetti et al. [20] discussed Address Resolution Protocol (ARP) spoofing in cloud, fog, or hybrid platforms employing software-defined networking. Torquato and Vieira [21] employed a moving target defense (MTD) method involving virtual machine (VM) migration to counter existing denial-of-service (DoS) attacks and neutralize or defend against further attacks. Their methodology evaluated VM migrations' timing as a key element of the MTD strategy. In [22], Valdovinos et al. presented a novel network paradigm by introducing software-defined networking, which can potentially overcome the limitations of current switching networks by decoupling the control and data planes.

Significant research has focused on scaling control plane resources in SDNs. Abdulqadder et al. [23] proposed a hierarchical distribution of the control layer, both physically and logically, to improve utilization and scalability. DDoS-specific defense mechanisms have also been explored, including the IP-rule integrated system by Dimolianis et al. [24] to mitigate attacks. Attack success probability modeling in Infrastructure as a Service (IaaS) cloud settings utilizing virtual machine migration [25] is another approach presented. Nguyen et al. [26] characterized DDoS taxonomy across traditional, semi-SDN, and fully virtualized networks to facilitate analytical attack simulations. Alavizadeh et al. [27] introduced a heterogeneous dynamic defense system that enhances security by dynamically altering attack surfaces.

Emerging decentralized technologies offer promising capabilities for DDoS defense. Shakil et al. [28] combined blockchain with cryptography to develop an intermediary-free solution that integrates security through a cryptographic algorithm in a trustless manner. In contrast, Balarezo and Wang [29] discussed non-moving target defense-based denial-of-service strategies for both cloud and non-cloud infrastructures.

Machine learning and traffic fingerprinting have enabled detection advancements. Roshani [30] detected volumetric DDoS attacks using a hybrid machine learning model with pre-training. Jiang et al. [31] proposed BSD-Guard, a scalable blockchain-based intrusion detection and prevention system intended to protect software-defined networks from DDoS campaigns. Additional detection mechanisms have leveraged traffic fingerprints. Rios et al. [32] categorizes low-rate DDoS attack types and subsequently halts or reduces incidents based on fingerprints. The security framework SFCSA [33] implements path selection as a Markov decision process, employs reward systems for accurate malicious traffic inference, and demonstrates method effectiveness across various scenarios.

III. RESEARCH METHOD

The main aspect of the research methodology involves selecting and altering Moving Target Defense (MTD) strategies. This research derives its core defensive paradigm from the ancient Greek stratagem of the ''Shell Game,'' a tactic predicated on misdirection and deception. Building upon this foundational concept, we propose a novel network defense strategy that effectively thwarts attacks without relying solely on vulnerability patching or inherent hardware and software limitations. A thorough examination of the model proposed in [6] reveals a need for more consideration for system bandwidth and overhead. We can significantly enhance network defenses by modifying the strategies employed in overlay networks and replacing random algorithms with measurement-based methods.

Figure 1 provides an overview of the MTD components constituting the core research focus areas. This figure comprises three components: the SDN switch, VCP, and SDN controller. These components operate respectively within the network layer, application layer, and control layer.

The system follows a multi-stage approach to protect against attacks. First, upon receiving a packet, the system checks the route to determine if the destination belongs to the Virtual Collector Point (VCP). If so, the packet is routed accordingly. Subsequently, all packets are passed through a firewall and pertinent information is extracted and stored in a database (watchlist) for further analysis. Packets that are deemed benign, based on the absence of suspicious attributes, ar then forwarded to their intended destinations, such as websites or servers.

Following the selection of a strategy (as detailed in Section A) the packet in VCP undergoes examination by the forensic rules integrated into the VCP machine. Subsequently, the NFV Orchestrator grants authorization to regular packets. Ultimately, upon traversing through VCP, these normal packets facilitate the establishment of an optimal path or modifications to the network topology.

A. MOVING TARGET DEFENSE STRATEGY

Considering the computing resources of the involved virtual machines, we can dynamically mutate packets on the least occupied virtual machine. This enables efficient processing of larger traffic volumes, reducing latency and improving system efficiency. Subsequently, we can initiate a decision-making process based on the number of times the packet traverses the controller. It is discarded if the packet's review count exceeds a predefined limit. Conversely, if the count falls below the limit, it is routed to the moving target defense section. This section makes decisions based on the overhead of the Virtual Shadow Host (VSH). This approach dynamically adapts in the network based on traffic conditions and system resources to defend against DDoS attacks.

Also, this method minimizes disruption to normal traffic. replacing random algorithms with measurement-based methods, we can significantly enhance network defenses. In other words, if the available bandwidth of a VSH falls below a defined threshold, incoming packets undergo mutation and re-entry into the deception network for additional scrutiny. When the bandwidth conditions are normal, the packets continue on their path through the deception networks for forensic evaluation using the method performed in [34].

B. FINAL IMPLEMENTATION

After evaluating existing controllers and their capabilities, Floodlight (version 1.2 with OpenFlow 1.1) was selected

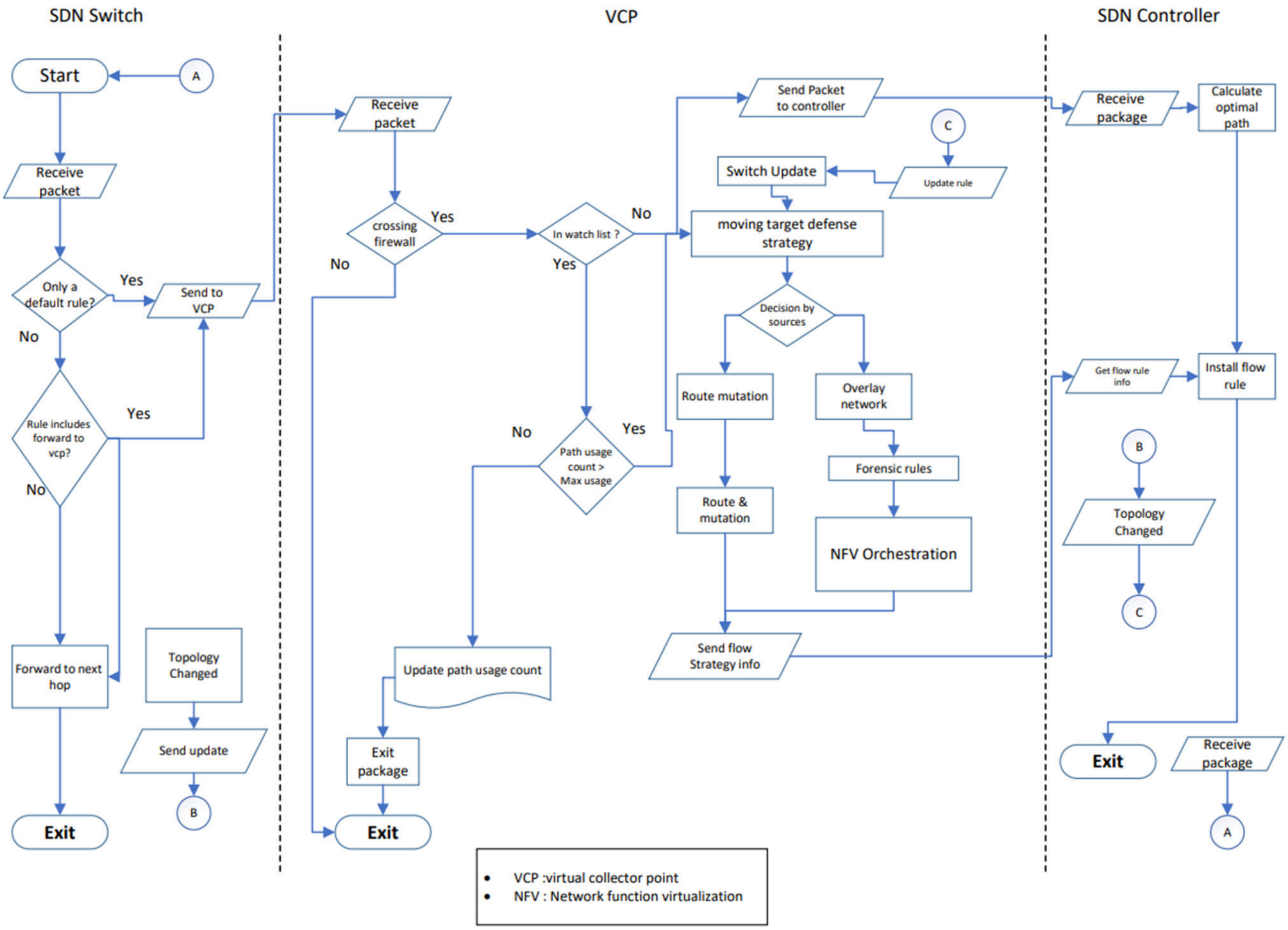


FIGURE 1. Flowchart for moving target defense management.

as the controller managed on the Virtual Collector Point server. This Java-based controller maintains all network rules and provides the underlying infrastructure necessary instructions to direct traffic. Additionally, the controller supports functional REST APIs to facilitate programming. Mininet version 2.3.0d6 was utilized to simulate the desired topology, which is comprised of switches, bots, and the main server. This simulator allows us to replicate virtual hosts and network topology. Both are based on Linux and employ OpenFlow technology [35]. Specifically, the virtual testbed comprised 11 hosts, four switches, and one controller instantiated within the Mininet environment for experimental purposes.

Upon initial connection of a switch to the controller, Floodlight defaults to clearing all tables. Subsequently, all routing and firewall roles are stored in the flow table. Additionally, the optimal network path is determined based on Floodlight's default routing protocol (RIP) for packets. One of the eleven hosts created assumes the role of a web server for defensive purposes. This web server is established using the ''Python SimpleHTTPServer'' module. The Floodlight firewall possesses an access control list utilized for watchlist functionality, constituting a stateless firewall.

While this approach is lightweight and efficient in creating these nodes, it cannot generate virtual machines operating independently from each other. Another point that The hosts' configurations are not retained when they are powered off. A key feature within Mininet is its ability to establish a connection with the Floodlight controller, facilitating dynamic rule exchange for enhanced network control and flexibility. Further, Mininet enabled direct Floodlight controller integration for rule exchanges. A script was developed using Python version 3 to execute a DDoS attack targeting websites hosted on one of the Mininet hosts and sending fake traffic to the servers. As soon as the script receives the desired port number and packet rate for the website, it initiates the attack. In this approach, the links for attack packets undergo route mutation, making it challenging for attackers to identify the network's actual topology and launch a DoS attack. Simultaneously, it enables the defender to gather attacker information through forensics. Consequently, the adopted approach effectively reduces latency and resource consumption.

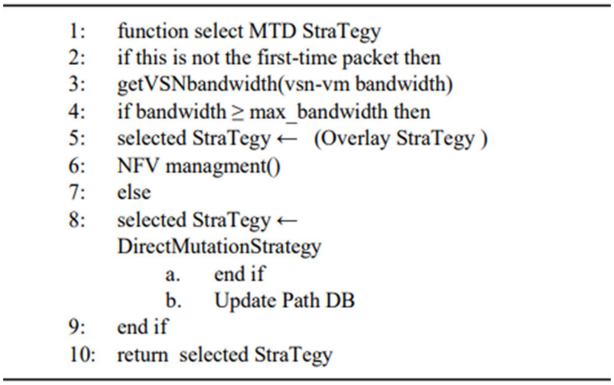
The modifications to the moving target defense in the strategy section are delineated in the two pseudo-codes section. Figure 2 presents a pseudocode detailing MTD's strategy selection process. The initial lines (1-4) handle packet reception and bandwidth analysis, followed by the core strategy selection logic (lines 5-12). In case the bandwidth exceeds the user-defined threshold, the algorithm triggers the introduction of virtual network functions (VNFs), effectively isolating the compromised traffic. Conversely, if the bandwidth remains within acceptable limits, the algorithm initiates route mutation, dynamically altering packet routing and updating the routing database accordingly.

Figure 3 depicts the pseudo-code for packet entry monitoring, an ongoing process that continues until the packet reaches the strategy selection section. This algorithm identifies entities requiring watchlist inclusion, such as srcIP (source IP) and dstIP (destination IP), and determines the appropriate response based on packet frequency. The pseudo-code receives packet-specific variables, including SwitchID, srcIP, dstIP, protocol, and maxUsage (maximum watch count). Based on the maxUsage value, a decision is made to either admit or discard the packet. For route mutation instances, the algorithm transmits the values of srcIP, dstIP, protocol, and MTDStrategy to the router for immediate mutation execution. Additionally, the triggeredUpdate value is updated, and the new value is recorded in the database. The provided pseudo-codes articulate a well-structured framework for implementing dynamic MTD strategies, adeptly countering emerging network threats.

To enhance comprehension of the operational methodology of the proposed approach, Figure 4 has showed the scenario of transmitting a normal packet across the three layers. The executed components are depicted in this figure. Initially, a random host sends a request to access a web server. The packet enters the VCP after routing through the network layer. After passing through the installed firewall (Floodlight controller) and the local database, it reaches the section responsible for examining defensive strategies against the moving target. In this section, depending on the load of authorized virtual machines, the packet either undergoes route mutation or proceeds to the NFV machine section. Ultimately, with the assistance of Open vSwitch within the controller, the packet reaches the web server host.

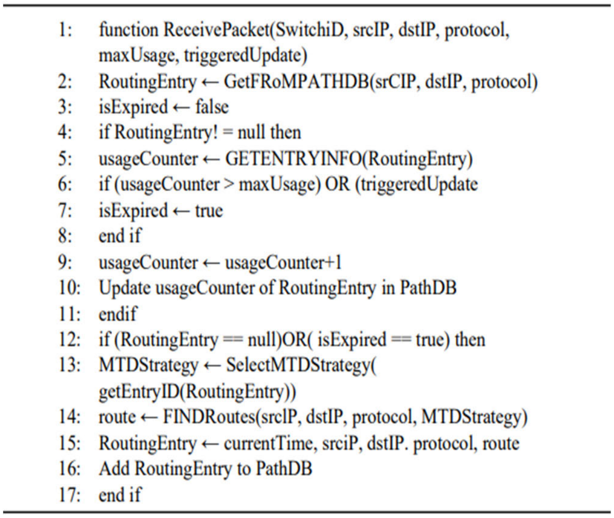
IV. FINDINGS

For this research, a virtual machine with the following specifications was utilized as the testing platform: 20 processor cores, 32 GB RAM, and running the Ubuntu operating system. Floodlight was chosen as the primary network controller, requiring Java prerequisites for its operation. The controller's compatibility with various physical switches, such as Dell Z9000, Arista 7050, and HP 3500, demonstrates its feasibility in real-world deployments. The performance of Open vSwitch on SDN and NFV platforms, along with the presentation of two new frameworks for routing overload management, has been investigated in previous articles [36].



```
1:  function select MTD StraTegy
2:  if this is not the first-time packet then
3:  getVSNbandwidth(vsn-vm bandwidth)
4:  if bandwidth ≥ max_bandwidth then
5:  selected StraTegy ←  (Overlay StraTegy )
6:  NFV managment()
7:  else
8:  selected StraTegy ←
    DirectMutationStrategy
        a.  end if
        b.  Update Path DB
9:  end if
10: return  selected StraTegy
```

FIGURE 2. Choosing moving target defense strategy.



```
function ReceivePacket(SwitchiD, srcIP, dstIP, protocol,
maxUsage, triggeredUpdate)
RoutingEntry ← GetFRoMPATHDB(srCIP, dstIP, protocol)
isExpired ← false
if RoutingEntry! = null then
usageCounter ← GETENTRYINFO(RoutingEntry)
if (usageCounter > maxUsage) OR (triggeredUpdate
isExpired ← true
end if
usageCounter ← usageCounter+1
Update usageCounter of RoutingEntry in PathDB
endif
if (RoutingEntry == null)OR( isExpired == true) then
MTDStrategy ← SelectMTDStrategy(
getEntryID(RoutingEntry))
route ← FINDRoutes(srcIP, dstIP, protocol, MTDStrategy)
RoutingEntry ← currentTime, srciP, dstIP. protocol, route
Add RoutingEntry to PathDB
end if
```

FIGURE 3. Packet received on the defense system.

The sFlow-RT tool (version 3.0.1425) is also used for bandwidth profiling, performance, and monitoring in SDN and NFV connections [37]. The sFlow-RT analysis engine operates as a continuous measurement system, receiving information from agents on network devices, hosts, and applications. Raw data is transformed into usable variables accessible through an API. The variable used in the SFlow-RT graphs is mn_flow, which displays the maximum number of flows passing through. This variable uses sFlow-RT to display the number of packets during an attack and after. The graph generated in the defense execution shows a downward trend. Figure 5 illustrates the number of flows in a normal request without an attack.

The attack scenario involved specifying the IP address of the web server, sending 4096 packets, and targeting the destination port HTTP. After the execution of the attack, the configured firewall in the controller is employed to mitigate and prevent further instances of the attack. Floodlight enables the definition of an access control list that functions as a firewall, shared among all switches. Figure 6 demonstrates the significant reduction in attacks after implementing firewall controls. In this graph, the attack peak reached 15,000 packets per minute, ultimately neutralizing the attacked.

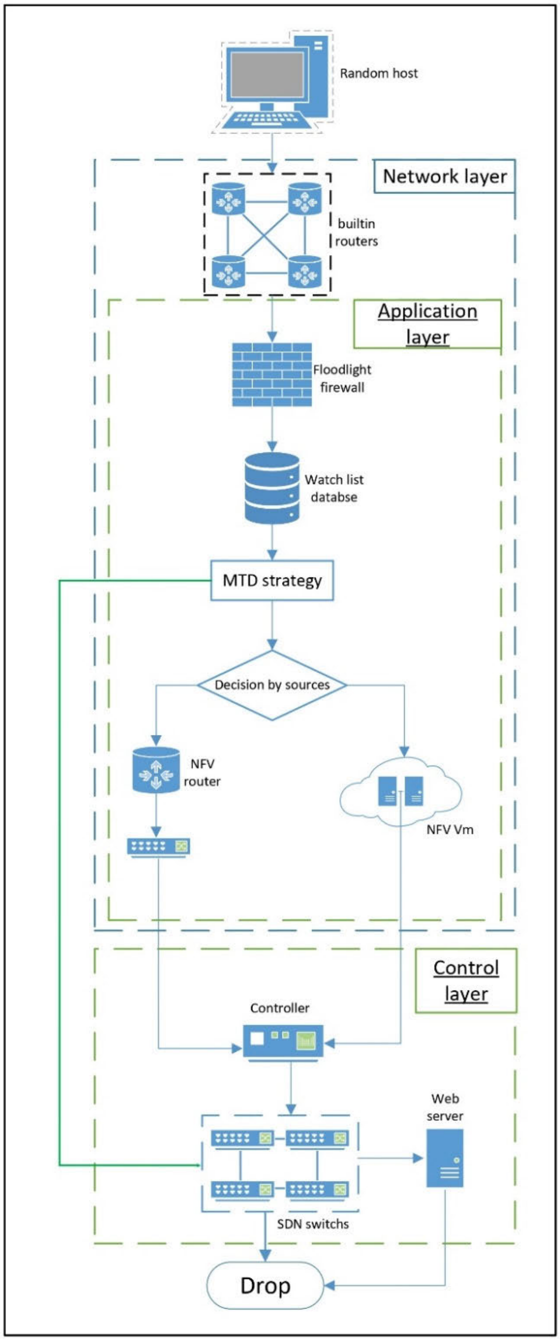


FIGURE 4. The scenario for transmitting a normal packet in the proposed method.

A. SMART DEFENSE BY MOVING TARGET DEFENSE STRATEGY

In the succeeding phase, once the controller, indexing engine, and attacks are executed, the implementation of moving target defense becomes feasible. The firewall was programmed utilizing the React API to construct intelligent defense capabilities by dispatching API commands to the controller and analysis engine. The user sets a threshold value for permissible traffic, identifying attacks and alerting the analysis engine.

The engine extracts the IP addresses of the attackers and relays them to the controller's firewall for defense. Upon receiving attack parameters from the controller, the firewall performs data cleansing and successfully thwarts the attack, denying the attacker access to the desired web service. By the moving target defense approach, the maximum flow algorithm is employed in the subsequent phase to execute route mutation, prioritizing different routes and effecting alterations. To facilitate the algorithm's functionality, two parameters, namely IP address and Time to Live (TTL), must be incorporated into the switches to serve the role of layer three. Using OpenFlow controllers, functions spanning layers 1 to 4 can be implemented [38].

Consequently, by integrating routers and NFV network functions, bandwidth can be effectively monitored within the controller. If the bandwidth falls below the user response threshold, route mutation is triggered; otherwise, it navigates through overlay networks to reach the ultimate server.

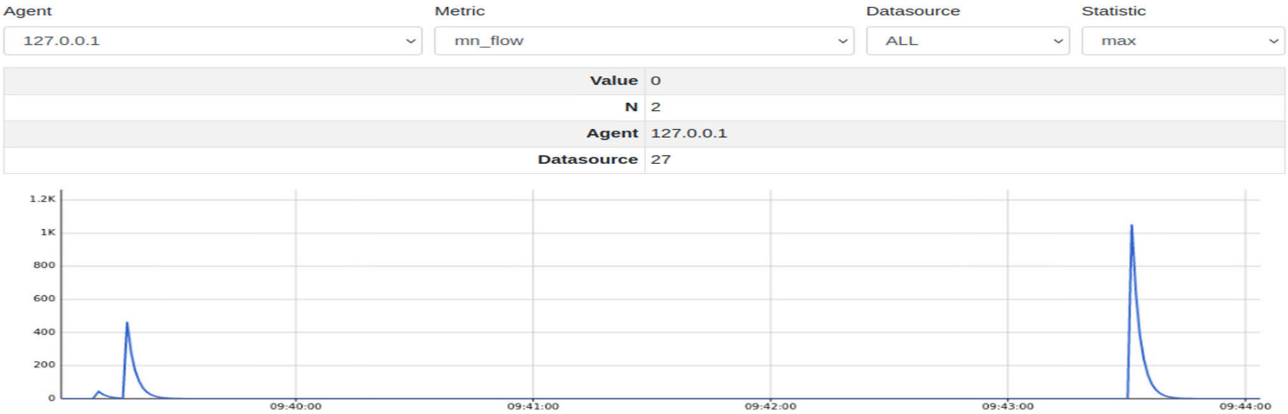
V. EVALUATION

The proposed defense technique effectively thwarts DDoS attacks, demonstrating its superiority to previously reported strategies. While existing methods such as [6] rely on factors like mutation probability, time limit, and alternative probability to determine MTD strategies, the proposed method prioritizes minimizing the overlay network system's overhead, allowing freed resources to handle normal traffic more efficiently. This approach also reduces storage and processing requirements, enabling healthy packets to reach their destination faster than previous mutation methods. An innovation in the proposed method involves replacing the Russian roulette algorithm with the suggested resource usage method. The controller detects potential attackers by classifying transmitted packets, and preemptive measures are taken to prevent the attack.

The evaluation of the proposed method is conducted in two sections. The first section assesses the effectiveness of the proposed method in preventing DDoS attacks, as demonstrated in Figure 6. Figure 6 illustrates a significant reduction in attacks following the implementation of firewall controls. As depicted in this graph, at one point, the maximum attack peaked at 15,000 packets per minute, ultimately being neutralized.

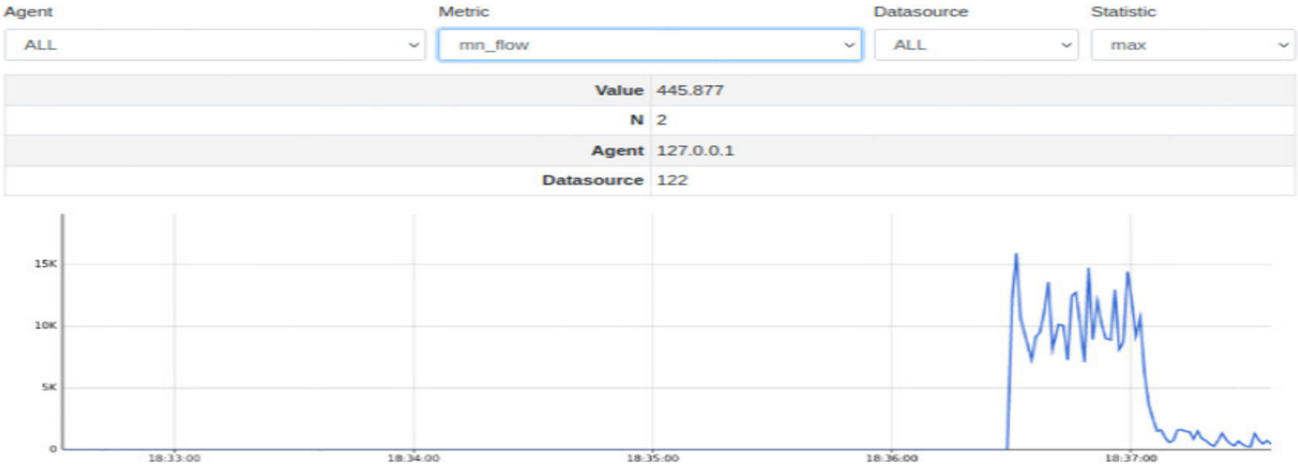
The second section evaluates the success rate of the proposed method in reducing processor overhead. A significant observation in the findings is the decrease of virtual machine processing resources by 10 to 20 percent in the proposed defense framework. In contrast, inspecting all incoming traffic would elevate this figure to over 90% of the processor's capacity. The system overhead, driven by the controller software and forensic properties, peaks when the system is fully loaded.

During system updates to the route, topology information may be stored in RAM, or alternatively, Link Layer Discovery Protocol (LLDP) packets may be employed to update network information. The total route mutation is directly related to the


Agent: 127.0.0.1
Metric: mn_flow
Datasource: ALL
Statistic: max

Value	0
N	2
Agent	127.0.0.1
Datasource	27

FIGURE 5. Packet received on the defense system.


Agent: ALL
Metric: mn_flow
Datasource: ALL
Statistic: max

Value	445.877
N	2
Agent	127.0.0.1
Datasource	122

FIGURE 6. Number of blocked packets by method.

increase in the number of routers. The complexity of the route mutation execution time is O(n* log(n)). Route mutation only inflicts a few milliseconds of delay. The database records contain input ID (2 bytes), source IP (4 bytes), destination IP (4 bytes), protocol type (1 byte), and time (7 bytes). Furthermore, the packet route size directly correlates with the length of the route mutation. A designated value of 1 byte indicates the probability of mutation within the MTD strategy.

VI. DISCUSSION

This study introduces three distinct moving target defense strategies for implementation, one operating at layer 7 and the others at layers 3 and 4, thereby enhancing overall defensive capability. In the proposed method, attackers face a continuously changing and random perspective of the underlying system. The proactive nature of this process, which thwarts attacks at the detection phase, proves time-consuming for attackers while imposing minimal overhead on the system.

Through route mutation, a divergent route is established within the packet's inbound trajectory, rendering it untraceable by the attacker. According to [11], the probability of choosing a decision through Russian Roulette is fixed at 33% in the decision-making process of moving target defense. However, the noteworthy advantage of the proposed method lies in its capacity to diminish the overlay network system's overhead by considering available system resources. This approach demonstrably reduces the likelihood of web

TABLE 1. Comparison between current study and previous work [6].

Parameters	Current Study	Previous work [6]
MTD Type	Based On sources	Randomly
Required processing resources	Only Ovlerlay networks	Use Shadow and ovlerlay networks
MTD Strategies	Ovlerlay networks	VSH and ovlerlay networks
Ovlerlay networks	Optimal use of resources	Only Change the route
Databases Used	Use a shared database	At least 2

server unavailability to less than 1%, representing a significant benefit in the domain of DDoS attack mitigation research.

An additional benefit of the proposed approach is storage space reduction by removing the tracking database. The entry in the tracking database comprises 39 bytes. Due to the prospect of memory overhead with increasing hops, this variable space has been completely supplanted. Table 1 demonstrates that comparing the proposed method to [6] reveals superior and reliable performance on certain criteria. However, the previous method retains MTD advantages.

VII. CONCLUSION

In this research, moving target defense techniques have been employed to mitigate DDoS attacks in Internet Service Providers (ISPs) by utilizing software network architecture and virtualizing network functions. This approach makes decisions grounded in resource considerations. It encompasses two dynamic defense strategies: the first strategy, route mutation, is implemented to obscure network topology information during the DDoS detection stage and divert the attacker away from the final target. The subsequent strategy facilitates the mutation of the route for malicious packets, redirecting them away from their ultimate destination, which is an important principle of moving target defense. Another strategy involves using covert networks for mutation, ensuring constant server migration, and preventing attackers from identifying the real server.

Notably, the implementation of overlay networks regarding resources was not addressed in [6]. This study endeavors to implement the defense method by assessing bandwidth and identifying the minimum resources required for defense through the virtualization of network functions. A comprehensive simulation of various aspects of attack and defense was undertaken in this study, intending to serve as a foundational framework for future research in the domain of moving target defense. The implementation and testing of this research were conducted on a small scale, which represents one of the primary limitations of this study. A real-world implementation within an ISP environment could provide a more comprehensive evaluation of the proposed method's effectiveness.

In future endeavors, incorporating the fault tolerance system in the controller, as introduced in [39], could mitigate potential failure points in the control layer. Adding fault tolerance would render attackers incapable of turning off the control layer, ensuring uninterrupted access to the desired defense services. Furthermore, the utilization of service function chaining [40] empowers operators to design Virtual Network Function (VNF) functions dynamically, catering to the specific needs of their clientele.

REFERENCES

[1] A. S. M. Rizvi, J. Mirkovic, J. Heidemann, W. Hardaker, and R. Story, "Defending root DNS servers against DDoS using layered defenses," in *Proc. 15th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2023, pp. 513–521, doi: 10.1109/COMSNETS56262.2023.10041415.

[2] *Akamai Blog | 2021: Volumetric DDoS Attacks Rising Fast*. Accessed: Apr. 20, 2023. [Online]. Available: https://www.akamai.com/blog/security/2021-volumetric-ddos-attacks-rising-fast

[3] *Azure DDoS Protection-2021 Q1 and Q2 DDoS Attack Trends | Azure Blog and Updates | Microsoft Azure*. Accessed: Apr. 20, 2023. [Online]. Available: https://azure.microsoft.com/en-us/blog/azure-ddos-protection-2021-q1-and-q2-ddos-attack-trends/

[4] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "IoT malicious traffic identification using wrapper-based feature selection mechanisms," *Comput. Secur.*, vol. 94, Jul. 2020, Art. no. 101863, doi: 10.1016/j.cose.2020.101863.

[5] M. Rawski, "Network topology mutation as moving target defense for corporate networks," *Int. J. Electron. Telecommun.*, vol. 65, no. 4, pp. 571–577, Oct. 2019. Accessed: Sep. 11, 2022. [Online]. Available: http://www.ijet.pl/index.php/ijet/article/view/10.24425-ijet.2019.129814

[6] A. Aydeger, N. Saputro, and K. Akkaya, "A moving target defense and network forensics framework for ISP networks using SDN and NFV," *Future Gener. Comput. Syst.*, vol. 94, pp. 496–509, May 2019, doi: 10.1016/j.future.2018.11.045.

[7] A. A. Sadi, M. Savi, D. Berardi, A. Melis, M. Prandini, and F. Callegati, "Real-time pipeline reconfiguration of P4 programmable switches to efficiently detect and mitigate DDoS attacks," in *Proc. 26th Conf. Innov. Clouds, Internet Netw. Workshops (ICIN)*, Mar. 2023, pp. 21–23, doi: 10.1109/ICIN56760.2023.10073501.

[8] Z. A. Bhuiyan, S. Islam, M. M. Islam, A. B. M. A. Ullah, F. Naz, and M. S. Rahman, "On the (in)Security of the control plane of SDN architecture: A survey," *IEEE Access*, vol. 11, pp. 91550–91582, 2023, doi: 10.1109/access.2023.3307467.

[9] M. J. Shayegan and A. Damghanian, "A review of methods to prevent DDOS attacks using NFV and SDN," in *Proc. 9th Int. Conf. Web Res. (ICWR)*, May 2023, pp. 346–355, doi: 10.1109/icwr57742.2023.10139112.

[10] I. H. Abdulqadder, D. Zou, I. T. Aziz, B. Yuan, and W. Dai, "Deployment of robust security scheme in SDN based 5G network over NFV enabled cloud environment," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 2, pp. 866–877, Apr. 2021, doi: 10.1109/TETC.2018.2879714.

[11] A. Chowdhary, A. Alshamrani, D. Huang, and H. Liang, "MTD analysis and evaluation framework in software defined network (MASON)," in *Proc. ACM Int. Workshop Secur. Softw. Defined Netw. Netw. Function Virtualization*, Mar. 2018, pp. 43–48, doi: 10.1145/3180465.3180473.

[12] C. C. Liu, B. S. Huang, C. W. Tseng, Y. T. Yang, and L. Der Chou, "SDN/NFV-based moving target DDoS defense mechanism," in *Proc. Int. Conf. Reliable Inf. Commun. Technol.*, vol. 843, 2019, pp. 548–556, doi: 10.1007/978-3-319-99007-1_51.

[13] A. K. Singh, R. K. Jaiswal, K. Abdukodir, and A. Muthanna, "ARDefense: DDoS detection and prevention using NFV and SDN," in *Proc. 12th Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops (ICUMT)*, Oct. 2020, pp. 236–241, doi: 10.1109/ICUMT51630.2020.9222443.

[14] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, and J. Yusupov, "Automated optimal firewall orchestration and configuration in virtualized networks," in *Proc. NOMS IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2020, pp. 1–7, doi: 10.1109/NOMS47738.2020.9110402.

[15] N. Alhebaishi, L. Wang, and S. Jajodia, "Modeling and mitigating security threats in network functions virtualization (NFV)," in *Proc. IFIP Annu. Conf. Data Appl. Secur. Privacy*, vol. 12122, 2020, pp. 3–23, doi: 10.1007/978-3-319-41483-6_21.

[16] J.-H. Cho, D. P. Sharma, H. Alavizadeh, S. Yoon, N. Ben-Asher, T. J. Moore, D. S. Kim, H. Lim, and F. F. Nelson, "Toward proactive, adaptive defense: A survey on moving target defense," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 709–745, 1st Quart., 2020, doi: 10.1109/COMST.2019.2963791.

[17] N. S. Bülbül and M. Fischer, "SDN/NFV-based DDoS mitigation via pushback," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6, doi: 10.1109/ICC40277.2020.9148717.

[18] K.-Y. Chen, S. Liu, Y. Xu, I. K. Siddhrau, S. Zhou, Z. Guo, and H. J. Chao, "SDNShield: NFV-based defense framework against DDoS attacks on SDN control plane," *IEEE/ACM Trans. Netw.*, vol. 30, no. 1, pp. 1–17, Feb. 2022, doi: 10.1109/TNET.2021.3105187.

[19] N. Agrawal and S. Tapaswi, "An SDN-assisted defense mechanism for the shrew DDoS attack in a cloud computing environment," *J. Netw. Syst. Manage.*, vol. 29, no. 2, pp. 1–28, Jan. 2021, doi: 10.1007/s10922-020-09580-7.

[20] A. K. Rangisetti, R. Dwivedi, and P. Singh, "Denial of ARP spoofing in SDN and NFV enabled cloud-fog-edge platforms," *Cluster Comput.*, vol. 24, no. 4, pp. 3147–3172, Jun. 2021, doi: 10.1007/s10586-021-03328-x.

[21] M. Torquato and M. Vieira, "VM migration scheduling as moving target defense against memory DoS attacks: An empirical study," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Sep. 2021, pp. 1–6, doi: 10.1109/ISCC53001.2021.9631397.

[22] I. A. Valdovinos, J. A. Pérez-Díaz, K.-K.-R. Choo, and J. F. Botero, "Emerging DDoS attack detection and mitigation strategies in software-defined networks: Taxonomy, challenges and future directions," *J. Netw. Comput. Appl.*, vol. 187, Aug. 2021, Art. no. 103093, doi: 10.1016/j.jnca.2021.103093.

[23] I. H. Abdulqadder, S. Zhou, I. T. Aziz, D. Zou, X. Deng, and S. M. A. Akber, "An effective lightweight intrusion detection system with blockchain to mitigate attacks in SDN/NFV enabled cloud," in *Proc. 6th Int. Conf. Converg. Technol. (I2CT)*, Apr. 2021, pp. 1–8, doi: 10.1109/I2CT51068.2021.9417961.

[24] M. Dimolianis, A. Pavlidis, and V. Maglaris, "Signature-based traffic classification and mitigation for DDoS attacks using programmable network data planes," *IEEE Access*, vol. 9, pp. 113061–113076, 2021, doi: 10.1109/ACCESS.2021.3104115.

[25] M. Torquato, P. Maciel, and M. Vieira, "Analysis of VM migration scheduling as moving target defense against insider attacks," in *Proc. 36th Annu. ACM Symp. Appl. Comput.*, Mar. 2021, pp. 194–202, doi: 10.1145/3412841.3441899.

[26] M. Nguyen and S. Debroy, "Moving target defense-based denial-of-service mitigation in cloud environments: A survey," *Secur. Commun. Netw.*, vol. 2022, pp. 1–24, Mar. 2022, doi: 10.1155/2022/2223050.

[27] H. Alavizadeh, S. Aref, D. S. Kim, and J. Jang-Jaccard, "Evaluating the security and economic effects of moving target defense techniques on the cloud," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 4, pp. 1772–1788, Oct. 2022, doi: 10.1109/TETC.2022.3155272.

[28] M. Shakil, A. F. Y. Mohammed, R. Arul, A. K. Bashir, and J. K. Choi, "A novel dynamic framework to detect DDoS in SDN using metaheuristic clustering," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 3, Mar. 2022, Art. no. e3622, doi: 10.1002/ett.3622.

[29] J. F. Balarezo, S. Wang, K. G. Chavez, A. Al-Hourani, and S. Kandeepan, "A survey on DoS/DDoS attacks mathematical modelling for traditional, SDN and virtual networks," *Eng. Sci. Technol., Int. J.*, vol. 31, Jul. 2022, Art. no. 101065, doi: 10.1016/j.jestch.2021.09.011.

[30] M. Roshani and M. Nobakht, "HybridDAD: Detecting DDoS flooding attack using machine learning with programmable switches," in *Proc. 17th Int. Conf. Availability, Rel. Secur.*, Aug. 2022, pp. 1–11, doi: 10.1145/3538969.3538991.

[31] S. Jiang, L. Yang, X. Gao, Y. Zhou, T. Feng, Y. Song, K. Liu, and G. Cheng, "BSD-guard: A collaborative blockchain-based approach for detection and mitigation of SDN-targeted DDoS attacks," *Secur. Commun. Netw.*, vol. 2022, pp. 1–16, Apr. 2022, doi: 10.1155/2022/1608689.

[32] V. D. M. Rios, P. R. M. Inácio, D. Magoni, and M. M. Freire, "Detection and mitigation of low-rate Denial-of-Service attacks: A survey," *IEEE Access*, vol. 10, pp. 76648–76668, 2022, doi: 10.1109/ACCESS.2022.3191430.

[33] B. Alhijawi, S. Almajali, H. Elgala, H. B. Salameh, and M. Ayyash, "A survey on DoS/DDoS mitigation techniques in SDNs: Classification, comparison, solutions, testing tools and datasets," *Comput. Electr. Eng.*, vol. 99, Apr. 2022, Art. no. 107706, doi: 10.1016/j.compeleceng.2022.107706.

[34] M. Li, S. Deng, H. Zhou, and Y. Qin, "A path selection scheme for detecting malicious behavior based on deep reinforcement learning in SDN/NFV-enabled network," *Comput. Netw.*, vol. 236, Nov. 2023, Art. no. 110034, doi: 10.1016/j.comnet.2023.110034.

[35] A. Agarwal, R. Singh, and M. Khari, "Detection of DDOS attack using IDS mechanism: A review," in *Proc. 1st Int. Conf. Informat. (ICI)*, Apr. 2022, pp. 36–46, doi: 10.1109/ICI53355.2022.9786899.

[36] A. Montazerolghaem, "Softwarization and virtualization of VoIP networks," *J. Supercomput.*, vol. 78, no. 12, pp. 14471–14503, Apr. 2022, doi: 10.1007/s11227-022-04448-w.

[37] M. Wang, Y. Lu, and J. Qin, "Source-based defense against DDoS attacks in SDN based on sFlow and SOM," *IEEE Access*, vol. 10, pp. 2097–2116, 2022, doi: 10.1109/ACCESS.2021.3139511.

[38] S. H. Darekar, M. Z. Shaikh, and H. B. Kondke, "Performance evaluation of various open flow SDN controllers by addressing scalability metric based on multifarious topology design on software-defined networks: A comprehensive survey," in *Proc. 3rd Int. Conf. Intell. Comput., Inf. Control Syst.*, 2022, pp. 327–338, doi: 10.1007/978-981-16-7330-6_25.

[39] P. Valizadeh and A. Taghinezhad-Niar, "DDoS attacks detection in multi-controller based software defined network," in *Proc. 8th Int. Conf. Web Res. (ICWR)*, May 2022, pp. 34–39, doi: 10.1109/ICWR54782.2022.9786246.

[40] M. Pattaranantakul, C. Vorakulpipat, and T. Takahashi, "Service function chaining security survey: Addressing security challenges and threats," *Comput. Netw.*, vol. 221, Feb. 2023, Art. no. 109484, doi: 10.1016/j.comnet.2022.109484.

MOHAMMAD JAVAD SHAYEGAN received the B.S. and M.S. degrees in computer engineering from IAU Science and Research Branch, Iran, and the Ph.D. degree in information technology and multimedia system from Universiti Putra Malaysia. He is currently an Associate Professor with the Department of Computer Engineering, University of Science and Culture, Tehran, Iran. He is also the Founder of the Web Research Center, IEEE International Conference on Web Research, and *International Journal of Web Research* in Iran. His research interests include distributed systems and web research.

AMIRREZA DAMGHANIAN received the master's degree in computer engineering from the Department of Computer Engineering, University of Science and Culture (USC), Tehran, Iran, where he is currently pursuing the Ph.D. degree in computer engineering. His research interest includes computer networks and security.